Checkpoint 4 - Grupo 25

Introducción

En este nuevo checkpoint nos dedicamos a explorar un nuevo modelo que antes no habíamos trabajado, el modelo de las redes neuronales. Este acercamiento nos derivó a la exploración de nuevas técnicas, como arquitectura de las mismas, la estandarización de los datos, optimización de hiper-parámetros, validación y testing.

Construcción del modelo

Los siguientes puntos describirán al mejor modelo obtenido:

- Arquitectura:
  - Capa entrada
    - La cantidad de columnas que posee el dataset. En nuestro caso 45 columnas.
    - La función de activación que se utilizó fue Relu.
    - Regularización escogida, Dropout.
    - Tipo de conexión: densa.
  - Capas ocultas
    - La cantidad de neuronas que elegimos en cada capa igualan a la cantidad de columnas que posee el dataset.
    - La función de activación elegida en todas las capas ocultas fue Relu.
    - Tipo de conexión: densa.
  - Capa de salida
    - La cantidad de neuronas iguala la cantidad de clases a predecir que en nuestro caso es solo 1 (una).
    - La función de activación para una sola neurona, considerándose que se trata de un caso binario, es sigmoidea.
- ¿Qué hiperparametros se optimizaron?

  Optimizamos bath_size y epochs ya que nos interesó saber hasta qué punto deberíamos entrenar la red y con cuantos datos era conveniente.

- ¿Qué optimizador se utilizó?

Utilizamos el optimizador Adam con un learning_rate de 0.001, beta_1 de 0.9 y beta_2 de 0.999.

- ¿Se utilizó alguna técnica de regularización? ¿Cuál?

  Utilizamos Dropout con un hiper parámetro de 0.2.
- ¿Cuántos ciclos de entrenamiento utilizó?

  Se utilizaron 100 epochs.

Cuadro de Resultados

Modelo	F1-Test	Presicion Test	Recall Test	Kaggle
modelo_1	0.82	0.84	0.79	0.81737
modelo_2	0.81	0.86	0.77	0.81358
modelo_3	0.81	0.83	0.80	0.81035
modelo_4	0.82	0.82	0.82	0.8238

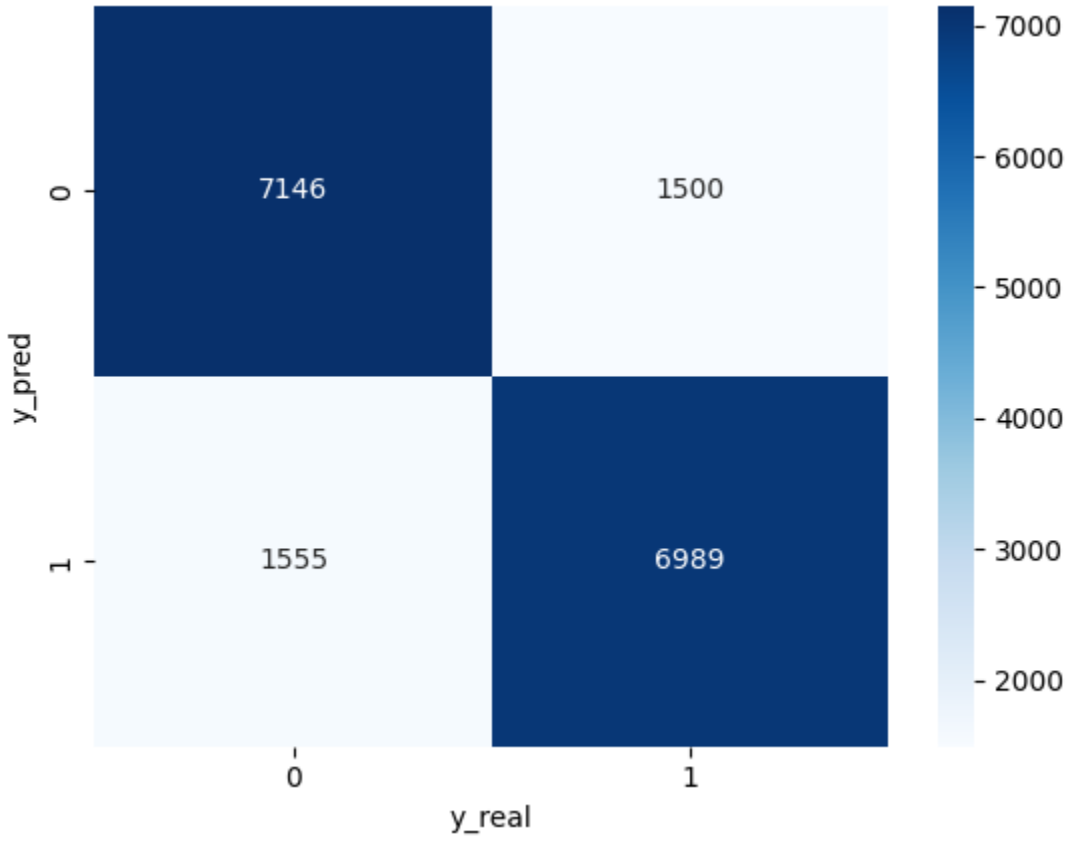
Todos los modelos descritos posteriormente contarán con la regularización Dropout, la capa de salida se activará con la función sigmoidea y con la cantidad de neuronas como con tantas clases se quieren predecir. En nuestro caso, es solo 1. La principal diferencia entre estos se basa en la cantidad de capas o de neuronas por capas.

- modelo_1: consiste en una arquitectura piramidal donde las capas ocultas tienen la mitad de neuronas que su anterior. A su vez la función de activación de cada capa consiste en la función ReLu.
- modelo_2: cuenta con una sola capa oculta activada por la función ReLu con la misma cantidad de neuronas como entradas tenga.
- modelo_3: cuenta con tres capas de neuronas. La primera capa oculta contiene la misma cantidad de neuronas que de entradas se tengan. La segunda contiene la mitad a su anterior y la tercera sigue con este proceso. La función de activación en todas estas, es la función ReLu.
- modelo_4: tiene 2 capas ocultas con la misma cantidad de neuronas en cada una de ellas como de entradas se tenga y ambas están activadas por la función ReLu.

El mejor modelo fue detectado realizando pruebas en Kaggle, resultó ser el modelo_4. Aunque obtuvimos un score de 0.8238 y eso nos permitió elegirlo, luego cuando mejoramos sus hiper-parámetros con Random Search, la predicción en

Kaggle disminuyó a 0.81624. Creemos que esto sea por sobreentrenamiento con nuestro conjunto de datos; las redes neuronales para conjuntos de datos pequeños tienden a sobreentrenarse. Incluso, hicimos uso de random search para mejorar el score de kaggle ampliando la cantidad de particiones para el cross validation pero aún así, ésta no aumentó.

Matriz de Confusion



Dada la matriz de confusión se puede detectar un 0.82 de precisión. La precisión nos indica en proporción de los que yo dije que eran positivos, a cuantos acertamos. Luego para la métrica de recall podemos observar que para la clase 0 tenemos un valor de 0.83 y para la clase 1, 0.82. Esta métrica nos indica en proporción de nuestros valores positivos a cuantos acerté.

Comentarios

Nos hubiera gustado probar mejores configuraciones para evitar el overfitting, como así también explorar en otras técnicas para reducir la capacidad de cómputo, ya que esto nos hubiera ayudado a tener un modelo un poco más acertado.

Tareas Realizadas

Integrante	Tarea
Mariana Juarez Goldemberg	Creación de modelos base Armado de informe final
Miranda Marenzi	Creación de modelos Búsqueda de mejores-hiper parámetros
Roman Lisandro	Armado de reporte Creación de modelos