sc-lab1

August 12, 2024

[1]:
```
def MP_neuron(inputs, weights, bias):

    weighted_sum = sum(w * inp for w, inp in zip(weights, inputs))

    return 1 if weighted_sum >= bias else 0

inputs_list = [
    [0, 0],
    [0, 1],
    [1, 0],
    [1, 1]
]


print("AND Gate:")
weights = [1, 1]
bias = 2
for inputs in inputs_list:
    print(f"Inputs: {inputs}, Output: {MP_neuron(inputs, weights, bias)}")


print("\nOR Gate:")
weights = [1, 1]
bias = 1
for inputs in inputs_list:
    print(f"Inputs: {inputs}, Output: {MP_neuron(inputs, weights, bias)}")


print("\nNAND Gate:")
weights = [-1, -1]
bias = -1
for inputs in inputs_list:
    print(f"Inputs: {inputs}, Output: {MP_neuron(inputs, weights, bias)}")


print("\nNOR Gate:")
weights = [-1, -1]
```

```
bias = 0
for inputs in inputs_list:
    print(f"Inputs: {inputs}, Output: {MP_neuron(inputs, weights, bias)}")
```

```
AND Gate:
Inputs: [0, 0], Output: 0
Inputs: [0, 1], Output: 0
Inputs: [1, 0], Output: 0
Inputs: [1, 1], Output: 1

OR Gate:
Inputs: [0, 0], Output: 0
Inputs: [0, 1], Output: 1
Inputs: [1, 0], Output: 1
Inputs: [1, 1], Output: 1

NAND Gate:
Inputs: [0, 0], Output: 1
Inputs: [0, 1], Output: 1
Inputs: [1, 0], Output: 1
Inputs: [1, 1], Output: 0

NOR Gate:
Inputs: [0, 0], Output: 1
Inputs: [0, 1], Output: 0
Inputs: [1, 0], Output: 0
Inputs: [1, 1], Output: 0
```