

Checkpoint 4 - Redes neuronales

En este checkpoint, se buscó construir una red neuronal para clasificación, optimizando su rendimiento mediante la búsqueda de una arquitectura e hiper parámetros adecuados.

Comenzamos importando los datasets y preprocesándolos como se viene haciendo en las entregas anteriores, esta vez agregando una normalización usando el escalado estándar.

Luego armamos una red neuronal secuencial de cuatro capas densas. La primera es una capa oculta compuesta por 100 neuronas, y tiene una función de activación sigmoideal. La segunda es una capa de dropout que desactiva el 20% de las unidades de la capa anterior durante el entrenamiento. La tercera es otra capa oculta de 50 neuronas, y la última contiene 100 neuronas, utiliza una función de activación tangente hiperbólica (tanh) y aplica una regularización L2.

Se probaron cuatro optimizadores, cambiando los hiperparametros para llegar al mejor resultado, y se obtuvo el siguiente F1 Score con cada uno:

- SGD: 0.7504
- Adam: 0.7986
- Adadelta: 0.7502
- RMSprop: 0.8011

Después armamos una segunda red neuronal optimizada utilizando Keras Tuner para la búsqueda de hiper parámetros, donde variamos la función de activación de la capa de entrada, función de activación de las capas ocultas, la función de pérdida, la tasa de aprendizaje, el dropout, el regularizador, el número de unidades en la capa de entrada, las capas ocultas y la capa de salida y el optimizador. A continuación se puede observar la arquitectura conseguida de manera dinámica usando keras-tuner y su matriz de confusión.

```
model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 192)	8064
dropout_4 (Dropout)	(None, 192)	0
dense_7 (Dense)	(None, 32)	6176
dropout_5 (Dropout)	(None, 32)	0
dense_8 (Dense)	(None, 2)	66

=====
Total params: 14,306
Trainable params: 14,306
Non-trainable params: 0

