```
import numpy as np

# Define the Perceptron class
class Perceptron:
    def __init__(self, input_size):
        self.weights = np.random.rand(input_size)
        self.bias = np.random.rand()

    def sigmoid(self, x):
        return 1 / (1 + np.exp(-x))

    def predict(self, inputs):
        summation = np.dot(inputs, self.weights) + self.bias
        return self.sigmoid(summation)

    def train(self, inputs, labels, epochs=10000, learning_rate=0.1):
        for epoch in range(epochs):
            for i in range(len(inputs)):
                prediction = self.predict(inputs[i])
                error = labels[i] - prediction
                self.weights += learning_rate * error * inputs[i]
                self.bias += learning_rate * error


# Prepare the dataset
# ASCII values for '0' to '9' are 48 to 57
ascii_numbers = list(range(48, 58))

# Represent inputs as binary (7-bit representation)
inputs = np.array([[int(bit) for bit in format(num, '07b')] for num in
ascii_numbers])

# Labels: 0 for even, 1 for odd
labels = np.array([0 if num % 2 == 0 else 1 for num in ascii_numbers])

# Create and train the Perceptron
perceptron = Perceptron(input_size=7)  # 7 bits for binary
representation
perceptron.train(inputs, labels, epochs=10000, learning_rate=0.1)

# Test the Perceptron
test_numbers = list(range(48, 58))  # Testing on 0-9
for num in test_numbers:
    binary_input = np.array([int(bit) for bit in format(num, '07b')])
    prediction = perceptron.predict(binary_input)
    print(f"ASCII {num} ('{chr(num)}') is {'odd' if prediction > 0.5
else 'even'} (prediction: {prediction:.4f})")
```

```
ASCII 48 ('0') is even (prediction: 0.0006)
ASCII 49 ('1') is odd (prediction: 0.9997)
ASCII 50 ('2') is even (prediction: 0.0005)
ASCII 51 ('3') is odd (prediction: 0.9996)
ASCII 52 ('4') is even (prediction: 0.0005)
ASCII 53 ('5') is odd (prediction: 0.9996)
ASCII 54 ('6') is even (prediction: 0.0004)
ASCII 55 ('7') is odd (prediction: 0.9995)
ASCII 56 ('8') is even (prediction: 0.0004)
ASCII 57 ('9') is odd (prediction: 0.9996)
```