clase_perceptron

August 17, 2019

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
```

1 Construcción de un perceptrón utilizando una clase.

- Vamos a construir una clase llamada perceptrón.

```
In [2]: class perceptron:
            def __init__(self, nentradas,  alfa):
                self.nentradas = nentradas
                self.entradas = np.array([1])
                self.pesos = np.random.randn(nentradas)
                self.salidad = 0
                self.alfa=alfa
                self.grad_pesos=self.pesos

            def calcular_salida(self, entradas):
                self.salida = 1*(self.pesos.dot(entradas)>0) # función step()
                self.entradas = entradas

            def actualizar_pesos(self, salidad):
                for i in range(0, self.nentradas):
                    self.pesos[i]=self.pesos[i]+self.alfa*(salidad-self.salida)*sel
```

2 Cargamos los datos de entrenamiento.

```
In [3]: ejemplos = np.array([[1., 0., 0.],
                             [1, 0., 1.],
                             [1, 1., 0.],
                             [1, 1., 1.]]) # Entradas de la función OR
```

```
In [4]: y = np.array([[0.], [1.], [1.], [1.]]) # Salidas de la función OR
```

3 Programa principal.

```
In [5]: # Llamamos a la clase perceptron para construir un perceptron de tres entra
        # El coeficiente de aprendizaje lo fijamos en 0.5.
```

```
In [6]: neurona=perceptron(3,  0.6)

In [7]: neurona.calcular_salida(ejemplos[1])
        print(ejemplos[1])

[1. 0. 1.]
```

4 Entrenamiento del perceptrón.

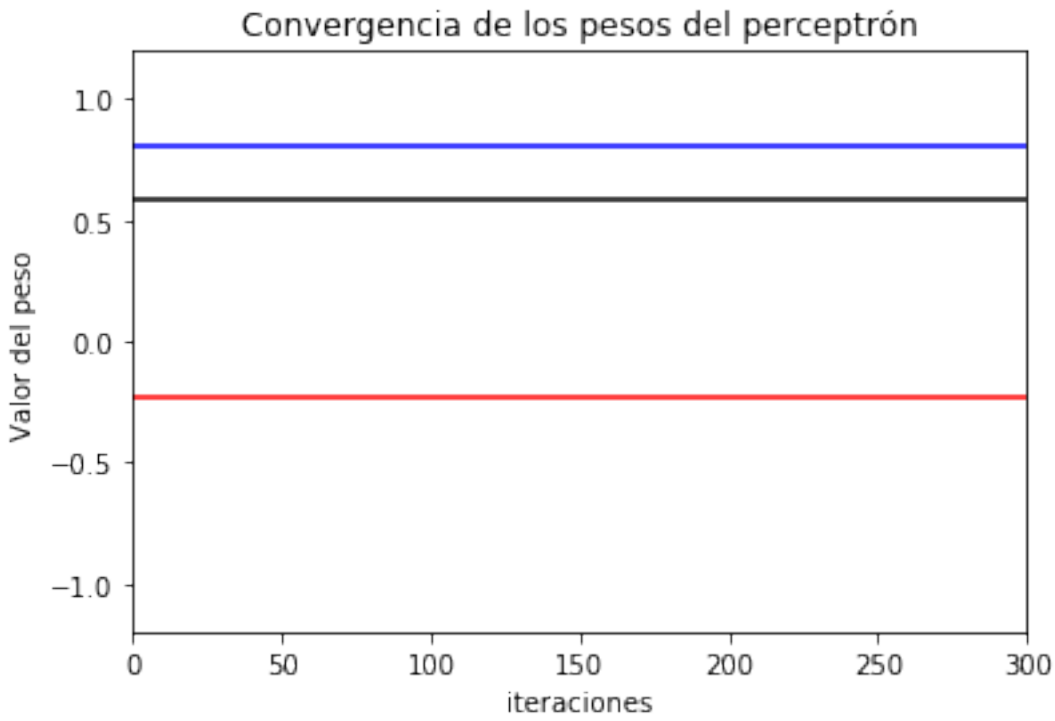
1. Hacemos un forward pass y calculamos la salida y.
2. Actualizamos los pesos.
3. Repetimos hasta que los pesos converjan.

```
In [8]: # Guardar el histórico de los pesos
        grad_pesos=[neurona.pesos]

In [9]: epoch=100;
        for j in range(0,epoch):
            for i in range(0, 3):
                neurona.calcular_salida(ejemplos[i])
                neurona.actualizar_pesos(y[i])
                grad_pesos=np.concatenate((grad_pesos,[neurona.pesos]),axis = 0)

In [10]: plt.plot(grad_pesos[:,0],'r')
         plt.plot(grad_pesos[:,1],'b')
         plt.plot(grad_pesos[:,2],'k')
         plt.axis([0, 300, -1.2, 1.2])
         plt.title('Convergencia de los pesos del perceptrón')
         plt.ylabel('Valor del peso')
         plt.xlabel('iteraciones')
         plt.rc('axes', labelsize=16)

         plt.show()
```




```
In [11]: for i in range(0, 4):
             neurona.calcular_salida(ejemplos[i])
             print(neurona.salida)
```

```
0
1
1
1
```