

Desarrollo de un traductor Hñähñu-Español mediante un algoritmo estadístico para redes neuronales y palabras vectorizadas

Development of a hñähñu-spanish translator using a statistical algorithm for neural networks and vectorized words

José Manuel Cruz Olguín¹, Salvador Santos Romero¹, Virgilio López Morales¹, Manuel Alejandro Ojeda Misses^{1*}

¹ Instituto de Ciencias Básicas e Ingeniería, Área Académica de Computación y Electrónica, Universidad Autónoma del Estado de Hidalgo, Mineral de la Reforma, Hidalgo, México. CP. 42090. cr233109@uaeh.edu.mx, sa513025@uaeh.edu.mx, virgilio@uaeh.edu.mx, manuel_ojeda@uaeh.edu.mx

*Autor de correspondencia

Resumen

Palabras clave:

Algoritmo;
estadística; red
neural; Hñähñu;
traducción.

En este trabajo se presenta un algoritmo de aprendizaje estadístico-probabilístico (EyP) para el entrenamiento de redes neuronales aplicado a la traducción léxica Hñähñu-Español. El método emplea la normalización estadística del error y la función de distribución acumulada normal para generar parámetros adaptativos en la actualización de los pesos sinápticos y el sesgo. La validación se implementó mediante un corpus bilingüe compuesto por palabras representadas mediante vectores numéricos de longitud fija. El desempeño del algoritmo fue evaluado mediante el error cuadrático medio y comparado con los algoritmos Backpropagation y Levenberg-Marquardt. Los resultados obtenidos muestran que el algoritmo EyP es capaz de aprender asociaciones léxicas entre ambos idiomas. Asimismo, esto sugiere que la metodología propuesta constituye una alternativa viable para escenarios con disponibilidad limitada de datos y evidencian su potencial para el desarrollo de herramientas computacionales orientadas a la preservación y revitalización digital de las lenguas indígenas.

Abstract

Keywords:

Algorithm; neural
networks; Hñähñu;
translation.

This work presents a statistical-probabilistic (EyP) learning algorithm for training artificial neural networks, applied to Hñähñu-Spanish lexical translation. The proposed method uses statistical normalization of the prediction error and the normal cumulative distribution function to generate adaptive parameters for updating the synaptic weights and bias of the neural network. For validation, a neural network was trained using a bilingual corpus composed of words represented by fixed-length numerical vectors. The performance of the proposed algorithm was evaluated using the mean squared error and compared with the Backpropagation and Levenberg-Marquardt algorithms. The results show that the EyP algorithm is capable of learning lexical associations between the two languages while maintaining stable behavior during the training process. Furthermore, the findings suggest that the proposed methodology represents a viable alternative for scenarios with limited data availability and demonstrate its potential for the development of computational tools aimed at the preservation and digital revitalization of indigenous languages.

Recibido: 22 de agosto de 2025

Aceptado: 03 de julio de 2026

Publicado: 30 de septiembre de 2026

Cómo citar: Cruz, J. M.; Santos, S.; López, V.; & Ojeda, M. A. (2026). Desarrollo de un traductor Hñähñu-Español mediante un algoritmo estadístico para redes neuronales y palabras vectorizadas. *Acta Universitaria*, 36, e4778. doi: <https://doi.org/10.15174/au.2026.4778>

Introducción

México se caracteriza por una amplia diversidad lingüística y cultural reflejada en la presencia de numerosas lenguas indígenas distribuidas en diferentes regiones del país. De acuerdo con el Instituto Nacional de Estadística y Geografía (INEGI, 2024), los estados con mayor proporción de población hablante de lengua indígena son Oaxaca (27.3%), Yucatán (26.1%), Chiapas (23.4%), Quintana Roo (14.1%) y Guerrero (13.9%). Estas lenguas constituyen un elemento fundamental del patrimonio cultural mexicano, ya que representan no solo medios de comunicación, sino también formas de preservar conocimientos, tradiciones e identidad colectiva. Entre las lenguas indígenas con mayor número de hablantes destacan el náhuatl (23.6%), el maya (12.4%), el tzeltal (7.9%) y el zapoteco (7.2%) (INEGI, 2024).

De acuerdo con el Instituto Nacional de Estadística y Geografía (INEGI, 2024), la población hablante de lenguas indígenas mantiene una estrecha relación con su territorio de origen, ya que el 86% reside en su estado natal, mientras que únicamente el 13.7% ha migrado a otra entidad federativa, y el 0.3% nació en el extranjero. Asimismo, la movilidad reciente es reducida, pues el 96.4% de los hombres y el 97.6% de las mujeres no han cambiado de residencia durante los últimos cinco años. Estos indicadores reflejan el fuerte vínculo existente entre lengua, identidad cultural y territorio, aspecto que ha impulsado el desarrollo de políticas orientadas a la preservación, promoción y fortalecimiento de las lenguas originarias en ámbitos educativos, culturales y sociales.

Entre estas lenguas originarias se encuentra el otomí, denominado Hñähñu por sus hablantes, perteneciente a la familia lingüística otomangue y al subgrupo otopame (Lastra, 2010). Su presencia se concentra principalmente en los estados de Hidalgo, Querétaro y Estado de México. Según datos del INEGI (2015), existen 307 928 hablantes de esta lengua, de los cuales aproximadamente el 39.4% reside en el estado de Hidalgo, particularmente en la región del Valle del Mezquital. El uso del término Hñähñu constituye además una forma de autoidentificación cultural que fortalece el sentido de pertenencia de sus comunidades, mientras que la denominación otomí corresponde al término comúnmente utilizado en español.

Diversos esfuerzos se han desarrollado para preservar esta lengua. En su estudio, Aguilar & García (2023) documentan iniciativas que desde el año 2000 han sido orientadas a fortalecer las lenguas originarias (Dyson *et al.*, 2007; Dyson *et al.*, 2016; Hinton & Hale, 2001). Entre estas acciones destacan corpus digitales, diccionarios y materiales educativos. Por su parte, en Gallardo (2012) se presenta un libro con vocabulario básico otomí creado con el apoyo de estudiantes en actividades de vinculación comunitaria, el cual abarca frutas, animales, números, colores y partes del cuerpo. Hernández-Cruz *et al.* (2010) y Vargas (2017) documentan un diccionario Hñähñu en el Valle del Mezquital, con recursos escritos y orales. Además, De la Vega (2017) publicó en el Estado de México *Aprendiendo otomí (Hñähñu)*, un libro que promueve la cultura e identidad otomí a través de temas cotidianos. Ante la escasez de materiales pedagógicos en lenguas originarias para nivel básico, se diseñó un libro electrónico enfocado al rescate del Hñähñu (Hernández *et al.*, 2020), respondiendo a las necesidades de niños indígenas y enfrentando el desplazamiento lingüístico. Finalmente, en el trabajo de Escorza-Sánchez *et al.* (2018) se desarrolló una aplicación móvil para promover el aprendizaje del Hñähñu en escuelas bilingües de Ixmiquilpan.

El trabajo propuesto se basa en un algoritmo de aprendizaje estadístico-probabilístico (EyP) que traduce palabras del español al Hñähñu mediante una red neuronal y la normalización del error. El propósito principal es desarrollar un traductor Español-Hñähñu con fines culturales y lingüísticos, alimentado con vocabulario relacionado con familia, frutas, animales y partes del cuerpo. Las redes neuronales, como los modelos de inteligencia artificial, imitan la toma de decisiones humanas y permiten automatizar tareas. Su uso en escenarios experimentales ha mostrado ventajas al superar barreras culturales y tecnológicas.

En el ámbito del procesamiento de lenguaje natural para lenguas de bajos recursos, diversos estudios han utilizado técnicas basadas en aprendizaje automático y redes neuronales para abordar problemas de traducción, clasificación y recuperación de información. Sin embargo, la mayoría de estos enfoques requieren grandes volúmenes de datos paralelos, condición que rara vez se cumple para lenguas indígenas como el Hñähñu. Esta limitación motiva la búsqueda de estrategias alternativas de entrenamiento que puedan operar bajo escenarios de información limitada.

A diferencia de los sistemas modernos de traducción neuronal, que generalmente requieren grandes corpus paralelos y arquitecturas complejas basadas en codificadores-decodificadores o mecanismos de atención, el presente trabajo aborda el problema desde una perspectiva distinta. En lugar de proponer una nueva arquitectura de traducción, se propone un algoritmo de aprendizaje estadístico-probabilístico denominado EyP para el entrenamiento de redes neuronales artificiales. El objetivo es analizar la capacidad del algoritmo para aprender asociaciones bilingües a partir de representaciones vectoriales de palabras en un escenario de recursos limitados.

Las palabras del corpus Hñähñu-Español son transformadas en vectores numéricos de longitud fija, permitiendo que la red neuronal aprenda correspondencias léxicas mediante un proceso de entrenamiento basado en la caracterización estadística del error. Esta aproximación resulta especialmente relevante para lenguas indígenas con disponibilidad reducida de datos, donde los métodos neuronales convencionales suelen presentar dificultades de entrenamiento y generalización.

Aunque diversos trabajos han incorporado conceptos probabilísticos y estadísticos dentro del aprendizaje profundo, hasta donde alcanza el conocimiento de los autores del presente estudio, no se identificaron trabajos que utilicen explícitamente la media muestral, la desviación estándar y la función de distribución acumulada normal como parte del mecanismo de actualización de pesos y sesgos en redes neuronales. En este contexto, se propone el algoritmo estadístico-probabilístico EyP como una alternativa de entrenamiento orientada a escenarios con disponibilidad limitada de datos. Las principales contribuciones de este trabajo son:

1. El desarrollo de un algoritmo de aprendizaje estadístico-probabilístico (EyP) para redes neuronales artificiales basado en la normalización estadística del error y la función de distribución acumulada normal.
2. La implementación del algoritmo en una red neuronal capaz de aprender asociaciones bilingües a partir de palabras vectorizadas.
3. La validación experimental del algoritmo en un corpus Hñähñu-Español de bajos recursos, orientado a la preservación digital de lenguas indígenas.

Estado del arte

Realizando una revisión literaria con base en técnicas estadísticas y probabilísticas, se tienen los siguientes trabajos. Linka *et al.* (2022) integran redes neuronales con inferencia bayesiana para mejorar predicciones bajo incertidumbre; Kabir *et al.* (2019) usan densidades de probabilidad para reducir costos computacionales; Mohebbali *et al.* (2020) combinan reducción de características con métodos bayesianos; Khoussi *et al.* (2021) ajustan distribuciones a través de pruebas estadísticas y limpieza de datos mediante el método de cuatro gráficas; y Li *et al.* (2024) proponen un marco probabilístico donde las distribuciones de probabilidad alimentan redes neuronales para modelar estructuras complejas.

En los últimos años, el desarrollo de algoritmos de aprendizaje para redes neuronales ha experimentado una transición significativa desde enfoques puramente deterministas hacia metodologías fundamentadas en la estadística, la teoría de la probabilidad y los procesos estocásticos. Esta evolución ha permitido mejorar la capacidad de generalización de los modelos, incrementar su robustez ante datos ruidosos y proporcionar mecanismos para cuantificar la incertidumbre asociada a las predicciones. Como consecuencia, el aprendizaje profundo moderno puede entenderse no solo como un problema de optimización, sino también como un proceso de inferencia estadística sobre datos observados (Nalisnick *et al.*, 2023).

Una de las principales líneas de investigación actuales se enfoca en la cuantificación de incertidumbre dentro de las redes neuronales profundas. Estos enfoques buscan estimar simultáneamente las predicciones y el grado de confianza asociado a ellas, lo que resulta particularmente importante en aplicaciones críticas donde los errores pueden tener consecuencias significativas. Para ello, se han desarrollado técnicas basadas en inferencia bayesiana, ensamblados de modelos, inferencia variacional y métodos de muestreo probabilístico, los cuales permiten representar explícitamente diferentes fuentes de incertidumbre presentes en los datos y en los parámetros del modelo (Abdar *et al.*, 2021; Gawlikowski *et al.*, 2023).

De manera complementaria, el aprendizaje profundo probabilístico ha adquirido una creciente relevancia debido a su capacidad para integrar redes neuronales con modelos estadísticos capaces de representar fenómenos aleatorios complejos. En estos enfoques, los pesos sinápticos y las salidas de la red pueden modelarse mediante distribuciones de probabilidad, permitiendo capturar incertidumbre epistémica y aleatoria durante el proceso de aprendizaje. Este paradigma ha demostrado ventajas en tareas de clasificación, regresión y toma de decisiones bajo incertidumbre, particularmente cuando los conjuntos de datos son limitados o presentan alta variabilidad (Fortuin, 2022).

Otra línea de investigación relevante corresponde a los algoritmos de optimización estocástica utilizados durante el entrenamiento. Estos métodos emplean subconjuntos aleatorios de datos para actualizar los parámetros de la red, reduciendo significativamente el costo computacional y favoreciendo la convergencia en problemas de gran escala. Las investigaciones recientes muestran que las variantes modernas de descenso de gradiente estocástico y los algoritmos adaptativos continúan desempeñando un papel fundamental en el rendimiento de las redes neuronales profundas, debido a su capacidad para equilibrar velocidad de aprendizaje, estabilidad numérica y capacidad de generalización (Abdulkadirov *et al.*, 2023).

Asimismo, la integración de conceptos estadísticos dentro del propio mecanismo de aprendizaje constituye una tendencia emergente. Diversos estudios exploran el uso de distribuciones probabilísticas, funciones de densidad, estimadores muestrales, momentos estadísticos y medidas de incertidumbre para modular la actualización de los parámetros de la red. Estos enfoques buscan aprovechar la información contenida en la distribución del error de predicción para construir algoritmos más robustos frente a observaciones atípicas, ruido y variabilidad inherente a los datos, favoreciendo una convergencia más estable y una mejor capacidad de generalización (Abdar *et al.*, 2021; Nalisnick *et al.*, 2023).

Finalmente, la convergencia entre aprendizaje profundo, inferencia estadística y modelado probabilístico ha impulsado el desarrollo de nuevos algoritmos capaces de incorporar explícitamente conocimiento estadístico dentro del proceso de entrenamiento. Esta tendencia representa una de las áreas más activas de investigación en inteligencia artificial, ya que permite diseñar modelos más interpretables, confiables y adaptables a escenarios reales caracterizados por incertidumbre y variabilidad muestral. En este contexto, los algoritmos que emplean parámetros estadísticos derivados del error, tales como la media, la desviación estándar y funciones de distribución probabilística, constituyen una alternativa prometedora para mejorar la estabilidad y robustez del aprendizaje neuronal (Fortuin, 2022; Gawlikowski *et al.*, 2023).

En traducción automática neuronal, los enfoques actuales se basan principalmente en modelos Encoder-Decoder, redes recurrentes y arquitecturas Transformer, los cuales han demostrado resultados sobresalientes cuando se dispone de grandes corpus paralelos. Sin embargo, para lenguas de bajos recursos, la escasez de datos limita significativamente el desempeño de estos modelos. Por esta razón, han surgido investigaciones enfocadas en estrategias alternativas de representación y aprendizaje que permitan aprovechar conjuntos reducidos de datos sin requerir millones de parámetros entrenados.

En este contexto, la utilización de representaciones vectoriales de palabras junto con mecanismos alternativos de actualización de pesos constituye una línea de investigación relevante para el desarrollo de herramientas computacionales orientadas a la preservación de lenguas indígenas.

Materiales y métodos

El presente trabajo aborda el problema de la traducción léxica Hñähñu-Español mediante una red neuronal entrenada a partir de palabras representadas como vectores numéricos de longitud fija. Bajo esta formulación, el aprendizaje consiste en establecer correspondencias entre las representaciones vectoriales de ambos idiomas utilizando un corpus reducido de palabras. En lugar de emplear algoritmos de optimización convencionales basados exclusivamente en gradientes, se propone una estrategia de aprendizaje fundamentada en principios estadísticos y probabilísticos que aprovecha la información contenida en la distribución del error de predicción para actualizar los pesos sinápticos de la red neuronal. De esta manera, se busca proporcionar una alternativa de entrenamiento adecuada para escenarios donde la disponibilidad de datos es limitada y la preservación de lenguas indígenas representa un desafío tecnológico relevante.

El algoritmo estadístico-probabilístico EyP se ha propuesto como una estrategia de entrenamiento para redes neuronales artificiales. El método utiliza información estadística obtenida a partir del error de predicción para adaptar la actualización de los pesos sinápticos durante el proceso de aprendizaje.

En este trabajo se emplea una red neuronal monocapa. El diagrama de la red neuronal emplea una entrada $y(k)$, un estado interno $h(k)$, los pesos sinápticos definidos con $W(k)$ y el sesgo dado por $b(k)$, matemáticamente definidos como:

$$y(k) = [y_1(k), y_2(k), y_3(k), \dots, y_n(k),] \quad (1)$$

$$W(k) = [w_1(k), w_2(k), w_3(k), \dots, w_n(k),] \quad (2)$$

$$h(k) = W^T(k)y(k) + b(k) \quad (3)$$

$$\hat{y}(k) = \psi[h(k)] \quad (3)$$

donde el algoritmo de entrenamiento propuesto es basado en distribuciones probabilísticas y datos generados a partir de operaciones estadísticas, estimando la salida de la red $\hat{y}(k)$ y aplicando una función de activación $\psi[\cdot]$ al estado interno $h(k)$. Las funciones de activación más comunes empleadas en redes neuronales son: la función de saturación, sigmoide, tangente hiperbólica, entre otras. En este caso, se emplea la función tangente hiperbólica como activación (Baruch *et al.*, 2017; Ojeda *et al.*, 2019).

El objetivo del entrenamiento consiste en que el error converja a 0 mediante el ajuste de los pesos sinápticos y el sesgo de la red neuronal presentada en la Figura 1.

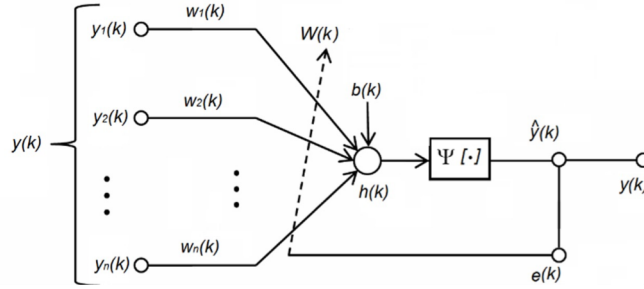


Figura 1. Modelo de una red neuronal *feedforward* mediante corrección de error.
Fuente: Elaboración propia.

La señal de error actúa como un mecanismo de retroalimentación que permite modificar el vector de pesos de la red neuronal con el propósito de minimizar el índice de desempeño definido en la ecuación 5.

$$J(k) = \frac{1}{2} e(k)^T e(k) = \frac{1}{2} e(k)^2 \quad (5)$$

A diferencia de los métodos usados para el aprendizaje de redes neuronales basando en gradientes, este algoritmo EyP incorpora una etapa de normalización estadística del error, calculando la media muestral y la desviación estándar de la salida durante el entrenamiento:

$$\mu(k) = \frac{\hat{y}_1(k) + \hat{y}_2(k) + \hat{y}_3(k) + \dots + \hat{y}_n(k)}{n} = \frac{1}{n} \sum_{i=1}^n \hat{y}_i(k) \quad (6)$$

$$\sigma(k) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\hat{y}_i(k) - \mu(k))^2} \quad (7)$$

Posteriormente, mediante el error y la desviación estándar se estima el z-score:

$$z(k) = \frac{e(k)}{\sigma(k) + \varepsilon} = \frac{y(k) - \hat{y}(k)}{\sigma(k) + \varepsilon} \quad (8)$$

Una vez obtenido el z-score, se calcula la probabilidad acumulada asociada utilizando la función de distribución acumulada de la distribución normal estándar:

$$P(k) = \Phi(z(k)) = \int_{-\infty}^{z(k)} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \quad (9)$$

donde $\Phi[\cdot]$ representa la función de distribución acumulada. En el algoritmo propuesto, esta probabilidad no se interpreta como un gradiente, sino como una medida estadística de la magnitud relativa del error. De esta forma, errores de diferente tamaño producen distintos niveles de actualización sobre los pesos sinápticos y el sesgo mediante el uso del Teorema del límite central (Anderson *et al.*, 2008; Oakland & Oakland, 2024).

La función de distribución acumulada transforma el error normalizado en una variable acotada dentro del intervalo $[0,1]$. Esta transformación reduce la influencia de errores extremos, evita actualizaciones excesivas de los parámetros y proporciona un mecanismo de aprendizaje más estable. Cuando el error se encuentra cerca de la media muestral, la probabilidad acumulada toma valores próximos a 0.5, produciendo ajustes moderados. Por el contrario, errores alejados de la media generan probabilidades cercanas a 0 o 1, incrementando el nivel de corrección aplicado a los parámetros de la red.

En este trabajo, el término z-score se utiliza como una medida de normalización estadística del error respecto a su variabilidad muestral. Esta normalización permite expresar el error en unidades de desviación estándar, proporcionando una medida relativa de su magnitud y favoreciendo una actualización más estable de los parámetros de la red neuronal.

Finalmente, la probabilidad acumulada se utiliza como entrada de funciones no lineales, tales como la función sigmoide o la tangente hiperbólica:

$$\Delta W = \frac{e^{P(k)} - e^{-P(k)}}{e^{P(k)} + e^{-P(k)}} = \tanh(P(k)) \quad (10)$$

$$\Delta b = \frac{L}{1 + e^{-\alpha P(k)}} \quad (11)$$

Las ecuaciones 10 y 11 generan los factores adaptativos utilizados para modificar los parámetros de la red neuronal. En particular, $\Delta W(k)$ se obtiene mediante la función tangente hiperbólica aplicada a la probabilidad acumulada, $P(k)$, produciendo una señal acotada que regula la magnitud de la actualización de los pesos sinápticos. De forma análoga, $\Delta b(k)$ se calcula mediante una función sigmoide escalada por el parámetro L , permitiendo ajustar de manera gradual el sesgo de la red (Ojeda-Misses *et al.*, 2024).

Posteriormente, ambos factores son incorporados en las ecuaciones de actualización 12 y 13, donde η representa la tasa de aprendizaje que controla la velocidad de adaptación de los parámetros. Valores pequeños de η producen modificaciones más suaves y estables, mientras que valores mayores aceleran el aprendizaje, aunque pueden incrementar el riesgo de oscilaciones durante el proceso de entrenamiento. Las salidas de estas funciones generan el factor adaptativo empleado para actualizar los pesos sinápticos y el sesgo de la red neuronal.

$$W(k+1) = W(k) + \eta \Delta W(k) \quad (12)$$

$$b(k+1) = b(k) + \eta \Delta b(k) \quad (13)$$

Finalmente, el algoritmo estadístico presentado para el aprendizaje de la red neuronal unicapa se resume en el pseudocódigo mostrado en la Figura 2. La descripción secuencial de cada etapa del procedimiento, incluyendo el cálculo de los parámetros estadísticos, la estimación probabilística y las reglas de actualización de pesos y sesgos, permite que el método pueda ser implementado y reproducido por otros investigadores bajo las mismas condiciones experimentales. De esta manera, se favorece la transparencia metodológica, la validación independiente de los resultados y la comparación objetiva con otros algoritmos de aprendizaje para redes neuronales.

Algoritmo 1. Algoritmo estadístico-probabilístico EyP para el aprendizaje de redes neuronales

1:	Inicializar $W(k), b(k), \eta, \alpha, L, \varepsilon > 0$ y el número máximo de épocas $E_{\max}$	
2:	$k \leftarrow 0$	
3:	Repetir	
4:	Para cada patrón de entrenamiento $y(k)$ hacer	
5:	Calcular el estado interno:	$h(k) = W^T(k) y(k) + b(k)$
6:	Calcular la salida de la red:	$\hat{y}(k) = \psi(h(k))$
7:	Calcular el error instantáneo:	$e(k) = y(k) - \hat{y}(k)$
8:	Fin para	
9:	Formar la muestra de salidas $\{\hat{y}_1(k), \hat{y}_2(k), \dots, \hat{y}_n(k)\}$, $n = 1, \dots, N$	
10:	Calcular la media muestral:	$\mu(k) = \frac{\hat{y}_1(k) + \hat{y}_2(k) + \dots + \hat{y}_n(k)}{n} = \frac{1}{n} \sum_{i=1}^n \hat{y}_i(k)$
11:	Calcular la desviación estándar muestral:	$\sigma(k) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\hat{y}_i(k) - \mu(k))^2}$
12:	Calcular el z-score:	$z(k) = \frac{e(k)}{\sigma(k) + \varepsilon} = \frac{y(k) - \hat{y}(k)}{\sigma(k) + \varepsilon}$
13:	Calcular la probabilidad acumulada (función de distribución normal estándar):	$P(k) = \Phi(z(k)) = \int_{-\infty}^{z(k)} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$
14:	Calcular el factor adaptativo para los pesos:	$\Delta W(k) = \tanh(P(k))$
15:	Calcular el factor adaptativo para el sesgo:	$\Delta b(k) = \frac{L}{1 + e^{-\alpha P(k)}}$
16:	Actualizar los pesos sinápticos:	$W(k+1) = W(k) + \eta \Delta W(k)$
17:	Actualizar el sesgo:	$b(k+1) = b(k) + \eta \Delta b(k)$
18:	$k \leftarrow k + 1$	
19:	Calcular el error cuadrático medio:	$J(k) = \frac{1}{2} e(k)^T e(k) = \frac{1}{2} \sum_{i=1}^n e_i(k)^2$
20:	Hasta que $J(k) \leq \text{umbral}$ o $k \geq E_{\max}$	
21:	Regresar $W(k), b(k)$	

Donde: $y(k)$ es el vector de entrada, $\hat{y}(k)$ la salida de la red, $e(k)$ el error instantáneo, n el número de muestras utilizadas, $\mu(k)$ la media muestral, $\sigma(k)$ la desviación estándar muestral, $\Phi(\cdot)$ la función de distribución acumulada de la distribución normal estándar, η la tasa de aprendizaje, α el parámetro de pendiente de la función sigmoide, L el límite superior del factor logístico y ε una constante pequeña para evitar divisiones entre cero.

Figura 2. Pseudocódigo del algoritmo de aprendizaje para la red neuronal basado en estadística y probabilidad (EyP).
Fuente: Elaboración propia.

Asimismo, se especifican las condiciones experimentales empleadas durante el entrenamiento, incluyendo la inicialización de los parámetros, el número máximo de épocas y el criterio de terminación, con el propósito de facilitar la implementación independiente del algoritmo y la comparación con otros métodos de aprendizaje reportados en la literatura.

Para calcular los parámetros estadísticos del error se empleó una muestra formada por todos los errores obtenidos durante una época de entrenamiento. El conjunto de datos utilizado estuvo compuesto por un corpus bilingüe formado por 1000 pares léxicos Hñähñu-Español, por lo que durante cada época la muestra estadística quedó integrada por 1000 errores individuales asociados a las salidas de la red neuronal. La selección corresponde a un muestreo exhaustivo sobre todos los patrones disponibles, por lo que no se realizó un muestreo aleatorio adicional ni una selección parcial de los datos. En consecuencia, los parámetros estadísticos, tales como la media muestral y la desviación estándar, fueron calculados utilizando la totalidad de los errores generados en cada época de entrenamiento. Este procedimiento garantiza que las medidas estadísticas empleadas sean representativas del comportamiento global del error y permite reproducir las condiciones experimentales bajo las cuales fue validado el algoritmo propuesto.

Resultados y discusión

A continuación, se presenta la aplicación del algoritmo de aprendizaje estadístico-probabilístico (EyP) en la arquitectura neuronal propuesta. La red neuronal emplea un esquema de realimentación de error y un identificador neuronal dinámico capaz de ajustar sus pesos sinápticos mediante información estadística derivada del error de predicción.

Para garantizar la reproducibilidad del método propuesto, el algoritmo EyP se implementó utilizando una red neuronal unicapa con una capa de entrada formada por vectores de longitud fija y una capa de salida con activación tangente hiperbólica. Los pesos sinápticos y el sesgo fueron inicializados aleatoriamente mediante una distribución uniforme en el intervalo $[-0.5, 0.5]$. El entrenamiento se llevó a cabo durante 1000 épocas utilizando una tasa de aprendizaje $\eta = 0.005$. Como criterio de terminación se empleó el número máximo de épocas o un error cuadrático medio (ECM) inferior a 10^{-3} . Estas condiciones experimentales permiten reproducir el procedimiento de entrenamiento y facilitan la validación independiente del algoritmo propuesto; sin embargo, los resultados presentados aquí solamente son de cuatro apalabras en Hñähñu: badu (pato), boxi (gallo), dāmoni (guajolota) y mey'o (chivo).

En los sistemas modernos de traducción automática neuronal (*Neural Machine Translation*, NMT), las secuencias lingüísticas son transformadas en representaciones numéricas que permiten a las redes neuronales aprender relaciones entre diferentes idiomas a partir de corpus bilingües. Los modelos Encoder-Decoder introducidos por Cho *et al.*, (2014a); Cho *et al.*, (2014b) demostraron que una secuencia de entrada puede representarse mediante vectores numéricos capaces de capturar información lingüística útil para tareas de traducción automática. En este trabajo se adopta un principio similar, aunque utilizando una arquitectura más simple orientada a escenarios de bajos recursos, donde cada palabra del corpus Hñähñu-Español es representada mediante un vector numérico de longitud fija.

Esta representación transforma el problema lingüístico en un problema de aprendizaje numérico, permitiendo que la red neuronal aprenda asociaciones léxicas entre palabras pertenecientes a ambos idiomas. A diferencia de los sistemas NMT de gran escala que emplean millones de parámetros y grandes corpus paralelos, el enfoque propuesto busca evaluar la capacidad del algoritmo EyP para aprender correspondencias bilingües utilizando un conjunto reducido de palabras vectorizadas. Bajo esta formulación, la red neuronal recibe como entrada el vector asociado a una palabra origen y produce como salida el vector correspondiente a la palabra destino, aprendiendo una función de mapeo entre ambos espacios vectoriales mediante la actualización estadístico-probabilística de los pesos sinápticos como se muestra en la Figura 3.

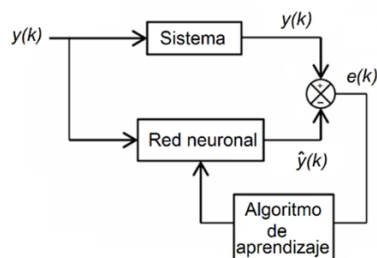


Figura 3. Modelo directo basado en identificador neuronal.
Fuente: Elaboración propia.

La Figura 4 muestra el proceso de aprendizaje para la palabra vectorizada *ts'ints'u* (pájaro), evidenciando el comportamiento de la red neuronal durante el entrenamiento. Se presentan tres aspectos fundamentales: la aproximación de la salida neuronal al vector objetivo, la evolución del error cuadrático medio (ECM) y la adaptación de los pesos sinápticos mediante el algoritmo EyP.

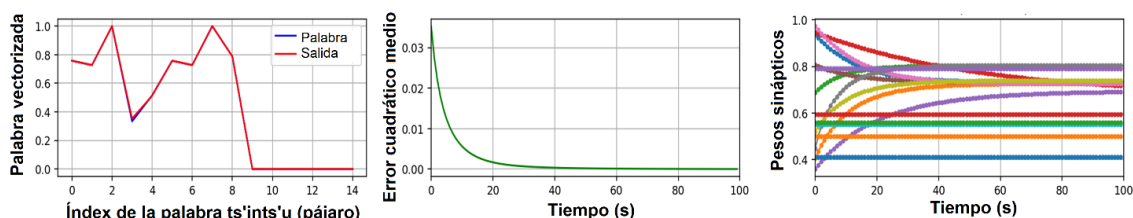


Figura 4. Aprendizaje de la palabra *ts'ints'u*, su error cuadrático medio y pesos sinápticos.
Fuente: Elaboración propia.

En la primera gráfica se observa que la salida generada por la red neuronal presenta inicialmente diferencias respecto al vector esperado; sin embargo, conforme avanza el entrenamiento, la aproximación mejora progresivamente hasta reproducir con mayor precisión la representación vectorial de la palabra objetivo. La segunda gráfica muestra una reducción sostenida del ECM desde valores cercanos a 0.035 hasta aproximadamente 0.001, indicando una disminución significativa del error de predicción. Finalmente, la tercera gráfica evidencia que los pesos sinápticos experimentan ajustes importantes durante las primeras iteraciones y, posteriormente, convergen hacia valores estables, comportamiento consistente con la reducción observada en el ECM.

Para construir el traductor propuesto, fue necesario implementar un proceso de vectorización de palabras, ampliamente utilizado en aplicaciones de procesamiento de lenguaje natural y traducción automática neuronal. La vectorización transforma palabras en representaciones numéricas que pueden ser procesadas por algoritmos de aprendizaje automático, permitiendo que la red neuronal identifique patrones y asociaciones entre vocablos pertenecientes a diferentes idiomas. En este trabajo, cada palabra del corpus Hñähñu-Español se representa mediante un vector de longitud fija construido a partir de la codificación numérica de sus caracteres.

El resultado puede interpretarse como un proceso de aprendizaje de asociaciones bilingües entre representaciones vectoriales. De esta forma, el aporte principal del trabajo no radica en proponer una nueva arquitectura de traducción neuronal, sino en introducir un algoritmo de aprendizaje estadístico-probabilístico alternativo para el ajuste de pesos sinápticos y evaluar su capacidad para aprender correspondencias léxicas en un escenario de recursos limitados. Este enfoque resulta particularmente relevante para lenguas indígenas como el Hñähñu, donde la disponibilidad de corpus paralelos es considerablemente menor que en los idiomas ampliamente estudiados.

Llevando a cabo el proceso de identificación con la red neuronal, ahora es necesario emplear el proceso de vectorización, que es de gran ayuda para desarrollar diccionarios de traducción. La vectorización convierte palabras en representaciones numéricas, lo que permite a los modelos de aprendizaje automático —especialmente redes neuronales— procesar eficazmente datos textuales. Su objetivo es transformar datos cualitativos en cuantitativos manejables por algoritmos (HaCohen-Kerner *et al.*, 2020). Matemáticamente, el alfabeto se define como un conjunto ordenado. Si denotamos el conjunto de caracteres del alfabeto Hñähñu como A :

$$A = \{a_1, a_2, a_3, \dots, a_n\} \quad (14)$$

donde $a_1, a_2, a_3, \dots, a_n$ representa los caracteres del alfabeto en Hñähñu, ordenados según su posición. En este caso, el conjunto incluye no solo los caracteres estándar del alfabeto español, sino también las vocales acentuadas y la letra ñ. La posición de cada carácter en el conjunto se puede usar para asignarle un valor numérico real. Entonces, un carácter $c \in A$ se define como:

$$f(c) = \frac{\text{index}(c)+1}{n} \quad (15)$$

Donde $\text{index}(c)$ es la posición del carácter c , y n es el número total de caracteres en el alfabeto. Es importante mencionar que, si el carácter no pertenece al alfabeto, se le asigna el valor de 0.

La vectorización de palabras constituye una etapa fundamental en el procesamiento del lenguaje natural (PLN), ya que permite transformar información textual en representaciones numéricas que pueden ser procesadas por modelos de aprendizaje automático y redes neuronales (HaCohen-Kerner *et al.*, 2020). Este principio es empleado por sistemas modernos de traducción automática neuronal, como Google Translate, DeepL y Microsoft Translator, donde las palabras o secuencias lingüísticas son convertidas en vectores capaces de capturar relaciones entre diferentes idiomas. Entre las técnicas clásicas de representación se encuentran la codificación one-hot, TF-IDF y métodos distribucionales como Word2Vec, GloVe y FastText (Kurniawan & Maharani, 2020; Nurdin *et al.*, 2020).

Aunque los sistemas actuales utilizan representaciones vectoriales más complejas y arquitecturas profundas basadas en mecanismos de atención y transformers, el concepto fundamental sigue siendo el mismo: transformar las palabras en representaciones numéricas que permitan a las redes neuronales aprender asociaciones lingüísticas. Debido a su simplicidad y eficacia, métodos como TF-IDF continúan siendo ampliamente utilizados en tareas de clasificación de textos, análisis de sentimientos y recuperación de información (Afifah *et al.*, 2021; Çelik & Koç, 2021; Kurniawan & Maharani, 2020; Nurdin *et al.*, 2020; Singh *et al.*, 2022).

Dado que cada carácter tiene un valor numérico, una palabra se transforma en un vector de longitud fija. Supongamos que una palabra está formada por una secuencia de caracteres. El vector de la palabra se define como:

$$\vec{X} = [f(c_1), f(c_2), f(c_3), \dots, f(c_m)] \quad (16)$$

Si $m < \text{index}$ del vector, se rellena con ceros hasta alcanzar la longitud deseada. Si $m > \text{index}$, se trunca a los primeros 15 elementos. Esta normalización asegura que todos los vectores tengan la misma dimensión, lo cual es un requisito para procesarlos con la red neuronal. En este caso, la longitud de las palabras es a lo más de 15 letras, por lo tanto, el vector se define como:

$$\vec{X} = [f(c_1), f(c_2), f(c_3), 0, \dots, 0] \in \mathbb{R}^{15} \quad (17)$$

Finalmente, el corpus vectorizado se construye aplicando este proceso a cada palabra de la lista de palabras en Hñähñu. Por ejemplo, si tenemos una lista de tres palabras en Hñähñu, cada palabra se transforma en un vector de longitud fija, y el conjunto completo de vectores se presenta como una matriz definida como:

$$D = \begin{bmatrix} \vec{X}_1 \\ \vec{X}_2 \\ \vec{X}_3 \end{bmatrix} \in \mathbb{R}^{3 \times 15} \quad (18)$$

La matriz D es el corpus de palabras de entrada, definido como un conjunto numérico que representa palabras en una forma interpretable por la red neuronal. Formalmente, $D \in \mathbb{R}^{k \times m}$ actúa como el tensor de entrada, donde cada fila corresponde a una palabra vectorizada de longitud fija m , y k es el número total de palabras en el corpus como:

$$D = \begin{bmatrix} \vec{X}_1 \\ \vec{X}_2 \\ \vec{X}_3 \\ \vdots \\ \vec{X}_n \end{bmatrix} \in \mathbb{R}^{k \times m} \quad (19)$$

Con parte de la validación del entrenamiento de la red neuronal, el sistema activa un módulo de prueba: un traductor interactivo que permite ingresar palabras en Hñähñu para traducirlas automáticamente al español, validando así la eficacia práctica del modelo y apoyando la preservación de lenguas indígenas en riesgo. La Figura 5 muestra la interfaz del traductor español a Hñähñu, donde el usuario ingresa "perro" y obtiene la traducción "tsat'io". La interfaz simple e intuitiva facilita consultas rápidas, fundamental para aplicaciones de apoyo lingüístico.

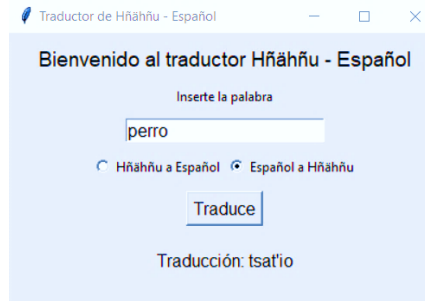


Figura 5. Ventana del traductor con palabra traducida de Español a Hñähñu (perro-tsath'io).

Fuente: Elaboración propia.

La Figura 6 muestra la misma interfaz del traductor, con la palabra en Hñähñu "y'o" ingresada para traducir al español. La opción seleccionada es "Hñähñu a Español", el sistema devuelve "borrego" como traducción. Esto refleja la flexibilidad del traductor o una posible omisión del usuario al cambiar la dirección. Este ejemplo resalta la capacidad bidireccional del sistema, facilitando la comprensión del Hñähñu.

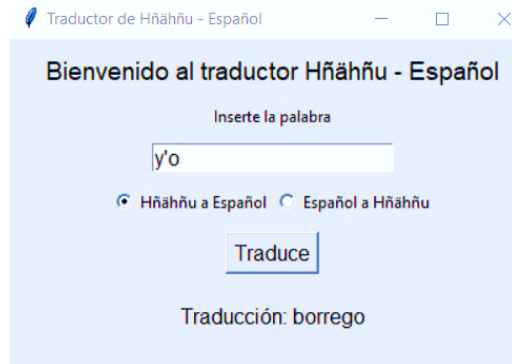


Figura 6. Ventana del traductor con palabra traducida de Hñähñu a Español (y'o-borrego).

Fuente: Elaboración propia.

Por otro lado, se comparan tres algoritmos de aprendizaje automático —Backpropagation (BP) (Baruch & Gaspar, 2009; Baruch *et al.*, 2017), Levenberg-Marquardt (LM) (Baruch *et al.*, 2017) y el algoritmo propuesto basado en la normalización del error con distribuciones probabilísticas, en la misma red neuronal implementada en Python. El objetivo es desarrollar un traductor Español-Hñähñu que convierte palabras en vectores numéricos y ajusta pesos mediante funciones no lineales para modelar relaciones complejas. El desempeño se evalúa con gráficos de convergencia, aprendizaje, ECM y tiempo de convergencia.

Los resultados experimentales con la red neuronal son mostrados a continuación en la Figura 7. En primer lugar, el aprendizaje de la palabra *badu* refleja un claro desempeño superior del algoritmo basado en estadística y probabilidad (EyP), en comparación con el algoritmo de BP y el algoritmo de LM, particularmente en iteraciones medias y avanzadas.

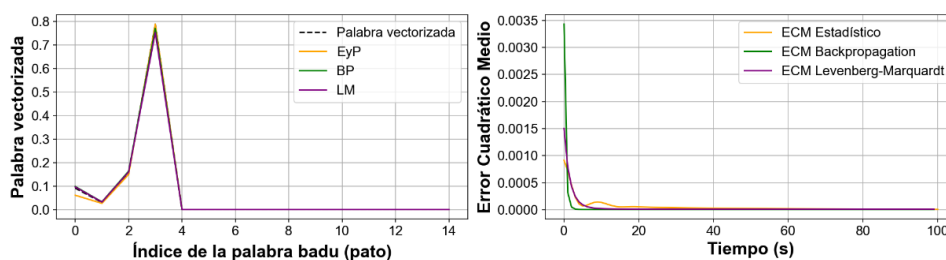


Figura 7. Identificación de la palabra vectorizada *badu* (pato) y sus respectivos errores cuadráticos medios.
Fuente: Elaboración propia.

En la Tabla 1 puede observarse el ECM para la palabra *badu* (pato). El algoritmo estadístico-probabilístico (EyP) inicia con un ECM bajo (~ 0.000914), mostrando una tendencia general decreciente, aunque presenta oscilaciones significativas entre las iteraciones 5 y 10. Tras este periodo, el error disminuye consistentemente hasta alcanzar valores cercanos a 0 en la iteración 100, lo que indica un proceso de aprendizaje exitoso. Backpropagation (MP) reduce el error rápidamente desde 0.003430 a niveles mínimos (0.000005) en solo unas pocas iteraciones. Por su parte, Levenberg-Marquardt (LM) muestra una mejora constante, alcanzando un error cercano a 0 desde la iteración 10, lo que refleja su alta eficiencia en convergencia.

Tabla 1. Errores cuadráticos medios para los algoritmos de aprendizaje basado en estadística y probabilidad (EyP), Backpropagation (BP) y Levenberg-Marquardt (LM) para la palabra *badu* (pato) en Hñähñu.

Iteración	ECM (EyP)	ECM (MP)	ECM (LM)
0	0.000914	0.003430	0.001503
2	0.000456	0.000052	0.000435
3	0.000236	0.000014	0.000248
5	0.000069	0.000006	0.000093
8	0.000136	0.000005	0.000033
10	0.000137	0.000005	0.000022
20	0.000058	0.000005	0.000014
40	0.000031	0.000005	0.000014
60	0.000020	0.000005	0.000014
80	0.000014	0.000005	0.000014
100	0.000009	0.000000	0.000000

Fuente: Elaboración propia.

Para la segunda palabra *boxi*, se observa nuevamente que el algoritmo EyP alcanza los valores de error más bajos en etapas avanzadas del entrenamiento (Figura 8).

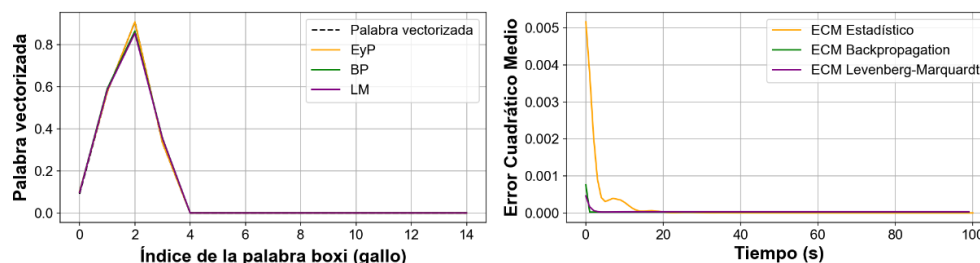


Figura 8. Identificación de la palabra vectorizada *boxi* (gallo) y sus respectivos errores cuadráticos medios.

Fuente: Elaboración propia.

La Tabla 2 muestra los errores cuadráticos medios para cada algoritmo, en su caso, para la palabra *boxi* (gallo). El algoritmo estadístico-probabilístico (EyP) reduce gradualmente el ECM desde 0.0051 hasta valores cercanos a 0, aunque con mayor número de iteraciones. Backpropagation (MP) muestra una rápida convergencia desde la iteración 2 (0.000028), manteniéndose constante sin mejorar más. Levenberg-Marquardt (LM) también alcanza un error mínimo similar, destacando por su precisión y estabilidad desde las primeras iteraciones.

Tabla 2. Errores cuadráticos medios para los algoritmos de aprendizaje basado en estadística y probabilidad (EyP), Backpropagation (BP) y Levenberg-Marquardt (LM) para la palabra *baxi* (gallo) en Hñähñu.

Iteración	ECM (EyP)	ECM (MP)	ECM (LM)
0	0.005163	0.000758	0.000460
2	0.002066	0.000028	0.000069
3	0.000918	0.000027	0.000037
5	0.000310	0.000027	0.000028
8	0.000374	0.000027	0.000032
10	0.000288	0.000027	0.000033
20	0.000026	0.000027	0.000034
40	0.000004	0.000027	0.000034
60	0.000003	0.000027	0.000034
80	0.000002	0.000027	0.000034
100	0.000001	0.000000	0.000000

Fuente: Elaboración propia.

En la Figura 9 se muestra la identificación para la palabra *dāmoni*, cuyo comportamiento del error muestra un indicativo de un buen desempeño, similar a los algoritmos presentados.

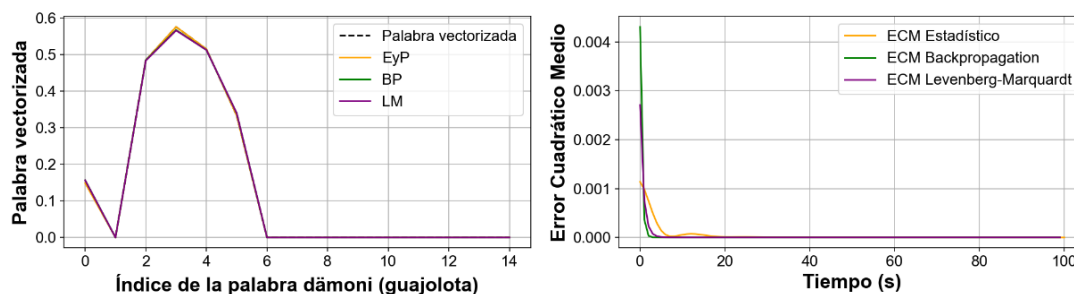


Figura 9. Identificación de la palabra vectorizada *dämoni* (guajolota) y sus respectivos errores cuadráticos medios.
Fuente: Elaboración propia.

Como puede verse en la Tabla 3, en el caso de la palabra *dämoni* (guajolota), EyP presenta una disminución continua del ECM, aunque con leves repuntes en la iteración 10. MP y LM ofrecen mejores resultados iniciales, con una reducción drástica del error desde las primeras iteraciones. MP destaca por alcanzar y mantener un error prácticamente nulo desde la iteración 10, superando a los otros dos métodos en precisión final.

Tabla 3. Errores cuadráticos medios para los algoritmos de aprendizaje basado en estadística y probabilidad (EyP), Backpropagation (BP) y Levenberg-Marquardt (LM) para la palabra *dämoni* (guajolota) en Hñähñu.

Iteración	ECM (EyP)	ECM (MP)	ECM (LM)
0	0.001137	0.004309	0.002710
2	0.000745	0.000032	0.000229
3	0.000496	0.000003	0.000072
5	0.000139	0.000001	0.000009
8	0.000023	0.000001	0.000002
10	0.000055	0.000001	0.000002
20	0.000010	0.000001	0.000002
40	0.000003	0.000001	0.000002
60	0.000001	0.000001	0.000002
80	0.000000	0.000001	0.000002
100	0.000000	0.000000	0.000000

Fuente: Elaboración propia.

Para la Figura 10, que muestra el aprendizaje de la palabra *mey'o*, se observa un rendimiento competitivo entre los tres algoritmos, aunque EyP tiene una magnitud mayor.

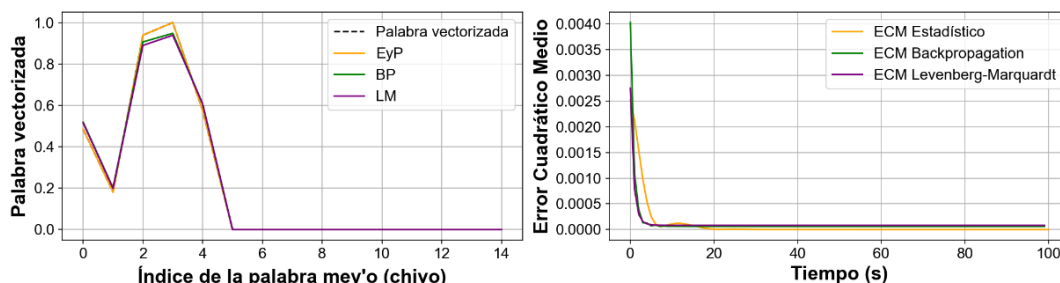


Figura 10. Identificación de la palabra vectorizada mey'o (chivo) y sus respectivos errores cuadráticos medios.

Fuente: Elaboración propia.

A pesar de tener una magnitud mayor en el ECM para *mey'o* (chivo), el algoritmo EyP es el único algoritmo que logra reducir el ECM hasta 0, mostrando un aprendizaje efectivo, y en su mayor parte progresivo. BP reduce el error rápidamente, pero se estanca en 0.000063 sin mayor mejora posterior. LM mantiene un comportamiento similar al BP, con un error estable en 0.000084, pero sin alcanzar la precisión lograda por el algoritmo EyP como se muestra en la Tabla 4.

Tabla 4. Errores cuadráticos medios para los algoritmos de aprendizaje basado en estadística y probabilidad (EyP), Backpropagation (BP) y Levenberg-Marquardt (LM) para la palabra *mey'o* (chivo) en Hñähñu.

Iteración	ECM (EyP)	ECM (BP)	ECM (LM)
0	0.002520	0.004022	0.002746
2	0.001544	0.000390	0.000296
3	0.000973	0.000132	0.000155
5	0.000249	0.000071	0.000095
8	0.000069	0.000070	0.000085
10	0.000112	0.000066	0.000084
20	0.000009	0.000064	0.000084
40	0.000001	0.000063	0.000084
60	0.000000	0.000063	0.000084
80	0.000000	0.000063	0.000084
100	0.000000	0.000000	0.000000

Fuente: Elaboración propia.

La red neuronal identifica dependencias a nivel de carácter y palabra, clave en lenguas morfológicamente complejas como el Hñähñu (otomí). Se compararon tres algoritmos de entrenamiento, Backpropagation (BP), Levenberg-Marquardt (LM) y uno innovador basado en estadística y probabilidad (EyP), aplicados a palabras vectorizadas en Python, donde LM demuestra mayor precisión inicial, mientras EyP logra la menor magnitud de error final en la mayoría de las palabras evaluadas, aunque presenta oscilaciones en etapas intermedias. La arquitectura usa activación no lineal y retroalimentación para minimizar el error cuadrático medio, con el objetivo de crear un traductor Hñähñu-español.

Los resultados obtenidos muestran que el algoritmo EyP presenta un comportamiento competitivo respecto a los algoritmos Backpropagation y Levenberg-Marquardt. Si bien estos métodos constituyen estrategias ampliamente utilizadas para el entrenamiento de redes neuronales, la propuesta estadístico-probabilística mostró una evolución estable del error y una adecuada capacidad para aprender asociaciones léxicas bajo condiciones de información limitada. Estos resultados sugieren que EyP puede representar una alternativa de entrenamiento para escenarios donde la disponibilidad de datos es reducida.

El algoritmo EyP destacó por su capacidad de adaptarse a la variabilidad morfológica del Hñähñu, logrando convergencia estable y sin depender fuertemente de condiciones iniciales, ideal para lenguas con pocos datos o estructuras complejas. Así, el sistema no solo compara métodos tradicionales con uno novedoso, sino que también ofrece una herramienta práctica para la revitalización del Hñähñu, con potencial para aplicaciones móviles y educativas en contextos bilingües y multiculturales.

El presente trabajo fue validado utilizando un corpus bilingüe compuesto por 1000 pares léxicos Hñähñu-Español correspondientes a palabras aisladas. Por lo tanto, los resultados obtenidos representan una evaluación inicial del algoritmo estadístico-probabilístico propuesto y no consideran estructuras sintácticas complejas ni la traducción de frases u oraciones completas. Asimismo, la disponibilidad limitada de recursos lingüísticos para el Hñähñu restringe el análisis de la capacidad de generalización del algoritmo en escenarios de mayor escala. Aunque el método mostró un comportamiento estable durante el entrenamiento, futuros trabajos deberán considerar corpus más extensos, conjuntos de datos balanceados y arquitecturas neuronales más complejas con el fin de evaluar la escalabilidad y robustez del algoritmo EyP en tareas de procesamiento del lenguaje natural más avanzadas.

El presente trabajo fue evaluado utilizando un corpus reducido de palabras aisladas en Hñähñu y español, por lo que los resultados obtenidos corresponden a una validación inicial del algoritmo propuesto. La metodología se enfocó en el aprendizaje de asociaciones léxicas y no considera estructuras sintácticas complejas ni traducción de frases completas. Asimismo, la disponibilidad limitada de recursos lingüísticos restringe la evaluación de capacidades de generalización en escenarios de mayor escala. Trabajos futuros deberán considerar corpus más extensos y arquitecturas neuronales modernas para analizar la escalabilidad del algoritmo EyP.

Conclusiones

En este trabajo se presentó un algoritmo de aprendizaje estadístico-probabilístico (EyP) para el entrenamiento de redes neuronales artificiales aplicado a la traducción léxica Hñähñu-Español. El método incorpora información estadística derivada del error de predicción y utiliza la función de distribución acumulada normal para generar factores adaptativos empleados en la actualización de los parámetros de la red.

Los resultados experimentales obtenidos sugieren que el algoritmo EyP constituye una alternativa viable para el entrenamiento de redes neuronales en escenarios de recursos limitados. La metodología propuesta permitió aprender asociaciones léxicas entre palabras en Hñähñu y español, mostrando un comportamiento estable durante el proceso de entrenamiento y evidenciando su potencial como herramienta tecnológica para apoyar la preservación y fortalecimiento de las lenguas indígenas.

Desde una perspectiva metodológica, la propuesta representa una aproximación basada en principios estadísticos y probabilísticos que puede extenderse a otros problemas de identificación, clasificación y procesamiento de señales. Asimismo, futuras investigaciones deberán considerar corpus de mayor tamaño, estructuras lingüísticas más complejas y arquitecturas neuronales modernas con el fin de analizar la escalabilidad y capacidad de generalización del algoritmo propuesto, así como su aplicación en tareas de procesamiento del lenguaje natural de mayor escala.

Finalmente, la incorporación de módulos de reconocimiento y síntesis de voz, la ampliación del vocabulario y la inclusión de expresiones y oraciones completas permitirán desarrollar herramientas computacionales más robustas orientadas a la preservación y revitalización del Hñähñu y otras lenguas indígenas. De igual manera, se considera relevante promover investigaciones interdisciplinarias que integren conocimientos de inteligencia artificial, lingüística computacional, procesamiento del lenguaje natural, educación y preservación del patrimonio cultural, con el propósito de desarrollar sistemas más completos que contribuyan tanto al avance científico como a la documentación, enseñanza y fortalecimiento de las lenguas originarias.

Conflicto de interés

Los autores no tienen conflicto de interés con la investigación realizada.

Referencias

- Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarek, V., & Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Information Fusion*, 76, 243–297. <https://doi.org/10.1016/j.inffus.2021.05.008>
- Abdulkadirov, R., Lyakhov, P., & Nagornov, N. (2023). Survey of optimization algorithms in modern neural networks. *Mathematics*, 11(11), 2466. <https://doi.org/10.3390/math11112466>
- Afifah, K., Yulita, I. N., & Sarathan, I. (2021). Sentiment analysis on telemedicine app reviews using XGBoost classifier. *International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA)*, 22–27. <https://doi.org/10.1109/ICAIBDA53487.2021.9689735>
- Aguilar, C. A., & García, H. A. (2023). Tecnologías del lenguaje aplicadas al procesamiento de lenguas indígenas en México: una visión general. *Lingüística y Literatura*, 44(84), 79–102. <https://doi.org/10.17533/udea.lyl.n84a04>
- Anderson, D. R., Sweeney, D. J., Williams, T. A., Camn, J. D., & Martin, K. (2008). *Estadística para administración y economía* (10ª ed.). Cengage Learning.
- Baruch, I. S., Arellano-Quintana, V., & Pérez-Reynaud, E. (2017). Complex-valued neural network topology and learning applied for identification and control of nonlinear systems. *Neurocomputing*, 233, 104–115. <https://doi.org/10.1016/j.neucom.2016.09.109>
- Baruch, I. S., & Gaspar, M. C. R. (2009). A Levenberg-Marquardt learning applied for recurrent neural identification and control of a wastewater treatment bioprocess. *International Journal of Intelligent Systems*, 24, 1094–1114. <https://doi.org/10.1002/int.20377>
- Çelik, Ö., & Koç, B. C. (2021). TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 23(67), 121–127. <https://doi.org/10.21205/deufmd.2021236710>
- Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014a). On the properties of neural machine translation: encoder–decoder approaches. *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 103–111. <https://doi.org/10.3115/v1/W14-4012>
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). Learning phrase representations using RNN encoder–decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>
- De la Vega, M. (2017). *Aprendiendo otomí (Hñähñu)*. Comisión Nacional para el Desarrollo de los Pueblos Indígenas (CDI). <https://www.gob.mx/cms/uploads/attachment/file/302157/aprendiendo-otomi-temoaya-estado-de-mexico-web.pdf>
- Dyson, L. E., Grant, S., & Hendriks, M. (2016). *Indigenous people and mobile technologies*. Routledge.

- Dyson, L. E., Hendriks, M., & Grant, S. (2007). Information technology and indigenous people. En L. E. Dyson, M. Hendriks, & S. Grant (eds.), *Information Technology and Indigenous People* (pp. 1-12). IGI Global. <https://doi.org/10.4018/978-1-59904-298-5>
- Escorza-Sánchez, Y. M., Martínez-Martín, G., Saldaña-Tapia, Y., & Maldonado-Catalán, O. (2018). Aplicación móvil para reforzar el aprendizaje de la lengua Hñähñu. *Revista de Tecnología y Educación*, 2(6), 23-31. https://www.ecorfan.org/republicoperu/research_journals/Revista_de_Tecnologia_y_Educacion/vol2n um6/Revista_de_Tecnolog%C3%ADa_y_Educaci%C3%B3n_V2_N6_4.pdf
- Fortuin, V. (2022). Priors in Bayesian deep learning: a review. *International Statistical Review*, 90(3), 563-591. <https://doi.org/10.1111/insr.12502>
- Gallardo, P. (2012). *Ritual, palabra y cosmos otomí: yo soy costumbre, yo soy de antigua*. UNAM. <https://historicas.unam.mx/publicaciones/publicadigital/libros/ritualpalabra/cosmos.html>
- Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., & Zhu, X. X. (2023). A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56, 1513-1589. <https://doi.org/10.1007/s10462-023-10562-9>
- HaCohen-Kerner, Y., Miller, D., & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*, 15(5), e0232525. <https://doi.org/10.1371/journal.pone.0232525>
- Hernández, Y., Guzmán, M. G., & Simón, E. (2020). Un libro electrónico en lengua hñähñu para promover el rescate y la ingesta de alimentos saludables. *Revista Lengua y Cultura*, 1(2), 77-84. <https://doi.org/10.29057/lc.v1i2.5432>
- Hernández-Cruz, L., Victoria-Torquemada, M., & Sinclair-Crawford, D. (2010). *Diccionario del Hñähñu (otomí) del Valle del Mezquital, Estado de Hidalgo*. <http://docencia.uaeh.edu.mx/estudios-pertinencia/docs/hidalgo-municipios/Valle-Del-Mezquital-Diccionario-Hnahnu.pdf>
- Hinton, L., & Hale, K. (eds.). (2001). *The green book of language revitalization in practice*. Academic Press. <https://www.journals.uchicago.edu/doi/10.1086/424555>
- Instituto Nacional de Estadística y Geografía (INEGI). (2015). *Encuesta Intercensal 2015. Lenguas indígenas nacionales y su distribución geográfica*. Instituto Nacional de Estadística y Geografía. <https://www.inegi.org.mx/programas/intercen-sal/2015/>
- Instituto Nacional de Estadística y Geografía (INEGI). (2024). *Estadísticas a propósito del día Internacional de los Pueblos Indígenas* [Comunicado de prensa núm. 472/24]. https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2024/EAP_PueblosInd24.pdf
- Kabir, H. M. D., Kebria, P. M., Khosravi, A., & Nahavandi, S. (2019). *Red neuronal para el cálculo de densidad de probabilidad en series temporales*. En 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Sydney, Australia (pp. 341-346). <https://doi.org/10.1109/CloudCom.2019.00059>
- Khoussi, S., Heckert, A., Battou, A., & Bensalem, S. (2021). *A neural networks-based methodology for fitting data to probability distributions*. En 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA), Tangier, Morocco (pp. 1-7). IEEE. <https://doi.org/10.1109/AICCSA53542.2021.9686821>
- Kurniawan, F. W., & Maharani, W. (2020). *Indonesian Twitter sentiment analysis using Word2Vec*. En 2020 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia (pp. 1-6). <https://doi.org/10.1109/icodsa50139.2020.9212906>
- Lastra, Y. (2010). *Los otomíes, su lengua y su historia*. IIA-UNAM. https://editorialiaia.unam.mx/omp/index.php/publicaciones/catalog/book/otomies_lengua_historia
- Li, C., Liu, Y., & Ma, Z. M. (2024). Neural networks taking probability distributions as input: a framework for analyzing exchangeable networks. *Neurocomputing*, 584, 127572. <https://doi.org/10.1016/j.neucom.2024.127572>
- Linka, K., Schäfer, A., Meng, X., Zou, Z., Karniadakis, G. E., & Kuhl, E. (2022). Bayesian physics-informed neural networks for real world nonlinear dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 402, 115346. <https://doi.org/10.1016/j.cma.2022.115346>

- Mohebbi, B., Tahmassebi, A., Meyer-Baese, A., & Gandomi, A. H. (2020). Probabilistic neural networks: a brief overview of theory, implementation, and application. En P. Samui, D. T. Bui, S. Chakraborty, & R. C. Deo (eds.), *Handbook of probabilistic models* (pp. 347–367). Butterworth-Heinemann. <https://doi.org/10.1016/B978-0-12-816514-0.00014-X>
- Nalisnick, E., Smyth, P., & Tran, D. (2023). A brief tour of deep learning from a statistical perspective. *Annual Review of Statistics and its Application*, 10, 219–246. <https://doi.org/10.1146/annurev-statistics-032921-013738>
- Nurdin, A., Seno, B. A., Bustamin, A., & Abidin, Z. (2020). Perbandingan kinerja word embedding Word2Vec, GloVe, dan FastText pada klasifikasi teks. *Jurnal Tekno Kompak*, 14(2), 74–79. <https://doi.org/10.33365/jtk.v14i2.732>
- Oakland, J., & Oakland, R. (2024). *Statistical process control and data analytics* (8a ed.). Routledge. <https://doi.org/10.4324/9781003439080>
- Ojeda, M. A., Solomon, I., & Soria, A. (2019). A real-time identification for hand-based movements using Recurrent Complex-Valued Neural Networks. 2019 IEEE 4th Colombian Conference on Automatic Control (CCAC), Medellin, Colombia (pp. 1–6). <https://doi.org/10.1109/CCAC.2019.8920864>
- Ojeda-Misses, M. A., Martinez-Arango, H., López-Morales, V., Franco-Árcega, A., & Márquez-Grajales, A. (2024). Self-tuned closed-loop controller based on statistical data using a servomechanism. 2024 XXVI Robotics Mexican Congress (COMRob), Torreón, Coahuila, México (pp. 27–32). <https://doi.org/10.1109/COMRob64055.2024.10777440>
- Singh, S., Kumar, K., & Kumar, B. (2022). Sentiment analysis of Twitter data using TF IDF and machine learning techniques. En 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM IT CON), Faridabad, India (pp. 252–255). <https://doi.org/10.1109/com-it-con54601.2022.9850477>
- Vargas, I. (2017). Experiencias de un proyecto de revitalización lingüística del hñähñu (otomí) del Valle del Mezquital, Hidalgo: de actores, discursos y prácticas. *Zeitschrift für romanische Philologie*, 133(4), 1064–1090. <https://doi.org/10.1515/zrp-2017-0055>