A perceptron implementation

Brief history of neural networks and a perceptron implementation report complement

Brandon Marquez Salazar

I. Code implementation

We'll use numpy as a very common package used for scientific computation.

```
import numpy as np
```

Now, in order to implement a perceptron we have to define the three steps needed

A. Neuron Model

First the perceptron output, which is the most important element, it's the neuron model. It will receive a vector of inputs X_{jk} a vector of weights W_{jk} and a reference for the activation function $S(\cdot)$.

```
def perceptronOutput(W,X,bias,ActivationFunction):
  WeighedInputs = np.dot(W,X)
  Net = WeighedInputs - bias
  O = ActivationFunction(Net)
  return O
```

B. Error equation

This function computes the error of the perceptron based on its output compared to the desired output. It receives the perceptron output vector O_j, the desired output vector Y_j and the number of patterns N.

```
def computeError(Y,O,N):
  DeltaSum = np.sum(np.abs(Y-O))
  Err = DeltaSum/N
  return Err
```

C. Weights update function

This function receives the current weights vector W_j, the perceptron and desired output vectors O_j and Y_j and learning rate r. Then returns the new weights vector.

```
def updateWeights(Y,O,W,X,r):
  NewWeights = W - (Y-O)*r
  return  NewWeights
```