

## Experiment-1.3

### Aim of the Experiment:

Implementation of Perceptron Learning Algorithm.

### Theory:

The Perceptron Learning Algorithm is a simple algorithm used for supervised learning of binary classifiers. It's designed to find the optimal separating hyperplane for linearly separable data points. The algorithm was developed by Frank Rosenblatt in the late 1950s and has since been foundational in the development of neural network models.

#### Steps of Perceptron Learning Algorithm:

- 1) Initialize the weights and threshold to small random numbers.
- 2) Multiply feature value by its corresponding weight and Sum up weighted features.
- 3) Apply an activation function (step or sign function) to the weighted sum.
- 4) Update the weights according to:

$$w_i = w_i + \alpha(y - \hat{y})x_i$$

where,

$w_i$  is the weight for feature  $x_i$

$\alpha$  is the learning rate

$y$  is the true class label.

$\hat{y}$  is the predicted class label.

$x_i$  is the feature value.

- 5) Repeat Steps 2 to 4 for a fixed number of iterations or until convergence (i.e., when no misclassified examples remain or error falls below a certain threshold).

One important thing to note is that the Perceptron Learning Algorithm only converges if the data is linearly separable. Otherwise, it may not converge, and the weights may oscillate indefinitely.

Additionally, the solution obtained by the Perceptron Learning Algorithm may not be unique if multiple separating hyperplanes exist.

An AND gate is a fundamental logic gate in digital electronics that produces an output of 1 (or TRUE) only when all of its inputs are 1 (or TRUE). Otherwise, it produces an output of 0 (or FALSE). The truth table for an AND gate is as follows:

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

We have two input features (A and B) and one bias feature (usually represented as  $x_0 = 1$ ). Weights are assigned to each input feature, and the perceptron's output is determined by whether the weighted sum of inputs exceeds a certain threshold.

### Code for Experiment :

```
% Input data and targets
x1 = [-1 -1 1 1];
x2 = [-1 1 -1 1];

% Combine them into one array
x = [x1' x2'];
t = [-1 -1 -1 1];

% Initialize values
w1 = input('Enter weight w1: ');
w2 = input('Enter weight w2: ');
w = [w1 w2];
a = input('Enter the value for the Learning Rate, Alpha: ');
theta = input('Enter the value for the Threshold, Theta: ');
b = input('Enter the value for the b: ');

fprintf('\n\n');
disp('Perceptron Algorithm for AND function: ');
fprintf('x1 = [%d, %d, %d, %d]\n', x1);
fprintf('x2 = [%d, %d, %d, %d]\n', x2);
fprintf('t = [%d, %d, %d, %d]\n', t);

% Start Training for model
fprintf('\nTraining Started ..... \n');
```

```
% Set maximum number of epochs
max_epochs = 1000;

% Number of input samples
[row_of_possible_inputs, ~] = size(x);
for epoch = 1:max_epochs
    fprintf('\nEpoch = %d\n', epoch);
    errors = 0;
    for i = 1:row_of_possible_inputs
        % Calculate s
        s = b + dot(w, x(i,:));
        if s > theta
            y = 1;
        else
            y = -1;
        end

        if y ~= t(i)
            % Update weights
            w = w + a * t(i) * x(i,:);
            % Update bias
            b = b + a * t(i);
            errors = errors + 1;
        end
        fprintf('s = %.2f \ty = %d\tb = %.2f\n', s, y, b);
    end

    % Stop if all samples are classified correctly
    if errors == 0
        fprintf('\nTraining Converged\n');
        break;
    end
end

if epoch == max_epochs
    fprintf('\nMaximum epochs reached. Training may not have converged.\n');
end
```

**Result/Output :**

```
>> SC_EXP3
Enter weight w1: 0.2
Enter weight w2: 0.3
Enter the value for the Learning Rate, Alpha: 0.1
Enter the value for the Threshold, Theta: 0
Enter the value for the b: 0

Perceptron Algorithm for AND function:
x1 = [-1, -1, 1, 1]
x2 = [-1, 1, -1, 1]
t = [-1, -1, -1, 1]

Training Started .....

Epoch = 1
s = -0.50    y = -1    b = 0.00
s = 0.10     y = 1     b = -0.10
s = 0.00     y = 1     b = -0.20
s = 0.30     y = 1     b = -0.20

Epoch = 2
s = -0.70    y = -1    b = -0.20
s = -0.10    y = -1    b = -0.20
s = -0.30    y = -1    b = -0.20
s = 0.30     y = 1     b = -0.20

Training Converged
```

**Learning outcomes (What I have learnt):**

1. Learnt about the Neurons and Neural Networks.
2. Learnt about the Perceptron Learning Algorithm.
3. Learnt about the AND function and its Implementation.
4. Learnt about the weights and Threshold value in Neural Network.
5. Learnt about how weights and bias value are updated if error exists.