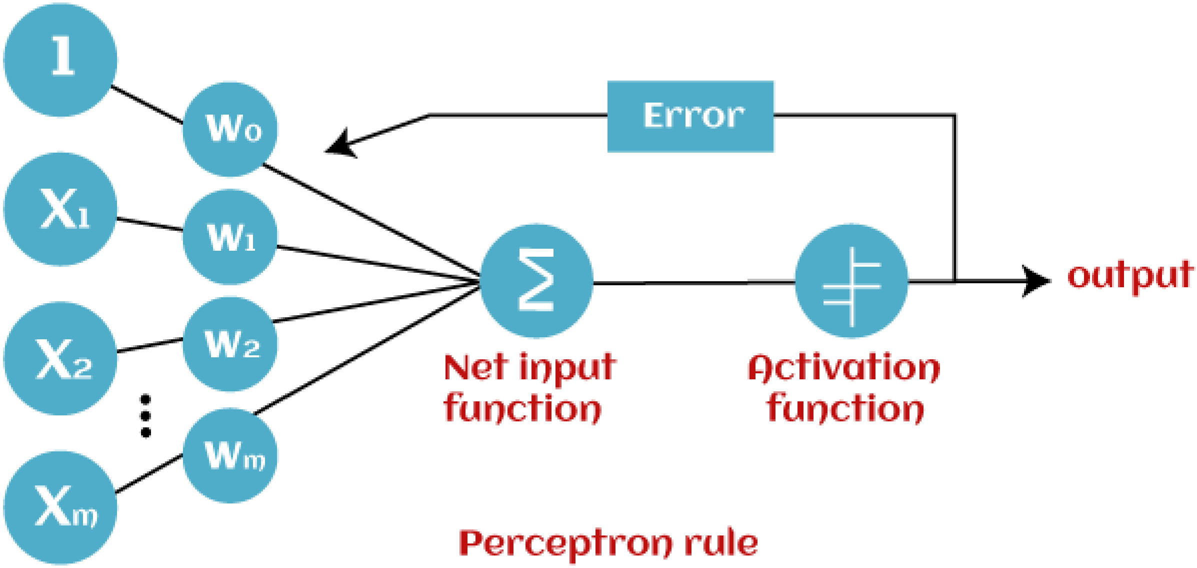
4.3. Experiment No. 3

Aim: Program using Perceptron Neural Network to recognize even and odd numbers. Given numbers are in ASCII from 0 to 9

Objective: To learn about Perceptron Neural Network and its function.

Theory:

Perceptron is a machine learning algorithm which mimics how a neuron in the brain works. It is also called a single layer neural network consisting of a single neuron. The output of this neural network is decided based on the outcome of just one activation function associated with the single neuron. In perceptron, the forward propagation of information happens. Deep neural network consists of one or more perceptrons laid out in two or more layers. Input to different perceptrons in a particular layer will be fed from the previouslayer by combining them with different weights.



Types of Perceptron models:

Single Layer Perceptron model: One of the easiest ANN(Artificial Neural Networks) types consists of a feed-forward network and includes a threshold transfer inside the model. The main objective of the single-layer perceptron model is to analyze the linearly separable objects with binary outcomes. A Single-layer perceptron can learn only linearly separable patterns.

Multi-Layered Perceptron model: It is mainly similar to a single-layer perceptron model but has more hidden layers.

Forward Stage: From the input layer in the on stage, activation functions begin and terminate on the output layer.

Backward Stage: In the backward stage, weight and bias values are modified per the model's requirement. The backstage removed the error between the actual output and demands originating backward on the output layer. A multilayer perceptron model has a greater processing power and can process linear and non-linear patterns. Further, it also implements logic gates such as AND, OR, XOR, XNOR, and NOR.

Algorithm:

Inputs:

x: input features, representing an image of a number

y: binary target class label (0 for even, 1 for odd)

w: weight vector

b: bias term

alpha: learning rate

num_iterations: number of iterations to run gradient descent

Outputs:

w: the learned weight vector

b: the learned bias term

Steps:

Initialize the weight vector w and bias term b to zero.

For each iteration of gradient descent:

For each training example (x, y):

Compute the predicted output y_hat =1 if w* x+b>0, else y_hat =0.

Update the weight vector and bias term using the following formulas:

Ww = w + alpha * (y - y_hat) * x

b =b + alpha * (y - y_hat)

Return the learned weight vector w and bias term b.

Applications:

1. A multi-layered perceptron model can be used to solve complex non-linear problems.
2. It works well with both small and large input data.
3. It helps us to obtain quick predictions after the training.
4. It helps to obtain the same accuracy ratio with large as well as small data.

Input:

6

Output:

```
even
```

Conclusion:

In this way have successfully implemented Perceptron Neural Network to recognise even and odd numbers. Given numbers are in ASCII form 0 to 9.

Outcome:

Upon completion of this experiment, students will be able to:

Experiment level outcome (ELO1): Perform Perceptron Neural Network to identify Even and Odd number for ASCII number.

Questions:

1. What is Perception Neural Network?
2. What are types of Perception Model?
3. What are the characteristics of perception?
4. What do you understand by perceptron learning algorithm?
5. What do you mean by a neural network?