Assignment-1-A

April 20, 2025

1 Assignment 1 A:

1.0.1 Neuron class:

[1]:
```python
class Neuron:
    def __init__(self, weights: list[float], bias: float):
        '''
        Initialize a neuron with given weights and bias

        Parameters:
            weights: List[float]
                The weights of the neuron
            bias: float
                The bias of the neuron
        '''

        self.weights: list[float] = weights
        self.bias: float = bias

    def activate(self, x: float) -> float:
        '''
        Activate the neuron

        Parameters:
            x: float
                The input to the neuron
        Returns:
            float
                The output of the neuron
        '''
        # using heaviside activation return 1.0 if the input is greater than or␣
↪equal to 0, otherwise return 0.0
        return 1.0 if x >= 0 else 0.0

    def feedforward(self, inputs: list[float]) -> float:
        '''
        Feedforward the neuron
```

```
        Parameters:
            inputs: list[float]
                The inputs to the neuron
        Returns:
            float
                The output of the neuron
        '''
        # check if the length of inputs is equal to the length of weights
        assert len(inputs) == len(
            self.weights), 'Length of inputs must be equal to length of weights'

        # calculate the total sum of the inputs multiplied by the weights and add the bias
        total: float = sum(
            w * i for w, i in zip(self.weights, inputs)) + self.bias

        return self.activate(total)
```

1.0.2 Test Neuron class

[2]:
```
# test the neuron
neuron = Neuron(weights=[-0.5, -0.6, 0.2, -0.3, 0.4], bias=0.0)
print(neuron.feedforward([-1.0, -2.0, -3.0, -4.0, 5.0]))
```

```
1.0
```