[[1]](#footnote-2)

*Abstract*—This paper reports on the recently developed pruning algorithm for the Feed Forward Multi Layer Perceptron. The importance of finding the ideal structure for an artificial neural networks lies in its efficient computations, fast convergence and its ability to generalize for yet unseen data. For that a new approach to pruning a neural network was recently developed and introduced in [1] based on a decorrelation approach between the nodes of the hidden layer. In this paper, that algorithm is applied to six classification problems and three prediction problems. The goal in both cases is to find the optimal hidden layer structure for each of these problems. Finally, we compare the results of our computed structure to the accuracies reported in some of the recent literature.

Preparation of Papers for IEEE TRANSACTIONS and JOURNALS(December2013)

Samy Shihata, Hossam Mossalam

*Index Terms*—Artificial Intelligence, Machine Learning, Neural Networks, Pruning.

# INTRODUCTION

Artificial neural networks are a paradigm of biologically inspired learning used in a wide range of machine learning applications. Ever since the conception of the Multi Layer Perceptron, artificial neural networks have been at the forefront of the research efforts in the field of learning systems and intelligent agents. In this paper, we discuss one of the main problems of using a Multi Layer Preceptron, choosing a structure. The significance of this problem is apparent from the sheer number of algorithms and heuristics that have been developed in order to address it. However despite these efforts there is yet to be a concrete solution, that can satisfactorily guarantee an optimal or a near optimal structure. Here we report on the performance of a recently developed algorithm that is based on the idea of pruning the network till it reaches its optimal structure [1].

# THE PROBLEM

A standard Multi Layer Preceptron consists of three components: input layer, output layer and a variable number of hidden layers. The number of nodes in the input and output layer is determined by the problem's domain and range respectively. Thus the only variables become the number of hidden layers and the number of nodes in each layer. It has been shown that a single hidden layer is sufficient for most standard problems, leaving only the number of hidden nodes to be determined.

Several heuristics for determining the size of the hidden layer exist based on different tradeoffs. Increasing the size of the hidden layer can lead to faster convergence, however the increase in the number of variables (degrees of freedom) risks over fitting. This can impact the networks ability to generalize for new unseen data. On the other hands reducing the number of hidden nodes can make it hard for the network to converge, increase the error value and reduce output confidence.

Ideally the structure of the MLP should be the smallest structure that can adequately generalize for new data. In practice, however, such structure is hard to determine, mainly due to the lack of semantic connection between the data and the hidden nodes. While we know that the hidden layer is responsible for mapping the input to the desired output, we have no notion of a semantic contribution of a single node to such mapping.

# The Algorithm

This algorithm belongs to a group of algorithms for optimizing artificial neural network structure called pruning algorithms. In a pruning algorithm, the network starts with a large number of hidden nodes, guaranteed to be more than required by the problem. During training, the structure of the network is gradually improved by removing nodes that are determined to be redundant. Training terminates once no more nodes can be removed [1].

## Overview

This algorithm tries to minimize the hidden layers size by reducing the variance of hidden nodes. This maximizes the correlation between the activation of each node and the net output of the network. Intuitively, it can be expected that the nodes whose activation has the least correlation to the output of the network can be removed without greatly impacting the performance of the network.

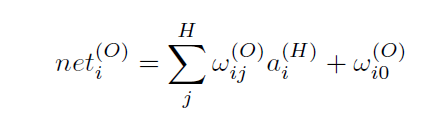
## Network Structure

In order to determine the correlation between the activation of a hidden node and the output of an output node, the variance of hidden layer activations needs to be considered. Thus this algorithm introduces a lateral connection from each hidden node to the following hidden nodes in the same layer. The lateral connections are only fed to the nodes following the origin and not the preceding nodes to keep the network strictly feed forward. 

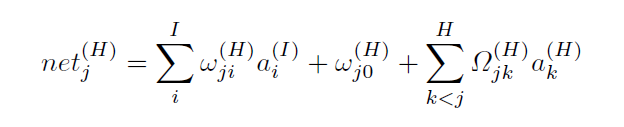
## Training

The nodes of the network are created with a sigmoid activation function for both the hidden nodes and the output nodes. The rules for the forward network propagation - taken from [1] - are as follows:

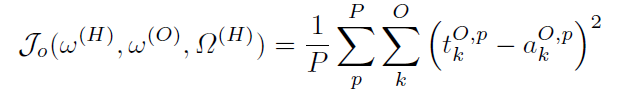
Net input for the hidden nodes

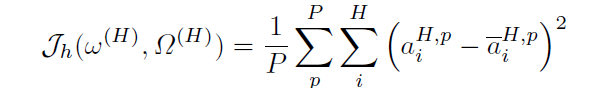


Net input for the output nodes

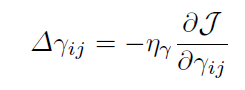


In order to train the network, we calculate an error function with

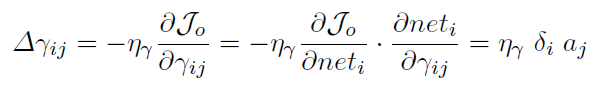




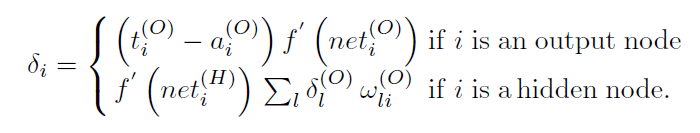
To find the minimum of the error function we approximate gradient descent using the stochastic approach. For each feature vector in the dataset we update the network weights to minimize the error function Jo using the following update rules:



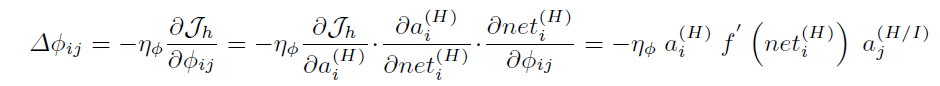
Where Δγij is calculated using the back propagation algorithm with



where η is the learning rate and δ is calculated from



For minimizing the Jh error function, we use the following update rule



# Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datasets | Results | | | |
| With Pruning | | Without Pruning | |
| Hidden layer size | Accuracy  % | Nodes = Optimum reached by algorithm | Nodes = Twice input size |
| Iris Plants | 8 | 38 |  |  |
| Wisconsin-breast-cancer | 18 | 97.22 | 91.9156 | 92.2671 |
| Hepatitis Domain | 11 | 58.75 | 58.75 | 58.75 |
| Pima Indians Diabetes | 12 | 65 | 73.0469 | 73.0469 |
| Ionosphere | 67 | 99.43 | 95.44 | 94.8718 |
| Wave form | 3 | 33.12 | 85.6400 | 91.7000 |

1. [↑](#footnote-ref-2)