

MNIST Feed-forward Autoencoder

Agustín Marcelo Domínguez

Redes Neuronales 2020 - Practico III

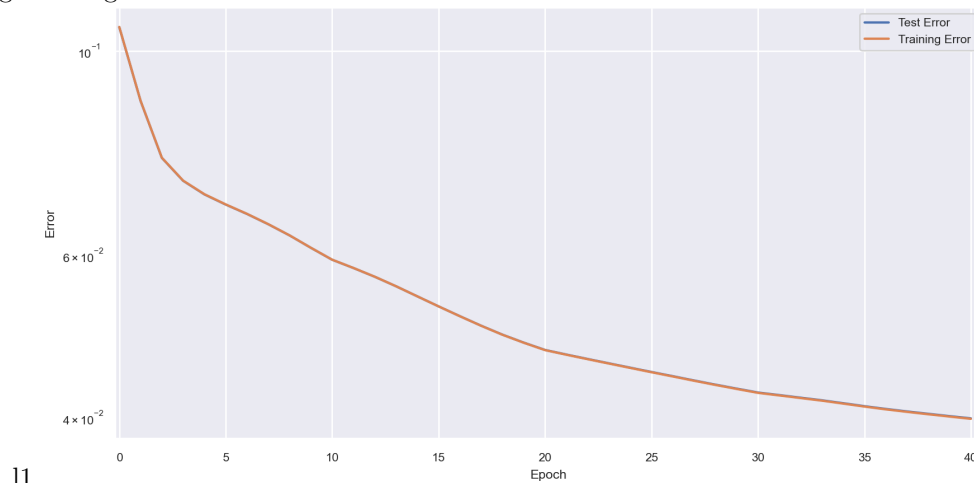
1 Introducción

Se implementó una red neuronal *feed-forward* con una capa oculta para aprender la función identidad sobre la base de datos de dígitos escritos a mano **MNIST**. Se comparó el error de entrenamiento y testeo para distintos números de neuronas en la capa oculta, y se comparó con otras funciones de optimización.

2 Primer desarrollo

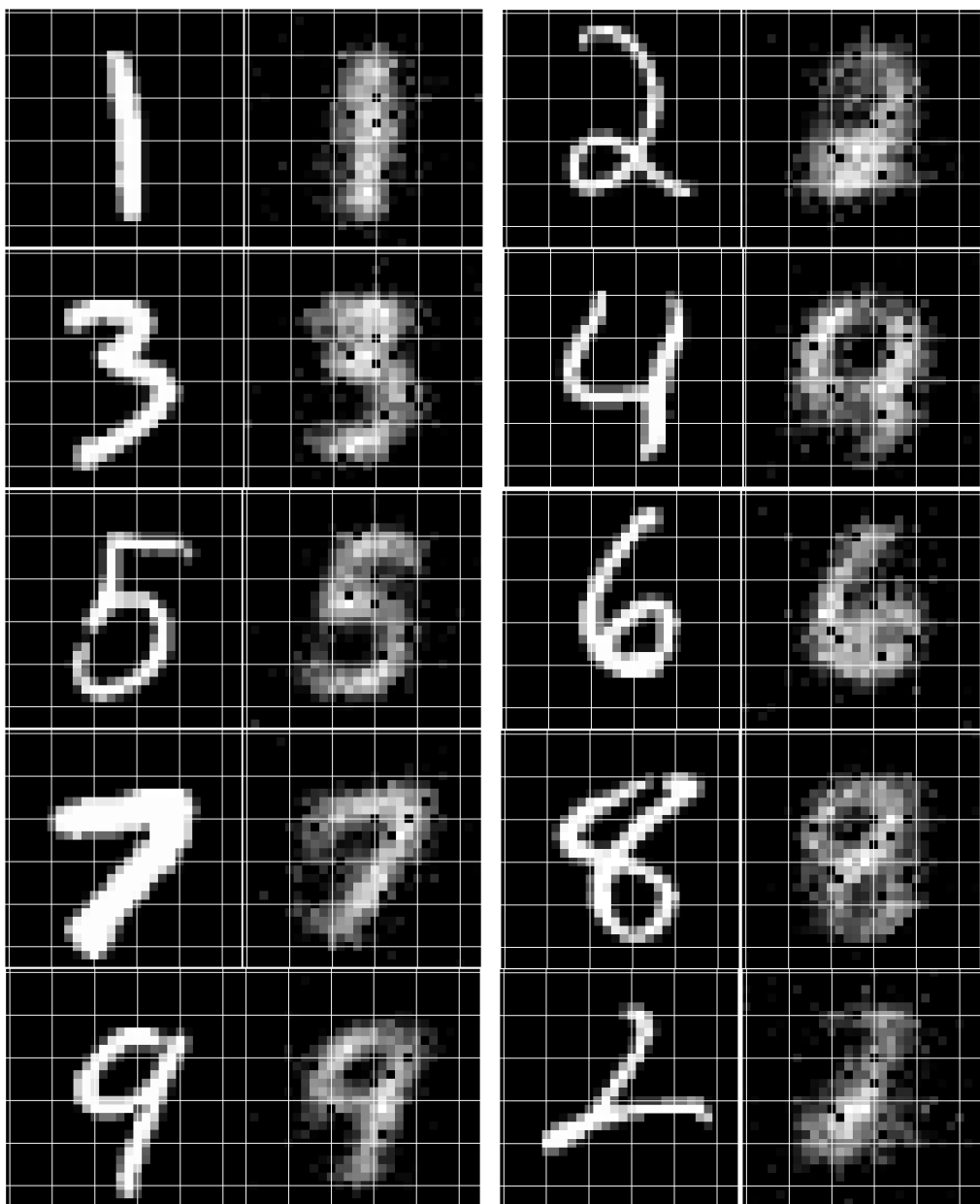
La red implementada tuvo **64** neuronas en la capa oculta, como función de error (*loss*) la función de error cuadrático medio (**MSELoss**), para el descenso la *función de descenso por el gradiente estocástico (SGD)*, minibatches de tamaño **1000**, y una división del conjunto de MNIST con 20000 imágenes de test y 50000 de entrenamiento.

Luego de ajustar algunos meta-parámetros (*learning rate* y *momento*), se llegó a la siguiente red:



11

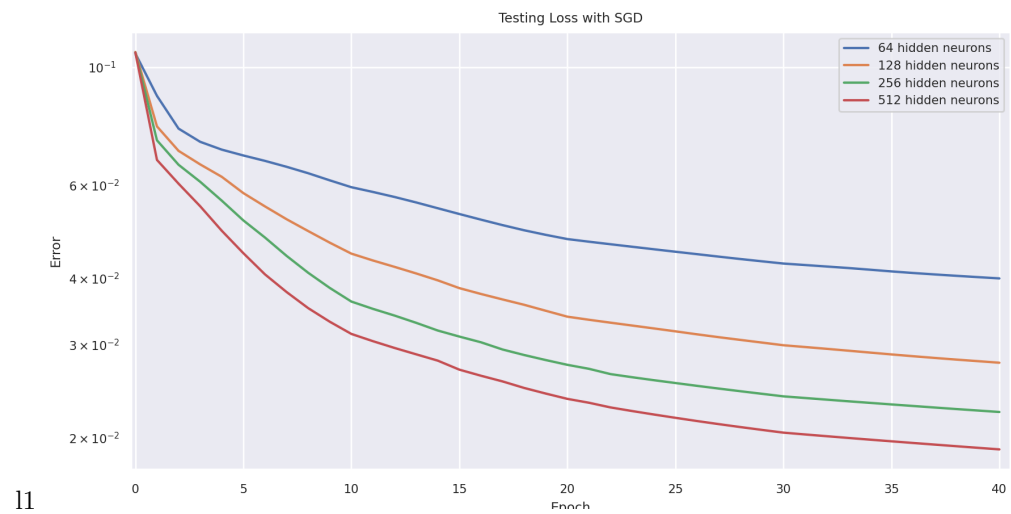
Que produce la siguiente función identidad:



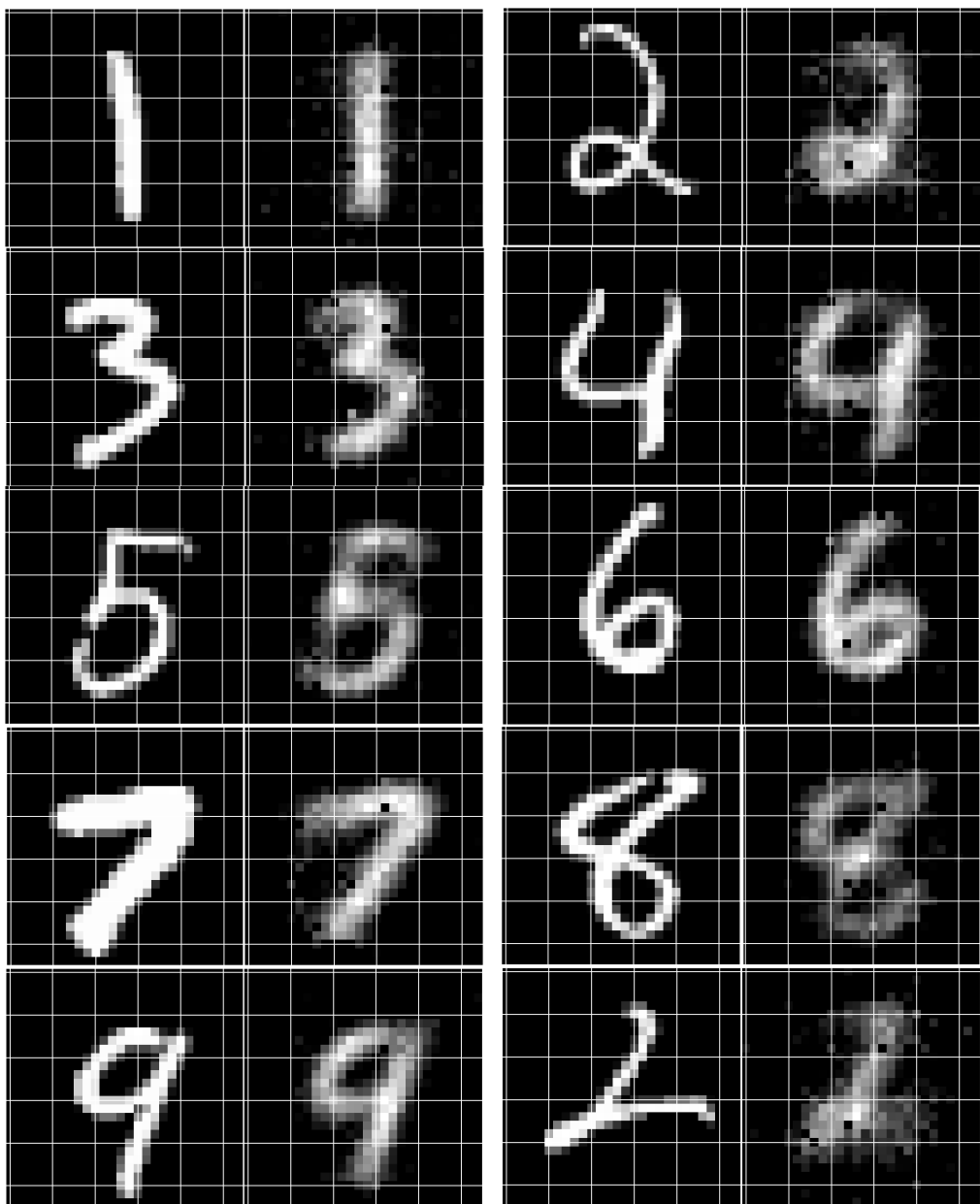
11

Una pista de los resultados deseados pero todavía no como se esperaba.

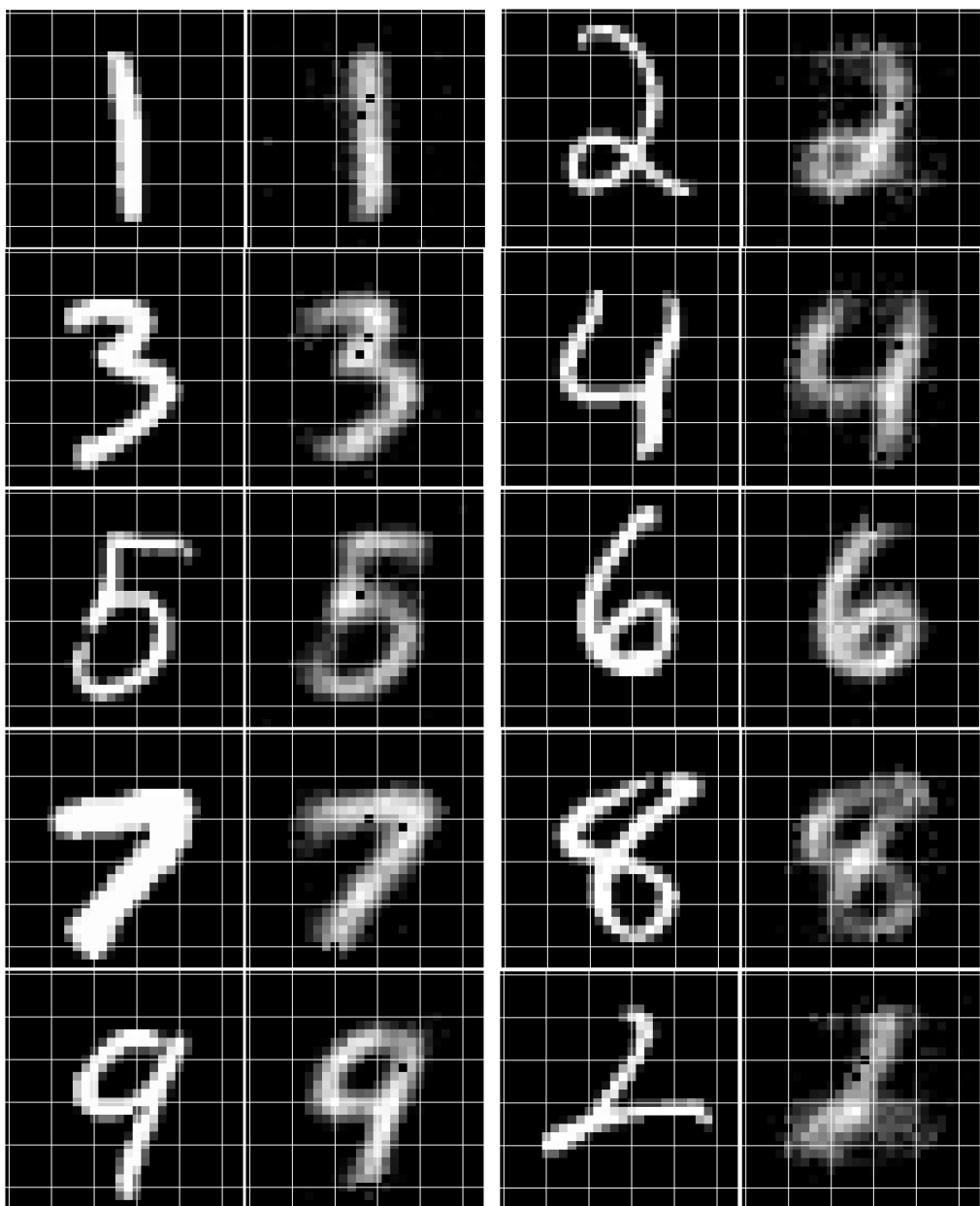
Luego se aumentó la cantidad de neuronas de la capa oculta, a **128**, luego **256** y **512**. Cada duplicado de neuronas produjo una nueva meseta de error en el conjunto de testeo:



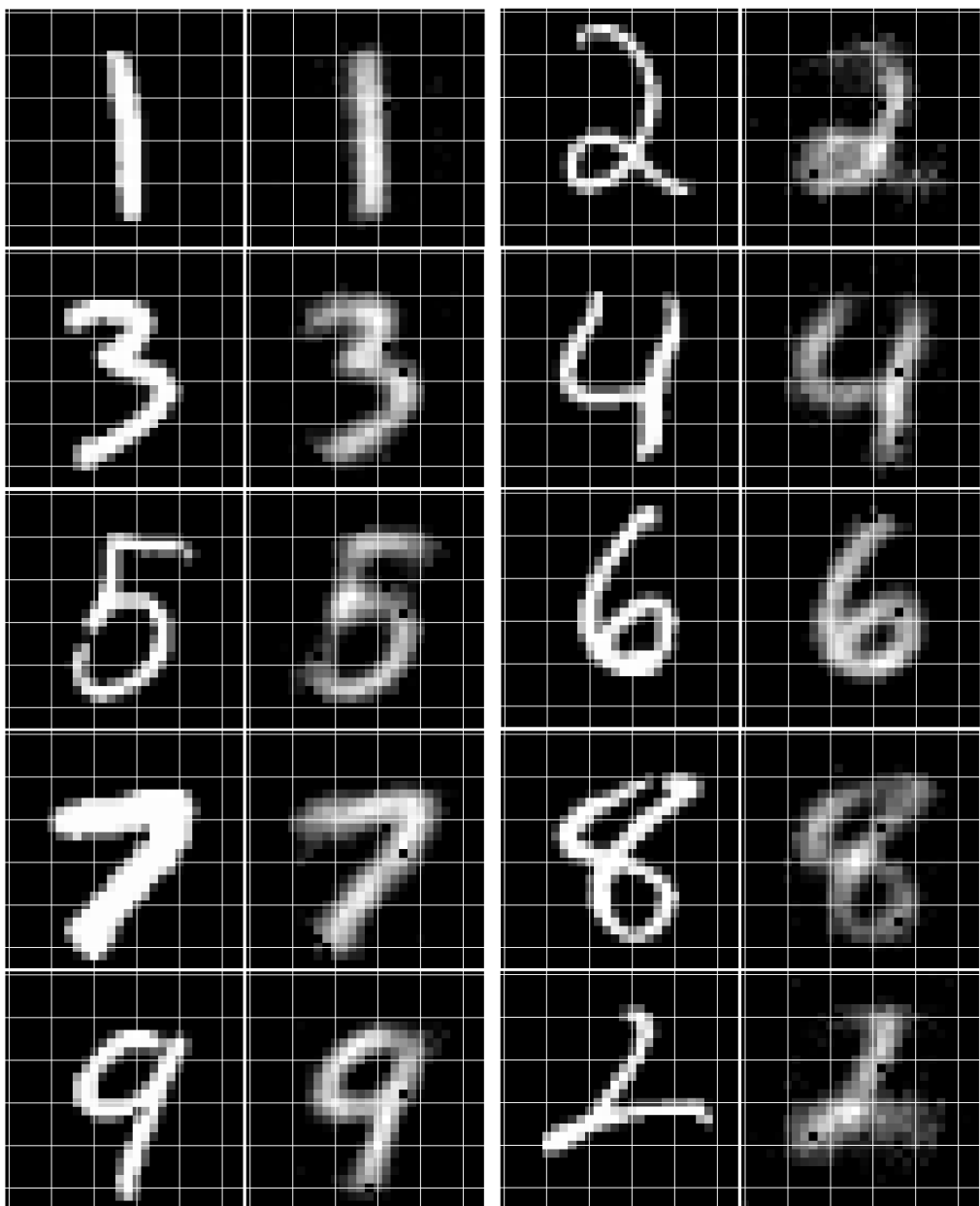
Que producen, en turno, los siguientes resultados evaluados (en orden 128, 256, 512):



11



11



11

Hay algunos pixeles 'muertos' que sin importar que número tienen permanecen con el mismo valor. Esa optimización ayuda a la reducción de dimensionalidad en los pixeles de las esquinas donde esa suposición no genera errores, pero claramente en los pixeles más cercanos al centro de la imagen no funciona tan bien.

Antes de ver otros optimizadores cabe mencionar que el modelo, incluso con 512 neuronas, no estaba memorizando el dataset ya que el error del training y el de testing se mantenían prácticamente iguales durante todas las épocas. Esto va a cambiar en los otros modelos donde se pudo reproducir la memorización.

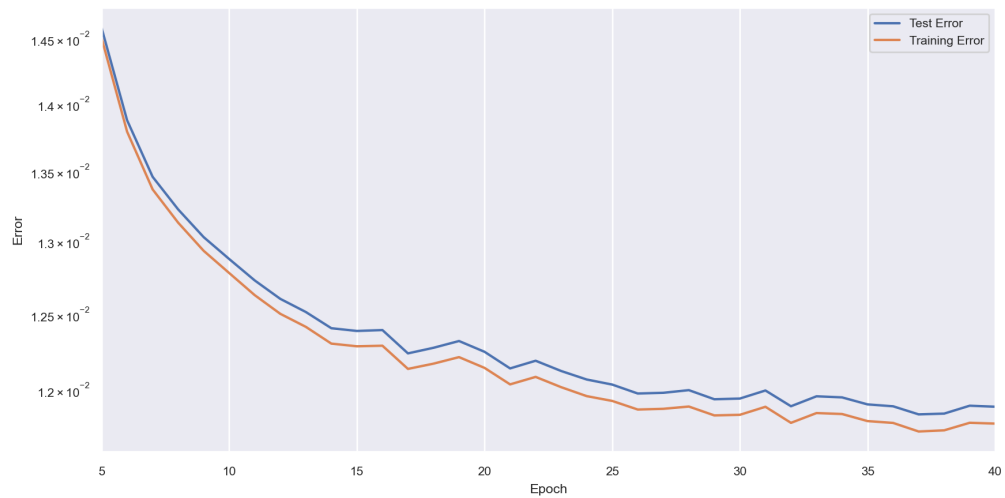
Como siempre, el código está disponible en el repositorio:

https://github.com/AgustinMDominguez/Redes_Neuronales_Practico_3

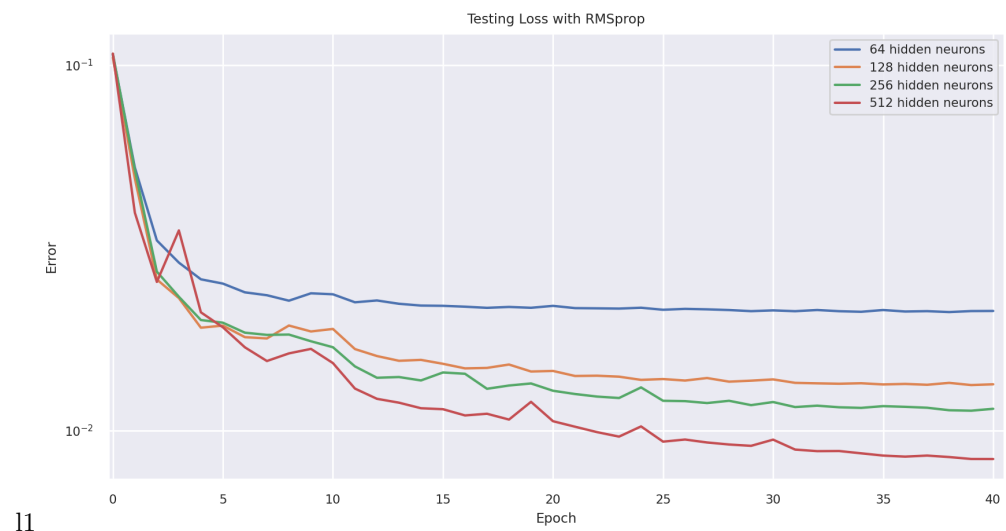
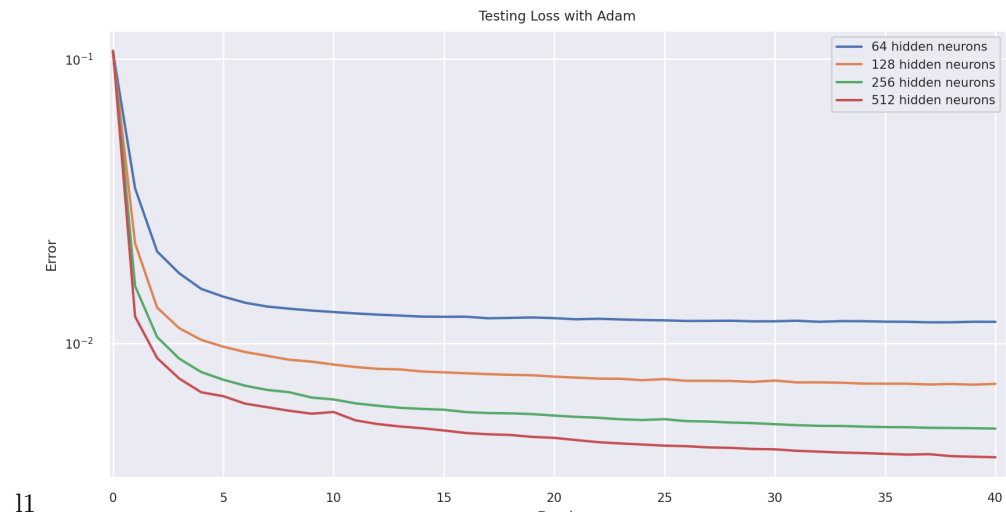
3 Otros optimizadores

Se comparó el mismo modelo con las mismas variaciones de neuronas pero con otros modelos de optimización: **Adam** y **RMSprop**. Estos modelos no solo produjeron resultados con menor error, sino que también superaron los mejores resultados de las redes anteriores en menos épocas (aunque es posible que con otros parámetros de SGD se podría mejorar el resultado).

Adam con 40 épocas y 64 neuronas



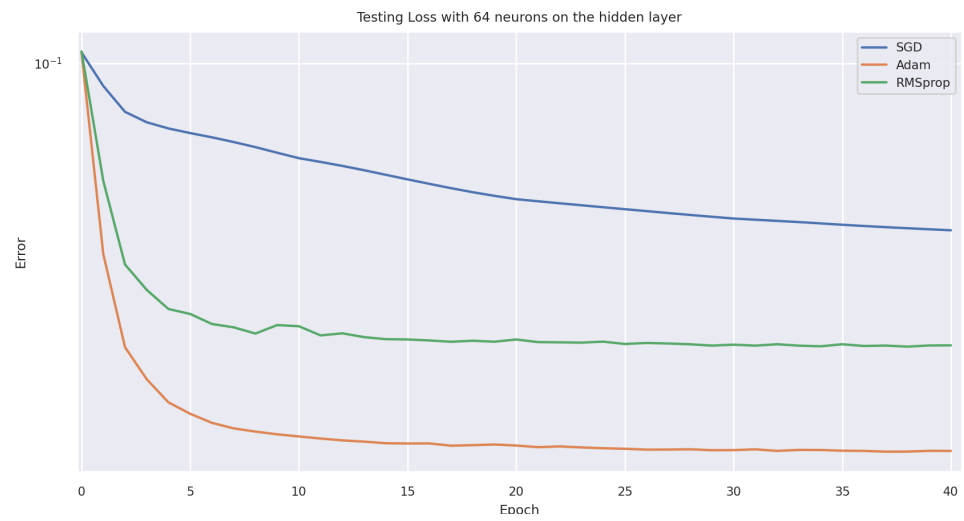
11



La irregularidad y saltos en RMSprop puede ser señal de metaparámetros que requieren un mejor ajuste, aunque incluso con esa inestabilidad la red llega a la meseta eventualmente y a un error bajo.

Comparación entre los optimizadores:

11



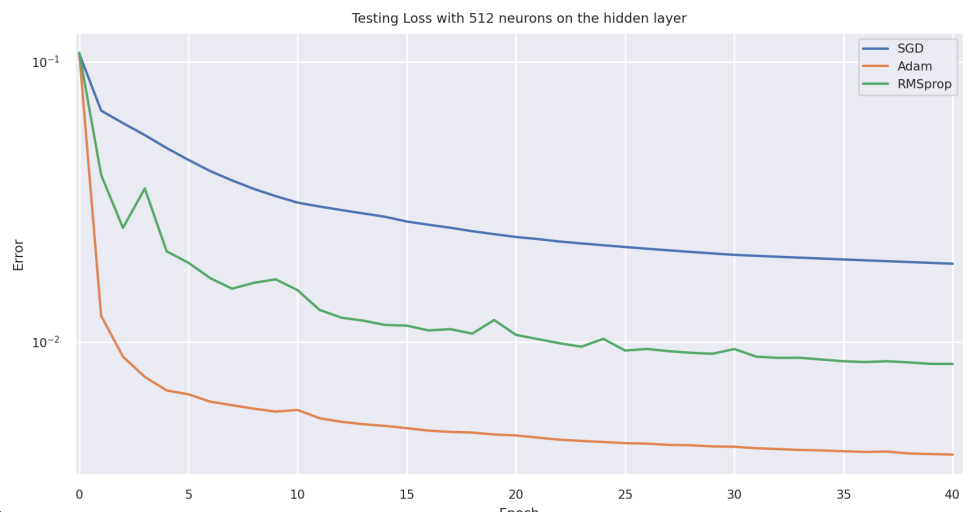
11



11

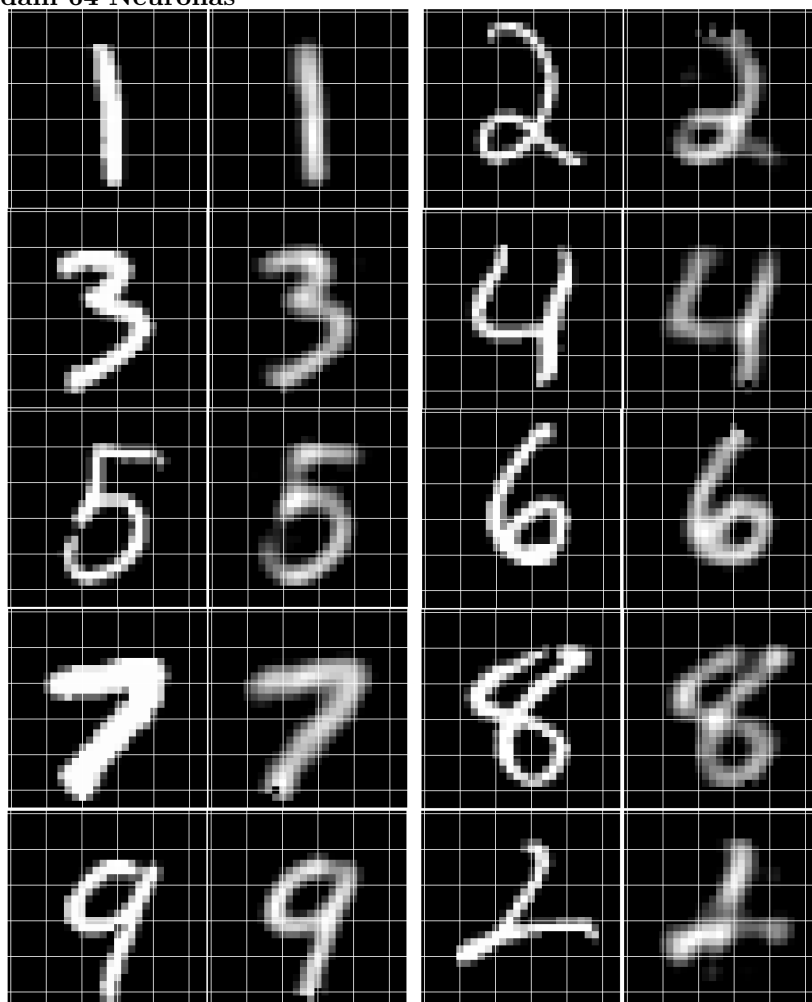


11



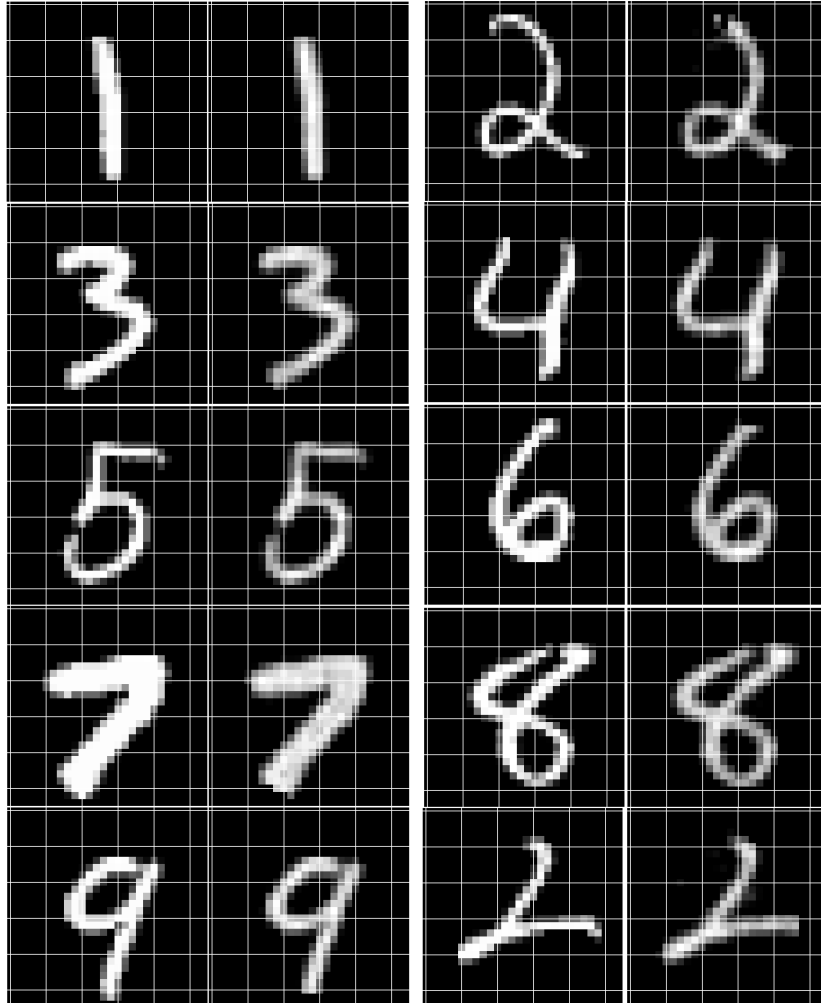
Por último, veamos la evaluación:

Adam 64 Neuronas



11

Adam 512 Neuronas



11

Si bien 512 neuronas para una imagen de 28x28 píxeles de blanco y negro no es una baja muy impresionante de dimensión, son notables los resultados obtenidos con solo una capa de neuronas ocultas y un sistema feed-forward que pierde espacialidad en su análisis.

Con redes convolucionales o una red ligeramente más profunda, los resultados deben ser aún mejores.