Perceptrons - Making Predictions

Assuming that we have

```
weights = [-0.1, 0.206, -0.234]
```

In [1]:

```
def predict2(X, w):
    activation = w[0] + w[1]* X[0] + w[2]* X[1]
    if activation >= 0.0:
        return 1.0
    else:
        return 0.0
```

In [2]:

```
# test predictions
dataset = [[2.7810836,2.550537003,0],
    [1.465489372,2.362125076,0],
    [3.396561688,4.400293529,0],
    [1.38807019,1.850220317,0],
    [3.06407232,3.005305973,0],
    [7.627531214,2.759262235,1],
    [5.332441248,2.088626775,1],
    [6.922596716,1.77106367,1],
    [8.675418651,-0.242068655,1],
    [7.673756466,3.508563011,1]]
```

In [3]:

```
weights = [-0.1, 0.206, -0.234]
```

In [4]:

```
for row in dataset:
    prediction = predict2(row, weights)
    print("Expected={}, Predicted={}".format(row[-1], prediction))
```

```
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
```

```
* More fancy !
```

In [1]:

```
def predict(row, weights):
    activation = weights[0]
    for i in range(len(row)-1):
        activation += weights[i + 1] * row[i]
    return 1.0 if activation >= 0.0 else 0.0
```

In [10]:

```
for row in dataset:
    prediction = predict(row, weights)
    print("Expected={}, Predicted={}".format(row[-1], prediction))
```

```
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=0, Predicted=0.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
Expected=1, Predicted=1.0
```

References

https://machinelearningmastery.com/implement-perceptron-algorithm-scratch-python/
(https://machinelearningmastery.com/implement-perceptron-algorithm-scratch-python/)