

Recibido 06 Jul 2026

ReCIBE, Año 15 No. 3, Sep 2026

Aceptado 10 Oct 2026

**Arquitectura perceptrón multicapa con reducción dimensional por
descomposición de valores singulares para el reconocimiento de
dígitos manuscritos**

**Multilayer Perceptron Architecture with Dimensionality Reduction Using
Singular Value Decomposition for Handwritten Digit Recognition.**

Álvaro Anzueto Rios¹

aanzueto@ipn.mx

Alma Sánchez Ramírez¹

aasanchezra@ipn.mx

Rafael Bayareh Mancilla¹

rbayarehm@ipn.mx

¹ Instituto politécnico nacional-UPIITA, México

Resumen

El reconocimiento de patrones visuales mediante redes neuronales de tipo perceptrón multicapa se emplea en tareas con recursos limitados de cómputo, memoria y tiempo de entrenamiento. En este trabajo se implementa un método que aplica la descomposición por valores singulares (SVD) y se evalúan en un perceptrón multicapa (MLP), para clasificar imágenes de dígitos manuscritos de 28×28 píxeles. La SVD sirve a dos fines simultáneos; reduce la dimensionalidad al proyectar las imágenes sobre el subespacio singular dominante, y a la vez ofrece a la primera capa una inicialización determinista e interpretable. Un punto por considerar en el diseño experimental es que la base singular se estima exclusivamente sobre la partición de entrenamiento y se aplica, sin recalcularse, a la de prueba; lo que evita por construcción la transferencia indebida de información entre particiones. La evaluación es realizada sobre una base de 200 imágenes de entrenamiento y 60 de prueba, balanceadas en diez clases; la SVD comprime la entrada a un intervalo comprendido entre el 8.2 % y el 18.4 % de su dimensión original e incluso superando la exactitud del modelo sin reducción, y disminuyendo de manera monótona la pérdida para la fase de prueba. El modelo alcanza una exactitud del 91.7 % y un puntaje F_1 macro de 0.915 con únicamente 19,850 parámetros entrenados, frente a los 101,770 de una arquitectura MLP equivalente sin reducción. La corrección de la retropropagación y la integridad del protocolo experimental se verifican numéricamente.

Palabras clave: perceptrón multicapa; red neuronal tipo vanilla; descomposición por valores singulares; reducción dimensional; extracción de características; retropropagación; clasificación de imágenes.

Abstract

Visual pattern recognition using multilayer perceptron neural networks is employed in tasks with limited computational resources, memory, and training time. This work implements a method that applies singular value decomposition (SVD) and is evaluated in a multilayer perceptron (MLP) to classify 28×28 pixel handwritten digit images. SVD serves two simultaneous purposes: it reduces dimensionality by projecting the images onto the dominant singular subspace, and at the same time provides the first layer with a deterministic and interpretable initialization. A key consideration in the experimental design is that the singular basis is estimated exclusively on the training partition and applied, without recalculation, to the test partition; this prevents, by design, the improper transfer of information between partitions. The evaluation is performed on a basis of 200 training images and 60 test images, balanced across ten classes. The SVD compresses the input to between 8.2% and 18.4% of its original size, even surpassing the accuracy of the unreduced model, and monotonically decreasing the loss for the testing phase. The model achieves an accuracy of 91.7% and a macro F_1 score of 0.915 with only 19,850 trained parameters, compared to 101,770 for an equivalent unreduced MLP architecture. Backpropagation correctness and the integrity of the experimental protocol are numerically verified.

Keywords: multilayer perceptron; vanilla neural network; singular value decomposition; dimensionality reduction; feature extraction; backpropagation; image classification.

Introducción

El reconocimiento de objetos o elementos contenidos en imágenes forma parte de las múltiples tareas que pueden resolverse aplicando el aprendizaje automático; un caso específico para analizar es la lectura automática de dígitos manuscritos, cuya resolución con redes neuronales fue establecida por LeCun *et al.* (1998) y actualmente continúa como banco de pruebas de referencia. Durante la última década, la investigación se ha visto influenciada por el principio de que añadir más capas y más parámetros garantiza un mejor rendimiento; sin embargo, esta búsqueda de profundidad implica altos consumos de cómputo, memoria y tiempo de entrenamiento, que

resultan prohibitivos en entornos con recursos limitados, tales como dispositivos embebidos o plataformas móviles. Citamos las arquitecturas VGG o ResNet como ejemplos de desarrollo, aunque ambas no siempre resultan adecuadas en el contexto de reducción de recursos, puesto que su capacidad de síntesis de información se obtiene a costa de un elevado número de parámetros.

En la búsqueda del equilibrio entre capacidad y eficiencia, surge una línea de trabajo alternativa, que implica mover la parte de la carga de la extracción de características, desde las capas profundas, hacia una etapa de procesamiento algebraico con fundamento teórico. La descomposición en valores singulares ofrece un marco apropiado para ello, desarrollando la tarea de factorizar la matriz de datos en sus direcciones de máxima varianza (Eckart & Young, 1936; Jolliffe & Cadima, 2016), permite descartar las componentes de menor energía, habitualmente asociadas a ruido y a detalles poco discriminativos y conservar una representación compacta que preserva la información de las componentes principales; esto no solo simplifica el problema, sino que además posee una interpretación geométrica establecida, en contraste con las representaciones aprendidas por las redes profundas, más difíciles de interpretar; al respecto, Hinton y Salakhutdinov (2006) mostraron que las redes neuronales convencionales pueden aprender códigos de baja dimensión, lo que motiva el interés por integrar la reducción dimensional en la arquitectura.

En este trabajo se aplica un diseño con complejidad reducida, la arquitectura de red neuronal desarrollada consta de una sola capa oculta y un número de parámetros entrenables, considerablemente menor que el de una arquitectura convencional equivalente, que recibe el vector de entrada sin reducir. Esta orientación, que prioriza la simplicidad estructural, la interpretabilidad y la facilidad de despliegue, es afín a la promovida por las redes *vanilla* (Chen et al., 2023). Por tanto, la propuesta en este trabajo consta de aplicar una etapa de reducción aplicando el proceso matemático de SVD, el cual cumple de manera simultánea dos funciones; uno comprimir la entrada y dos, inicializar de forma determinista la capa de proyección de la red.

Tomando como base estos puntos el trabajo diseña, formaliza y valida de manera empírica esta combinación mediante el reconocimiento de dígitos escritos a mano, buscando alcanzar un grado de exactitud competitivo y un valor de pérdida reducida. Las contribuciones de este trabajo no residen en la combinación de SVD y perceptrón multicapa cuyo uso en conjunto cuenta con antecedentes, sino en la forma específica en que ambos se integran y se evalúan. En concreto: (i) la capa de proyección se construye con la base singular y permanece fija durante todo el entrenamiento, de modo que opera como un extractor de características determinista e interpretable, sus pesos son las eigen-imágenes del conjunto, y no como una capa que se ajusta con los datos, a diferencia de los esquemas en que la inicialización por SVD se refina posteriormente (Lytvyn et al., 2024); (ii) la base singular se estima exclusivamente sobre la partición de entrenamiento y se aplica sin recalcularse a la de prueba, protocolo cuya pertinencia se corrobora mediante una contraprueba numérica; (iii) un estudio de ablación sobre el rango de truncamiento cuantifica el compromiso entre compresión y desempeño; y (iv) la corrección de la retropropagación se verifica numéricamente. El resultado es un modelo de una sola capa oculta que alcanza un desempeño competitivo con un número de parámetros entrenables cinco veces menor que el de un MLP equivalente, lo que lo hace apropiado para escenarios con recursos limitados.

El resto de este trabajo se organiza de la siguiente manera; la Sección 2 revisa los trabajos relacionados y la Sección 3 formaliza el método propuesto. Posteriormente, las Secciones 4 a 6 abordan los resultados y su verificación y las conclusiones de la propuesta.

Trabajos relacionados

La interacción que se da entre las redes neuronales y los métodos de factorización matricial ha producido una literatura amplia, de la cual se seleccionan los aportes más pertinentes para el presente estudio. El trabajo de Lytvyn et al. (2024) constituye el antecedente más directo: propone combinar la SVD con un MLP totalmente conectado, empleando la descomposición tanto para el preprocesamiento de las imágenes como para la inicialización de las capas, y reporta un incremento de la exactitud utilizando una fracción reducida de la dimensión original de la entrada. De ese trabajo tomamos la idea central de inicializar la proyección de los datos con los vectores singulares y aplicar una aproximación empleando un conjunto reducido de imágenes.

Considerando una perspectiva complementaria, Goodfellow et al. (2016) sistematizan el formalismo matricial de la propagación hacia adelante y hacia atrás, así como la verificación de la implementación mediante una comprobación numérica del gradiente, elementos que este trabajo retoma.

Ahora consideramos una perspectiva teórica desarrollada en el trabajo de Du et al. (2022), ellos presentan un tratamiento detallado del perceptrón; han considerado su aprendizaje, su generalización y las estrategias de inicialización de pesos, que fundamenta varias de las decisiones de diseño adoptadas, poniendo énfasis en el método de inicialización de He y el tipo de activación rectificada. El empleo de la metodología SVD, para el tópico del reconocimiento visual, tenemos un antecedente clásico que es conocido como eigenfaces (Turk & Pentland, 1991), que presenta los rostros mediante sus vectores singulares dominantes. En cuanto al carácter compacto del modelo, la red VanillaNet (Chen et al., 2023) demuestra que arquitecturas de escasa profundidad pueden alcanzar un desempeño competitivo.

Recientemente dos trabajos han desarrollado una metodología práctica que se incorporan al protocolo presente. En el primero de ellos, Galis et al. (2025) se considera la ponderación de características para agrupamiento no supervisado, subrayan la conveniencia de evaluar el desempeño con múltiples métricas complementarias; siguiendo esa recomendación, se reportan no solo la exactitud, sino también la precisión, la exhaustividad y el puntaje F_1 por clase. En el segundo, Abaza y Gheorghita (2025) revisa en un marco de redes neuronales para el control de operaciones de torneado, documentan un proceso iterativo y estructurado de configuración del modelo que motiva la descripción explícita de cada decisión arquitectónica presentada en la Sección 3.5. La Tabla 1 sintetiza estos aportes y su relación con la propuesta, que integra la SVD de Lytvyn et al. (2024), el formalismo de entrenamiento de Huang et al. (2020) y Du et al. (2022), el diseño compacto de Chen et al. (2023) y las prácticas de evaluación de Galis et al. (2025) y Abaza y Gheorghita (2025).

Tabla 1: Matriz comparativa con los trabajos relacionados.

Criterio	Lytvyn	Huang	VanillaNet
Uso de SVD	Sí (núcleo)	No	No
Reducción dim.	144/784	—	—
Dataset	MNIST	F-MNIST	ImageNet
Aplicabilidad	Alta	Media	Conceptual

Método

En el trabajo se considera una secuencia de componentes que parten de la imagen original sin procesar hasta concluir con la predicción de la clase. Las imágenes se vectorizan y se normalizan para construir las matrices de datos (Sección 3.1). Luego se avanza con el cálculo de la descomposición por valores singulares de la partición de entrenamiento y se conserva un número reducido de componentes, lo que define el operador de reducción dimensional (Sección 3.2); dicho operador proyecta los datos sobre el subespacio singular, cuya distribución se analiza (Sección 3.3). Tanto la base de vectores singulares como los pesos iniciales de la red, los cuales se estiman únicamente con la partición de entrenamiento y se aplican, sin recalcularse, en la fase de prueba (Sección 3.4); así, ninguna información, en esa fase, interviene en el ajuste (protocolo sin fuga de información).

Por último, las representaciones reducidas alimentan la arquitectura de la red, que se describe en detalle (Sección 3.5), y se definen las métricas de evaluación (Sección 3.6). La Figura 1 resume esta secuencia en una visión de conjunto organizada en dos carriles, y cada bloque remite a la subsección que lo detalla.

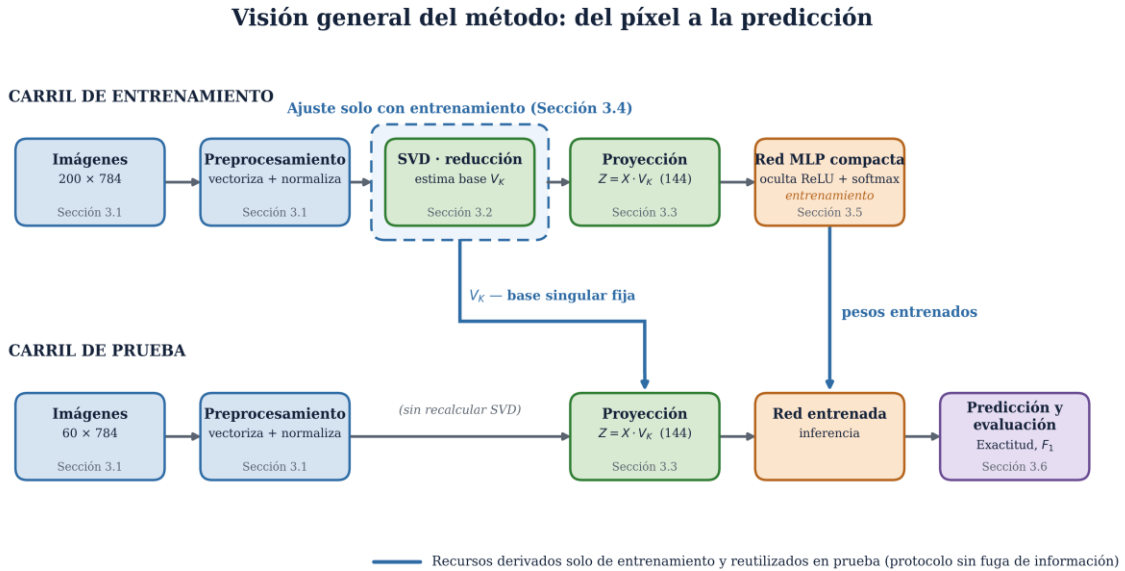


Figura 1: Visión general del método en dos carriles. La base singular V_K (región punteada) y los pesos de la red se estiman solo con datos de entrenamiento y se reutilizan, sin recalcularse, en el carril de prueba (conectores en azul); así, ninguna información de prueba interviene en el ajuste (protocolo sin fuga de información). Cada bloque remite a la subsección que lo detalla.

Conjunto de datos y preprocesamiento

El conjunto de datos está constituido por imágenes de dígitos manuscritos en escala de grises, de 28×28 píxeles y profundidad de ocho bits, organizadas en una partición de entrenamiento de 200 ejemplares y una de prueba de 60, ambas balanceadas entre las diez clases. Cada imagen $I_i \in \mathbb{R}^{28 \times 28}$ se transforma en un vector columna mediante la operación de vectorización (reordenamiento de los píxeles en una sola columna), $x_i = \text{vec}(I_i) \in \mathbb{R}^{784}$, y sus intensidades se reescalan al intervalo unitario mediante $x_i \leftarrow x_i/255$. Al ser una transformación elemento a elemento y sin parámetros estimados a partir de los datos, la normalización no introduce

dependencia entre las particiones. Las imágenes vectorizadas se apilan por filas en dos matrices de datos: $X_{tr} \in \mathbb{R}^{200 \times 784}$ y $X_{te} \in \mathbb{R}^{60 \times 784}$, donde los subíndices tr y te denotan, respectivamente, las particiones de entrenamiento (*training*) y de prueba (*test*). Esta convención de subíndices se mantiene en todo el documento.

Descomposición en valores singulares

La descomposición en valores singulares factoriza una matriz en el producto de tres factores con propiedades geométricas precisas. Para la matriz de entrenamiento se expresa como

$$X_{tr} = U \Sigma V^T, \quad (1)$$

donde $U \in \mathbb{R}^{m \times m}$ (con $m = 200$, el número de imágenes de entrenamiento) contiene los vectores singulares izquierdos y cumple $U^T U = I$; $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_m)$ es la matriz diagonal de valores singulares ordenados de forma no creciente, $\sigma_1 \geq \dots \geq \sigma_m \geq 0$; y las filas de V^T constituyen un conjunto ortonormal de direcciones de características, interpretables como las “eigen-imágenes” del conjunto. Los valores singulares cuantifican la energía —la varianza— que cada dirección aporta a la representación de los datos.

El sustento teórico de la reducción lo proporciona el teorema de Eckart–Young (1936), según el cual la mejor aproximación de rango K de la matriz, en norma de Frobenius, se obtiene reteniendo los primeros K términos de la descomposición, con un error residual igual a la energía descartada:

$$\|X_{tr} - U_K \Sigma_K V_K^T\|_F^2 = \sum_{i>K} \sigma_i^2. \quad (2)$$

La elección del rango K se guía por la fracción de energía acumulada $E(K) = \sum_{i \leq K} \sigma_i^2 / \sum_i \sigma_i^2$, representada en la Figura 2 junto con el espectro de valores singulares. El espectro decae con rapidez, lo que indica que un número reducido de componentes concentra la mayor parte de la energía: con $K = 144$ se retiene el 99.85 %, mientras que con $K = 64$ y $K = 100$ se conservan el 96.39 % y el 99.01 %, respectivamente. Estos valores justifican la adopción de $K = 144$ como configuración primaria.

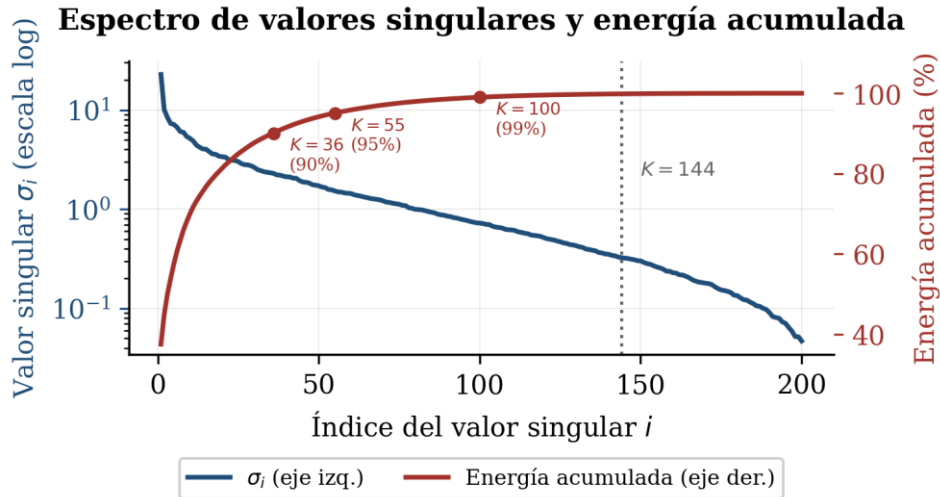


Figura 2: Espectro de valores singulares de X_{tr} en escala logarítmica (azul) y energía acumulada (rojo). Se indican los rangos necesarios para retener el 90 %, 95 % y 99 % de la energía, así como la configuración primaria $K = 144$.

Proyección y distribución en el subespacio singular

El operador de reducción dimensional es el proyector $V_K \in \mathbb{R}^{784 \times K}$, formado por las primeras K columnas de V ; es decir, por las K direcciones singulares de mayor energía. Aplicarlo a una partición consiste en una única multiplicación matricial,

$$Z = X V_K, \quad Z_{tr} \in \mathbb{R}^{200 \times K}, \quad Z_{te} \in \mathbb{R}^{60 \times K}, \quad (3)$$

que transforma cada imagen de 784 dimensiones en un vector de solo K coordenadas. Cada una de esas coordenadas posee una interpretación geométrica directa,

$$z_i = x^\top v_i, \quad (4)$$

el producto interno entre la imagen x y el i -ésimo vector singular v_i , que mide la proyección de la imagen sobre esa dirección; en otras palabras, z_i cuantifica cuánta presencia tiene en la imagen el patrón asociado a v_i .

Para revisar la conformación o estructura del espacio final, la Figura 3 presenta la interpretación de las imágenes en la base de entrenamiento, sobre sus tres primeras coordenadas (z_1, z_2, z_3) que representan las componentes principales, coloreadas por clase; estas tres direcciones concentran el 50 % de la varianza total. En esa representación; análoga a las del análisis de componentes principales (Jolliffe & Cadima, 2016), las clases tienden a posicionarse en regiones separadas y diferenciadas; ahora si consideramos que esta separación se aprecia ya con solo tres de las K direcciones, es posible interpretar que el subespacio singular preserva información discriminativa relevante para la clasificación posterior, pese a la considerable reducción de la dimensionalidad de los datos.

Distribución de las imágenes en el espacio de los tres primeros componentes singulares (entrenamiento)

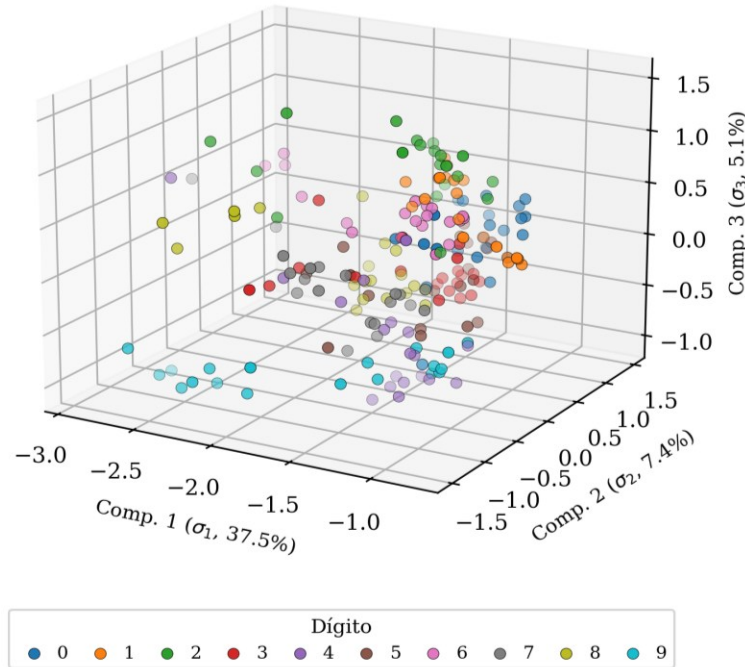


Figura 3: Distribución de los tres primeros componentes singulares (z_1, z_2, z_3), donde $z_i = x^\top v_i$. Cada punto es una imagen y el color indica la clase (dígito), para la base de datos de entrenamiento. Entre paréntesis se indica la varianza calculada por cada eje.

Separación de las particiones sin transferencia de información

La separación de las particiones sigue el principio de que, el vector considerado como proyección, V_K se estima *únicamente* a partir de X_{tr} (Ec. (1)) y se aplica un tiempo después, sin recalcularse, sobre X_{te} . Esto asegura que ninguna información de los datos de prueba interviene en la construcción de la base de información en el entrenamiento, lo que evita transferencia indebida de información entre la fase de entrenamiento y la de prueba (un fenómeno conocido como *fuga de información*, que se produce cuando datos de prueba influyen en el ajuste del modelo y conducen a estimaciones de desempeño optimistas). La sección 4.6 documenta este proceso y explica la influencia de incluir el conjunto de prueba en el cálculo de la descomposición y cómo esto permite medir la influencia de los datos al construir la matriz de la base singular, lo que confirma la relevancia del protocolo presente en el trabajo.

Arquitectura de la red

La arquitectura se construye siguiendo un criterio de complejidad reducida lo que provoca una arquitectura compacta: considerando el empleo del menor número de capas y de parámetros que permita clasificar con exactitud, trasladando la tarea de la extracción de características hacia la proyección singular descrita en la Sección 3.2. Formalmente, el modelo es una función compuesta $f = L_2 \circ L_1 \circ L_0$ de cuatro elementos; es decir, una capa de proyección, una capa oculta, una capa de salida y la función de pérdida que guía el aprendizaje, que se presentan a continuación. La Figura 4 ofrece una representación en forma de grafos de neuronas, y la Tabla 2 presenta de manera cuantificada las dimensiones y los parámetros de cada componente.

Grafo del perceptrón multicapa entrenable (entrada → oculta → salida)

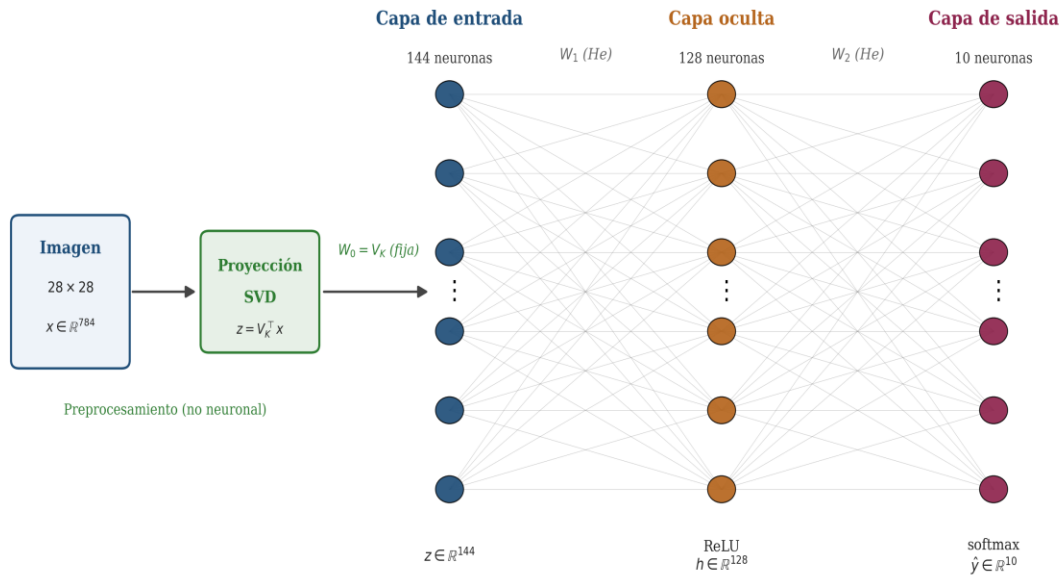


Figura 4: Grafos del perceptrón multicapa con parámetros entrenables. Únicamente las capas neuronales se representan como columnas de nodos: la capa de entrada ($z \in \mathbb{R}^{144}$), la capa oculta (128 unidades con ReLU) y la capa de salida (10 unidades con softmax). La imagen y la proyección por SVD no constituyen capas de neuronas, sino una etapa de preprocesamiento (bloques presentados al lado izquierdo) que produce el vector z mediante la transformación fija $W_0 = V_K$. Por claridad se considera dibujar un subconjunto representativo de neuronas por capa (los puntos suspensivos indican las restantes); W_1 y W_2 se inician aplicando el esquema de He.

Capa de proyección L_0 (reducción por SVD)

La primera capa es una transformación lineal sin sesgo cuyos pesos coinciden con el proyector singular, $W_0 = V_K \in \mathbb{R}^{784 \times K}$. Dada una imagen vectorizada $x \in \mathbb{R}^{784}$, produce el vector de características

$$z = W_0^\top x = V_K^\top x \in \mathbb{R}^K, \quad (5)$$

con $K = 144$. A diferencia de una capa densa convencional, W_0 no se inicializa de forma aleatoria, sino con la SVD *exacta* de la Ec. (1); al ser ortonormales sus columnas, la capa realiza un cambio de base hacia el subespacio de máxima varianza del conjunto de entrenamiento. Para la configuración primaria de la arquitectura neuronal la matriz de pesos sinápticos W_0 permanece fija durante el proceso de entrenamiento y se comporta como un extractor de características determinista e interpretable.

Capa oculta L_1 (transformación afín y ReLU)

La segunda capa es una transformación afín seguida de una no linealidad rectificada. A partir de z calcula

$$a^{(1)} = W_1^\top z + b_1, \quad W_1 \in \mathbb{R}^{K \times H}, \quad b_1 \in \mathbb{R}^H, \quad (6)$$

$$h = \text{ReLU}(a^{(1)}) = \max(0, a^{(1)}) \in \mathbb{R}^H, \quad (7)$$

con $H = 128$ unidades. La función de activación ReLU es considerada debido a su gradiente constante en el semieje positivo a diferencia de la función sigmoideal y logra acelerar la convergencia (Du et al., 2022). Los pesos sinápticos de la matriz W_1 se inicializan considerando el esquema propuesto por He et al. (2015), $W_1 \sim \mathcal{N}(0, 2/K)$, apropiado para funciones de activación rectificadas.

Capa de salida L_2 (softmax)

La tercera capa proyecta la representación oculta sobre las diez clases y la normaliza a una distribución de probabilidad:

$$a^{(2)} = W_2^\top h + b_2, \quad W_2 \in \mathbb{R}^{H \times 10}, \quad b_2 \in \mathbb{R}^{10}, \quad (8)$$

$$\hat{y}_i = \frac{\exp(a_i^{(2)})}{\sum_{c=1}^{10} \exp(a_c^{(2)})}, \quad i \in \{1, \dots, 10\}. \quad (9)$$

La composición de las tres capas equivale a

$$\hat{y} = \text{softmax}(W_2^\top \text{ReLU}(W_1^\top V_K^\top x + b_1) + b_2), \quad (10)$$

que hace explícita la naturaleza estrictamente prealimentada de la red.

Función de pérdida

Durante el proceso de entrenamiento se busca minimizar la entropía cruzada de manera categórica, entre la etiqueta verdadera, codificada en *one-hot* como t , y la distribución predicha $\hat{y}$, considerando:

$$\mathcal{L}(\hat{y}, t) = - \sum_{c=1}^{10} t_c \log \hat{y}_c. \quad (11)$$

Conteo de parámetros

En la Tabla 2 se presentan las dimensiones y los valores de los parámetros de cada capa. Al mantener fija la capa de proyección L_0 , el número de parámetros entrenables se reduce a 19,850,

frente a los 101,770 de un MLP equivalente que recibiera los 784 píxeles sin reducir: una disminución de un factor de 5.13. La reducción de parámetros responde directamente al diseño: la proyección singular toma a su cargo la extracción de características, y con ello una sola capa oculta resulta suficiente para la clasificación.

Tabla 2: Dimensiones y parámetros de la arquitectura ($K = 144$, $H = 128$). Los parámetros de L_0 no son entrenables cuando la proyección permanece fija.

Capa	Operación	Salida	Parámetros
L_0 proyección	$z = V_K^\top x$	144	112,896 (fijos)
L_1 oculta	$\text{ReLU}(W_1^\top z + b_1)$	128	18,560
L_2 salida	$\text{softmax}(W_2^\top h + b_2)$	10	1,290
Entrenables (con L_0 fija)			19,850
MLP de referencia sin SVD ($784 \rightarrow 128 \rightarrow 10$)			101,770

Métricas de evaluación

Además de la exactitud global; la proporción de ejemplos correctamente clasificados, se reportan métricas por clase. Para una clase c , a partir de los verdaderos positivos TP_c , los falsos positivos FP_c y los falsos negativos FN_c , se definen la precisión y la exhaustividad (*recall*):

$$\text{Prec}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \quad \text{Rec}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \quad (12)$$

y el valor de F_1 como su media armónica:

$$F_{1,c} = \frac{2 \text{Prec}_c \text{Rec}_c}{\text{Prec}_c + \text{Rec}_c}. \quad (13)$$

Promediar cada métrica por clase con igual peso, sin importar cuán frecuente sea cada una, es lo que resume el desempeño global; es la estrategia conocida como promedio macro.

Configuración experimental

Las decisiones de implementación que sustentan la reproducibilidad de los resultados se detallan a continuación. El modelo se entrenó con el optimizador Adam (Kingma & Ba, 2015), que adapta la tasa de aprendizaje de cada parámetro mediante estimaciones de los momentos de primer y segundo orden, con una tasa base $\eta = 10^{-3}$ y los coeficientes $\beta_1 = 0.9$ y $\beta_2 = 0.999$. La función objetivo es la entropía cruzada categórica de la Ec. (11). El entrenamiento se prolongó durante 200 épocas con un tamaño de lote de 32 ejemplares, y todos los pesos entrenables se inicializaron con el esquema de He. Para asegurar la reproducibilidad, se fijó una semilla aleatoria única para la inicialización de pesos y para el ordenamiento de los lotes. La Tabla 3 presenta los hiperparámetros y los valores obtenidos. La implementación se realizó directamente en NumPy, sin marcos de aprendizaje profundo, lo que permite un control explícito de la propagación y facilita la verificación numérica del gradiente (Sección 4.6).

Tabla 3: Valores de los hiperparámetros del entrenamiento.

Parámetro	Valor
Optimizador	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Tasa de aprendizaje	$\eta = 10^{-3}$
Tamaño de lote	32
Épocas	200
Capa oculta	128 unidades (ReLU)
Función de pérdida	entropía cruzada categórica
Inicialización	He (W_1, W_2); SVD exacta (W_0)
Semilla aleatoria	42

Resultados

Para presentar esta sección se considera el planteamiento de cinco ejes y la relación entre ellos. Para el primero, que corresponde a la Sección 4.1, se valora la fidelidad de la reconstrucción empleando los vectores de SVD en función de su rango de influencia. Para la segunda parte, que corresponde a la Sección 4.2, se revisa el estudio de ablación sobre K . En la Sección 4.3 se revisa la dinámica de convergencia. Luego avanzamos en la Sección 4.4 con reportar las métricas de clasificación por clase para finalmente, abordar en la Sección 4.5 la parte de caracterizar los errores mediante la matriz de confusión.

Fidelidad de la reconstrucción por SVD

La Figura 5 muestra, para cinco dígitos representativos, la imagen original y su reconstrucción $\hat{X} = ZV_K^T$ para tres rangos de truncamiento. Con $K = 16$ la reconstrucción retiene solo la silueta global y presenta un error relativo de Frobenius del 47.1 %; con $K = 64$ el error desciende al 19.0 %; y con $K = 144$ es de apenas el 3.9 %, de modo que las reconstrucciones resultan visualmente indistinguibles de las originales. Esta progresión confirma de forma directa la implicación de la Ec. (2): un número reducido de componentes singulares preserva la estructura esencial de los dígitos.

Reconstrucción por SVD truncada al aumentar el rango K

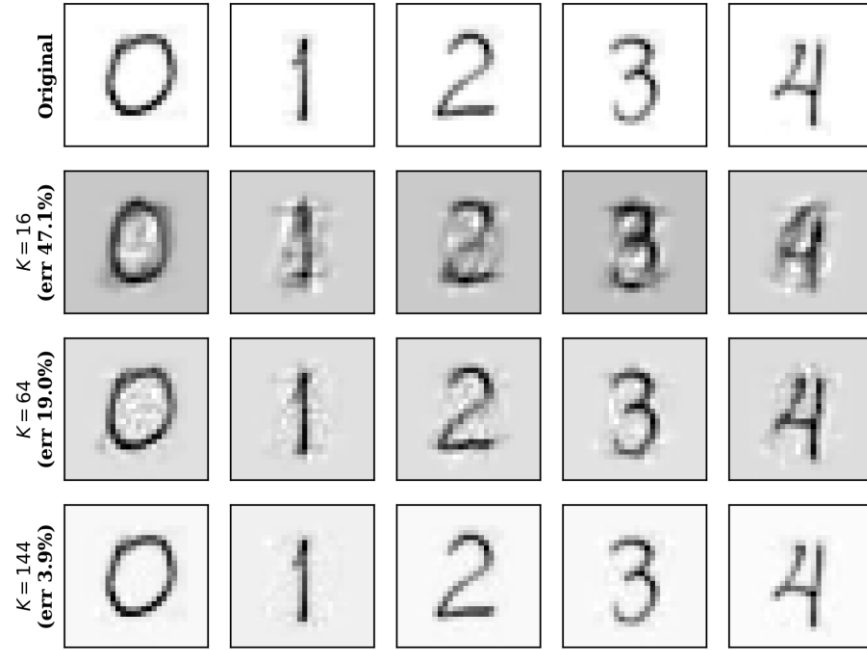


Figura 5: Ejemplo de reconstrucción de cinco dígitos de entrenamiento para $K \in \{16, 64, 144\}$. La fila superior presenta las versiones originales de las imágenes, las siguientes filas se encargan de presentar las diferentes reconstrucciones, el error relativo de Frobenius disminuye del 47.1 % al 3.9 % al considerar un mayor número de componentes.

Ablación del rango de truncamiento

La Tabla 4 compara cuatro configuraciones con SVD frente al modelo sin reducción, y la Figura 6 representa conjuntamente la exactitud de prueba y la energía retenida en función de K . La reducción dimensional a un intervalo comprendido entre el 8.2 % y el 18.4 % del espacio original no degrada la exactitud; por el contrario, las configuraciones $K \in \{64, 100\}$ alcanzan una exactitud del 93.3 %, superior al 91.7 % del modelo de referencia, y reducen los parámetros entrenables en un orden de magnitud. La pérdida de prueba decrece de forma monótona al aumentar el rango, desde 0.2387 en el modelo de referencia hasta 0.1692 con $K = 144$. Dado que la exactitud se satura por la granularidad del conjunto de prueba, la pérdida constituye un criterio de comparación más fino.

Tabla 4: Ablación del rango de truncamiento K frente al modelo sin SVD. En negritas, el mejor valor por columna.

Configuración	Dim.	Energía (%)	Acc. prueba	Pérdida prueba
Baseline (sin SVD)	784	–	0.9167	0.2387
SVD $K = 64$	64	96.39	0.9333	0.2157
SVD $K = 100$	100	99.01	0.9333	0.1836
SVD $K = 144$	144	99.85	0.9167	0.1692

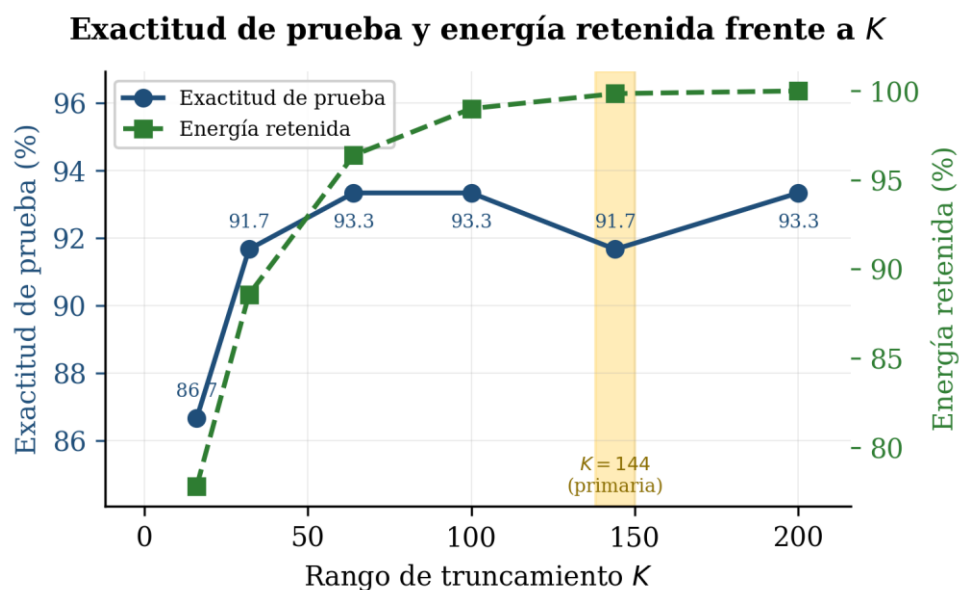


Figura 6: Exactitud de prueba (azul, eje izquierdo) y energía retenida (verde, eje derecho) en función del rango de truncamiento K . La franja resalta la configuración primaria $K = 144$.

Dinámica de convergencia del entrenamiento

La Figura 7 contrasta la evolución de la pérdida y de la exactitud durante las 200 épocas, para el modelo con SVD ($K = 144$) y el modelo sin reducción. El panel (a) muestra que ambos modelos alcanzan una pérdida de entrenamiento prácticamente nula, pero el modelo con SVD converge a una pérdida de prueba menor (0.169 frente a 0.239), señal de una mejor capacidad de generalización. El panel (b) muestra que la exactitud de prueba se estabiliza de manera temprana y se mantiene constante, sin signos de degradación por sobreajuste. La proyección singular, al proporcionar un punto inicial informativo y un espacio de características de menor dimensión, contribuye a esta dinámica de aprendizaje estable.

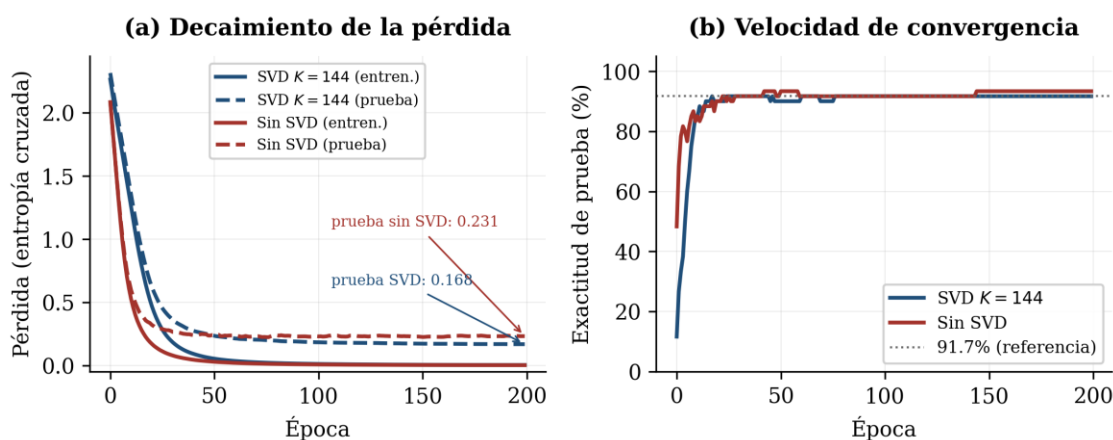


Figura 7: (a) Decaimiento de la pérdida de entrenamiento y de prueba; (b) evolución de la exactitud de prueba. Se comparan el modelo con SVD ($K = 144$) y el modelo sin reducción; se anotan los valores finales de pérdida.

Métricas de clasificación por clase

La Tabla 5 reporta la precisión, la exhaustividad y el puntaje F_1 de cada clase, junto con sus promedios macro (Ecs. (12)–(13)). El modelo alcanza un puntaje F_1 macro de 0.915, con una precisión macro de 0.916 y una exhaustividad macro de 0.917; la proximidad de estos tres valores indica un comportamiento equilibrado entre clases. La Figura 8 representa el puntaje F_1 por clase: la mayoría de los dígitos se reconocen con un F_1 igual o superior a 0.9, y solo las clases con confusiones morfológicas —analizadas en la Sección 4.5— presentan valores algo menores.

Tabla 5: Métricas de clasificación por clase sobre el conjunto de prueba (modelo SVD $K = 144$). Se reportan precisión, exhaustividad (recall) y puntaje F_1 .

Clase	Precisión	Exhaustividad	F_1	Soporte
0	1.000	1.000	1.000	6
1	0.833	0.833	0.833	6
2	1.000	1.000	1.000	6
3	0.800	0.667	0.727	6
4	1.000	1.000	1.000	6
5	0.833	0.833	0.833	6
6	1.000	1.000	1.000	6
7	0.833	0.833	0.833	6
8	0.857	1.000	0.923	6
9	1.000	1.000	1.000	6
Macro	0.916	0.917	0.915	60

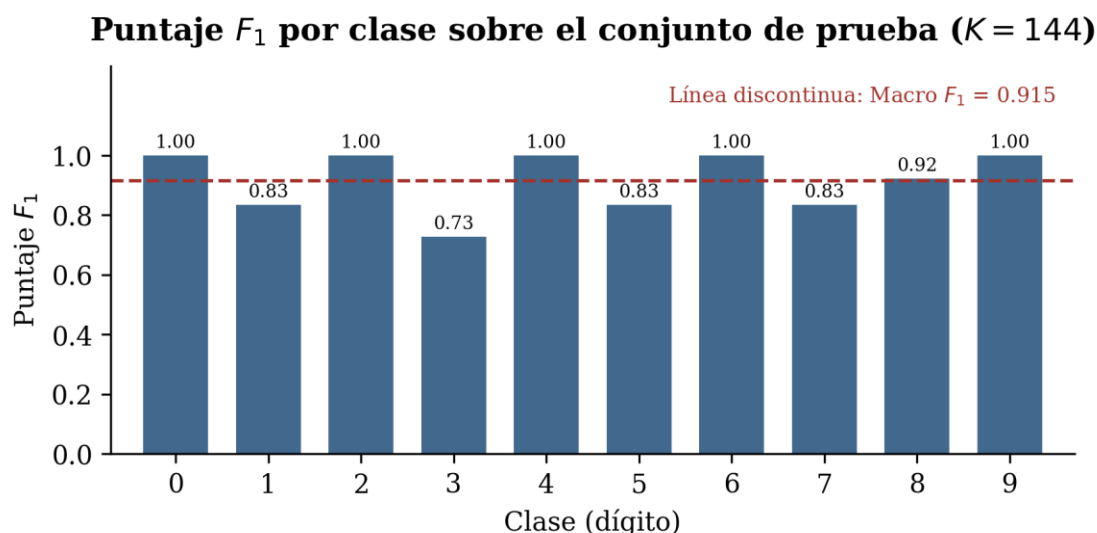


Figura 8: Puntaje F_1 por clase sobre el conjunto de prueba ($K = 144$). La línea discontinua indica el promedio macro (0.915).

Análisis de errores mediante la matriz de confusión

La Figura 9 contiene la matriz de confusión obtenida al evaluar la base de datos de prueba y se tiene una versión normalizada por filas. La exactitud global de 91.7 % puede ser confirmada al revisar los valores numéricos sobre la diagonal principal. Los errores observados son pocos y después de analizarlos podemos decir que no son aleatorios, sino que se concentran en pares de clases de morfología similar, es decir, se registran confusiones entre el dígito 3 y los dígitos 5 y 8, así como entre el 1 y el 7. Por tanto, estas confusiones son plausibles desde el punto de vista perceptual, pues dichos dígitos comparten trazos curvos o verticales que, a la resolución de 28×28 píxeles pueden ser confusos.

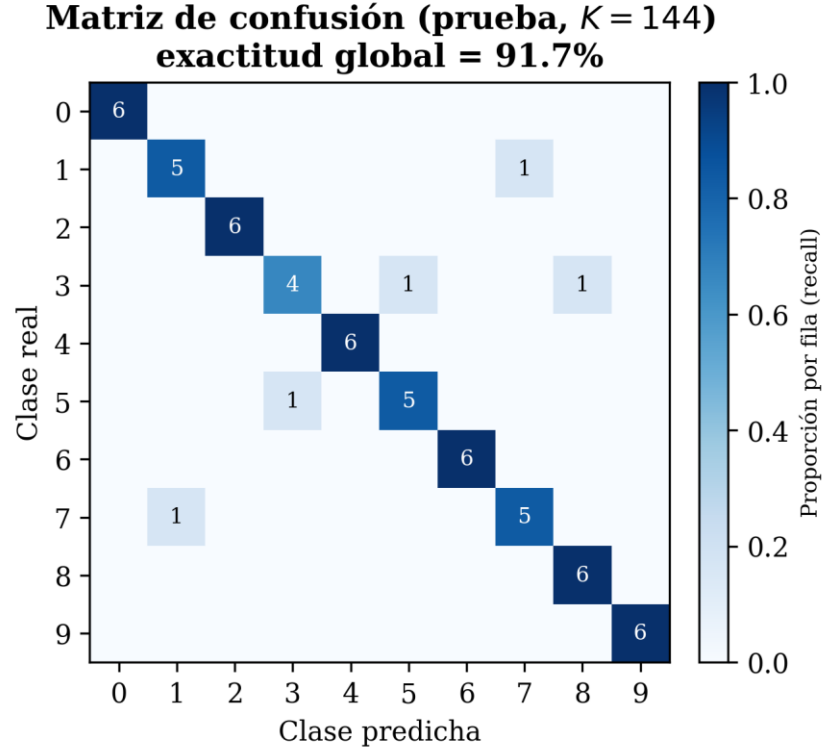


Figura 9: Matriz de confusión al evaluar la base de datos de prueba, considerando para SVD la cantidad de $K = 144$ vectores.

Verificación

Como base del análisis se considera la corrección de las formulaciones y el seguimiento riguroso del protocolo experimental; en ambos casos se aplicó una verificación numérica y el proceso puede ser resumido siguiendo cuatro puntos. En primer lugar, el residual de la aproximación truncada coincide con el valor teórico de la Ec. (2), con una diferencia del orden de 10^{-15} , atribuible al redondeo de punto flotante. En segundo lugar, la base V_K resultó ortonormal hasta la precisión de máquina: $\|V_K^T V_K - I\| \approx 2.5 \times 10^{-14}$. En tercer lugar, la corrección de la retropropagación se validó comparando el gradiente analítico con la aproximación numérica por diferencias centradas, $\partial \mathcal{L} / \partial w \approx [\mathcal{L}(w + \epsilon) - \mathcal{L}(w - \epsilon)] / (2\epsilon)$, con un error máximo de 2.0×10^{-11} , varios órdenes de magnitud por debajo del umbral habitual de 10^{-4} . Como último punto, la ausencia de transferencia indebida de información se corroboró mediante una contraprueba: al incluir la base de imágenes de prueba en el cálculo de la descomposición, el conjunto de vectores singulares cambia de manera apreciable, $\|V_{tr} - V_{tr+te}\| = 9.81$, lo que demuestra que ajustar únicamente sobre el entrenamiento es necesario y efectivo para preservar la independencia de la evaluación.

Discusión

Después de aplicar la SVD y recopilar los resultados del proceso, se logra confirmar que al usar un número predeterminado de componentes, funciona como un extractor de características eficiente, que prioriza las direcciones con mayor varianza, reduce la cantidad de datos a procesar; lo cual plantea la disminución del costo computacional y, al mismo tiempo, logra regularizar el aprendizaje. Esta doble función puede explicar el funcionamiento visto en las pruebas de

ablación, que nos indica que a mayor compresión de la entrada la exactitud mejora de forma constante y la pérdida de prueba disminuye. Por tanto, como punto de interés se expresa que la proyección de vectores singulares actúa como un filtro que elimina el ruido, un proceso que coincide con lo reportado por Lytvyn et al. (2024).

Partiendo del punto de vista de diseño, el comportamiento del proceso matemático expresado en el trabajo respalda la búsqueda de una complejidad reducida, además de plantear que con una sola capa oculta; precedida por una proyección lineal interpretable e inicializada de forma determinista, es suficiente para lograr un desempeño competitivo en la clasificación de los datos, de hecho, este enfoque requiere cinco veces menos parámetros entrenables que un MLP convencional. Además, la capa de proyección es transparente, ya que sus pesos pueden ser considerados como las *eigen-imágenes* del conjunto de datos; esto contrasta favorablemente con la opacidad típica de las representaciones en arquitecturas profundas.

Sin embargo y a pesar de estas ventajas, es importante tener en consideración y delimitar el alcance de los hallazgos a través de dos puntos clave. En primer lugar, el tamaño reducido del conjunto de prueba (con 60 ejemplares) otorga a la exactitud una resolución de 1.7 puntos porcentuales por muestra, lo que ha demostrado una mejora con respecto a la pérdida y el puntaje F_1 macro como métricas principales que cuantifican la comparación. En segundo lugar, los resultados presentados están basados en el tratamiento de dígitos manuscritos dentro de imágenes de baja resolución; por lo que es honesto describir que su generalización a dominios con mayor variabilidad intraclase requerirá de validaciones adicionales.

Conclusiones

Se ha desarrollado, formalizado y validado una metodología para el reconocimiento de dígitos escritos a mano que logra combinar la descomposición por valores singulares (SVD) y una arquitectura neuronal multicapa, logrando niveles de desempeño que compiten con los de arquitecturas profundas con una complejidad mucho menor. La metodología planteada ha logrado la compresión de los vectores en un rango de entre el 8.2 % y el 18.4 %, y aun así ha logrado conservar; e incluso mejorar la exactitud, con una pérdida de prueba que disminuye de manera monótona con el paso de las épocas de entrenamiento. Los resultados se han obtenido siguiendo un protocolo de evaluación riguroso; sin fuga de información entre las particiones de datos y verificando numéricamente la retropropagación.

La capa de proyección fue inicializada de manera determinista y construida a partir de la base de vectores singulares, lo cual ha favorecido a una convergencia estable y ha logrado reducir en un factor de 5.13 la cantidad de parámetros a entrenar comparado con un MLP convencional. Con apenas 19,850 parámetros, el modelo ha alcanzado una exactitud del 91.7 % y un puntaje F_1 macro de 0.915, con una base de datos balanceados para cada clase evitando que alguna logre dominar el resultado por su frecuencia. Esto confirma que el método propuesto logra trabajar en escenarios con recursos limitados.

Como perspectivas a tareas por desarrollar se considera la validación cruzada estratificada, con el objetivo de mitigar el efecto del tamaño del conjunto de prueba; el propio uso de técnicas de aumento de datos y el ajuste meticuloso de la capa de proyección, considerando la inclusión de un tratamiento bayesiano de la incertidumbre (Hubin & Storvik, 2024).

Declaración de uso de IA

Herramienta utilizada: Claude Anthropic- Sonnet 4.6 julio de 2026

Descripción del uso: Se utilizó para la lluvia inicial de ideas, organización de conceptos teóricos y el apoyo en el diseño de los diagramas que se ilustran en las Figuras 1 y 4.

En concordancia con las políticas de ReCIBE, los autores ratifican que el uso de estas herramientas se limitó a tareas de asistencia técnica o editorial. La concepción del estudio, el desarrollo del método, la ejecución de los experimentos, la interpretación de los resultados y la redacción final fueron realizadas en su totalidad por seres humanos, quienes asumen la responsabilidad ética y legal del contenido publicado.

Referencias

- Abaza, B. F., & Gheorghita, V. (2025). Artificial neural network framework for hybrid control and monitoring in turning operations. *Applied Sciences*, 15(7), Article 3499. <https://doi.org/10.3390/app15073499>
- Chen, H., Wang, Y., Guo, J., & Tao, D. (2023). *VanillaNet: The power of minimalism in deep learning* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2305.12972>
- Du, K.-L., Leung, C.-S., Mow, W. H., & Swamy, M. N. S. (2022). Perceptron: Learning, generalization, model selection, fault tolerance, and role in the deep learning era. *Mathematics*, 10(24), Article 4730. <https://doi.org/10.3390/math10244730>
- Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3), 211–218. <https://doi.org/10.1007/BF02288367>
- Galis, F., Onchis, D. M., & Istin, C. (2025). Refining filter global feature weighting for fully unsupervised clustering. *Applied Sciences*, 15(16), Article 9072. <https://doi.org/10.3390/app15169072>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <https://www.deeplearningbook.org>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 1026–1034). <https://doi.org/10.1109/ICCV.2015.123>
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507. <https://doi.org/10.1126/science.1127647>
- Huang, Y., Lu, J., & Xu, Y. (2020). *Image classification using multilayer perceptron* [Unpublished course report]. University of California San Diego. https://jingpeilu.github.io/doc/253HW2_Report.pdf
- Hubin, A., & Storvik, G. (2024). Sparse Bayesian neural networks: Bridging model and parameter uncertainty through scalable variational inference. *Mathematics*, 12(6), Article 788. <https://doi.org/10.3390/math12060788>
- Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065), Article 20150202. <https://doi.org/10.1098/rsta.2015.0202>
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.1412.6980>
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- Lytvyn, V., Peleshchak, I., Peleshchak, R., Shakleina, I., Mozol, N., & Svysch, D. (2024). Object image recognition using multilayer perceptron combined with singular value decomposition. In *Proceedings of the Modern Machine Learning Technologies Workshop (MoMLet 2024)* (pp. 225–234). CEUR-WS. <https://ceur-ws.org/Vol-3711/paper13.pdf>
- Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71–86. <https://doi.org/10.1162/jocn.1991.3.1.71>



Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.