# Redes Neuronales - Práctico 3

Luis Biedma

8 de marzo de 2021

## Ejercicio 1

Para resolver el ejercicio se implementó el autoencode feed-forward usando el paquete Keras (junto con Tensorflow), ya que nos permite crear redes de forma sencilla y directa. El contenido de la red por capas es el siguiente:

1. Capa de entrada con Dropout de 0.1, con un input de tamaño 784 ($28 \times 28$).
2. Capa intermedia con función de activación RELU de tamaño 64.
3. Capa de salida, nuevamente con tamaño 784, con función de activación sigmoidal.

El dataset utilizado es MNIST, que puede ser obtenido directamente desde Keras con la función $keras.datasets.mnist.load\_data()$. Nuestros conjuntos tienen 60000 imágenes de entrenamiento y 10000 de test o validación.

Como la función de pérdida (loss) a utilizar es el error cuadrático medio, debemos tener especial cuidado al revisar la métrica en base a los valores básicos de los píxeles. Para hacer un poco más fácil el trabajo del entrenador, se optó por dividir todos los valores de los píxeles por el máximo posible (255), para tener valores entre 0 y 1 para nuestros inputs.

Debemos optimizar con Stochastic Gradient Descent (SGD), por lo que es necesario realizar una exploración de hiperparámetros más minuciosa que utilizando otros optimizadores más avanzados. Tenemos 3 opciones al ejecutarlo:

- Tasa de aprendizaje o *learning rate*. Se buscaron valores entre 0.01 y 10.
- Momento o *momentum*. Se utilizaron valores entre 0 (sin momento) y 1.
- Aplicar o no Nesterov.
- Además, debemos utilizar un batch de 1000 imágenes en cada época de entrenamiento.

Después de una búsqueda de Grid Search, se estableció que los mejores valores de la función de pérdida se podrían conseguir con la combinación Nesterov, learning rate de 1 y un momentum de 0.5, con 100 épocas de entrenamiento. A continuación, se muestran algunos gráficos de diferentes dígitos aleatorios del conjunto de test y su correspondiente transformación al ser pasados por el autoencoder (arriba: original, abajo: transformado). A modo cualitativo, se puede ver que el autoencoder trabaja de forma aceptable, se podría decir que le aplica una ligera difuminación a las imágenes. Realizar 100 épocas más mejora ligeramente la visualización.

Figura 1: Comparación de dígitos aleatorios. Arriba original, abajo transformación.

Se puede visualizar en el avance de la función de pérdida que la misma tiene un gran decaimiento al comienzo y luego comienza a bajar de a poco. Es posible que, con más épocas, se consiga una función de pérdida más baja, pero esto sería a costa de un mal comportamiento en el conjunto de validación, por lo que se decidió parar en 100.

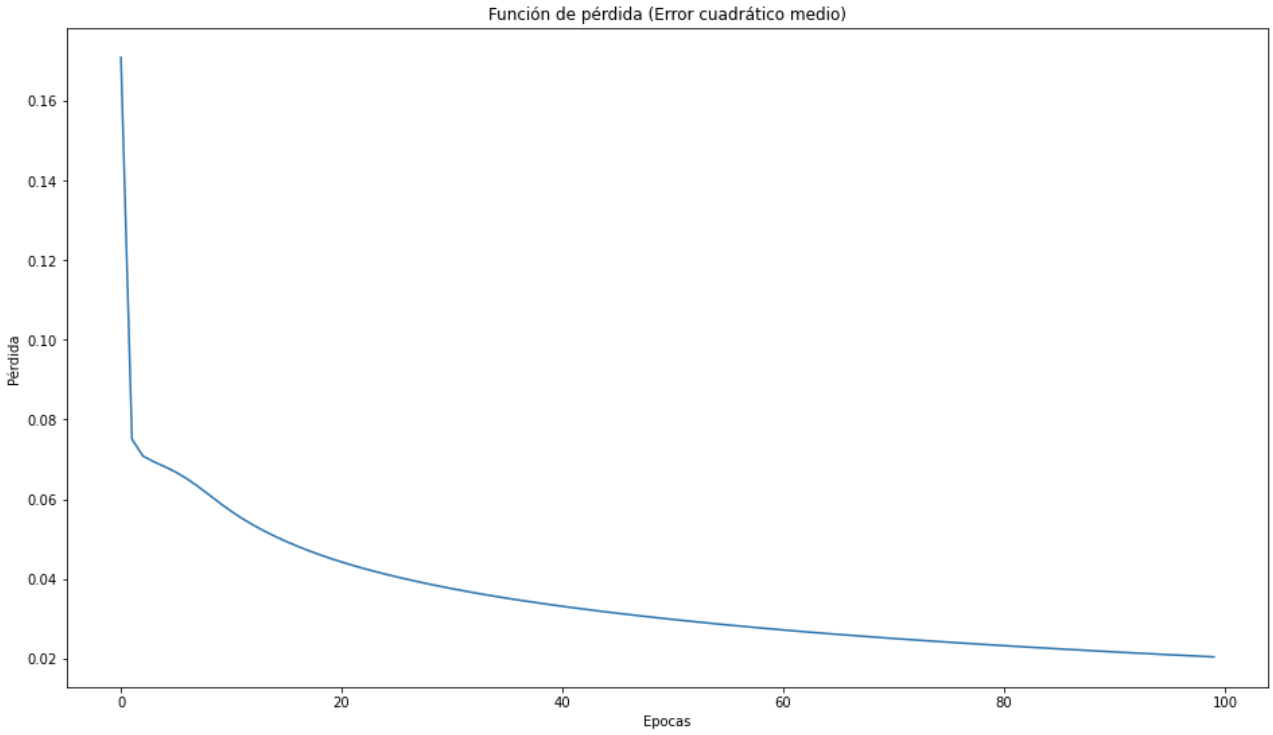


Figura 2: Pérdida (error cuadrático medio) en función de las épocas de entrenamiento

# Ejercicio 2

Para realizar el ejercicio 2 y facilitar la obtención de los datos para graficar, es posible utilizar nuestro conjunto de test como conjunto de validación en el entrenamiento de la red (dentro de la función $fit$ del modelo).

A continuación, se muestran las curvas de funciones de pérdida para las 4 configuraciones propuestas del autoencoder (64, 128, 256 y 512 neuronas en la capa intermedia). Esta vez se utilizaron 50 épocas de entrenamiento (para bajar el tiempo de preparación).

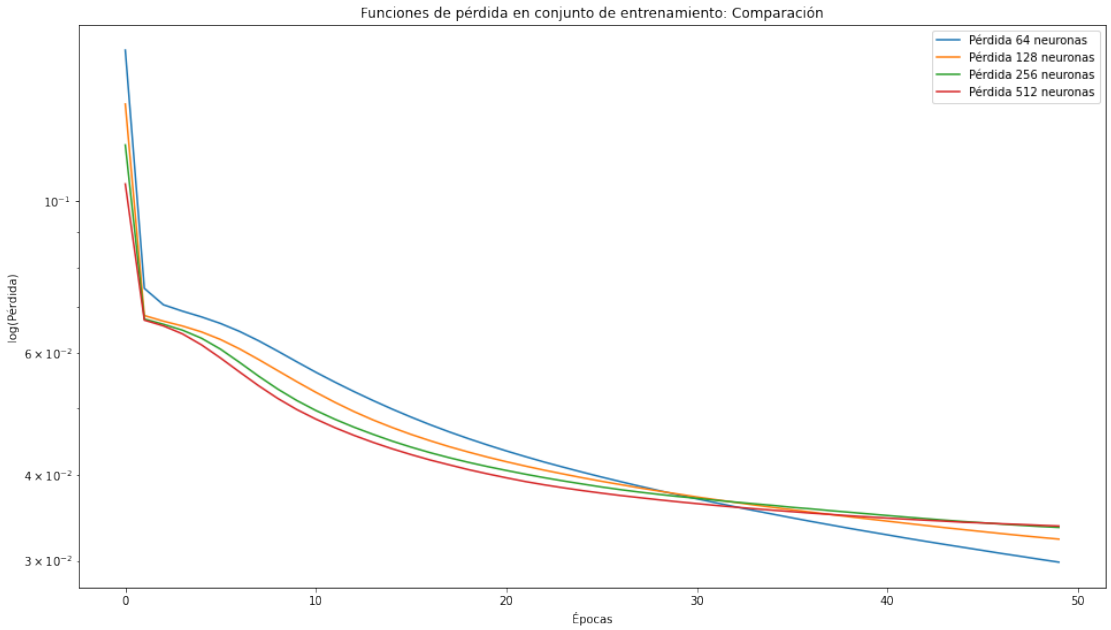


Figura 3: Pérdida en logaritmo en función de las épocas de entrenamiento

Podemos ver que no hay grandes diferencias de función de pérdida al aumentar la cantidad de neuronas en la capa intermedia, tanto para el conjunto de entrenamiento como para el de test. Esto puede indicar que la función de pérdida es altamente dependiente de los hiperparámetros que elijamos para inicializar el Stochastic Gradient Descent. Se utilizaron los mismos parámetros que se encontraron para el modelo de 64 neuronas en la capa oculta. Esto nos hace pensar que conviene a veces quedarse con el modelo de 64 neuronas, ya que los tiempos de entrenamiento se van casi duplicando con cada experimento.

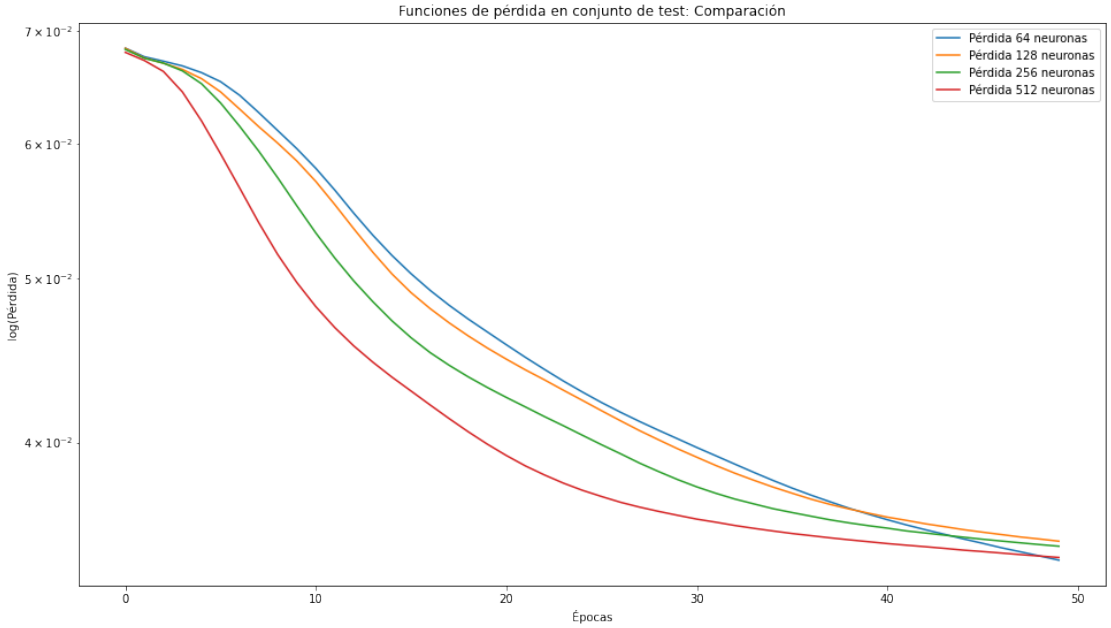


Figura 4: Pérdida en logaritmo en función de las épocas de entrenamiento (conjunto test)