<u>INDIVIDUAL TASK-3</u>

Error-Correction Learning Demo: Train a perceptron on AND/OR tasks; plot error decreases and identify learning rate effects on convergence

1. Introduction:-

Error-correction learning is one of the foundational learning paradigms in artificial neural networks. It is based on a simple but powerful principle:

Adjust the model parameters in proportion to the error made.

The perceptron is the earliest and simplest model that uses this principle.

1.1 What is a Perceptron?

A perceptron is a single-layer binary classifier that:

- Takes weighted inputs
- Computes a weighted sum
- Applies a threshold activation
- Produces a binary output

$$z = w_1x_1 + w_2x_2 + b$$

$$\hat{y} = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Where:

- w = weight vector
- b = bias
- $\hat{y}$ = predicted output

1.2 Error-Correction Rule

$$w \leftarrow w + \eta(y - \hat{y})x$$

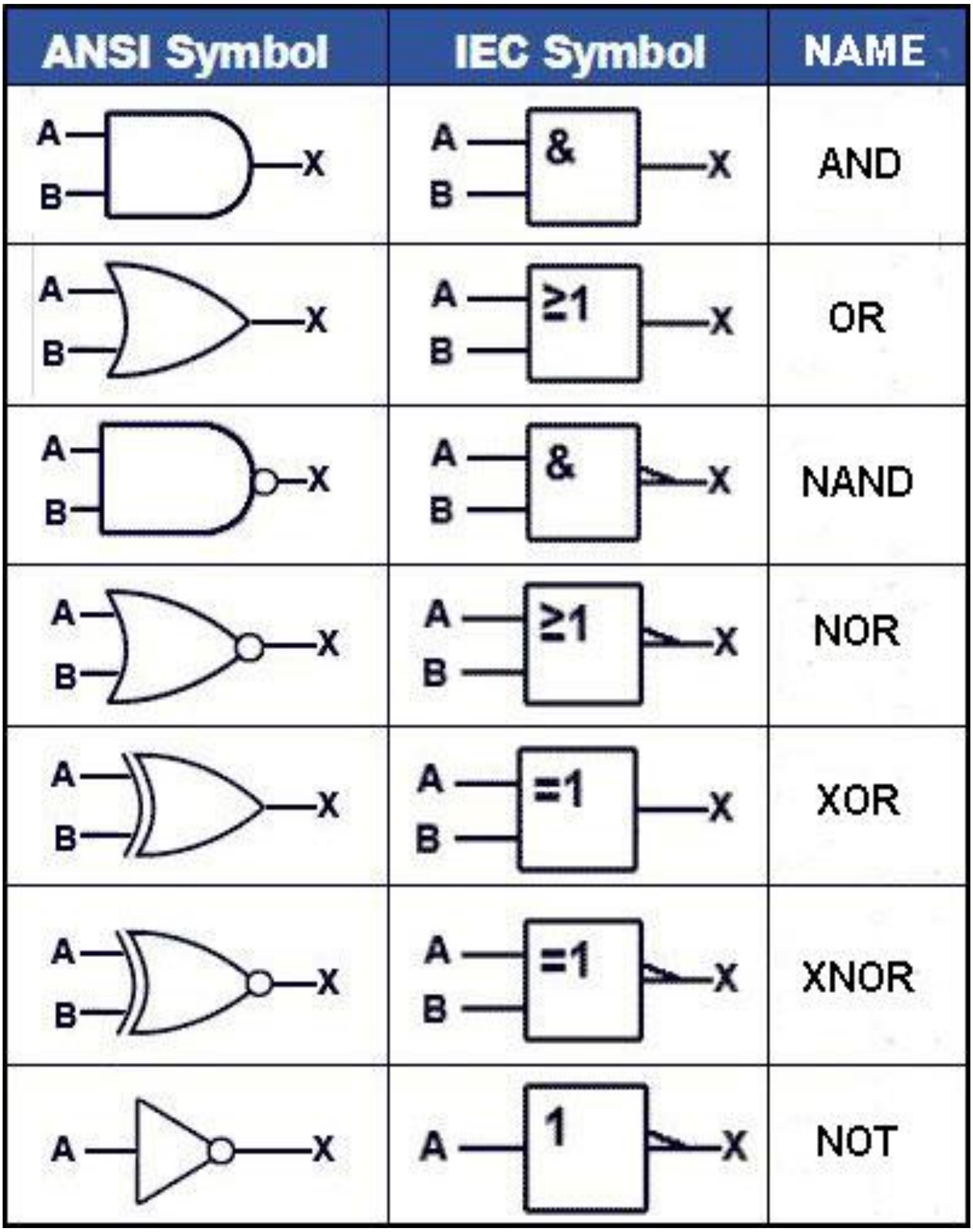
$$b \leftarrow b + \eta(y - \hat{y})$$

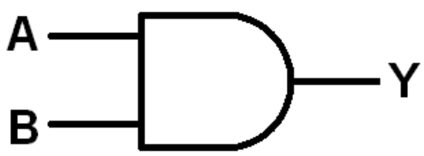
Where:

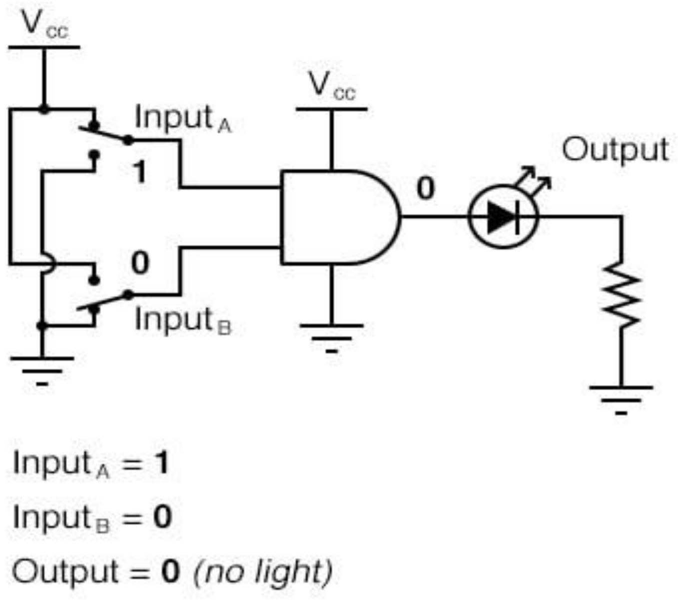
- η = learning rate
- $y - \hat{y}$ = error

2. AND Gate Learning:-

AND Truth Table







x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

2.1 Geometry of AND

- Only one positive class point (1,1)
- Decision boundary must isolate this point
- Linearly separable

Graphically:

- A straight line separates (1,1) from other three points

2.2 Training Behaviour

Initial weights are typically zeros.

During training:

- Early epochs → multiple misclassifications
- Gradually → weights adjust toward isolating (1,1)
- Eventually → zero classification error
- Typical convergence: 5–15 epochs.

2.3 Error Pattern

Error decreases monotonically for moderate learning rates.

Small learning rate:

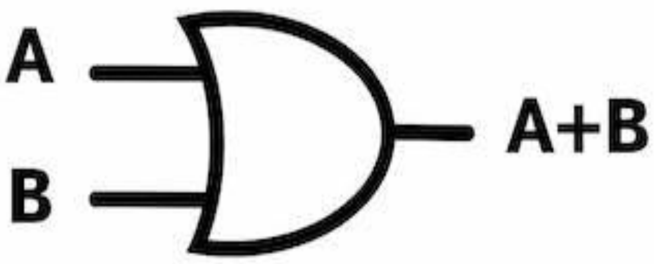
- Smooth but slow decrease

Large learning rate:

- Faster decrease
- Possible oscillations before stabilizing

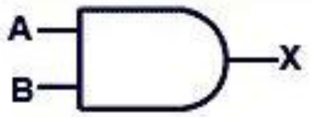
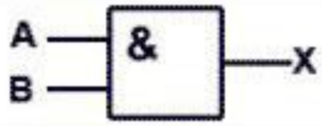
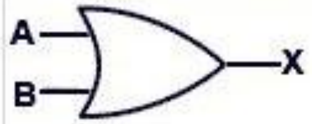
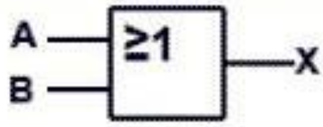
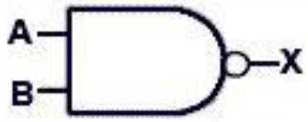
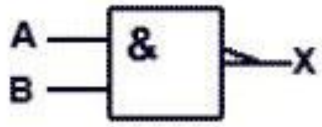
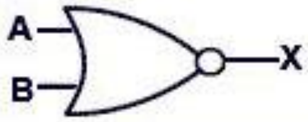
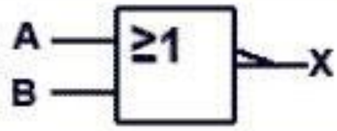
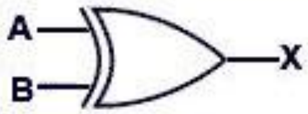
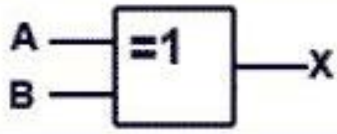
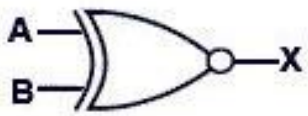
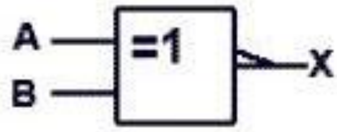
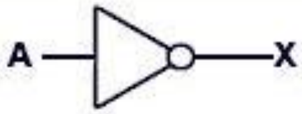
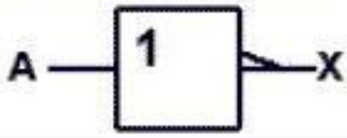
3. OR Gate Learning:-

OR Truth Table

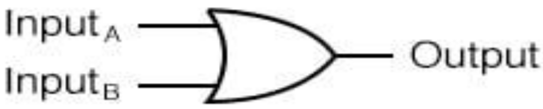


2 input OR gate

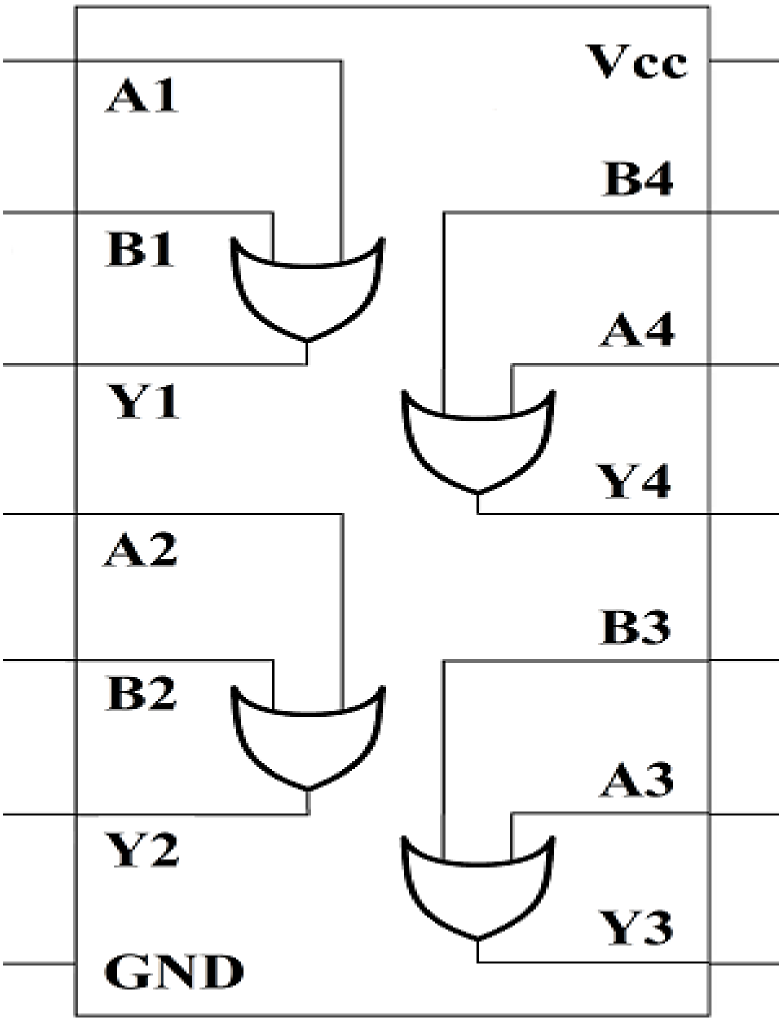
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

ANSI Symbol	IEC Symbol	NAME
		AND
		OR
		NAND
		NOR
		XOR
		XNOR
		NOT

2 - input OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



x1	**x2**	**y**
0	0	0
0	1	1
1	0	1
1	1	1

3.1 Geometry of OR

- Only one negative class point (0,0)
- Easier separation compared to AND
- Also linearly separable

3.2 Training Behaviour

Observations:

- Converges faster than AND
- Often reaches zero error in fewer epochs

Why faster?

Because:

- Only one negative point needs isolation
- Larger margin typically exists

4. Python Implementation & Error Plotting:-

4.1 Implementation

```
plt.figure(figsize=(10,5))
for lr in learning_rates:
    _, _, errors = train_perceptron(X, y_or, lr, epochs=20)
    plt.plot(errors, label=f"OR (lr={lr})")
plt.title("Perceptron Training Error (OR Task)")
plt.xlabel("Epochs")
plt.ylabel("Total Errors")
plt.legend()
plt.grid(True)
plt.show()
```

4.2 Plotting Error Decrease

```
learning_rates = [0.01, 0.1, 1.0]

# AND
plt.figure()
for lr in learning_rates:
    plt.plot(train_perceptron(X, y_and, lr)[2])
plt.title("AND")
plt.show()

# OR
plt.figure()
for lr in learning_rates:
    plt.plot(train_perceptron(X, y_or, lr)[2])
plt.title("OR")
plt.show()
```

Case 1: Small Learning Rate (0.01)

- Very gradual updates
- Smooth convergence
- Requires many epochs

Case 2: Medium Learning Rate (0.1)

- Balanced updates
- Fast and stable convergence
- Typically ideal

Case 3: Large Learning Rate (1.0)

- Large jumps in weight space
- Rapid error reduction
- Possible oscillations
- May overshoot before settling

5. Theoretical Insight:-

According to the Perceptron Convergence Theorem:

If data is linearly separable:

- The perceptron converges
- Convergence depends on:
  - Margin size
  - Learning rate
  - Initial weights

AND and OR are separable → guaranteed convergence.

5.1 Why OR Converges Faster Than AND

Because:

- OR has larger separable margin
- AND isolates only one positive point
- Geometry affects convergence speed

6. Final Discussion and Key Insights:-

6.1 Observations from the Experiment

1. Error decreases over epochs.
2. Both AND and OR converge to zero error.
3. OR typically converges faster.
4. Learning rate controls convergence speed.
5. Too large η may cause oscillation.

7. Conclusion:-

This Error-Correction Learning demonstration showed that a single-layer perceptron can successfully learn linearly separable Boolean functions such as AND and OR using the perceptron learning rule. During training, the weight updates driven by classification errors progressively reduced the total error across epochs, confirming the effectiveness of supervised error-correction learning.

For both AND and OR tasks, convergence was achieved because these problems are linearly separable. The error plots demonstrated a monotonic decrease in misclassifications until the perceptron reached a stable solution. Differences in convergence speed were strongly influenced by the learning rate:

- **Small learning rate (η)**: Produced gradual, stable convergence but required more epochs.
- **Moderate learning rate**: Achieved faster convergence with stable updates.
- **Large learning rate**: Led to oscillations around the decision boundary and, in some cases, slower overall convergence due to overshooting.

These results highlight an important trade-off: while larger learning rates accelerate weight changes, excessively large values can destabilize training. For simple linearly separable problems like AND/OR, a moderate learning rate ensures fast and stable convergence.

Overall, the experiment validates the Perceptron Convergence Theorem: for linearly separable datasets, the perceptron learning algorithm is guaranteed to converge in a finite number of updates. This demo reinforces foundational concepts in supervised learning, gradient-based optimization, and the role of hyperparameters in model performance.

8. References:-

1. Rosenblatt, Frank (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65(6), 386–408.
2. Minsky, Marvin, & Papert, Seymour (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.
3. Pattern Classification. (2001). John Wiley & Sons.
4. Neural Networks and Learning Machines. (2009). Pearson Education.
5. Deep Learning. (2016). MIT Press.
6. Machine Learning. (1997). McGraw-Hill.
7. Artificial Intelligence: A Modern Approach. (2010). Prentice Hall.
8. Neural Networks for Pattern Recognition. (1995). Oxford University Press.
9. Adaptive Filter Theory. (1985). Prentice Hall.
10. Widrow, Bernard, & Hoff, Marcian (1960). Adaptive switching circuits. IRE WESCON Convention Record.