

- Name :: Rishabh Patil
- SAP :: 60009200056
- Div :: K/K2

import numpy as np

### Step & Perceptron Fn

```
def step(v):
    if v >= 0:
        return 1
    else:
        return 0

def perceptron(x, w, b):
    v = np.dot(w, x) + b
    y = step(v)
    return y
```

### AND Gate

```
def AND_percep(x):
    w = np.array([1, 1])
    b = -2
    return perceptron(x, w, b)
```

```
example1 = np.array([1, 1])
example2 = np.array([1, 0])
example3 = np.array([0, 1])
example4 = np.array([0, 0])
```

```
print("AND({}, {}) = {}".format(1, 1, AND_percep(example1)))
print("AND({}, {}) = {}".format(1, 0, AND_percep(example2)))
print("AND({}, {}) = {}".format(0, 1, AND_percep(example3)))
print("AND({}, {}) = {}".format(0, 0, AND_percep(example4)))
```

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

### OR Gate

```
def OR_percep(x):
    w = np.array([1, 1])
    b = -1
    return perceptron(x, w, b)
```

### # Test

```
example1 = np.array([1, 1])
example2 = np.array([1, 0])
example3 = np.array([0, 1])
example4 = np.array([0, 0])
```

```

print("OR({}, {}) = {}".format(1, 1, OR_percep(example1)))
print("OR({}, {}) = {}".format(1, 0, OR_percep(example2)))
print("OR({}, {}) = {}".format(0, 1, OR_percep(example3)))
print("OR({}, {}) = {}".format(0, 0, OR_percep(example4)))

```

```

OR(1, 1) = 1
OR(1, 0) = 1
OR(0, 1) = 1
OR(0, 0) = 0

```

### NOT Gate

```

def NOT_percep(x):
    return perceptron(x, w=-1, b=0.5)

```

```

print("NOT(0) = {}".format(NOT_percep(0)))
print("NOT(1) = {}".format(NOT_percep(1)))

```

```

NOT(0) = 1
NOT(1) = 0

```

### NAND Gate

```

def NAND_percep(x):
    a=AND_percep(x)
    b=NOT_percep(a)
    return b

```

*# Test*

```

example1 = np.array([1, 1])
example2 = np.array([1, 0])
example3 = np.array([0, 1])
example4 = np.array([0, 0])
print("NAND({}, {}) = {}".format(1, 1, NAND_percep(example1)))
print("NAND({}, {}) = {}".format(1, 0, NAND_percep(example2)))
print("NAND({}, {}) = {}".format(0, 1, NAND_percep(example3)))
print("NAND({}, {}) = {}".format(0, 0, NAND_percep(example4)))

```

```

NAND(1, 1) = 0
NAND(1, 0) = 1
NAND(0, 1) = 1
NAND(0, 0) = 1

```

### NOR Gate

```

def NOR_percep(x):
    a=OR_percep(x)
    b=NOT_percep(a)
    return b

```

*# Test*

```

example1 = np.array([1, 1])
example2 = np.array([1, 0])
example3 = np.array([0, 1])

```

```

example4 = np.array([0, 0])
print("NOR({}, {}) = {}".format(1, 1, NOR_percep(example1)))
print("NOR({}, {}) = {}".format(1, 0, NOR_percep(example2)))
print("NOR({}, {}) = {}".format(0, 1, NOR_percep(example3)))
print("NOR({}, {}) = {}".format(0, 0, NOR_percep(example4)))

```

```

NOR(1, 1) = 0
NOR(1, 0) = 0
NOR(0, 1) = 0
NOR(0, 0) = 1

```

## XOR Gate

```

def XOR(x):
    return AND_percep(np.array([NAND_percep(x), OR_percep(x)]))
print("XOR({}, {}) = {}".format(1, 1, XOR(example1)))
print("XOR({}, {}) = {}".format(1, 0, XOR(example2)))
print("XOR({}, {}) = {}".format(0, 1, XOR(example3)))
print("XOR({}, {}) = {}".format(0, 0, XOR(example4)))

```

```

XOR(1, 1) = 0
XOR(1, 0) = 1
XOR(0, 1) = 1
XOR(0, 0) = 0

```