Autoencoder feed-forward sobre dataset Fashion-MNIST

Borgnino, L.* and Salomón, N.†
Fundación Fulgor, Ernesto Romagosa 518, Córdoba, Argentina and
Universidad Nacional del Sur, Bahía Blanca, Argentina

Spinazzola, M.T.‡
Facultad de Ciencias Exactas y Naturales (FCEyN) de la Universidad de Buenos Aires (UBA), Buenos Aires, Argentina
(Dated: December 5, 2023)

El presente trabajo presenta una implementación simple, pero funcional, de una red feed-forward de tipo autoencoder empleando capas neuronales completamente conectadas. El análisis se realiza sobre el dataset Fashion-MNIST, compuesto por distintas prendas de vestir. Se evaluan diferentes arquitecturas, funciones de activación y de error, así como también distintos optimizadores, para obtener el mejor desempeño posible. Nuestros resultados experimentales demuestran la efectividad del autoencoder propuesto al reconstruir fielmente imágenes del dataset Fashion-MNIST. El código de desarrollo se encuentra disponible en `https://colab.research.google.com/drive/1M9w2Lnj3dNM1ADOJVey7ijGN2m9xX5Do?usp=drive_link`.

I. INTRODUCCIÓN

La creciente complejidad de los conjuntos de datos en el campo de la visión por computadora ha impulsado el desarrollo de técnicas avanzadas para la representación eficiente de información visual. Entre estas técnicas, los autoencoders han emergido como herramientas poderosas capaces de aprender representaciones latentes significativas de datos. En este contexto, el conjunto de datos Fashion-MNIST [1, 2] se ha convertido en un banco de pruebas esencial para evaluar el rendimiento de modelos de aprendizaje automático en la tarea de reconocimiento de prendas de moda.

Este trabajo se sumerge en la aplicación de autoencoders, específicamente de una arquitectura neuronal completamente conectada, para abordar el desafío de reconstruir imágenes y extraer características intrínsecas del conjunto de datos Fashion-MNIST. La complejidad y diversidad de prendas presentes en este conjunto de datos proporcionan un escenario ideal para evaluar la capacidad de los autoencoders para capturar características de la imágen, codificarlas y decodificarlas en nuevas imágenes.

II. DESARROLLO

El proceso de entrenamiento de una red neuronal feed-forward puede dividirse en pasos perfectamente definidos.

- **Recopilación y preparación del Dataset:** El entrenamiento de una red neuronal supervisada requiere de datos de entrada con un formato específico acorde a cada tipo de red.

- **Creación de la arquitectura e inicialización de hiperparámetros:** Se crea la arquitectura a analizar y se fijan valores iniciales para los hiperparámetros y los pesos de la red.

- **Establecimiento de la función de error y el optimizador a emplear:** Se fija la función que se empleará para cuantificar la diferencia entre la salida predicha por la red y la salida esperada. Adicionalmente se establece la función encargada de optimizar los parámetros de la red, considerando el error calculado previamente.

- **Entrenamiento:** Proceso iterativo, por medio del uso de batches (o lotes) de datos de entrada, para calcular el error y realizar el backpropagation de los gradientes para actualizar los pesos de la red.

- **Validación:** Empleando un conjunto de datos de validación, se evalua el rendimiento de la red en datos no vistos durante el entrenamiento y se modifican hiperparámetros, de ser necesario, para mejorar el rendimiento.

- **Prueba:** Finalmente, la red entrenada se evalúa en un conjunto de datos de prueba independiente para medir su rendimiento general en datos no vistos.

A. Dataset: Fashion-MNIST

El conjunto de datos Fashion-MNIST es una gran base de datos de imágenes de moda disponible gratuitamente que se utiliza comúnmente para entrenar y probar varios sistemas de aprendizaje automático.[1, 2]. Originalmente Fashion-MNIST estaba destinado a servir como reemplazo de la base de datos MNIST original [3] para comparar algoritmos de aprendizaje automático, ya que comparte el mismo tamaño de imagen, formato de datos y la estructura de divisiones de entrenamiento y prueba.

El conjunto de datos contiene 70.000 imágenes en escala de grises de 28x28 píxeles de prendas de vestir, repartidas en 10 categorías, con 7.000 imágenes por categoría (ver Tabla I). Algunos ejemplos de los distintos tipos de prendas pueden apreciarse en la Figura 1. El conjunto de entrenamiento consta de 60.000 imágenes y el conjunto de prueba consta de 10.000 imágenes. El conjunto de datos suele incluirse en bibliotecas estándar de aprendizaje automático.

Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

TABLE I. Clases del Dataset Fashion-MNIST

FIG. 1. Ejemplos de prendas del Dataset Fashion-MNIST

B. Arquitectura: Autoencoder Feed-forward

La arquitectura elegida para resolver el presente problema se basa en un autoencoder. El mismo consta de dos partes principales: un codificador que reduce la dimensionalidad de los datos de entrada a través de capas ocultas, resultando en una etapa intermedia conocidad como espacio latente; y un decodificador que reconstruye los datos originales a partir de la representación comprimida generada por el codificador, ver Figura 4. Matemáticamente:

$$v = h(x) \tag{1}$$
$$y = f(v) = f(h(x)) \tag{2}$$

Donde:

- x es la entrada al Autoencoder.
- v es la salida del Codificador (espacio latente).
- y es la salida del Autoencoder.
- h y f se corresponden con las funciones aplicadas por el Codificador y Decodificador respectivamente.

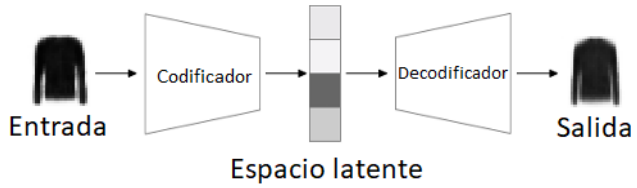

FIG. 2. Arquitectura general de una red Autoencoder

En nuestros experimentos se emplearán redes feed-forward con una capa oculta, con un número variable de neuronas (entre 64 a 512 neuronas), mientras que las entradas y salidas estarán compuestas por 784 neuronas, debido a las dimensiones de las imágenes del dataset serializadas ($28*28 = 784$). Para la capa oculta se empleó la función de activación *Leaky ReLU*, mientras que para la capa de salida se empleó la función *Sigmoide*, con el fin de contemplar los valores negativos obtenidos para los píxeles, y llevarlos al rango adecuado para las imágenes (0.0 - 1.0).

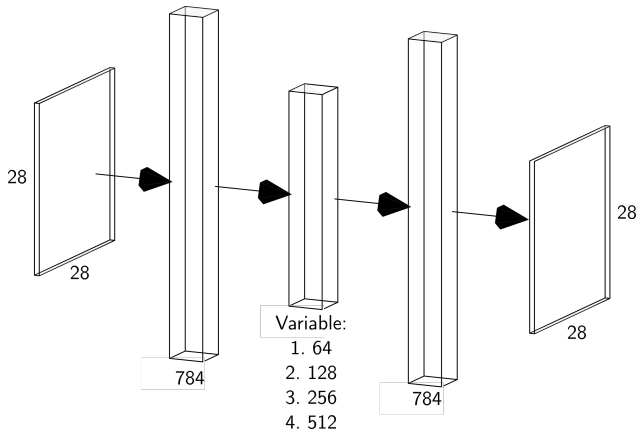

FIG. 3. Arquitectura Autoencoder utilizada

C. Entrenamiento y Validación del Modelo

El sistema utilizado para el entrenamiento consta del modelo de autoencoder a optimizar que toma como entrada cada una de las imágenes del set de datos de entrenamiento. El modelo realizará una predicción de la imagen que se comparará con la imagen de entrada original mediante la métrica de Error Cuadrático Medio (MSE). De acuerdo al error entre la imagen original y la imagen recostruida a partir del espacio latente, a través de gradiente descendente estocástico, se actualizarán los pesos del modelo. Se llevará un registro de cómo evoluciona la pérdida de entrenamiento y validación para determinar si hay overfitting y el número óptimo de épocas.

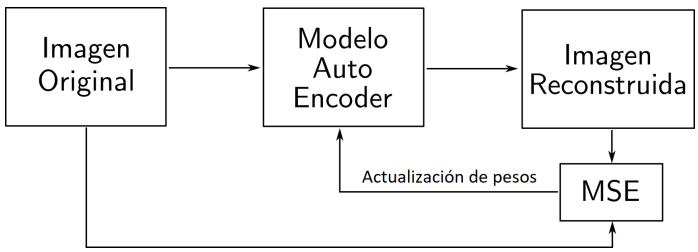

FIG. 4. Flujo de entrenamiento del modelo

Para validar el modelo se utilizará un grupo de datos reducido, lo cuales que no han sido provistos al modelo en la etapa de entrenamiento, y se calculará el error entre las predicciones y las imágenes originales.

III. RESULTADOS

A. Fase de entrenamiento y validación

Se realizó el entrenamiento de los cuatro modelos de autoencoder (cada uno con un número diferente de capas ocultas) y se evaluó la pérdida de entrenamiento y validación de cada uno de ellos.

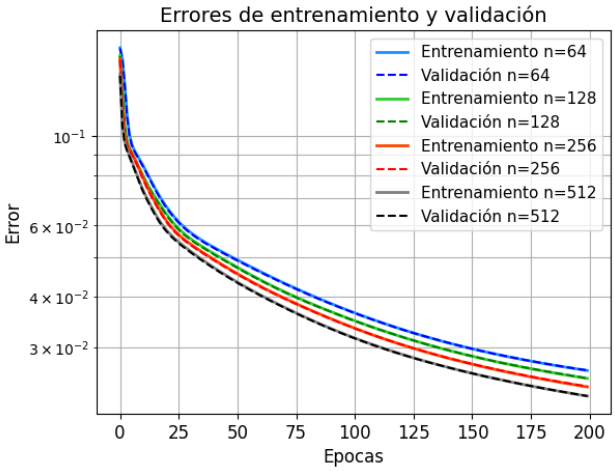

FIG. 5. Pérdida de entrenamiento y validación de cada modelo respecto a la época. Escala semilogarítmica.

Tanto para el error de entrenamiento cómo para el error de validación (Figura 5) puede observarse que aumentar el numero de neuronas en la capa oculta reduce el error. En las distintas pruebas realizadas no se encontró una cota superior para el numero de épocas a partir de la cual modelo empieza a fallar. Por lo tanto se determinó que 200 épocas fue un valor apropiado a partir del cual aumentar las iteraciones no mejora significativamente el rendimiento de la red.

Por otro lado, una forma de corroborar el correcto funcionamiento de la red es confirmar que la red logre reconstruir adecuadamente las imágenes. En la Figura 6 puede observarse cómo al aumentar la época de entrenamiento en la que se encuentra el modelo, se van refinando las predicciones de las imágenes.

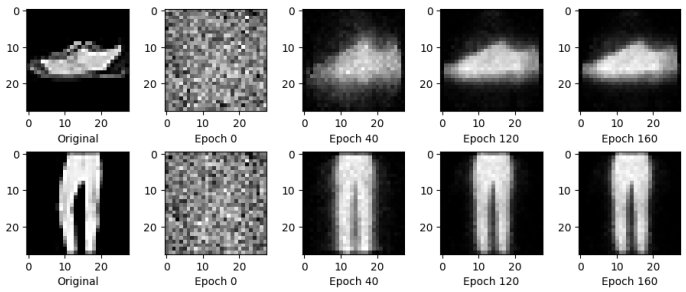

FIG. 6. Predicción de una imagen respecto a la época

B. Fase de prueba

Respecto al conjunto de prueba de los modelos, se muestra que, al igual que durante el entrenamiento y validación, aumentar el numero de neuronas en la capa oculta se traduce a una reducción en el error. Puede verse que este error es del mismo orden que los errores de prueba y validación, lo que indica que el entrenamiento que recibió la red (numero de épocas) fue adecuado.

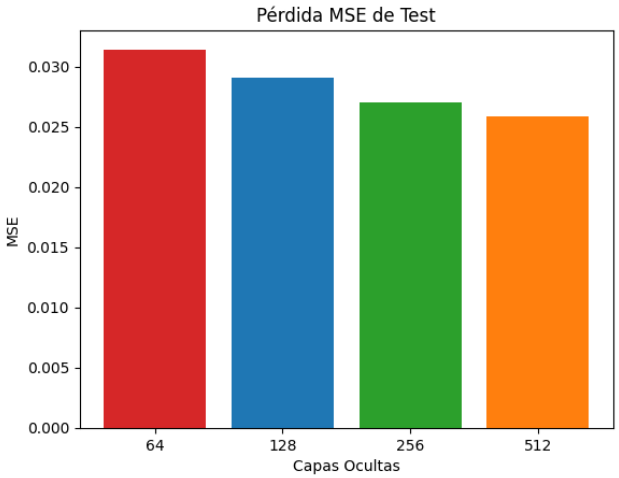

FIG. 7. Comparación de la pérdida de Test entre los diferentes modelos

IV. CONCLUSIONES

En conclusión, en este trabajo se diseñó una red Feed-forward tipo autoencoder para codificar y decodificar imágenes pertenecientes a 10 categorías de ropa. Utilizando la función error cuadrático medio, distintas funciones de activación y el método de descenso por el gradiente, fue posible entrenar 4 versiones de esta red con un número creciente de neuronas en la capa oculta (64, 128, 256 y 512), obteniendo como resultado una mejora progresiva en el rendimiento de la red. Además, se pudo verificar el proceso de aprendizaje de la red a través de muestras de imágenes intermedias durante el ciclo de entrenamiento, partiendo de imágenes completamente "ruidosas", hasta llegar a representaciones similares a las imágenes de entrada.

* leo.borgnino@gmail.com
† nicolassalomon96@gmail.com
‡ tomasspinazzola@gmail.com

[1] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017), arXiv:1708.07747 [cs.LG].
[2] Fashion-mnist dataset, `https://github.com/zalandoresearch/fashion-mnist`.
[3] Y. LeCun and C. Cortes, The mnist database of handwritten digits (2005).