

# Tarea 1

## Redes Neuronales

Estoy utilizando la base de datos de Iris, donde con datos de determinado tipo de planta se clasifica la subespecie de la misma.

Los datos corresponden al alto y el peso del tallo y las hojas, y hay tres categorías posibles.

El repositorio de Github utilizado es:

<https://github.com/Afromatico/RedesNeuronales.git>

para correr el programa, basta desde el shell invocar `python2 main.py`, este comando corre los experimentos con los datos, crea la red etc.

Para utilizar el programa se necesitan las siguientes librerías:

`matplotlib`

`sklearn`

`pandas`

`numpy`

Si se quiere editar el código, en el archivo `main.py` la línea `per = MLPerseptron(0.01,[4, 7, 3])` es la que define la red neuronal. para modificar el learning rate basta cambiar el primer parámetro.

No note un gran efecto con el número de capas ocultas respecto al learning rate, lo único destacable fue que la precisión aumentó notablemente para mayor cantidad de experimentos y un learning rate más pequeño del orden del 0.01%, donde llegue incluso a 98% de precisión.

Mi red para 4 neuronas de input, 7 en la capa oculta y 3 en la salida del clasificador demora para analizar un dataset e 150 elementos 1000 veces menos de un minutos, en torno a las 20 segundos sin optimizaciones por gpu.

Para learning rates muy altos note que es muy probable que se pierda el avance logrado en la precisión, en cambio dado que son categorías simples conviene no arriesgar tanto y avanzar de a poco hacia el mínimo error posible.

En el caso de cómo diseñe la red el orden no afecta el producto dado que cada epoch se calcula en un solo paso y se actualizan los pesos de la red en conjunto considerando todos los datos del dataset. Aun así si afecta los valores que tomen los parámetros en primera instancia los pesos y el bias de la red.