

## Práctico N° 3

### Machine Learning

#### Ejercicios obligatorios

1. Familiarizarse con el código de la red neuronal feedforward fully connected de 1 capa oculta explicada en clase
2. Modificar el programa para que
  - a. Mida la precisión de clasificación (accuracy) además del valor de Loss
  - b. Utilice un conjunto de test independiente para realizar dicha medición (en lugar de utilizar los mismos datos de entrenamiento). Este punto requiere generar más ejemplos.
3. Agregar parada temprana, utilizando un conjunto de validación, distinto del conjunto de entrenamiento y de test (este punto requiere generar más ejemplos). Esto es: verificar el valor de loss o de accuracy cada N epochs (donde N es un parámetro de configuración) utilizando el conjunto de validación, y detener el entrenamiento en caso de que estos valores hayan empeorado (puede incluirse una tolerancia para evitar cortar el entrenamiento por alguna oscilación propia del proceso de entrenamiento).
4. Experimentar con distintos parámetros de configuración del generador de datos para generar sets de datos más complejos (con clases más solapadas, o con más clases). Alternativamente experimentar con otro generador de datos distinto (desarrollado por usted). Evaluar el comportamiento de la red ante estos cambios.
5. Modificar el programa para que funcione para resolver problemas de regresión
  - a. Debe modificarse la función de pérdida y sus derivadas, utilizando por ejemplo MSE
  - b. Debe crearse un generador de datos nuevo para que genere datos continuos (pueden mantenerse igualmente 2 entradas; en caso de usar más entradas puede requerirse más capas en la red neuronal)
6. Realizar un barrido de parámetros (learning rate, cantidad de neuronas en la capa oculta, comparación de ReLU con Sigmoide)

#### Ejercicios opcionales

7. Re-implementar la red feedforward fully connected entera
8. Desarrollar analíticamente e implementar una capa oculta adicional
9. Probar la red neuronal con sets de datos tomados de alguna de las siguientes fuentes
  - a. Machine Learning Repository de la Universidad de California en Irvine ([link](#))
  - b. Kaggle ([link](#))
10. Implementar características avanzadas en la red neuronal (regularización, otros algoritmos de optimización de pesos como Momentum o Adam, Dropout, etc)
11. Familiarizarse con el uso de la biblioteca Keras con Tensorflow e implementar una red Feedforward fully connected similar a la vista con esta biblioteca