EXPERIMENTNO.9

Aim: Implement McCulloch Pitts model.

Theory:

The McCulloch-Pitts (MCP) neuron model, proposed by Warren McCulloch and Walter Pitts in 1943, laid the foundation for modern artificial neural networks. It is a simplified mathematical model of a biological neuron andprovides insights into how neurons might process informationinthe brain. The MCP neuron model consists of three main components: input signals, weights, and an activation function.

BasicConcepts

- *Neuron*: The fundamental building block of neural networks, a neuron receives input signals, processes them,andproduces anoutput signal. In the MCP neuron model, the neuron performs a simple computation based on the inputs and weights.

- *Input Signals*: Neurons receive input signals from other neurons or external sources. Each input signal is associated with a weight, which represents the strength of the connection between the input and the neuron.

- *Weights*: Weights determine the influence of input signals on the neuron's output. Larger weights indicate stronger connections, while smaller weights indicate weaker connections. The weights can be adjusted during the learning process to improve the performance of the neuron.

- *Activation Function*: The activation function determines whether the neuron will produce an output signal based on the weighted sum of its inputs. In the MCP neuron model, the activation function is a threshold functionthatcomparestheweightedsum toathreshold.Iftheweighted sum exceeds the threshold, the neuron fires and produces an output signal; otherwise, it remains inactive.

McCulloch-PittsNeuronModel

The McCulloch-Pitts neuron model describes how a neuron processes input signals to produce an output signal. Mathematically, the output of an MCP neuron y is computed as follows:

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i \cdot x_i \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

Where:

- y is the output of the neuron.
- x_i are the input signals.
- w_i are the weights associated with the input signals.
- threshold is a threshold value.
- n is the number of input signals.

Properties

- *Binary Output*: The output of an MCP neuron is binary, typically representedas0(inactive)or1(active).Thisbinaryoutputsimplifies computations and enables the modeling of logical functions.

- *Thresholding Operation*: The MCP neuron performs a thresholding operation, where the weighted sum of input signals is compared to a threshold.Ifthesumexceedsthethreshold,theneuronfires;otherwise, it remains inactive.

- *SimplifiedModel*:TheMCPneuronmodelisasimplifiedrepresentation of the biological neuron.It neglects details suchas temporal dynamics, synapse strengths, and non-linearities observed in real neurons.

Applications

TheMcCulloch-Pittsneuronmodelhasinfluencedvariousfields,including:

- **ArtificialNeuralNetworks**:TheMCPneuronmodelformsthebasisof artificial neural networks, which are used for pattern recognition, classification, regression, and other machine learning tasks.
- **ComputationalNeuroscience**:TheMCPneuronmodelprovidesinsights into the computational properties of biological neurons and helps researchers understand how neurons process information in the brain.
- **CognitiveScience**:TheMCPneuronmodelinformstheoriesofcognition and provides a framework for understanding how the brain performs cognitive functions such as perception, learning, and memory.

The McCulloch-Pitts neuron model is a foundational concept in neuroscience and artificial intelligence. It describes a simplified mathematical model of a biological neuronandserves asthe buildingblockfor artificialneuralnetworks. By understanding the principles of the MCP neuron model, researchers can develop more complex neural network architectures and learning algorithms for solving a wide range of tasks in machine learning and cognitive science.

ProgramCode:

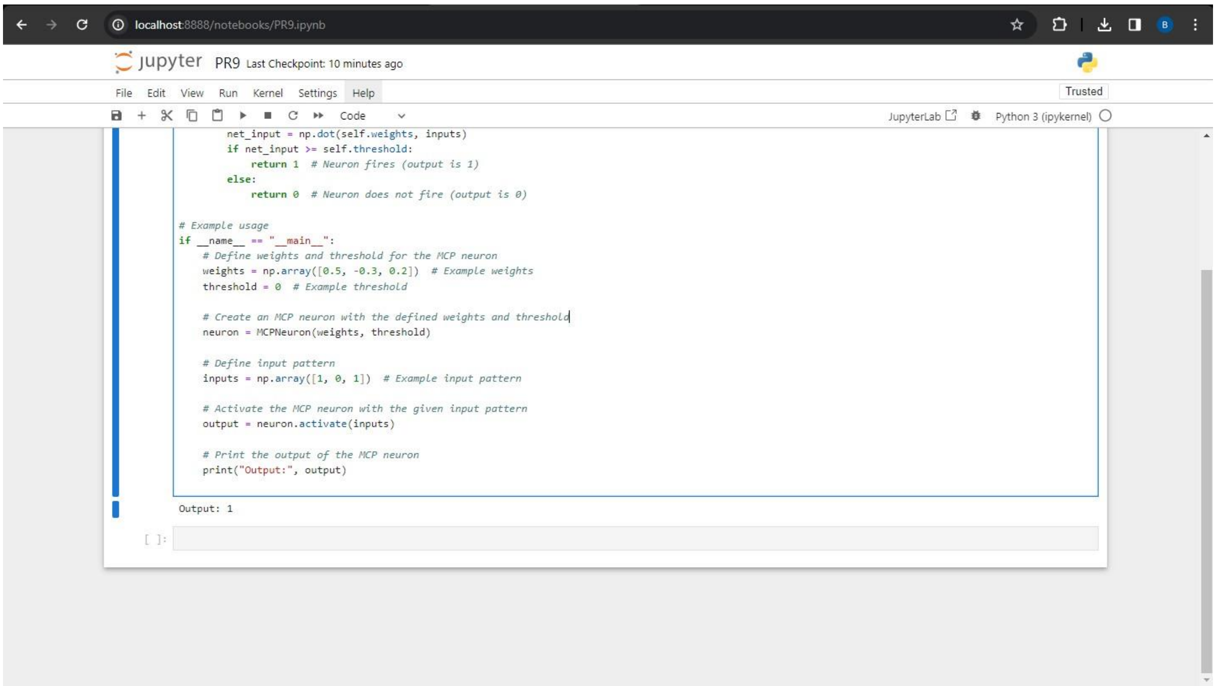
```
importnumpyasnp
```

```
classMCPNeuron:
  def __init__(self,weights,threshold):
    self.weights = weights
    self.threshold = threshold

  defactivate(self,inputs):
    #Checkifthesumofweightedinputsexceedsthethreshold net_input =
    np.dot(self.weights, inputs)
    ifnet_input>=self.threshold:
      return1#Neuronfires(outputis1) else:
      return0#Neurondoesnotfire(outputis0)

#Exampleusage
if __name___=="___main__":
  # Define weights and threshold for the MCP neuron
  weights=np.array([0.5,-0.3,0.2])#Exampleweights
  threshold = 0# Example threshold

  #CreateanMCPneuronwiththedefinedweightsandthreshold neuron =
  MCPNeuron(weights, threshold)

  #Defineinputpattern
  inputs=np.array([1,0,1])#Exampleinputpattern

  #ActivatetheMCPneuronwiththegiveninputpattern output
  = neuron.activate(inputs)
```

#PrinttheoutputoftheMCPneuron

print("Output:", output)

Output:



Conclusion:

By performing this practical we learnt to implement McCulloch Pitts model in python.