Perceptrons - Making Predictions

Creating a gate with Perceptron

In [1]:

```
import numpy as np
```

AND Gate

In [27]:

```
def AND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.7
    tmp = w[0]*x[0] + w[1]*x[1] + b # tmp = np.sum(w*x) + b

    if tmp <= 0:
        return 0
    else:
        return 1
```

In [28]:

```
AND(1,0)
```

Out[28]:

```
0
```

In [29]:

```
AND(1,1)
```

Out[29]:

```
1
```

In [30]:

```
AND(0,0)
```

Out[30]:

```
0
```

In [31]:

```
AND(0,1)
```

Out[31]:

```
0
```

NAND Gate

In [32]:

```
def NAND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([-0.5, -0.5])
    b = 0.7
    tmp = w[0]*x[0] + w[1]*x[1] + b
    if tmp <= 0:
        return 0
    else:
        return 1
```

In [33]:

```
NAND(0,0)
```

Out[33]:

```
1
```

In [34]:

```
NAND(1,0)
```

Out[34]:

```
1
```

In [35]:

```
NAND(1,1)
```

Out[35]:

```
0
```

OR Gate

In [36]:

```
def OR(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.2
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1
```

In [37]:

```
OR(0,0)
```

Out[37]:

```
0
```

In [38]:

```
OR(1,0)
```

Out[38]:

```
1
```

In [39]:

```
OR(0,1)
```

Out[39]:

```
1
```

In [40]:

```
OR(1,1)
```

Out[40]:

```
1
```

XOR Gate

In [51]:

```
def XOR(x1, x2):
    s1 = NAND(x1, x2)
    s2 = OR(x1, x2)
    y = AND(s1, s2)
    return y
```

In [52]:

```
XOR(0,0)
```

Out[52]:

```
0
```

In [53]:

```
XOR(1,1)
```

Out[53]:

```
0
```

In [54]:

```
XOR(0,1)
```

Out[54]:

1

In [55]:

```
XOR(1,0)
```

Out[55]:

1

XOR cannot be expressed as a single layer Perceptron.