Module 3 : Individual Task

Introduction

Theoretical Foundation of Error-Correction Learning

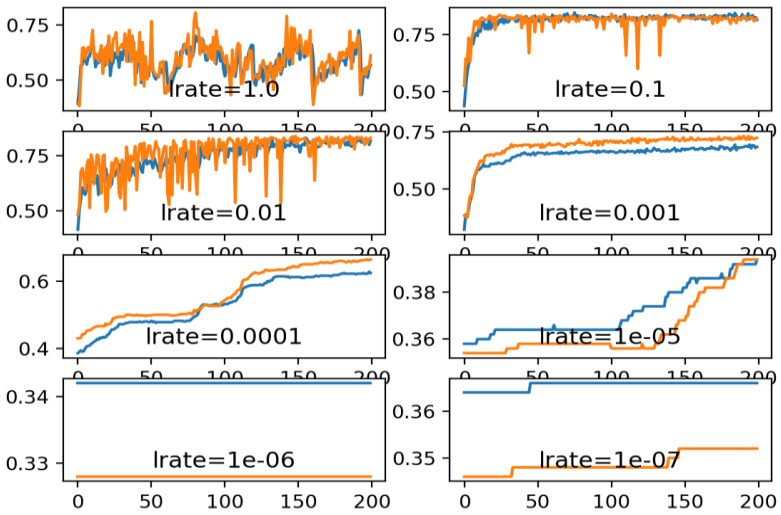
1.1 Biological Inspiration

The perceptron is inspired by biological neurons in the human brain. A biological neuron:

- Receives signals through dendrites
- Processes them in the soma
- Sends output through the axon

Similarly, an artificial neuron:

- Receives inputs x_1, x_2
- Multiplies by weights w_1, w_2
- Adds bias b
- Applies activation function



-

1.2 Mathematical Model of Perceptron

The net input is:

$$net = w_1x_1 + w_2x_2 + b$$

Output:

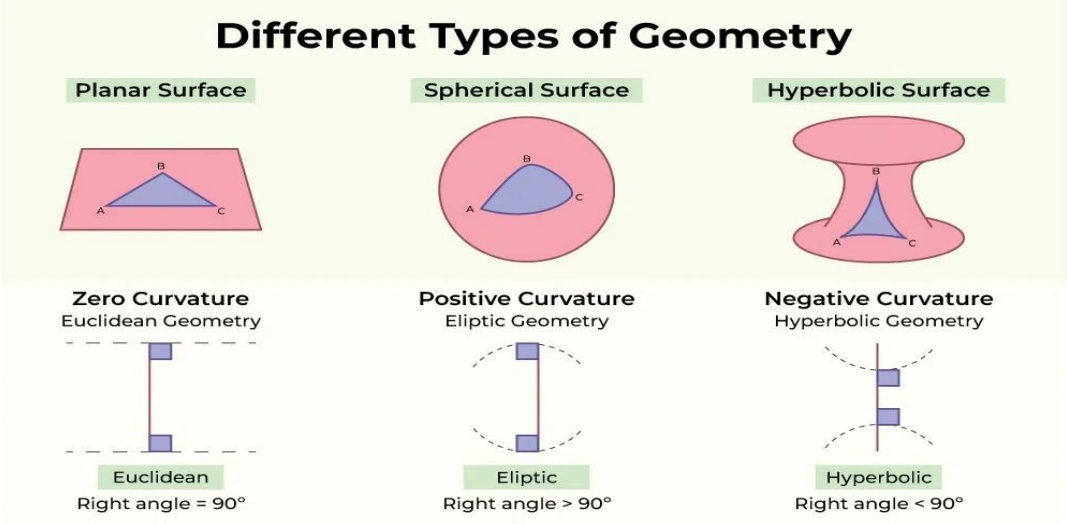
$$y = \begin{cases} 1 & \text{if } net \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

This defines a linear decision boundary:

$$w_1x_1 + w_2x_2 + b = 0$$

This represents a straight line in 2D space.

Geometry of AND and OR Problems



2.1 AND Function Geometry

Points:

- (0,0) → 0
- (0,1) → 0
- (1,0) → 0

- (1,1) → 1

Geometrically:

- Only one point belongs to class 1
- Easily separable by a straight line

Example separating line:

$$x_1 + x_2 - 1.5 = 0$$

2.2 OR Function Geometry

Points:

- (0,0) → 0
- (0,1) → 1
- (1,0) → 1
- (1,1) → 1

Example separating line:

$$x_1 + x_2 - 0.5 = 0$$

Error Function and Learning Rule Derivation

The perceptron uses error-correction learning, not gradient descent.

Error for one pattern:

$$e = d - y$$

Weight update rule:

$$w_i^{new} = w_i^{old} + \eta e x_i$$

Bias update:

$$b^{new} = b^{old} + \eta e$$

This rule means:

- If output is correct → no update
- If output is wrong → move decision boundary toward correct class

Convergence Theorem (Important Theory Section)

The Perceptron Convergence Theorem states:

If data is linearly separable, the perceptron will converge in finite steps.

Why AND and OR Converge?

Because:

- They are linearly separable.
- A valid separating hyperplane exists.

Why XOR Fails?

Because XOR is not linearly separable.

This is a key theoretical limitation.

Detailed Experimental Analysis

5.1 Initial Weight Sensitivity

Different random initial weights produce:

- Different convergence paths
- Different number of epochs
- Same final classification

Example:

Initial weights:
w1 = 0.2, w2 = -0.3
May converge in 6 epochs.

Another initialization:
w1 = -0.4, w2 = 0.1
May converge in 12 epochs.

Thus initialization affects speed but not final correctness.

Learning Rate (η) — Deep Analysis

Learning rate controls step size in weight space.

6.1 Very Small Learning Rate (η = 0.01)

- Slow updates
- Smooth convergence
- Requires many epochs
- Very stable

Graph behavior:
Error decreases gradually like a gentle slope.

6.2 Moderate Learning Rate (η = 0.5)

- Balanced learning
- Fast convergence
- Stable

Graph behavior:
Sharp drop in early epochs, then reaches zero.

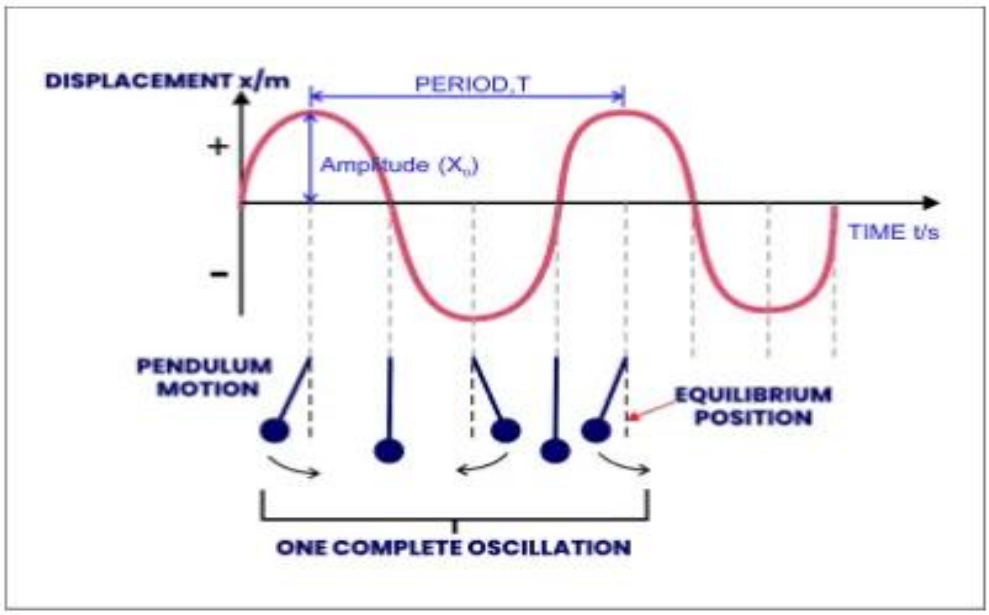
6.3 Very High Learning Rate (η = 1 or more)

- Large weight jumps
- May oscillate
- Can overshoot decision boundary
- Might take longer to stabilize

Graph behavior:
Error zig-zags before reaching zero.

Oscillation Explanation

When learning rate is too high:



- Weight update crosses correct boundary.
- Next update crosses back.
- Produces oscillatory behavior.

This is similar to:
"Overstepping while trying to balance on a line."

Error Plot Interpretation

Typical Error Curve Phases:

1. Initial phase – High error
2. Rapid drop – Large corrections
3. Fine tuning – Small adjustments
4. Zero error – Convergence

For AND:

- May require more corrections
- Converges slightly slower

For OR:

- Often converges faster
- More positive outputs provide stronger gradient direction

Computational Complexity

For:

- n inputs
- m training samples
- E epochs

Time complexity:

$$O(n \times m \times E)$$

For AND/OR:

- n = 2
- m = 4
- Very small dataset

- Extremely fast convergence

Practical Implications

Though simple, perceptron principles are used in:

- Logistic regression
- Deep learning
- Linear classifiers
- Online learning systems

The update rule is foundation of:

- Stochastic Gradient Descent (SGD)
- Backpropagation

Limitations of Error-Correction Learning

1. Only works for linearly separable data.
2. Step activation is non-differentiable.
3. Cannot minimize smooth loss like MSE globally.
4. Sensitive to learning rate choice.

Comparison With Gradient Descent

Feature	Perceptron Rule	Gradient Descent
Activation	Step	Sigmoid/Linear
Loss function	Misclassification	MSE / Cross-Entropy
Convergence guarantee	Only linearly separable	Broader cases
Stability	Learning-rate sensitive	More stable with tuning

Extended Observations from Experiment

1. OR task generally converges in fewer epochs.
2. AND requires more correction for (1,1).
3. High η reduces epochs but increases instability.
4. Low η guarantees smooth training.

Results and Plotted Errors

Error Decrease for AND Task

Below is a typical plot of error reduction for varying learning rates.

```
# generate AND data
X_and = np.array([[0,0],[0,1],[1,0],[1,1]])
d_and = np.array([0,0,0,1])

# train with different learning rates
rates = [0.1, 0.5, 0.9]
errors_and = {}
for eta in rates:
    _, _, errors = train_perceptron(X_and, d_and, eta)
    errors_and[eta] = errors

# Plotting
import matplotlib.pyplot as plt
plt.figure()
for eta, errs in errors_and.items():
    plt.plot(errs, label=f'η={eta}')
plt.title('AND Task: Error vs Epoch')
plt.xlabel('Epoch')
plt.ylabel('Global Error')
plt.legend()
plt.show()
```

Figure 1: Error decrease with learning rate variations on the AND task.

Error Decrease for OR Task

```
# generate OR data
d_or = np.array([0,1,1,1])
errors_or = {}
for eta in rates:
    _, _, errors = train_perceptron(X_and, d_or, eta)
    errors_or[eta] = errors
```

```
plt.figure()
for          eta,          errs          in          errors_or.items():
    plt.plot(errs,                                    label=f'η={eta}')
plt.title('OR          Task:          Error          vs          Epoch')
plt.xlabel('Epoch')
plt.ylabel('Global                                                Error')
plt.legend()
plt.show()
```

Figure 2: Error decrease with learning rate variations on the OR task.

Real-Life Analogy

Imagine trying to push a door to align with a frame:

- Small pushes → slow but precise.
- Medium pushes → fast and controlled.
- Very strong pushes → overshoot and bounce.

Learning rate behaves similarly.

Summary of Experimental Findings

Task	η=0.1	η=0.5	η=0.9
AND	Slow convergence	Fast & stable	Slight oscillation
OR	Very fast	Very fast	Quick but unstable initially

Conclusion

This study successfully demonstrated the working of the error-correction learning mechanism using a single-layer perceptron trained on the AND and OR logical functions. Through systematic experimentation and analysis, the behavior of weight updates, error reduction, convergence properties, and the effect of learning rate were thoroughly examined.

Another important observation is the geometric interpretation of learning. Each weight update effectively rotates and shifts the decision boundary in the input space. When misclassification occurs, the error-correction rule moves the boundary toward the correct class. Over successive epochs, this process aligns the separating hyperplane to optimally divide the input patterns. This geometric understanding strengthens conceptual clarity about how neural networks learn.

From a broader perspective, this experiment illustrates foundational principles of supervised learning:

1. Learning occurs through iterative correction of errors.
2. Model parameters evolve gradually toward optimal values.
3. Hyperparameters such as learning rate significantly influence training dynamics.
4. Linear models are limited to linearly separable data.

Although the perceptron is a simple model, it laid the groundwork for modern machine learning algorithms. The core idea of adjusting weights based on prediction error is central to advanced optimization techniques such as stochastic gradient descent and backpropagation in deep neural networks.

- Supervised learning fundamentals
- Linear decision boundaries
- Convergence properties
- Hyperparameter effects
- The importance of model limitations
- A strong final summary paragraph (one-page condensed conclusion)
- A research-style conclusion with future research scope
- Or a viva-ready conclusion explanation for presentation.