Explanation of Feed-Forward Neural Networks and Back-Propagation Algorithm

Introduction

This document uses as a references the following scientific paper : A Brief Review of Feed-Forward Neural Networks.

1 Feed-Forward Neural Networks (FNNs)

1.1 Definition and Structure

Feed-Forward Neural Networks (FNNs) are a type of artificial neural network where connections between the nodes do not form cycles. The basic structure of an FNN consists of an input layer, one or more hidden layers, and an output layer. Each layer is fully connected to the next, meaning every neuron in one layer connects with a certain weight to every neuron in the subsequent layer.

1.2 Components

- **Neurons (Nodes):** The basic processing units of the network. Each neuron receives inputs, processes them using an activation function, and passes the output to the next layer.
- **Weights:** The strength of the connections between neurons. These are adjusted during the training process.
- **Activation Functions:** Functions like Sigmoid, ReLU (Rectified Linear Unit), or Tanh, which introduce non-linearity into the model, allowing it to learn complex patterns.

1.3 Types

- **Single Layer FNNs:** These networks have only one layer of weights (excluding the input layer). They are limited in their ability to model complex patterns (see Figure 1 in the document).
- **Multi-Layer FNNs:** These networks have one or more hidden layers, enabling them to model more complex relationships (see Figure 2 in the document).

2 Back-Propagation Algorithm

2.1 Definition

The Back-Propagation Algorithm is a supervised learning algorithm used to train FNNs. It works by adjusting the weights of the network to minimize the error between the predicted output and the actual output.

2.2 Steps of the Algorithm

- **Forward Pass:**
  - Inputs are passed through the network to generate an output.
  - The output is compared with the desired output to calculate the error.
- **Backward Pass:**
  - The error is propagated backward through the network.
  - The weights are adjusted to minimize the error using gradient descent.

2.3 Key Points

- The error signal for a neuron in the output layer is defined by the difference between the desired output and the actual output (see Equation (1) in the document).
- The instantaneous value of the error energy for a neuron is given by a specific formula (see Equation (2) in the document).
- The total error energy for the network is the sum of the error energies for all neurons in the output layer (see Equation (3) in the document).
- The weight updates are proportional to the partial derivative of the error with respect to the corresponding weight (see Equation (6) in the document).
- The correction applied to the weights is defined by the delta rule (see Equation (12) in the document).

3 Justification of the Choice

The Back-Propagation Algorithm is widely used for training FNNs due to several reasons:

- **Efficiency:** It is computationally efficient and can handle large datasets.
- **Flexibility:** It can be applied to various types of neural networks, not just FNNs.
- **Effectiveness:** It has been proven effective in minimizing the error and improving the performance of the network.

4 Scientific Reference

The document references the work of S. Haykin, "Neural Networks, A Comprehensive Foundation," 2nd edition, Prentice Hall, 1999. This book provides a thorough discussion of artificial neural networks and the Back-Propagation Algorithm, serving as a foundational text in the field (see Reference [1] in the document).

5 Conclusion

Feed-Forward Neural Networks are powerful tools for various applications, and the Back-Propagation Algorithm is a crucial component in training these networks. The algorithm's ability to adjust weights based on the error gradient makes it an effective and widely used method in machine learning.

By understanding the structure of FNNs and the mechanics of the Back-Propagation Algorithm, one can appreciate the complexity and potential of neural networks in solving real-world problems.