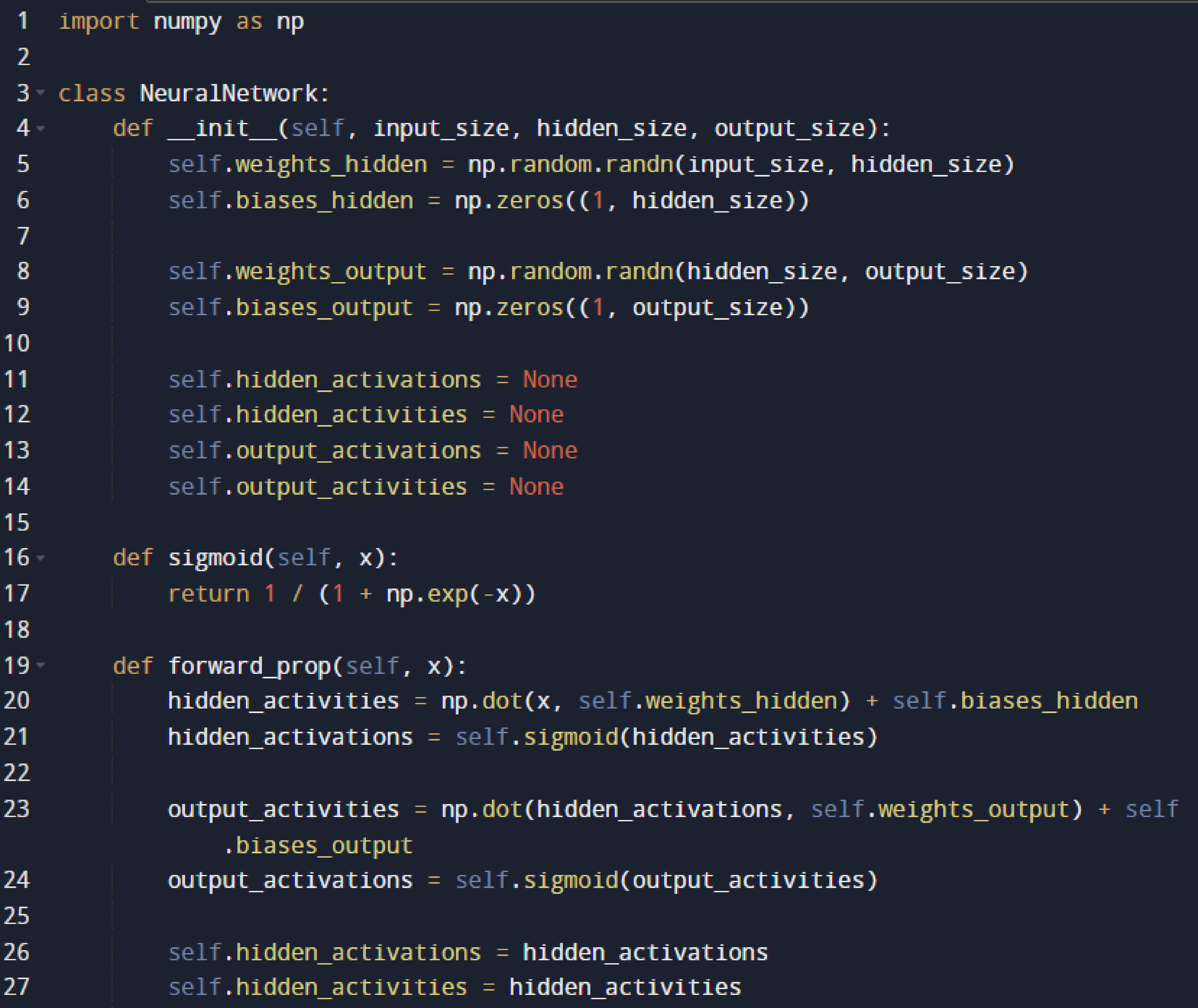
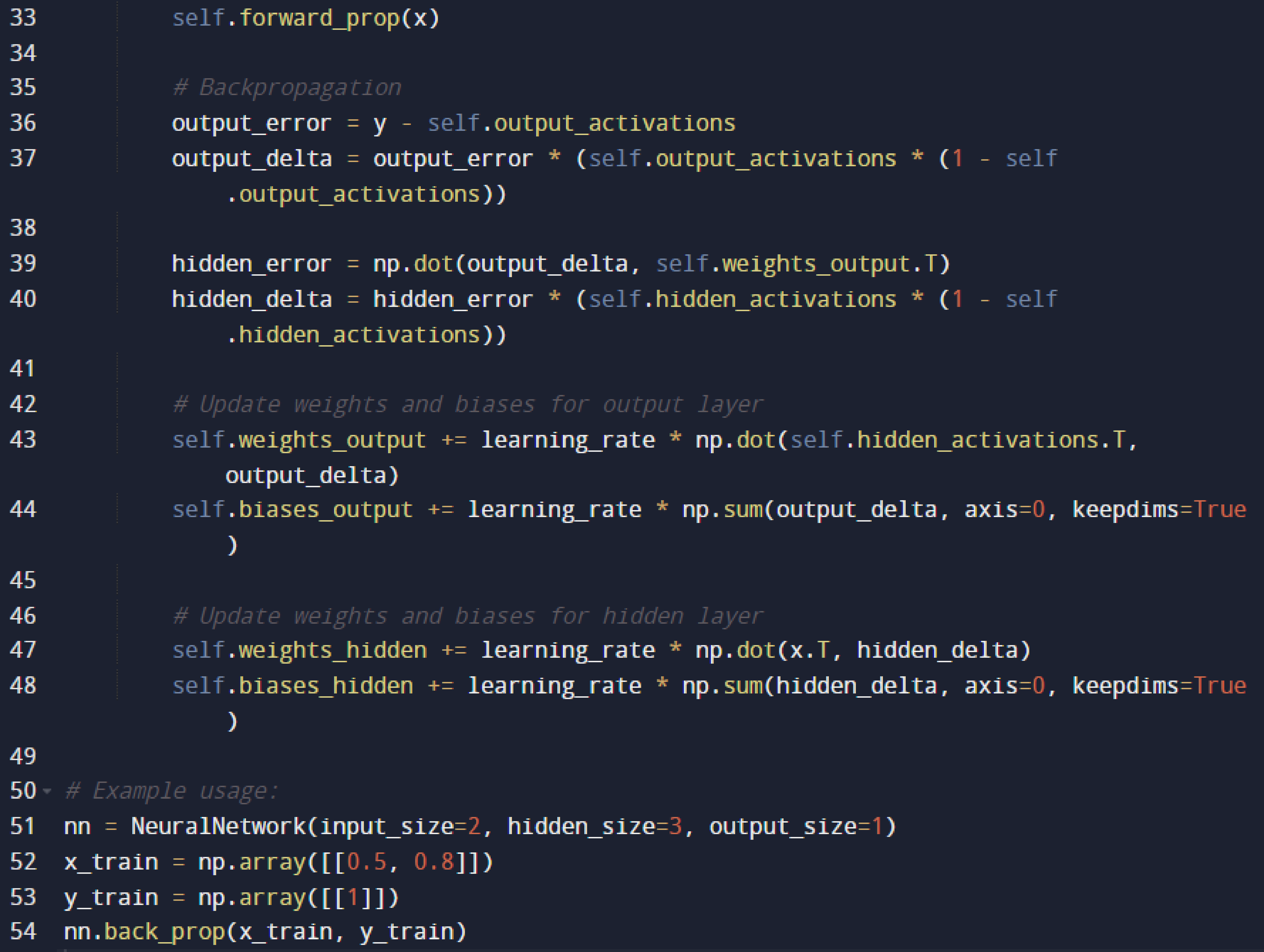


```python
import numpy as np

class NeuralNetwork:
    def __init__(self, input_size, hidden_size, output_size):
        self.weights_hidden = np.random.randn(input_size, hidden_size)
        self.biases_hidden = np.zeros((1, hidden_size))

        self.weights_output = np.random.randn(hidden_size, output_size)
        self.biases_output = np.zeros((1, output_size))

        self.hidden_activations = None
        self.hidden_activities = None
        self.output_activations = None
        self.output_activities = None

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

    def forward_prop(self, x):
        hidden_activities = np.dot(x, self.weights_hidden) + self.biases_hidden
        hidden_activations = self.sigmoid(hidden_activities)

        output_activities = np.dot(hidden_activations, self.weights_output) + self.biases_output
        output_activations = self.sigmoid(output_activities)

        self.hidden_activations = hidden_activations
        self.hidden_activities = hidden_activities
```



```python
        self.forward_prop(x)

        # Backpropagation
        output_error = y - self.output_activations
        output_delta = output_error * (self.output_activations * (1 - self.output_activations))

        hidden_error = np.dot(output_delta, self.weights_output.T)
        hidden_delta = hidden_error * (self.hidden_activations * (1 - self.hidden_activations))

        # Update weights and biases for output layer
        self.weights_output += learning_rate * np.dot(self.hidden_activations.T, output_delta)
        self.biases_output += learning_rate * np.sum(output_delta, axis=0, keepdims=True)

        # Update weights and biases for hidden layer
        self.weights_hidden += learning_rate * np.dot(x.T, hidden_delta)
        self.biases_hidden += learning_rate * np.sum(hidden_delta, axis=0, keepdims=True)

# Example usage:
nn = NeuralNetwork(input_size=2, hidden_size=3, output_size=1)
x_train = np.array([[0.5, 0.8]])
y_train = np.array([[1]])
nn.back_prop(x_train, y_train)
```