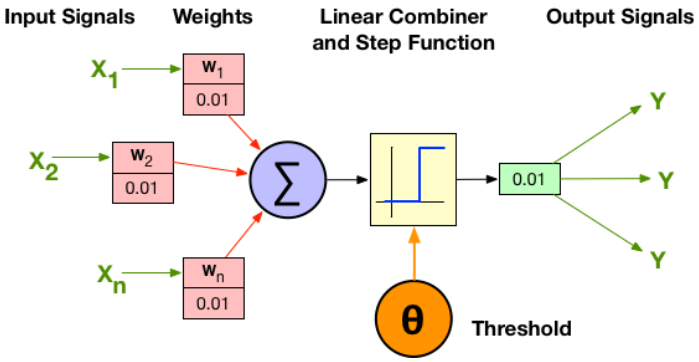
Department of Computer Science & Applied Physics

Training a Perceptron the Logical AND and OR Operations

Overview

Recall that the aim of a Rosenblatt perceptron is to classify inputs and that the model consists of a linear combiner and a hard limiter. The weighted sum of the inputs is applied to the hard limiter to compute an activation. If the activation is at or above a threshold (**θ**), the perceptron fires.



The weighted sum is computed with a simple ***for*** loop and a step activation function is then used as a hard limiter. In the context of Boolean operations, a single layer perceptron can be trained in a small number of epochs compared to other activation functions, such as the sigmoidal and hyperbolic tangent functions. However, a single-layer perceptron can only classify inputs that are linearly separable. While hidden layers can be added to create a multi-layer perceptron, that can potentially classify multi-dimensional and highly complex data, in practice a sigmoidal function is used instead of a step function to create a **multi-layer neural network**.

Exercises

- The permutations of the two operands for the logical OR and AND operators can be represented as a 2D array. Add the following declaration to the ***main()*** method:

```
float[][] data = {
  {0.00f, 0.00f},
  {1.00f, 0.00f},
  {0.00f, 1.00f},
  {1.00f, 1.00f}
};
```

- The following array of floats corresponds to the expected truth values for the **logical**

AND operations for each row of operands in the 2D array.

```
float[] expected = {0.00f, 0.00f, 0.00f, 1.00f};
```

Add this declaration to the ***main()*** method.

- Instantiate a Rosenblatt perceptron with two input variables and train the perceptron with the training data in 10000 epochs.

```
Perceptron p = new Perceptron(2);
p.train(data, expected, 10000);
```

- Test the perceptron by iterating through the values in the 2D array of training data as follows and then manually inspect the results for accuracy.

```
for (int row = 0; row < data.length; row++){
  int result = p.activate(data[row]);
  System.out.println("Result " + row + ": " + result);
}
```

- Change the expected truth values to the following array of data for the **logical OR** operation and then train and test the perceptron.

```
float[] expected = {0.00f, 1.00f, 1.00f, 1.00f};
```