Universidad Nacional de Córdoba
Facultad de Matemática, Astronomía, Física y Computación
Curso de Posgrado “Redes Neuronales”
Docentes: Dr. Francisco Tamarit, Dr. Juan Ignacio Perotti
Alumna: María Virginia Romero Messein
Trabajo Final Integrador

Redes Neuronales Convolucionales para la clasificación de imágenes

Introducción

Las redes neuronales convolucionales (CNN) son arquitecturas de aprendizaje profundo muy populares y se han convertido en un modelo de referencia para resolver principalmente problemas relacionados con imágenes. Una de las tareas más frecuentes es la clasificación de imágenes, tarea en la cual se diseñan modelos que imitan la corteza visual humana y su capacidad de categorización. Una CNN se caracteriza por realizar una secuencia de transformaciones de los datos disponibles con una estructura jerárquica. En la primera capa se encuentran los datos de entrada y luego cada capa de la red acomoda la información representando los datos de forma distinta. Se trata entonces, de una serie de representaciones hasta llegar a la representación más accesible para la tarea que se pretende realizar.

El objetivo de este trabajo es implementar una red neuronal feedforward convolucional para el aprendizaje de clasificación de imágenes. Se utilizará la base de datos Fashion- MNIST que contiene 60.000 imágenes en el conjunto de entrenamiento y 10.000 en el conjunto de testeo. Las imágenes son de de 28 * 28 píxeles en escala de grises y se dividen en 10 categorías de diversas prendas y calzados: 1-Camiseta (T-shirt), 2-Pantalón (Trouser), 3-Pullover (Pullover), 4- Vestido (Dress), 5-Campera (Coat), 6-Sandalia (Sandal), 7-Camisa (Shirt), 8-Zapatilla (Sneaker), 9-Bolso (Bag), 10-Bota (Ankle boot). A partir de la exploración de los hiperparámetros se diseñaron varias arquitecturas, se evaluaron sus desempeños y se seleccionó la arquitectura mejor comportada.

El trabajo consta de tres partes: en la primera, se exponen los conceptos más importantes presentes en las CNN. En la segunda, se presentan los diferentes modelos explorados y se evalúan sus desempeños. En la tercera, se establece una comparación entre el desempeño de una red neuronal convolucional y una red neuronal simple con una sola capa oculta. Por último, se exponen las reflexiones finales.

Conceptos preliminares

La característica principal de las CNN es que las distintas capas permiten la representación de diversas características de los datos. Estas representaciones se realizan a través del proceso de convolución. La convolución es una operación matemática cuyo objetivo es extraer características de alto nivel de una imagen. Para esta tarea se utilizan filtros que son los encargados de capturar determinados aspectos o algo de información de una imagen y a su vez cada filtro aprende cosas distintas.

Como se puede visualizar en la figura 1, en el proceso de convolución se desliza el filtro sobre la imagen de entrada. En cada ubicación se multiplica el tamaño de la imagen por el tamaño del filtro y se suman ambos resultados. Esta suma entra en un mapa de características. A medida que el filtro se va corriendo por toda la imagen los resultados se van ubicando en el mapa de características hasta completarse.

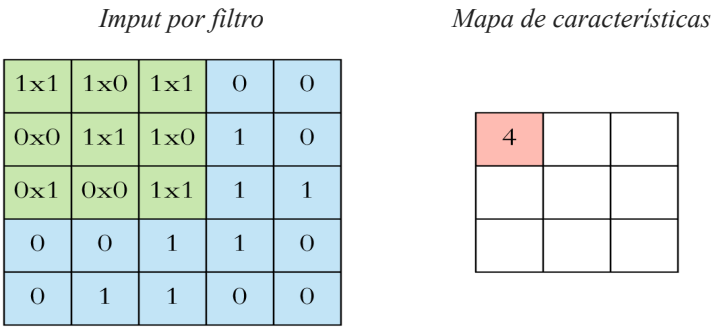

Figura 1

Además, las CNN utilizan el Pooling (subsampling) para reducir la imagen y agilizar el proceso de entrenamiento. El tipo que se utilizará en este trabajo es el max pooling, mediante el cual la imagen es dividida en regiones del mismo tamaño y para cada región se extrae el valor máximo que corresponderá a un píxel en la imagen resultante.

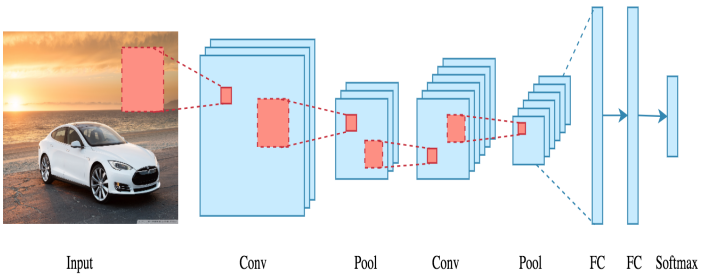

Figura 2

Para sintetizar, el funcionamiento de una CNN, tal como se observa en la figura 2, consta de: una primera capa de entrada con el conjunto de datos de entrenamiento; luego sigue una primera capa

convolucional con sus respectivos filtros y un max pooling, que generarán un mapa de representaciones; después, sigue una segunda capa de convolución con sus respectivos filtros y max pooling que generarán otro mapa de representaciones. Estas representaciones se envían a un perceptrón multicapa el cual se conecta a una capa de salida. Cabe aclarar que para complejizar los modelos de CNN se pueden incrementar el número de capas convolucionales y la cantidad de capas ocultas.

Además, en las CNN se encuentran variables llamadas hiperparámetros que son las encargadas de regir el proceso de entrenamiento. A través del ajuste de los mismos se intenta llegar a valores mínimos de pérdida puesto que es prácticamente imposible llegar a un mínimo absoluto. Lo que interesa es la capacidad de generalización, es decir, la capacidad de la red para aprender cosas nuevas y evitar el overfitting. Cuando se da el overfitting significa que la red memorizó las imágenes y no adquirió la capacidad de predicción. Para evitar este fenómeno se le presentan al modelo las 10.000 imágenes de testeo, las cuales no ha mirado en la etapa de entrenamiento. Con este conjunto de datos reservados exclusivamente para el testeo se corrobora que el modelo efectivamente está aprendiendo.

Los hiperparámetros son los siguientes:

Learning rate (tasa de aprendizaje): determina cuán rápido convergerá la red neuronal a los mínimos.

Número de épocas: determina cuánto tiempo andará el algoritmo de entrenamiento. Indica el número de veces que se va a iterar sobre el conjunto de datos. Cada época consta de dos partes: el train loop, que itera sobre el conjunto de datos de entrenamiento e intenta converger hacia los mínimos; y el validation/test loop, que itera sobre el conjunto de datos de testeo.

Batch size (tamaño del lote): es el número de iteraciones necesarias para completar una época. En cada iteración se actualizan los pesos del modelo. El número de iteraciones es el número total de muestras de entrenamiento presentes en un solo lote.

Dropout: es un método que desactiva el número de neuronas de forma aleatoria. En cada iteración de la red neuronal el dropout desactiva diversas neuronas para obligar a las neuronas cercanas a no depender de las demás y trabajar de manera independiente. El dropout contribuye a evitar el overfitting.

Arquitecturas posibles

Para las CNN se utilizó la función de activación ReLu, la función Cross Entropy Loss para evaluar el error, el Accuracy como métrica para calcular la precisión sobre un conjunto de datos y el optimizador Adam.

Las arquitecturas exploradas tienen una estructura similar pero con la modificación de los siguientes hiperparámetros: el *learning rate,* el *batch* y el *número de épocas*.

Se implementa una red con 784 unidades de entrada; 2 capas convolucionales; 2 capas ocultas, la primera 600 neuronas y la segunda de 120 neuronas; y una capa de salida de 10 neuronas.

En la primera capa convolucional se utilizan 32 filtros de tamaño 3*3 y en la segunda capa convolucional se utilizan 64 filtros de tamaño 3*3.

Primer modelo: *learning rate*=0,001; *dropout*=0,25; *batch*=64; *número de épocas*=5.

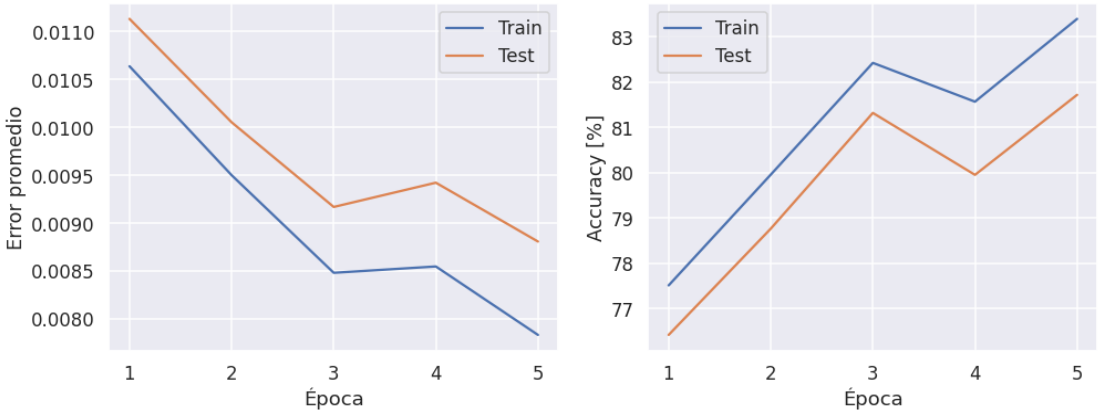

Figura 3: Error promedio y accuracy para train y test

En la figura 3 se observa que las loss de entrenamiento y de testeo descienden con más énfasis desde la época 1 hasta la época 3. En las dos últimas épocas el decrecimiento es más paulatino. La curva de testeo es similar a la de entrenamiento lo cual indica la ausencia de overfitting y un buen desempeño del modelo. También, se visualiza que los valores de Accuracy llegan al 83% aproximadamente en la última época.

Segundo modelo: *learning rate*=0,00001; *dropout*=0,25; *batch*=1000; *número de épocas*= 30.

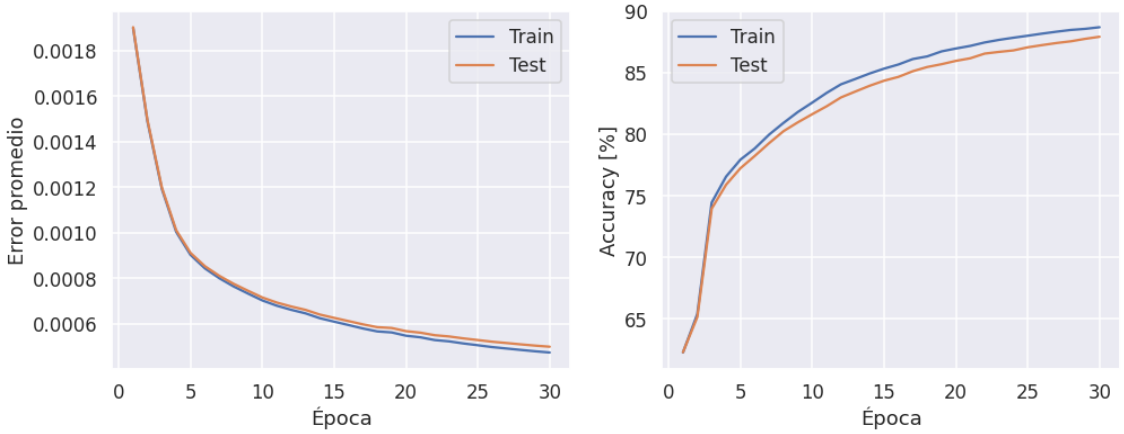

Figura 4: Error promedio y accuracy para train y test

La figura 4 demuestra que disminuyendo el learning rate y aumentando el tamaño del batch mejora el rendimiento del modelo. Las curvas descienden con mayor énfasis hasta la época 5. Luego el decaimiento es más paulatino. Ambas curvas presentan un recorrido similar por lo que no se detecta el riesgo de overfitting. Además, se visualiza que los valores del Accuracy están cerca del 87% en la época 30.

Tercer modelo: *learning rate*=0,00001; *dropout*=0,25; *batch*=1000; *número de épocas*= 100.

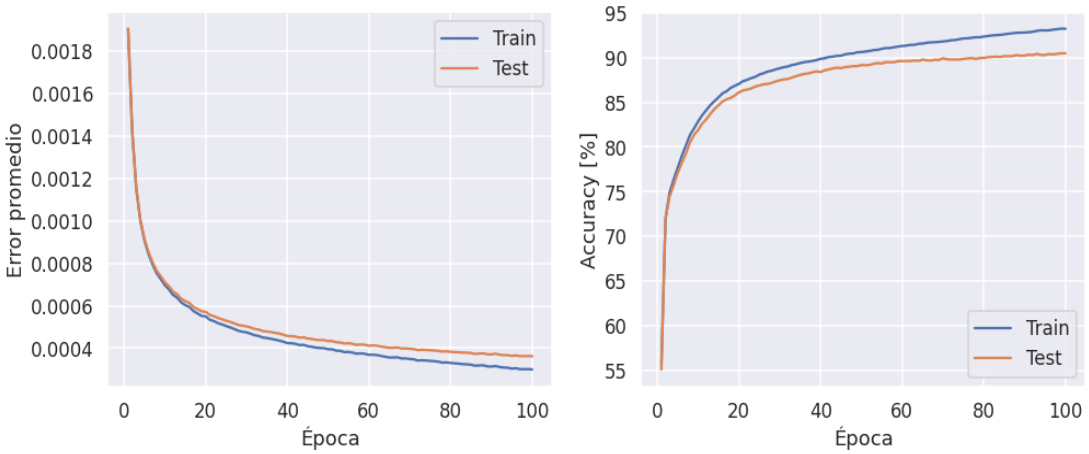

Figura 5: Error promedio y accuracy para train y test

En la figura 5 se observa que los valores de loss del entrenamiento y del testeo descienden con mayor énfasis hasta la época 20. Luego el decrecimiento es más paulatino hasta la época 100. Además, ambas curvas presentan un recorrido similar y por lo tanto no hay riesgo de overfitting. También, se visualiza que los valores de Accuracy superan el 90% en la época 100.

Se concluye que este último modelo es el que presenta un mejor rendimiento con respecto a los dos anteriores.

Comparación entre red neuronal de una sola capa oculta y sin convolución y una red neuronal convolucional

A continuación, se muestran los resultados obtenidos en el entrenamiento de una red neuronal con una sóla capa oculta de tamaño n para la tarea de clasificación de imágenes con la base de datos Fashion MNIST. Se exploraron dos tamaños diferentes para la capa oculta y se modificó el tamaño del batch.

Entonces:
- para $n = 2048$: *learning_rate* = 1e-3; *batch_size* = 500; *num_epochs* = 100
- para $n = 512$: *learning_rate* = 1e-3; *batch_size* = 1000; *num_epochs* = 40

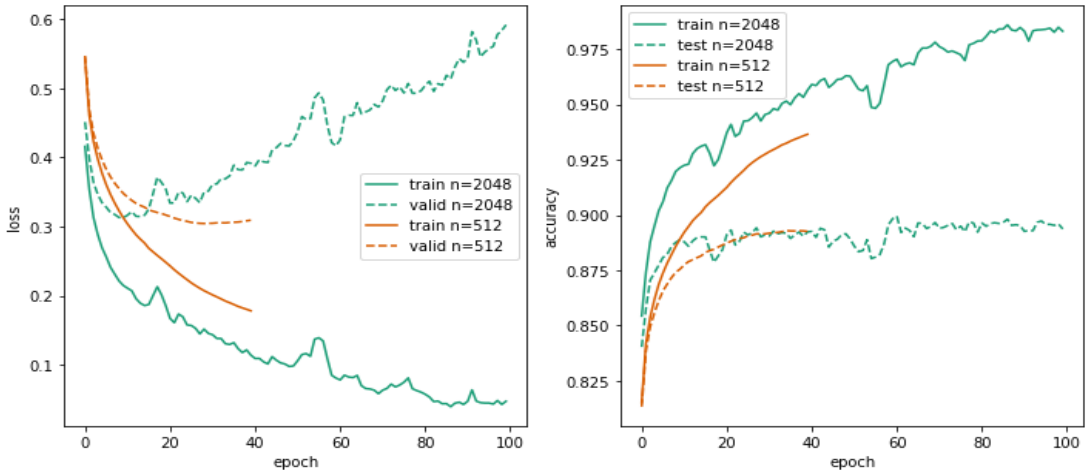

Figura 6: Error promedio y accuracy para train loss y test loss

En la figura 6 se observa que para $n = 2048$ el desempeño de la red es virtuoso hasta la época 10 aproximadamente, puesto que las loss de entrenamiento y validación decrecen mientras que el Accuracy crece. A partir de dicha época este fenómeno se detiene porque comienza a subir la loss de validación y el accuracy deja de crecer. Esto demuestra que a partir de la época 10 el modelo comienza a sobreajustar. Por otro lado, para $n = 512$ el desempeño de la red es virtuoso hasta la época 30 aproximadamente, dado que las loss de entrenamiento y validación decrecen mientras que el accuracy crece. A partir de dicha época este fenómeno se detiene porque comienza a subir el error de validación y el accuracy deja de crecer, lo cual demuestra que el rendimiento no mejora. Comparando ambos tamaños de n se concluye que $n = 512$ es el tamaño óptimo de la capa oculta.

A diferencia de la red neuronal con una sola capa oculta de tamaño n=512, el tercer modelo de CNN presenta valores de loss notablemente más bajos y los porcentajes de Accuracy son más elevados. Por último se estima que el aprendizaje en las redes neuronales sin convolución y con una capa oculta es un tanto más precario porque la ausencia de filtros impide una comprensión sofisticada de las imágenes.

Reflexiones finales

Luego de este trabajo exploratorio, se concluye que con una arquitectura simple de CNN, con dos capas convolucionales y dos capas ocultas, se llega a resultados satisfactorios. El tercer modelo escogido demuestra que, en las primeras épocas, se llega a una disminución considerable de los valores de loss y un aumento del Accuracy. Sin embargo, cabe resaltar que la cantidad de épocas de entrenamiento para llegar a los mínimos incrementa notablemente el costo computacional. Si en un futuro se quisieran obtener desempeños más virtuosos, se podría aumentar el número de convoluciones, filtros y capas ocultas, así como continuar explorando el espacio de los hiperparámetros.

Bibliografía:

- Arden Dertat (2018) *Applied Deep Learning - Part 4: Convolutional Neural Networks.* Recuperado de: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

- Sumit Sasha (2017) *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* Recuperado de: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

- Páginas web:
  https://cs231n.github.io/convolutional-networks/
  Optimizing Model Parameters — PyTorch Tutorials 1.10.1+cu102 documentation