

# Redes Neuronales – Curso 2020

---

## Actividad 2

### Adaline

#### Ejercicio 1

Se desea entrenar un asociador lineal que pueda resolver el problema de convertir un número binario de tres bits a su correspondiente número decimal:

X1	X2	X3	T
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

- a) Seleccione cualquier  $W$  inicial y ejecute manualmente el algoritmo de entrenamiento del asociador lineal sin utilizar un  $W_0$ . Utilice un valor de alfa de 0.5. ¿Cuántas iteraciones fueron necesarias para encontrar la respuesta?
- b) Idem a) pero usando un valor de  $W_0$ . ¿Cuál es la diferencia con el resultado obtenido en el inciso anterior?
- c) ¿Cuál cree que es la explicación para el fenómeno observado en los incisos anteriores? Justifique su respuesta.

#### Ejercicio 2

Si luego del entrenamiento de un Adaline su vector de pesos quedó con estos valores:

$$\langle W_0 = -3.32; W_1 = 2.15; W_2 = 4.72 \rangle$$

Sabiendo que se entrenó utilizando un conjunto de datos correspondientes a margaritas (clase 0) y rosas (clase 1).

Del siguiente dataset ¿Cuáles son margaritas y cuáles rosas?

x1	x2
0.4	0.54
0.3	0.57
0.6	0.42

### Ejercicio 3

Utilizando las imágenes del archivo `imagenes.zip` qué diferencia observa al entrenar una neurona no lineal si al dataset se le aplican las siguientes transformaciones:

- a) Se escalan las variables en un rango 0..1
- b) Los datos se ordenan, primero los datos de una clase, luego los de la otra y se presentan al perceptrón en ese orden.
- c) ¿Qué características tiene que tener el dataset utilizado para el entrenamiento de un adaline?
- d) ¿Qué diferencias observa entre los distintos resultados encontrados con cada imagen?
- e) Si el dataset cumple con las características mencionadas en c) ¿Cuántas soluciones posibles puede encontrar un adaline en distintos entrenamientos con el mismo dataset?
- f) ¿Encuentra diferencias entre entrenar la neurona no lineal con la función *logsig* y la función *tansig*?
- g) ¿Podría resolver estos mismos problemas con la función *puerlin*? ¿Si fuera un problema de clasificación tendría sentido usar *purelin*?

### Ejercicio 4

Del ejercicio anterior tome los pesos finales resultado de algún entrenamiento de cualquiera de las imágenes (`w_final`, `b_final`). Invierta las clases de los datos, es decir los datos de la clase 0, ahora serán de la clase 1 y los datos de la clase 1 ahora serán los de la clase 0.

$$T2 = \text{abs}(T-1)$$

Entrene un nuevo Adaline con la misma imagen (con las clases invertidas) y los pesos resultado del entrenamiento anterior.

```
train_con_pesos(P, T2, alfa, max_ite, cota, funcion, True,  
                w_final, b_final)
```

- a) ¿Qué observa durante el entrenamiento?
- b) ¿Cómo puede explicar ese fenómeno?
- c) Entrene un nuevo Adaline con el dataset con las clases invertidas utilizado en el ensayo anterior, pero usando una inicialización de pesos al azar. ¿Qué relación encuentra entre los pesos finales del entrenamiento original (`w_final`, `b_final`) y los obtenidos en este ensayo?

### Ejercicio 5

Utilice la base de datos `radar_muestras.csv` de la actividad anterior para entrenar una neurona no lineal que clasifique los datos de ambas rutas aéreas. ¿Qué diferencias observa en el resultado comparado con el conseguido con el perceptrón? ¿Qué sucede al presentarle a la neurona no lineal los datos del dataset `radar_desconocidos`?

## Ejercicio 6

Entrene un adaline para realizar una regresión entre las variables del dataset R\_train.csv. Separe el dataset en subconjuntos train y test.

- a) ¿En cuántas iteraciones converge el adaline?
- b) ¿Cuál es el error promedio y el error absoluto entre los ejemplos pertenecientes al subconjunto de train?
- c) ¿Cuál es el error promedio y el error absoluto entre los ejemplos pertenecientes al subconjunto de test?

## Ejercicio 7

Dadas las siguientes clases y sus ejemplos:

Clase A = { (1, 3) , (1.5, 4), (2, 3), (1.5, 2.5), {2.5, 3) }

Clase B = { (2.5, 4) , (2.5, 2), (3, 3), (3, 2.5), (1.5, 5), (2, 4.5) }

Plantee una arquitectura (neuronas y como se conectan) de un multiperceptrón capaz de resolver el problema de clasificar las dos clases y responda a las siguientes preguntas.

- a) ¿Cuántas neuronas posee la arquitectura planteada?
- b) ¿Cuál es la tarea asignada a neurona?
- c) ¿Cuáles son los patrones de entrada y su salida esperada para cada neurona?

## Ejercicio 8

El archivo "MP\_Training.csv" contiene 75 registros, los cuales están divididos en tres clases. Se desea formar un modelo de los datos que permita hacer la clasificación de nuevas muestras en una de las tres clases posibles. Este modelo sólo puede estar formado por neuronas no lineales.

Sabiendo que una neurona no lineal sólo divide los datos en dos clases se deberá armar una arquitectura formada por dos neuronas que logre el objetivo pedido.

Por lo tanto para conseguir este modelo de clasificación deberá seguir los siguientes pasos:

- a) Entrenar una neurona no lineal que permita separar los datos de una de las clases de las dos restantes.  
La neurona no lineal entrenada será capaz de clasificar una nueva muestra como perteneciente a una clase o bien a las otras dos.  
Preste atención que el valor asociado a la clase (valor esperado) está relacionado con la función de salida utilizada (logsig o tansig). Por ello al separar los dos subconjuntos deberá renombrar las clases a 0 y 1 (o -1 y 1) según la función utilizada.
- b) Repita el paso anterior con el subconjunto restante.
- c) ¿Cómo es la arquitectura de la red de neuronas hallada? Recuerde que la arquitectura es la cantidad de neuronas, como se conectan entre ellas, qué pesos tiene cada neurona y cual es la función de activación de cada neurona.
- d) ¿Qué clases de datos (A, B o C) separa cada una de las neuronas entrenadas?
- e) Utilice el modelo encontrado para clasificar los cuatro registros que se encuentran en el archivo MP\_Unknown.csv

**ACLARACIÓN:** Las tres clases de la base de datos “MP\_Training.csv” pueden ser separadas linealmente en forma de cascada. Por lo tanto, deberá asegurarse, que cada neurona entrenada separe correctamente el 100% de los ejemplos que le correspondan.