This Python code provides a simple implementation of a neural network with configurable activation functions (Sigmoid, Tanh, ReLu). It's designed to work with the Iris dataset, which consists of features related to iris flowers and their corresponding species. Here is a link to the Iris dataset that was used: https://www.kaggle.com/datasets/uciml/iris

Requirements:
- Python 3.x
- NumPy
- pandas

Installation:
1. Clone or download this repository to your local machine.
2. Make sure you have Python installed on your machine.
3. Install the required dependencies using "pip install numpy pandas"

Running the Neural Network:
1. Ensure that your Iris dataset file (Iris.csv) is located in the same directory as the NeuralNetwork class file.
2. Import the NeuralNetwork class into your session.
3. Create an instance of the NeuralNetwork class, specifying the activation function (the default is sigmoid).
4. Load and preprocess the Iris dataset using the provided methods (load_iris_data, preprocess_iris_dataset, and split_dataset).
6. Set attributes and create weights and biases using the set_attributes and create_weights_bias methods.
7. Train the neural network using the train method, specifying the activation function, learning rate, and number of epochs.
8. Test the trained model using the test method.

Adjust the activation function, learning rate, and number of epochs as necessary.

Here is an example of the code to run the neural network:

```
from NeuralNetwork import NeuralNetwork

# Create an instance of the NeuralNetwork class
nn = NeuralNetwork(activation="sigmoid")

# Load and preprocess the Iris dataset
nn.load_iris_data()
nn.preprocess_iris_dataset()
nn.split_dataset(random_state=2)
# Create the input, hidden, and output layers.
# Hidden layer has default 6 neurons
# Can change hidden_layer_size
# nn.set_attributes(hidden_layer_size=10)
nn.set_attributes()
nn.show_attributes()
print("\n")
nn.create_weights_bias()

nn.train(epochs=100, activation="tanh", learning_rate=0.1)
nn.test()
```