

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

Perceptron class

```
In [2]: class Perceptron:
    def __init__(self, input_size, lr=1, epochs=100):
        self.W = np.zeros(input_size+1)
        self.epochs = epochs
        self.lr = lr

    #Relu activation function
    def activation_fn(self, x):
        return 1 if x >= 0 else 0

    def predict(self, x):
        x = np.insert(x, 0, 1)
        z = self.W.T.dot(x)
        a = self.activation_fn(z)
        return a

    def train(self, X, labels):
        for _ in range(self.epochs):
            for i in range(len(labels)):
                x = np.insert(X[i], 0, 1)
                y_pred = self.predict(X[i])
                error = labels[i] - y_pred
                self.W = self.W + self.lr * error * x
```

Training data - ASCII representation of numbers 0 to 9

```
In [3]: X_train = [
    [0,0,1,1,1,1,1,0,0,0], # 0
    [0,1,0,0,0,0,0,1,0,0], # 1
    [1,0,0,0,0,0,0,0,1,0], # 2
    [1,0,0,0,0,0,0,0,0,1], # 3
    [1,0,0,0,0,0,0,0,1,1], # 4
    [1,0,0,0,0,0,0,1,0,0], # 5
    [1,0,0,0,0,0,1,0,0,0], # 6
    [1,0,0,0,0,1,0,0,0,0], # 7
    [1,0,0,0,1,0,0,0,0,0], # 8
    [1,1,1,1,1,1,1,1,1,1] # 9
]
```

```
In [4]: # Labels - 0 for even, 1 for odd
y_train = np.array([0, 1, 0, 1, 0, 1, 0, 1, 0, 1])
```

Creating and training the perceptron

```
In [11]: perceptron = Perceptron(input_size=10)
perceptron.train(X_train, y_train)
```

Test the trained perceptron

```
In [14]: test_numbers = [  
    [0,0,1,1,1,1,1,0,0,0], # 0  
    [0,1,0,0,0,0,0,1,0,0], # 1  
    [1,0,0,0,0,0,0,0,1,0], # 2  
    [1,0,0,0,0,0,0,0,0,1], # 3  
    [1,0,0,0,0,0,0,0,1,1], # 4  
    [1,0,0,0,0,0,0,1,0,0], # 5  
    [1,0,0,0,0,0,1,0,0,0], # 6  
    [1,0,0,0,0,1,0,0,0,0], # 7  
    [1,0,0,0,1,0,0,0,0,0], # 8  
    [1,1,1,1,1,1,1,1,1,1]  # 9  
]  
  
for i, pred in enumerate(test_numbers):  
    predictions = perceptron.predict(pred)  
    print(f"Predicted label for {i}: {'Odd' if predictions == 1 else 'Even'}")
```

```
Predicted label for 0: Even  
Predicted label for 1: Odd  
Predicted label for 2: Even  
Predicted label for 3: Odd  
Predicted label for 4: Even  
Predicted label for 5: Odd  
Predicted label for 6: Even  
Predicted label for 7: Odd  
Predicted label for 8: Even  
Predicted label for 9: Odd
```

In []: