

Facultad de Matemática, Astronomía, Física y Computación

Benjamin Martinez Picech

19 de julio de 2023

Resumen

Este documento tiene como objetivo implementar y evaluar dos arquitecturas de redes neuronales feedforward, como clasificadoras de imágenes de la base de datos Fashion-MNIST. La comparación será entre una red convolucional multicapa y un perceptron simple de una capa oculta, evaluando sus clasificaciones de los distintos tipos de prendas de vestir, analizando los errores en distintos sets de datos así como la máxima precisión alcanzada por cada una.

1. Introducción

Se explicará la implementación de una red neuronal de tipo feedforward multicapa y convolucional, sumado a un perceptron simple con una sola capa oculta, con el objetivo de aprender a clasificar las imágenes de la base de datos Fashion-MNIST. Esta base de datos consta de imágenes en escala de grises de 28x28 píxeles, y cada imagen está etiquetada con una de las 10 clases correspondientes a diferentes tipos de prendas de vestir. Es importante destacar que la red deberá tener 784 unidades de entrada (correspondientes a los píxeles de las imágenes) y 10 unidades de salida (correspondientes a las clases de prendas de vestir). Además, se utilizará la Cross Entropy Loss como función de pérdida, ya que es la más apropiada para problemas de clasificación, la red convolucional utilizará ADAM (Adaptive Moment Estimation) y el perceptron el descenso por el gradiente estocástico como optimizadores del método del descenso por el gradiente. Luego se realizará una comparación del rendimiento de la red convolucional profunda con los mejores resultados obtenidos utilizando una red de una sola capa oculta y sin convolución. Para entrenar a cada una de las redes, se utilizarán los conjuntos de entrenamiento, validación y prueba. Se analizará y comentará cómo la adición de capas convolucionales y la profundización de la arquitectura influyen en el rendimiento y la capacidad de clasificación de la red. Este trabajo tiene como objetivo principal explorar y comprender el funcionamiento y los efectos de distintas arquitecturas de redes neuronales en la clasificación de imágenes.

2. Teoría

Introduciremos los conceptos fundamentales relacionados con la implementación de redes neuronales convolucionales y porque son particularmente útiles en la tarea de la clasificación.

2.1. Filtros

También conocidos como Kernels son matrices $N \times N$ en general con N número impar, utilizados para procesar los inputs aplicando la operación de convolución con una determinada cantidad de kernels. Aplicando filtros a pequeñas regiones de la imagen, nos permiten capturar patrones locales como bordes, texturas y formas de la imagen. La operación de convolución también depende de otros hiperparámetros como el padding y el stride.

2.2. Pooling

Estas capas convolucionales usualmente son seguidas por capas de pooling, que reducen la dimensionalidad y preservan las características más relevantes. Esto es útil porque tenemos la capacidad dada por los filtros de resaltar y diferenciar las distintas partes de la imagen y luego condensamos la información de las neuronas más excitadas con operaciones de max-pooling o avg-pooling.

2.3. Adam(Adaptive Moment Estimation)

Optimizador del método del gradiente que mezcla las ideas de los metodos AdaGrad(Adaptive Gradient Algorithm) y RMSProp(Root Mean Square Propagation) para solucionar los problemas del método tradicional. Solucionando el problema de tener una única tasa de aprendizaje para todas las actualizaciones de los pesos sinápticos, además utilizando el momento de los gradientes como una introduccion de memoria en el calculo de los mismos, para eso utiliza el promedio y la varianza de las tasas de aprendizaje. Por eso este algoritmo tambien logra convativir con aquellas regiones llanas en las que el gradiente de aprendizaje tiende a ser cercano a 0 y asi disminuir considerablemente la cantidad de epocas necesarias para entrenar la red.

3. Resultados

En esta sección se explicara la arquitectura de las redes mas en detalle y también se mostraran gráficos con los resultados obtenidos para cada una. Para ambas redes se utiliza softMax como la función de activación para las neuronas de salida. Por otro lado decidí entrenar 12 épocas la red convolucional y 20 la del perceptron, dada la diferencia sustancial en la velocidad de aprendizaje.

3.1. Perceptron

Luego de una exploración de los hiperparametros me decidí por utilizar el descenso por el gradiente estocástico con un learning rate = 0.001 para el perceptron simple ya que para valores mas grandes el modelo no convergía, ademas ya sabia por resultados vistos en clase que la mejor arquitectura tenia 512 neuronas para la capa oculta utilizando en este caso dropout para evitar overfitting del 25 %.

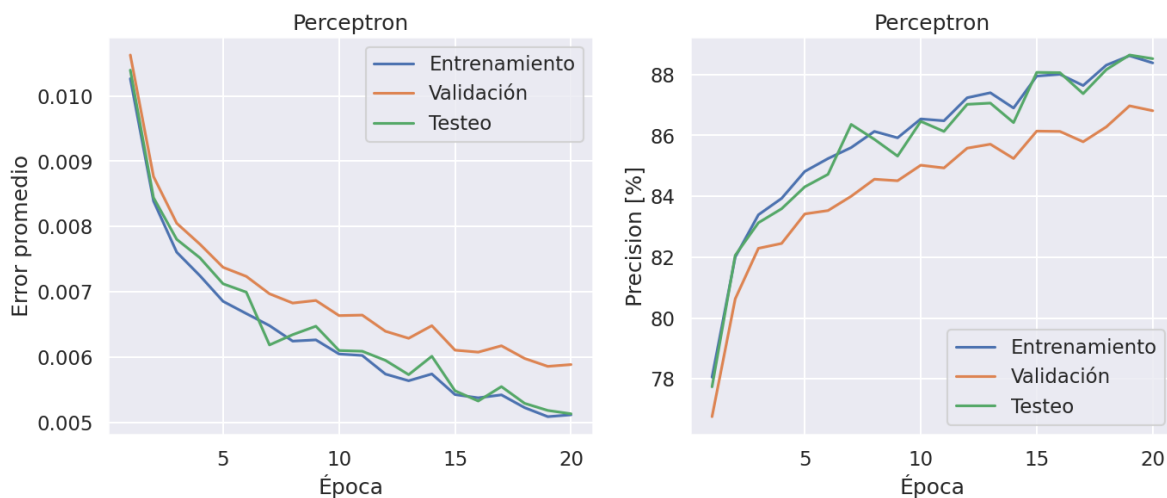


Figura 1: Error Promedio y Precisión del Perceptron en función de las épocas de entrenamiento

3.2. Red Convolucional Multicapa

Para la red convolucional me decidí por armar una red con 2 convoluciones, la primera con un kernel=3x3, padding=1 y con 32 canales de salida utilizando Relu, la segunda con un kernel=5x5, padding=2 y con 64 canales de salida. Utilizando un único MaxPooling en el final de la capa convolucional de 2x2 y un stride de 2. Para la capa totalmente conectada utilizo 64 x 7 x 7 neuronas de entrada con un dropout=0.25, capa oculta con 600 neuronas también con el mismo dropout y una capa de salida con las 10 neuronas representando cada una de las clasificaciones posibles, particularmente para esta red encontré que el learning rate óptimo esta cerca del 0.0001.

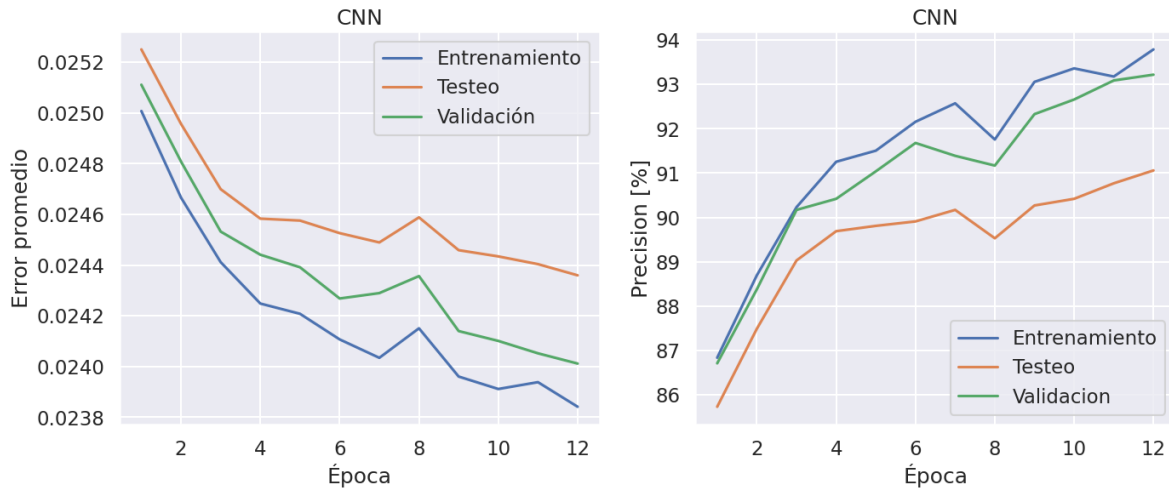


Figura 2: Error Promedio y Precisión de la Red Convolutiva en función de las épocas de entrenamiento

La siguiente imagen muestra los filtros obtenidos luego del entrenamiento de 12 épocas de la red convolutiva multicapa. Es difícil de entender que características resalta cada uno pero la red aprende cuales son los valores necesarios para que en conjunto puedan entender las características particulares de cada imagen.

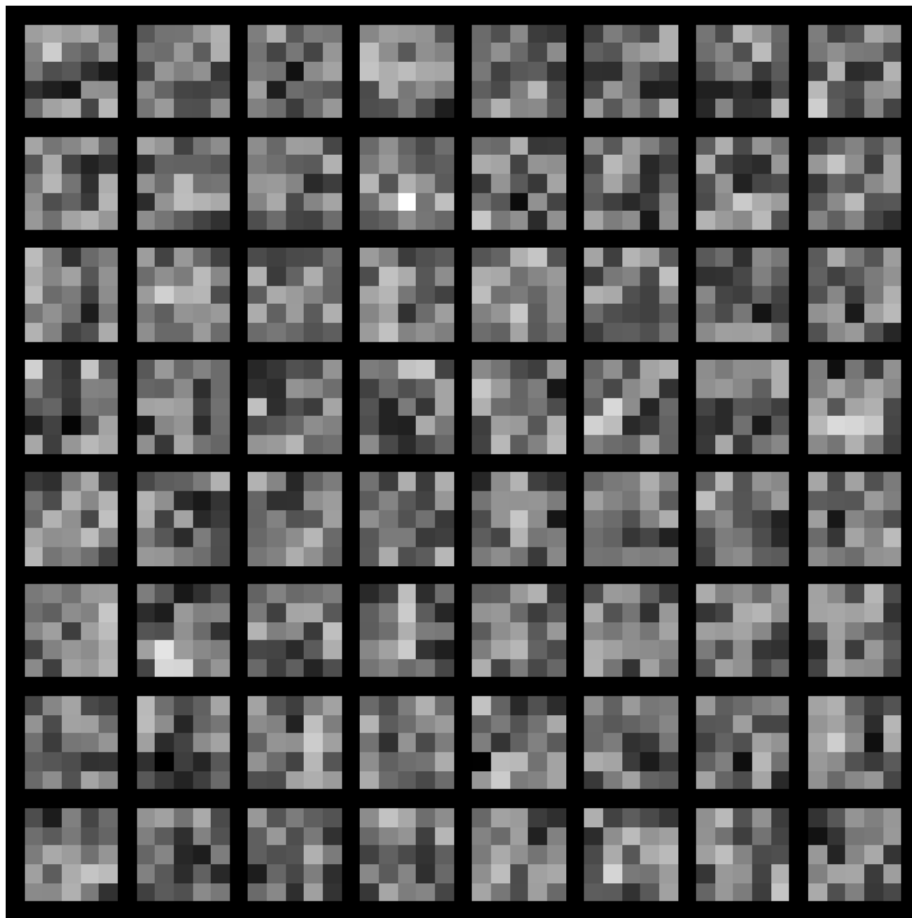


Figura 3: Los 64 Filtros de la segunda capa de convolución

Por ultimo una demostración de las predicciones realizadas por cada red neurona ante una imagen elegida al azar del set de datos de testeo. Podemos notar que la suma de las probabilidades asignadas en cada una de las predicciones es igual a 1, esto se da por el uso de la función de activación softMax como la activación de salida para ambas redes.

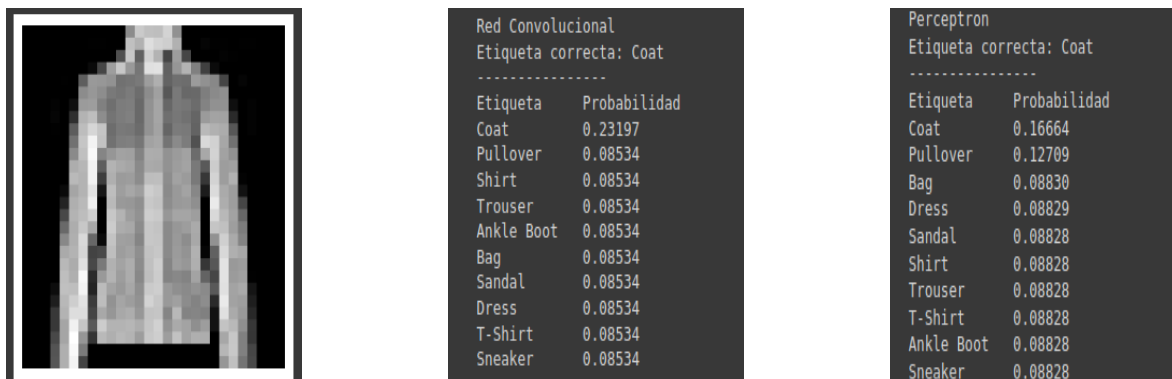


Figura 4: Predicción de ambas redes entrenadas de una imagen etiquetada como un 'COAT'

4. Conclusión

Las imágenes tienen una estructura espacial que puede ser explotada por las redes convolucionales. Al utilizar filtros, las redes convolucionales pueden capturar eficientemente la información espacial en las imágenes. Esto les permite detectar características en diferentes ubicaciones de la imagen sin necesidad de tener una conexión completa entre todas las unidades de la red. Por otro lado, una red feedforward con una sola capa oculta no puede aprovechar esta estructura espacial de manera tan eficiente, lo que puede llevar a una pérdida de información y un rendimiento inferior en la clasificación de imágenes. Por otro lado las redes convolucionales están compuestas por múltiples capas, lo que les permite aprender jerárquicamente características más complejas a medida que se profundiza en la red. Las primeras capas convolucionales pueden aprender características simples como bordes y texturas, mientras que las capas posteriores pueden aprender características más abstractas y de alto nivel, como formas y estructuras completas. Esta capacidad de aprendizaje jerárquico es fundamental para la clasificación precisa de imágenes, ya que las características de las imágenes se componen de manera compleja. Esto sumado a la gran diferencia en la cantidad de neuronas entre las dos capas resultan en una gran diferencia de bias entre las redes y como ya vimos en los gráficos una diferencia inmensa en la velocidad de aprendizaje, siendo la red neuronal convolucional multicapa mucho mejor en ese sentido.

5. Bibliografía

- [Clase 23. El descenso por el gradiente con memoria y adaptativo](#)
- [Clase 24. Las redes neuronales convolucionales](#)
- [Aplicación de filtros de concurrencias en redes neuronales convolucionales](#)
- [CNN — Introduction to Pooling Layer](#)
- [Optimización de descenso de gradiente de Code Adam](#)
- [Different Types of CNN Architectures Explained](#)