



Trabajo Práctico I — Checkpoint IV

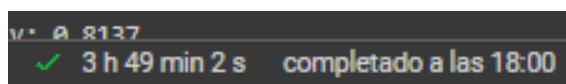
[75.06/95.58] Organización de Datos
Primer cuatrimestre de 2023

Alumno	Número de padrón	Email
Masri, Noah	108814	noahmasri19@gmail.com
Ayala, Camila	107440	cayala@fi.uba.ar
Loscalzo, Melina	106571	mloscalzo@fi.uba.ar

En este checkpoint, el objetivo era el de implementar una red neuronal compleja a la que entrenar y optimizar. Para esto, generamos diversos modelos en los cuales usamos una cantidad variada de neuronas, viendo a ojo como cambiaba el modelo. Además, usamos métodos de regulación, por sí solos y combinados, entre ellos Dropout, con el fin de evitar las fuertes codependencias, y Early Stopping, para frenar el entrenamiento antes de que el error del set de validación empiece a aumentar, busca el punto óptimo entre la varianza y el bias. Comenzamos usando como optimizador Adam, pero luego optamos por usar Nadam, ya que estos dos son similares, solo que este segundo hace uso de la corrección de Nesterov, es decir, tiene en cuenta la dirección del gradiente en el paso anterior, lo que puede generar una convergencia más rápida.

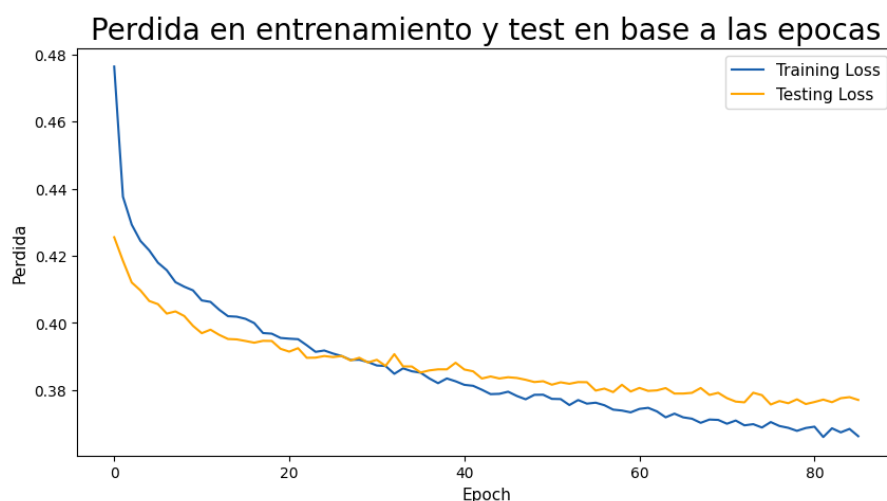
Dado que nuestro problema es de clasificación de clases excluyentes, usamos siempre en la capa de salida la función de activación sigmoidea. Usamos frecuentemente como función de pérdida la entropía cruzada binaria, dado que es una muy usada en problemas de clasificación binaria.

Debimos tomar decisiones como la cantidad de capas a usar, los optimizadores, los métodos de regulación, la cantidad de neuronas, entre otros, ya que no podíamos optimizar todo mediante k-folds. Optamos por optimizar con este método la cantidad de épocas, la cantidad de batchs y el learning rate del optimizador, usando el mejor modelo de los propuestos anteriormente como base. Usamos 5 folds, tomando 10 valores aleatorios de tres sets de parámetros de 3 valores cada uno. Este entrenamiento tomó una excesiva cantidad de tiempo, y consideramos que su ejecución no valió la pena.



Previo a tomar este camino, en un modelo más básico al que entrenamos con una cantidad de épocas muy elevada, realizamos un gráfico de la relación entre el Accuracy y la cantidad de épocas registrados en el historial del entrenamiento, y en base a ese máximo, entrenamos nuevamente el modelo.

Luego de crear todos nuestros modelos, concluimos que el mejor fue el que cuenta con dropout y early stopping, y que no es necesario un batchsize tan grande para obtener buenos resultados.



Vemos que justo antes de que frene el modelo, la pérdida en el set de test ya se veía estancada, por lo que posiblemente pronto subiría.