

Trabajo Práctico 1 - CHP4

Reservas de Hotel

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

ALUMNO	PADRON	CORREO
BARTOCCI, Camila	105781	cbartocci@fi.uba.ar
LOURENGO, Lucía	104880	llourenco@fi.uba.ar
PATÍÑO, Franco	105126	fpatino@fi.uba.ar

Sobre los modelos construidos

Experimentamos con distintos tipos de arquitecturas de Redes Neuronales, variando la cantidad de capas, el número de neuronas en cada una, las funciones de activación utilizadas y los optimizadores, entre otras cosas. También probamos distintas cantidades de lotes y épocas a la hora de entrenar nuestros modelos, optimizando F1 score en todos los casos. Tras los primeros entrenamientos de los modelos más simples, pudimos notar que había señales de overfitting al comparar el score obtenido en el entrenamiento versus el obtenido en validación, presentando diferencias de hasta 10 puntos porcentuales en la métrica a optimizar. Por este motivo, introducimos parámetros de regularización, como L2 y L1 en algunas capas, evitando así el sobreajuste y la alta varianza en los modelos.

En cuanto a la métrica a optimizar, notamos que Keras no cuenta con un F1 score predefinido al momento de compilar las Redes Neuronales. Por este motivo, creamos nuestra propia función 'get_f1' para poder optimizarla en el entrenamiento, basándonos en la implementación que se muestra en [este artículo](#).

Por último, tuvimos en cuenta aplicar validación cruzada para optimizar los hiperparámetros del modelo, pero decidimos no apoyarnos en esta técnica para las Redes Neuronales debido al alto costo computacional y el largo tiempo que tomaba en ejecutarse, además de que no logramos obtener mejores métricas con validación cruzada frente a aquellas que logramos probando distintas combinaciones de hiperparámetros por nuestra cuenta. Nos enfocamos, en cambio, en el análisis de la evolución de la métrica a optimizar en el entrenamiento, analizando los gráficos de la métrica en entrenamiento versus la métrica en validación y aplicando early stopping (además de los parámetros de regularización mencionados anteriormente en algunas capas) cuando veíamos riesgo de overfitting.

Sobre el mejor modelo obtenido

La arquitectura de nuestro mejor predictor es de 3 capas ocultas, siendo la cantidad de neuronas 150, 100 y 30 respectivamente. La función de activación presente en las capas ocultas elegida fue ReLu y la función sigmoide en la capa de salida. El modelo fue entrenado en 100 épocas y con un batch size de 32. El optimizador utilizado es Nadam con learning rate de 0.001 y evaluando con la función de pérdida binary cross entropy. Los métodos de regularización aplicados fueron L2 y Dropout. En cuanto a sus métricas, obtuvimos:

- 82 % de Accuracy
- 83 % de Recall
- 81 % de Precision
- 82 % de F1 Score

En cuanto a las métricas obtenidas en entrenamiento vs en validación, vimos que la primera tuvo scores entre 81.4 % y 82.4 %, mientras que en la segunda obtuvimos entre 80 % y 81.5 %, presentando así una diferencia porcentual de pocos puntos, habiendo logrado reducir la varianza y evitar el overfitting. También analizamos el área bajo la curva ROC, siendo ésta de aproximadamente de 90, logrando una buena capacidad de discriminación entre instancias positivas y negativas.