Assignment 3

Name: : Chukka Chanakya Devendra

Roll Number: : AP21110011577

Submission Date: : 09-09-2024

Exercise

1. Implement AND, and OR gate using ADALINE in python
2. Train NAND and NOR gates using ADALINE for 100 epochs in python.

In [1]:
```python
import numpy as np

class ADALINE:
    def __init__(self, lr=0.01, epochs=100):
        self.lr = lr
        self.epochs = epochs
        self.weights = None
        self.bias = None

    def activation(self, x):
        return np.where(x >= 0.0, 1, 0)

    def predict(self, X):
        linear_output = np.dot(X, self.weights) + self.bias
        return self.activation(linear_output)

    def fit(self, X, y):
        n_samples, n_features = X.shape
        self.weights = np.zeros(n_features)
        self.bias = 0

        for _ in range(self.epochs):
            for idx, x_i in enumerate(X):
                linear_output = np.dot(x_i, self.weights) + self.bias
                y_pred = self.activation(linear_output)
                error = y[idx] - y_pred
                self.weights += self.lr * error * x_i
                self.bias += self.lr * error
```

In [3]:
```python
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])

y_and = np.array([0, 0, 0, 1])

y_or = np.array([0, 1, 1, 1])

adaline_and = ADALINE(lr=0.1, epochs=10)
adaline_and.fit(X, y_and)

print("AND Gate Predictions:", adaline_and.predict(X))
```

```
adaline_or = ADALINE(lr=0.1, epochs=10)
adaline_or.fit(X, y_or)

print("OR Gate Predictions:", adaline_or.predict(X))
```

```
AND Gate Predictions: [0 0 0 1]
OR Gate Predictions: [0 1 1 1]
```

In [5]:

```
y_nand = np.array([1, 1, 1, 0])

y_nor = np.array([1, 0, 0, 0])

adaline_nand = ADALINE(lr=0.1, epochs=100)
adaline_nand.fit(X, y_nand)

print("NAND Gate Predictions:", adaline_nand.predict(X))

adaline_nor = ADALINE(lr=0.1, epochs=100)
adaline_nor.fit(X, y_nor)

print("NOR Gate Predictions:", adaline_nor.predict(X))
```

```
NAND Gate Predictions: [1 1 1 0]
NOR Gate Predictions: [1 0 0 0]
```