Perceptrons - Making Predictions

Assuming that we have

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

In [1]:

```
def predict2(X, w):
    bias = w[0]
    activation = bias + 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]:

```
bias = -0.1
w0 = 0.206
w1 = -0.234
weights = [bias, w0, w1]
print(weights)
```

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

In [4]:

```
for row in dataset:
    print(row[0], row[1], row[2], row[-1])
```

```
2.7810836 2.550537003 0 0
1.465489372 2.362125076 0 0
3.396561688 4.400293529 0 0
1.38807019 1.850220317 0 0
3.06407232 3.005305973 0 0
7.627531214 2.759262235 1 1
5.332441248 2.088626775 1 1
6.922596716 1.77106367 1 1
8.675418651 -0.242068655 1 1
7.673756466 3.508563011 1 1
```

In [5]:

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

Plot the results

In [6]:

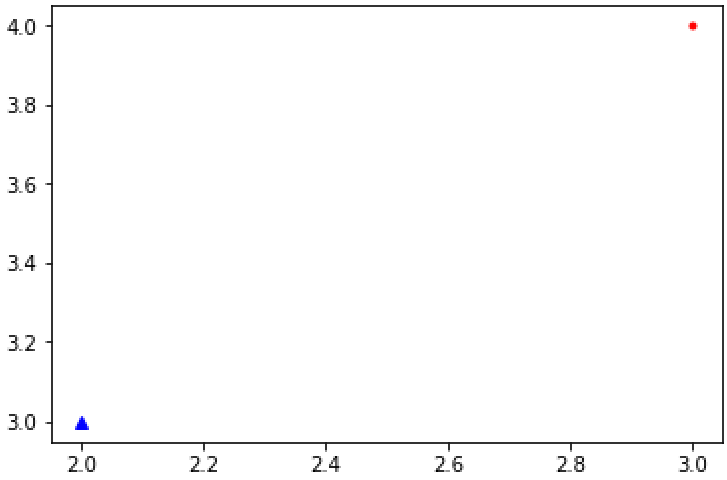
```
import matplotlib.pyplot as plt
```

In [7]:

```
plt.plot(3,4,'r.')
plt.plot(2,3,'b^')
```

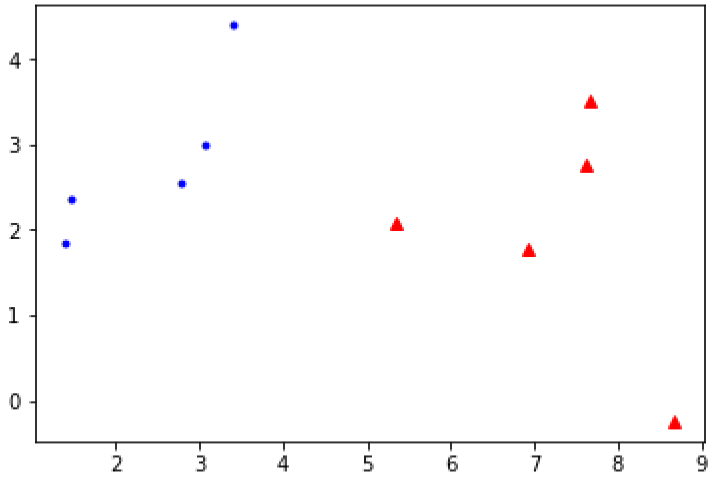
Out[7]:

```
[<matplotlib.lines.Line2D at 0x10eb61400>]
```



In [8]:

```
for row in dataset:
    prediction = predict2(row, weights)
    answer = row[-1]
    if prediction ==0:
      plt.plot(row[0], row[1],'b.')
    else:
      plt.plot(row[0], row[1],'r^')
```



- Random test(임의의 테스트 수행)

In [9]:

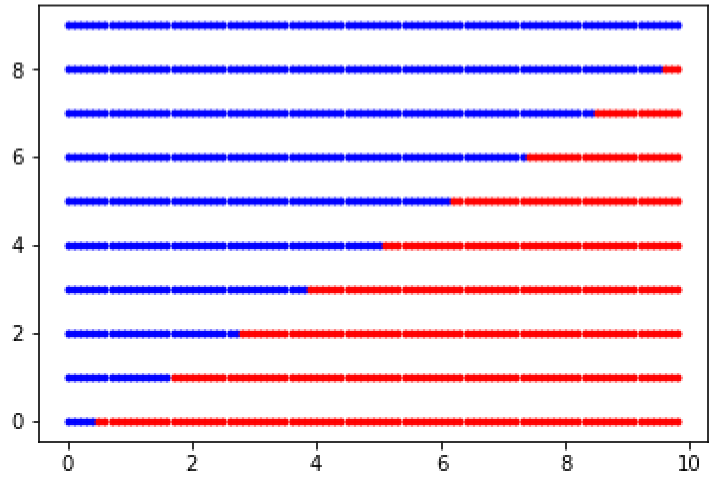
```
pred = predict2([3.8, 3], weights)
print(pred)
```

```
0.0
```

In [10]:

```
for j in range(10):
  for i in range(99):
    x = i * 0.1
    y = j
    pred = predict2([x, y], weights)

    if pred ==0:
        plt.plot(x, y,'b.')
    else:
        plt.plot(x, y,'r.')
```



References

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