Perceptrons - Making Predictions

Creating a gate with Perceptron

In [1]:

```
import numpy as np
```

AND Gate

In [2]:

```
def AND(x1, x2):
    x = np.array([x1, x2])
    #w = np.array([0.1, 0.9])
    w = np.array([0.5, 0.5])
    b = -0.7#-2
    #activation 계산
    tmp = w[0]*x[0] + w[1]*x[1] + b # tmp = np.sum(w*x) + b
    print(tmp)
    if tmp <= 0:
        return 0
    else:
        return 1
```

In [3]:

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

```
-0.19999999999999996
```

Out[3]:

```
0
```

In [4]:

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

```
0.30000000000000004
```

Out[4]:

```
1
```

In [5]:

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

```
-0.7
```

Out[5]:

```
0
```

In [6]:

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

```
-0.19999999999999996
```

Out[6]:

```
0
```

NAND Gate

In [7]:

```
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 [8]:

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

Out[8]:

```
1
```

In [9]:

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

Out[9]:

```
1
```

In [10]:

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

Out[10]:

```
0
```

OR Gate

In [11]:

```
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 [12]:

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

Out[12]:

0

In [13]:

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

Out[13]:

1

In [14]:

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

Out[14]:

1

In [15]:

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

Out[15]:

1

XOR Gate

In [16]:

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

In [17]:

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

```
-0.19999999999999996
```

Out[17]:

```
0
```

In [18]:

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

```
-0.19999999999999996
```

Out[18]:

```
0
```

In [19]:

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

```
0.30000000000000004
```

Out[19]:

```
1
```

In [20]:

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

```
0.30000000000000004
```

Out[20]:

```
1
```

XOR cannot be expressed as a single layer Perceptron.

In [21]:

```
def AND2(x1, x2):
    if x1 == 1 and x2 ==1:
      return 1
    elif x1 ==1 and x2 ==0 :
      return 0
    elif x1 ==0 and x2 ==1 :
      return 0
    else:
      return 0
```