SimpleArtificialNeuron

This project implements a basic artificial neuron component, a fundamental building block of artificial neural networks. The component provides a simple and flexible way to model neurons with customizable weights, bias, and activation functions.

Features

- **Weighted Sum and Bias:** Calculates the weighted sum of input values and adds a bias term.
- **Activation Function:** Applies an activation function to the weighted sum to produce the neuron's output.
- **Weight Updates:** Supports updating weights based on a learning rate and error signal.

Getting Started

Using Eclipse:

1. **Download the latest release:** Go to the Releases page of the repository and download the latest ZIP file.

2. **Import the project into Eclipse:**

    - Open Eclipse and go to **File > Import...**
    - Select **General > Existing Projects into Workspace** and click **Next**.
    - Click **Browse** next to the **Select root directory** field and choose the downloaded ZIP file.
    - Make sure the project is checked and click **Finish**.

3. **Run the demo:**

    - Open the `SimpleArtificialNeuronDemo.java` file in the `src` folder.
    - Right-click anywhere in the file and select **Run As > Java Application**.

Usage

See the `SimpleArtificialNeuronDemo.java` file in the `src` folder for an example of how to use the this component.

Future Improvements

- **Multiple Activation Functions:** Currently, the component only supports the sigmoid activation function. In the future, it would be beneficial to allow users to choose from a variety of activation functions, such as ReLU, tanh, Leaky ReLU, and others. This could be achieved by:
    - Creating an `ActivationFunction` interface with a common `activate` method.
    - Implementing various activation functions as concrete classes that implement the interface.
    - Allowing the `SimpleArtificialNeuron` to accept an `ActivationFunction` object and use its `activate` method.
- **Multi-Layer Networks:** Extend the component to support the creation of multi-layer neural networks, allowing for more complex and powerful models.

- **Visualization:** Develop visualization tools to display the neuron's structure, weights, and activation values.

License