SRINIVAS UNIVERSITY
INSTITUTE OF ENGINEERING & TECHNOLOGY

(SUBJECT: ARTIFICIAL NEURAL NETWORK)

(SUBJECTCODE:24SBT113)

A Individual Task on

"Error-Correction Learning Demo using Perceptron (ANN) "

Submitted in the partial fulfillment of the requirements for the fourth semester

BACHELOR OF
TECHNOLOGY IN AIML

Submitted By,

SUDESH(01SU24AI104)

UNDER THE GUIDANCE OF

Prof. Mahesh Kumar V B

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

SRINIVAS UNIVERSITY INSTITUTE OF ENGINEERING & TECHNOLOGY
MUKKA, MANGALURU-574146

2025-26

SRINIVAS UNIVERSITY

INSTITUTEOF ENGINEERING & TECHNOLOGY

Department of ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

This is to certify that **SUDESH(01SU24AI104)** has satisfactorily completed the assessment (Individual-Task – Module 1) in **"ARTIFICIAL NEURAL NETWORK "** prescribed by the Srinivas University for the 4st semester B. Tech course during the year **2025-26**.

MARKS AWARDED

<u>Staff In charge</u>

Name: Prof. Mahesh Kumar V B

Assistant Professor, Department Of AIML

TABLE OF CONTENTS:

Error-Correction Learning Demo using Perceptron (ANN)

1. Introduction:-

Artificial Neural Networks (ANN) are computational models inspired by the human brain. They consist of interconnected neurons that process information and learn patterns from data. One of the earliest and simplest neural network models is the **Perceptron**, which is used for binary classification problems.

The perceptron learns using an **error-correction learning rule**, where weights are adjusted based on the difference between the predicted output and the actual target value. In this report, we demonstrate perceptron training on **AND and OR logic gates** and analyse how the **learning rate affects convergence** and error reduction.

2. Artificial Neural Networks (ANN):-

An Artificial Neural Network is made up of three main layers:

1. Input Layer
   Receives the input features from the data-set.
2. Hidden Layer
   Processes the inputs using weights and activation functions.
3. Output Layer
   Produces the final prediction.

ANN are widely used in:

- Pattern recognition
- Image classification
- Speech recognition
- Medical diagnosis
- Forecasting systems

3. Perceptron Model:-

The perceptron is a single-layer neural network used for linear classification problems.

Components of a Perceptron

- Inputs ($x_1, x_2, \dots, x_n$)
- Weights ($w_1, w_2, \dots, w_n$)

- Bias (b)
- Weighted sum
- Activation function

Mathematical Model

Weighted sum:

$z = w_1x_1 + w_2x_2 + b$

Activation function (Step Function):

$y = 1$, if $z \geq 0$
$y = 0$, if $z < 0$

4.Learning Rate (η):-

Definition

The learning rate (η) is a hyperparameter that controls how much the weights are adjusted during each update step in the error-correction learning rule.

$$w_i(new) = w_i(old) + \eta(t - y)x_i$$

Why It Is Important

- It determines the speed of learning
- It affects stability of convergence
- It influences whether the algorithm reaches the correct solution

Effects of Different Learning Rates

Very Small Learning Rate ($\eta \rightarrow 0$)

- Very slow learning
- Many epochs required
- Stable but inefficient
- May get stuck in local regions (in complex networks)

Moderate Learning Rate

- Balanced convergence
- Faster error reduction
- Stable training

- Preferred in practical systems

Very Large Learning Rate

- Very fast updates
- May overshoot optimal solution
- Oscillations may occur
- May fail to converge

Practical Insight

In modern neural networks:

- Learning rate scheduling is used
- Adaptive optimizes (Adam, RMS Prop) automatically adjust learning rate

5. Training Dataset for AND Gate:-

Input (x_1, x_2)	**Target (t)**
(0,0)	0
(0,1)	0
(1,0)	0
(1,1)	1

6. Training Dataset for OR Gate:-

Input (x_1, x_2)	**Target (t)**
(0,0)	0
(0,1)	1
(1,0)	1
(1,1)	1

7. Algorithm for Perceptron Learning:-

Step 1: Initialize weights and bias to small values or zero
Step 2: Select learning rate η
Step 3: For each training example
Step 4: Compute weighted sum
Step 5: Apply activation function
Step 6: Calculate error
Step 7: Update weights and bias
Step 8: Repeat until error becomes zero or convergence is achieved

8. Error-Correction Learning Demonstration:-

Initial values:
$w_1 = 0$
$w_2 = 0$
$b = 0$
$\eta = 0.1$

Example training iteration for AND gate:

Input = (1,1)
Target = 1

$z = (0\times1) + (0\times1) + 0 = 0$
$y = 1$

Error = 1 − 1 = 0

No weight update needed.

9. Error Reduction During Training:-

Epoch	Error
1	3
2	2
3	1

During training, the perceptron gradually reduces classification errors by adjusting weights. As epochs increase, the error decreases until the model correctly classifies all training samples.

4	0

Example error progression:

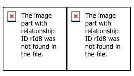
This shows the learning process improving over time.

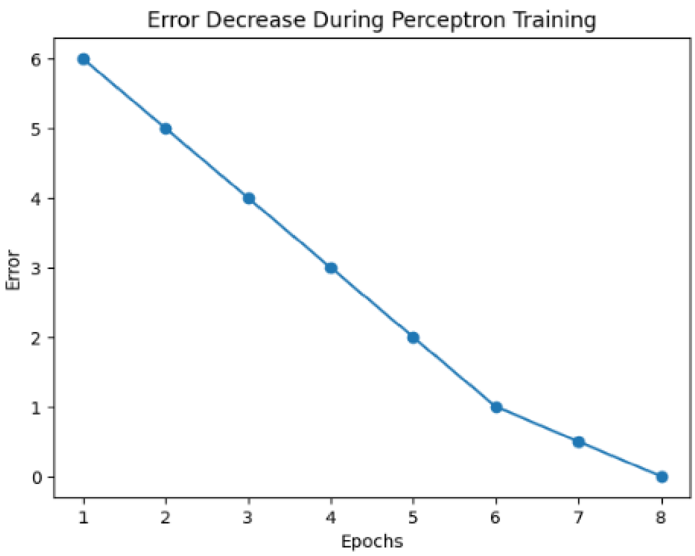
10. Graph of Error Decrease:-

The error curve typically looks like this:

High error at beginning → gradually decreases → reaches zero.

This demonstrates that the perceptron successfully learns the pattern.





11. Learning Rate and Its Effect:-

The **learning rate (η)** controls how much the weights change during training.

Small learning rate:

- Slow learning
- Stable convergence
- More training time required

Large learning rate:

- Faster learning
- May cause instability
- Can overshoot optimal solution

Moderate learning rate:

- Best performance
- Balanced convergence speed

Example:

Learning Rate	Convergence Speed
0.01	Slow
0.1	Moderate
0.5	Fast but unstable

12.Convergence in Perceptron Learning:-

What is Convergence?

Convergence means the algorithm reaches a state where:

- All training samples are classified correctly
- Weight updates stop
- Error becomes zero

Perceptron Convergence Theorem

If the data-set is linearly separable, the perceptron algorithm is guaranteed to converge in finite steps.

Examples of linearly separable problems:

- AND gate
- OR gate

Example of a non-linearly separable problem:

- XOR gate

Why Convergence Matters

- Ensures stability of the model
- Guarantees reliable predictions
- Prevents infinite training loops

13.Applications of Error-Correction Learning:-

Although simple, the principle is widely used in:

Pattern Recognition

Recognising handwritten digits or characters.

Image Classification

Detecting objects in images.

Spam Detection

Classifying emails as spam or not spam.

Medical Diagnosis

Binary classification of disease presence.

Financial Fraud Detection

Identifying fraudulent transactions.

Speech Recognition

Classifying sound signals.

14.Advantages of Error-Correction Learning:-

1. Simplicity:-

- Easy mathematical formulation
- Simple update rules

2. Computational Efficiency:-

- Requires minimal computation
- Suitable for small datasets

3. Guaranteed Convergence (for linear problems):-

- Strong theoretical foundation

4. Interpret-ability:-

- Decision boundary can be visualized as a straight line
- Easy to understand model behaviour

5. Foundation of Deep Learning:-

Modern neural networks use the same idea:

- Error calculation
- Gradient-based weight updates

15.Limitations of Perceptron Learning:-

1. Cannot Solve Non-Linear Problems:-
Fails for the XOR problem because it is not linearly separable.

2. Binary Output Only:-
Produces only 0 or 1.

3. No Hidden Layer:-
Single-layer architecture limits representation power.

4. Sensitive to Learning Rate:-
Improper selection may cause instability.

5. No Probabilistic Output:-
Unlike logistic regression, a perceptron does not give probability scores.

16. Conclusion:-

The perceptron model demonstrates the fundamental principle of learning in Artificial Neural Networks: adjusting weights based on error feedback. Through error-correction learning, the network gradually minimizes classification error and improves performance over time.

By training on AND and OR tasks, we observe:-

- Error decreases across epochs

- Convergence occurs for linearly separable problems
- Learning rate significantly affects training behaviour
- Stable learning requires careful parameter selection

Although limited in complexity, the perceptron laid the foundation for multi-layer neural networks and modern deep learning architectures. The core idea of error-driven weight updates remains central to today's artificial intelligence systems.

17. Future Scope:-

- **Implement experiment in Python**
  Automating the perceptron training process in Python would allow faster experimentation, easier weight updates, and calculation of performance metrics such as accuracy and loss.
- **Plot error vs epoch graphs**
  Visualizing how the total error changes across epochs can provide a clearer understanding of the learning behavior and convergence pattern of the model.
- **Extend to non-linear datasets**
  Applying the experiment to more complex datasets and using advanced models like multi layer perceptron can help overcome the limitations of linear separability.
- **Compare with other learning rules**
  Studying alternative learning approaches such as Hebbian learning or reinforcement learning would provide deeper insight into different training strategies and their effectiveness.