Deep Learning: Regularizers № 4

Daniela Pinto Veizaga, dpintove@itam.mx
Diego Villa Lizárraga, dvillali@itam.mx

19/02/2020

Introducción

Para la implementación de la presente tarea, emplearemos la base de datos MNIST; esta base de datos contiene imágenes binarias de dígitos escritos a mano. Usaremos las imágenes extendidas en forma de vector como datos *x* y sus respectivas etiquetas (enteros) como valores de salida *y*.

El objetivo es diseñar redes neuronales que, con ayuda de regularizadores, sean capaces de obtener buen desempeño de clasificación multi-clase, tanto en los datos de entrenamiento como en los de validación y prueba.

Pregunta 1

Según los resultados que hayas obtenido, ¿cuál de los dos modelos es preferible y por qué?

A pesar de que los dos modelos de redes neuronales no son propiamente comparables, puesto que –si bien emplean funciones de activación ReLu en las capas intermedias y funciones Softmax en las capas de salida– sus arquitecturas son ligeramente diferentes[1], el segundo modelo presenta un mejor desempeño, tanto en el *training loss* como en el *validation loss*.

Modelo	*Precision Test*	*Test Loss*
Modelo 1: Sin Regularizadores	0.11349	2.3024
Modelo 2: Con Regularizadores	0.1427	2.2539

Table 1: Resultados de Modelos Iniciales.

[1] El modelo 1 es una red neuronal con tres capas ocultas, cada capa de 4, 2 y 1 nodos, respectivamente; el modelo 2, es una red neuronal con tres capas ocultas, cada capa de 4, 4 y 4 nodos, respectivamente.)

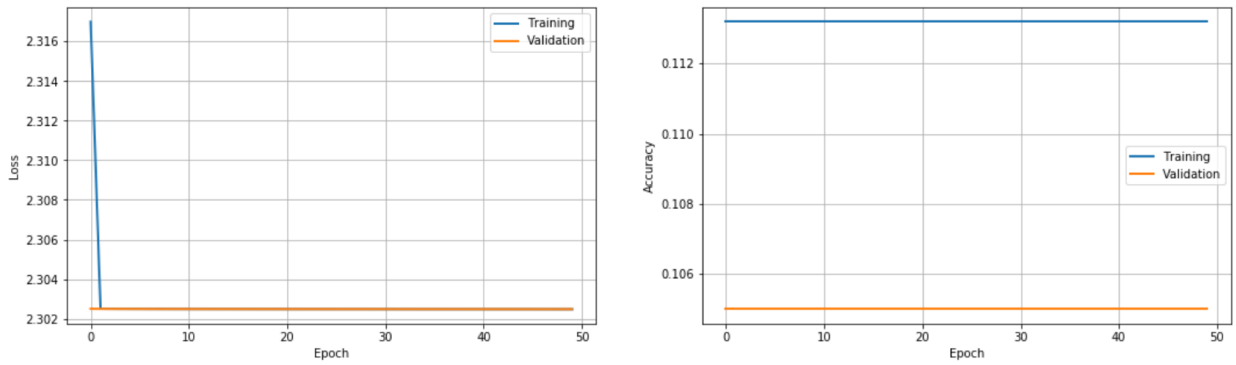

Figura 1. Modelo 1, Precisión y Pérdida

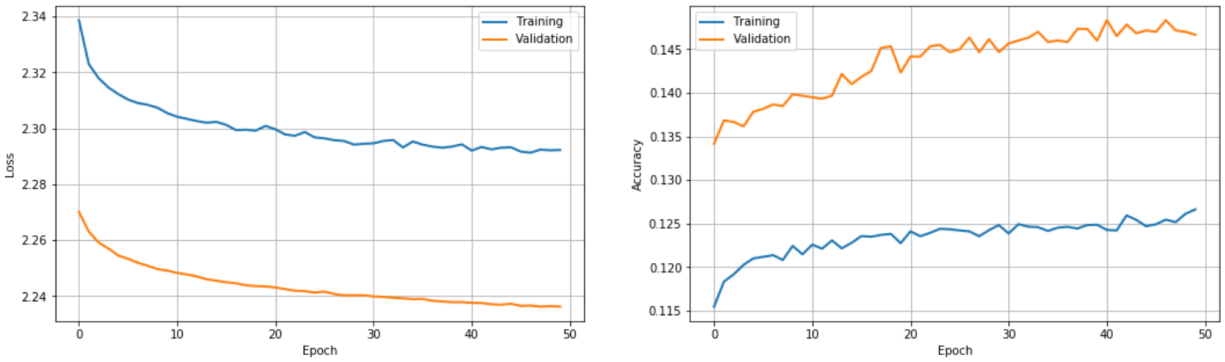

Figura 2. Modelo 2, Precisión y Pérdida

Como se puede observar, ambos modelos son bastantes ingenuos en la clasificación de las imagénes que se les proveyó: por cada 100 imágenes, únicamente son capaces de clasificar correctamente 11 y 14 de ellas, respectivamente.

Para visualizar lo antes comentado de mejor manera, remitámonos a las Figuras 3 y 4, donde se presentan las matrices de confusión asociadas a los modelos. De la figura 3 se deduce que el modelo 1, sin regularizar, clasifica la mayoría de las imágenes como imágenes del número 1; de la figura 4 se concluye que el modelo 2 clasifica la mayoría de las imágenes como imágenes del número 3, 7 y 8.

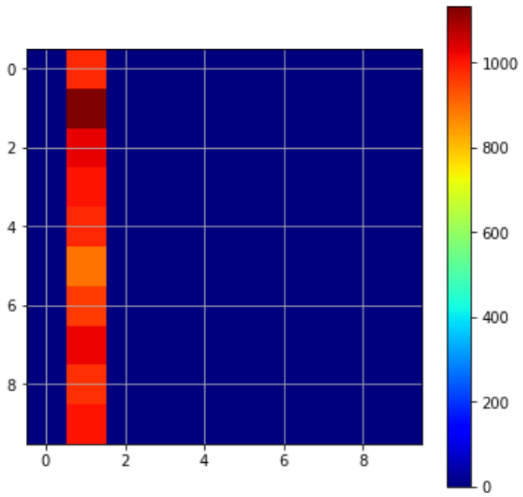

Figura 3. Modelo 1, Matriz de Confusión

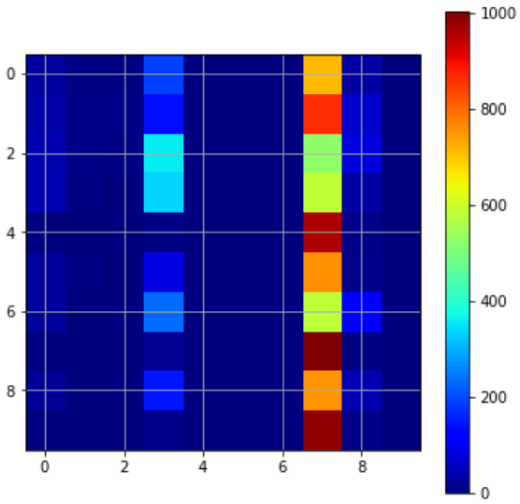

Figura 4. Modelo 2, Matriz de Confusión

Pregunta 2

¿Por qué usamos softmax en la salida de la red?

Softmax lleva esta idea al plano de las clases múltiples. Es decir, softmax asigna probabilidades decimales a cada clase en un caso de clases múltiples. Esas probabilidades decimales deben sumar 1.0. Esta restricción adicional permite que el entrenamiento converja más rápido.

Pregunta 3

Ajusta el primer modelo (sin regularizadores) para obtener una pérdida de "entrenamiento" menor o igual a 0.08 y exactitud mayor o igual a 98 por ciento. Reporta el número de capas y sus tamaños.

Implementamos una red profunda con la siguiente arquitectura: 5 capas ocultas, cada capa con 64, 64, 128, 64, 64 nodos, respectivamente. Para todas las capas ocultas, se emplearon funciones de activación ReLU.

Como se observa en la **Figura 5.**, el modelo implementado mejora sustancialmente en comparación con los primeros modelos implementados puesto que se obtienen los siguientes valores:

Modelo	Test Accuracy	Test Loss	Training Loss
Modelo cinco capas ocultas: Sin Regularizadores	0.9768	0.1782	3.2868e-5

Table 2: Resultados de Modelo con cinco capas ocultas, sin regularizadores.

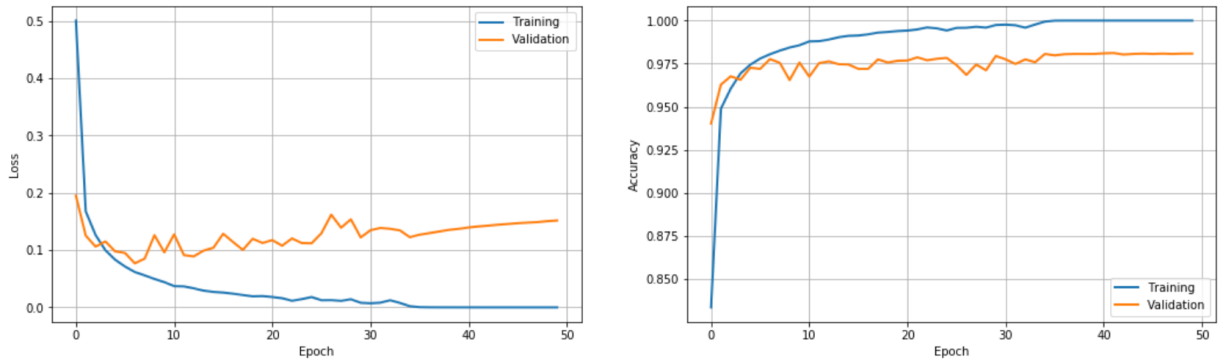

Figura 5. Modelo sin regularizadores

Observando más detenidamente la gráfica de la izquierda de la **Figura 5.**, observamos que nuestro modelo presente una comportamiento de sobre-entrenamiento a partir de la época 30: la pérdida del *validation set* incrementa, mientras la pérdida del *training set* disminuye.

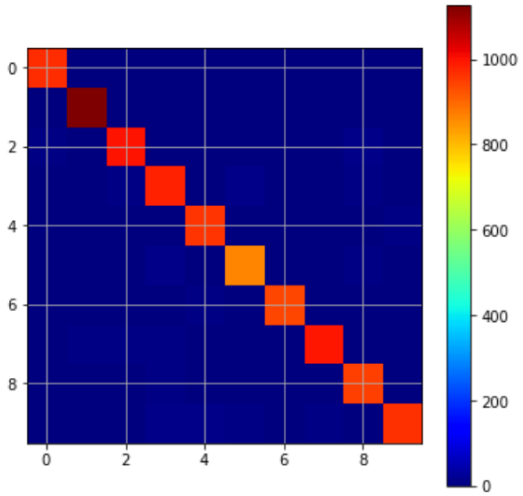

Figura 6. Modelo sin regularizadores, Matriz de Confusión

La **Figura 6.** muestra que, la arquitectura, empleada a propósito de este inciso, no tiene problemas particulares para clasificar ningún número. Entre los nu mero que presentan mayor dificultad para ser clasificados, se resaltan el 5. Por otro lado, la red clasifica de mejor manera las imágenes de los números 1.

Pregunta 4

Ahora usa esos mismos valores de hiperparámetros (número de capas y sus tamaños) en el segundo modelo, y ajusta la taza de dropout, y las alfas en los regularizadores l1 y l2 para disminuir el error de generalización (validación). Reporta el modelo regularizado que te haya dado mejores resultados.

Con la arquitectura de la red neuronal implementanda a propósito de la pregunta No 3[2], se incluyeron los siguientes regularizadores: 1) dropout (rate de 0.33); 2) batch normalization.

[2]Red Neuronal profunda con 5 capas ocultas ReLU, con nodos 64, 64, 128, 64, 64.

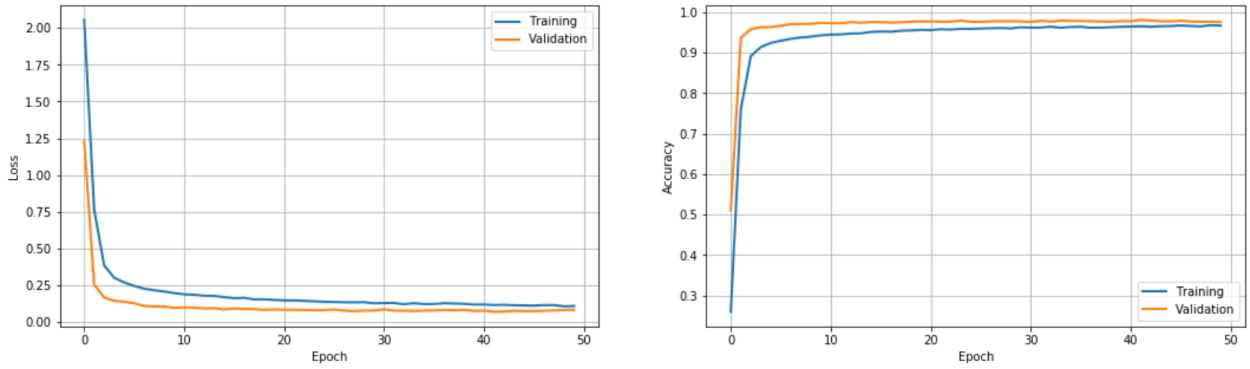

Figura 7. Modelo con Regularizadores, Despempeño

Después de entrenar los modelo con cinco capas ocultas, obtuvimos un *training loss* de 0.1084 contra un *test loss* de 0.094 y un *test accuracy* de 0.9718.

Algunas anotaciones: Es importante destacar que en la **Figura 7.** se observa que la distancia entre el *training loss* y el *test loss* es mucho mas estrecha gracias a la implementación de regularizadores. Ademas, la tendecia de esta distancia es decreciente conforme avanzan las epocas, se observa la misma tendecia en la grafica de accuracy.

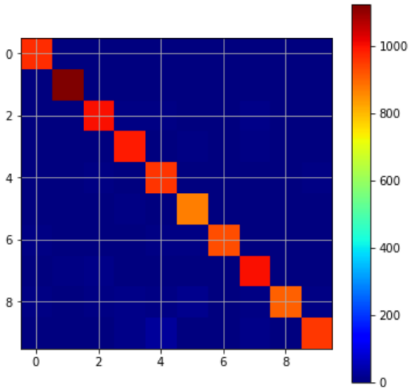

Figura 8. Modelo con Regularizadores, Matriz de Confusión

Pregunta 5

Partiendo del mejor modelo que hayas obtenido anteriormente, modifica el número de sus capas y tamaños para disminuir aún más los errores, tanto el de entrenamiento como el de validación. Reporta tu mejor modelo.

Finalmente, implementamos una red profunda con 5 capas ocultas ReLU, con nodos 64, 64, 128, 64, 64, incluyendo cuatro regularizadores: a) dropout (rate de 0.33); b) batch normalization; c) l; d) l2.

En general, el rendimiento de esta arquitectura es superior a las implementadas previamentes, en términos del *test loss*: 0.0889. Sin embargo, en términos de test accuracy, el mejor modelo fue el de cinco capas SIN regularizadores. Remitirse a la **Tabla 3.** para ver el comparativo entre los tres modelos implementados con cinco capas.

Modelo	Test Accuracy	Test Loss	Training Loss
Modelo cinco capas ocultas: Sin Regularizadores	0.9768	0.1782	3.2868e-5
Modelo cinco capas ocultas: 2 Regularizadores	0.9718	0.094	0.1084
Modelo cinco capas ocultas: 4 Regularizadores	0.975	0.0889	0.0878

Table 3: Resultados de Modelo con cinco capas ocultas, sin regularizadores.

```
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_59 (Dense)             (None, 64)                50240
_________________________________________________________________
dropout_8 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_60 (Dense)             (None, 64)                4160
_________________________________________________________________
batch_normalization_6 (Batch (None, 64)                256
_________________________________________________________________
dense_61 (Dense)             (None, 128)               8320
_________________________________________________________________
dense_62 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_63 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_64 (Dense)             (None, 10)                650
=================================================================
Total params: 76,042
Trainable params: 75,914
Non-trainable params: 128
_________________________________________________________________
```

Figura 9. Modelo Final, Arquitectura

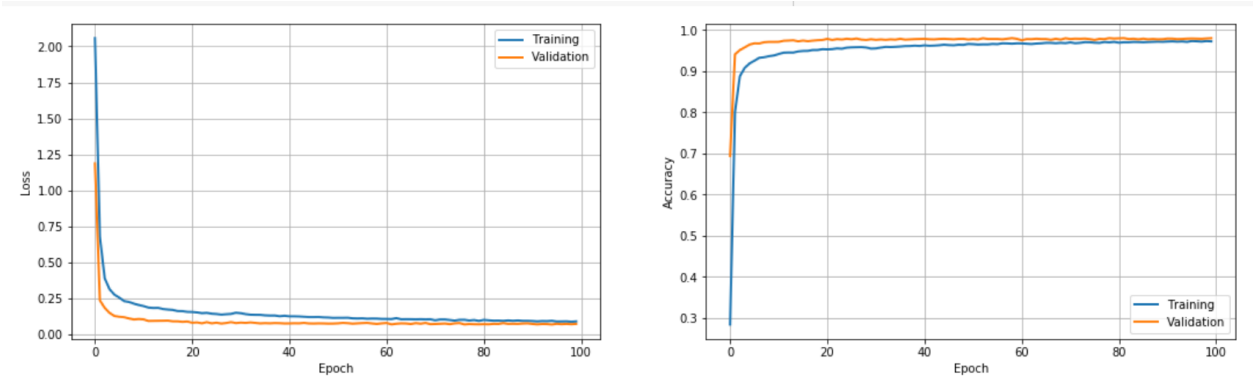

Figura 10. Modelo Final, Desempeño

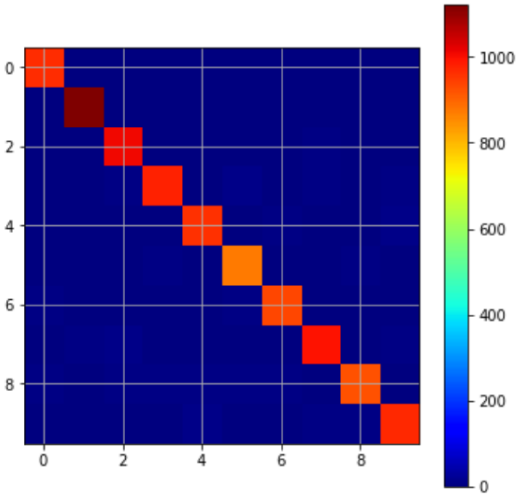

Figura 11. Modelo Final, Matriz de Confusión