INTELIGENCIA ARTIFICIAL

TRABAJO PRÁCTICO n°3

MPL

(Multilayer Perceptron)

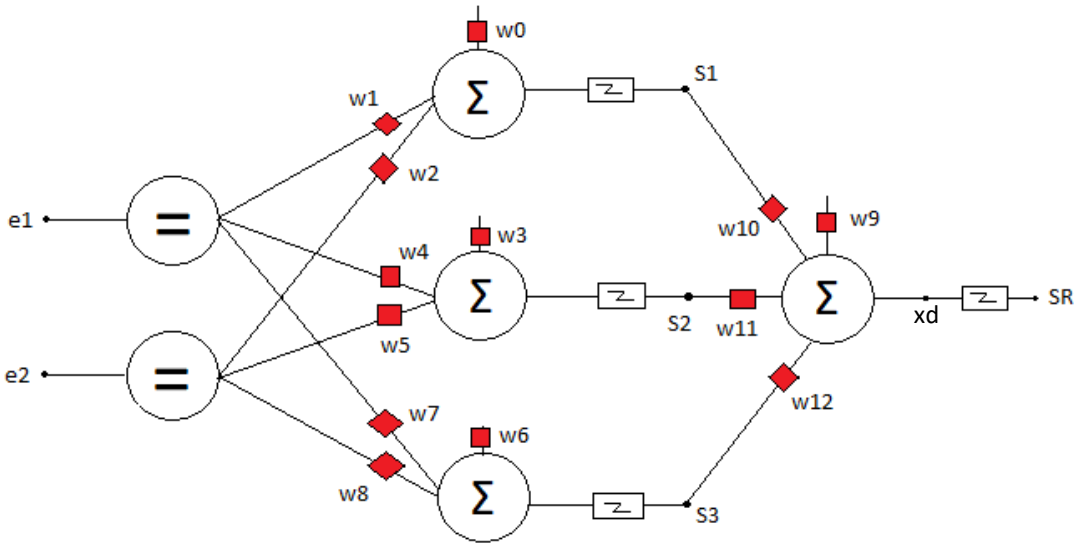
Alumna: M. Valentina Scalco

Ingeniería en Informática

2022

Consignas

1- Obtener la salida, asignando los pesos que vimos en clase, para la tabla de verdad XOR, primera iteración, primer registro.
2- Entrenamiento para que se comporte como una compuerta XOR.



w0 = 0.9	w3 = 0.3	w6 = 0.8	w9 = -0.23	w12 = 0.6
w1 = 0.7	w4 = -0.9	w7 = 0.35	w10 = -0.79	
w2 = 0.5	w5 = -1	w8 = 0.1	w11 = 0.56	

Resolución

1- Para la primera iteración los valores serán:

```
ya: 0.7109495026250039
yb: 0.574442516811659
yc: 0.6899744811276125
xd: -0.05597760898265647
Salida real: 0.48600925088941466
```

"Ya" es correspondiente a S1, "yb" a S2 e "yc" a S3.

Para obtener esto realizamos la sumatoria entre los pesos y sus respectivos valores de la tabla xor, excepto para w0, w3, w6 y w9, los cuales siempre se multiplican por uno.

```
xa = (self.w[1] * self.valor_e1[i]) + (self.w[2] * self.valor_e2[i]) + (self.w[0] * self.valor_w0)
ya = 1 / (1 + e**(-xa))
xb = (self.w[4] * self.valor_e1[i]) + (self.w[5] * self.valor_e2[i]) + (self.w[3] * self.valor_w0)
yb = 1 / (1 + e**(-xb))
xc = (self.w[7] * self.valor_e1[i]) + (self.w[8] * self.valor_e2[i]) + (self.w[6] * self.valor_w0)
yc = 1 / (1 + e**(-xc))
xd = (self.w[9] * self.valor_w0) + (self.w[10] * ya) + (self.w[11] * yb) + (self.w[12] * yc)
# Salida Real (yd)
yd = 1 / (1 + e**(-xd))
```

2- Los valores iniciales que tenemos serán los de los pesos, desde w0 hasta w12 y la tabla de la compuerta XOR.

```
def initialize_game(self):
    self.w = [0.9, 0.7, 0.5, 0.3, -0.9, -1,
              0.8, 0.35, 0.1, -0.23, -0.79,
              0.56, 0.6]
    self.valor_w0 = 1

def compuerta_XOR(self):
    self.valor_e1 = [0, 0, 1, 1]
    self.valor_e2 = [0, 1, 0, 1]
    self.salida_d = [0, 1, 1, 0]
    self.resolucion()
```

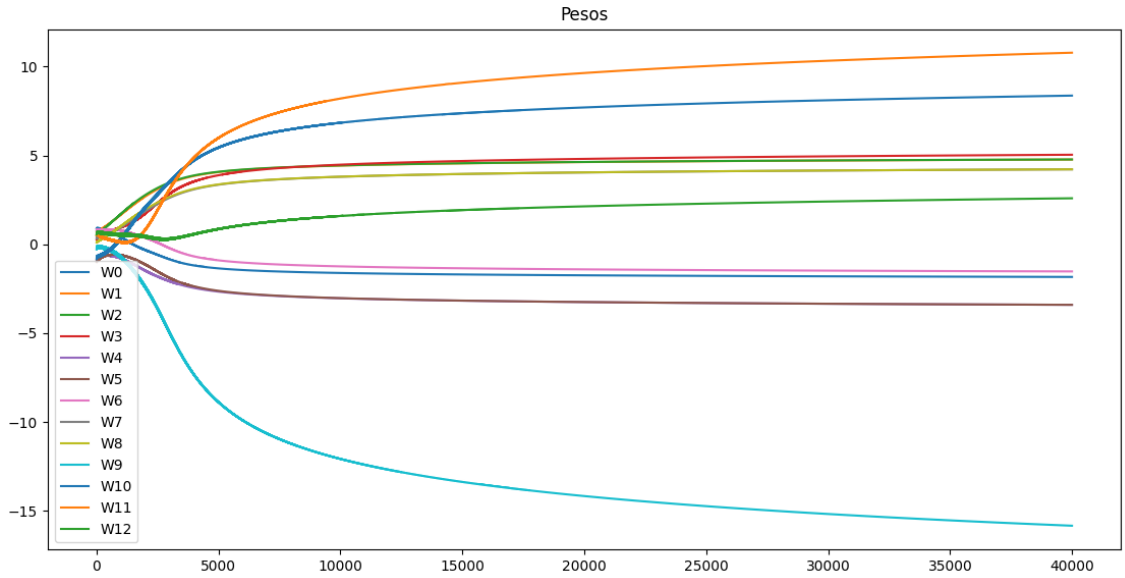
Para resolver o entrenarlo, lo recorremos 10000 veces, por lo que, pondremos un for para realizar esta tarea. Cada iteración, será pasar por los 4 valores de la tabla, eso quiere decir que debemos tener un for para recorrer esos cuatro valores. El valor de Lr que tomaremos será de 0,5.

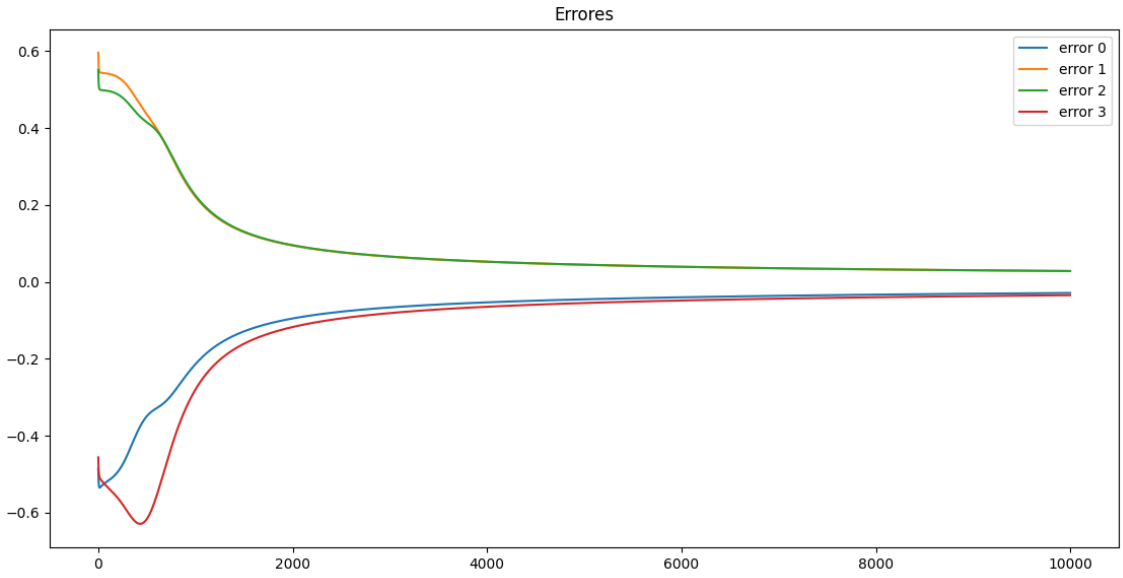
Primero sacamos x e y para cada una de las neuronas. Y obtendremos la salida real. Luego sacamos los errores y calculamos los nuevos pesos.

Una vez que terminemos el entrenamiento, graficaremos el cambio de los pesos y de los errores.

RESULTADO EN TERMINAL

```
Salida real para salida deseada 0 : 0.02890450775505159
Salida real para salida deseada 1 : 0.9719575196912781
Salida real para salida deseada 1 : 0.9719136443546669
Salida real para salida deseada 0 : 0.03476136411886884
Entrenamiento realizado: 10000 veces
```





NOTA: El trabajo fue desarrollado en python 3.