



KOMPUTER SAPIENS

Revista de Divulgación de la Sociedad Mexicana de Inteligencia Artificial

Año 11
Volumen 2
Mayo-Agosto 2019

TUTORES INTELIGENTES: VALIDACIÓN, PERFILES Y AFECTIVIDAD

LA IA EN
AGRICULTURA,
CLIMA, COSMOLOGÍA,
PROPIEDAD
INTELLECTUAL Y
CONVERSIONES

APRENDIZAJE
PROFUNDO
PARA PLN

SEGUIDORES
DE LÍNEAS Y
DETECCIÓN DE
OBSTÁCULOS
MÓVILES





©Komputer Sapiens, Año XI Volumen II, marzo-agosto 2019, es una publicación cuatrimestral de la Sociedad Mexicana de Inteligencia Artificial, A.C., con domicilio en Ezequiel Montes 56 s/n, Fracc. los Pilares, Metepec, Edo. de México, C.P. 52159, México, <http://www.komputersapiens.org>, correo electrónico: editorial@komputersapiens.org, tel. +52 (833)357.48.20 ext. 3024, fax +52 (833) 215.85.44. Impresa por Sistemas y Diseños de México S.A.

de C.V., calle Aragón No. 190, colonia Álamos, delegación Benito Juárez, México D.F., C.P. 03400, México, se terminó de imprimir el 30 de agosto 2019, este número consta de 1000 ejemplares.

Reserva de derechos al uso exclusivo número 04-2009-111110040200-102 otorgado por el Instituto Nacional de Derechos de Autor. ISSN 2007-0691.

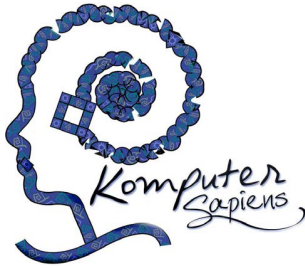
Los artículos y columnas firmados son responsabilidad exclusiva de los autores y no reflejan necesariamente los puntos de vista de la Sociedad Mexicana de Inteligencia Artificial. La mención de empresas o productos específicos en las páginas de Komputer Sapiens no implica su respaldo por la Sociedad Mexicana de Inteligencia Artificial.

Queda estrictamente prohibida la reproducción total o parcial por cualquier medio, de la información aquí contenida sin autorización por escrito de los editores.

Komputer Sapiens es una revista de divulgación en idioma español de temas relacionados con la inteligencia artificial. Creada en \LaTeX , con la clase **papertex** disponible en el repositorio *CTAN*: Comprehensive TeX Archive Network, <http://www.ctan.org/>

Indizada en el IRMDCT de CONACYT y en Latindex.

	Directorio SMIA	Directores Fundadores	Elisa Schaeffer
Presidente	Félix A. Castro Espinoza	Carlos Alberto Reyes García	Julio Berbel
Vicepresidente	Ildar Batyrshin	Ángel Kuri Morales	Ángel Kuri
Secretario	Oscar Herrera Alcántara		J. David Terán-Villanueva
Tesorero	María de Lourdes G. Martínez Villaseñor	Comité Editorial	Carlos Brizuela
Vocales:	Sofía Natalia Galicia Haro	Félix A. Castro Espinoza	Pastor López-Monroy
	Sabino Miranda Jiménez	Jesús Favela Vara	Aurora Torres
	Roberto A. Vázquez Espinoza de los Monteros	Sofía Natalia Galicia Haro	César Torres
	Hiram Erendin Ponce Espinosa	Miguel González Mendozao	Miguel Ángel Álvarez Carmona
	Francisco Viveros-Jiménez	Oscar Herrera Alcántara	Angélica Muñoz
	Noé Alejandro Castro Sánchez	Raúl Monroy Borja	Aurelio López-López
	Obdulia Pichardo Lagunas	Eduardo F. Morales Manzanares	Francisco Domínguez-Mota
	Omar Montaña Rivas	Leonardo Garrido Luna	Eric S. Tellez
	Antonio Marín Hernández	Carlos Alberto Reyes García	Carlos Jaime Barrios Hernández
	Néstor Velasco Bermeo	Angélica Muñoz Meléndez	Leonardo Romero
	Iris Iddaly Méndez Gurrola	Antonio Sánchez Aguilar	Juan Bekios
	Gustavo Arroyo Figueroa	Luis Enrique Sucar Succar	Carlos Lara-Álvarez
		Ángel Kuri Morales	María Del Pilar Gómez Gil
		José A. Martínez Flores	Leopoldo Morán
	Komputer Sapiens	Juan Manuel Ahuactzin Larios	Felix Calderón
Director general	Félix A. Castro Espinoza	Manuel Montes y Gómez	Edgar Tello
Editora en jefe	Karina Mariela Figueroa Mora	Ofelia Cervantes Villagómez	Enrique Sucar
Editores invitados	Jorge A. Ruiz Vanoye	Alexander Gelbukh	Manuel Mejía-Lavalle
	Ocotlán Díaz Parra	Grigori Sidorov	Alexa Ramírez
Editor científico	Héctor Gabriel Acosta Mesa	Laura Cruz Reyes	Carlos Monserrat Aranda
Editores asociados	Marco Antonio Aguirre Lam	Elisa Schaeffer	José Carranza
	Laura Cruz Reyes	Ramon Brena Pinero	Mario Graff Guerrero
Coordinadora de producción e-Tlakuilo	Viridiana Mena Gómez	Juan Humberto Sossa Azuela	Rosario Baltazar
	Jorge A. Ruiz-Vanoye		Aidé Maldonado
	Ocotlán Díaz-Parra	Árbitros	Diego Tlapa
Estado del IArte	Ma del Pilar Gómez Gil	Ramón Zatarain Cabada	Cuahtémoc Sánchez Ramírez
	Jorge Rafael Gutiérrez Pulido	Carlos Alberto Reyes García	René Navarro
Sakbe	Claudia G. Gómez Santillán	Julia Yazmín Arana Llanes	Aurelio López
	Laura Cruz Reyes	Antonio Camarena-Ibarrola	
	Laura Gómez Cruz	Edgar Chávez	
IA & Educación	María Yasmín Hernández Pérez	Yadira Quiñonez	
	María Lucía Barrón Estrada	Hugo Rodríguez Cortés Roberto	
Deskubriendo Konocimiento	J. Julieta Noguez Monroy	Munguia Silva Erick Rodríguez	
	Alejandro Guerra Hernández	C. Alberto Ochoa-Zezatti	
	Leonardo Garrido Luna	Luis Castro	
Asistencia técnica	Alan G. Aguirre Lam	César Cárdenas	
Corrección de estilo	Miguel Antonio Lupián Soto	Luis Valero Elizondo Leonel Vinicio	
	Gilberto Rivera Zárate	Morales Díaz	
	Marcela Quiroz Castellanos	Lucía Barrón	
	Ruth Esmeralda Barreda Guajardo	Ángel G. Andrade	
	Denisse Alvarado	Leonardo Trujillo	
	Héctor Fraire	José Antonio González Cervera	
	Guillermo de Jesús Hoyos Rivera	Juan Humberto Sossa Azuela	
	Susana Aurora Velasco Montiel	Juan José Flores	
Edición de imagen	Laura Gómez Cruz	Eduardo Morales	
Portada	Irene Morales Pagaza, Mopi Diseño		



Contenido

Optimización de una red convolucional para la predicción la velocidad del viento

Pág. 7 ⇒ por Hector Rodriguez-Rangel, Emmanuel F. Ramírez-Hernández, Luis A. Morales, Juan J. Flores y J. Armando Guerra

Una Aplicación de las Redes Neuronales Artificiales en la Cosmología

Pág. 12 ⇒ por Isidro Gómez Vargas, Ricardo Medel Esquivel, Ricardo García Salcedo y J. Alberto Vázquez

Validación de un Sistema Tutor Inteligente Web para el aprendizaje de las Matemáticas

Pág. 18 ⇒ por Alfonso Rosas Escobedo, Erica María Lara Muñoz y N. Sofía Huerta Pacheco

Obtención de perfiles de tutores académicos utilizando técnicas de Soft Computing

Pág. 23 ⇒ por M. Rosario Vazquez, Rafael Sánchez-Lara, Y. El Hamzaoui y Patricia Zavaleta-Carrillo

La agricultura de precisión impulsada por la Inteligencia Artificial

Pág. 29 ⇒ por Rogelio B. Olvera Rosales, Julio C. Ramos Fernández, Marco A. Márquez Vera Issac Hernández Santos

Pronóstico de Velocidad de Viento por Medio de un Conjunto de Modelos

Pág. 34 ⇒ por Hector Rodriguez-Rangel, Juan R. Valenzuela-Barraza, Carlos Lara-Alvarez, Luis A. Morales Rosales y Giovanni Manjarrez

Sistema Tutor Afectivo para la enseñanza de métodos numéricos de Euler y Runge-Kutta

Pág. 40 ⇒ por Gustavo Padrón Rivera, Cristina Joaquin Salas, Pedro Omar Silva Vásquez y Ricardo Serrano Herrera

Inclusión de la Inteligencia Artificial en Disputas Legales sobre Propiedad Intelectual

Pág. 45 ⇒ por Rodrigo Vidal-López y José A. Hernández-Servín

Redes Neuronales Recurrentes y su uso para la conversión de cadenas

Pág. 51 ⇒ por Jared D.T. Guerrero-Sosa, Víctor Hugo Menéndez-Domínguez y Víctor Uc-Cetina

Procesamiento de lenguaje natural con aprendizaje profundo

Pág. 56 ⇒ por Fernando Pech May, Luis A. López Gómez y Jorge Magaña Govea

Detección de Obstáculos Móviles Aplicada a la Conducción Autónoma Vehicular

Pág. 62 ⇒ por Lauro Reyes Cocoltzi, Ivan Olmos Pineda y J. Arturo Olvera López

Construyendo robots seguidores de línea avanzados

Pág. 67 ⇒ por Leonardo Romero, Moisés García, Antonio Camarena-Ibarrola e Ignacio Juárez-Campos

Columnas

Sapiens Piensa.
Editorial pág. 2

e-Tlakuilo pág. 4

Estado del IArte pág. 5

Sakbe pág. 6

IA & Educación pág. 73

Deskubriendo
Konocimiento pág. 75

Sapiens Piensa

Jorge A. Ruiz-Vanoye y Ocotlán Díaz-Parra

La Inteligencia Artificial (IA) es un tema de investigación e innovación que evoluciona continuamente. Una de las metas de la IA es imitar la Inteligencia Humana y aplicarla a la robótica. Es por ello que diversas empresas e instituciones públicas de algunos países están haciendo proyectos para conocer como el cerebro aprende nuevos conocimientos. Recientemente se están realizando experimentos de como conectar cerebros humanos a Internet (Internet de las Cosas Cerebrales) en el proyecto “Cerebro Humano” (proyecto médico-científico y tecnológico financiado por la Unión Europea). Estos proyectos tienen ventajas y desventajas a corto, mediano y largo plazo. Una de las grandes ventajas es la de poder determinar donde se encuentra el problema en enfermos de Parkinson, Alzheimer o síndrome de Down, empero la desventaja es la falta de leyes que protejan los neuroderechos de los ciudadanos.

En esta edición de Komputer Sapiens presentamos artículos de divulgación seleccionados mediante un proceso cuidadoso de arbitraje.

Para el tema de IA aplicada se presentan los trabajos de los siguientes investigadores:

Olvera Rosales, Ramos Fernández, Márquez Vera y Hernández Santos (La agricultura de precisión impulsada por la Inteligencia Artificial) mencionan la importancia de aplicar tecnología a la producción agrícola nacional y el desarrollo de sistemas inteligentes para la producción de alimentos, lo que se conoce mundialmente como “agricultura inteligente” (*Smart Farming*). Adicionalmente, mencionan que en la Universidad Politécnica de Pachuca (UPP) y en el Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos se están desarrollando varios proyectos para dar solución a problemas en esta área mediante la IA. Por ejemplo, proyectos en tractores autónomos, robots móviles para producción agrícola protegida, sistemas de control bio-climáticos en invernaderos, desarrollo de algoritmos con procesamiento de imágenes para el reconocimiento de surcos de maíz, entre otros.

Vidal López y Hernández Servín (Inclusión de la IA en disputas legales sobre propiedad intelectual) mencionan la importancia de la integración de la Minería de Datos, teoría de juegos y sistemas multi-agentes para las disputas legales sobre la propiedad intelectual.

Reyes Cicoletzi, Olmos Pineda y Olvera López (Detección de obstáculos móviles aplicada a la conducción autónoma vehicular) mencionan que con los avances recientes de la tecnología se ha intentado alcanzar e igualar de forma artificial las capacidades del sentido de la vista. Por ejemplo, el uso de visión artificial (visión estereoscópica computacional, sensores LIDAR, coincidencia

semi-global, mapas de disparidad y regiones de interés, percepción del movimiento, seguimiento de objetos) aplicada al sector automotriz, con orientación al manejo autónomo, a la identificación de obstáculos en el recorrido de un automóvil, y la estimación de trayectorias para evitar colisiones y daños a terceros.

Leonardo Romero, Moisés García, Antonio Camarena e Ignacio Juárez (Construyendo Robots seguidores de línea avanzados) abordan uno de los desafíos inherentes para estos robots avanzados: usar sensores para medir los desplazamientos angulares en los motores que impulsan las ruedas del robot de manera que sea posible construir un mapa del camino y usarlo para acelerar en segmentos rectos de la pista y frenar antes de entrar a las curvas.

Para el tema de Sistemas Tutores Inteligentes y *Soft Computing* se presentan las contribuciones de los siguientes investigadores:

Rosario Vázquez, Rafael Sánchez, Y. El Hamzaoui y Patricia Zavaleta (Obtención de perfiles de tutores académicos utilizando técnicas de Soft Computing) utilizaron la Lógica Borrosa para obtener perfiles de personalidad de los tutores académicos. Explotaron las bondades del servicio *Personality Insights*, para obtener los rasgos de personalidad de tutores a través del análisis lingüístico de reportes redactados en sesiones tutoriales. Extendieron la funcionalidad de este servicio, expresando los resultados arrojados en términos de lógica borrosa, para obtener el grado de afinidad del tutor evaluado con cada uno de los prototipos de perfiles de tutores.

Padrón Rivera, Joaquin Salas, Silva Vásquez y Serrano Herrera (Sistema Tutor Afectivo para la enseñanza de métodos numéricos de Euler y Runge-Kutta) mencionan que el cómputo afectivo es la parte de las ciencias de la computación que se encarga del estudio y el desarrollo de sistemas y dispositivos que pueden reconocer, interpretar, procesar y estimular las emociones humanas. Ellos desarrollaron un sistema tutor afectivo para el aprendizaje de los métodos de Euler y Runge-Kutta, el cual es un tipo de sistema tutor inteligente que mejora la experiencia de aprendizaje del usuario. Usaron 18 de 30 estudiantes del Instituto Tecnológico Superior de Teziutlán, Puebla. La edad promedio fue de 20 años en ambos sexos, y el 76

Rosas Escobedo, Lara Muñoz y Huerta Pacheco (Validación de un Sistema Tutor Inteligente web para el aprendizaje de las matemáticas) analizaron el funcionamiento y características de un sistema tutor inteligente de matemáticas. Mencionan que los sistemas tutores inteligentes permiten incrementar los resultados individuales de los estudiantes; y que el proceso de análisis jerárquico

permite validar la igualdad de funcionalidades de los sistemas tutores inteligentes. Adicionalmente, se procedió a probarlo con un grupo piloto, obteniendo resultados que mostraron que los estudiantes incrementaron sus calificaciones de manera individual.

Sobre el tema de *Deep Learning*, redes neuronales y pronósticos, este número se place en presentar los escritos de los siguientes investigadores:

Pech May, López Gómez y Magaña Govea (Procesamiento de lenguaje natural con aprendizaje profundo) mencionan que el aprendizaje profundo es una poderosa herramienta de aprendizaje automático, capaz de solucionar problemas que van desde la clasificación hasta el modelado de secuencias y árboles Markovianos; y que una de las técnicas de aprendizaje profundo más utilizadas para el procesamiento de lenguaje natural son las redes neuronales recurrentes y sus variantes

Héctor Rodríguez, Emmanuel Ramírez, Luis Morales, Juan Flores y Armando Guerra (Optimización de una red convolucional para la predicción de la velocidad del viento) proponen una metodología para el pronóstico de la velocidad del viento utilizando las redes neuronales convolucionales (RNC) como pronosticador de una serie de tiempo del viento, en conjunto con un algoritmo genético compacto para la búsqueda del mejor conjunto de hiper-parámetros de la RNC. La metodología propuesta se probó en un caso de estudio concreto sobre el pronóstico a corto plazo de la velocidad del viento. En particular, fueron utilizadas cuatro series de tiempo de velocidad del viento de estaciones meteorológicas ubicadas en distintas

regiones del estado de Michoacán, México.

Guerrero-Sosa, Menéndez Domínguez y Uc Cetina (Redes neuronales recurrentes y su uso para la conversión de cadenas) nos comparten el origen de las redes neuronales, así como los dos modelos tradicionales que se han enfocado en resolver problemas de clasificación y de reconocimiento de patrones. Exponen la importancia del uso de las redes neuronales recurrentes para la aplicación en problemas donde la red requiere tener el conocimiento previo para realizar su finalidad de manera correcta.

Rodríguez-Rangel, Valenzuela Barraza, Lara Álvarez, Morales Rosales y Manjarrez Montelongo (Pronóstico de velocidad del viento por medio de un conjunto de modelos) utilizan métodos de la IA para realizar los pronósticos de velocidad del viento, y se realiza una fusión de datos para buscar resultados con una mejor aptitud en el resultado. Ellos muestran evidencia de que se pueden obtener mejores resultados al utilizar diferentes métodos de pronóstico que son integrados mediante un esquema de fusión de datos.

Isidro Gómez Vargas, Ricardo Medel Esquivel, Ricardo García Salcedo y Alberto Vázquez (Una aplicación de las redes neuronales artificiales en la cosmología) mencionan que la generación de datos cosmológicos seguirá al alza en los próximos años. Será crucial escudriñarlos para mejorar nuestra concepción del universo. Indudablemente, la cosmología mantendrá la mirada atenta en los avances de la IA. Esperamos que este número sea del interés y agrado de los lectores.*



Dr. Jorge A. Ruiz-Vanoye es profesor - Investigador de la Universidad Politécnica de Pachuca, e Investigador Nacional. Mayor información en www.ruivanoye.com.



Dra. Ocotlán Díaz-Parra es profesor-Investigador de la Universidad Politécnica de Pachuca e Investigador Nacional. Mayor información en www.diazparra.net.

e-Tlakuilo: Cartas de nuestros lectores

Jorge A. Ruiz-Vanoye y Ocotlán Díaz-Parra

etlakulo@komputersapiens.org

En *Komputer Sapiens* nos hemos esforzado por estar “a sólo un click de distancia” a través de diferentes medios como Facebook, Twitter y correo electrónico. Les presentamos algunas de las preguntas que hemos recibido a través de estos medios.



Ernesto de la Cruz – Alumno de Universidad. (vía redes sociales)

Buenos días, actualmente soy estudiante de la carrera de Ingeniería de Software y me interesan los temas de los neuroderechos. Tengo una pregunta: ¿Cuál es el estado actual de la tecnología de extracción de mis pensamientos para trasladarlos a una computadora?

Hola gracias por escribir. Los neuroderechos son los derechos humanos de mantener los datos de nuestro cerebro de una manera privada y dentro de nuestro cuerpo. Roberto Adorno y Marcello Ienca han propuesto la necesidad de proteger de robo a los pensamientos y las memorias almacenadas en los cerebros de los ciudadanos. Ellos mencionan la necesidad de protección de lo que llaman neuroderechos. Los cuales contemplan el derecho a la libertad cognitiva (protección contra el uso no consentido de la información del cerebro humano), privacidad mental, integridad mental (protección de accesos no autorizados o manipulación de las señales del cerebro que puedan derivar en daños psicológicos o físicos), continuidad psicológica (mantener la identidad personal y la coherencia del comportamiento individual, principalmente evitar adición o supresión de memorias esenciales del cerebro).

Las principales tecnologías que se están desarrollando para gestionar la información de los cerebros son a través de las interfaces cerebro-máquinas (BCI). La idea fundamental de las BCIs

es ayudar a personas con alguna discapacidad motriz a controlar los dispositivos electrónicos.

Existen diversas compañías trabajando sobre tecnología BCI, por ejemplo, Neuralink, Facebook, Kernel, Emotiv, Neurosky entre otras. Facebook mencionó un plan para crear dispositivos que lean las ondas cerebrales de la gente y le permita escribir con solo pensarlo. Por otro lado, Neuralink (doi: 10.1101/703801) está diseñando un implante cerebral para conectar la mente de la gente directamente a una computadora, en otras palabras, conectar el cerebro a Internet. Los doctores Ruiz-Vanoye y Díaz-Parra mencionan que el Internet de las cosas cerebrales es la acción de conectar el cerebro humano a través de interfaces cerebro-máquina u otras tecnologías al Internet. El cerebro tiene diversos tipos de ondas cerebrales: Ondas Delta (1 a 3 Hz, un nivel adecuado de ondas delta favorece al sistema inmunitario, el descanso y la capacidad de aprender), ondas Theta (3.5 a 8 Hz, un nivel adecuado favorece la creatividad, la conexión emocional y la intuición; un nivel elevado pudiera relacionarse con trastornos depresivos y de falta de atención), ondas alfa (8 a 13 Hz, un nivel elevado nos indicaría una sensación de pocas fuerzas para realizar una actividad, un nivel bajo causaría ansiedad, estrés e insomnio), ondas Beta (12 a 32 Hz, un nivel óptimo de ondas ayudaría a estar más receptivos y mejorar nuestra capacidad para resolver problemas, un nivel bajo nos conduciría a un estado relajado, laxo y depresivo), ondas Gamma (25 a 100 Hz, un nivel óptimo permitiría un alto pensamiento cognitivo, capacidad de procesar nueva información en el cerebro, estados de felicidad; las personas con problemas mentales o de aprendizaje tienen pocas o nulas ondas Gamma).

El desafío principal que se encuentran las empresas que están desarrollando tecnología para leer la mente es la búsqueda de los patrones en la actividad neuronal entre grandes cantidades de información generada por nuestro cerebro. El uso de BCIs para gestionar la información traería ventajas y desventajas. En el mundo ya existen muchos biohackers experimentales que hacen modificaciones a su cuerpo para ser seres humanos mejorados.

Estado del IArte

María del Pilar Gómez Gil (@pgomezgil) y Jorge Rafael Gutiérrez Pulido (@jrpgpulido)
estadoiarte@komputersapiens.org

“Muestras” es el término que se usa para denominar el nuevo *petróleo digital*, también conocido coloquialmente como “datos” o “big data” al presentarse en cantidades significativas. El término “Internet de las Cosas” (IoT) se ha utilizado en los últimos años para referenciar a la red de dispositivos interconectados vía internet que captan dichas muestras. Los relojes inteligentes, telescopios, computadoras, drones, vehículos, cámaras, edificios y un sinnúmero más de sensores se encuentran actualmente extrayendo este petróleo. Se calcula que el número de dispositivos en IoT pronto llegará a los 50,000 millones, de acuerdo con la empresa de telecomunicaciones CISCO.

Estas muestras forman un monolito que tiene que romperse para poder realmente sacarle provecho. Por ejemplo, los relojes inteligentes colectan información acerca de quien los viste o las computadoras de sus usuarios, pero a tales muestras hay que aplicar algoritmos específicos para producir información valiosa para la toma de decisiones, las cuales podrían ser autónomas. IoT puede aprovecharse en diferentes áreas, como la seguridad o la mejora del medio ambiente y en todos los ámbitos: social, público, privado o educativo. La mayoría de los algoritmos de inteligencia artificial, son directamente aplicables en estos casos, ya que aprovechan el número grande de muestras para entrenar sus modelos y tener un mejor desempeño. En una institución de educación superior las posibilidades no son menores. Desde el punto de vista de desarrollo de aplicaciones de software, por ejemplo, los relojes inteligentes permiten identificar a quienes los visten, las computadoras sus preferencias en línea y los sensores de monitoreo de energía proporcionan información sobre qué debe optimizarse al interior de la institución. De esta manera, las instituciones de educación superior se pueden convertir en espacios de trabajo eficientes, seguros, y completamente conectados, en donde se promueven el aprendizaje proactivo y la investigación innovadora.

Pero más no significa necesariamente mejor. El gran reto que se enfrenta al tener grandes cantidades de datos en tiempo real está en poder manipularlos de forma tal que pueda obtenerse información útil para la mejora de la productividad, seguridad, confort o cualquier otro valor

que se busque en el campo específico de aplicación. Según el artículo “las cosas industriales producen salidas de tamaño industrial” escrito por Kareem Yusuf en el blog de la IBM del IoT en septiembre de 2018, el 80 % de los datos generados por dispositivos IoT no se encuentran estructurados de ninguna manera, por lo que la capacidad de las compañías para dar sentido a toda esa información está siendo sobrepasada. Entonces, los expertos consideran que solamente con el uso de tecnología basada en Inteligencia Artificial podrá aprovecharse realmente esta cantidad de información. Además, es indispensable que las empresas u organizaciones que desean sacar provecho de IoT tengan implementados procesos rigurosamente estructurados en las áreas donde desean aprovechar IoT. Un ejemplo impactante de la aplicación de IoT a través del sistema comercial “Watson IoT,” se encuentra en el sistema de transporte ferroviario francés SNCF. Esta empresa, que da servicio a alrededor de 2,000 millones de pasajeros a través de 3,000 estaciones ferroviarias en Francia, utiliza IoT a través de datos recolectados por millones de sensores colocados en sus rieles. Con el uso de lo que se conoce actualmente como “analítica de datos,” SNCF consigue una calidad impresionante en el servicio, confort y productividad en sus transportes.

IoT no es futuro, sino presente, que entre más nos tardemos en entender y dominar más lejos estaremos de aprovecharlo y de competir en este mundo globalizado. Algunas iniciativas promovidas por organizaciones profesionales internacionales como la “*IEEE Internet of Things*” pueden ayudarnos a conocer las herramientas y procesos requeridos para dominar esta tecnología.*

Para saber más sobre éstas y otras aplicaciones del uso inteligente de IoT consulta:

- <https://bit.ly/2qsonxB>.
- <https://www.ibm.com/mx-es/internet-of-things>.
- <https://www.ibm.com/blogs/internet-of-things/iot-industrial-ai/>.
- <https://iot.ieee.org/>.

Sakbe

Claudia Guadalupe Gómez Santillán, Laura Cruz Reyes y Laura Gómez Cruz
sakbe@komputersapiens.org

<https://www.patroneshermososmexico.org>

Mucho se habla de impulsar el empoderamiento de las mujeres, pero, ¿qué significa empoderar realmente? Como la misma palabra lo implica, “empoderar” significa literalmente dar poder a alguien, y este “poder” ayuda a adquirir la fortaleza necesaria para resistir las adversidades que puedan presentarse. En las ciencias de la computación, las mujeres aún necesitan luchar contra corriente para poder triunfar, y el programa “Patrones Hermosos” lo reconoce.

Este programa parte de la premisa de que el pensamiento computacional es una nueva habilidad tan fundamental e importante como lo son la lectura, la escritura y las matemáticas. Por lo tanto, el programa busca impulsar esta habilidad en jóvenes latinoamericanas entre los 13 y 17 años para que alcancen su máximo potencial y no se queden atrás en esta revolución educativa.

Particularmente, el programa se enfoca en el reconocimiento de patrones mediante el desarrollo de algoritmo divertidos en torno a tareas cotidianas. La identificación de patrones es una tarea importante de la computación que facilita llegar a la óptima toma de decisiones. Y para participar, no se necesitan conocimientos previos al respecto, solo se necesita tener interés en colaborar en un proyecto que aunque es de alto nivel, también es de alto entretenimiento. Entre las actividades que las participantes pueden realizar se encuentran la búsqueda de patrones para maximizar la felicidad, patrones para guardar y descubrir secretos, para almacenar y ordenar cosas, para hacer rutas y horarios más eficientes y para generar estrategias para ganar juegos. Todo esto acompañado de talleres, conferencias y presentaciones de la mano de expertas.

“Patrones Hermosos” es impartido por otras mujeres que estudian en universidades de renombre tal como el MIT, el ITESM y el CSOFTMTY, entre otras. Llegó por primera vez a Monterrey en el 2017 y se prevé que beneficie a más de dos mil mexicanas si se continúa apoyando, lo cual se puede lograr de tres maneras distintas, ya sea con una capacitación como instructora mexicana con el MIT, con un patrocinio, o fungiendo como una sede. De

continuar, estas jóvenes mujeres podrían poner lo aprendido en práctica para incentivar la economía desde un buen posicionamiento educativo o laboral disminuyendo a su vez la desigualdad de género en el área.

Para saber más puedes consultar:

- <https://dreamgrande.io/>.
- <https://sg.com.mx/buzz/patrones-hermosos>.
- <http://www.lania.mx/patrones-hermosos-2019/>.



Las recetas de cocina son como algoritmos, con ellas puedes hacer patrones hermosos y deliciosos. Foto de postre libanés, Laura Gómez.

ARTÍCULO ACEPTADO

Optimización de una red convolucional para la predicción la velocidad del viento

Hector Rodriguez-Rangel, Emmanuel F. Ramírez-Hernández, Luis A. Morales, Juan J. Flores y J. Armando Guerra

A lo largo de los años la generación de energía a nivel mundial se ha producido utilizando recursos naturales no renovables (energía nuclear, combustibles fósiles y minerales, entre otros). Los medios para la producción de este tipo de energía no pueden ser producidos, regenerados, o reutilizados a una escala tal que se pueda sostener su consumo.

En el caso de la energía eólica, la materia necesaria para la producción de energía es el viento. Pero el comportamiento del viento es caótico, generando una incertidumbre al cuantificar la materia prima disponible; no se sabrá con certeza la cantidad de energía a generar. Por lo que es necesario contar con un pronóstico acertado sobre el comportamiento del viento en las próximas horas, días o meses, dependiendo de la tarea a realizar.

En este artículo se propone una metodología para el pronóstico de la velocidad del viento utilizando las redes neuronales convolucionales (RNC) como la herramienta de aprendizaje. Para esto fue necesaria la trans-

formación de una serie de tiempo a imagen, dado que es la entrada de nuestra herramienta de aprendizaje.

Existen una gran variedad de implementaciones de una red convolucional utilizadas en el área de predicción [1]. Sin embargo, la elección del número de capas, tipo de capa, tamaño de filtro, etc. se realiza de manera artesanal. Por otro lado, en la literatura existen métodos de optimización de la red. Sin embargo, estos pueden consumir mucho tiempo de procesamiento y no ser efectivos.

La metodología propuesta permite explorar un mayor número de configuraciones para una red convolucional. El objetivo de este proceso consiste en encontrar la mejor configuración, la que nos otorgue mejores resultados. Aunado a una mejora en los tiempos de entrenamiento obtenida mediante el uso de la paralelización del proceso de entrenamiento en las tarjetas gráficas (GPU). Con ello, es posible explorar un conjunto mayor de combinaciones de todas las posibles mejores configuraciones.

Los recursos no renovables se han ido agotando debido a los precios bajos y la disponibilidad de algunos de estos recursos. Gracias a estos hechos las energías no renovables están en desventaja frente a las energías renovables.

La metodología propuesta fue probada con series de tiempo de la velocidad del viento de diferentes localidades del estado de Michoacán en un servidor con una tarjeta GPU integrada. Donde la ejecución de la RNC se realizó utilizando una unidad de procesamiento gráfico (GPU), mientras que la ejecución del algoritmo genético compacto (AGc) se realizó en la unidad central de procesamiento (CPU) del servidor. Al paralelizar el proceso se pretende tener una mayor exploración del espacio de modelos en un menor tiempo.

Metodología propuesta

El modelo Convolutivo-Evolutivo consta de 3 subprocesos principales, los cuales son: *Pre-Procesamiento de datos*, *Generación del banco de imágenes*, *Proceso de entrenamiento*. El proceso consiste en: 1) recibir una serie de tiempo realizando un preprocesamiento de datos, 2) generar un banco de datos al transformar la serie de tiempo en imágenes, y 3) generar el entrenamiento del

modelo de aprendizaje a partir del banco de datos (imágenes). Como salida se obtiene un modelo predictivo de la serie de tiempo estudiada. Este proceso se ilustra en la Figura 1.

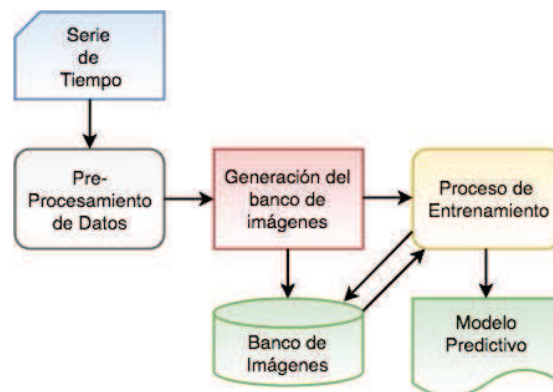


Figura 1. Generación de un modelo predictivo de la velocidad del viento.

Pre-Procesamiento

En el Pre-Procesamiento de datos se realiza la revisión de datos atípicos (outliers) en la serie de tiempo y la normalización de los datos. Un dato atípico puede ser entendido como una observación que cae fuera del patrón general de la distribución de probabilidad de los datos.

Un ejemplo de datos atipicos se muestra en la Figura 2.

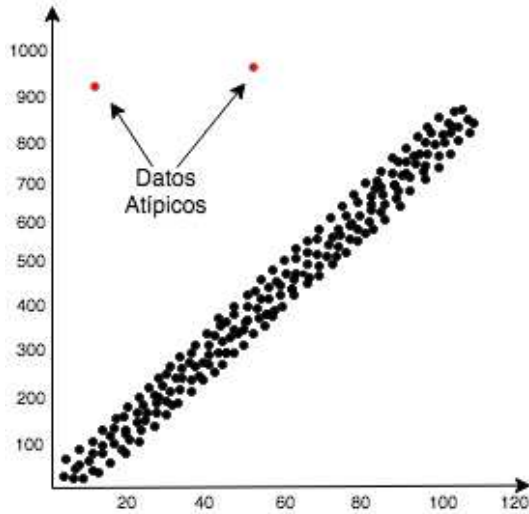


Figura 2. Datos atípicos encontrados en una serie de tiempo.

Para la eliminación de los datos atípicos, se inicia con un suavizado de la serie de tiempo. La técnica utilizada para el suavizado es llamada Moving Average o en español Media Móvil [2]. En estadística esta técnica es utilizada para realizar un análisis de los datos u observaciones creando una serie a partir de promedios.

Después del proceso de suavizado, los datos pasan a un proceso de normalización. Esto es, se mapean los

datos a otro rango de valores (i.e. $0 \leq x_i \leq 1$ donde x_i es una observación normalizada).

Generación del banco de imágenes

Con el fin de usar las RNC en el proceso de predicción es necesario transformar la serie de tiempo a un conjunto de imágenes.

La Figura 3 muestra como una ventana w se va deslizando a través de la serie de tiempo. Y a partir de esta ventana de datos se genera una imagen asignándole el valor predictivo P_i .

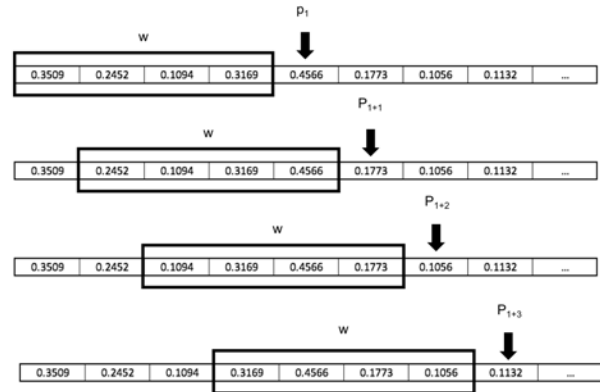


Figura 3. Deslizamiento de la ventana w y asignación de valor esperado P_i .

El método utilizado en este trabajo para la transformación de serie de tiempo a imagen es descrito en el trabajo por Wang & Oates "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks"[3]. En este trabajo se exploraron las técnicas Grammar Angular Field y Escaneo lineal para la conversión de un vector a imagen.

Las RNC reciben como entrada la información estructurada en un vector de matrices, donde cada una de estas matrices representan los datos de una imagen.

Entrenamiento

El proceso de entrenamiento se realiza mediante la implementación de un Algoritmo Genético Compacto para la exploración de una óptima configuración (hiperparámetros) de una red convolucional. El algoritmo genético se encarga de realizar una búsqueda estocástica de un conjunto de hiperparámetros óptimos. Estos hiperparámetros o posible solución, serán utilizados por la Red Neuronal Convolucional para realizar el entrenamiento de un modelo predictivo de la velocidad del viento.

La RNC es la encargada de realizar el aprendizaje

del comportamiento del viento. La arquitectura general de RNC utilizada para este trabajo se basa en trabajo realizado por Pérez-Espinosa et al. [4], donde la red está conformada por 12 capas, de las cuales 2 capas de convolución van transformando la entrada a través de cada una de ellas.

La arquitectura propuesta tiene dos capas de convolucion, activación y pooling, para después, transformar los datos (reshape) y pasar a la red completamente conectada. La Figura 4 muestra estas 12 capas, que forman la arquitectura de la RNC propuesta para resolver el pro-

blema.

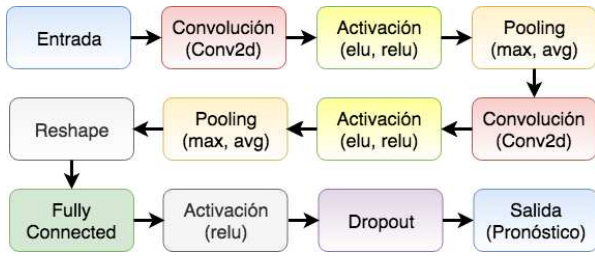


Figura 4. Arquitectura utilizada para la RNC.

La implementación de la RNC se realizó utilizando la librería Keras y el backend Tensorflow para GPU. Donde el AGc es ejecutado en el CPU del servidor (Intel Core i5) y el entrenamiento de la RNC es ejecutado en una tarjeta aceleradora gráfica NVIDIA GTX 1080.

El determinar los mejores hiper-parámetros para la RNC es un problema de optimización. Las Redes Neuronales, por su complejidad, toman una serie de hiper-parámetros que necesitan ser configurados correctamente para obtener un mejor resultado. Normalmente esta serie de hiper-parámetros son configurados utilizando un método muy común de prueba y error hasta llegar a un resultado deseado. A pesar de eso, este método no garantiza un rendimiento óptimo en el proceso de entrenamiento y el tiempo para llegar a un resultado deseado puede ser demasiado alto.

A continuación se describen cada uno de los hiper-parámetros considerados; en total son 10 hiper-parámetros clasificados entre General y Convolutacional.

1. General

- a) **Épocas:** Es un número entero positivo que limita el número de pasos en que el conjunto de entrenamiento es evaluado.
- b) **Factor de aprendizaje:** Es un número de punto flotante que representa la magnitud de la actualización por cada época de entrenamiento.
- c) **Entrenamiento:** Es un número que representa el porcentaje de datos que se incluyen en el conjunto de entrenamiento.
- d) **Optimizador:** Son funciones que computan gradientes para una medida de pérdida y aplican el gradiente a las variables de la red. Los tres posibles valores de este hiper-parámetro son Stochastic gradient descent (SGD), AdamOptimizer (ADAM) y RMSPropOptimizer (RMSProp).
- e) **Activación:** Funciones que proveen la no linealidad del modelo producido por una RNA.

Este hiper-parámetro puede tomar dos valores, una función rectificadora lineal (relu) y una función para suavizar la no linealidad (elu).

2. Convolutacional

- a) **Tamaño del Filtro:** Es un número que representa el tamaño del filtro que recorrerá la matriz que representa la imagen.
- b) **Strides:** Una lista de números enteros que representa el número de características que serán recorridas de izquierda a derecha en la matriz, moviendo el filtro por cada dimensión en el vector de entrada.
- c) **Padding:** Indica si el filtro puede ir más allá de los límites de la matriz. Los valores que puede tomar este hiper-parámetro son *same* y *valid*.
- d) **Pooling:** Reduce la dimensionalidad de la entrada permitiendo hacer suposiciones de características contenidas en una región de la entrada. Los valores que puede tomar este hiper-parámetro son *maxpooling* y *avgpooling*.
- e) **Dropout:** Es una técnica de regularización para reducir el sobre entrenamiento en redes neuronales. Reduce el número de neuronas en la RNA.

Para la optimización de los hiper-parámetros, anteriormente mencionados se utiliza un AGc [5], el cual es una versión no poblacional del algoritmo genético simple. En el algoritmo AGc cada individuo (Cromosoma) es definido como un vector binario que codifica cada uno de los hiper-parámetros. El AGc es usado para definir el mejor conjunto de hiper-parámetros para la RNC.

La Figura 5 muestra las posiciones del cromosoma utilizados en el proceso de optimización (AGc). Por ejemplo, para obtener el hiper-parámetro Épocas se toman como referencia las 3 primeras posiciones del vector binario se decodifica a decimal y se conoce el valor correspondiente a la variable Épocas.

Épocas	Aprendizaje	Entrenamiento	Optimizador	Activación	Tamaño del Filtro	Strides	Padding	Pooling	Dropout
1	0	1	0	0	0	0	1	1	0

Figura 5. Codificación de los diferentes hiper-parámetros en el cromosoma del AGc.

Resultados

Con el propósito de comprobar la efectividad de la metodología propuesta, ésta se probó en el caso de estudio concreto del pronóstico a corto plazo de la velocidad del viento. En particular fueron utilizadas 4 series

de tiempo de velocidad de viento de estaciones meteorológicas de viento de distintas regiones del estado de Michoacán, México. Las series de tiempo contienen observaciones cuyo intervalo de captura entre observaciones es de una hora promedio. Donde, cada una de las series obtenidas consta de 4,416 observaciones.

Una vez que el pre-procesamiento de los datos fue realizado para cada serie se procedió a realizar el entrenamiento. Dentro del proceso de entrenamiento se realiza la generación del banco de imágenes, generando las imágenes con las que será realizado el entrenamiento y asociándolas a un valor a pronosticar. El tamaño del banco de imágenes está directamente relacionado con el tamaño de la serie de tiempo utilizada, el tamaño de las imágenes generadas fue de 24 x 24 pixeles.

La última fase es la de pronóstico en la cual se utiliza el modelo generado en la fase de entrenamiento para generar el pronóstico. En esta fase también se evalúa que el modelo generado cumpla con la aptitud buscada.

Para la transformación de las series de tiempo a imágenes se utilizó una técnica basada en la presentada por Wang, Z. and Oates, T [4]. Se realizaron pruebas con dos técnicas, Escaneo Lineal y Gramian Angular Field para crear bancos de imágenes. Se observó que en los bancos de modelos creados usando la técnica de escaneo lineal, el tamaño del banco de imágenes disminuye significativamente.

Para el proceso de entrenamiento se realizó una división de los datos, separándolos en un conjunto de entrenamiento y un conjunto de validación. Se tomó un 20 % fijo para el conjunto de validación y el porcentaje de datos que se utilizó para el conjunto de datos de entrenamiento fue seleccionado por el AGc dentro del rango de 60 % - 80 %. Para la obtención de cada uno los modelos se realizó la búsqueda de los diferentes hiper-parámetros con los cuales se obtuviera el modelo que mejor se apegue a los datos de entrenamiento utilizando un AGc para la optimización de hiper-parámetros de estos. La Tabla 1 muestra los parámetros clasificados como Generales. e.g. para el hiper-parámetro Épocas solo se podrá seleccionar un valor entre los siguientes números: 20, 40, 60, 80, 100, 120, 160. La Tabla 2 muestra los hiper-parámetros clasificados como Convolucionales. e.g. para el hiper-parámetro Polling solo se podrá seleccionar un valor entre: MaxPooling2D y AveragePooling2D.

Para el AGc se utilizaron como configuración 100 generaciones, el tamaño del cromosoma de 19 posiciones y 100 individuos.

Tabla 1. Hiper-parámetros generales y sus posibles valores

hiper-parámetro	Rango
Épocas	[20, 40, 60, 80, 100, 120, 160]
Tasa de Aprendizaje	[0.0001, 0.0006, 0.0011, 0.0016, 0.0021, 0.0026, 0.0031]
Tasa de Entrenamiento	[0.70, 0.80, 0.90, 1.00]
Optimizador	[SGD, ADAM, RMSprop]
Activación	[relu, elu]

Tabla 2. Hiper-parámetros convolucionales y sus posibles valores

hiper-parámetro	Rango
Tamaño del Filtro	[3, 4, 5, 6]
Strides	[2, 3, 4, 5]
Padding	[valid, same]
Polling	[MaxPooling2D, AveragePooling2D]
Dropout	[0.3, 0.4, 0.5, 0.6]

Se realizó una comparación de resultados obtenidos con la metodología propuesta utilizando RNC y los resultados obtenidos utilizando la técnica de estimación Naïve $\hat{y}_{t+1} = y_t$. La Tabla 3 muestra una comparación de los resultados de las diferentes implementaciones de las series de tiempo utilizadas de el Fresno, La Piedad, Aristeo Mercado y La Piedad Michoacán. La comparación se realiza utilizando las medidas de aptitud MSE ($MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$, ver definición en [8]).

Tabla 3. Tabla comparativa entre el método Naïve y

Ciudad	RNC	
	Método	MSE
El Fresno	RNC	0.0369
	Naïve	0.1922
La Piedad	RNC	0.0246
	Naïve	0.1568
La Palma	RNC	0.0194
	Naïve	0.1394
Aristeo Mercado	RNC	0.0148
	Naïve	0.1219

Como se observa en la Tabla 3, la propuesta de este trabajo supera al método de Naïve.

Los experimentos fueron divididos en tres fases, la fase de pre-procesamiento de los datos, la fase entrenamiento del modelo, y la fase de pronóstico.

Conclusiones

En este trabajo se utilizaron Redes Neuronales Convolucionales como pronosticador de una serie de tiempo de viento, en conjunto con un Algoritmo Genético Compacto para la búsqueda del mejor conjunto de hiper-parámetros de la Red Neuronal Convolutiva. Con ello se evita realizar el clásico proceso de búsqueda a prueba y error, ya que se encuentra un conjunto de hiper-parámetros con los cuales se obtienen los resultados deseados con un mejor desempeño del modelo resultante. La metodología desarrollada se le dió el nombre de enfoque Convolutiva Evolutiva.

Se utilizó una técnica para la codificación de una serie de tiempo a imagen. Para el proceso de entrenamiento se utilizaron RNC; la optimización de los hiper-parámetros de la RNC se realizó con un Algoritmo Genético Compacto, el cual explora una gran cantidad de soluciones de los hiper-parámetros de la RNC.

Para finalizar se presentó una comparación de los resultados obtenidos utilizando RNC y la optimización de hiper-parámetros de la RNC contra el pronóstico obtenido con el método Naïve, obteniendo mejores resultados con la metodología planteada. Con ambas técnicas o metodologías es posible obtener un pronóstico acertado a

corto plazo de la velocidad del viento. En este caso con la metodología Convolutiva Evolutiva se obtuvieron mejores resultados.*

REFERENCIAS

1. Chang, W-Y. (2013). Short-Term Wind Power Forecasting Using the Enhanced Particle Swarm Optimization Based Hybrid Method. *Energies*.
2. Montgomery, D. C., Jennings, C. L., y Kulahci, M. (2015). Introduction to time series analysis and forecasting. *John Wiley & Sons*.
3. Wang, Z., y Oates, T. (2015). Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. En *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
4. Pérez-Espinosa, H., Avila-George, H., Rodriguez-Jacobo, J., Cruz-Mendoza, H. A., Martínez-Miranda, J., y Espinosa-Curiel, I. E. (2016). Tuning the Parameters of a Convolutional Artificial Neural Network by Using Covering Arrays. *Research in Computing Science*, 121, 69-81.
5. Harik, G. R., Lobo, F. G., y Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE transactions on evolutionary computation*. 3(4), 287-297.

SOBRE LOS AUTORES



Hector Rodríguez Rangel es egresado del Instituto Tecnológico de Morelia en la carrera de Ingeniería en Sistemas Computacionales (2009). La maestría y el doctorado los realizó en el departamento de Posgrado de la Facultad de Ingeniería Eléctrica en la Universidad Michoacana (2009, 2014). Actualmente es Profesor Titular en el Tecnológico Nacional de México campus Culiacán.



Emmanuel Francisco Ramírez Hernández es egresado del Instituto Tecnológico de Culiacan en la carrera de Ingeniería en Sistemas Computacionales (2015). Actualmente estudia el posgrado de Maestría en Ciencias Computacionales en el departamento de Posgrado e Investigación del Instituto Tecnológico de Culiacán.



Luis A. Morales Rosales obtuvo el grado de doctor en Ciencias Computacionales en el Instituto Nacional de Astrofísica Óptica y Electrónica. Actualmente es catedrático Conacyt en la Universidad Michoacana de San Nicolás de Hidalgo.



Dr. Juan José Flores Romero es egresado de la Universidad Michoacana de San Nicolás de Hidalgo de Ingeniería en Electrica (1984). Cuenta con Maestría en ciencias del Centro de Investigación y Estudios Avanzados, del Instituto Politécnico Nacional. En 1997 obtuvo el grado de Doctor en Filosofía, especialidad en Computación, de la Universidad de Oregon, en Estados Unidos.



Jesus Armando Guerra García realizó sus estudios en el Instituto Tecnológico de Culiacán en la carrera de Licenciatura de Ingeniería en Sistemas Computacionales (2011). Cuenta con maestría en Ciencias de la Computación en el Instituto Tecnológico de Culiacán (2018).

ARTÍCULO ACEPTADO

Una Aplicación de las Redes Neuronales Artificiales en la Cosmología

Isidro Gómez Vargas, Ricardo Medel Esquivel, Ricardo García Salcedo y J. Alberto Vázquez

La búsqueda científica de un modelo que describa al Universo ha producido sofisticadas exploraciones cosmológicas y una inmensidad de datos observacionales. Para explorar y analizar esta información, la ciencia computacional resulta imprescindible.

El idilio entre cosmología y computación podría ser tema de varios libros; en este artículo nos centraremos en una aplicación de las redes neuronales artificiales (RNAs) en la estimación de parámetros de modelos cosmológicos.

La cosmología y sus datos

La cosmología es la ciencia que estudia el Universo a gran escala, donde las galaxias pueden ser descritas como puntos (Figura 1). Su objetivo es describir el origen, evolución, composición y destino del Universo a través de teorías físicas que se confrontan con observaciones astronómicas.

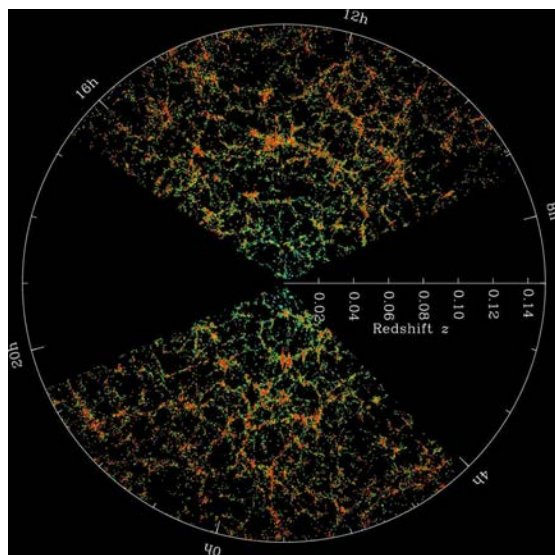


Figura 1. Distribución de galaxias locales generada con los datos del SDSS¹.

¹Fuente: www.sdss.org/science/orangepie

²Premio Turing en 1998, creador del concepto de Cubo de Datos y desarrollador de la base de datos del SDSS.

³science.nasa.gov/missions/cobe

⁴map.gsfc.nasa.gov

⁵www.phy.cam.ac.uk/research/research-groups/ap/plank

⁶SDSS son las siglas de Sloan Digital Sky Survey (www.sdss.org)

⁷www.lsst.org

En la actualidad, el modelo cosmológico que mejor describe los datos observacionales es el modelo de materia oscura fría con constante cosmológica (abreviado Λ CDM). Se basa en la teoría general de la relatividad de Einstein, el Big Bang, la existencia de materia oscura fría (solo detectable por efectos gravitacionales) y de energía oscura (componente misterioso que produce la expansión acelerada del Universo).

Sin embargo, el modelo Λ CDM es incapaz de explicar algunas cuestiones físicas importantes. Esto ha propiciado la generación de una amplia gama de modelos, algunos basados en teorías alternas a la relatividad general y otros en hipótesis sobre la existencia de materia no ordinaria.

La cosmología es una ciencia muy singular debido a su objeto de estudio. El Universo es un sistema único, de fronteras inaccesibles, que debe estudiarse desde adentro sin la posibilidad de realizar experimentos. Esto exige dos características metodológicas en la cosmología contemporánea: análisis intensivo de datos y estadística bayesiana.

Jim Gray² pronosticó el advenimiento de un cuarto paradigma en la ciencia. El primer paradigma ocurrió hace mil años cuando la ciencia era descriptiva y empírica; el segundo tuvo su inicio con las investigaciones de Isaac Newton al adquirir un carácter teórico; el tercero surgió en el siglo XX, al incorporar simulaciones computacionales. El cuarto paradigma sería la ciencia de datos o *e-ciencia*, una mezcla de ciencia empírica-teórica con análisis computacional de una cantidad exorbitante de datos [1].

Información sobre supernovas, galaxias, radiación cósmica de fondo, oscilaciones acústicas de bariones y otros fenómenos cósmicos es recolectada desde la Tierra o el espacio. Proyectos como COBE³, WMAP⁴, el satélite Planck⁵ y el SDSS⁶ proveen un laboratorio cosmológico con datos importantes para validar o descartar modelos teóricos. Esta nueva etapa en el estudio del Universo se conoce como *cosmología de precisión*.

Además, dentro de pocos años, el telescopio LSST⁷ fotografiará toda la bóveda celeste con la cámara más

grande jamás construida (3.2 Gigapíxeles), y se estima que recopilará 15 Terabytes por noche. Se vislumbra que la gestión y minería de datos del telescopio sean la parte técnica más difícil del proyecto. Por otro lado, el próximo radiotelescopio más grande del mundo, SKA⁸, coleccionará aproximadamente 14 exabytes por día, suficientes datos como para llenar 15 millones de iPods de 64 GB; además, su computadora central tendrá el poder de, más o menos, 100 millones de unidades de procesamiento.

Puede apreciarse que la cosmología contemporánea, con su torrente de datos, cristaliza en una realidad la visión de Jim Gray del cuarto paradigma.

El análisis de los datos cosmológicos requiere de técnicas estadísticas y computacionales para ajustar los parámetros de los modelos teóricos. En este contexto, las RNAs y otras técnicas de inteligencia artificial pueden aportar sus virtudes.

La inferencia bayesiana

La inferencia estadística utilizada en cosmología tiene como fundamento el teorema de Bayes, una controvertida aplicación de la probabilidad condicional con más de dos siglos de historia, que ha conquistado grandes logros. Dada su naturaleza predictiva, se puede afirmar que el teorema de Bayes ha sido pionero del aprendizaje automático, de hecho, se puede emplear para detectar spam y hacer diagnósticos médicos [2].

Al lector interesado en la inferencia bayesiana se le recomienda la Ref. [2] para una introducción, la Ref. [3] presenta un tratamiento más formal y la Ref. [4] es una buena guía en el contexto cosmológico.

En la prueba de modelos teóricos, el teorema de Bayes adquiere el siguiente aspecto:

$$P(\theta|D, H) = \frac{P(D|\theta, H)P(\theta|H)}{P(D|H)},$$

donde D representa el conjunto de datos observacionales, H es la hipótesis (modelo) y θ el conjunto de parámetros (un modelo lineal, por ejemplo, tendría como parámetros libres la pendiente de una recta y su ordenada al origen). La probabilidad *previa* $P(\theta|H)$ ⁹ representa nuestro conocimiento de los parámetros θ antes de considerar los datos observables. Esta probabilidad se modifica a través de la *verosimilitud* $P(D|\theta, H)$ al incluir datos experimentales D . El objetivo final es obtener la probabilidad *posterior* $P(\theta|D, H)$, que representa el estado de nuestro conocimiento de los parámetros del modelo al considerar la información de los datos. La constante de normalización $P(D|H)$, o *evidencia bayesiana*, es el promedio de la *verosimilitud* sobre la probabilidad *previa*:

$$P(D|H) = \int d^N \theta P(D|\theta, H)P(\theta|H),$$

donde N es la dimensión del espacio de parámetros. Esta cantidad, fundamental en la comparación de modelos, es una constante que puede omitirse en la estimación de parámetros.

La *verosimilitud* se determina al suponer una distribución estadística para los datos D , la probabilidad *previa* se define a partir de conocimientos preliminares. Al calcular la distribución *posterior*, se pueden obtener predicciones basadas en el modelo teórico y en las observaciones empleadas.

La estimación de parámetros encuentra el valor más probable de éstos, a la luz de los datos. En cosmología, los parámetros son las principales cantidades que describen el Universo; como no existe una teoría fundamental que las prediga, sus valores deben estimarse estadísticamente.

Esto se realiza mediante muestreos aleatorios en el espacio de parámetros e iteraciones del teorema de Bayes hasta que los resultados alcanzan un estado estacionario. Los valores de los parámetros en ese momento conforman la mejor estimación y también se pueden conocer las regiones de confianza.

Como ejemplo, la Figura 2 muestra gráficas resultantes de una inferencia bayesiana. A la derecha, la distribución *posterior* 1-Dimensional con la región más probable del parámetro Ω_m . La imagen de la derecha representa la distribución *posterior* 2-Dimensional, con regiones de probabilidad de los parámetros Ω_m y Ω_b ; la parte interna representa la región de confianza 1σ (68.3%), mientras que la externa corresponde a 2σ (95.4%).

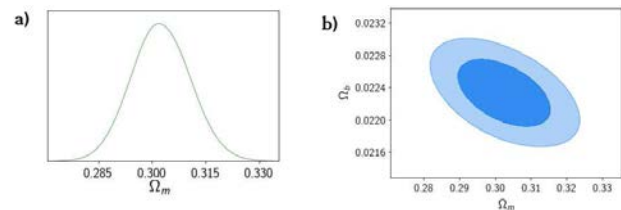


Figura 2. Distribución *posterior*: a) 1-Dimensional, b) 2-Dimensional.

En cosmología, como en otras ciencias, la inferencia bayesiana se implementa, en general, mediante métodos Monte Carlo vía Cadenas de Markov (MCMC), con algoritmos como Metropolis-Hastings, Gibbs u otras propuestas de muestreo.

El espacio de parámetros que exploran estos algoritmos tiene tantas dimensiones como parámetros tiene el modelo. Un número grande de parámetros produce funciones de probabilidad muy complejas. Así, a pesar de

⁸www.itweb.co.za/content/mraYAyqowZnqJ38N

⁹Las probabilidades involucradas son condicionales, por ejemplo: $P(\theta|D, H)$ es la probabilidad de que ocurra θ dado que han ocurrido D y H .

que los algoritmos MCMC suelen generar buenas estimaciones, también gastan muchos recursos computacionales. Dado que los modelos cosmológicos suelen tener al menos seis parámetros, diversos grupos de investigación han desarrollado códigos especializados, por ejemplo: CosmoMC¹⁰ y CosmoSIS¹¹.

En algunos casos, además de la estimación de parámetros es necesario comparar modelos para determinar al mejor. La comparación de dos modelos distintos se realiza mediante el cociente de los valores de la *evidencia bayesiana* de ambos. Esto suele ser difícil o imposible de obtener analíticamente, incluso mediante cómputo en paralelo el cálculo puede tomar varias semanas.

La *evidencia bayesiana* es significativa porque penaliza a los modelos con mayor número de parámetros, a través de la distribución previa. Así, se obedece la Navaja de Occam, una guía filosófica que establece: "En igualdad de condiciones, la explicación más sencilla suele ser la más probable". En otras palabras, mientras más sencillo sea un modelo, es mejor.

Los métodos MCMC son buenos para estimar parámetros, pero ineficaces al calcular la *evidencia bayesiana*, por ello se han propuesto otros algoritmos, más eficientes, en este ámbito. Tal es el caso del muestreo anidado propuesto por John Skilling [5].

Muestreo anidado

La idea de John Skilling fue simplificar la integral de la *evidencia bayesiana* mapeando todo el espacio de parámetros a un intervalo unidimensional. El problema se reduce a calcular una integral de una dimensión; los contornos de las regiones con distinta probabilidad, ordenados, se mapean a un intervalo cerrado de 0 a 1 (Figura 3). Además, el muestreo se produce después de ordenar un primer muestreo más pequeño. Cada nuevo elemento de la muestra se compara con el de menor probabilidad y, en cada nueva iteración, el espacio de parámetros se reduce, aglomerando los nuevos puntos muestreados en una zona de probabilidad de alta ocurrencia.

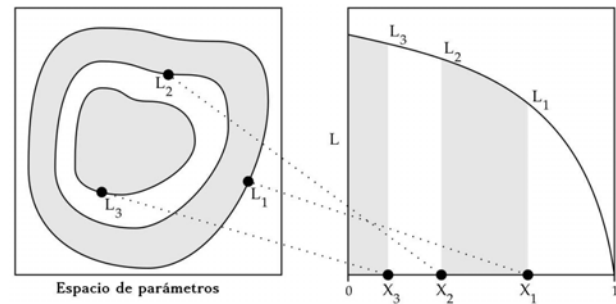


Figura 3. Propuesta de John Skilling. Fuente: [5].

El radiotelescopio SKA colectará 14 exabytes por día, suficientes datos para llenar 15 millones de iPods de 64 GB.

Las necesidades de la cosmología han motivado mejoras en el muestreo anidado. MULTINEST [6], por ejemplo, agrupa los puntos del muestreo mediante el algoritmo *k-means* y ha resultado ser bastante robusto, capaz de reducir el número de muestreos en uno o dos órdenes de magnitud respecto a los métodos MCMC [6] y por su éxito ha sido empleado en diversas disciplinas.

Optimizar el recorrido del espacio de parámetros, como hace MULTINEST con *k-means*, abre la puerta para que algoritmos de aprendizaje automático ayuden a mejorar las herramientas con las que se analizan los secretos del Universo.

El perceptrón multicapa (PMC)

Una de las arquitecturas de RNAs más simples y populares, desde hace un par de décadas, es el PMC (Figura 4).

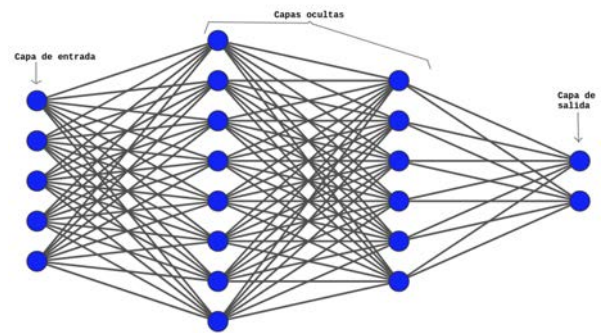


Figura 4. Arquitectura del PMC.

El Teorema Universal de Aproximación [7] establece que una RNA, con al menos tres capas, se puede aproximar a cualquier función, siempre que su función de activación esté localmente acotada, sea continua por partes y no sea un polinomio. Además, mientras mayor número de nodos posean sus capas ocultas más precisión tendrá la aproximación.

¹⁰<https://cosmologist.info/cosmomc>

¹¹<https://bitbucket.org/joezuntz/cosmosis>

Las funciones de activación asociadas al PMC (por ejemplo, sigmoide o \tanh) cumplen los requerimientos del Teorema Universal de Aproximación.

SKYNET

SKYNET, desarrollado por el mismo grupo de MULTINEST, es un algoritmo que consiste en una RNA capaz de realizar regresión, clasificación, estimación de densidad, agrupamiento y reducción dimensional. En el artículo donde los autores liberan el código [8], se prueba

en las tareas mencionadas mediante ejemplos de juguete y aplicaciones astrofísicas sencillas.

La RNA utilizada es un PMC de tres capas con una función de activación sigmoide. Los autores encontraron que el número óptimo de nodos en la capa oculta, para las estimaciones cosmológicas, está entre 50 y 100.

Una característica sobresaliente de SKYNET es su método de pre-entrenamiento: utiliza los muestreos de MULTINEST como entradas del PMC, lo cual reduce el tiempo de entrenamiento.

¿Qué hacer si dos modelos distintos se ajustan igual de bien a las observaciones? La respuesta es simple: compararlos.

En el caso de la estimación de parámetros, la función objetivo de la RNA es el logaritmo de la función de *verosimilitud*, dada por una función de ajuste chi-cuadrada. El algoritmo optimiza su función objetivo en cada iteración, evita sobre-ajuste y garantiza convergencia [8].

BAMBI

BAMBI (Blind Accelerated Multimodal Bayesian Inference) [9] es un algoritmo de los autores de MULTINEST y SKYNET. Su relevancia radica en que, para problemas típicos de cosmología, el tiempo de cálculo de la función de *verosimilitud* se reduce de segundos a milisegundos [9].

A diferencia de SKYNET, pretende sustituir en algún momento los cálculos resultantes del muestreo anidado. Sin embargo, se entrena con ellos y sus predicciones las va comparando con las respectivas salidas del muestreo anidado. Una vez que se alcanza la precisión establecida, prescinde del cálculo de las funciones de *verosimilitud* y emplea la RNA entrenada para etapas posteriores.

En problemas de estimación de parámetros simples, el tiempo que se pierde entrenando la RNA no es fructífero. No obstante, cuando se trata de funciones multivariadas, el tiempo invertido en el entrenamiento, se recompensa en una disminución del tiempo total. Más aún, mientras mayor complejidad tengan el modelo o los datos observacionales, más eficiente es BAMBI comparado con MULTINEST y la precisión de los resultados es del mismo orden.

Ejemplo

Para ilustrar y comprobar el funcionamiento de MULTINEST y BAMBI, hemos implementado un ejemplo sencillo mediante la interfaz en Python de [10]¹².

Elegimos el casquete de un paraboloides circular como función de objetivo (Figura 5):

$$z = f(x, y) = x^2 + y^2$$

Se siguieron las especificaciones de las Refs. [6,8,9]: BAMBI calculó el 75 % de las muestras mediante MULTINEST, y las restantes vía un PMC.

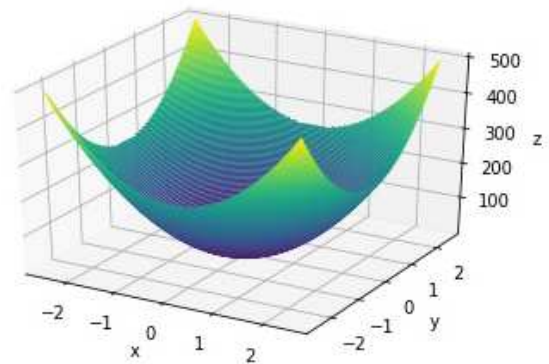


Figura 5. Paraboloides (función objetivo).

Una manera de validar los resultados de estos algoritmos y, en general, de cualquier método MCMC o de estimación de parámetros, es verificar si los puntos muestreados reproducen la función objetivo. Para este ejemplo (Figura 6), ambos algoritmos alcanzaron una precisión similar y satisfactoria. En la Figura 7, se puede apreciar la evolución del muestreo con BAMBI.

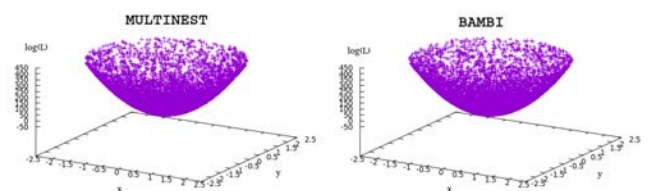


Figura 6. Resultados de MULTINEST y BAMBI.

¹²Se utilizó pyBambi: `pybambi.readthedocs.io`

...la cosmología mantendrá la mirada atenta en los avances de la inteligencia artificial.



Figura 7. Etapas inicial, intermedia y avanzada del muestreo con BAMBI.

También realizamos un comparativo entre los tiempos de ejecución con diferentes números de evaluaciones de la función de *verosimilitud* con ambos algoritmos (Tabla 1).

Tabla 1. Tiempos de ejecución para evaluaciones de la *verosimilitud*.

No. Eval.	Tiempo MULTINEST	Tiempo BAMBI
1 000	0.3 s	0.7 s
10 000	7.1 s	4.2 s
100 000	735.4 s	75.9 s

Nótese que mientras más se evalúa la función *verosimilitud*, mejor desempeño tiene BAMBI gracias a su RNA. En modelos complejos, como los cosmológicos, la función *verosimilitud* se debe evaluar millones de veces y métodos que optimicen los cálculos son necesarios.

Por último, en la Tabla 2 se muestra que al utilizar BAMBI no se pierde precisión en el cálculo de la *evidencia bayesiana*.

Tabla 2. Cálculos de la *evidencia bayesiana*.

	MULTINEST	BAMBI
Log(Evidencia)	$-0,782 \pm -0,06$	$-0,783 \pm -0,06$

Conclusiones

La estimación de parámetros de un modelo teórico mediante el muestreo anidado es una alternativa a los métodos MCMC y, además, permite calcular la *evidencia bayesiana* para comparación de modelos. Sin embargo, el uso de la RNA para evaluar las funciones de probabilidad mejora la velocidad y no pierde precisión.

La generación de datos cosmológicos seguirá a la alza en próximos años. Será crucial escudriñarlos para mejorar nuestra concepción del Universo y la cosmología mantendrá la mirada atenta en los avances de la inteligencia artificial. *

Agradecimientos. Al CONACYT, ICF-UNAM y CICATA-Legaria-IPN. Este trabajo fue parcialmente apoyado por los proyectos SIP 20180252 y SIP 20190237 del IPN. J.A.V. agradece el apoyo proporcionado a los proyectos FOSEC SEP-CONACYT Investigación Básica A1-S-21925 y UNAM-PAPIIT IA102219.

REFERENCIAS

- Hey, T., Tansley, S. y Tolle, K. (2014). Jim Gray sobre la eficiencia: un método científico transformado. En *Hey, T. et al. (Eds.) El cuarto paradigma. Descubrimiento científico intensivo en datos*. México, UAM, 17-23.
- Castro Ortega, P. (2017). El Teorema de Bayes. Aproximándonos a la verdad. *RBA Coleccionables*.
- Sivia, D. y Skilling, J. (2006). *Data analysis: a Bayesian tutorial*. OUP Oxford.
- Padilla, L., Tellez, L. O., Escamilla, L. y Vazquez J. A. (2019). Cosmological parameter inference with Bayesian statistics. arXiv:1903.11127 [astro-ph.CO].
- Skilling, J. (2006). Nested sampling for general Bayesian computation. *Bayesian analysis*. 1(4), 833-859.
- Feroz, F., Hobson, M. P. y Bridges, M. (2009). MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society*. 398(4), 1601-1614.
- Hornik, K., Stinchcombe, M. y White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*. 3(5), 551-560.
- Graff, P., Feroz, F., Hobson, M. P. y Lasenby, A. (2014). SKYNET: an efficient and robust neural network training tool for machine learning in astronomy. *Monthly Notices of the Royal Astronomical Society*. 441(2), 1741-1759.
- Graff, P., Feroz, F., Hobson, M. P. y Lasenby, A. (2012). "BAMBI: blind accelerated multimodal Bayesian inference". *Monthly Notices of the Royal Astronomical Society*. 421(1), 169-180.
- Buchner, J., Georgakakis, A., Nandra, K., Hsu, L., Rangel, C., Brightman, M. y Kocevski, D. (2014). X-ray spectral modelling of the AGN obscuring region in the CDFS: Bayesian model selection and catalogue. *Astronomy & Astrophysics*. 564, p. A125.

SOBRE LOS AUTORES



Isidro Gómez Vargas es licenciado en física y matemáticas y, en la actualidad, estudiante de doctorado en tecnología avanzada del IPN. Ganó segundo lugar del “Premio de Ensayo Innovación Educativa 2016” y en el “XIX Concurso Internacional de Cuento Navideño, Súbito, Breve y Electrónico”. En su doctorado estudia la estimación de parámetros cosmológicos mediante técnicas estadísticas y computacionales.



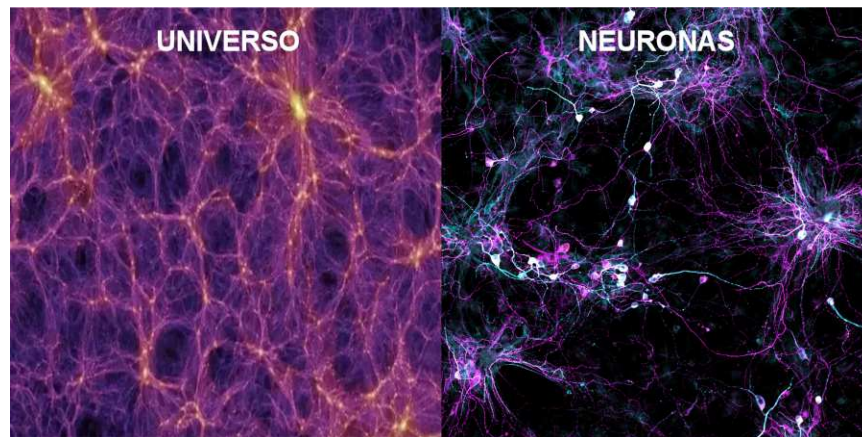
Ricardo Medel Esquivel es egresado de la Escuela Superior de Física y Matemáticas, actualmente cursa el Doctorado en Tecnología Avanzada en el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada Unidad Legaria del Instituto Politécnico Nacional, donde desarrolla un proyecto sobre análisis de datos cosmológicos. Ha publicado libros de texto y otros materiales educativos de nivel Secundaria en Ediciones Castillo. En 2018 el IPN le otorgó el Diploma a la Cultura, en reconocimiento a algunos de sus trabajos literarios y de educación.



Ricardo García Salcedo es profesor titular en el Departamento de Física Educativa en el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada Unidad Legaria del Instituto Politécnico Nacional. Licenciado y maestro en Ciencias por la Universidad Autónoma del Estado de México, doctorado en Ciencias por el Depto. de Física del CINVESTAV-IPN. Su investigación se centra en la cosmología, especialmente los modelos de energía oscura y TIC en la enseñanza de la Física.



J. Alberto Vázquez es investigador del Instituto de Ciencias Físicas (ICF) de la UNAM. Previo a su incorporación al ICF, fue catedrático CONACYT asociado al CINVESTAV, IPN. Estancia postdoctoral en el Laboratorio Nacional Brookhaven, NY. Doctorado y Maestría otorgados por el Kavli Institute for Cosmology de la Universidad de Cambridge, UK. Su investigación se enfoca en la exploración y análisis de observaciones cosmológicas, en el estudio de modelos de Energía oscura e Inflacionarios, a través de la estadística Bayesiana y el computo científico.



“La vida y la mente tienen un patrón abstracto común o un conjunto de propiedades organizacionales básicas. Las propiedades funcionales características de la mente son una versión enriquecida de las propiedades funcionales que son fundamentales para la vida en general. La mente es literalmente realista.”

—Godfrey-Smith, P. (1996). Complexity and the Function of Mind in Nature. *Cambridge University Press* —

ARTÍCULO ACEPTADO

Validación de un Sistema Tutor Inteligente Web para el aprendizaje de las Matemáticas

Alfonso Rosas Escobedo, Erica María Lara Muñoz y N. Sofía Huerta Pacheco

Resumen

Un sistema tutor inteligente (STI) es una herramienta diseñada como apoyo al proceso educativo, el cual utiliza técnicas de inteligencia artificial para brindar una enseñanza personalizada de acuerdo a las necesidades de cada estudiante. Estos sistemas actúan como si fuera un tutor humano el que interviniera en el proceso de enseñanza. Al diseñar estos STI, se debe tener cuidado de

integrar todas las características de inteligencia que permitan la transmisión del conocimiento que se pretende enseñar. El objetivo del presente artículo es mostrar la validación de un STI en su versión web, considerando un proceso de análisis jerárquico con la finalidad de corroborar si dicho sistema tutor cumple con las mismas funcionalidades y características que contempla el STI en su versión original de escritorio.

Los sistemas tutores inteligentes brindan la estrategia pedagógica de acuerdo a la necesidad de cada estudiante.

Introducción

El diseño de los STI varía considerando el dominio por el cual fue diseñado, sin embargo, aunque el dominio cambie, los demás elementos como el módulo tutor destinado a brindar la estrategia pedagógica, la ayuda cognitiva de acuerdo a las necesidades de cada estudiante [1], el *módulo estudiante* para captar la atención del mismo reconociendo sus fortalezas y debilidades, para adaptar los contenidos de acuerdo a su nivel de conocimiento [2], y el *módulo de la interfaz* por medio de la cual los estudiantes visualizan la información e interactúan con el sistema [3], son elementos que tiene todo STI en común, salvo alguna otra funcionalidad adicional que cada desarrollador requiera implementarle.

Estos sistemas tutores han nacido para su uso individual, sin embargo, algunos investigadores [4][5][6] han integrado un ambiente colaborativo para incrementar el rendimiento de grupos de estudiantes, así como para obtener resultados de aprendizaje de manera individual [7].

El diseño y desarrollo del STI web se dio debido a la necesidad de que los estudiantes pudieran trabajar desde cualquier sitio remoto, pero sobre todo, poder implementar una herramienta para su uso colaborativo desde una plataforma web. Para ello se analizó el funcionamiento y características de un sistema tutor inteligente de matemáticas en su versión original de escritorio, así, posteriormente irlo migrando a una plataforma web con las mismas funcionalidades, características y ejercicios que el sistema tutor original. El STI utilizado cuenta con dos versiones, la reactiva y la no reactiva, la primera contiene tanto un botón de ayuda como a un agente animado que reacciona de acuerdo a las respuestas que vaya seleccionando un estudiante, feliz si se contesta correcta-

mente, triste si se selecciona alguna respuesta incorrecta y enojado en caso de seleccionar varias veces respuestas incorrectas, entre algunos otros comportamientos. La versión no reactiva brinda el apoyo necesario al estudiante, tal y como lo hace la versión reactiva, con la diferencia que en lugar de reaccionar el agente, el estudiante debe buscar la respuesta en el botón de ayuda, la retroalimentación que el STI le brinda al estudiante se adapta al igual que en la versión reactiva, a las necesidades del estudiante y a las respuestas que este vaya contestando.

Descripción del procedimiento

Para el diseño y desarrollo del STI web, a partir del sistema tutor original en su versión no reactiva, se identificaron sus elementos, funcionamiento, manera de interactuar, se analizaron los módulos: dominio, tutor, estudiante e interfaz, el lenguaje en el que fue desarrollado y el código fuente, de esta manera se podría comenzar a desarrollar el nuevo STI web.

Posteriormente cuando el STI web estuvo desarrollado, para corroborar que este cumplía con las mismas funcionalidades y comportamiento que el STI anterior, se escribió una lista, a la cual se le llamó “requerimientos cumplidos”, estos no eran más que las funcionalidades que realizaba cada uno de los sistemas tutores inteligentes en ambas versiones. Para esto se realizó un proceso de análisis jerárquico [8], el cual es una técnica estructurada para la toma de decisiones por medio de comparaciones entre pares [9], en este caso se utilizó para conocer cuál de los dos sistemas tutores inteligentes (web y escritorio) cumplía con mejores funcionalidades que el otro, o si estos eran iguales o semejantes.

El proceso de análisis jerárquico permite validar la igualdad de funcionalidades de los sistemas tutores inteligentes.

En este proceso analítico se comenzó por clasificar los requerimientos de acuerdo a su funcionalidad, en este caso Administrador, General, Inicio e Interacción, una vez clasificados de esta manera, fueron evaluados por investigadores capacitados en el sistema tutor. La evaluación consistió en comparar entre dos requerimientos de cada una de las cuatro funcionalidades, cuál era el que tenía mayor importancia y cuál valor de intensidad entre 1 y 9, siendo 1 una preferencia de igualdad entre dos indicadores y 9 un valor de preferencia de nueve veces más grande o más importante que aquel con el que se está comparando. Los valores intermedios entre 2 y 8 van de moderado a fuerte y muy fuerte. Esta escala de clasificación del 1 a 9 se le denomina Escala de Saaty de comparación de juicios por pares para las preferencias del proceso de análisis jerárquico [10], mismas que se puede observar en la Tabla 1.

Tabla 1. Escala de Saaty de comparación de juicios por pares para las preferencias del proceso de análisis jerárquico.

Valor de intensidad	Importancia
9	A, extremadamente preferido que B
8	A, muy fuerte a extremadamente preferido que B
7	A, muy fuertemente preferido que B
6	A, fuertemente a muy fuertemente preferido que B
5	A, fuertemente preferido que B
4	A, moderadamente a fuertemente preferido que B
3	A, moderadamente preferido que B
2	A, igualmente a moderadamente preferido que B
1	A, igualmente preferido que B

Primeramente, la comparación se realizó entre los requerimientos del mismo STI en la funcionalidad Administrador con los siguientes tres (General, Inicio, Interacción), General contra los siguientes dos, e Inicio contra Interacción. Es decir, se tienen cuatro funcionalidades 1) Administrador, 2) General, 3) Inicio y 4) Interacción, en dónde la primera se comparó con 2, 3 y 4, la segunda con 3 y 4 y la tercera se compara con la funcionalidad 4. Este mismo procedimiento se hizo para ambos sistemas tutores inteligentes.

Una vez elaborada la clasificación y asignada la importancia e intensidad entre los requerimientos, se realizó la evaluación con los cálculos que propone este proceso analítico jerárquico, mismos que se describen en el siguiente apartado.

Evaluación de funcionalidades

La evaluación de los requerimiento a través del proceso de análisis jerárquico presentaron concordancias mayores al 84 % entre los investigadores capacitados en el Sistema Tutor Inteligente, es decir, que presentaron un alto acuerdo en el nivel de importancia de las funcionalidades y sus intersecciones contempladas en los requerimientos del mismo tutor. El proceso de análisis jerárquico permite validar la igualdad de funcionalidades de los sistemas tutores inteligentes.

Es importante destacar que para esta evaluación, las funcionalidades fueron ponderadas con base en la importancia e intensidad propuesta por Saaty en [11]. Para ello se realizó una matriz de comparación entre las cuatro funcionalidades (Administrador, General, Inicio e Interacción) considerando los juicios de valor de los expertos en el STI, una vez teniendo la matriz de pesos resultante, se normaliza. Con dicha matriz lo siguiente fue obtener el peso de cada criterio, esto se hizo obteniendo el valor promedio por cada fila de la matriz, los valores resultantes son el peso de cada criterio que determina la importancia de cada funcionalidad del STI.

El peso para Administrador fue de 0.4231, para General se obtuvo un 0.3330, Inicio alcanzó un 0.1685 e Interacción se quedó con 0.0754. Debido a que en ambos Sistemas Tutores Inteligentes las funcionalidades tienen igualdad de importancia, los pesos son considerados los mismos.

Una vez obtenida la importancia de cada funcionalidad de ambos tutores, se realizó la suma de los valores de intensidad que se obtuvieron de los requerimientos en cada funcionalidad donde los investigadores tuvieron concordancia, ésto se hizo para obtener una matriz de los pesos de los requerimientos por funcionalidad. Con esta matriz de pesos, se realiza la matriz normalizada para obtener los pesos de los requerimientos, debido a que varían un poco los requerimientos de cada STI, los pesos obtenidos fueron en el sistema web, para Administrador 0.6347, para general 0.2597, en Inicio se tuvo 0.1039 e Interacción se quedó en 0.0020. Para el caso de STI de escritorio, se obtuvo 0.5283 en Administrador, 0.4139 para General, en el caso de Inicio se tuvo un 0.0360 y en Interacción 0.0217.

Observándose que los investigadores muestran de igual forma un acuerdo considerable para poder establecer una ponderación a cada requerimiento definido para las dos versiones del Sistema Tutor Inteligente, donde se obtuvieron pesos de 0.3691 en la versión de escritorio y 0.3726 en la Web, las cuales presentan valores semejantes en las cuatro funcionalidades consideradas para ambos sistemas.

Es importante mencionar que la Figura 1, muestra las comparaciones por funcionalidad en la que se identifica que presentan consideraciones de importancia e intensidad mayores en Administrador (0.6347) e Inicio (0.1039) en el STI en la versión Web, y en la funcionalidad General (0.4139) e Interacción (0.0217) para el STI de escritorio.

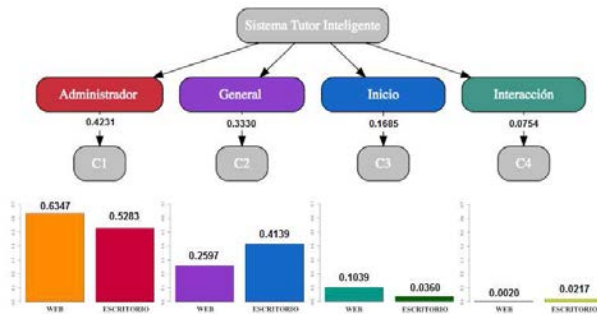


Figura 1. Comparaciones por funcionalidad.

Resultados

El entorno del STI original de escritorio en su versión no reactiva [12], al ejecutarse, muestra una serie de ventanas (Figura 2) como lo son el *escenario* (actividad), la *hoja de ayuda*, que es tomada como referencia para poder contestar el ejercicio presentado, la *herramienta tipo variable* en donde se deberá contestar los tipos de variables (categóricas o numéricas) que se están utilizando en la actividad y la *barra de avance*, en donde se puede observar como su nombre lo indica, el avance que se tenga en la actividad. El sistema muestra un signo de interrogación amarillo con la funcionalidad de ayuda, mismo que al presionar el usuario, le irá orientando para el avance de su actividad en caso de tener alguna duda referente al ejercicio en cuestión.

El entorno que muestra el STI web, una vez que se ingresa al mismo, es similar al STI de escritorio, simplemente que en lugar de que se ejecuten una serie de ventanas aisladas que no permiten identificar el inicio del ejercicio como es en el caso del STI de escritorio, en el STI web, la pantalla está organizada, considerando la secuencia de uso que el estudiante le dará al sistema para resolver su ejercicio, facilitando de esta manera la utilización del mismo. Los elementos que se presentan en el STI web son el escenario, la hoja de ayuda, la herramienta tipo variable y la barra de avance, como se muestra en la Figura 3.

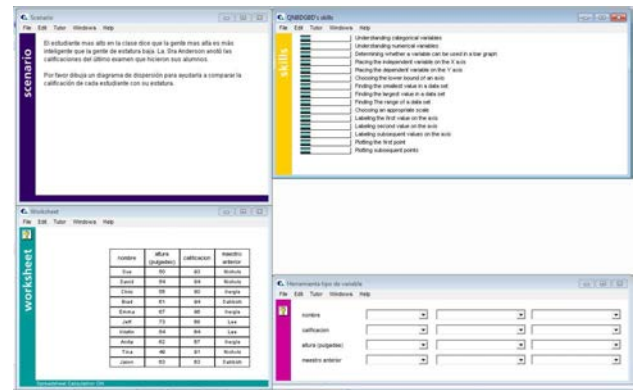


Figura 2. Entorno inicial STI de escritorio no reactivo.

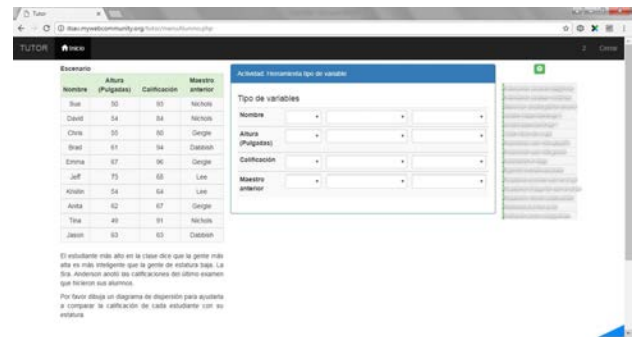


Figura 3. Entorno inicial STI web no reactivo.

En ambos STI al terminar la primera actividad del escenario, se van mostrando otras actividades que el estudiante deberá ir resolviendo para comprender y estudiar los ejercicios propuestos. Una vez que se concluyó el diseño y desarrollo del sistema tutor inteligente web y se comprobó que sus funcionalidades y comportamientos eran similares al STI original utilizando el proceso de análisis jerárquico [8], se procedió a probarlo con un grupo piloto.

Para el momento del desarrollo del STI web, el STI en su versión de escritorio, ya se había aplicado a un grupo de estudiantes [13], los cuales hicieron uso del mismo por pares, con la finalidad de realizar una actividad colaborativa, obteniendo resultados que mostraron que los estudiantes incrementaron sus calificaciones de manera individual con valor de $p.value < 0,05$ de un pre-test a un post-test, tal y como se observa en la Figura 4.

Al implementar el STI web a otro grupo de estudiantes con características homogéneas al anterior y aplicando el mismo procedimiento experimental, se observa que estos, de la misma manera incrementan su puntaje del pre-test al post-test, nuevamente obteniendo un valor de $p.value < 0,05$ (Figura 5). Comprobando que el desarrollo de este STI, de igual forma que el STI de escritorio, muestra resultados similares, incrementando los puntajes individuales de los estudiantes.

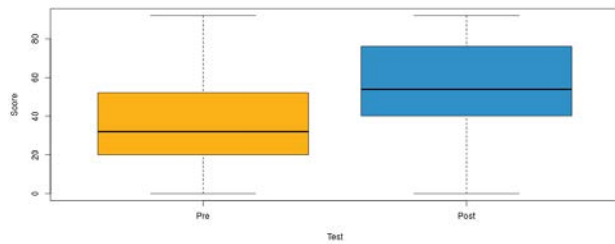


Figura 4. Resultados pre-test y post-test individual del STI de escritorio.

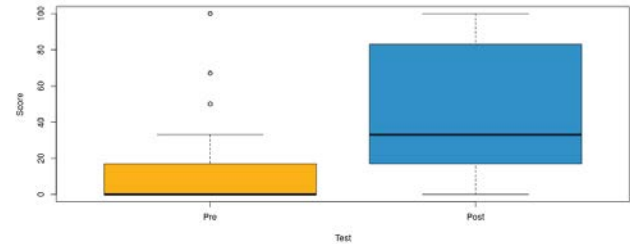


Figura 5. Resultados pre-test y post-test individual del STI web.

El desarrollo e implementación de sistemas tutores inteligentes, permite incrementar los resultados individuales de los estudiantes.

Conclusiones

Con base en el proceso de análisis jerárquico utilizado para calcular los pesos obtenidos por cada Sistema Tutor Inteligente (web y escritorio), se puede concluir que ambas versiones brindan semejanza en las funcionalidades de Administrador, General, Inicio e Interacción, siendo la funcionalidad de Administrador seguida por General las de mayor peso, ya que éstas presentan la definición de elementos relevantes antes de que se dé la interacción de los usuarios con el Tutor.

El desarrollo e implementación de sistemas tutores inteligentes, permite incrementar los resultados individuales de los estudiantes.

Se puede mencionar que los sistemas tutores brindan el apoyo fundamental para que los estudiantes o usuarios puedan adquirir e incrementar su conocimiento, de la misma forma, éstos pueden ser desarrollados considerando un STI ya elaborado y analizando su código, funcionamiento, características y sobre todo sus módulos: dominio, tutor, estudiante e interfaz, con la finalidad de que brinden el mismo soporte educativo y cumplan con el propósito por el que fueron desarrollados. *

REFERENCIAS

- Suárez, J., Arencibia, Y. y Pérez, A. (2016). Metodología para desarrollar un sistema tutor inteligente basado en la web, para estudiantes de ingeniería. *Revista Universidad y Sociedad*, 8(4), 108-115.
- Durango, J. y Pascuas, Y. (2015). Los sistemas tutores inteligentes y su aplicabilidad en la educación. *Revista Horizontes Pedagógicos*, 17(2), 104-116.
- Molina, Y., Pascuas, Y. y Millán, E. (2015). Sistemas Tutores Inteligentes como apoyo en el proceso de aprendizaje. *Redes de Ingeniería*, 6(1), 25-44.
- Harsley, R. (2015). When two heads are better than one: A critical review of four collaborative intelligent tutoring systems. *University of Illinois*, Chicago, IL USA, 1-24. Recuperado de <https://bit.ly/2IZLfgd>.
- Walker, E., Rummel, N. y Koedinger, K. (2009). Integrating collaboration and intelligent tutoring data in evaluation of a reciprocal peer tutoring environment. *Research and Practice in Technology Enhanced Learning*, 4(3), 221-251.
- Suebnuakarn, S. y Haddawy, P. (2004). COMET: A Collaborative Intelligent Tutoring System for Medical Problem-Based Learning. *IEEE Intelligent System*, 22(4), 14-21.
- Tchounikine, P., Rummel, M. y McLaren, B. (2010). Computer Supported Collaborative Learning and Intelligent Tutoring Systems. *Advances in Intelligent Tutoring Systems*. Springer, 308, 447-463.
- Saaty, T. (1990). How to make a decision: the analytic hierarchy process. *European Journal of Operational Research*, 48(1), 9-26.
- Márquez, L. y Baltierra, E. (2017). El proceso analítico jerárquico como metodología para seleccionar revistas científicas en el área biotecnológica. *E-Ciencias de La Información*, 7(2).
- Sutterfield, J., Swirsky, S. y Ngassam, C. (2014). Project Management Software Selection Using Analytical Hierarchy Process. *Academy of Information and Management Sciences Journal*, 4(6), 113-119.
- Saaty, T. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 1(1), 83-98.
- Baker, R. (2005). Designing Intelligent Tutors that adapt to when student game the system. Doctoral thesis, *Carnegie Mellon University*, Pittsburgh, U. S.
- Lara, E., Rebolledo, G. y Rojano, R. (2017). The influence of learning styles in collaborative activities. En *Proc. XVIII International Conference on Human Computer Interaction (Interacción '17)*. ACM, New York, USA, 33, 6.

SOBRE LOS AUTORES



Alfonso Rosas Escobedo es Maestro en Computación Aplicada por el Laboratorio Nacional de Informática Avanzada, Docente desde el 2010 de tiempo completo en el Instituto Tecnológico Superior de Alvarado, con reconocimiento como Profesor Perfil Deseable ante PRODEP, con más de 4 Proyectos financiados ante el TecNM, con áreas de interés en ingeniería de software, internet de las cosas e inteligencia artificial.



Erica María Lara Muñoz es Doctora en Ciencias de la Computación por la Universidad Veracruzana, y en Educación por el Centro de Estudios Superiores de Veracruz. Con maestría en Ciencias de la Computación y en Tecnología Educativa, Docente con Perfil Deseable PRODEP y Titular de tiempo completo en el Instituto Tecnológico Superior de Alvarado. Sus intereses de investigación incluyen áreas en Tecnología Educativa, Ingeniería de Software e Inteligencia Artificial en la Educación.



N. Sofía Huerta-Pacheco es Doctora en Ciencias de la Computación por la Universidad Veracruzana (2018) y Maestra en Estadística Aplicada (2015), con áreas de interés en cómputo afectivo en entornos educativos, análisis de información fisiológica en procesos cognitivos y enseñanza de la estadística con software libre.



“Las investigaciones realizadas en la Inteligencia Artificial y la educación se ocupan de la cuestión de qué es la experiencia y la forma en que se puede comunicar.”

—Lieberman, H. (1987). An example-based environment for beginning programmers. *Artificial Intelligence and Education: Learning environments and tutoring systems*. Eds. Lawler y Yazdani. 1(1), 135.—

ARTÍCULO ACEPTADO

Obtención de perfiles de tutores académicos utilizando técnicas de Soft Computing

M. Rosario Vazquez, Rafael Sánchez-Lara, Y. El Hamzaoui y Patricia Zavaleta-Carrillo

En la labor tutorial es indispensable que los tutores cuenten con ciertos rasgos de personalidad (amabilidad, concientización, apertura, entre otros) que les permitan desempeñar adecuadamente su labor como orientadores. En este trabajo se propone un método que extiende la funcionalidad del servicio Personality Insights de IBM para obtener los rasgos de personalidad del tutor. Se obtienen prototipos de perfiles de tutores usando el método de Prototipos Deformables Borrosos, que permite encon-

trar su afinidad con los rasgos de personalidad de cada tutor evaluado para describirlo adecuadamente. Se validaron los resultados arrojados por el método propuesto con los de un test de personalidad, obteniendo un error pequeño. El método propuesto servirá como una herramienta de apoyo para detectar fortalezas y áreas de oportunidad de los tutores, con el fin de mejorar el servicio tutorial proporcionado a los estudiantes.

Propuesta de método que extiende la funcionalidad del servicio Personality Insights de IBM para obtener perfiles de tutores académicos.

Introducción

La tutoría académica surgió como una alternativa de solución para reducir el rezago y abandono estudiantil. La tutoría funge como un acompañamiento al estudiante a lo largo de su vida académica en una institución de educación.

La función del tutor resulta de vital importancia, ya que consiste en orientar a los estudiantes asignados a su cargo a lo largo de su vida académica, así que es importante que cuente con ciertos rasgos de personalidad, como *amabilidad*, *responsabilidad*, *curiosidad*, *extraversión*, entre otros, que le permitan desempeñar adecuadamente su rol.

En este sentido, investigadores han enfocado sus esfuerzos para obtener el perfil del tutor, como [1], que, usando técnicas de machine learning, árboles de decisión, obtuvieron los principales atributos que debe poseer un tutor académico, que son: comunicación, autodirección y habilidades digitales. Por su parte, Goulart [2], basado en información existente en la literatura, presentó las habilidades y características de un buen tutor en el área de enseñanza de idiomas, como compartir su experiencia, desarrollo de relaciones interpersonales y comprender los aspectos que influyen en los tutorados. Finalmente concluyen que un buen tutor requiere dos elementos básicos: desafiar a los tutorados a pensar de manera crítica acerca de sus actividades y guiarlos en estos desafíos.

Brito, Durán y Sunza [3] identificaron las competencias del perfil de orientador, que debe cubrir el tutor a través de la experiencia de alumnos que colaboraron como tutores-pares en el programa de tutoría. Identificaron cuatro competencias: 1) Emocional, para manejar

de manera adecuada y eficaz las emociones ante determinadas situaciones y contextos. 2) Gestión de tareas, capacidad para diseñar y llevar a cabo tareas vinculadas con esta labor. 3) Aprender a aprender: capacidad para emprender y organizar un aprendizaje individual o grupal. 4) Competencias digitales: capacidad para usar las tecnologías de la Información y Comunicación.

Las investigaciones mencionadas describen el perfil, características y habilidades de tutores académicos, enfocados en el saber hacer, mientras que este trabajo se enfoca en el saber ser. Se parte de la hipótesis de que los rasgos de *Concientización* y *Amabilidad* en los tutores les permitirán establecer lazos de empatía y confianza con sus tutorados, de manera que puedan cumplir con su labor, potenciando las habilidades de los estudiantes.

Se propone un método para obtener la personalidad de los tutores académicos con base en los rasgos de *Concientización* y *Amabilidad*, para lo cual se hace uso del servicio Personality Insights de IBM [4]. Existen varias investigaciones que explotan las potencialidades de los servicios que ofrece IBM para el análisis de personalidad [5], [6]. Posteriormente, se obtienen prototipos de personalidad de los tutores usando el método de Prototipos Deformables Borrosos (PDB) [7]. Para obtener la personalidad de un nuevo tutor no considerado en el modelo, se evalúa su afinidad con los prototipos previamente definidos. Los prototipos se deforman de acuerdo a esta afinidad para describir adecuadamente la personalidad del tutor académico. Los PDB han sido ampliamente utilizados en la literatura: detección de enfermedades [8], predecir el rendimiento académico de los estudiantes que utilizan sistemas tutores [9], entre otras aplicaciones.

Los Prototipos Deformables Borrosos deforman los prototipos de perfiles para adaptarlos al perfil del tutor evaluado para describirlo.

Background

Se describe brevemente el servicio *Personality Insights de IBM*, técnicas de Softcomputing y el método *Prototipos Deformables Borrosos*.

Personality Insights de IBM

IBM Watson ofrece varios servicios, como *Personality Insights* [4], que permite obtener la personalidad de un individuo mediante análisis lingüístico de texto escrito en redes sociales, mensajería, foros, entre otros. Integra tres modelos: *Big Five*, *Needs* y *Values*. Este trabajo utiliza el *Big Five*, que agrupa los rasgos de personalidad de un individuo en cinco grupos: 1) *Extraversión*: capacidad de disfrutar la compañía de otros individuos. 2) *Amabilidad*: capacidad de un individuo de ser considerado, amigable y servicial. 3) *Concientización*: capacidad de un individuo para ser disciplinado, responsable y buen planificador del futuro. 4) *Neuroticismo*: mide la estabilidad emocional de un individuo. 5) *Apertura a experiencias*: describe qué tan abierto o cerrado es el pensamiento de una persona; individuos con una alta apertura son intelectualmente curiosos, se interesan por el arte y la ciencia, aprecian ideas inusuales, etc.

Técnicas de Soft Computing

En la computación blanda intervienen un conjunto de metodologías que explotan la tolerancia a la imprecisión e incertidumbre del mundo real. En la computación dura (tradicional) predomina la precisión, la certeza y el rigor. La lógica borrosa es una técnica que surgió como un formalismo para imitar la forma vaga e imprecisa en que los seres humanos razonan [11]; partió de la hipótesis de que el pensamiento humano está regido por etiquetas y no por números.

Con la Lógica Borrosa es posible modelar el cambio gradual de un estado a otro. Un conjunto borroso se define como una colección de objetos que pertenecen en cierto grado a la clase a la que representan. Cada conjunto borroso tiene asociadas funciones de pertenencia, que permiten calcular la pertenencia de un elemento a un conjunto. Estas funciones de pertenencia se representan gráficamente a través de Figuras geométricas, como triángulos, trapecios, sigma, entre otras. Por simplicidad, en este trabajo se utilizan las funciones de pertenencia triangular y trapezoidal (Figura 1).

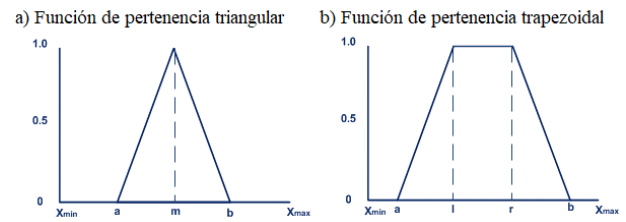


Figura 1. Funciones de pertenencia.

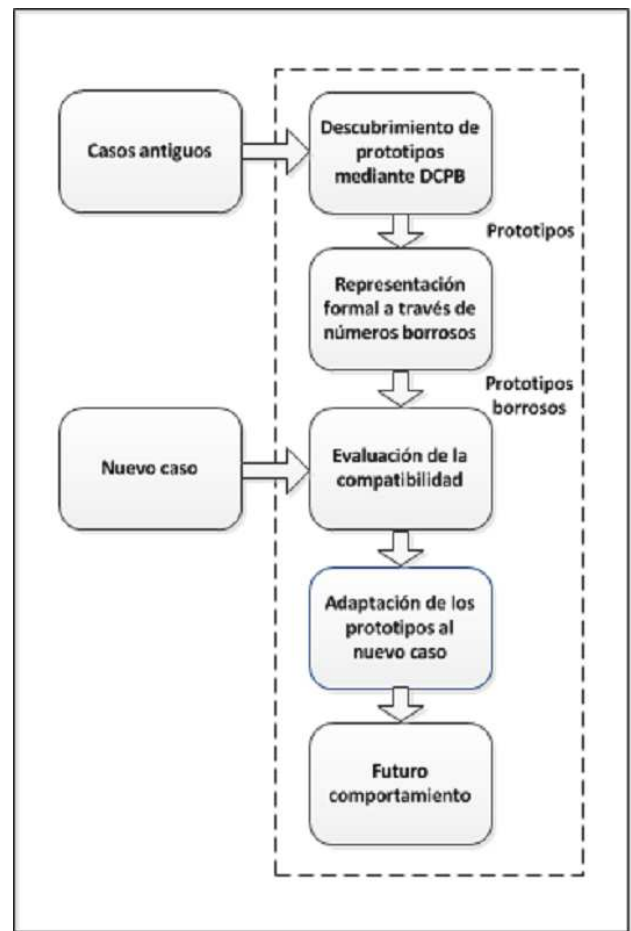


Figura 2. Diagrama de bloques del método Prototipos Deformables Borrosos.

Método Prototipos Deformables Borrosos (PDB)

El *PDB* [7], proveniente del Área de Inteligencia Artificial, permite deducir conocimiento a partir de casos históricos. Consiste en encontrar prototipos a través de conjuntos de datos. Un prototipo se puede definir como

un conjunto de datos que describen perfectamente a la clase a la que pertenecen. Para cada prototipo se definen sus respectivos conjuntos borrosos, que se deformarán de acuerdo a su afinidad con la nueva situación a evaluar.

En la Figura 2 se muestra el diagrama de bloques de los PDB. En ésta se observa que a partir de casos históricos se descubren prototipos mediante un proceso de extracción de conocimiento. Se definen conjuntos borrosos (representación borrosa) para cada uno de los prototipos descubiertos, que se deforman de acuerdo a su afinidad con la nueva situación a evaluar con el fin de describirlo completamente.

Propuesta para obtener el perfil del tutor

En la Figura 3 se muestra el diagrama de bloques del método propuesto. El tutor introduce información sobre la sesión tutorial, analizada lingüísticamente con el servicio *Personality Insights* de IBM [4]. Se definen prototipos borrosos para cada una de las sub-características de las características *Amabilidad* y *Concientización*. Los prototipos se deforman de acuerdo a su afinidad con los rasgos de personalidad del tutor a evaluar.

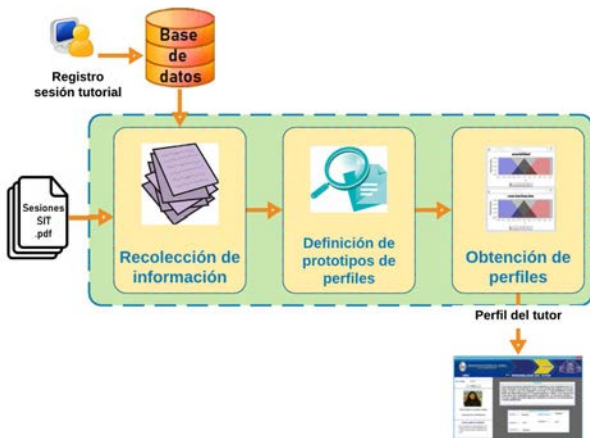


Figura 3. Arquitectura para la obtención de perfiles de los tutores.

Descripción de la arquitectura propuesta:

- **Recolección de la información:** se obtienen textos de las sesiones tutoriales. Estos provienen de reportes.
- **Definición de prototipos de perfiles:** en esta etapa, usando el servicio *Personality Insights*, se lleva a cabo un análisis lingüístico de los reportes de sesiones tutoriales. Se obtienen las representaciones borrosas de las características *Amabilidad* y *Concientización*, así como la afinidad de los rasgos de personalidad del tutor con cada una de las representaciones borrosas.

- **Obtención de perfiles:** se obtienen prototipos de perfiles a partir de los resultados que arrojan las evaluaciones del servicio tutorial que los estudiantes reciben. Se deforman los prototipos con base en la afinidad del tutor con cada uno de los prototipos obtenidos.

En este trabajo se obtuvieron los rasgos de personalidad de tutores académicos de la Facultad de Ciencias de la Información de la Universidad Autónoma del Carmen.

Recolección de información

Por lo general, los tutorados acuden a sesiones tutoriales con el fin de informar a su tutor sobre su rendimiento académico o cuando tienen alguna duda o inquietud. Los tutores registran en el Sistema Institucional de Tutorías (SIT-WEB) [11] los datos de la sesión, como matrícula y nombre del tutorado, fecha, hora de inicio, hora de finalización, motivo y la descripción de la sesión tutorial.

Los datos que alimentan al sistema propuesto provienen de los reportes del SIT-WEB [11] de agosto-diciembre de 2017. También se puede ingresar información de las sesiones directamente en el sistema desarrollado en este trabajo.

Definición de prototipos de perfiles

En el proceso de inferencia para obtener los rasgos de personalidad de los tutores (Figura 4), los datos de entrada son los registros de las sesiones tutoriales de un determinado tutor. Estos datos se transforman a formato JSON (JavaScript Object Notation), que permite almacenar información de una manera organizada y fácil de acceder. El servicio *Personality Insights* proporciona un archivo en formato JSON con las características de personalidad del tutor. Estos datos servirán de entrada para la siguiente etapa.

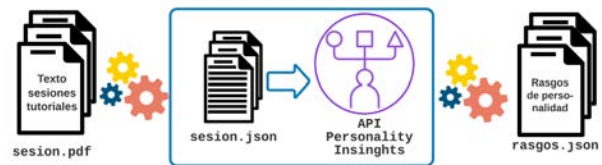


Figura 4. Componentes de la etapa de inferencia de la Personalidad.

Se utilizaron conjuntos borrosos para representar los rasgos de personalidad. Cada rasgo está definido por tres conjuntos borrosos definidos por: *Bajo*, *Promedio* y *Alto*. Cada conjunto permitirá calcular el grado en que un tutor cumple con cada característica. Estos conjuntos están definidos en el rango de cero a uno. Se hizo uso del conocimiento experto para definir los rangos de las representaciones borrosas para cada rasgo de personalidad.

En la Figura 5 se muestran las representaciones borrosas (conjuntos borrosos) para los rasgos de *Amabilidad* y *Concientización*.

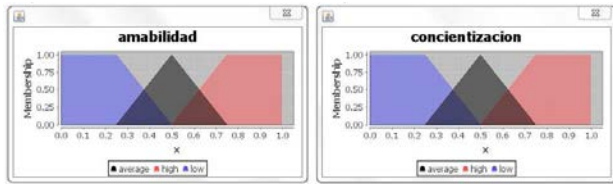


Figura 5. Conjuntos borrosos para los rasgos de personalidad.

En el proceso para obtener el perfil del tutor (Figura 6) se obtienen las representaciones borrosas de los rasgos a evaluar. Para cada rasgo del tutor se obtiene su afinidad con los conjuntos borrosos, que definen el nivel en que cumple con esa característica.



Figura 6. Proceso para la obtención de perfiles.

Obtención de perfiles

Se hace uso del proceso *Descubrimiento de Conocimiento Prototípico Borroso* (DCPB) para descubrir prototipos. El DCPB tiene las mismas etapas que el KDD (Knowledge Discovery in Databases); la diferencia radica en que en cada etapa del DCPB interviene el conocimiento experto (Figura 7).

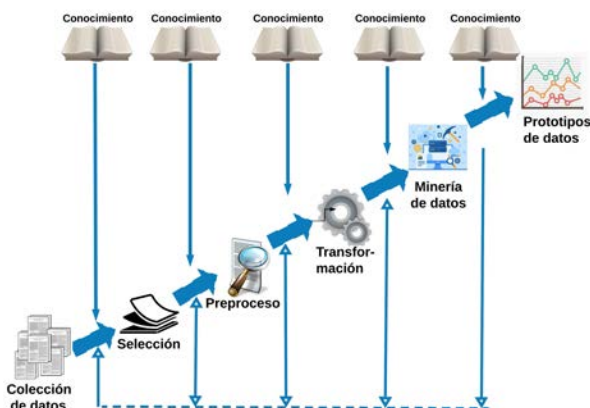


Figura 7. Proceso de Descubrimiento de Conocimiento Prototípico Borroso DCPB.

Los prototipos de perfiles del tutor se obtuvieron a través de los resultados arrojados en la evaluación que realizan los estudiantes sobre el servicio tutorial que reciben.

Cada rasgo a evaluar está compuesto por un conjunto de sub-características. Los prototipos definidos para todas las sub-características son *Bajo*, *Promedio* y *Alto*.

Para *Amabilidad* las sub-características son *Disposición* del tutor para atender a los tutorados, *Confianza* que le brinda al tutorado, *Respeto* y *Apertura* a la comunicación. Los valores de cada sub-característica están definidos en el rango de cero a uno. En la Figura 8 se muestran los tres prototipos descubiertos para *Amabilidad*.

Sub-características	Prototipos		
	Bajo	Promedio	Alto
Disposición	0-0.35	0.36-0.7	0.71-1
Confianza	0-0.4	0.41-0.79	0.8-1
Respeto y disposición	0-0.3	0.31-0.6	0.61-1
Abierto a la comunicación	0-0.3	0.31-0.7	0.71-1

Figura 8. Prototipos para la característica Amabilidad.

Las sub-características de *Concientización* son: interés del tutor por problemas académicos de sus estudiantes, facilidad con la que los estudiantes localizan a sus tutores, entre otros. En la Figura 9 se muestran los prototipos descubiertos para esta característica. En el prototipo *Bajo* caen los tutores con un interés por problemas académicos en un rango de 0 a 0.3, localizables en el rango de 0 a 0.35, y así sucesivamente.

Sub-características	Prototipos		
	Bajo	Promedio	Alto
Interés problemas académicos	0-0.3	0.31-0.7	0.71-1
Localizable	0-0.35	0.36-0.7	0.71-1
Detecta dificultades	0-0.4	0.41-0.7	0.71-1
Exhorta al estudio	0-0.35	0.36-0.7	0.71-1
Normatividad	0-0.35	0.36-0.65	0.66-1
Orienta carga académica	0-0.3	0.36-0.65	0.66-1
Canaliza	0-0.32	0.33-0.75	0.76-1
Seguimiento dificultades	0-0.32	0.33-0.7	0.71-1
Detecta situaciones de riesgo	0-0.3	0.31-0.6	0.61-1
Seguimiento AFIs	0-0.4	0.41-0.65	0.66-1
SS y PP	0-0.4	0.41-0.75	0.76-1
Programas de apoyo	0-0.4	0.41-0.8	0.81-1
Anomalías	0-0.37	0.38-0.68	0.69-1

Figura 9. Prototipos para la característica Concientización.

En la Figura 10 se muestran los rasgos de personalidad de uno de los tutores académicos evaluados. En la sección izquierda se muestra información del tutor, como: número de empleado, nombre, facultad de adscripción.

En la parte inferior de la sección se indica el número de palabras analizadas para obtener el perfil. La parte superior de la sección derecha contiene una descripción de los rasgos del tutor. En la parte inferior se indica el nivel que posee en cada uno de los rasgos evaluados.



Figura 10. Personalidad del tutor académico.

En la Figura 11 se observa que el tutor evaluado posee un nivel de *Neuroticismo* (Rango emocional) promedio, significando que su nivel emocional es equilibrado. Un alto nivel de *Conciencia*, es decir: alta capacidad para ser disciplinado. Un nivel de *Extraversión* promedio, que disfruta de manera equilibrada de la compañía de otras personas, así como de un ambiente de tranquilidad. Un alto nivel de *Amabilidad* significa que es considerado con los demás. Y un nivel promedio de *Franqueza*, es decir: curioso y abierto a nuevas experiencias de manera equilibrada.



Figura 11. Perfil del tutor académico.

Los resultados obtenidos en esta propuesta se validaron mediante el Error Cuadrático Medio (ECM) (Ecuación 1) de los errores obtenidos para cada rasgo respecto al arrojado en un test de personalidad [12] aplicado a los tutores evaluados, que implementa el modelo *Big Five* usando preguntas directas, mientras que *Personality Insights* lo obtiene a partir de análisis lingüístico de textos.

Para el tutor de la Figura 8, en *Amabilidad*, se obtuvo un error del 20 %, y para el de *Concientización*, 16 %.

$$EMC = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Donde: Y_i es un vector, que contiene los n perfiles calculados; \hat{Y}_i es un vector de los perfiles obtenidos de la evaluación de los estudiantes.

El método propuesto servirá como una herramienta de apoyo para detectar fortalezas y áreas de oportunidad de los tutores.

Conclusiones

La tarea del tutor académico es un desafío, ya que debe contar con ciertos rasgos de personalidad que le permitan desempeñar adecuadamente su labor. En este trabajo se utilizó la Lógica Borrosa para obtener perfiles de personalidad de los tutores académicos.

Se explotaron las bondades del servicio *Personality Insights*, para obtener los rasgos de personalidad de tutores a través del análisis lingüístico de reportes redactados en sesiones tutoriales. Se extendió la funcionalidad de este servicio, borrosificando los resultados arrojados, para obtener el grado de afinidad del tutor evaluado con cada uno de los prototipos de perfiles de tutores.

Usando los resultados de la evaluación del servicio tutorial, se descubrieron los prototipos mediante el método PDB. Éstos se deformaron con base en su afinidad con los resultados arrojados en el análisis lingüístico de las sesiones tutoriales del tutor evaluado, con el fin de

describirlo exactamente.

Los resultados obtenidos en este trabajo se contrastaron con los resultados que arrojó un test de personalidad aplicado a los tutores evaluados. Se obtuvo un margen de error pequeño, lo que indica que los resultados del método propuesto son próximos a la realidad.*

REFERENCIAS

1. Urbina, A.B. y De la Calleja, J. (2018). Selection of academic tutors in higher education using decision trees. *Revista Española de Orientación y Psicopedagogía*, 29, 108-124.
2. Goulart, L. (2017). From teacher to mentor: a case study on the development of mentoring skills. *Revista Odisseia*. 2(1), 55-71.
3. Durán, M., Sunza, S.P. (2017). Desarrollo de competencias del orientador a través de la tutoría entre pares. *Revista de Ciencias de la Educación*. Academicus. 1(11), 6-12.
4. IBM. (2015). "Personality insights". Recuperado el 21 de marzo de 2019 de <https://www.ibm.com/watson/services/personality-insights/>.

5. Pereira, R.A. y Inkpen, D. (2017). Using cognitive computing to get insights on personality traits from twitter messages. En *Malek Mouhoub and Philippe Langlais, (Eds.), Advances in Artificial Intelligence. Springer International Publishing.* 278–283.
6. Gain, U. y Hotti, V. (2017). Tones and traits - experiments of text-based extractions with cognitive services. *Finnish Journal of eHealth and eWelfare*, (9)(2-3), 82–94.
7. Olivas, J.A. (2000). Contribución al estudio experimental de la predicción basada en categorías deformables borrosas. Tesis de doctorado. Universidad de Castilla-La Mancha.
8. Romero, R., et al. (2016). An application of fuzzy prototypes to the diagnosis and treatment of fuzzy diseases: Diagnosis and treatment of fuzzy diseases. *International Journal of Intelligent Systems*. 32(2), 194–210.
9. Vazquez, M.R., et al. (2016). An approach to academic performance prediction in tutoring systems based on fuzzy deformable prototypes. *Progress in Artificial Intelligence*. 5(1), 55–64.
10. Zadeh, L.A. (1965). Fuzzy sets, *Information and Control*, 8(3), 338-353.
11. UNACAR. (2019). "SIT Sistema Institucional de Tutorías". Universidad Autónoma del Carmen. Recuperado el 21 de marzo de 2019 de <http://www.unacar.mx>.
12. Potter, J. (1996). "Test de Personalidad Los Cinco Grandes". Recuperado el 18 de febrero de 2019. De <http://es.outofservice.com/bigfive/>.

SOBRE LOS AUTORES



M. Rosario Vazquez obtuvo el grado de doctor en Informática en la Universidad de Castilla-La Mancha en Ciudad Real, España, en 2016. Actualmente es profesor investigador en la Universidad Autónoma del Carmen. Sus áreas de interés son: Aplicación de técnicas inteligentes en la educación.



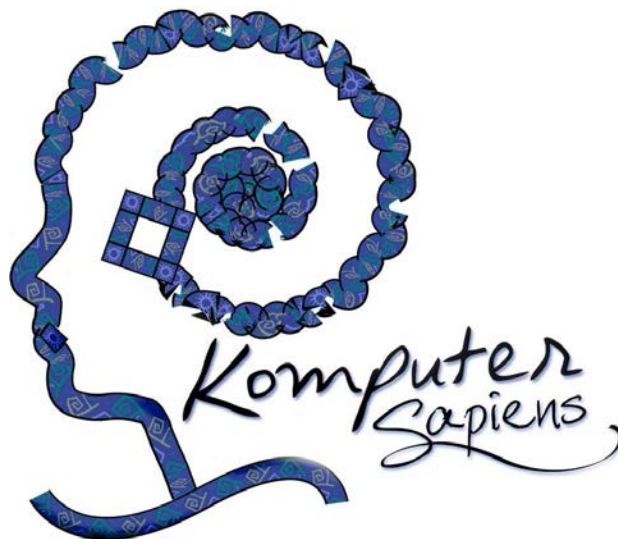
Rafael Sanchez-Lara obtuvo el grado de doctor en Tecnología Avanzada en el Centro de Investigación e Innovación Tecnológica (CIITEC) del Instituto Politécnico Nacional (IPN). Pertenece al Sistema Nacional de Investigadores con nivel candidato. Sus áreas de investigación de interés son: Fenómenos lineales en la fibra óptica y Sistemas de control borroso.



Youness El Hamzaoui obtuvo el grado de doctor en el Centro de Investigación, Ingeniería y Ciencia aplicada de la Universidad Autónoma de Morelos. Pertenece al Sistema Nacional de Investigadores con nivel I. Sus áreas de investigación de interés son: Machine learning, Deep learning, Algoritmos evolutivos, Inteligencia computacional y Estadística inferencial.



Patricia Zavaleta-Carrillo obtuvo el grado de maestra en Ciencias en Ciencias Computacionales en el Centro Nacional de Investigación y Desarrollo Tecnológico en Temixco, Morelos, en 2002. Actualmente es profesor investigador en la Universidad Autónoma del Carmen, Campeche. Sus áreas de investigación de interés son: Programación, Minería de datos y Enseñanza de la programación.



ARTÍCULO ACEPTADO

La agricultura de precisión impulsada por la Inteligencia Artificial

Rogelio B. Olvera Rosales, Julio C. Ramos Fernández, Marco A. Márquez Vera y Issac Hernández Santos

La Inteligencia Artificial (IA) en conjunto con la ingeniería mecatrónica han contribuido de manera esencial a la evolución de la Agricultura de Precisión (AP).

La Evolución de la Agricultura

La Oficina de Información Científica y Tecnológica para el congreso de la Unión [1] menciona que en la agricultura convencional se consideran las condiciones del terreno como homogéneas, y se aplican los insumos (fertilizantes, semillas, fungicidas, insecticidas, etc.) con base en valores promedio a toda la superficie de siembra. Ésto provoca un incremento en costos, tiempo y espacio. La AP tiene como principal característica el análisis y control de la variabilidad espacio-temporal del terreno y los cultivos. Con el estudio y aplicación de la AP se realizan suministros eficientes de los insumos y toma en cuenta la variación en los componentes del suelo como: acidez, textura, relieve, el desarrollo vegetal y las condiciones del ambiente en cada temporada.

Para trabajar en el campo con AP, se requieren tres etapas. Primero la recolección de datos, que se realiza con equipos de comunicación radiofrecuencia (RF) y satelital, sensores para medir variables físicas y químicas, unidades meteorológicas, vehículos aéreos y terrestres no tripulados. Después se realiza el análisis de datos; donde un humano o un sistema computacional experto analiza los datos y emite recomendaciones para el adecuado manejo de la variación espacio-temporal detectada. Finalmente, se implementan las estrategias, orientando al productor cómo hacer el trabajo agrícola según las sugerencias dadas por el sistema experto.

Las principales tecnologías que se usan en la AP son: (i) Sistemas de posicionamiento global (GPS por su sigla en inglés) se emplean para técnicas de guiado automático lo que permite mejorar la calidad del trabajo y las condiciones de guiado en los vehículos agrícolas. (ii) Sistemas de información geográfica (GIS por su sigla en inglés): son sistemas que incluyen hardware y software, permiten el almacenamiento de datos que se han referenciado para posteriormente procesarlos y mostrarlos a una ubicación geográfica. (iii) Sensores y electrónica especializada: son sistemas los cuales obtienen datos referentes al cultivo co-

mo son: la gestión del agua de riego, enfermedades de los cultivos, mapeo de malezas, detección de plagas. Para la eficiencia de estos sensores, deben estar bien calibrados y poseer suficiente resolución. (iv) Monitores de rendimiento: estos dispositivos obtienen la información de la cantidad y calidad del cultivo. (v) Maquinaria inteligente: son máquinas capaces de realizar de forma autónoma diferentes tareas, tales como asperjado, cosecha, fertilización que actualmente se hace con vehículos aéreos y terrestres no tripulados.

La evolución de la agricultura ha permitido la integración de sistemas mecatrónicos en los campos agrícolas, se han desarrollado equipos, herramientas, maquinaria y robots que ayudan a mejorar la producción agrícola en calidad y cantidad de productos agrícolas.

De Silva [2] define a la mecatrónica como un campo de ingeniería multidisciplinario que implica una integración sinérgica de las ingenierías: mecánica, eléctrica, electrónica, de control y ciencias de la computación. Las tecnologías para detección, accionamiento, acondicionamiento de señales, interconexión, comunicación y control son esenciales para los sistemas mecatrónicos, que se ha adaptado favorablemente al desarrollo y mejora de la AP.

Esto conlleva también al estudio y aplicación de las ciencias en computación, en especial un tópico que resalta es la IA, en este aspecto contribuye sustancialmente en el progreso de la AP con: sistemas expertos, lógica difusa, redes neuronales artificiales, reconocimiento de patrones, visión artificial y algoritmos genéticos. Por ejemplo, Zhu y colegas en [3] formulan un modelo de vehículo basado en redes neuronales para la estimación del comportamiento en terrenos agrícolas inclinados, para el aprendizaje del modelo utilizan algoritmos genéticos y el algoritmo de retro-propagación del error, con importantes resultados mostrados de manera experimental. También Noguchi [4], menciona el desarrollo de un método capaz de crear una ruta óptima de un robot tractor agrícola aplicando una técnica de control combinando una red

neuronal con un algoritmo genético, afirmando que este método debería ser aplicado a gran variedad de problemas de control no lineal en la agricultura. En el trabajo de Revathi y Sivakumaran [5], se menciona el uso de controladores difusos para mantener las condiciones del clima favorables en invernaderos, para proporcionar un ambiente controlado en la producción de plantas, con adecuados valores de humedad y temperatura demuestran cómo el controlador difuso mantiene el microclima en los valores deseados, sin un modelo matemático exacto.

La IA en un enfoque a sistemas que actúan racionalmente está relacionada con conductas inteligentes en artefactos, así menciona Nilsson [6]. Un enfoque propio y basado en la experiencia del desarrollo de sistemas inteligentes en la Universidad Politécnica de Pachuca (UP-PACHUCA), definimos a la IA como el razonamiento, la experiencia y las heurísticas del humano experto compiladas en código de programación y embebidas en una computadora, que actúa y toma decisiones como el experto.

Desde finales de los años 60 del siglo XX, los trabajos científicos y tecnológicos han sido dirigidos hacia el estudio de sistemas inteligentes para: robótica, navegación autónoma, visión artificial, control de temperatura, etc., apareciendo así la necesidad de incorporar a los sistemas agrícolas, una cantidad importante de conocimiento

específico para resolver los problemas cotidianos. Esto da como resultado el estudio de los sistemas expertos, que es una rama de la IA en la que se tiene mayor interés en el ámbito agrícola como lo menciona Vecino [7], donde define a los sistemas expertos como programas de cómputo que imitan el comportamiento de un humano experto, manipulando conocimiento codificado para resolver problemas de un dominio especializado con el fin de dar solución a una problemática dentro de un dominio concreto mediante deducción lógica de conclusiones y máquinas de inferencia.

Los beneficios que tiene la aplicación de la mecatrónica en conjunto con la IA han respaldado a la agricultura haciendo frente a las necesidades del nuevo mundo. Su uso ha reducido el tiempo de siembra y cosecha, el desperdicio de insumos y del espacio en el que se aplica, ha brindado mayor seguridad a los operarios de máquinas y ha elevado la calidad y cantidad de cultivos, entre otros aspectos.

El desarrollo de sistemas inteligentes para producir alimentos

En la actualidad el desarrollo de sistemas inteligentes en la agricultura contribuye de manera directa en la calidad y cantidad de los productos agrícolas, estos se ven beneficiados de manera constante y cada vez más por las innovaciones en la tecnología que se aplica en el campo.

Los países altamente desarrollados tienen como política de estado mejorar la calidad y cantidad de productos del campo, para esto utilizan técnicas de IA.

Sin embargo, pocas naciones han dado prioridad a esto y han desarrollado tecnología que lo permita. Un ejemplo de ellos es Japón, un país de grandes desarrollos científicos y tecnológicos, en manos del Dr. Noboru Noguchi, especialista en automatización con robots tractores para la agricultura de precisión, sus desarrollos científicos y tecnológicos en acompañamiento de sus estudiantes y colaboradores ha impulsado la modernización de la agricultura a nivel internacional con asociación de empresas como YANMAR Co., Ltd, que se dedican a la producción de maquinaria agrícola, han desarrollado una línea de robots tractores con grandes beneficios para los agricultores en Japón, así lo ilustra YANMAR [8]. Al igual por otra parte, en la Unión Americana las empresas CASE IH, New Holland y John Deere, han desarrollado maquinaria agrícola autónoma, con grandes beneficios para los agricultores.

En la UPPACHUCA, se desarrolla una línea de investigación con enfoque a la AP, con estudiantes y profesores especialistas en mecatrónica e inteligencia artificial, con

el objetivo de proponer soluciones científicas y tecnológicas que actualmente en México se requieren. Por ejemplo en Ramos-Fernández y colegas [9], ilustran el desarrollo de un sensor con técnicas de la IA en específico modelado difuso y Takagi-Sugeno, para medir el Déficit de Presión de Vapor (DPV), muestran que el sensor desarrollado es útil para prevenir problemas de regulación en la humedad relativa y temperatura dentro de un invernadero, todo esto con el fin de evitar enfermedades en los cultivos de agrícolas.

Actualmente en la UPPACHUCA se trabaja en el desarrollo de un prototipo de Robot Tractor (RT), utilizando la integración sinérgica de la mecatrónica con IA, donde se desarrollan algoritmos de navegación autónoma, procesamiento de imágenes, para lograr un RT plenamente autónomo, como los que fabrican países altamente desarrollados. Se utiliza tecnología de alta precisión como RTK-GPS (Real Time Kinematics por su sigla en inglés) con un error no mayor a un centímetro, equipos de medición para controlar la navegación y ope-

ración en el seguimiento de trayectorias predefinidas en la agricultura, en este proyecto se estudian y desarrollan algoritmos de control con técnicas de la IA.

En la Figura 1, se observa un esquema del funcionamiento básico para el seguimiento de trayectorias del sistema que se desarrolla en la UPPACHUCA. El primer paso es el desarrollo de un algoritmo para adquisición

de coordenadas a una trayectoria deseada, con puntos establecidos en la parcela, posteriormente se realiza el aprendizaje con algoritmos de control con técnicas de la IA, que reproducen los movimientos deseados (planificados) de forma autónoma para el robot tractor, esto es el posicionamiento en cada punto marcado con coordenadas específicas utilizando los datos del GPS Rover.

Hay una gran reto para las universidades y centros de investigación para la formación de ingenieros e investigadores que proporcionen soluciones a costos accesibles.



Figura 1. Diagrama básico del seguimiento de trayectoria del RT.

Otro proyecto que se desarrolla en la UPPACHUCA asociado al Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos (LANAVEX-CONACYT), es un robot móvil tipo diferencial de tracción electro-mecánica, con una capacidad de carga útil de cien kilogramos, para realizar tareas dentro de invernaderos agrícolas, esto para ayudar a resolver los problemas con inspección y aplicación de agroquímicos preventivos y curativos a la producción agrícola en espacios cerrados. Estas tareas normalmente se realizan con equipo manual y son efectuadas por un humano, que en ocasiones arriesgan su salud por contacto físico o inhalación de químicos. De igual manera la producción agrícola se ve afectada, ya que las plantas no reciben los tratamientos en el tiempo adecuado, por estas razones el robot móvil tendrá la capacidad de seguir trayectorias definidas dentro del invernadero y proporcionará la correcta aplicación de los tratamientos preventivos. Otra tarea es la polinización artificial con aire forzado, así como supervisar con técnicas de visión artificial, tareas repetitivas que son necesarias en la producción de cultivos agrícolas.

En la Figura 2, se muestra un esquema del diseño asistido por computadora (CAD) del robot móvil tipo diferencial.

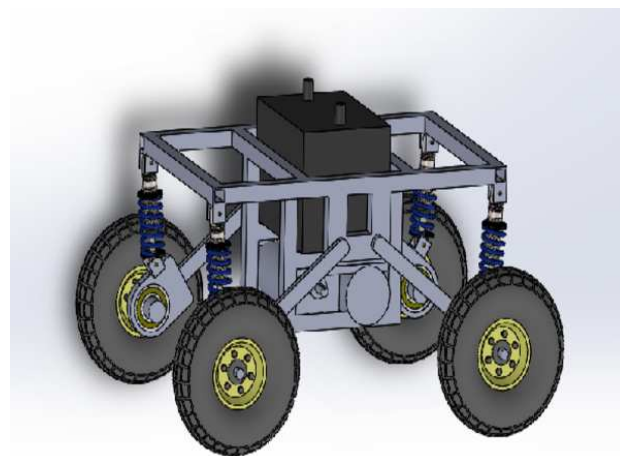


Figura 2. Esquema CAD de robot móvil diferencial.

Limitantes de la AP en la sociedad Mexicana

La AP en México tiene serias limitaciones, algunas son: la falta de especialización de profesionales y operarios, mínimo avance científico y tecnológico interdisciplinario desarrollado por mexicanos, los altos costos de la tecnología agrícola y servicios para su mantenimiento, entre otras que existen en el país las cuales no permiten adoptar plenamente la filosofía de la AP.

En lo social y económico, un porcentaje importante de agricultores en México no están de acuerdo en introducir la AP en sus campos agrícolas, creen que es una técnica innecesaria, y que no impactaría en mucho su adopción. Los agricultores de pequeñas extensiones están acostumbrados a trabajar a su manera, y ven la AP como un rival que puede desplazar la mano de obra del trabajador en el campo. Y aunque algunos ven esto a largo plazo, la mayoría de agricultores duda mucho del potencial que esta

tecnología puede aportar, la personas están muy arraigadas a que el trabajo agrícola debe ser pesado, de alguna otra forma no ven los beneficios que la AP provee.

Una muestra del avance que proponen investigadores de la UPPACHUCA en la robótica móvil aplicada a la AP se puede apreciar en [10], donde se muestra el desarrollo de un RT utilizando técnicas de la mecatrónica en conjunto con la IA, aplicando modelado y control difuso a los diferentes servomecanismos del RT. Dada la variedad de condiciones que se encuentren en el campo no existe un solo sensor que pueda garantizar la confiabilidad en cada situación ambiental. Es por ello que los temas de investigación a resolver en el prototipo de robot móvil son: sistemas de visión artificial para detectar obstáculos en la trayectoria, reconocimiento de plagas y la calidad en producción agrícola, así como posibles riesgos civiles mediante sistemas de aprendizaje, por ejemplo, redes neuronales artificiales, redes neuro-difusas que sean evaluadas con la ayuda de expertos en agricultura, para realizar aplicaciones reales en campos agrícolas con implementos mecánicos y la consideración de utilizar a futuro sensores de imagen con procesamiento por estereovisión, detección del entorno de navegación con luz láser (LIDAR por su acrónimo al inglés), el objetivo es lograr un prototipo de RT con sistemas multisensoriales e inteligentes.

Agradecimientos

Los autores agradecen al Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos (LANAVEX-LN299146) y al Consejo Nacional de Ciencia y Tecnología por el apoyo técnico y económico recibido para el desarrollo de proyectos en AP, así como a los revisores de la revista Komputer Sapiens por sus valiosos comentarios para la redacción del presente trabajo de divulgación científica.

Conclusiones

Con la revolución industrial mejoró la cantidad de producción y transporte. Actualmente, el desarrollo científico, como se presentó en la revisión bibliográfica, muestra el uso de la robótica, la automatización y las computadoras han sido de mucha utilidad en la producción y calidad de los alimentos provenientes del campo.

Así es como la AP se ha fortalecido con la aportación de las ciencias de la computación, los algoritmos de la IA juegan un rol importante prácticamente en todas las ciencias, sin embargo en la agricultura aún hay un camino extenso por recorrer y sobre todo en la formación académica y profesional, los jóvenes de provincia, hijos

de agricultores que estudian en las ciudades, en una mínima escala regresan a las labores agrícolas, es por esto que la forma de hacer la agricultura debe cambiar mediante el uso de la robótica, la inteligencia artificial y el uso de las tecnologías de la información, para que los jóvenes retomen las labores agrícolas con el enfoque de la tecnología.

Por esta razón y debido a que la producción agrícola es un problema nacional, en la UPPACHUCA se realizan proyectos de investigación, para dar solución a los retos de la AP. Uno de los proyectos realizado automatiza un tractor, asimismo se desarrolla un robot móvil diferencial para su uso en producción agrícola protegida. También se ha contribuido con el desarrollo de sistemas para el control bio-climático en invernaderos usando IA, así como el desarrollo de algoritmos con procesamiento de imágenes para reconocer surcos de maíz, con lo que se fortalece la formación de recursos humanos especializados para brindar soluciones mediante aplicación de la mecatrónica en la AP. *

REFERENCIAS

1. Oficina de Información Científica y Tecnológica para el Congreso de la Unión (2018) "Agricultura de Precisión". *Foro Consultivo*, Nota INCyTU, 15, 1-6.
2. De Silva, C.W. (2008). *Mechatronic systems: devices, design, control, operation and monitoring*. Primera edición. Editorial CRC Press, 492.
3. Zhu, Z.X., Torisu, R., Takeda, J.I., Mao, E.R. y Zhang Q. (2005). Neural Network for estimating Vehicle Behaviour on Sloping Terrain. *Biosystems Engineering*, 91(4), 403-411.
4. Noguchi, N. y Terao, H. (1997) Path planning of an agricultural mobile robot by neural network and genetic algorithm. *Computer and Electronics in Agriculture*, 18, 187-204.
5. Revathi, S., Sivakumaran, N. (2016). Fuzzy Based Temperature Control of Greenhouse. *IFAC-PapersOnLine*, 49, 549-554.
6. Nilsson, N.J. (1998). *Artificial Intelligence A New Synthesis*. Morgan Kaufmann, 513 p.
7. Berbel-Vecino, J. (1989). La inteligencia artificial en la agricultura: perspectivas de los sistemas expertos. *Revista de estudios Agro-sociales*, 149, 61-77.
8. YANMAR (2017) "Will the Robot be Our Savior? The future of agriculture changed by technology". Recuperado el 19 de Marzo de 2019, de https://www.yanmar.com/global/about/ymedia/product/agri_robot.html
9. Ramos-Fernández, J.C., Balmat, J.F., Márquez-Vera, M.A., Lafont, F., Pessel, N. y Espinoza-Quesada, E.S. (2016). "Fuzzy Modeling Vapor Pressure Deficit to Monitoring Microclimate in Greenhouses". *IFAC-PapersOnLine*, 49(16), 371-374.
10. LANAVEX-CONACyT UPPACHUCA (2019). "Robot Tractor Autónomo". Recuperado el 5 de Julio de 2019, de <https://www.youtube.com/watch?v=jwFgIRw1rLI&t=1s>.

SOBRE LOS AUTORES



Rogelio Benjamín Olvera Rosales es Ingeniero en Mecatrónica egresado de la Universidad Politécnica de Pachuca, actualmente estudiante de la maestría en mecatrónica en la Universidad Politécnica de Pachuca, realiza su proyecto de investigación en el Laboratorio Nacional en Vehículos Autónomos (LANAVEX) de CONACYT en la UPPACHUCA asociado con el CINVESTAV, sus intereses científicos incluyen el desarrollo de algoritmos para la navegación autónoma de un robot tractor, utilizando RTK-GPS, así como el modelado y control neuro-difuso para su uso en el campo agrícola.



Julio C. Ramos-Fernández es Doctor en ciencias de la computación por la Universidad Autónoma del Estado de Hidalgo, Dr. en Ciencias de la Ingeniería con aplicaciones Industriales por la Université de Sud Toulon Var-Francia, es responsable técnico del Laboratorio Nacional en Vehículos Autónomos (LANAVEX) en la UPPACHUCA asociado con el CINVESTAV, sus intereses científicos incluyen el desarrollo de nuevos algoritmos para el control autónomo de robots tractores, así como el modelado y control neuro-difuso para invernaderos agrícolas.



Marco Antonio Márquez Vera es Profesor Investigador en la maestría en mecatrónica, cuenta con una maestría en Ciencias en Automatización y Control con especialidad en sistemas no lineales, su área de investigación abarca temas de soft-computing como modelado y control difuso, aprendizaje profundo y sistemas bio-inspirados, así como electrónica de potencia con aplicaciones en biotecnología.



Issac Hernández Santos Ingeniero en mecatrónica egresado de la Universidad Politécnica de Pachuca, actualmente estudiante de maestría en mecatrónica en la Universidad Politécnica de Pachuca en el Laboratorio Nacional en Vehículos Autónomos (LANAVEX) de CONACYT, asociado con el CINVESTAV, su interés científico incluye el desarrollo de nuevos algoritmos para el control autónomo de robots móviles del tipo diferencial.



Inteligencia Artificial Emergente en la agricultura

— Imagen tomada de <https://www.technologytimes.pk/2019/07/15/artificial-intelligence-agriculture/>.—

ARTÍCULO ACEPTADO

Pronóstico de Velocidad de Viento por Medio de un Conjunto de Modelos

Hector Rodriguez-Rangel, Juan R. Valenzuela-Barraza, Carlos Lara-Alvarez, Luis A. Morales Rosales y Giovanni Manjarrez-Montelongo

En la actualidad la energía eléctrica tiene gran importancia en las actividades realizadas por la humanidad; pero más de dos tercios (73.5 %) se produce a partir de combustibles fósiles [2]. Una alternativa es la energía generada por medio de recursos renovables. Sin embargo, estos recursos renovables utilizados en la producción de

energía eléctrica, no son constantes. Por lo que la cantidad de energía a suministrar se vuelve incierta.

Se han buscado mecanismos [1] que permitan efficientar la producción de energía por medios renovables para poder hacer costeable su producción y así, facilitar su adopción.

Una buena estimación permite saber con antelación la cantidad de energía a suministrar

Un buen pronóstico permite determinar los plazos de mantenimiento; una buena sincronización con las demás plantas productoras de energía. Además se pueden reducir los costos de producción y el impacto negativo al medio ambiente.

Obtener buenas predicciones de la velocidad del viento es una tarea complicada; ya que el viento presenta un comportamiento caótico que es difícil de predecir. Este artículo explora un ensamble de técnicas de aprendizaje como redes neuronales (ANN) y k vecinos más cercanos (k-NN) y métodos estadísticos como regresión lineal (LR) y bosques aleatorios (RF) para obtener una mejor predicción.

Después de realizar la predicción con cada técnica, los pronósticos se fusionan para mejorar la predicción. Se evaluaron diferentes técnicas de fusión: i) mediana aritmética, ii) media aritmética y iii) agrupamiento espacial. Los resultados muestran que el error cuadrático medio de los nuevos pronósticos es significativamente mejor que el error obtenido por los métodos por separado.

Antecedentes

El pronóstico de series de tiempo es útil en diferentes áreas de la investigación. En el área de energía eólica es especialmente útil porque ayuda a reducir el riesgo de falta de fiabilidad del suministro eléctrico. Muestra de esto; es que en la literatura se reporta una amplia gama de opciones para la velocidad del viento.

Chang [3] describen diferentes enfoques de predicción de la velocidad del viento como: persistencia, físicos, estadísticos, correlación espacial, inteligencia artificial y métodos híbridos. La predicción se divide en cuatro categorías: muy corto, corto, mediano y largo plazo [3]. Cada método tiene una mejor aptitud que otro dependiendo

estas categorías, y de su aplicación.

El método de persistencia [4] supone que la velocidad del viento en un determinado momento futuro será la misma que cuando se hace la predicción. Este método es de alguna manera más exacto que otros métodos de pronóstico del viento en periodos a muy corto plazo, pero la precisión de este método se degrada rápidamente cuando la escala de tiempo se aumenta [4].

Los métodos físicos se basan en pronóstico numérico de clima o por su nombre en inglés *Numerical Weather Prediction* (NWP), y utilizan datos como temperatura, presión, rugosidad de superficies y obstáculos [3].

Los métodos de correlación espacial [5] consideran la relación espacial de la velocidad del viento de diferentes sitios. En dichos métodos, se emplean las series temporales de la velocidad del viento del punto predicho y sus puntos vecinos para predecir la velocidad del viento.

Metodología Propuesta

En esta metodología, el pronóstico se inicia normalizando los datos para organizarlos y formatearlos adecuadamente para su almacenamiento en una base de datos. Los modelos predictivos (ANN, kNN, etc) toman estos datos para su entrenamiento, generando un pronóstico por cada modelo. Con los pronósticos resultantes se establece un mecanismo de fusión con la finalidad de mejorar el pronóstico individual. La Figura 1 muestra el proceso realizado en el pronóstico de velocidad de viento utilizado en este trabajo.

Preprocesamiento de Datos

El objetivo de realizar la normalización de la serie de tiempo es tener los valores de la serie de tiempo en un mismo rango; esto permite que sean medibles para el pro-

ceso de entrenamiento y validación en el rango entre 0 y 1.

La serie de tiempo normalizada $X_t = (x_1, \dots, x_n)$ se obtiene con (1)

$$y_n = \frac{x_n - x_{min}}{x_{max} - x_{min}} \quad (1)$$

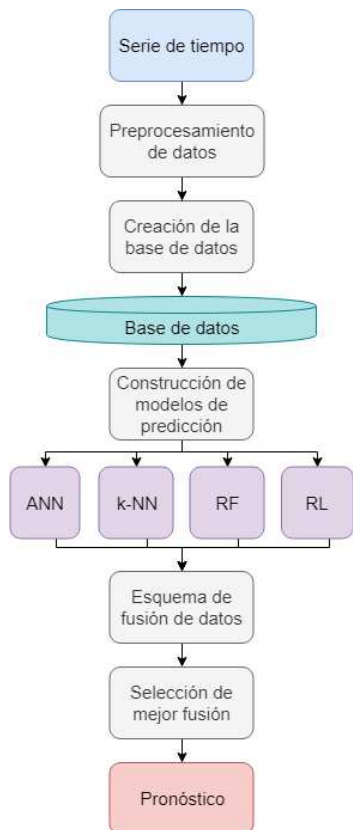


Figura 1. Proceso del pronóstico de series de tiempo de la velocidad del viento.

donde y_n es el valor normalizado resultante, x_n el valor original de la serie de tiempo, y_{max} es el valor máximo

de la muestra de datos de la serie de tiempo y y_{min} el valor mínimo de la muestra de los datos de la serie de tiempo.

Teniendo los datos normalizados, la serie de tiempo se almacenará en una base de datos para ser utilizados en el proceso de entrenamiento de los modelos de aprendizaje mencionados anteriormente.

Creación de la base de datos

Regularmente, en la tarea de pronosticar series de tiempo se tiene un vector de retardo que se utiliza como entrada al modelo predictivo, y este a partir de esa entrada genera una salida. Este vector se toma de una ventana de tamaño m , la cual se desliza a través del tiempo en la serie de tiempo. Al trasladar los vectores de retardo obtenido junto con su valor predictivo a un registro de la base de datos se elimina el factor tiempo. Dado que los registros pueden cambiar de posición y no importa el tiempo en el que fueron recolectados. La Tabla 1 muestra como se procesa la serie de tiempo a una base de datos y extrae las a características asociadas con su valor predictivo.

Tabla 1. Transformación de la serie de tiempo en registros almacenados en una base de datos.

	Entradas					Salida	
	$P(m) = a_{t+1}$						
1	$a_1,$	$a_2,$	$a_3,$	$a_4,$	\dots	a_m	a_m
2	$a_2,$	$a_3,$	$a_4,$	$a_5,$	\dots	a_{m+1}	a_{m+1}
3	$a_3,$	$a_4,$	$a_5,$	$a_6,$	\dots	a_{m+2}	a_{m+2}
4	$a_4,$	$a_5,$	$a_6,$	$a_7,$	\dots	a_{m+3}	a_{m+3}
5	$a_5,$	$a_6,$	$a_7,$	$a_8,$	\dots	a_{m+4}	a_{m+4}
6	$a_6,$	$a_7,$	$a_8,$	$a_9,$	\dots	a_{m+5}	a_{m+5}
7	$a_7,$	$a_8,$	$a_9,$	$a_{10},$	\dots	a_{m+6}	a_{m+6}
8	$a_8,$	$a_9,$	$a_{10},$	$a_{11},$	\dots	a_{m+7}	a_{m+7}
9	$a_9,$	$a_{10},$	$a_{11},$	$a_{12},$	\dots	a_{m+8}	a_{m+8}
10	$a_{10},$	$a_{11},$	$a_{12},$	$a_{13},$	\dots	a_{m+9}	a_{m+9}
\dots							
n	$a_{n-w-1},$	$a_{n-m},$	$a_{n-m+1},$	$a_{n-m+2},$	\dots	a_{m-1}	a_m

Los métodos implementados en este trabajo fueron redes neuronales artificiales, k vecinos más cercanos, regresión lineal y bosques aleatorios

Pronóstico mediante técnicas de Inteligencia Artificial

Cada método fue implementado de manera individual para posteriormente realizar una fusión de datos con los valores pronosticados con cada técnica. La Figura 2 especifica el proceso del pronóstico con cada modelo de predicción. Donde a partir de los datos en la base de datos, se construye el modelo de aprendizaje, para posteriormente, entrenar ese modelo.

Pronóstico con ANN

La arquitectura utilizada en la red neuronal es un *Feed-forward Multilayer Perceptron* (MLP); entrenada por métodos basados en gradiente. En la red neuronal se tiene un conjunto de m observaciones pasadas que se consideran como los datos de entrada, la capa oculta consta de h neuronas, la capa de salida corresponde al valor predictivo \hat{y}_{t+1} . La función de activación utilizada fue la sigmoideal [6]. En la Figura 3 se observa la estructura general de una red neuronal MLP.

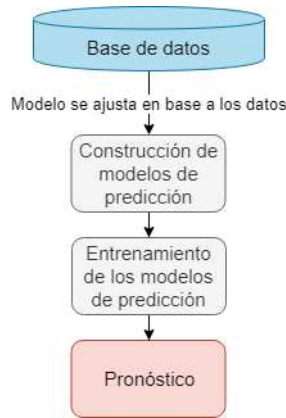


Figura 2. Proceso del pronóstico de series de tiempo de la velocidad del viento aplicando cada pronosticador.

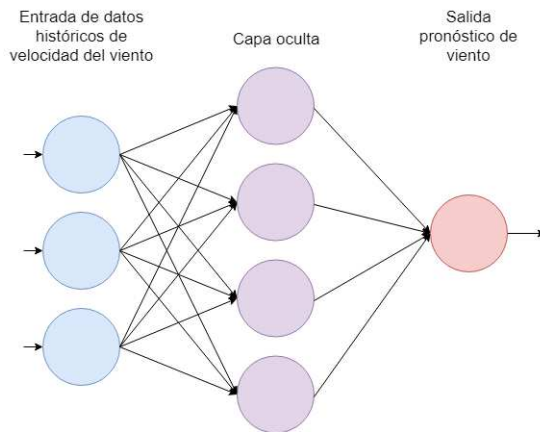


Figura 3. Representación general de red neuronal artificial MLP.

Para encontrar la mejor topología de la red neuronal se aplicó una búsqueda en malla iterando las entradas y las neuronas en la capa oculta en un rango de 2 a 41.

Pronóstico con k -NN

El método de aproximación k -NN (k vecinos más cercanos) es muy simple, pero poderoso. Se ha usado en muchas aplicaciones, particularmente en tareas de clasificación [7]. Este algoritmo es una técnica de clasificación y regresión basada en la similitud de los individuos de una población. La premisa de esta técnica indica que los miembros de una población coexisten rodeados de individuos similares que tienen propiedades similares.

En la Figura 4 se puede observar el proceso básico del funcionamiento de k -NN. El método de k -NN categoriza los datos basándose en la distancia entre los individuos.

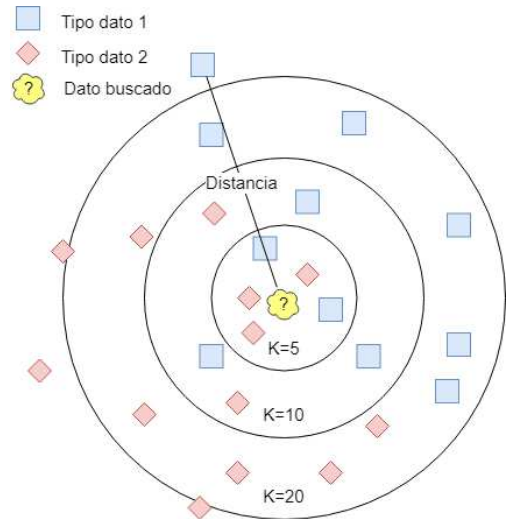


Figura 4. Diagrama del comportamiento de k -NN en tareas de clasificación.

Pronóstico con Regresión Lineal

La regresión lineal es un análisis que busca explicar la relación entre una variable Y (salidas) y una variable o variables $X_1 \dots X_P$ (entradas), con el objetivo de: realizar observaciones futuras, buscar el efecto, causa o relación entre la respuesta de las variables o una descripción general de los datos [8]. Para este trabajo se utilizó con el objetivo de realizar observaciones futuras para series de tiempo de la velocidad del viento.

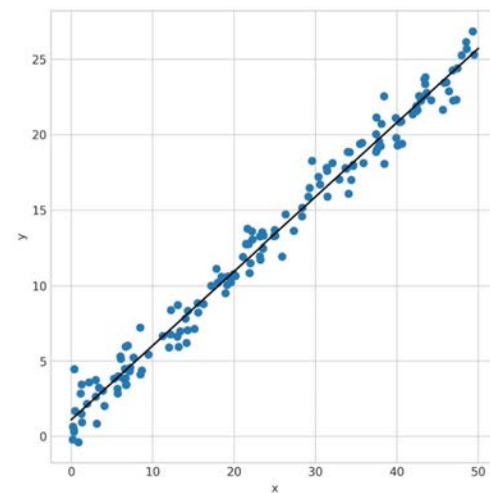


Figura 5. Ejemplo gráfico de regresión lineal.

En [9] utilizaron regresión lineal para realizar el pronóstico de 28 periodos de 24 horas de la velocidad del viento donde utilizaron datos históricos de los meses de febrero de los años 2006 al 2012. Los resultados observados muestran que la regresión lineal pueden obtener resultados con una buena aptitud.

Pronóstico con Bosques Aleatorios

Los Bosques Aleatorios son una combinación de árboles predictores; tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos [10]. En la Figura 6 se puede observar una representación simplificada de Bosques Aleatorios.

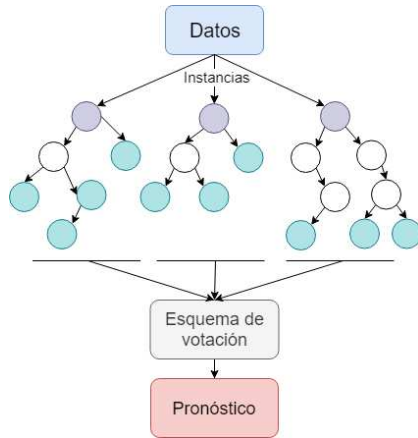


Figura 6. Diagrama básico de Bosques Aleatorios.

La predicción del árbol se obtiene tomando las clases más comunes o las predicciones del árbol de valor medio. Cada árbol de decisión se entrenó en un subconjunto aleatorio del conjunto de entrenamiento y solo se utilizó un subconjunto aleatorio de las características.

Fusión de Datos

La fusión de datos se realiza con el objetivo de mejorar el pronóstico realizado de manera individual. Para esto tres esquemas de fusión de datos fueron definidos: mediana aritmética, media aritmética y agrupamiento espacial.

En la Figura 7 se describe la estructura del proceso de fusión de datos, donde se obtendrá el pronóstico (y_{t+1}) con cada método de inteligencia artificial, se realiza la fusión de datos y se selecciona el mejor resultado.

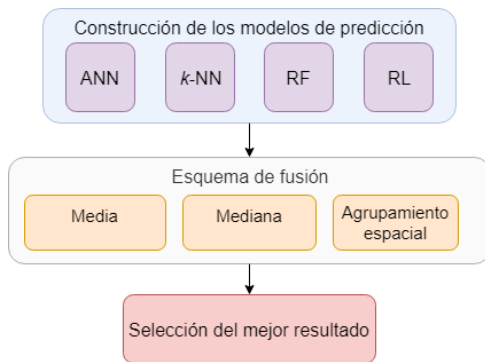


Figura 7. Implementación del esquema de fusión con los modelos de predicción utilizados en este trabajo.

Implementación de la Mediana Aritmética

El proceso del pronóstico mediante el uso de la mediana aritmética se realiza con los pronósticos obtenidos por cada técnica utilizada. Para obtenerlo, todas las predicciones obtenidas se almacenan en un vector ordenado. La predicción resultante es el dato que se encuentre en la posición que esta justo a la mitad del vector. Para determinar la posición que se encuentra justo a la mitad del vector en caso de que el tamaño de los datos sea impar (n), se utiliza la Ecuación 2:

$$m_e = x_{\frac{n+1}{2}} \tag{2}$$

Si n es par, la mediana es la media aritmética (promedio) de los dos valores centrales. Cuando n es par, los dos datos que están en el centro de la muestra ocupan las posiciones $\frac{n}{2}$ y $\frac{n}{2} + 1$. La Ecuación 3 muestra la fórmula del cálculo de la mediana cuando n es par.

$$m_e = \frac{(x_{\frac{n}{2}} + x_{\frac{n}{2}+1})}{2} \tag{3}$$

Implementación de la Media Aritmética

Para este tipo de fusión, el pronóstico resultante se obtiene del promedio de todos lo pronósticos obtenidos. La Ecuación 4 describe el proceso del promedio que se aplicó en la fusión de los datos pronosticados.

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} \tag{4}$$

En la Figura 9 se visualiza el proceso de la aplicación de la media con los diferentes pronosticadores.

Implementación Agrupamiento Espacial

La técnica de agrupamiento espacial se definió como base la mediana aritmética del resultado del pronóstico de series de tiempo. Durante el proceso de la mediana (V_m) del vector de los pronósticos resultantes, se promedia el valor de la mediana junto con algunos pronósticos que se encuentran en su vecindad.

El proceso de agrupamiento espacial agrupa los datos con otros de características similares, creando grupos de datos, como se observa en la Figura 8.

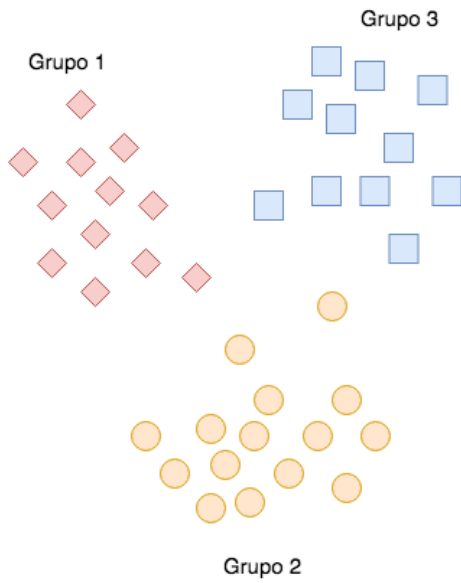


Figura 8. Ejemplo de Agrupamiento de Datos.

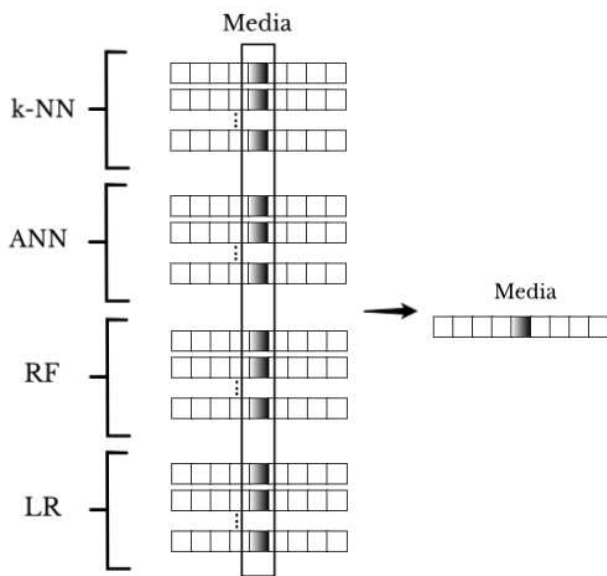


Figura 9. Proceso de aplicación de media con pronosticadores.

Resultados

La propuesta fue probada mediante experimentos con series de tiempo de la velocidad del viento obtenidas de diferentes plantas generadoras de energía eólica en el estado de Michoacán, México. Las series de tiempo tienen registros a intervalos horarios.

Por cuestión de espacio, los resultados individuales de cada pronosticador no se muestran. Solo los mejores resultados obtenidos por la fusión entre los diferentes métodos de inteligencia artificial se muestran en la Tabla 2, donde se muestra el mejor pronóstico, de

cada serie estudiada, el modelo resultante y su aptitud ($MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$).

Tabla 1. Pronóstico.

Serie de Tiempo	Modelo	MSE
Aristeo-Mercado	ANN + RF	0.004670
El Fresno	ANN + LR + RF	0.009451
La Palma	ANN + RF	0.009179
La Piedad	ANN + LR + RF	0.006253
Cointzio	ANN + kNN + RF	0.005278
Malpais	ANN + RF	0.005626
Pátzcuaro	ANN + RF	0.000458

Como se podrá observar en la Tabla 2, se obtuvieron mejores resultados en los métodos que utilizaron el esquema de fusión de datos, en comparación de los métodos individuales de inteligencia artificial. En los resultados de la Tabla 2 se puede observar que los mejores resultados involucraron el uso de ANN, seguido por el uso de RF. LR y k-NN no mostraron resultados mejores en la mayoría de los casos. Sin embargo cada uno de estos métodos demostró que en conjunto con otros pueden ser mejores.

En la Figura 10 se pueden observar los resultados de los pronósticos obtenidos en Pátzcuaro y Malpais (solo se presentaron las mejores 2 gráficas por cuestiones de espacio).

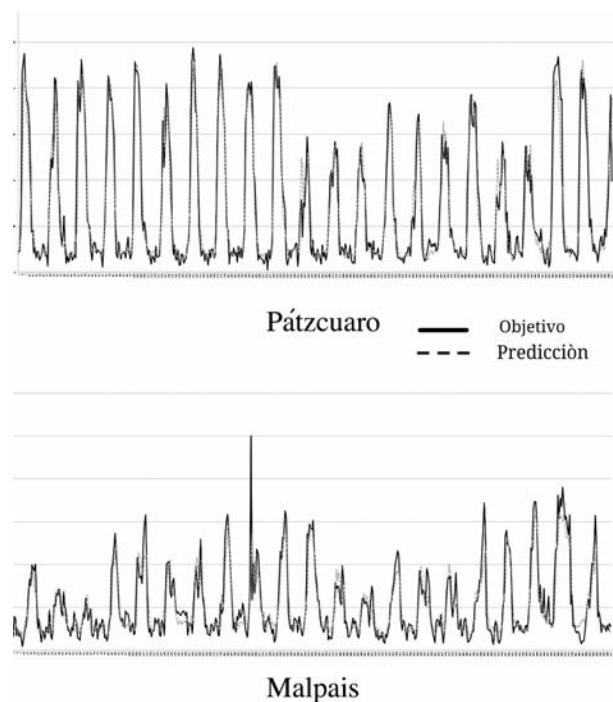


Figura 10. Proceso de cálculo de agrupamiento espacial para obtener el pronóstico de la serie de tiempo.

Conclusiones

La naturaleza caótica del viento nos lleva a la necesidad de buscar métodos para la predicción de su comportamiento para la producción de energía eólica.

Existen muchos métodos utilizados para el pronóstico de velocidades de viento, pero en muchas situaciones el uso de un solo método no es suficiente para obtener resultados deseables. Por esto se propone en este trabajo el uso de múltiples métodos, y realizar un esquema de fusión de datos para obtener un resultado con mejor aptitud.

En este trabajo se utilizan métodos de inteligencia artificial para realizar los pronósticos de velocidad del viento, y se realiza una fusión de datos para buscar resultados con una mejor aptitud en el resultado.

Se demuestra que se pueden obtener mejores resultados al utilizar diferentes métodos de pronóstico, y después utilizar un esquema de fusión de datos de los métodos individuales.*

REFERENCIAS

1. Sawin, J. L., Sverrisson, F., Rutovitz, J., Dwyer, S., Teske, S., Murdock, H. E. y Misra, A. (2018). Renewables 2018-Global status report. A comprehensive annual overview of the state of renewable energy, 41.
2. Rodriguez, H., Flores, J., Puig, V., Morales, L., Guerra, A. y Calderon, F. (2017). Wind speed time series reconstruction using a hybrid neural genetic approach. En *IOP Conference Series: Earth and Environmental Science*, 93, 012020.
3. Chang, W-Y. (2014). A literature review of wind forecasting methods, En *Journal of Power and Energy Engineering*, Scientific Research Publishing, 2, 161.
4. Wu, Y-K. y Hong, J-S. (2007). A literature review of wind forecasting technology in the world, En *Power Tech, 2007 IEEE Lausanne*, 2007, 504-509.
5. Lei, M., Shiyan, L., Chuanwen, J., Hongling, L. y Yan, Z. (2009). A review on the forecasting of wind speed and generated power. En *Renewable and Sustainable Energy Reviews*, 915-920.
6. Welch, R. L., Ruffing, S. M. y Venayagamoorthy, G. K. (2009). Comparison of feedforward and feedback neural network architectures for short term wind speed prediction. En *Neural Networks, IJCNN*. International Joint Conference on, 3335-3340.
7. Sorjamaa, A. y Lendasse, A. (2006). Time series prediction using dirRec strategy. En *European Symposium on Artificial Neural Networks Bruges (Belgium)*, 146-148.
8. Faraway, J. J. (2002). Practical regression and ANOVA using R., University of Bath Bath.
9. Kiranvishnu, K. y Sireesha, K. (2016). Comparative study of wind speed forecasting techniques. En *Power and Energy Systems: Towards Sustainable Energy (PESTSE)*, Biennial International Conference on IEEE, 2016. 1-6.
10. Köksoy, C. E., Özkan, M. B., Buhan, S., Demirci, T., Arslan, Y., Birtürk, A. y Karagöz, P. (2015). Improved Wind Power Forecasting Using Combination Methods. En *Machine Learning and Applications (ICMLA)*, IEEE 14th International Conference on. IEEE, 2015, 1142-1147.

SOBRE LOS AUTORES



Hector Rodríguez Rangel es egresado del Instituto Tecnológico de Morelia en la carrera de Ingeniería en Sistemas Computacionales (2009). La maestría y el doctorado los realizó en el departamento de Posgrado de la Facultad de Ingeniería Eléctrica en la Universidad Michoacana (2009, 2014). Actualmente es Profesor Titular en el Tecnológico Nacional de México campus Culiacán.



Juan Ramón Valenzuela Barraza es egresado del Instituto Tecnológico de Culiacán en la carrera de Ingeniería Mecatrónica (2016). Actualmente estudia Maestría en Ciencias Computacionales en el departamento de Posgrado e Investigación del Instituto Tecnológico de Culiacán.



Carlos Lara Álvarez es doctor en ciencias en Ingeniería Eléctrica por la Universidad Michoacana, es Ingeniero en Comunicaciones y Electrónica por la Universidad de Guanajuato. Actualmente labora en el Centro de Investigación en Matemáticas (CIMAT), unidad Zacatecas.



Luis Alberto Morales Rosales obtuvo el grado de doctor en Ciencias Computacionales en el Instituto Nacional de Astrofísica Óptica y Electrónica. Actualmente es catedrático Conacyt en la Universidad Michoacana de San Nicolás de Hidalgo.



Giovanni Manjarrez Montelongo es egresado del Instituto Tecnológico de Culiacán en la carrera de Licenciatura de Ingeniería en Sistemas Computacionales (2014). Realizó la maestría en Ciencias de la Computación en el departamento de Posgrado de Sistemas en el Instituto Tecnológico de Culiacán (2018).

ARTÍCULO ACEPTADO

Sistema Tutor Afectivo para la enseñanza de métodos numéricos de Euler y Runge-Kutta

Gustavo Padrón Rivera, Cristina Joaquin Salas, Pedro Omar Silva Vásquez y Ricardo Serrano Herrera

Introducción

El cómputo afectivo, es la parte de las ciencias de la computación que se encarga del estudio y el desarrollo de sistemas y dispositivos que pueden reconocer, interpretar, procesar y estimular las emociones humanas. Las investigaciones en esta área contribuyen a la inteligencia

artificial, reconocimiento de patrones, aprendizaje automático, interacción humano-computadora, robótica social, agentes autónomos, neurociencia afectiva, neuroeconomía, salud, y donde la tecnología se usa para detectar, reconocer, medir, comunicar, obtener, manejar o de otra manera entender e influenciar las emociones [7].

Análisis facial de las UAs por Affdex SDK, rastreando los músculos de la cara para el reconocimiento de estados afectivos.

Estados afectivos y su relación con el aprendizaje

En la actualidad, existe una amplia evidencia en la literatura que respalda la idea de las emociones y su influencia en el aprendizaje, principalmente, en matemáticas y ciencias exactas, al cometer errores y tratar de recuperarse de ellos [5]. Estudios previos han explorado a los estados afectivos que ocurren durante el aprendizaje complejo y sus efectos, identificando cinco: aburrimiento, concentración/flujo, confusión, frustración y un estado neutral [3][8]. La adaptación del reconocimiento de emociones en la tecnología, tiene la finalidad de buscar adaptarse al ritmo de trabajo, y a mejorar el estado de ánimo del usuario. Por ejemplo, cuando está presente el estado de concentración, el cansancio desaparece y el tiempo parece detenerse. En estos casos, la productividad aumenta considerablemente, por lo que es importante fomentarlo.

Sistema tutor Afectivo (STA)

Un Sistema Tutor Afectivo (STA), es un tipo de Sistema Tutor Inteligente (STI) que mejora las experiencias de aprendizaje del usuario, en matemáticas y ciencias, creando respuestas afectivas, que dependen de las emociones que presenten los usuarios, en actividades que requieren un gran esfuerzo, con el propósito de detectar el afecto del alumno y diseñar respuestas apropiadas con el fin de motivarlo [6]. Los STI son entornos amigables para el aprendizaje, también definidos como maestros virtuales que ofrecen atención personalizada al estudiante que interactúa con ellos, y son capaces de dar retroalimentación cuando es necesario [8], y se adaptan al ritmo de aprendizaje de los alumnos. Por lo tanto, los STA son STI que se adaptan a los estados afectivos de los sujetos [1].

Identificación automática de estados afectivos)

Se desarrolló un STA, denominado Tamaxtil, que significa «maestro» en lengua náhuatl, el cual tiene la capacidad de detectar la frustración y la confusión de los estudiantes. Para esto, hace uso del SDK de Affective/Affdex (ver Figura 1), herramienta de distribución libre, bajo licenciamiento GNU, diseñada para el análisis facial a través de la identificación de Unidades de Acción Facial (UAF), las cuales son etiquetas independientes de los 44 músculos del rostro humano. Las combinaciones de las UAFs sirven para identificar emociones [10]. Este SDK permite una fácil integración, y su capacidad de identificación de emociones ha sido probada desde diferentes ángulos, grados de iluminación y tipos de personas, etnia y el uso de lentes en el sujeto; además, de tener la capacidad de identificar el sexo. Cabe mencionar que esta tecnología es de acceso libre, para su distribución con fines de investigación, por lo que fue posible el desarrollo y adaptación de esta herramienta con gran facilidad. El reconocimiento se realiza a través de la identificación de las combinaciones de Unidades de Acción Facial, que como ya se mencionó anteriormente, son etiquetas de cada uno de los músculos del rostro y sus combinaciones sirven para identificar emociones. Después, a través de las características del histograma de orientación de gradiente (HOG), se obtiene la región de interés de la imagen definida por los puntos de referencia del rostro y con una máquina de soporte de vectores (SVM), el cual es un tipo de algoritmo de aprendizaje automático para clasificación, el cual fue entrenado con 10,000 imágenes faciales codificadas, se proporcionan puntajes de 0 (ausente) a 100 (presente) para cada UAF, para saber su presencia, y a partir de combinaciones, saber cual emoción está presente [9].

Es posible detectar la frustración por las combinaciones de UAF 1+12 y 2+12, y la confusión por 4+25 y 4+26.

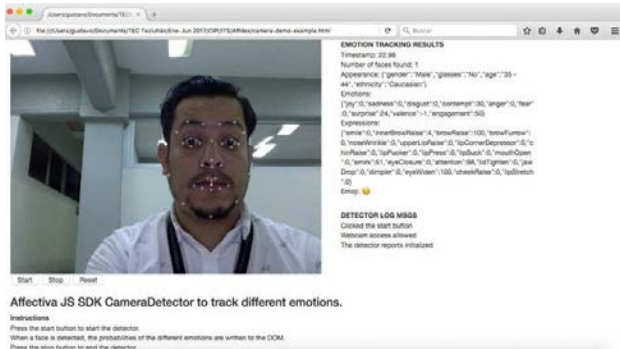


Figura 1. Análisis facial de las UAs por Affdex SDK, rastreando los músculos de la cara para el reconocimiento de estados afectivos.

Se ocultó el reflejo de lo que captura la cámara Web, para evitar que el estudiante se distrajera, y lograr enfocar su atención en la resolución de los ejercicios (ver Figura 2).



Figura 2. El STA oculta la cámara y solicita permiso al usuario de activar la cámara web para el análisis facial.

Cuando un sujeto presenta un estado de frustración o confusión, existe una gran probabilidad de caer en un estado de aburrimiento, que causaría el abandono de la meta o la adquisición de nuevo conocimiento [4]. Es posible detectar la frustración por las combinaciones de UAF 1+12 y 2+12, y la confusión por UAF 4+25 y 4+26 [11].

Metodología

El STA (ver Figura 3), está diseñado para la enseñanza de los métodos de aproximación de Euler y Runge-Kutta. Se llevaron a cabo diversos experimentos piloto, para verificar la efectividad del STA. Cada uno de los experimen-

tos, se llevó a cabo en cuatro días. Durante el primer día, se presentaron los métodos de Euler y Runge-Kutta en diapositivas por un profesor; esto es necesario para ayudar a la comprensión del tema, como sería en clase, y los estudiantes resuelven una examen con ejercicios denominado «pre-prueba». En el segundo y tercer día, se lleva a cabo la interacción con el STA (ver figura 4); mientras se encuentran resolviendo ejercicios propuestos por la aplicación virtual son analizados facialmente para detectar frustración o confusión. Cuando es detectado alguno de estos, se muestra un mensaje que les sugiere revisar la ayuda.



Figura 3. Interfaz de identificación y escenario de trabajo, del STA, versión 1.0.



Figura 4. Estudiantes interactuando con Tamaxtil, la primer versión (arriba), y la última versión 6.2.1 (abajo).

El material basado en video genera emociones positivas en los estudiantes, que ayudan al proceso de adquisición de nuevo conocimiento.

El reconocimiento de los estados afectivos, tiene como objetivo regularlos para mejorar el compromiso de aprendizaje de los estudiantes. Para llevar a cabo esta tarea, cada cinco segundos se realiza el análisis facial con SDK, las combinaciones de UAFs detectadas y se registran en una base de datos. El tiempo promedio de interacción, es de 40 minutos. Finalmente, el día cuatro, los estudiantes resuelven otra prueba denominada «pos-prueba», con ejercicios diferentes a los usados en la «pre-prueba», pero solubles con los mismos métodos. Dichas pruebas, permiten medir el progreso de los estudiantes.

Resultados

Se identificó que el estado afectivo de frustración, está presente en casi todos los estudiantes que obtuvieron un bajo promedio de calificación en las pruebas al finalizar los días de experimentación coincidiendo con investigaciones similares a esta, pero en las que el análisis facial era llevado a cabo por expertos humanos en detección de emociones. Los participantes del experimento piloto que terminaron los cuatro días de experimentación, fueron 18 de 30 estudiantes del Instituto Tecnológico Superior de Teziutlán, Puebla. La edad promedio fue de 20 años, en ambos sexos y el 76 % obtuvo un desempeño bajo en ambas pruebas. A través de la técnica de Análisis de Componentes Principales (ver Figura 5), fue posible identificar las características de esta población [12], en la que se encontraron patrones de conducta relacionadas con su desempeño, haciendo evidente una correlación entre la frustración y un pobre desempeño.

Posteriormente, se aplicó un análisis por agrupamiento (ver Figura 6), con la intención de identificar similitudes en los usuarios. El tercer grupo de derecha a izquierda, que es el de mayor tamaño, identificó a estudiantes que presentaron un alto índice de lapsos de confusión y frustración, pero también de clicks en el menú de ayuda. En experimentos similares, con herramientas de aprendizaje, se demostró que el material multimedia basado en video suele generar emociones más positivas en estudiantes [2]. Por lo que se buscó agregar la capacidad de identificación de dichas emociones, cuando tienen un alto grado de incidencia en un lapso de tiempo corto, y automáticamente hacer la reproducción de videos con ayuda del tema, con el fin de incentivar emociones que ayuden a la comprensión del tema por parte del sujeto.

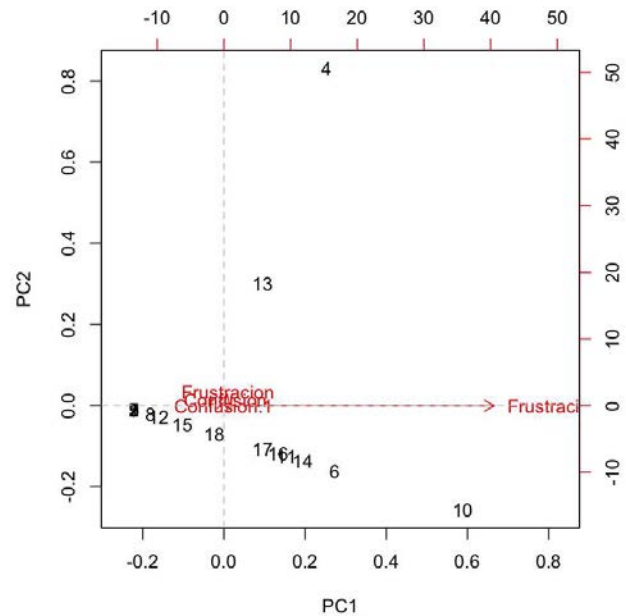


Figura 5. Análisis de Componentes Principales para la población que utilizó la primer versión del STA, en la que se puede observar una tendencia de los estudiantes hacia el componente principal no. 1, identificado como “PC1”, el cual es la frustración.

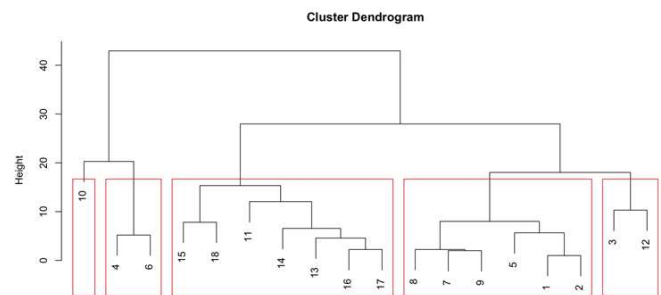


Figura 6. A través de la técnica por agrupamiento se analizó el tercero, de izquierda a derecha, que mostró a estudiantes con lapsos grandes de frustración y confusión, y al mismo tiempo de clicks en búsqueda de ayuda.

Hasta este momento, se utilizó una versión del STA que al parecer no era intuitiva en el uso y diseño. Además, debido a la complejidad de los ejercicios fue necesario mejorar la ayuda para personalizar el tipo de material que se le presenta al alumno, esperando que pueda salir de dichos ciclos de frustración y poder comprender el tema completamente. Por lo que se agregó este tipo

de elementos a la interfaz de Tamaxtil, en los que incluyeron video tutoriales de los temas (ver Figura 8), y un asistente virtual, con forma de un mapache, con voz y movimiento dando retroalimentación al estudiante al ejecutar correctamente, equivocarse o tratar de adivinar las respuestas, que aparece en cada una de estas acciones (ver Figura 9).

Cabe mencionar, que los estudiantes presentaron cierto rechazo al análisis facial, al sentir intromisión en su privacidad, debido a que la luz led de las cámaras Web de sus computadoras se enciende. Para esto, se realizó un video explicativo que puede ser visto desde el inicio, cuando el usuario se está registrando, explicando la importancia de las emociones en el aprendizaje, y cuál es el análisis que se lleva a cabo. Se agregó una casilla de confirmación para permitir o no la identificación de estados afectivos (ver Figura 7).



Figura 7. Video explicativo en la interfaz de identificación del STA Tamaxtil_v6.2.1.

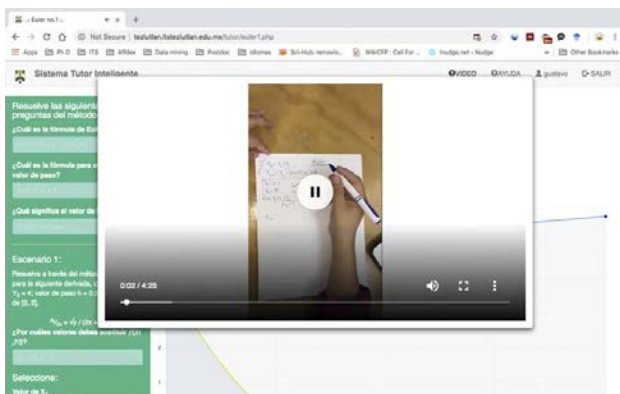


Figura 8. Ejecución de video tutoriales desde la interfaz del STA Tamaxtil_v6.2.1.

En base a los resultados anteriores, se realizaron mejoras en la interfaz, agregando mensajes sobre todo tipo de errores que el usuario pudiera cometer, se llevaron a cabo nuevos experimentos, usando la misma metodología, para una nueva población de 40 sujetos, en los que

34 concluyeron los 4 días de experimentación, y el resto se descartó. Todos ellos, del Instituto Tecnológico Superior de Teziutlán, Puebla; con un promedio de edad de 20 años, ambos sexos. Como se puede ver en la Figura 10, al aplicar nuevamente un análisis de Componentes Principales, la búsqueda de ayuda y revisión de los videotutoriales incorporados en la interfaz del STA fue la característica de estos estudiantes.

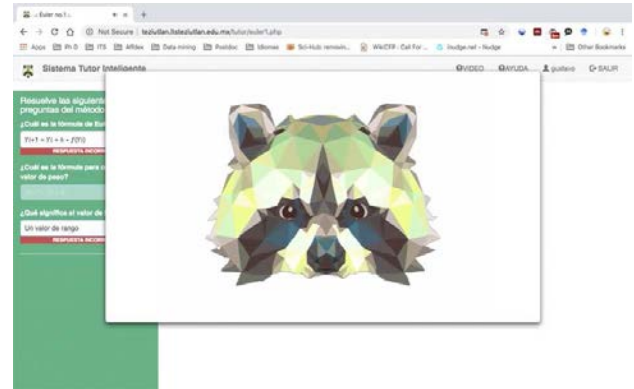


Figura 9. El asistente virtual, alertando al sujeto que está tratando de adivinar las respuestas y que su estado afectivo detectado pudiera influir de manera negativa en el aprendizaje, por lo que sugiere la búsqueda ayuda en la interfaz o al maestro.

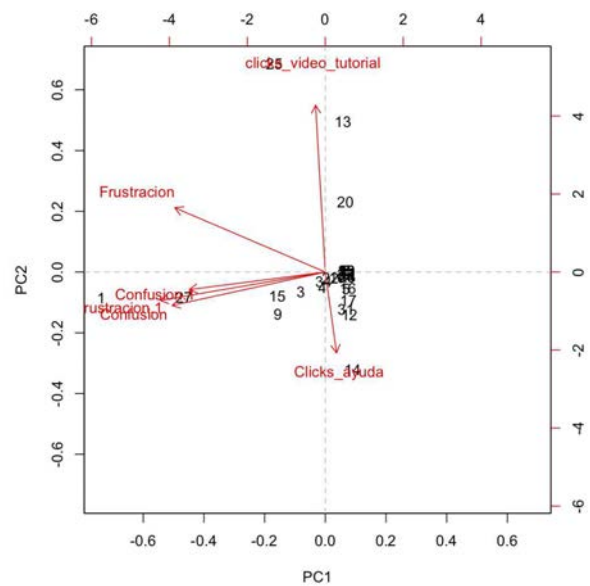


Figura 10. Análisis de Componentes Principales para la población que utilizó la versión mejorada del STA, en la que se puede observar una tendencia de los estudiantes hacia los componentes principales 1 y 2, identificados “PC1” y “PC2” respectivamente, los cuales son los clicks en las ventanas de ayuda y los video tutoriales.

Se infiere que a pesar de sentirse frustrados o confundidos como en grupos anteriores, debido a la mejora en la interfaz y la ayuda. Además, hubo una mejoría en los lapsos de emociones positivas, ya que al calificar el avance entre las pre y pos-prueba, el 70 % de los estudiantes obtuvieron una calificación aprobatoria, de una a otra.

Conclusiones

La elección de los métodos de Euler y Runge-Kutta es adecuada, pues su presentación en los cursos de las Facultades de Ingeniería está asociada a materias con un muy alto nivel de reprobación. Se espera extenderse a otros temas cuya enseñanza también presenta dificultades, no solamente de ecuaciones diferenciales y de métodos numéricos, sino también de temas más básicos de cálculo y álgebra. El STA, a través de la adecuación de tareas y actividades según la emoción que está percibiendo del aprendiz, pueda ayudar en la adquisición de nuevo conocimiento, partiendo de identificar aquellas emociones que afectan su desempeño, y guiarlo en todo momento, como lo haría un maestro humano en el aula de clases. Las mejoras a la interfaz y la incorporación de ayuda multimedia en forma de video ayudaron considerablemente a la adquisición de nuevo conocimiento de los primeros a los últimos experimentos piloto.*

REFERENCIAS

1. Ben, M., Neji, M., Alimi, A. M. y Gouardères, G. (2010). The Affective Tutoring System. *Expert Systems with Applications*, 37(4), 3013-3023.
2. Chen, A. (2017). Assessing the effects of different multimedia materials on emotions and learning performance for visual and verbal style learners. 38-40.
3. Craig, S., Graesser, A., Sullins, J. y Gholson, B. (2004). Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media*, 29(3), 241-250.
4. D'Mello, S. D., Taylor, R. S. y Graesser, A. (2007). Monitoring Affective Trajectories during Complex Learning. En *Proceedings of the 29th Annual Cognitive Science Society*, 203-208.
5. D'Mello, S. K., Lehman, B. y Person, N. (2010). Monitoring affect states during effortful problem solving activities. *International Journal of Artificial Intelligence in Education*, 20(4), 361-389.
6. Defalco, J., Rowe, J. P. y Mott, B. W. (2017). Detecting and Addressing Frustration in a Serious Game for Military Training.
7. Lin, H. C. K., Su, S. H., Chao, C. J., Hsieh, C. Y. y Tsai, S. C. (2016). Construction of multi-mode affective learning system: Taking Affective Design as an Example. *Educational Technology and Society*, 19(2), 132-147.
8. Malekzadeh, M., Mustafa, M. B. y Lahsasna, A. (2015a). A Review of Emotion Regulation in Intelligent Tutoring Systems. 18, 435-445.
9. McDuff, D., Mahmoud, A., Mavadati, M., Amr, M., Turcot, J. y Kaliouby, R. el. (2016). AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit. En *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*, 3723-3726.
10. McDuff, D., Member, S., Kaliouby, R. El y Picard, R. W. (2012). Crowdsourcing Facial Responses to Online Videos. 3(4), 456-468.
11. Padrón-Rivera, G., Rebolledo-Mendez, G., Parra, P. P. y Huerta-Pacheco, N. S. (2016). Identification of action units related to affective states in a tutoring system for mathematics. *Educational Technology and Society*, 19(2), 77-86.
12. Tharwat, A. (2016). Principal component analysis - a tutorial. *International Journal of Applied Pattern Recognition*, 3(3), 197.

SOBRE LOS AUTORES



Gustavo Padrón Rivera obtuvo el grado de doctor en Ciencias de la Computación, Maestría en Inteligencia Artificial en la Universidad Veracruzana. Actualmente es profesor investigador en el Instituto Tecnológico Superior de Teziutlán. Sus intereses científicos incluyen Cómputo Afectivo, Minería de Datos e Inteligencia Artificial.



Cristina Joaquín Salas obtuvo el grado de Maestría en Educación Matemática en la Universidad Popular Autónoma del Estado de Puebla. Actualmente es profesora investigadora en el Instituto Tecnológico Superior de Teziutlán. Sus intereses científicos incluyen Minería de Datos y Educación Matemática.



Pedro Omar Silva Vásquez obtuvo el grado de Maestría en Sistemas Interactivos Centrados en el Usuario en la Universidad Veracruzana. Actualmente es profesor de tiempo completo en el Instituto Tecnológico Superior de Teziutlán. Sus intereses científicos incluyen la interacción Humano-Computadora, Diseño de Interacción y Desarrollo de aplicaciones Web y Móvil.



Ricardo Alberto Serrano Herrera obtuvo el grado de Ingeniería en Sistemas Computacionales en el Instituto Tecnológico Superior de Teziutlán. Actualmente es profesor en la misma institución. Sus intereses incluyen el desarrollo de aplicaciones móviles y realidad aumentada.

ARTÍCULO ACEPTADO

Inclusión de la Inteligencia Artificial en Disputas Legales sobre Propiedad Intelectual

Rodrigo Vidal-López y José Antonio Hernández-Servín

La Importancia de las Disputas Legales sobre Propiedad Intelectual

Las disputas legales sobre la propiedad intelectual son un recurso utilizado por los inventores como medio de protección de sus creaciones [1]. Estas disputas legales se complican cuando la infracción en propiedad intelectual es cometida en territorios con procesos legales distintos, generalmente, internacionales. La importancia de estas disputas se remonta a que tan sólo en 2015 la solitu-

des de patentes creció un 9.2 % en los Estados Unidos, el cual es el crecimiento más rápido en los últimos 18 años [2]. Para 2017, la Organización Mundial de la Propiedad Intelectual, reportó un récord en solicitudes y registro de patentes, pero esta vez en China [3]. Existen dos formas para llevar a cabo una disputa legal, una de ellas es la negociación y la otra el litigio. La primera de ellas, es un común acuerdo que se da entre las partes involucradas en mutuo beneficio y el litigio por otro lado es el conflicto entre las partes involucradas en un juicio.

Las disputas legales son un recurso utilizado por los inventores como medio de protección de sus creaciones.

De acuerdo a Fainshmidt *et al.*[1], existen dos conceptos importantes que influyen en la toma de decisiones para la resolución de los conflictos, los cuales son: **la distancia cognitiva** y **la distancia legal**. La **distancia cognitiva** se refiere a las malas interpretaciones que pueden surgir de entre las partes involucradas, es decir, la educación y los contextos sociales de las partes involucradas son diferentes, por lo cual refuerza la idea de una resolución de conflictos mediante un litigio y no una negociación. La **distancia legal**, se asocia de manera positiva a llevar una negociación como resolución de conflictos en lugar de un litigio. Como se puede observar, estas dos alternativas generan a su vez otro conflicto entre sí, de acuerdo a las definiciones y referencias actuales ofrecen poca información en la elección de la solución de conflictos.

Los litigios como resolución de conflictos

Los litigios son la parte medular de los conflictos ya que existen temas de rivalidad y contrademandas, lo que en ocasiones llevan demasiado tiempo en resolverse, sin embargo, se impone a la idea de negociar un acuerdo fuera de los tribunales de justicia, como sugiere la distancia legal, ya que este acuerdo puede ser considerado como un juego de suma-cero en el que "el ganador se lleva todo"[2] y por supuesto que ninguna de las partes involucradas quiere perder.

Por tal razón, en el estudio realizado por Fainshmidt *et al.* [1] se examinaron las controversias internacionales en materia de propiedad intelectual, resueltas en tribunales de los Estados Unidos. Los datos analizados se

obtuvieron de la base de datos de 94 tribunales federales, teniendo como resultados una correlación de la distancia legal y cognitiva, donde a mayores distancias legales y cognitivas darán lugar a litigios, en vez de negociaciones, porque la distancia cognitiva refuerza el efecto negativo de la distancia legal sobre elegir la negociación como una estrategia de resolución de conflictos. Al mismo tiempo, la distancia legal se asocia positivamente con la probabilidad de obtener una negociación como resolución, porque las partes involucradas no quieren tener un conflicto legal, al no saber las leyes con las que se rige la parte anfitriona, lo que podría considerarse una ventaja para esta, es decir, el lugar y con las leyes donde se lleva a cabo la disputa legal.

Inteligencia Artificial (IA) en Disputas Legales

La IA se ha interesado en los años recientes en el problema que se presenta, al tratar de obtener una resolución de conflictos en las disputas legales, específicamente en las que están relacionadas con la propiedad intelectual.

Existen dos ramas de la IA con las que se ha tratado el problema de la resolución de conflictos, como son: la **Teoría de Juegos** y la **Minería de Datos**. La Teoría de Juegos se ha utilizado en litigios para negociar la solución a la controversia que logre resultados favorables para los oligopolistas (un grupo pequeño de involucrados). La solución a la controversia con teoría de juegos se hace mediante un análisis del comportamiento y toma de decisiones, para encontrar una solución óptima o formar una estrategia basada en el conocimiento de una situa-

ción, como las alternativas y los riesgos. La teoría de juegos es útil para observar la evolución del comportamiento cooperativo, ya que la interacción entre los involucrados afecta a las decisiones que toman [2].

La teoría de juegos tiene elementos básicos como son:

- Jugadores: En este caso en particular, los jugadores pueden ser irracionales, por ejemplo, pueden hacer creer a otros jugadores que las amenazas que hagan son reales.
- Movimientos/acciones: Se tiene un número limitado de movimientos, estos movimientos pueden ser secuenciales o paralelos:
 - Secuenciales: Estos movimientos son realizados con árboles de decisión, árboles de inducción inversa, además de que las estrategias son irreversibles.
 - Paralelos: Estos movimientos se hacen mediante árboles de decisión, matriz de resultados y el equilibrio de Nash.
- Pagos: Son los números que representan la motivación de los jugadores.

En este caso, los movimientos en paralelo son los más importantes con respecto a los secuenciales, ya que en las disputas legales los abogados pueden realizar movimientos aislados de la contraparte involucrada.

Por lo tanto, para una resolución de conflictos con teoría de juegos, es necesario involucrar los movimientos en paralelo, los cuales involucran árboles de decisión,

matriz de resultados y el equilibrio de Nash. En la matriz de resultados es necesario tener mapeados todos los posibles resultados de todos los posibles movimientos. Y en el equilibrio de Nash cada jugador hace su mejor movimiento dado el movimiento del oponente; sin embargo esto no necesariamente conduce a los mejores resultados para uno o ambos jugadores, este equilibrio se puede encontrar mediante el Teorema de minimax [2].

El Teorema de minimax se utiliza para juegos de movimientos simultáneo y aplica las estrategias maximin y minimax. La estrategia maximin es donde el pago mínimo es el más alto. Por el contrario, la minimax limita el mejor resultado esperado del oponente y así se minimiza la ventaja máxima del oponente [2].

Por otra parte, en el año 2018, un grupo de la Universidad de Stanford llamado *Lex Machina* [4] desarrolló un software llamado *Legal Analytics* para servir de apoyo en los casos de litigio, realizando minería de datos sobre millones de páginas sobre litigios, después realizan un preprocesamiento para limpiar, etiquetar y estructurar los datos, algo que no se hace en [1], lo que permite sentar bases acerca de jueces, abogados y sujetos de los casos. La información que ofrece *Legal Analytics* es estratégica y crítica para crear una estrategia legal exitosa y obtener una ventaja ganadora [4].

Una rama de la IA, conocida como Sistemas Multi-agentes, podría aportar a la resolución de conflictos en las disputas legales. En la siguiente sección se explicará la propuesta con la que se puede llegar a una resolución de conflictos en otra perspectiva en contraste con la Minería de Datos y Teoría de Juegos.

***Legal Analytics* es un software que sirve de apoyo en los casos de litigio, utilizando Minería de Datos.**

Sistemas multi-agentes, una rama de la IA

Los Sistemas Multi-agentes se componen de múltiples elementos informáticos interactivos, es decir, agentes inteligentes¹ y se utilizan cuando se tiene un objetivo que un sólo agente no es capaz de satisfacer la tarea, sin embargo, por la naturaleza de los agentes el sistema multi-agente puede ser diseñado para que los agentes trabajen bajo un ambiente de colaboración o de competencia, en los sistemas multi-agentes uno de los factores más importantes es la comunicación entre ellos [6], por esta razón se requieren elementos como las ontologías², la taxonomía³

y finalmente un lenguaje de contenido que permita poder hacer una representación de la taxonomía. Para ello, existe un Lenguaje de Comunicación del Agente (ACL) [7] el cual permite que los agentes se puedan comunicar entre sí, mediante de un conjunto de primitivas, de esta manera sólo es necesario generar el mensaje de un agente a otro, pero no del lenguaje de contenido.

Los agentes inteligentes (*Ver Figura 1*) como se mencionó anteriormente, actúan conforme a sus objetivos, sin embargo, los ambientes en los que están inmersos, cambian, es decir, que los agentes se encuentren en un ambiente dinámico, donde pueden cambiar los recursos

¹agentes inteligentes: en el contexto informático son programas de computadora situados en un ambiente, capaces de actuar de manera autónoma para cumplir con los objetivos para los cuales fueron diseñados [5].

²ontología: en inteligencia artificial, es un sistema de datos para establecer las relaciones existentes entre los conceptos de un área de conocimiento.

³Taxonomía: se define como los términos básicos y relaciones de un dominio dado.

y estados, por tal motivo puede que el agente se quede sin posibilidades de cumplir con los objetivos planteados inicialmente y exhibir comportamientos indeseables o inesperados [8]. Aunado a lo anterior, los agentes pueden estar dotados de restricciones u obligaciones que deben tomar en cuenta para cumplir con sus objetivos. A estas restricciones u obligaciones se les conoce como: normas.

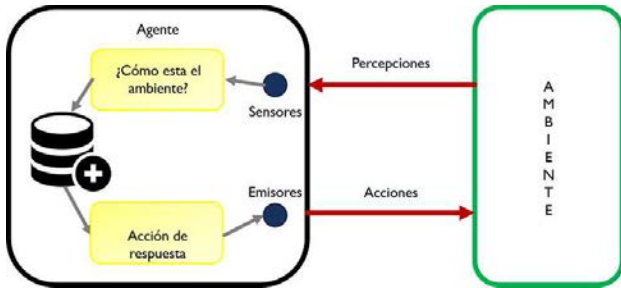


Figura 1. Arquitectura de un Agente Inteligente.

Las normas para los agentes son las que definen el comportamiento ideal que deben de tener en los ambientes que están inmersos, permitiéndoles saber así sus obligaciones y restricciones. Existen dos clases de normas: duras y suaves; las normas duras son estrictamente obligaciones, por lo que limitan la autonomía del agente, por el contrario las normas suaves no lo hacen y éstas se utilizan para instruir, coordinar u orientar las acciones que realizan [9].

El razonamiento práctico normativo, no es otra cosa que un estado mental para orientar a un agente a tomar decisiones sobre lo que es mejor para alcanzar sus objetivos en ciertas situaciones [10]. El problema del razonamiento práctico normativo se modela con la siguiente quintupla:

$$P = (FL, \Delta, A, G, N) \quad (1)$$

Donde:

- P representa el Problema del Razonamiento Práctico Normativo.
- FL es el conjunto de fuentes (posibles acciones a realizar por el agente).
- Δ indica el estado inicial donde se encuentra el agente.
- A conforma el conjunto de acciones que realizará el agente.
- G son las metas u objetivos que el agente debe cumplir.

- N es el conjunto de normas que existen en el entorno y el agente debe acatar.

Para poder abordar el problema planteado en la Ecuación 1 es necesario elaborar un plan⁴. Un plan no debe infringir normas duras; sin embargo puede infringir algunas normas suaves para alcanzar el mayor número de objetivos posibles [11]. Sin embargo, ya que las normas son establecidas para respetarse, en un plan la infracción de una norma trae como consecuencia una penalización al agente, mientras que su cumplimiento una recompensa [12].

El problema del razonamiento práctico normativo fue abordado por Shams *et al.* [10] con un enfoque basado en acciones. Una acción es un cambio de estado, el cual debe tener una duración de tiempo para llevarlo a cabo y sólo es posible que el agente realice el cambio si cumple con las condiciones previas del estado en el que se encuentra en ese momento. La propuesta de Shams *et al.* logra generar planes mediante un mecanismo que maximiza la utilidad de los mismos, dicha utilidad se define restando los costos causados por las normas infringidas a la ganancia de los objetivos cumplidos, por lo tanto el conjunto de planes óptimos son aquellos que maximizan dicha utilidad.

Answer Set Programming (ASP)

En el campo de la programación lógica, se puede representar el conocimiento y razonamiento a través de un modelo que permita determinar las soluciones del problema, esto se logra mediante el paradigma declarativo conocido como (Answer Set Programming). Este paradigma permite modelar especificaciones incompletas y restricciones, también la satisfacción de dichas restricciones y de las propuestas [13]. Una de las principales ventajas de ASP es que programar el conjunto de reglas es relativamente sencillo comparado con los problemas que resuelve, ofreciendo de manera sencilla programar el conjunto de respuestas que corresponden a las soluciones, para después con un solucionador de conjuntos de respuestas encontrar el conjunto de respuestas del programa. Ahora, la sencillez que ofrece ASP no quiere decir que los problemas que se traten sean simples, la forma de programar los conjuntos de respuestas es la parte simple; sin embargo modelar el conjunto de respuestas adecuados tiene un grado de complejidad mayor, ya que por supuesto si las representaciones del conocimiento y razonamiento no son generadas de manera adecuada, ASP dará como resultado soluciones equivocadas. ASP es utilizado para solucionar problemas difíciles por ejemplo configuración de productos, diagnósticos, problemas de teoría de gráficas y generación de planes. Este último problema, es del interés de esta investigación, ya que la propuesta de solución tiene como base la generación de planes para

⁴Plan: secuencia de acciones satisfactoria para el cumplimiento de uno o más objetivos

que los agentes puedan establecer una secuencia de acciones que los lleve a cumplir el mayor número de objetivos posibles.

Anger [13], menciona que para poder escribir programas en ASP se sigue una estrategia en dos partes: la primera, es escribir un grupo de reglas cuyos conjuntos de respuestas corresponderían a las soluciones y la segunda, es agregar un nuevo conjunto de reglas, pero ahora siendo estas las restricciones de integridad, eliminando así a aquellos conjuntos que representen soluciones inválidas.

Por otra parte, es importante mencionar, que a pesar de que ASP ha tenido un rápido crecimiento y aceptación en la relación existente entre la lógica y la programación lógica, no es óptimo, por lo que se han realizado estudios para encontrar programas equivalentes, los cuales contienen una parte común, lo que quiere decir es que esta parte en común se calcula en repetidas ocasiones para los conjuntos de respuestas. Osorio [14] hace una propuesta de optimización del cálculo de los conjuntos de respuestas. Esto es importante para un agente, debido a que una gran parte de su base de conocimiento inicial no va a cambiar a lo largo del tiempo, sino que se agrega nuevo conocimiento que puede modificar el curso de su plan; sin embargo, para ASP es necesario volver a computar la base de conocimiento que no cambió más la parte que si lo hizo, es en este punto donde se podría evitar esa parte de cómputo y obtener los conjuntos de respuestas en un tiempo de cómputo polinomial⁵.

Sistemas multi-agentes aplicando el Problema del Razonamiento Práctico Normativo y ASP

Los métodos anteriores permiten desarrollar una investigación para realizar una propuesta de resolución de conflictos en temas de disputas legales de propiedad intelectual. Primeramente, desde el punto de vista de los agentes inteligentes, donde éstos pueden fungir como las partes involucradas en el conflicto (verbigracia, abogados y jueces), en integración con el problema del razonamiento práctico normativo de tal manera que se establezca un diálogo entre los agentes. Para esto, es necesario establecer un modelo de sistemas multi-agentes, con una arquitectura BDI (*Beliefs, Desires, Intentions*) con el problema del razonamiento práctico normativo, ya que la propuesta de Shams está implementada para un sólo agente.

En segundo lugar, establecer el modelo implicaría que cada uno de los agentes que conforme el sistema tendría su propio problema de razonamiento práctico normativo, es decir, una quintupla que involucre conjuntos de flujos, acciones, metas, normas y estados iniciales, que se presentaría en una disputa legal, entre 2 abogados y el

juez encardado del caso (Ver Figura 2), donde P_A, P_B y P_J representan el Problema del Razonamiento Práctico Normativo para cada agente, por lo cual es necesario que estos agentes generen planes para que cómo se plantea en [10] puedan cumplir con al menos uno de los objetivos de cada uno de ellos y así el plan sea válido.

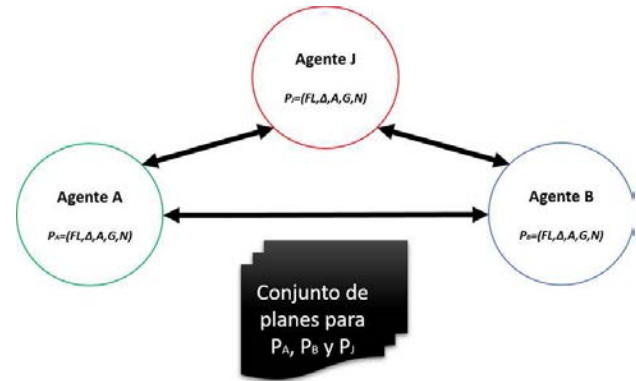


Figura 2. Modelado de una disputa legal con agentes inteligentes, aplicando el Problema del Razonamiento Práctico Normativo.

Aunado a lo anterior, una vez teniendo el modelo de sistema multiagente en integración que el problema del razonamiento práctico normativo, incluir este modelo en ASP para poder obtener el conjunto de respuestas óptimo para la resolución de conflictos en las disputas legales de propiedad intelectual. Obtener este conjunto de respuestas es relevante ya que en múltiples casos de disputas legales se han gastado una gran cantidad de recursos [1][2], tiempo y dinero principalmente, por lo que tener un modelo que nos permita integrar a las partes involucradas, las normas que se deben seguir, los intereses de cada parte, los objetivos que tienen y su postura de inicio, ahorraría una gran parte de estos recursos y saber cuales son las posibles alternativas que pueden seguir las partes para obtener mutuos beneficios.

Por supuesto, el realizar este modelado tiene un grado de complicación alto, esto principalmente por los temas de leyes que en este caso serían las normas del sistema, por lo que es fundamental tener una correcta interpretación de estas leyes, con un error en la interpretación el modelo no se colpasa, pero el conjunto de planes óptimos estarían en una falsedad. Al igual que la comunicación entre los agentes tiene que ser la adecuada, es decir, que el resto de los agentes que conforman el sistema tengan conocimiento de las acciones que realizan los demás agentes, así como los cambios normativos que se presenten en el ambiente en el que están inmersos.

⁵Tiempo polinomial: Tiempo de ejecución de un algoritmo para obtener la solución, en sujeto a las variables del algoritmo.

La Organización Mundial de la Propiedad Intelectual eligió la IA como primer tema de su nueva serie de investigaciones sobre tendencias tecnológicas de la OMPI.

Conclusiones

En conclusión, existen diversas formas de abordar el caso de estudio planteado (Minería de Datos y Teoría de Juegos, principalmente) como metodologías que se ajustan a la resolución de conflictos. Ahora con este planteamiento de utilizar sistemas multi-agentes, se pretende enriquecer esta resolución de conflictos para poder simular la interacción que resultaría entre las partes involucradas y dar a conocer la mejor decisión para ellas. No obstante, no se propone descalificar a los trabajos que se han realizado con Minería de Datos o Teoría de Juegos, al contrario, se propone en este caso de estudio retomar de estas dos áreas algunos conceptos o métodos utilizados para el enriquecimiento de la propuesta presentada.

La integración de los tres enfoques (Minería de Datos, Teoría de Juegos y Sistemas Multi-agentes) puede resultar muy productiva tomando en cuenta lo que han logrado, como por ejemplo la Minería de Datos, poder recopilar grandes cantidades de información, procesarla e interpretarla para poder dar a conocer estadísticas, las cuales requieren de realizar una interpretación para la resolución de conflictos y así ayudar a la toma de decisiones. Un punto desfavorable de la Minería de Datos, es precisamente la interpretación de los resultados, una mala interpretación, llevaría a una toma de decisiones equívoca y por lo tanto una resolución de conflictos que puede no ser la mejor, el desarrollo realizado por Lex Machina [?] ha tratado de solventar este punto desfavorable ofreciendo una aplicación como guía, hasta donde se tiene conocimiento, esta aplicación no toma en cuenta los movimientos o respuestas de la contraparte, por lo que no existe la posibilidad de cambiar de estrategia en referencia a los movimientos de la otra parte involucrada. Por el punto de vista de la Teoría de Juegos, se basa principalmente en el Equilibrio de Nash, donde se intenta buscar la mejor jugada donde ambos jugadores sean beneficiados; sin embargo, esto no siempre es garantizado, incluso para uno o más jugadores, por lo que la solución de conflictos puede no beneficiar a nadie, además utiliza una matriz de resultados, la cual tiene todos los posibles resultados de todos los posibles movimientos, por lo tanto una vez que uno de los jugadores realice su jugada, los demás movimientos que se tienen registrados en la matriz en esa jugada quedarían descartados y al tener generada esa matriz, no es posible realizar movimientos que se acoplen a la situación actual.

Es por estas razones que se percibe que el enfoque de sistemas multi-agentes y el razonamiento práctico nor-

mativo pueden solventar estos puntos desfavorables, ya que sería posible ver los movimientos de cada agente que representa a una parte involucrada y la reacción que tiene conforme a las acciones que los demás agentes realizan, así como actualizar la información de como se encuentra el ambiente en ese momento y poder actuar conforme a esa instancia. Pero como se mencionó anteriormente, no podemos descartar los otros dos enfoques realizados ya que es muy posible que sean necesarios para poder generar este nuevo enfoque con sistemas multi-agentes.*

REFERENCIAS

1. Fainshmidt, S., White, G. y Cangioni, C. (2014). Legal distance, cognitive distance, and conflict resolution in international business intellectual property disputes. *Journal of International Management*.
2. Wan, J. (2015). Application of game theory in a patent dispute negotiation. *les Nouvelles*.
3. Organización Mundial de la Propiedad Intelectual. (2018). Otro año record en las solicitudes de títulos de propiedad intelectual.
4. Lex Machina. (2018). Legal analytics. Website.
5. Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.
6. Finin, T. y Labrou, Y. (1997). Semantics for an agent communication language. *Intelligent Agents IV Agent Theories, Architectures, and Languages*, 1365, 209-214.
7. Li, S. y Kokar, M. (2012). *Agent Communication Language*. Springer.
8. García-Camino, A., Rodríguez-Aguilar, A., Sierra, C. y Vasconcelos, W. (2008). Constraint rule-based programming of norms for electronic institutions. *Autonomous Agents and MultiAgent Systems*, 18(1), 186-217.
9. Esteva, M., Padget, J. y Sierra, C. (2002). Formalizing a language for institutions and norms. *Intelligent Agents VIII*, 2333, 348-366.
10. Shams, Z., De Vos, M., Padget, J. y Vasconcelos, W. (2017). Practical reasoning with norms for autonomous software agents. *Engineering Applications of Artificial Intelligence*, 65, 388-399.
11. Subrahmanian, V. y Zaniolo, C. (1995). Relating stable models and ai planning domains. En *Proceedings of the Twelfth International Conference on Logic Programming*. MIT Press.
12. Alrawagfeh, W., y Meneguzz, F. (2015). Utilizing permission norms in bdi practical normative reasoning. *Coordination, Organizations, Institutions, and Norms in Agent Systems X*, 9372, 1-18.
13. Anger, C., Konczak, K., Linke, T., Schaub, T. (2005). A glimpse of answer set programming. *KI*, 19(1), 12. [14] Osorio, M., Navarro, J. y Arrazola, J. (2001). Equivalence in answer set programming. En *International Workshop on Logic-Based Program Synthesis and Transformation*, 57-75.

SOBRE LOS AUTORES



Rodrigo Vidal-López Candidato a Doctor en Ciencias de la Ingeniería por la Universidad Autónoma del Estado de México. Los intereses en investigación son acerca de Sistemas Multi-agentes, Lógica y Métodos Formales, en busca de resolver problemas actuales relevantes y el bien común para la sociedad.



José Antonio Hernández-Servín Actualmente es profesor de tiempo completo en la Universidad Autónoma del Estado de México. En 1999, recibió su licenciatura en Matemáticas y Ciencias Físicas, y una maestría en Matemáticas en 2001 de la UMSNH (Univ. Aut. San Nicolás de Hidalgo). En 2005, obtuvo un doctorado de la Universidad de Nottingham, Reino Unido. Miembro del Sistema Nacional de Investigadores de México, Nivel I.

*Inteligencia Artificial y Propiedad Intelectual*

— Imagen tomada de <http://www.iprhelphdesk.eu/ip-highlights/special-artificial-intelligence-and-IP>. —

ARTÍCULO ACEPTADO

Redes Neuronales Recurrentes y su uso para la conversión de cadenas

Jared D.T. Guerrero-Sosa, Víctor Hugo Menéndez-Domínguez y Víctor Uc-Cetina

En este trabajo se presentan las redes neuronales convencionales y sus principales problemas para trabajar con el almacenamiento del conocimiento previo, dando origen a las redes neuronales recurrentes. Se explica el modelo de red recurrente GRU y un ejemplo de su uso para otro modelo, *seq2seq*, para la conversión de nombres de investigadores de forma abreviada (como muchas veces registran su producción científica) a su nombre completo. Los resultados fueron satisfactorios, demostrando que los modelos, además de traducir cadenas de un idioma a otro, pueden tener otras funciones

Introducción

El cerebro humano realiza varias funciones de forma compleja por medio de las neuronas y sus conexiones. Este aspecto fue la inspiración para las redes neuronales artificiales para la solución de problemas que tienen un alto nivel de complejidad para ser resueltos por medio de la programación convencional [1]. Las redes neuronales se conforman de un conjunto de unidades simples (neuronas), donde cada una toma cierto número de entradas de valor real y generan una salida de valor real.

Existen tres tipos de neuronas: 1) neuronas de entrada, mismas que reciben estímulos desde el exterior; 2) neuronas ocultas, las cuales reciben la información de las neuronas mencionadas anteriormente; y 3) neuronas de salida, encargadas de dar la respuesta del sistema.

Un tipo de red neuronal se basa en una unidad llamada perceptrón, el cual emite como salida 1 si el resultado es mayor que el límite establecido, y -1 en caso contrario. Este tipo de red neuronal ayuda a la solución de problemas de clasificación binaria.

Otro tipo de red neuronal tradicional es la basada en retropropagación, el cual se ha utilizado, para la resolución de problemas como el reconocimiento de voz [2], de código postal en manuscrito [3], de dígitos en manuscrito [4], mientras que por otra parte, se ha estudiado acerca de un vehículo terrestre autónomo [5], entre otros. Básicamente, el algoritmo de retropropagación busca aprender los pesos de una red de varias capas de neuronas con un número fijo de neuronas y conexiones. El peso es la fuerza de las conexiones entre unidades. Es decir, la influencia que tiene la unidad A al conectarse con la unidad B.

El problema con ambos tipos de redes neuronales es que no son útiles en la aplicación de problemas que requieren tener un conocimiento previo por cada instante en que ingresen datos de entrada a la red. Para ello

es necesario que ésta tenga memoria para que con base en el aprendizaje previo, funcione adecuadamente. Esto ocasionó que se realice un nuevo diseño de red neuronal con las especificaciones mencionadas, dando lugar a las redes neuronales recurrentes, las cuales se explican en la siguiente sección, así como el modelo de red de Unidad Recurrente Cerrada (Gated Recurrent Unit, abreviado como GRU) para posteriormente exponer un ejemplo de su aplicación para la conversión de cadenas, más específico, de nombres abreviados de investigadores a sus respectivos nombres completos. Finalmente, se presentan las conclusiones del trabajo.

GRU, un tipo de red neuronal recurrente

Para comprender las redes neuronales recurrentes se puede hacer por medio de una analogía. Cuando una persona lee un libro, entiende cada suceso porque ha leído las páginas previas, no reinicia su conocimiento desde cero para comprender. De la misma forma trabajan las redes recurrentes. Éstas necesitan contexto para realizar las predicciones de manera correcta y de acuerdo con [6], es la parte elemental de este tipo de redes neuronales. Las redes recurrentes trabajan con bucles para que de esta forma la información aprendida sea persistente. Una propuesta para la comprensión del funcionamiento de las redes recurrentes es la propuesta en [7], representada en la Figura 1, donde, en el extremo izquierdo se observa una red recurrente convencional, cada entrada x_t entra a la capa oculta (rectángulo con la letra A) y emite una salida (rectángulo con las letras h_t) t veces; y en el extremo derecho se presenta la misma red recurrente pero de manera desenrollada.

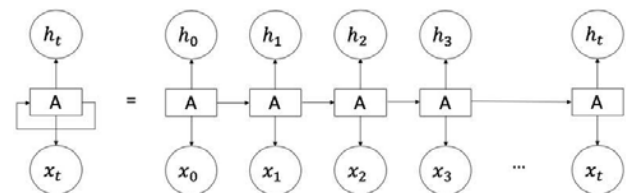


Figura 1. Desarrollo del bucle de una red recurrente.

Existen diseños de redes neuronales recurrentes que utilizan compuertas, es decir, elementos que se encargan de realizar operaciones para lograr emitir la salida final, y las más comunes son las Redes de Gran Memoria de Corto Plazo (LSTM) y con base en ella se ha diseñado otro modelo de red recurrente, conocida como GRU,

mencionada brevemente en la introducción. A continuación, se presenta su funcionamiento por cada paso del tiempo (denominado como t).

1. Compuerta de actualización. Su función es determinar qué parte de la información pasada (es decir, perteneciente a los pasos del tiempo anteriores) deben prevalecer. Observe la Figura 2. Esta compuerta (representada con el recuadro color gris y con el símbolo σ) se calcula

1) multiplicando el valor de la entrada actual (x_t) por su propio peso para la compuerta de actualización; 2) multiplicando el valor de la información previa (h_{t-1}) por su peso para la compuerta de actualización; y 3) sumando ambos resultados. Posteriormente, el resultado pasa por la función sigmoide, cuyo resultado puede ser 0 ó 1. Si el resultado es 0, la información del paso del tiempo anterior se elimina. En caso contrario, se conserva.

Las redes neuronales recurrentes son útiles para problemas en los que se requiere considerar la información previa.

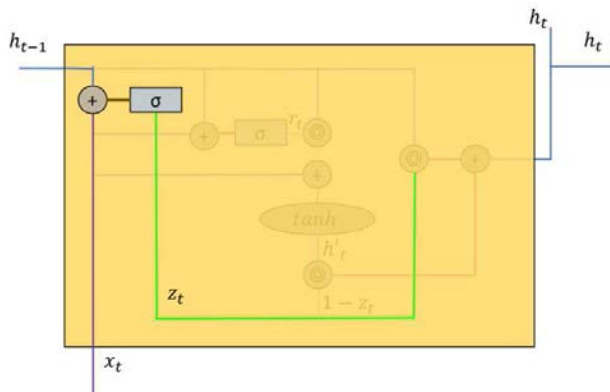


Figura 2. Compuerta de actualización en GRU.

2. Compuerta de reinicio. Tiene como finalidad decidir qué tanto de la información previa se olvidará. Observe la Figura 3. Similar al paso anterior, se representa con el recuadro gris con el símbolo σ y para calcularla, se realiza 1) la multiplicación de la entrada actual (x_t) por su peso para la compuerta de reinicio; 2) la multiplicación de la información previa (h_{t-1}) por su peso para la compuerta de reinicio; y 3) se suman los dos resultados obtenidos. El resultado de la suma pasa a la función sigmoide para obtener un resultado entre 0 y 1.

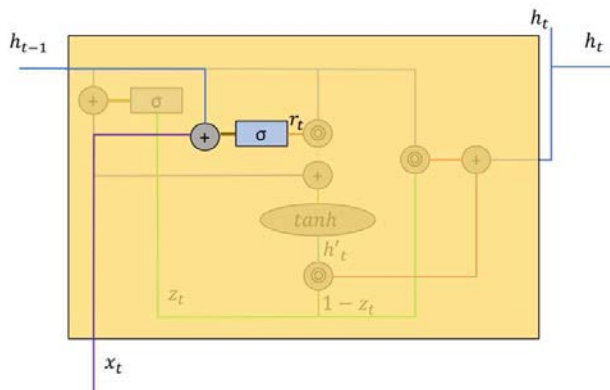


Figura 3. Compuerta de reinicio en GRU.

3. Contenido de la memoria actual. En este paso se presenta cómo las compuertas afectan a la salida final. Observe la Figura 4. El procedimiento a seguir es el siguiente: 1) Multiplicar la entrada (x_t) por su peso y la información previa (h_{t-1}) por su peso; 2) Determinar lo que se eliminará de los pasos anteriores por medio del producto Hadamard (se mencionará como elemento sabio, círculo gris junto a r_t) entre la compuerta de reinicio (r_t) y el resultado de la multiplicación de la información previa (h_{t-1}) por su peso, para que de esta forma se almacene la información previa relevante; 3) sumar los resultados anteriores; y 4) al resultado anterior, aplicar la función tangente hiperbólica (óvalo con la etiqueta $tanh$) para obtener valores entre -1 y 1.

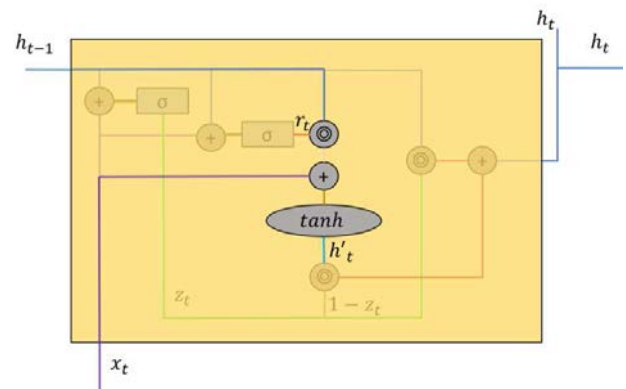


Figura 4. Contenido de la memoria actual en GRU.

4. Memoria final en paso del tiempo actual. Observe la Figura 5. La red requiere obtener la información de la unidad actual de la red y transmitirla a la siguiente unidad. En este caso, se hace uso de la compuerta de actualización para determinar qué recolectar del contenido de la memoria actual (h_t) y qué de la información previa ($h_{(t-1)}$). El procedimiento a seguir es: 1) Aplicar el elemento sabio a la multiplicación de la compuerta de actualización (z_t) por la información previa ($h_{(t-1)}$); 2) aplicar el elemento sabio a la multiplicación de $1 - z_t$ (la compuerta de actualización) por h'_t (que representa el

contenido actual de la memoria); y 3) sumar ambos valores obtenidos previamente. El resultado se envía a la siguiente unidad de la red. Si es la última unidad, entonces el resultado emitido es el final.

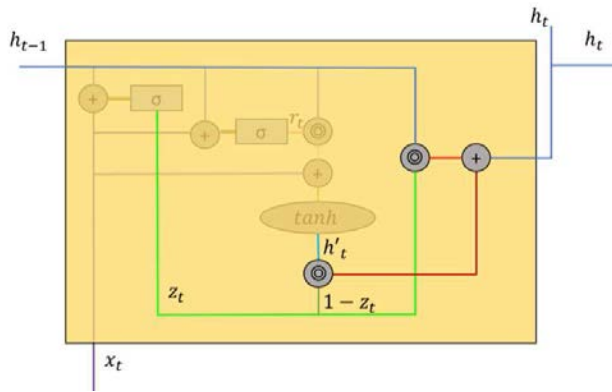


Figura 5. Memoria final en el paso del tiempo actual en GRU.

seq2seq: Uso en un caso real

neuronales recurrentes se encuentra el uso de secuencias para entradas y salidas. Esta idea ha impulsado a la realización de tareas de traducción de un idioma a otro, ya que utilizan como entrada una secuencia de cierto tamaño y la salida es otra secuencia de otro tamaño. Un modelo que cumple con esto es Sequence to Sequence (*seq2seq*) y utiliza dos redes recurrentes: una que opera como codificador y otra como decodificador. Para comprender cómo funciona, observe la Figura 6. El codificador lee cada elemento de la secuencia de entrada y utiliza una celda de red recurrente, ya sea LSTM o GRU. El estado interno (o capa oculta) de la red recurrente codifica la comprensión del modelo de la secuencia. Después de que la secuencia de entrada concluye (observe *<EOS>*, que determina que una entrada ha ingresado por completo), el decodificador produce elementos de la secuencia de salida, uno por uno. La idea es sencilla, y para comprender mejor su uso, en lugar de utilizar *seq2seq* como una herramienta de traducción, se puede utilizar de igual forma para la identificación de nombres de personas escritos en diferentes formatos. El caso real se basa en el siguiente hecho, siguiendo el modelo propuesto por [8].

Los autores muchas veces registran sus nombres con diversos formatos, ya sea abreviando los nombres, utilizando un solo apellido o ambos con un guión, entre otros. Cuando se quiere recuperar la producción científica de los investigadores de una institución a partir de diversos repositorios, no siempre es posible localizar al autor en cuestión por medio de su nombre completo, por lo que es necesario conocer a qué investigador pertenece cada nombre abreviado o bien, en un formato diferente. El modelo *seq2seq* resulta útil para solucionar este problema porque trabaja con secuencias como datos de

entrada y datos de salida y ambas pueden tener longitudes diferentes.

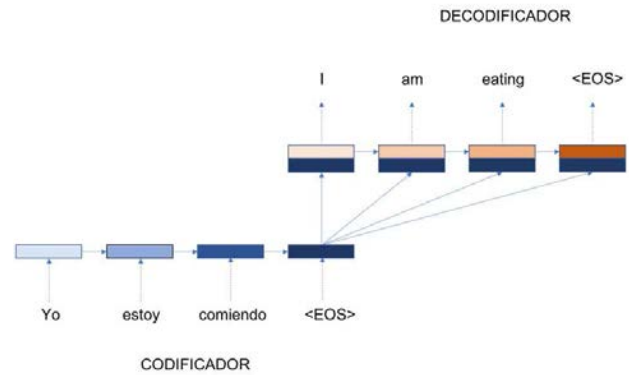


Figura 6. Idea básica de *seq2seq*.

A continuación, se describe el procedimiento realizado, el cual consiste en ingresar el nombre de un autor investigador en un formato diferente al nombre completo y la red neuronal debe emitir como salida el nombre completo de manera correcta, utilizando *seq2seq* con un codificador y decodificador basados en redes GRU.

Carga de archivo con los datos. Con base en una recolección previa de los nombres de autores en la base de datos de citas bibliográficas científicas Scopus con diversos formatos, se creó un archivo donde cada renglón consiste en dos columnas. La primera consiste en el nombre completo del autor, y la segunda corresponde a una de sus variantes (nombre abreviado). Si un autor tiene 3 variaciones, se hacen 3 renglones siguiendo la misma regla. Observe el ejemplo de la Figura 7.

Columna 1	Columna 2
Alberto López Pérez	Alberto López
Alberto López Pérez	A. López
Alberto López Pérez	A. López-Pérez

Figura 7. Ejemplo de archivo con los datos necesarios.

Uso de diccionarios. Como en este caso se trabaja con cadenas y no con números, y para que la red pueda utilizarlas como datos de entrada, es necesario representarlas por medio de vectores, y se utilizaron tres diccionarios: uno que se encarga de asignar una palabra nueva a su índice correspondiente de acuerdo con su orden de aparición (*word2index*); otro con el índice por palabra (*index2word*); y otro para realizar el conteo por cada palabra encontrada (*word2count*). Los tres diccionarios se aplican tanto para los nombres completos (de

la primera columna) como para los nombres abreviados (segunda columna).

Uso del codificador. Para cada palabra que entra al codificador, éste genera un vector de salida y uno de estado oculto, y utiliza el último para la siguiente palabra de entrada.

Uso del decodificador. Se utiliza un decodificador de atención, la cual permite que la red del decodificador se enfoque en una parte diferente de las salidas del codificador para cada paso de las salidas del decodificador. Lleva a cabo los siguientes pasos:

1. Calcular los pesos de cada entrada (conocidos como pesos de atención).
2. El conjunto de los pesos de atención se multiplican por los vectores de salida del codificador para crear una combinación ponderada.
3. La combinación ponderada debe contener información sobre esa parte en específico de la secuencia de entrada, y de esta forma ayuda al decodificador a elegir las palabras adecuadas.
4. El cálculo de los pesos de atención se lleva a cabo en una capa la cual utiliza la entrada del decodificador y el estado oculto como entradas.

Preparación para el entrenamiento. Para entrenar cada par, se necesitan los index2word para la entrada y el resultado objetivo. A ambas secuencias se les agrega <EOS>, que recordando, representa el fin de un nombre. Posteriormente, se realiza lo siguiente:

1. Se ejecuta la oración de entrada a través del codificador.
2. Se realiza el seguimiento de la salida y el último estado oculto correspondiente.
3. El decodificador recibe el <SOS> que indica que un nombre inicia, como primera entrada y el último estado oculto del decodificador como el primer estado oculto.
4. Se ejecuta el entrenamiento de forma iterativa. Al finalizar, el programa se encarga de tomar algunos nombres de la entrada y presenta el resultado objetivo, es decir, el nombre correspondiente de acuerdo con el archivo que contiene los datos del entrenamiento, y la salida real, resultado del conocimiento que adquirió la red neuronal. Dichos resultados se observan en la Tabla 1.

Tabla 1. Resultados del entrenamiento de seq2seq para formatos de nombres de autores.

No.	Entrada	Resultado objetivo	Salida real
1	felipe c zaragoza caulich .	felipe carlos zaragoza caulich .	felipe carlos zaragoza caulich . <EOS>
2	luis erales .	luis asuncion erales aguilar .	luis asuncion erales aguilar . <EOS>
3	david aguilar castro .	david ruben aguilar castro .	david ruben aguilar castro . <EOS>
4	valentin perez molina .	valentin gabriel perez molina .	valentin gabriel perez molina . <EOS>
5	manuel sierra .	manuel rene sierra canton .	manuel rene sierra alfaros . <EOS>
6	miguel ortiz carvajal .	miguel margarita ortiz carvajal .	miguel margarita ortiz carvajal . <EOS>
7	jorge a contreras guerrero .	jorge alberto contreras guerrero .	jorge alberto contreras manzaneros . <EOS>
8	adolfo e mendez vargas .	adolfo ermilo mendez vargas .	adolfo ermilo mendez vargas . <EOS>
9	joel sandoval .	joel antonio sandoval castellanos .	joel antonio sandoval castellanos . <EOS>
10	marco navarro perez .	marco cesar navarro perez .	marco cesar navarro perez . <EOS>

Los aspectos que se pueden inferir a partir de los resultados obtenidos por el modelo son:

- Al ser pocos datos de entrenamiento, comparados a los 135,842 que presenta [8] aún así el modelo aprendió correctamente, además de que los nombres tienen al menos 3 palabras, y las variaciones 2 palabras mínimo.
- De los 10 ejemplos de resultados objetivos presentados por la red, las salidas reales de las filas 5 y 7 no fueron exactas a las esperadas, pero no se alejan tanto, ya que ambas se componen de cuatro palabras y se confunden en una.
- GRU expone todo el contenido de su capa oculta y sigue un procedimiento más simple que LSTM y aún así, el modelo aprendió correctamente, por lo que se ha obtenido un resultado óptimo con un tipo de red neuronal sencilla, pero potente para una

tarea con entradas y salidas de secuencias de diferentes longitudes.

- La utilidad del modelo para el aprendizaje de los nombres en sus diversos formatos se basa en la necesidad de la recolección de la producción científica de un autor, aún cuando éste tiene diversas publicaciones registradas con formatos diferentes de su nombre completo.
- *seq2seq* no sólo ha tenido éxito en tareas de traducción de un lenguaje a otro, y esta tarea es un ejemplo claro. Aunque la mecánica es similar (dada una secuencia como entrada, obtener una secuencia de salida), al utilizar nombres, es muy probable que algunos se repitan por ser muy comunes (por ejemplo, José), y a pesar de esto, el modelo cumple con lo que se desea.

Conclusiones

Se presentó el origen de las redes neuronales, así como los dos modelos tradicionales que se han enfocado en resolver problemas de clasificación y de reconocimiento de patrones. Posteriormente, se expone la importancia del uso de las redes neuronales recurrentes para la aplicación en problemas donde la red requiere tener el conocimiento previo para realizar su finalidad de manera correcta. Se menciona la existencia de las redes de gran memoria de corto plazo (LSTM) y las redes de unidad recurrente cerrada (GRU), siendo este último explicado a través de los cálculos de la compuerta de actualización, de la compuerta de reinicio, del contenido de la memoria actual y de la memoria final en el paso del tiempo actual. Finalmente, se presentó el uso de *seq2seq* utilizando redes GRU como codificador y decodificador, que en lugar de realizar traducciones de un idioma a otro, se realiza

la transcripción de los nombres de autores científicos en un formato diferente a su nombre completo a su nombre real, obteniendo resultados positivos, considerando que al utilizar nombres de personas, es muy probable que algunos se repitan por ser muy comunes (por ejemplo, José), y a pesar de esto, el modelo cumple con lo que se desea debido al uso de una red neuronal recurrente y su naturaleza basada en la consideración del contexto para realizar la predicción, permitiendo mejores resultados al considerar el nombre completo como el contexto.*

REFERENCIAS

1. Graupe, D. (2013). Principles of Artificial Neural Networks. *World Scientific*, 384.
2. Mitchell, T. (1997). Machine Learning. *McGraw-Hill*, 432.
3. Le Cun, Y. (1990). Handwritten Zip Code Recognition with Multilayer Networks. En *Proceedings. 10th International Conference on Pattern Recognition*, pp. 35-40.
4. Le Cun, Y., Boser, B., et al. (1989). Handwritten Digit Recognition with a Back-Propagation Network. En *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, 396-404.
5. Pomerleau, D. (1989). Alvin: An autonomous land vehicle in a neural network. En *Touretzky D. (Ed.) Advances in Neural Information Processing Systems*. Morgan Kaufmann Publishers Inc., 305-313.
6. Mikolov, T. y Zweig, G. (2012). Context dependent recurrent neural network language model. En *2012 IEEE Spoken Language Technology Workshop. Miami, Florida*, 234-239.
7. Olah, C. (2015). Translation with a Sequence to Sequence Network and Attention. Recuperado el 18 de octubre de 2018, de <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
8. Robertson, S. (2017). Translation with a Sequence to Sequence Network and Attention. Recuperado el 18 de octubre de 2018, de https://pytorch.org/tutorials/intermediate/seq2seq-translation_tutorial.html.

SOBRE LOS AUTORES



Jared D.T. Guerrero-Sosa es Ingeniero en Tecnologías de la Información y Comunicaciones por el Instituto Tecnológico de Chetumal, México. Es estudiante de la Maestría en Ciencias de la Computación en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, México. Sus intereses de investigación se enfocan en temas relacionados con los repositorios digitales, la representación del conocimiento y la Informática Educativa.



Víctor Hugo Menéndez-Domínguez es Doctor en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha, España. Es Profesor Titular en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, México. Su trabajo de investigación se centra en temas relacionados con repositorios de documentos digitales, la representación del conocimiento y la gestión de Objetos de Aprendizaje.



Víctor Uc-Cetina es Doctor en Ciencias de la Computación por la Humboldt-Universität zu Berlin, Alemania. Es Maestro en Sistemas Inteligentes por el Instituto Tecnológico y de Estudios Superiores de Monterrey y es Licenciado en Ingeniería de Software por el Instituto Tecnológico de Mérida. Su trabajo de investigación se enfoca en el diseño y aplicación de algoritmos de Aprendizaje Automático que presenten un comportamiento de Inteligencia Artificial. El Dr. Uc-Cetina es profesor y miembro del grupo de investigación Computational Learning and Imaging Research en la Universidad Autónoma de Yucatán.

ARTÍCULO ACEPTADO

Procesamiento de lenguaje natural con aprendizaje profundo

Fernando Pech May, Luis A. López Gómez y Jorge Magaña Govea

El lenguaje natural se refiere a la forma en que las personas se comunican y el procesamiento de lenguaje natural se encarga de analizar y procesar el lenguaje humano a través de herramientas y tecnologías de *software*. Esto involucra distintas áreas de la computación, tales como inteligencia artificial, lingüística computacional, etc. Mediante el procesamiento del lenguaje natural es posible procesar documentos de texto, mensajes SMS, email, página web, etc. y organizar el conocimiento para realizar tareas como análisis de sentimientos, análisis de contexto, generación de resúmenes, traducción automática, sistemas de diálogos, etc.

Dada la importancia del procesamiento del lenguaje natural (PLN), actualmente se han realizado distintos enfoques, entre los que se destacan métodos de aprendizaje profundo para realizar las tareas antes mencionadas. Los métodos de aprendizaje profundo han emergido exitosamente en los últimos años. Este éxito se debe a la capacidad de abordar problemas complejos de aprendizaje a través de múltiples niveles de representación y abstracción que ayudan a dar sentido a datos, imágenes y sonido.

Contexto histórico del aprendizaje profundo (Deep Learning)

En la década de los 90 las herramientas para el PLN eran muy diferentes, los enfoques clásicos basados en reglas fueron reemplazados por métodos de aprendizaje estadísticos. Los campos aleatorios condicionales (CRF) [11] fueron por excelencia el modelado para el etiquetado y segmentación de datos secuenciales o extraer información de documentos. Sin embargo, el aprendizaje profundo ha reemplazado a los CRF ocasionando el desplazamiento del enfoque de ingeniería de características al diseño e implementación de redes neuronales. El principal desafío para el PLN consistió en encontrar buenas características que capturaran bien los datos en los documentos.

Sutton y McCallum [21] describen ampliamente los CRFs y su utilidad en los distintos campos. Para el desarrollo de un CRF se dedica mucho tiempo y esfuerzo para encontrar aquellas funciones que ayudan al modelo a predecir la probabilidad de cada etiqueta dada. La principal debilidad del CRF es que son incapaces de modelar la similitud semántica entre dos palabras. Para superar este problema, muchos CRFs se basan en gazettters (lista de personas, organizaciones, lugares, etc.). Sin embargo, esto es una solución sub-óptima, ya que los gazettters están intrínsecamente limitados y pueden ser muy costosos al desarrollarse. Además, sólo están disponibles para entidades nombradas, de modo que la similitud semántica entre dos palabras es difícil de realizar.

Por otra parte, en el aprendizaje profundo la primera etapa de una red neuronal, para el procesamiento de texto, es por lo general la representación de las palabras en vectores continuos para entrenar el modelo. En este contexto, los *word embedding* [7] son, principalmente, los que han dado el éxito al aprendizaje profundo. Este éxito ha sido en gran parte a la popularidad de word2vec [22]. Word2vec ofrece dos formas de representación vectorial de palabras: 1) bolsa de palabras continuas (CBOW), para predecir la palabra dada su contexto y 2) Skip-gram, para predecir el contexto dada la palabra. Word2vec presenta el problema de asignar cada palabra a un solo *embedding*, lo que implica que, si una palabra tiene varios sentidos, estos están codificados en el mismo vector. Para dar solución a dicho problema, desarrollaron Sense2vec, que es una adaptación de word2vec que utiliza etiquetas supervisadas para distinguir entre los sentidos de la palabra. Otra alternativa para la representación de palabras es mediante vectores globales (GloVe) [17], los cuales son métodos basados en conteo; su entrenamiento es rápido y escala a grandes corpus y toma como base que la ocurrencia de palabras es la principal fuente de información disponible para aprender representaciones de palabras mediante métodos de aprendizaje no supervisado.

Una red neuronal artificial es un paradigma de programación de inspiración biológica que permite a una computadora aprender de los datos observados.

Aprendizaje profundo para el procesamiento de lenguaje natural

Una red neuronal artificial (RNA) es un paradigma de programación de inspiración biológica que permite a una computadora aprender de los datos observados. Está compuesto por un gran número de elementos de procesamiento interconectados y que trabajan conjuntamente para resolver un problema. Consta de 3 capas: 1) capa de entrada, 2) capa oculta y 3) capa de salida. La RNA tiene un número limitado de capas, lo que hace imposible representar problemas complejos. Las RNAs que consisten de más capas se le denomina redes neuronales profundas, y entrenarlos se denomina aprendizaje profundo. Por consiguiente, el aprendizaje profundo es un conjunto de técnicas poderosas para el aprendizaje en redes neuronales y con ella es posible resolver problemas complejos.

Existen distintas técnicas de aprendizaje profundo, las de mayor relevancia y más utilizadas para el procesamiento del lenguaje natural son las redes neuronales recurrentes. Sin embargo, algunos enfoques están haciendo uso de las redes neuronales convolucionales para el mismo propósito.

Redes neuronales recurrentes (RNN)

Las redes neuronales recurrentes (RNNs) son modelos poderosos de aprendizaje profundo, y son excelentes para el procesamiento de lenguaje natural. Construyen una representación vectorial para cada token, leyendo token por token, y “recordando” información importante. Es decir, permiten que la información persista y son capaces de manejar datos secuenciales. Se llaman recurrentes porque realizan la misma tarea para cada elemento de una secuencia, y la salida depende de los cálculos previos. Otra forma de pensar sobre las RNNs es que tienen una memoria que captura información sobre lo que se ha calculado hasta ahora. En algunas tareas que requieren predecir la siguiente palabra en una oración, las RNNs pueden tener un mal desempeño; sin embargo, actualmente se han desarrollado distintas mejoras en las RNNs que permiten hacer frente a las deficiencias del modelo original.

Existen tres versiones RNN: 1) la versión simple, 2) LSTM (Long Short Term Memory) y 3) GRU (Gated Recurrent Units). La diferencia entre ellas radica en la arquitectura de la unidad de memoria.

RNN simple. Tiene la forma de una cadena de módulo repetitivo de red neuronal. Tiene una estructura muy simple con una sola capa tanh, que es la función de activación, con valores entre -1 y 1. En la Figura 1 se muestra la estructura de un fragmento de RNN simple. Se tiene una entrada X_t y una salida Y_t . Un ciclo permite que la información pase de una etapa de la red a la siguiente, dicho ciclo es un estado interno H . Esto

puede considerarse como copias múltiples de la misma red, cada copia pasando un mensaje a su sucesor.

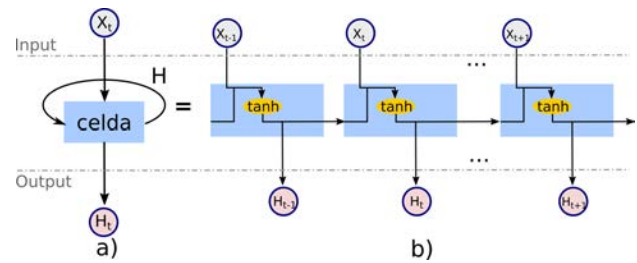


Figura 1. Estructura de la red neuronal simple.

Un atractivo de la RNN es que puede ser capaz de conectar información previa a la tarea actual. En algunos problemas sólo se requiere información reciente para realizar la tarea actual (por ejemplo, predecir la siguiente palabra en función a las anteriores), por lo que no se necesita ningún contexto adicional (Figura 2a). Sin embargo, existen casos donde se demanda mayor contexto, por lo que se requiere información en celdas anteriores y, a medida que crece la brecha, las RNNs no pueden aprender a conectar la información (Figura 2 b). A pesar de que las RNNs exhiben una capacidad superior de modelar problemas, en tiempos no lineales, son incapaces de aprender a largo plazo [5]. Este inconveniente puede solucionarse con las RNN-LSTM.

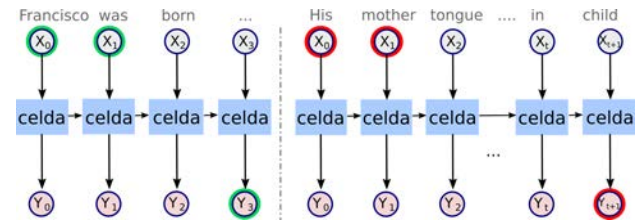


Figura 2. Red neuronal simple: a) con información reciente y b) RNN que requiere información de celdas anteriores lejanas.

RNN-LSTM. Consiste de múltiples puertas (gates) y son capaces de aprender dependencias a largo plazo debido a que su estructura tiene la capacidad de recordar información por periodos largos de tiempo, olvidar información innecesaria y exponer cuidadosamente la información en cada paso del tiempo. Consta de cuatro capas que interactúan; por lo tanto, el módulo de repetición realiza más operaciones: esto permite que recuerde dependencias a largo plazo.

En la Figura 3 se muestra la estructura RNN-LSTM. Consta de 4 capas (círculos o elipses en amarillo) y de operaciones puntuales (círculos o elipses en verde), tales como la suma de vectores. Cada línea lleva un vector completo, desde la salida de un nodo hasta las entradas de otros. La clave de las RNN-LSTM es el estado de las celdas (C_t), que es la línea roja horizontal que

se extiende en el diagrama de la figura. El estado de la celda corre directamente por todo el diagrama con interacciones lineales menores. También tiene la capacidad de remover o agregar información al estado de la celda regulado cuidadosamente por estructuras llamadas puertas. Se compone de 3 puertas (entrada, olvido y salida) y son las que controlan el paso de la información dentro o fuera de la celda. Las puertas constan de una capa de red neuronal llamada sigmoide (σ) y una operación puntual de multiplicación (líneas verdes de la figura).

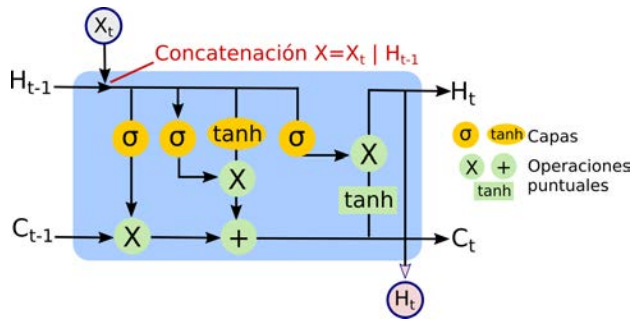


Figura 3. Estructura de RNN-LSTM.

Las RNN-LSTM constan de cuatro etapas. En la Figura 4 se muestran cada una de ellas y en la Tabla 1, sus expresiones. A continuación, se describe cada etapa:

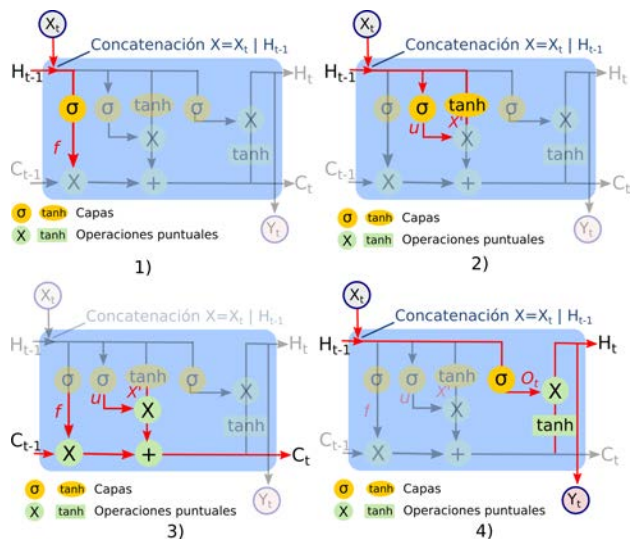


Figura 4. Etapas de RNN-LSTM: 1) Forget gate layer, 2) capa de puerta de enlace y valores candidatos, 3) actualización de nueva celda y 4) nueva salida.

1. Decidir la información a arrojar del estado de la celda. Esta decisión se toma mediante una capa de sigmoide llamada “Forget gate layer” (f). La concatenación X de h_{t-1} y la entrada X_t emite una salida entre 0 y 1 para cada número en el estado

de la celda C_{t-1} , donde 1 representa “mantenerlo por completo” y -1 “deshacerlo por completo”.

2. Decidir la nueva información que se almacenará en el estado de la celda. Consta de dos partes: 1) una capa sigmoide llamada *capa de puerta de entrada* (u) que decide los valores a actualizar y 2) capa \tanh , que crea un vector de nuevos valores candidatos (X') que podrán agregarse al estado.
3. Actualización del estado de la celda anterior C_{t-1} . Para actualizar C_{t-1} a una nueva celda C_t , se realiza con las etapas anteriores multiplicando f con el estado anterior C_{t-1} y sumándolo con la multiplicación de u con la concatenación X' .
4. Decidir lo que se generará como salida. Para generar la salida se basará en el estado de la celda, pero será una nueva versión de salida filtrada H_t (nuevo estado oculto). Para esto, primero se ejecuta una capa sigmoide que decide la parte del estado de la celda que se generará como salida (O_t). Luego se pone el estado de la celda C_t a través de \tanh , para que los valores sean entre -1 y 1, y multiplicarlo por O_t .

Tabla 1. Expresiones de las etapas de RNN-LSTM.

Etapa	Nombre	Representación
1	Forget gate layer	$f = \sigma(X \cdot W_f + b_f)$
2	Input gate layer	$u = \sigma(X \cdot W_u + b_u)$
2	New candidates values	$X' = \tanh(X \cdot W_c + b_c)$
3	New cell	$C_t = f \cdot C_{t-1} + u \cdot X'$
4	Output gate	$O_t = \sigma(X \cdot W_c + b_c)$
4	New output	$H_t = O_t \cdot \tanh(C_t)$

Debido a que LSTM es unidireccional, sólo es capaz de conservar información del pasado, no proporciona buenos resultados en algunas tareas de procesamiento de lenguaje natural como detección de entidades y similitud semántica [18]. Esto se debe, principalmente, a que con LSTM sólo se detecta el contexto izquierdo de la palabra cuando se le asigna una etiqueta. Sin embargo, existen LSTM bidireccionales denominados BILSTM, que procesan los datos con dos capas separadas, una de pasado a futuro y otra de futuro a pasado.

LSTM-bidireccional (BILSTM). Se constituye de dos capas: una capa que aprende representaciones previas y otra que retrocede en el tiempo, lo que ayuda a aprender de representaciones futuras. Son capaces de analizar oraciones hacia adelante y hacia atrás, lo que mejora el rendimiento de la RNN. Entrenan dos LSTM en lugar de uno en la secuencia de entrada. El primer LSTM es unidireccional (*forward*) y funciona tal como se ha explicado anteriormente; la segunda LSTM es una copia invertida (*backward*) de la secuencia de entrada

(ver Figura 5), lo que puede proporcionar un contexto adicional a la red y un aprendizaje más rápido y completo sobre el problema. Ambas capas producen un vector de salida para cada palabra.

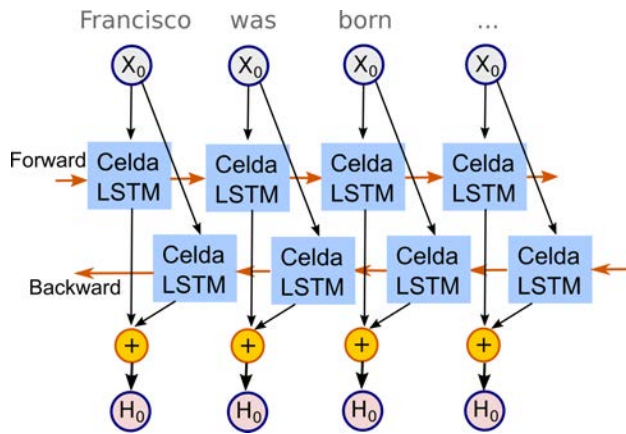


Figura 5. LSTM-bidireccional.

El reemplazo de la capa LSTM simple a BILSTM ha sido de gran impacto en su rendimiento. Asimismo, se han utilizado en la clasificación fonemas y reconocimiento de voz [8].

RNN-encoder-decoder. También conocida como Seq2seq [5], se ha convertido en un modelo para distintas tareas, tales como traducción, reconocimiento de voz, resumen de texto, etc. y ha obtenido buenos resultados. Consiste de dos módulos de RNN (ver Figura 6), uno para codificar (encoder) y el otro para decodificar (decoder). El Encoder convierte una secuencia de símbolos en un vector de características de tamaño fijo y lee la secuencia de entrada (oración), procesa palabra por palabra y emite un contexto (una función del estado oculto final del encoder) que servirá en el siguiente paso. El decoder genera una secuencia de salida, una palabra a la vez mientras mira el contexto y la palabra anterior durante cada paso del tiempo. El contexto (obtenido del encoder) es proporcional el estado inicial del decoder.

También se han presentado distintas estrategias que, unidas con seq2seq, permiten obtener mejores resultados en las RNN, tales como *máquinas neuronales de Turing*, *interfaces de atención*, *tiempo de cálculo adaptativo* y *programadores neuronales*. Entre estas estrategias destacan las interfaces de atención.

Las interfaces o mecanismos de atención en una red neuronal se basan vagamente en la atención visual de las personas, es decir: se enfocan o prestan atención a secuencias en particular. Es un mecanismo que, aplicado con BILSTM, ha obtenido buenos resultados en distintos trabajos de PLN. Este mecanismo se ha utilizado para distintos problemas, tales como traducción de texto, clasificación de imágenes y resumen de texto. La distribución de la atención es una distribución de probabilidad

sobre las palabras en el texto fuente y dice a la red neuronal dónde buscar para ayudarlo a producir la siguiente palabra.

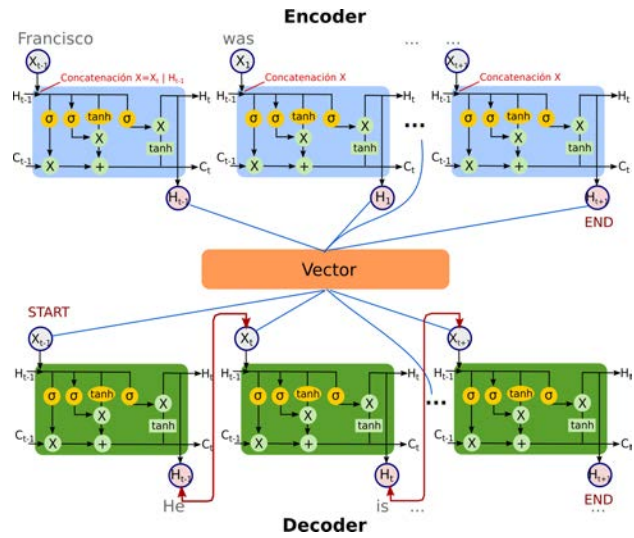


Figura 6. RNN-encoder-decoder.

RNN-GRU. Las RNN GRU (Gated Recurrent Unit) están diseñadas similarmente a las LSTM, y producen resultados similares. Utiliza dos puertas (actualización y reinicio), en lugar de las 3 que utiliza RNN-LSTM. El funcionamiento de las puertas es parecido a las LSTM. Básicamente, estas dos puertas deciden qué información se debe pasar a la salida. Lo especial de ellos es que pueden ser entrenados para mantener la información a largo plazo.

Por otra parte, las redes neuronales convolucionales (CNN) están dirigidas al procesamiento de imágenes y son el núcleo de un gran número de sistemas de visión por computadora. Sin embargo, recientemente se han propuestos trabajos de PLN utilizando CNN con resultados interesantes [10].

Enfoques para el procesamiento de lenguaje natural utilizando Deep Learning

Las tareas del PLN se pueden clasificar en extracción de información, clasificación de texto, sumarización, búsqueda de respuesta (QA) y traducción automática. En la literatura se han presentado distintos enfoques para la solución de estas tareas mediante el aprendizaje profundo. En este trabajo se analizaron algunos enfoques; sin embargo, existen muchos trabajos de procesamiento de texto que utilizan el aprendizaje profundo y, por cuestión de espacio, sólo se mencionan algunos.

La extracción de información es el proceso de usar algoritmos para extraer de manera implícita o explícita información en un corpus de documentos para el reconocimiento de entidades nombradas, extracción de eventos,

extracción de relaciones, etc. (ver Tabla 2). La clasificación de texto es el proceso de asignar o clasificar las palabras (ver Tabla 3). La generación de resúmenes es la tarea de producir texto a partir de uno o más textos que contienen una porción significativa de la información respecto al texto original (ver Tabla 4). La traducción automática involucra técnicas para traducir documentos de un lenguaje a otro (ver Tabla 5).

Tabla 2. Enfoques de extracción de información con aprendizaje profundo.

Trabajo	Tarea	Enfoque
[8]	NER	Extracción de entidades nombradas con LSTM-bidireccional y con un CNN a nivel de caracteres.
[19]	NER	LSTM bidireccional y uso de CRF para el etiquetado de entidades.
[7]	Extracción de eventos	Uso de CNN para extraer automáticamente las funciones de nivel léxico y oración de palabras.
[26]	Extracción de eventos	RNN-encoder-decoder para identificar eventos y roles el texto.
[40]	Extracción de relaciones	LSTM-bidireccional y una CNN para el reconocimiento de la entidad y clasificación relaciones.
[33]	Extracción de relaciones	GRU basado en la atención con un mecanismo de copia. Utiliza una estructura de datos llamado mecanismo de cobertura.
[20]	POS-Tagging	LSTM-bidireccional para el etiquetado gramatical.

Tabla 3. Enfoques en clasificación de texto con aprendizaje profundo.

Trabajo	Tarea	Enfoque
[37]	Clasificación	Proponen la decodificación de tweets con LSTM, el estado oculto es usado para predecir la polaridad de los sentimientos.
[10]	Clasificación	Utilizan CNN para el análisis de sentimiento y RNN para el reconocimiento de entidades nombradas.
[16]	Clasificación de texto	Usa una arquitectura híbrida que combina una red de creencia profunda y regresión softmax.

Tabla 4. Enfoques para la generación de resúmenes que utilizan métodos de aprendizaje profundo.

Trabajo	Tarea	Enfoque
[25]	Extracción de resúmenes extractivo	RNN-GRU Extrae resumen de texto utilizando RNN con GRU.
[36]	Extracción de resúmenes extractivo	Uso de Restricted Boltzmann Machine (RBM).
[38]	Extracción de resúmenes extractivo	Deep auto-encoder (AE).
[24]	Extracción de resúmenes abstractivo	RNN encoder-decoder.
[27]	Extracción de resúmenes abstractivo	CNN para la extracción de resúmenes de texto.

Tabla 5. Enfoques para traducción automática que utilizan métodos de aprendizaje profundo.

Trabajo	Tarea	Enfoque
[2]	Traducción automática	RNN-encoder-decoder.
[9]	Traducción automática	RNN-encoder-decoder.
[31]	Traducción automática	Uso de GRU con mecanismos de atención.
[1]	Traducción automática	RNN con mecanismos de atención y una capa de CNN.

Conclusiones

El aprendizaje profundo es una poderosa herramienta de aprendizaje, capaz de solucionar problemas que van desde la clasificación hasta el modelado de secuencias y árboles Markovianos. Se ha convertido en un nuevo campo de aprendizaje automático y ha ganado amplio interés en diferentes áreas de investigación, entre ellas el procesamiento de lenguaje natural. Una de las técnicas de aprendizaje profundo más utilizadas para el procesamiento de lenguaje natural son las redes neuronales recurrentes. En este documento han sido descritas, así como sus variantes. Asimismo, se analizaron distintos trabajos que hacen uso de ellas para el procesamiento de texto. Dichos trabajos han demostrado las ventajas sobre los métodos tradicionales de aprendizaje automático en algunos campos. El aprendizaje profundo se ha aplicado al procesamiento del lenguaje natural con cierto éxito, y los resultados parecen prometedores; sin embargo, los resultados son preliminares y aún están lejos de ser satisfactorios.*

REFERENCIAS

- Ahmed, K., Keskar, N.S. y Socher R. (2018). Weighted Transformer Network for Machine Translation. CoRR. abs/1711.02132.
- Bahdanau D., Cho, K. y Bengio Y. (2016). Neural machine translation by jointly learning to align and translate. arXiv:1409.0473].
- Chen, Y., Xu, L., Liu, K., Zeng, D. y Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. En *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 1, 167-176.
- Chiu, J. y Nichols, E. (2015). Named entity recognition with bidirectional lstm-cnns. CoRR, abs/1511.08308.
- Cho, K., Merriënboer, B., Gülcehre, C. Bahdanau, D., Bougares, F., Schwenk, H. y Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. En *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1724-1734.
- Du, J., Zhang, Y., Luo, J., Jia, Y., Wei, Q., Tao, C. y Xu, H. (2018). Extracting psychiatric stressors for suicide from social media using deep learning. *BMC medical informatics and decision making*, 4, 43.
- Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*.
- Graves, A., Jaitly, N. y Mohamed, A. (2013). Hybrid speech recognition with Deep Bidirectional LSTM. *IEEE Workshop on Automatic Speech Recognition and Understanding*, 273-278.
- Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y. y Guan, R. (2018). Text classification based on deep belief network and softmax regression. *Neural Computing and Applications*, 29, 61-70.
- Zhang, D., Yan, Jin, W. y Wang, Z. (2017). Combining knowledge with deep convolutional neural networks for short text classification. En *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2915-2921.
- Lafferty, J., McCallum, A. y Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. En *Proceedings of the Eighteenth International Conference on Machine Learning*, 282-289.

12. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. y Dyer, C. (2016). Neural architectures for named entity recognition. En *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, 260-270.
13. Ling, W., Luís, T., Marujo, L., Ramón, A., Amir, S., Dyer, C., Black, A. y Trancoso, I. (2015). Finding function in form: Compositional character models for open vocabulary word representation. CoRR, abs/1508.02096.
14. Nallapati, R., Zhai, F. y Zhou, B. (2016). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. CoRR, abs/1611.04230.
15. Nguyen, T., Cho, K., y Grishman, R. (2016). Joint event extraction via recurrent neural networks. En *Proc. of Conference of the Association for Computational Linguistics*, 300-309.
16. Paulus, R., Xiong, C. y Socher, R. (2018). A deep reinforced model for abstractive summarization. En *International Conference on Learning Representations*.
17. Pennington, J., Socher, R. y Manning, C. (2014). Glove: Global vectors for word representation. En *EMNLP*, 14, 1532-1543.
18. Sankaran, B., Mi, H., Al-Onaizan, Y. y Ittycheriah, A. (2016). Temporal attention model for neural machine translation. CoRR, abs/1608.02927.
19. Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hirschler, J., Junczys-Dowmunt, M., Laubli, S., Miceli, A., Mokry, B. y Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. En *Proceedings of the Software Demonstrations*, 65-68.
20. Sun, M., Li, X., Wang, X., Fan, M., Feng, Y. y Li, P. (2018). Logician: A unified end-to-end neural approach for opendomain information extraction, 556-564.
21. Sutton, C. y McCallum, A. (2012). An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4, 267-373.
22. Trask, A., Michalak, P. y Liu, J. (2015). sense2vec - a fast and accurate method for word sense disambiguation in neural word embeddings. CoRR, abs/1511.06388.
23. Verma, S. y Nidhi, V. (2017). Extractive summarization using deep learning. CoRR, abs/1708.04439.
24. Wang X., Liu, Y., Sun, C., Wang, B. y Wang, X. (2015). Predicting polarities of tweets by composing word embeddings with long short-term memory. En *Proc. of the Association for Computational Linguistics*, 1, 1343-1353.
25. Yousefi-Azar, M. y Hamey, L. (2017). Text summarization using unsupervised deep learning. *Expert Syst. Appl.*, 68, 93-105.
26. Zheng, S., Hao, Y., Lu, D., Bao, H., Xu, J., Hao, H. y Xu, B. (2017). Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257, 59-66.

SOBRE LOS AUTORES



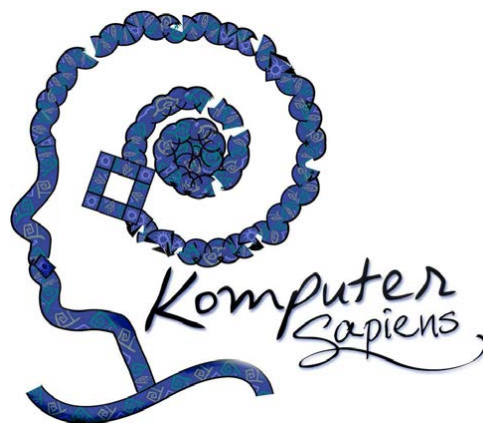
Fernando Pech May es profesor investigador del Instituto Tecnológico Superior de los Ríos. Doctor en ciencias de la Computación por el CENIDET y Maestro en ciencias de la computación por el CINVESTAV. Sus áreas de interés son el procesamiento de lenguaje natural, web semántica y Deep Learning.



Luis A. López Gómez es profesor investigador del Instituto Tecnológico Superior de los Ríos. Sus áreas de interés son las ontologías y el procesamiento de texto.



Jorge Magaña Govea es profesor investigador del Instituto Tecnológico Superior de los Ríos. Maestro en Ciencias de la Computación por la UJAT. Sus áreas de interés son las ontologías y la web semántica.



ARTÍCULO ACEPTADO

Detección de Obstáculos Móviles Aplicada a la Conducción Autónoma Vehicular

Lauro Reyes Cocoletzi, Ivan Olmos Pineda y J. Arturo Olvera López

Obtención de información del mundo que nos rodea

Una de las principales fuentes de información que tiene el ser humano para percibir el ambiente que lo rodea es el sentido de la vista. A través de los ojos se puede tener conocimiento de la forma, el tamaño, color e incluso estimar la distancia del observador hacia objetos cercanos. Además, con el sentido de la vista el humano es capaz de identificar y distinguir las características del entorno en el cual se encuentra.

Con los avances recientes de la tecnología se ha intentado alcanzar e igualar de forma artificial las capacidades del sentido de la vista. Sin embargo, la visión humana es tan compleja que tratar de desarrollar sistemas que iguallen todas sus capacidades es computacionalmente complejo y costoso. Por ello, las investigaciones recientes en el área de visión artificial o computacional se han centrado en resolver problemas específicos.

Un ejemplo de estos problemas, que ha captado la atención de una parte importante del sector académico y de investigación, es el uso de visión artificial aplicada al sector automotriz orientado al manejo autónomo. Es de especial interés el desarrollo de sistemas capaces de identificar obstáculos en el recorrido de un automóvil, además de estimar sus trayectorias para evitar colisiones y daños a terceros.

Tecnología desarrollada para vehículos autónomos

Actualmente existen distintos sistemas para el manejo autónomo de vehículos. Por ejemplo, la compañía *Daimler-Wymo-Google* desarrolla un sistema de taxi autónomo basado en sensores laser (lidar), el cual realiza un mapeo tridimensional a través de una nube de puntos para reconstruir el ambiente real en un rango de distancia de hasta 120 metros [1].

Los sensores lidar tienen un precio elevado por lo que es un dispositivo poco viable de instalar en un gran volumen de automóviles por su costo; además, sufren del problema de interferencia mutua (un sensor puede leer información de otro sensor cercano), provocando una distorsión en la señal del mapeo tridimensional.

Otros trabajos enfocados al desarrollo de vehículos

autónomos utilizan sensores tipo radar. Estos radares se basan en la emisión de ondas electromagnéticas que se reflejan en el objetivo (obstáculos) y retornan hacia el sensor. A partir de este eco se puede realizar una reconstrucción del entorno que rodea al vehículo; sin embargo, los radares sufren de problemas de interferencia mutua y ruido ambiental.

Finalmente, los sistemas con cámaras de video no emiten señales de energía sino que captan la luz visible que sirve para la construcción de imágenes del ambiente que los rodea. La imagen capturada proporciona información del entorno y sólo en condiciones ambientales adversas la adquisición de información se puede dificultar.

Por ejemplo, un automóvil dotado de autonomía en conducción se debe equipar con un sistema de visión (cámaras de video y algoritmos computacionales), para detectar gran variedad de objetos en su trayectoria (Figura 1).



Figura 1. Representación de un automóvil con un sistema computacional capaz de detectar peatones, ciclistas u otros vehículos.

La visión estereoscópica en el ser humano se encarga de integrar dos imágenes del ambiente en una sola y crear una representación tridimensional del medio que lo rodea. La percepción de profundidad del ambiente se obtiene debido a que las imágenes captadas por las retinas son ligeramente distintas entre sí por la diferencia de posición de los ojos, lo que resulta en un efecto de relieve [2]. Por lo tanto, emular este sistema de visión biológico en un sistema de visión artificial requiere de visión estereoscópica.

Al utilizar VEC es posible aproximar con cierto rango de error la profundidad de cada objeto en la escena.

Además, para que un sistema de visión artificial cuente con autonomía de manejo, éste debe lograr estimar profundidad. Por lo que, considerando el caso de la visión estereoscópica humana, el camino natural para un sistema artificial es dotarlo de dos cámaras de video que sirvan como ojos artificiales.

Visión estereoscópica computacional

La visión estereoscópica computacional (VEC) tiene como principal objetivo construir a partir de la información bidimensional, capturada por dos cámaras de video, una representación tridimensional. Esto se logra al colocar un par de cámaras a una distancia controlada para obtener dos imágenes, las cuales se procesan para identificar puntos de correspondencia y recuperar la posición espacial [3]. En general la VEC se basa en 4 etapas que se describen a continuación:

1. Obtener dos imágenes separadas a una distancia conocida $d/2$ (Figura 2).
2. Identificar puntos comunes en ambas imágenes, es decir, cada imagen capturada por el par de cámaras está compuesta por puntos (píxeles) que conforman las proyecciones de la escena del mundo real que tienen similitud o correspondencia entre sí. Por ejemplo, el punto P en la figura 2, tanto en las imágenes como en el esquema, representa el píxel de correspondencia.
3. Determinar la profundidad Z del punto proyectado P mediante geometría analítica y otras herramientas matemáticas, es decir, la proyección sobre las líneas Z_1, Z_2 con respecto a los lentes de la cámara y la distancia focal (ver figura 2).
4. Calcular la proyección del punto P sobre las líneas Y_1, Y_2 y su respectiva posición estimada (Figura 2). La profundidad es inversamente proporcional a la disparidad de la imagen, lo que sugiere que la distancia horizontal puede incrementarse para mejorar la exactitud de la profundidad medida.

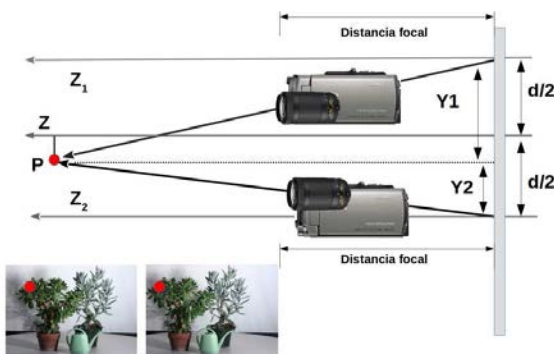


Figura 2. Esquema básico de un sistema VEC.

Al procesar las 4 etapas mencionadas, es posible estimar la posición espacial de los objetos detectados en la escena, lo cual es fundamental para la detección de obstáculos en la ruta de desplazamiento de un vehículo autónomo.

Detección de obstáculos

El principal reto de los algoritmos basados en VEC es que pueden presentar errores en la obtención de la información tridimensional. Esto se debe a diferentes características del ambiente, como: apariencia similar entre los objetos observados, oclusiones, variaciones de iluminación, ruido, etc. Aún con estos inconvenientes VEC es la forma más cercana a la visión estereoscópica humana, por lo que se deben seguir explorando técnicas para lograr cubrir las deficiencias de los algoritmos.

Distintos trabajos de investigación relacionados a la detección de obstáculos [4-7] se basan en coincidencia semi-global (*Semi-Global-Matching SGM*), mapas de disparidad y regiones de interés (*region of interest ROI*) para complementar a VEC (Figura 3).

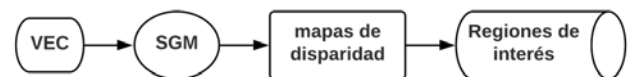


Figura 3. Sistema de detección de obstáculos.

A continuación se describe la metodología por bloques para dotar de percepción del entorno a los vehículos autónomos.

Método SGM y mapas de disparidad

El método denominado SGM tiene como base encontrar correspondencias pixel a pixel de imágenes estereoscópicas, haciendo múltiples aproximaciones con lo que se pretende lograr una correspondencia global. Este método propone una función que penaliza los cambios abruptos de disparidad en la vecindad del píxel, lo que implica omitir las regiones donde no hay correspondencia (falsos positivos), es decir, evitar la asignación de disparidades correctas cuando en realidad son erróneas. Tomando en cuenta las consideraciones mencionadas, se pueden construir los mapas de disparidad [4].

El mapa de disparidad es una imagen donde el nivel de gris de los píxeles corresponde al valor de disparidad, proporcional a la profundidad del punto en el espacio real. Los puntos más alejados poseen un valor de disparidad inferior y por tanto tienen niveles de gris próximos al tono negro, mientras que, los puntos cercanos tienen valores de disparidad altos y por tanto próximos al tono blanco [5]. En la Figura 4 se observa la captura izquierda-derecha de la imagen original y el mapa de disparidad resultante. Los mapas de disparidad son la primera parte del proceso, así que, zonas poco definidas y presencia de ruido son inherentes al resultado obtenido.

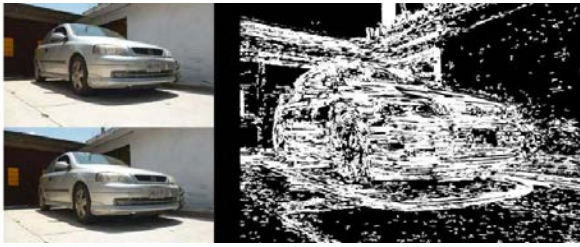


Figura 4. Imagen izquierda, derecha y mapa de disparidad procesado.

Detección ROI

Las ROI son el subconjunto de datos (píxeles) con características similares que identifican espacialmente la presencia de un objeto en una imagen (área o región). Como se mencionó, el método de disparidad sirve para tener una primera aproximación de los obstáculos presentes en la escena, sin embargo, es necesario refinar los resultados obtenidos para la detección de los obstáculos. Esto se logra a través de la detección de bordes en los objetos.

Las regiones obtenidas del procesamiento de bordes proporcionan áreas dentro de la escena que son viables de contener obstáculos (ROI), es decir, se realiza este proceso para diferenciar obstáculos con respecto al fondo.

La Figura 5 muestra la segmentación obtenida del proceso realizado en la detección de ROI.



Figura 5. Segmentación de las regiones de interés.

En la segmentación se eliminan texturas y se presentan los objetos con regiones y características similares de intensidad de color.

Hasta este punto se ha planteado como detectar obstáculos, sin embargo, para un automóvil dotado de visión artificial esto no basta ya que es imperativo determinar las posibles trayectorias de los obstáculos. Por lo tanto, se requiere una etapa de estimación de trayectorias, lo que implica el análisis de imágenes en movimiento.

Análisis de movimiento

Al analizar trayectorias, se debe tomar en cuenta que existe un movimiento relativo entre cámara y objetos. De esta forma, se obtiene una secuencia de imágenes entre las que existen pequeñas diferencias derivadas del desplazamiento. El análisis de movimiento considera la secuencia de imágenes capturadas como un patrón de flujo óptico de los objetos en una escena debido al cambio de posición entre un observador (cámara) y el ambiente que lo rodea (escena capturada)[6]. A continuación se menciona este proceso con más detalle.

El flujo óptico realiza un seguimiento continuo de objetos en un conjunto de imágenes relacionadas en el espacio-tiempo.

Flujo óptico

En visión artificial la percepción del movimiento considera cambios ligeros en la imagen que son percibidos como un flujo de puntos (píxeles) en el espacio. Esto se puede considerar como un arreglo vectorial que expresa la velocidad de desplazamiento instantánea de cada punto [7]. A este arreglo de vectores de velocidad se le llama flujo óptico y se obtiene de la secuencia de imágenes capturadas en video.

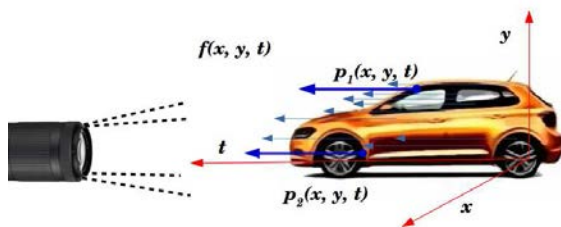


Figura 6. Función espacio-temporal del flujo óptico.

En la Figura 6 se muestra un par de vectores (p_1, p_2) que representan dos puntos distintos que componen a un objeto en movimiento capturado en imagen (un cuadro del video).

Para calcular el flujo óptico (figura 6), se considera la secuencia de imágenes en movimiento como una función continua en 3 dimensiones, x, y , y *tiempo*. La posición espacial de los objetos en movimiento están representados por columnas (x) y filas (y) que componen a la imagen. La posición temporal de los objetos es la posición espacial en distintos intervalos de tiempo (t).

El número total de puntos-vectores que conforman al objeto se considera una función $f(x, y, t)$, donde esta función representa la secuencia de imágenes en diferentes tiempos (cuadros que componen el video).

Las características relevantes que se deben de identificar en los objetos detectados en una escena y que tienen movimiento son: cambios de velocidad, movimiento co-

mún y consistencia del objeto a través del desarrollo de la escena, conocimiento *a priori* del tipo de movimiento de los objetos y/o del observador.

Un uso común del análisis de una secuencia de video es el seguimiento de objetos en movimiento, éste tiene múltiples aplicaciones prácticas tales como: el seguimiento de personas o vehículos para fines de seguridad, el seguimiento de las partes del cuerpo de una persona para reconocer actividades o ademanes, entre otras.

Seguimiento y estimación de trayectoria

El seguimiento en una secuencia de video consiste en determinar la posición y velocidad de un punto (o de una región u objeto) en una imagen, dada su posición y velocidad en una secuencia anterior de imágenes.

En primera instancia se puede determinar el recorrido de obstáculos en el transcurso del tiempo (trayectoria), a continuación se puede tener una noción de estimación de sus recorridos *a posteriori* [8].

La Figura 7 ejemplifica el seguimiento del vehículo rojo a través de los carriles de la carretera, así como, las posibles vías de desplazamiento de los vehículos en la escena (se ilustra con las flechas apuntando a las distintas direcciones de movimiento).



Figura 7. Seguimiento y estimación de trayectorias.

El seguimiento de objetos a través de imágenes consecutivas se puede aplicar para un conjunto de obstáculos en escena.

La trayectoria de los objetos se pueden considerar como un grupo de puntos desplazándose (en el espacio y el tiempo) simultáneamente. La forma de abordar el problema para la estimación de trayectoria tiene que ver con el seguimiento por detección. Para poder calcular con un umbral de validez la estimación de la siguiente posición de los obstáculos en el tráfico vehicular; en primera instancia se puede abordar como un problema de estimación de probabilidad.

Con base en la información previa (posición y ruta seguida dado por el bloque de seguimiento) se puede establecer una hipótesis de trayectoria. Una opción por medio de la cual se puede parametrizar la hipótesis de trayectoria son las redes bayesianas o un modelo matemático basado en la distribución de probabilidad de las opciones de movimiento (seguir de frente, girar a la derecha, girar izquierda, etc.) que tienen los vehículos.

Conclusiones

Los métodos abordados en este trabajo muestran las diferentes tareas que se tienen que resolver para la detección de obstáculos y la estimación de trayectoria. Un problema significativo es la predicción del desplazamiento conforme a la dinámica individual de cada obstáculo en escena. Usar VEC para plantear una hipótesis sobre la ubicación a futuro de un objeto en escena es una tarea compleja, de tal forma que completar la tarea con

métodos de flujo óptico y seguimiento con modelos de probabilidad aún está en desarrollo.*

REFERENCIAS

1. Khoury, J., Amine, K. y Abi, R. (2019). An Initial Investigation of the Effects of a Fully Automated Vehicle Fleet on Geometric Design. En *Journal of Advanced Transportation*, 150-175.
2. Szeliski, R. (2010). Computer vision: algorithms and applications. *Springer Science and Business Media*.
3. Sucar, E. y Gómez, G. (2017). Visión computacional. Instituto Nacional de Astrofísica, Óptica y Electrónica. Recuperado el 12 de Noviembre de 2018, de <https://ccc.inaoep.mx/esucar/Libros/vision-sucar-gomez.pdf>.
4. Tian, Y., Pei, K., Jana, S. y Ray, B. (2018). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. En *Proceedings of the 40th international conference on software engineering*, 303-314.
5. Song, W., et al. (2018). Real-time obstacles detection and status classification for collision warning in a vehicle active safety system. En *IEEE Transactions on Intelligent Transportation Systems*, 758-773.
6. Prabhakar, G., et al. (2017). Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. En *IEEE region 10 symposium (tensymp)*, 1-6.
7. Lin, F., Chan, Y., Fu, C., Hsiao, Y., Chuang, A., Huang, S. y Lo, F. Integrating appearance and edge features for sedan vehicle detection in the blind-spot area. En *IEEE Transactions on Intelligent Transportation Systems*, 737-747.
8. Hussain, R., y Zeadally, S. (2018). Autonomous Cars: Research Results, Issues, and Future Challenges. *IEEE Communications Surveys and Tutorials*, 21(2), 1275-1313.

SOBRE LOS AUTORES

Lauro Reyes Cocoltzi. Estudiante de doctorado en la Facultad de Ciencias de la Computación en la Benemérita Universidad Autónoma de Puebla (Puebla, México). Sus intereses de investigación incluyen procesamiento de imágenes, visión artificial y sistemas de navegación autónomos.



Ivan Olmos Pineda. Profesor de tiempo completo / investigador en la Facultad de Ciencias de la Computación en la Benemérita Universidad Autónoma de Puebla (Puebla, México). Sus intereses de investigación incluyen: aprendizaje máquina, reconocimiento de patrones, inteligencia artificial, árboles de decisión, entre otras.



José Arturo Olvera López. Profesor de tiempo completo / investigador en la Facultad de Ciencias de la Computación en la Benemérita Universidad Autónoma de Puebla (Puebla, México). Sus intereses de investigación incluyen: reconocimiento de patrones, minería de datos, aprendizaje máquina, análisis digital de imágenes, biometría, entre otras.

*Vehículos autónomos*

— Imagen tomada de <https://www.autoevolution.com/news/daimler-details-autonomous-car-testing-in-california-from-2019-129462.html#>.—

ARTÍCULO ACEPTADO

Construyendo robots seguidores de línea avanzados

Leonardo Romero, Moisés García, Antonio Camarena-Ibarrola e Ignacio Juárez-Campos

Construir robots seguidores de línea es, quizá, la primera tarea en la que se involucra un estudiante de robótica. Sin embargo, para un diseñador avanzado, la meta es lograr que el robot viaje tan rápido como sea posible. Este artículo aborda los desafíos inherentes para estos robots avanzados: usar sensores para medir los desplazamientos angulares en los motores que impulsan las ruedas del robot, de manera que sea posible construir un mapa del camino. Una vez construido el mapa del camino, la idea es aprovecharlo para diseñar estrategias de velocidad que le permitan al robot viajar tan rápido como sea posible, sin salirse del camino. Este ambiente simplificado proporciona una oportunidad para que los estudiantes se adentren en varias cuestiones relevantes de la robótica móvil, tales como: cinemática del robot, incertidumbre en los sensores, construcción de mapas usando segmentos rectos y circulares, y el problema de localización del robot. Los autores esperan que los lectores puedan beneficiarse de este artículo y motivarse a construir robots seguidores de línea más avanzados que los actuales, logrando mejores desempeños en competencias regionales, nacionales e internacionales.

Introducción

La Figura 1a muestra una competencia de robots seguidores de línea de persecución, donde dos robots son colocados en los lados opuestos del camino o pista. Los robots deben seguir la línea negra tan rápido como sea posible y el robot que alcanza al otro gana la competencia.

El robot mostrado en la Figura 1b fue construido por un estudiante de ingeniería en computación e incorpo-

ra: una batería LiPo de 7.4V (350mAh), un micromotor acoplado a cada rueda, 8 sensores reflexivos ubicados al frente del robot (que le permiten detectar la pista), y un microcontrolador Teensy 3.2 de 32 bits (que opera como la computadora a bordo del robot). Considerando la máxima velocidad de los motores de 3000rpm y el diámetro de las ruedas de 2cm, el robot puede alcanzar velocidades cercanas a los $3 \frac{m}{s}$.

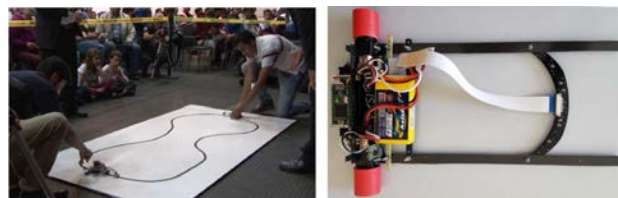


Figura 1.a) Competencia de dos robots seguidores de línea y b) Vista superior de un robot.

Usualmente los robots seguidores de línea reaccionan solo a las mediciones de los sensores reflectivos para seguir la pista sin salirse y no construyen ningún tipo de mapa de la trayectoria que siguen. La observación clave en este tipo de competencias es que es muy común que los robots viajen por los mismos lugares en varias ocasiones antes de alcanzar al otro robot. De esta manera, tiene sentido construir un mapa del camino y usarlo para acelerar en segmentos rectos del camino y frenar antes de entrar a las curvas, como lo hacen los corredores profesionales de autos de carreras.

La idea: construir un mapa del camino y usarlo para acelerar en segmentos rectos de la pista y frenar antes de entrar a las curvas

Para conocer cuanto ha girado cada rueda del robot, lo que nos permitiría estar en condiciones de poder construir un mapa del camino, se agregó un pequeño *encoder* magnético a cada micromotor. Un *encoder* es un sensor que permite medir los desplazamientos angulares (el *encoder* magnético utilizado tiene una precisión de $\frac{1}{12}$ de vuelta del motor).

Enseguida se describe brevemente la etapa de construcción del mapa, la determinación de los segmentos de línea recta y las curvas del camino, la localización del robot en el mapa y la estrategia de aceleración/frenado del robot.

Construyendo el mapa

La Figura 2 ayuda a comprender la trayectoria que sigue un robot de este tipo, llamado robot de locomoción diferencial, a partir de la velocidad de la rueda izquierda (V_l) y de la rueda derecha (V_r) del robot. Si asumimos que la velocidad de la rueda izquierda es un poco menor que la velocidad de la rueda derecha, se genera un movimiento de rotación de radio R con centro de rotación C , como se ilustra en la Figura 2. Un caso especial es cuando ambas velocidades son iguales, originando un movimiento hacia adelante o atrás, siguiendo una línea

recta. Otro caso especial es cuando ambas velocidades son iguales en magnitud, pero de signo contrario (una rueda gira en un sentido y la otra en el otro sentido), ocasionando que el robot gire sobre su centro.

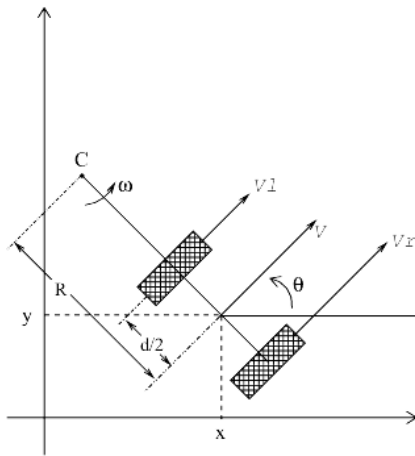


Figura 2. Rotación del robot respecto al punto C.

Al modelo matemático para calcular la trayectoria del robot (dadas las velocidades de las ruedas), se le conoce como cinemática directa. El lector puede consultar en [2,3,9] las ecuaciones que definen la cinemática directa de un robot de este tipo.

La Figura 3 muestra un ejemplo de una pista donde se agregó temporalmente una pequeña marca negra (en la parte inferior) para que el robot pueda identificar cuando culmina una vuelta completa. Se puede apreciar en la Figura 4 el mapa construido a partir de las velocidades de las ruedas sensadas por los encoders de los motores y las ecuaciones de la cinemática directa para este tipo de robots. El recorrido inicia en la parte inferior y avanza hacia la derecha. Se observa que la trayectoria estimada del robot se va alejando poco a poco de la trayectoria real debido a la incertidumbre asociada a los sensores del robot, de manera que el punto final del recorrido no coincide con el punto de partida.

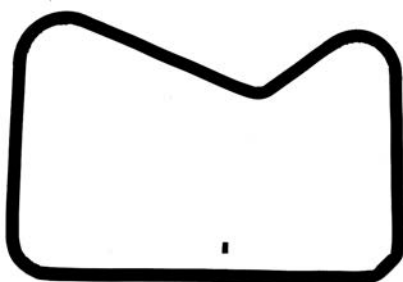


Figura 3. Pista real.

Una causa de la incertidumbre en las mediciones de los sensores se debe justamente a la precisión con la que

se realiza la medición (en múltiplos de $\frac{1}{12}$ de vuelta para nuestro robot). A mayor precisión, se tendrá menor incertidumbre.

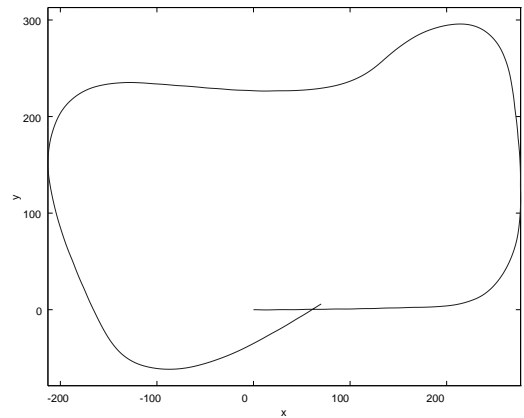


Figura 4. Mapa Construido.

Si construimos el mapa a partir de la posición final retrocediendo hasta el principio; esto es, a partir de la posición final consideramos los datos de las velocidades en orden inverso, podemos construir el mapa que se muestra en la Figura 5. Ahora bien, si consideramos las dos posiciones que resultan para los datos de cierta medición de velocidades de las ruedas, tanto hacia adelante como hacia atrás, tenemos una situación como se ilustra en la Figura 6. El radio de los círculos representa la incertidumbre asociada con la ubicación de los centros; a mayor distancia recorrida, mayor la incertidumbre sobre la posición del robot.

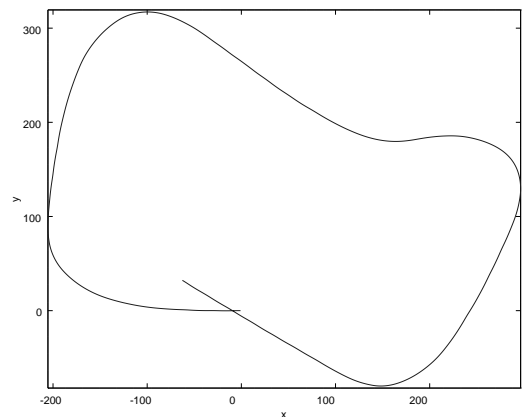


Figura 5. Mapa construido hacia atrás.

Si asumimos que la incertidumbre se modela adecuadamente con distribuciones de probabilidad normales con covarianza cero (es decir, con la forma de una campana), entonces la posición real del robot estará en un punto de

la línea que une los centros de los círculos (la línea punteada de la Figura 6). Sea $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$ los centros de los círculos. Las ecuaciones paramétricas de la línea que conecta a los puntos P_1 y P_2 están dadas por: $x = x_1 + (x_2 - x_1)t$, $y = y_1 + (y_2 - y_1)t$ donde t es un parámetro de la línea. Cuando $t = 0$, las coordenadas (x, y) corresponden al punto P_1 y cuando $t = 1$, las coordenadas (x, y) corresponden al punto P_2 . De manera que el valor de t (entre 0 y 1) corresponde a un punto sobre la línea que une a P_1 y P_2 . Se puede demostrar que si la varianza de las mediciones se hace proporcional a la longitud de la trayectoria, el valor óptimo del parámetro t está dado por: $t = L_1 / (L_1 + L_2)$ donde L_1 y L_2 corresponden a las longitudes de las trayectorias asociadas a los puntos P_1 y P_2 respectivamente. Con esta forma de combinar la información hacia adelante y hacia atrás, se obtiene el mapa mostrado en la Figura 7, el cual se acerca bastante a la forma del ambiente real.

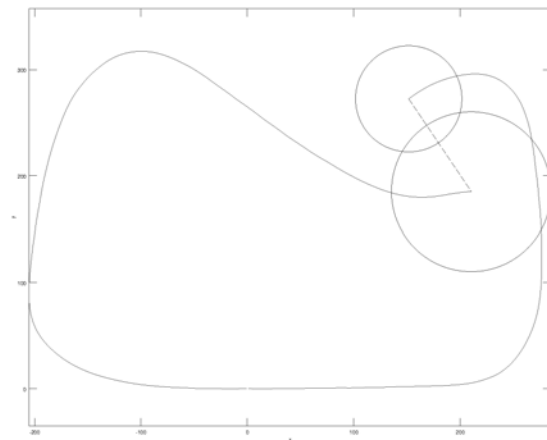


Figura 6. Dos posibles ubicaciones generada por la incertidumbre de las mediciones en los sensores.

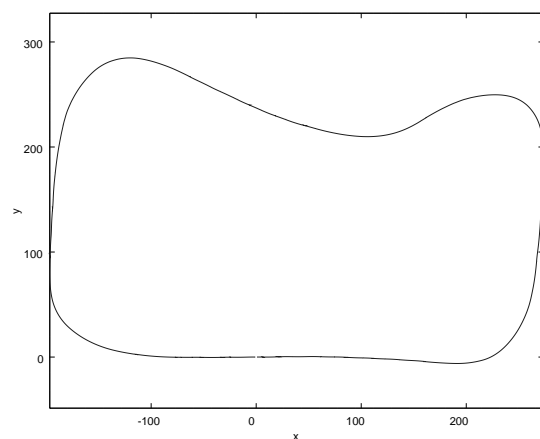


Figura 7. Mapa construido.

Determinación de los segmentos rectos y curvos del mapa

Una vez que tenemos el mapa construido, la siguiente etapa es encontrar los segmentos rectos y curvos del mapa. Primero abordemos el problema de encontrar una línea recta dada una secuencia de puntos que tienen la misma incertidumbre, como se ilustra en la Figura 8. La mejor línea es la que minimiza la suma de los cuadrados de las distancias de los puntos a la línea (conocidas como distancias ortogonales a la línea, por formar ángulos de 90 grados). A este método se le conoce como Mínimos Cuadrados Totales (MCT) [4,5,6,10].

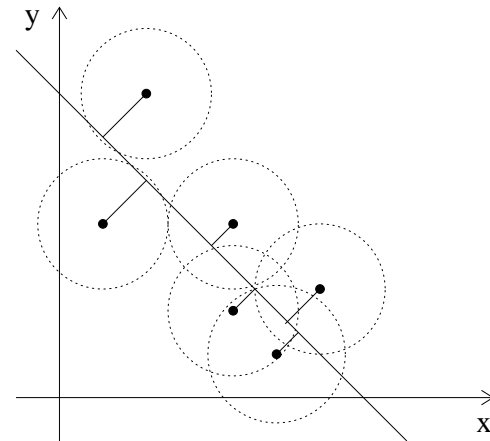


Figura 8. Ajustando una línea considerando distancias ortogonales de los puntos a la línea.

Si podemos calcular una línea recta a partir de una secuencia de puntos, el siguiente problema consiste en encontrar los mejores segmentos de línea recta asociados a una secuencia grande de puntos. El algoritmo de Seguimiento de Línea Extendido (SLE) [8] aborda este problema y es una versión extendida del algoritmo original de Seguimiento de Líneas (SL) [1]. Utilizando SLE, encontramos las líneas mostradas en la Figura 9.

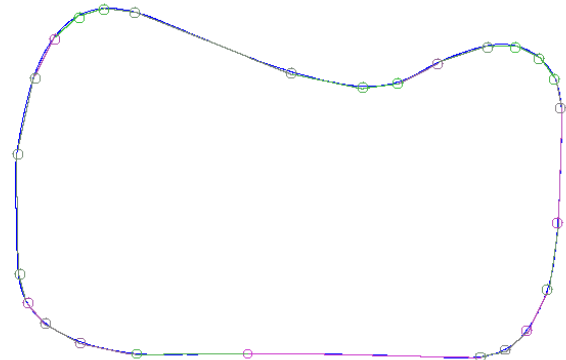


Figura 9. Segmentos de línea encontrados en el mapa construido.

El algoritmo SL es simple y rápido. Al inicio une con una línea los dos primeros puntos de la secuencia. Enseguida considera el siguiente punto de la secuencia. Si la distancia de todos los puntos a la nueva línea es menor a cierto umbral definido, se asocia el punto a la línea y se continúa con el siguiente punto. En caso contrario, el nuevo punto se considera que no pertenece a la línea y se inicia una nueva línea con dicho punto. Sin embargo, cuando el algoritmo SL agrega puntos a la línea, es posible que en realidad los últimos puntos de una línea pertenezcan a la siguiente línea. El algoritmo SLE agrega este paso de verificación, para corregir estos errores en las asociaciones de los puntos a las líneas.

Ahora bien, nos queda el problema de determinar cuales rectas en realidad son mejor explicadas por segmentos de círculo de determinado radio. Para ello, se utiliza el método de Pratt [7] que permite encontrar el círculo que mejor representa a una secuencia de puntos.

Para elegir el segmento de recta o de círculo que mejor explique una secuencia de puntos, se utiliza como criterio de decisión aquel que minimice la suma de las distancias al cuadrado de los puntos a la recta o al círculo. Se elige el segmento que tenga la mínima suma. La Figura 10 muestra los segmentos de rectas y de círculos encontrados. Para cada segmento circular se muestra su centro y dos radios que identifican el inicio y fin del segmento. Las partes que no se describen como círculos están representadas por los segmentos de recta correspondientes en la Figura 9.

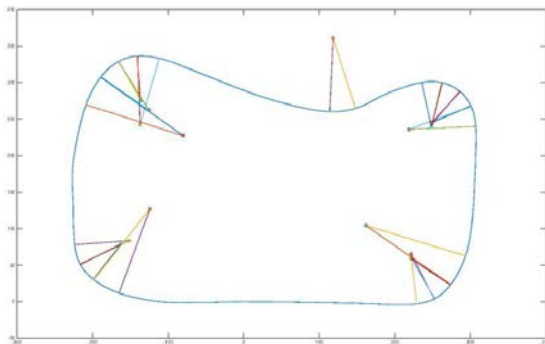


Figura 10. Mapa mostrando los segmentos de recta y de círculos.

El problema de localización del robot

En las competencias, normalmente la pista se encuentra disponible un poco antes, para realizar pruebas. En esta etapa es cuando un robot podría recorrer la pista y construir el mapa. Sin embargo, los jueces pueden cambiar, al inicio de la competencia, la ubicación de los robots. A esta situación, donde el robot desconoce su ubicación en el mapa, se le conoce como el problema de localización global del robot. Un enfoque para resolver este problema, conocido como Localización de Markov [3], consiste

en discretizar el camino en segmentos de igual longitud (por ejemplo de 1 centímetro), de manera que cada segmento de 1cm es un estado y está conectado con el estado siguiente y el estado anterior. Al inicio tenemos que todos los estados tienen la misma probabilidad de que el robot se encuentre ahí. El robot se encuentra localizado cuando solo un estado tiene una alta probabilidad, en comparación con todos los demás estados. Cada vez que el robot avanza un centímetro, se realiza una actualización de la distribución de probabilidad en la secuencia de estados. La Figura 11 ilustra esta actualización. Con probabilidad 0.8 el robot pasa al estado siguiente, con probabilidad 0.1 el robot se queda en el mismo estado y con probabilidad 0.1 el robot pasa dos estados adelante. De esta forma se modela la incertidumbre asociada en el avance real de robot.

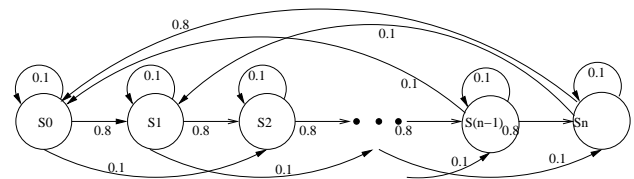


Figura 11. Modelo de actualización de la distribución de probabilidad en los estados por el movimiento de 1 cm del robot.

Por otro lado, utilizando las ecuaciones del modelo de cinemática directa de la locomoción diferencial se puede estimar el radio R_m del centro de rotación que sigue el robot. Con esta medición se diseña una función de la probabilidad $P(R_m|s_i)$ para obtener la medición R_m dado que el robot se encuentra en el estado s_i ; para ambos casos, tanto cuando el estado s_i corresponde a un segmento recto, como cuando corresponde a un segmento circular de determinado radio. En las pruebas realizadas, después de 30cm la secuencia de estados presenta dos estados con alta probabilidad, y después de 60cm el robot se encuentra localizado al tener sólo un estado con una alta probabilidad.

Estrategia de aceleración y frenado

Si se conoce el radio de cada segmento circular, es posible estimar la velocidad máxima que el robot debe alcanzar en los estados asociados a dicha curva. Utilizando la secuencia de estados, se puede calcular la distancia de cada estado a la siguiente curva (con su velocidad máxima permitida) y, por lo tanto, diseñar estrategias de aceleración/frenado para alcanzar altas velocidades en los segmentos rectos y frenar justo para alcanzar la velocidad límite de la siguiente curva.

Desafíos

Se deben abordar algunos desafíos para lograr un robot autónomo más competitivo. Si no es posible colocar una

marca temporal en el camino, que ayude a detectar cuando el robot da una vuelta completa, entonces se requiere tener un modelo muy completo de la incertidumbre asociada con los movimientos del robot. Dicho modelo debe detectar (con alta probabilidad) cuando el robot pasa por el mismo lugar. Este problema es conocido como el problema completo de Localización y Mapeo Simultáneo (en inglés, full SLAM).

La simetría del camino (ver Figura 12), presenta otro aspecto que es necesario considerar. Pueden haber dos, tres, cuatro o más lugares del camino donde el robot pueda estar (marcados por pequeños círculos en el mapa), y no existe forma de determinar exactamente en cual lugar está a partir de las mediciones de los sensores. En ese caso, puede asumirse que el robot se encuentra en cualquiera de dichas posiciones (así el robot se encuentra localizado).

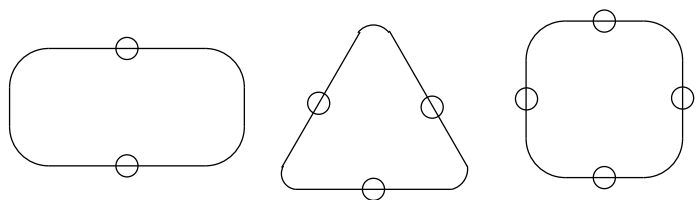


Figura 12. Diferentes tipos de caminos simétricos.

Otra dificultad se presenta cuando el robot se coloca en dirección opuesta a aquella en la cual el robot construyó el mapa. En este caso, se puede pensar en utilizar dos modelos: uno construido a partir del recorrido del robot en cierto sentido y otro modelo construido en la dirección contraria. Cuando uno de los modelos tiene un

estado con una alta probabilidad, el robot está localizado y conoce en que dirección está recorriendo la pista.

Finalmente, un robot podría agregar una pequeña turbina que empuja el aire hacia arriba y por lo tanto incrementa la fuerza de fricción de las ruedas del robot y su velocidad. La potencia desarrollada por la turbina podría estar en función de la ubicación del robot, para disminuirla en los segmentos rectos y aumentarla cuando se aproxima a curvas con pequeños radios.*

REFERENCIAS

1. Borges, G.A. y Aldon, M.-J. (2004). Line extraction in 2d range images for mobile robotics. *Journal of Intelligent and Robotic Systems* 40, 267-297.
2. Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms in Matlab*. 2nd edn. Springer.
3. Dudek, G., Jenkin, M. (2010). *Computational Principles of Mobile Robotics*. 2nd edn. Cambridge University Press.
4. Golub, G.H. (1973). Some modified matrix eigenvalue problems. *Siam Review*, 15(2), 318-334.
5. Nivergelt, Y. (1994). Total least squares: State-of-the-art regression in numerical analysis. *SIAM Rev.*, 36(2), 258-264.
6. Markovskiy, I. y Van Huffel, S. (2007). Overview of total least-squares methods. *Signal processing*, 87(10), 2283-2302.
7. Pratt, V. (1987). Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph.*, 21(4), 145-152.
8. Romero, L., Garcia, M. y Lara, C.A. (2013). An extended line tracking algorithm. En *2013 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, 1-5.
9. Siegwart, R., Nourbakhsh, I.R. y Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. 2nd edn. The MIT Press.
10. Van Huffel, S. y Vandewalle, J. (1991). The Total Least Squares Problem. *Computational Aspects and Analysis*, 9.

SOBRE LOS AUTORES



Leonardo Romero Muñoz estudió la licenciatura en ingeniería eléctrica en la Universidad Michoacana de San Nicolás de Hidalgo y la maestría y doctorado en computación en el Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Cuernavaca. Se desempeñó como investigador en el Instituto de Investigaciones Eléctricas en Cuernavaca, Morelos de 1986 a 1993. A partir de 1993 es profesor investigador en la Universidad Michoacana. Ha desarrollado trabajos en las áreas de visión computacional, robótica móvil y razonamiento probabilístico. Es miembro del Sistema Nacional de Investigadores.



Moisés García Villanueva obtuvo el grado de Maestro en Ingeniería Eléctrica en la Universidad Michoacana de San Nicolás de Hidalgo en el estado de Michoacán de Ocampo, México en el año de 2001. Actualmente es profesor investigador en la Facultad de Ingeniería Eléctrica de la Universidad Michoacana. Sus intereses científicos incluyen la robótica, procesamiento de lenguaje natural, minería de datos e Inteligencia Artificial.



Antonio Camarena Ibarrola nació en Morelia Michoacán, obtuvo el grado de Ingeniero Eléctrico en la Universidad Michoacana de San Nicolás de Hidalgo, el grado en Maestro en Ciencias Computacionales en el Instituto Tecnológico de Toluca y el Grado de Doctor en Ciencias en Ingeniería Eléctrica, opción Sistemas Computacionales en la Universidad Michoacana, trabaja como profesor-investigador en la División de Estudios de Postgrado de la Facultad de Ingeniería Eléctrica de la Universidad Michoacana, sus principales intereses como investigador están en el área de procesamiento digital de señales, reconocimiento de patrones y búsquedas en espacios métricos.



Ignacio Juárez Campos nació el 23 de mayo de 1966, en Zitácuaro, Michoacán. Estudió Ingeniería Mecánica en la UMSNH, 1988, maestría y doctorado en sistemas mecánicos y mecatrónica, 1996 y 2002, respectivamente, en la Facultad de Ingeniería de la UNAM. Actualmente, es profesor e investigador en la Facultad de Ingeniería Mecánica de la UMSNH. Sus intereses profesionales incluyen la concepción y el desarrollo de máquinas de locomoción discreta y exoesqueletos para asistencia y rehabilitación. Es integrante del Sistema Nacional de Investigadores.



Competencias en robótica 2019, robot seguidor de líneas

— Imagen tomada de <http://roboticday.org/2019/competitions/>.—

IA & Educación

Julieta Noguez, Lucía Barrón y Yasmín Hernández
iaeducacion@komputersapiens.org

Inteligencia Artificial Cognitiva

La Inteligencia Artificial Cognitiva (CAI, por sus siglas en inglés) es una nueva perspectiva en Inteligencia Artificial, que asocia un nuevo concepto de inteligencia el cual añade a la emulación del comportamiento, la forma en que el cerebro aumenta su conocimiento [1].

Una forma de inteligencia humana es la capacidad del cerebro humano para generar nuevo conocimiento a partir de la información percibida y entregada por los órganos sensoriales humanos. Y es el único que posee la capacidad para generar nuevo conocimiento. Este mecanismo ha sido investigado por muchos científicos en todo el mundo debido al interés de los seres humanos de crear una máquina inteligente capaz de ayudarlos en su vida laboral diaria. Para ello se han introducido e implementado muchos enfoques para garantizar que su rendimiento pueda imitar lo inteligente del cerebro humano.

Por otro lado, también se ha desarrollado tecnología llamada fusión de información, para predecir o estimar el futuro de un fenómeno. Esta tecnología trata de imitar el mecanismo del cerebro humano para obtener información completa como predicción o estimación del estado de un fenómeno a través de los órganos de los sentidos [2]. Ver figura 1. Es decir, el ser humano obtiene mucha información de sus órganos de detección que comprenden ojos, oídos, nariz, piel y lengua. Según esta observación, la clave principal de la generación de conocimiento en el cerebro humano es la fusión de información. Sumari y Ahmad [2] llaman a este mecanismo crecimiento del conocimiento, es decir, conocimiento que el cerebro desarrolla después de procesar la información entregada por los órganos sensoriales del sistema.

Estos conceptos se están ampliando rápidamente con el desarrollo de sensores habilitados para la red y algoritmos de Inteligencia Artificial. Varios autores proponen sistemas inteligentes centrados en el ser humano para proporcionar servicios de mayor calidad, como atención médica inteligente, interacción afectiva y conducción autónoma.

Teniendo en cuenta que la computación cognitiva es una tecnología indispensable para desarrollar estos sistemas inteligentes, algunos autores como Chen, Herrera y Hwang [3] proponen la computación centrada en el ser humano asistida por la computación cognitiva y la computación en la nube. Estos autores desarrollaron la arquitectura de un sistema de computación cognitiva, que consta de tres tecnologías: redes (Internet de las cosas), análisis (aprendizaje por refuerzo y aprendizaje

profundo) y computación en la nube. Finalmente, describen las aplicaciones representativas de la computación cognitiva centrada en el ser humano, incluida una tecnología de robots, un sistema de comunicación emocional y un sistema cognitivo médico.

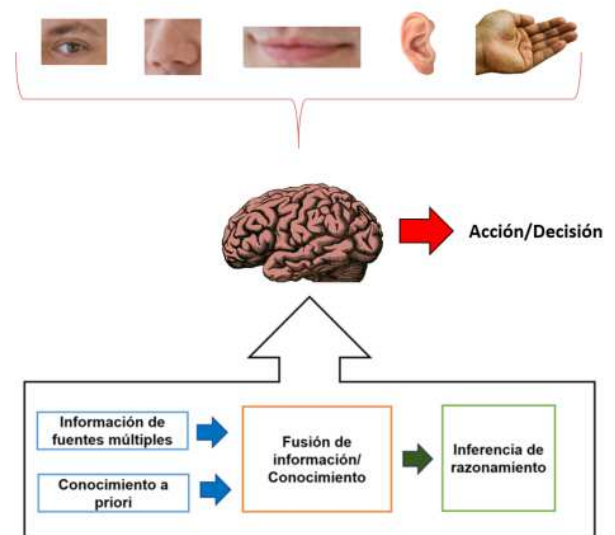


Figura 1. Sistema de Fusión de Información Humana. Basado en [2].

El panorama de este nuevo enfoque es promisorio y pueden verse otro tipo de aplicaciones científicas. Por ejemplo, algunos otros autores lo han aplicado con éxito a problemas de seguridad de hardware [4], y para diagnóstico de bloqueo cardíaco y arritmia [5]. También ha habido esfuerzo en desarrollar aplicaciones comerciales como Cortana (Microsoft), Siri (Apple) y Watson (IBM), quienes, gracias al uso de amplias bases de datos y de algoritmos de cómputo cognitivo, tienen la capacidad de comunicarse en lenguaje natural, facilitando la comunicación con la tecnología en un lenguaje humano.*

REFERENCIAS

1. Ahmad, A.S., y Sumari, A.D.W. (2017). Cognitive Artificial Intelligence: Brain-Inspired Intelligent Computation in Artificial Intelligence. *Proceedings of Computing Conference 2017*.
2. Sumari, A.D.W., y Ahmad, A.S. (2016). Cognitive Artificial Intelligence. The Fusion of Artificial Intelligence and Information Fusion. *Proceedings of International Symposium on Electronics and Smart Devices (ISESD)*.

3. Chen, M., Herrera, F., y Hwang, K. (2018). Cognitive Computing: Architecture, Technologies and Intelligent Applications. *Special Section on Human-Centred Smart Systems and Technologies*. IEEE Access. 6.
4. Putra, S.D., Ahmad, A.S., y Sutikno, S.(2016). DPA-countermeasure with knowledge growing system, En *Proceedings of 2016 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 16-20.
5. Sereati, C.O., Sumari, A.D.W., Adiono, T., y Ahmad, A.S. (2017). Cognitive Artificial Intelligence (CAI) based on Knowledge Growing Systems (KGS) for Diagnosing Heart Block and Arrhythmia. *Proceedings of 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*.

Publique en Komputer Sapiens



Deskubriendo Konocimiento

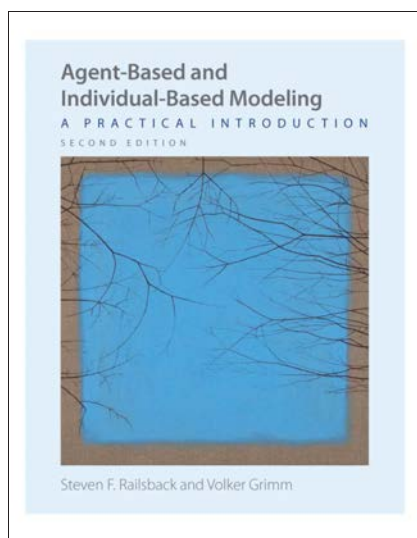
Alejandro Guerra Hernández y Leonardo Garrido
deskubriendokonocimiento@komputersapiens.org

Reseña de Agent-Based and Individual-Based Modeling: A Practical Introduction *

Dr. Alejandro Guerra-Hernández

Centro de Investigación en Inteligencia Artificial

Universidad Veracruzana



Portada de la revista.*

El modelado y simulación basado en agentes, ABMS por sus siglas en inglés, se ha constituido en una alternativa atractiva a los métodos tradicionales de simulación, particularmente en problemas caracterizados como complejos. Su atractivo reside en que su naturaleza ascendente, evita la adopción temprana de consideraciones sobre cómo mantener nuestros modelos matemáticamente tratables. Por ascendente quiero decir que, en lugar de partir de los principios generales de cómo debe comportarse el sistema globalmente, expresados muchas veces como un sistema de ecuaciones, el ABMS parte de la especi-

ficación del comportamiento de los diferentes agentes que participan en la simulación y sus interacciones. El comportamiento global del sistema es una propiedad emergente de estas interacciones y si las cosas van bien, será cercano a lo observado en la realidad.

Pues bien, el libro de Steven F. Railsback y Volker Grimm, cuya segunda edición hoy reseñamos, tiene como objetivo explicar cuándo y por qué usar el ABMS y cómo llevar las cosas a buen término. Los autores ponen especial atención en cómo diseñar un ABMS y para ello introducen detalladamente los fundamentos conceptuales del diseño y descripción de modelos, asuntos en los que tienen muchos años de experiencia. La parte práctica del libro está soportada por el uso de NetLogo, un lenguaje de programación dedicado al ABMS, bien documentado y fácil de aprender.

Para conseguir sus objetivos, los autores han organizado el libro en cuatro partes. La primera de ellas introduce los conceptos básicos del ABMS y NetLogo. El capítulo inicial ilustra el uso del ABMS, en comparación con una aproximación más tradicional, para gestionar el control de la rabia en Europa. Una vez ejemplificado el uso del ABMS, se define el concepto de modelo y se

introduce informalmente una metodología de diseño en torno al ciclo de modelado, que será de gran importancia a lo largo del todo el texto. También se resaltan aquí las diferencias entre el ABMS y otras técnicas de modelado y simulación. El segundo capítulo es básicamente un tutorial de NetLogo, que hace uso extensivo de la excelente documentación que se incluye en la distribución de este lenguaje de programación. El tercer capítulo introduce una herramienta de gran utilidad: el protocolo ODD (Overview, Design Concepts, and Details), una especificación de los ABMS que se ha vuelto un estándar de facto. Con estos elementos cubiertos, el capítulo 4 propone la implementación de un primer ABMS –Un caso bien documentado y abordable: la formación de corredores en las migraciones de mariposas. El capítulo 5, bajo el sugerente título "De la animación a la ciencia", introduce el tipo de análisis que se pueden realizar con los datos generados por nuestro primer modelo. Finalmente, el capítulo 6 aborda las técnicas de depuración de NetLogo y documenta los errores más comunes al iniciarse en el uso de este lenguaje de programación.

La segunda parte del libro tiene como objetivo presentar al ABMS

* Railsback, Steven F. and Grimm, Voler. Agent-Based and Individual-Based Modeling: A Practical Introduction. 2nd ed. Princeton University Press, USA, 2019.

como una forma específica de hacer ciencia, abordando más en detalle el marco conceptual que caracteriza a este tipo de modelado y simulación. Para ello, el capítulo 8 revisa el concepto de emergencia, analizando los resultados de la simulación para entender cuales son relativamente impuestos o predecibles de alguna manera; y cuales emergen realmente de forma compleja de las conductas de los agentes. Se introduce el uso de la herramienta conocida como espacio de conductas, para explorar el espacio de parámetros con NetLogo con este objetivo. Los capítulos 9 y 10 están dedicados respectivamente a la observación y la percepción en estos sistemas, es decir, qué salidas pueden generar, por ejemplo, animaciones, archivos, gráficas, etc., y cómo obtienen información los agentes que componen un ABMS. Los capítulos 11 al 13 se concentran en la toma de decisión de los agentes, en términos del comportamiento adaptativo de éstos, sus capacidades de predicción y las interacciones que se dan entre ellos. El capítulo 14 revisa cómo se implementa el ciclo principal de la simulación y la naturaleza del tiempo en NetLogo. El capítulo 15 cubre los aspectos estocásticos del ABMS, en particular cuando es que un proceso debe modelarse como aleatorio. Finalmente, el capítulo 16 aborda el concepto de los colectivos, formas de organización que agrupan agentes y afectan los comportamientos individuales. De especial interés es la emergencia de estos colectivos a lo largo de la simulación.

La tercer parte del libro tiene que ver con las técnicas de valida-

ción de los ABMS. Se trata de una profundización en los temas introducidos en el capítulo 6. Aunque el propósito de nuestras simulaciones funciona como un filtro para decidir que elementos de la realidad deben incluirse en el modelo, este criterio es insuficiente para saber si el modelo es estructuralmente realista, es decir, que aún siendo una simplificación de la realidad, captura la organización interna del fenómeno observado. Los autores sugieren un modelado orientado a patrones para ello, de forma que nuestros modelos finales sean generales, útiles, científicos y precisos. El capítulo 18 introduce el concepto de patrón en la estructura de un modelo, mientras que el 19 aborda la teoría de desarrollo, una metodología para explorar los patrones del modelo en el diseño de la conducta de los agentes. Finalmente el capítulo 20 aborda la parametrización y calibración de los ABMS, profundizando en el uso de la herramienta de espacio de conductas.

La cuarta, y última parte del libro, se concentra en el análisis de nuestros modelos, una profundización a los temas del capítulo 5. Por análisis entendemos el proceso de entender qué es lo que un modelo, o sub-modelo, hace -¿Porqué se comporta de ciertas maneras, bajo ciertas circunstancias? El capítulo 22 describe una serie de estrategias generales para el análisis de los ABMS, enfatizando la necesidad de ejecutar nuestros experimentos de manera controlada, guiados por nuestras hipótesis. El capítulo 23 se aboca al análisis, más tradicional, de sensibilidad, incertidumbre y robustez; por ejemplo, el efecto de los

valores de los parámetros en nuestras salidas. El capítulo 24 termina el libro con algunas recomendaciones sobre cómo proseguir con la práctica de los ABMS una vez terminada la lectura del mismo.

El sitio web (<http://www.railsback-grimm-abm-book.com>) del libro incluye ligas de interés como material para clases, errata, información sobre los autores, etc., y las novedades y actividades en torno a éste, como cursos, conferencias, etc. En mi opinión el libro puede usarse en un curso de un semestre sobre ABMS; o enfatizar en las partes I y II para algo más corto, por ejemplo mi curso de ABMS y Aprendizaje en la Universidad Veracruzana (<https://www.uv.mx/personal/aguerra/abms/>). Se trata además de un excelente libro para los autodidactas que cubre los aspectos conceptuales y prácticos del ABMS de manera auto-contenida. Los ejemplos utilizados reflejan una cuidadosa selección en diversos dominios de conocimiento, lo cual contribuye enormemente a la comprensión de los alcances y posibilidades de las técnicas presentadas. Las mejoras con respecto a la primera edición, incluyen la actualización del código a una versión más reciente de NetLogo (6.0.4) y una revisión de los ejercicios y sus soluciones disponibles para profesores. Por lo anterior, no me resta más que recomendar ampliamente la lectura cuidadosa y activa de este texto para introducirse a la practica del modelado y simulación basado en agentes.*



XI Congreso Mexicano de Inteligencia Artificial COMIA 2019

INFORMACIÓN GENERAL

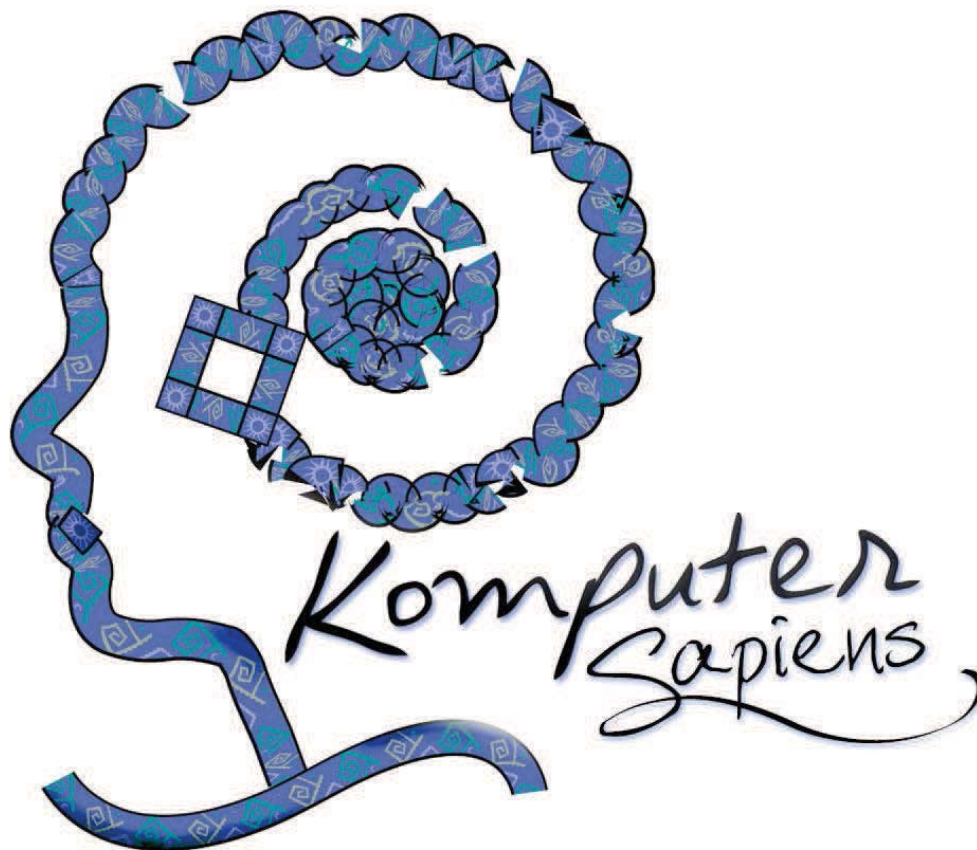
COMIA 2019 es organizado por la Sociedad Mexicana de Inteligencia Artificial (SMIA) con sede en el Centro de Investigación Científica y de Educación Superior de Ensenada - Unidad de Transferencia Tecnológica Tepic CICESE-UT3 y la Universidad Autónoma de Nayarit UAN. COMIA 2019 es un foro científico serio para la presentación y publicación de trabajos de investigación derivados de tesis o proyectos, terminados o en proceso, en español.

FECHAS IMPORTANTES

Fecha límite para envío de artículos:	16 de abril de 2019
Notificación de aceptación:	30 de abril de 2019
Envío de versión final:	11 de mayo de 2019
Límite para pago de artículos aceptados:	14 de mayo de 2019
Celebración del Congreso:	4 al 7 de junio de 2019

LLAMADA A ARTÍCULOS

<http://smia.mx/comia2019/>



¡Publique en Komputer Sapiens!



Komputer Sapiens solicita artículos de divulgación en todos los temas de Inteligencia Artificial, dirigidos a un amplio público conformado por estudiantes, académicos, empresarios, tomadores de decisiones y consultores. Komputer Sapiens es patrocinada por la SMIA, la Sociedad Mexicana de Inteligencia Artificial



www.smia.org.mx

Instrucciones para autores e información general: <http://www.komputersapiens.org>
Síguenos en las redes sociales: www.facebook.com/Komputer.Sapiens, twitter.com/KomputerSapiens

