ARTICLE IN PRESS

Neurocomputing xx (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Hierarchical Autoassociative Polynimial Network (HAP Net) for pattern recognition

Theus H. Aspiras[a,*], Vijayan K. Asari[a]

[a] University of Dayton, 300 College Park, Dayton, OH 45469, United States

ARTICLE INFO

Communicated by Ma Lifeng Ma

Keywords:
Neural networks
Pattern recognition
Polynomial network
MNIST
Backpropagation

ABSTRACT

Extensive research and evaluations have been conducted on neural networks to improve classification accuracy and training time. Many classical architectures of neural networks have been modified in several different ways for advancement in design. We propose a new architecture, the hierarchical autoassociative polynomial neural network (HAP Net), which is a formulation of different neural network concepts. HAP Net is a combination of polynomial networks, which provides the network with nonlinear weighting, deep belief networks, which obtains higher level abstraction of the incoming data, and convolutional neural networks, which localizes regions of neurons. By incorporating all of these concepts together along with a derivation of a standard backpropagation algorithm, we produce a strong neural network that has the strengths of each concept. Evaluations have been conducted on the MNIST Database, which is a well-known character database tested by many state of the art classification algorithms, and have found the HAP Net to have one of the lowest test error rates among many leading algorithms.

1. Introduction

Artificial neural network performance and advancement has been inspired by the human brain, which contains several neurons and synaptic junctions to retain memory. With regions of neurons grouped together, lobes are formed in the brain which govern a specific function and these lobes work together to perform complex tasks. Artificial models of different biological complex structures can be created to mimic the function of a brain system. These complex structure models include modeling the synaptic junctions created between neurons, the training of the synaptic junctions, and the overarching connectivity of the neurons.

Considering how complex these systems can be, we can take underlying conceptual ideas that formulate the biological structures of the brain. One consideration is the weighting structure between neurons, in which these weights can be modeled in a nonlinear fashion through a polynomial weight set. Another consideration is the modularity of brain structures creating lobes and region connectivity between neurons. A third consideration is the depth of the signal path for electrical pulses to radiate through the different lobes of the brain. By incorporating these three concepts together, we can provide an architecture that is inspired by the biological framework of the human brain. Thus, we propose a new architecture, the hierarchical autoassociative polynomial network (HAP net) which fuses these three concepts together to provide strong classification and regression abilities.

1.1. Objectives

The objectives of the research in this paper are:

Objective 1:

To develop a single layer polynomial network with different nodal relationships to evaluate the effectiveness of the weighting scheme.

Objective 2:

To introduce modularity into a multilayer network using polynomial weighting to create the hierarchical autoassociative polynomial network and test different configurations of the network to find the optimum structure.

Objective 3:

To develop a fully-tuned HAP net and compare results with state of the art algorithms in a well-known database.

1.2. Methods

Feed-forward neural networks are one of the most explored neural network architectures, providing a unidirectional signal flow of inputs through the weights of the network. Given that these networks can contain multiple layers of neurons and multiple neurons per layer [1], a training scheme must be developed to tune the weights of the neurons so that there will be desirable outputs based on the inputs. A backpropagation algorithm [2] can be used to iteratively train all of the weights that are initialized randomly in the architecture and reduce

* Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2016.10.002
Received 4 January 2016; Received in revised form 16 July 2016; Accepted 3 October 2016
Available online xxxx
0925-2312/

the error that is propagated throughout the system.

Weights given in a neural network system are linear with a nonlinear activation function provided after the summation of inputs, resulting in a nonlinear transformation of the inputs for a desired output. By introducing nonlinearity into the weights, we can improve the nonlinear nature of the network. Lin et al. [3] created higher-order neural networks by using combinations of the inputs together. Oh and Pedrycz [4] used a polynomial neural network, which was found to perform better than other neural network architectures. Huang et al. [5] also used polynomial neural networks to achieve 15% better detection rate than the multilayer perceptron. Yu et al. [6] used polynomial neural networks and improved the architecture by providing a rigid framework to increase efficiency and learning without unnecessarily increasing the complexity of the neural network. Oh et al. [7] looked at using the polynomial framework on fuzzy neural networks and found it every effective in regression tasks against other types of neural networks, like radial basis function neural networks.

Initial research went into small amounts of hidden layers in the network, thus providing limited transformation ability of the network, since each layer nonlinearly transforms the input for the next layer. With more amounts of hidden layers, multiple nonlinear transformations can result, thus providing even better recognition outcomes due to the higher level abstraction. Hinton et al. [8] used a deep belief network with several layers for the network and a faster training algorithm to train the system. Hinton et al. [9] then performed speech recognition and found that the capabilities of the network exceeded Gaussian Mixture Models (GMM) in accuracy. With deeper networks, modifications to the architecture can be created to improve further the recognition accuracy. Bai et al. [10] used auto-encoders and recurrent neural networks in a deep network fashion to provide high recognition results in RGB-D datasets. LeCun et al. [11] took a deep belief network and created convolutional layers, finding that it works well for different recognition tasks.

Convolutional neural networks are similar to visual fields in the brain, which create overlapping regions to formulate the network. These networks can have several layers to provide even higher levels of abstraction. Each overlapping region creates a filter, which is convolved with the corresponding region to obtain the result, which provides local information for the network. By combining several convolutional layers together, information is processed through several filters, thus resulting in the optimal transformation of the data for usage in a classification algorithm.

Ciresan et al. [12] found the convolutional neural network to have a 35% improvement over other state of the art architectures. Krizhevsky et al. [13] used the convolutional neural network on the ImageNet database, which is a very large collection of images within categories, and found that it provided the best results against various algorithms. There are many different applications for convolutional neural networks, like language processing prediction [14] and image recognition/classification tasks.

Convolutional neural networks model visual fields, which create modularity in the network. The regions created by the convolution neural network bring out the best features in each module that is optimized for recognition. By leveraging neighborhood connectivity, modularity can be created based on proximity with surrounding neurons. Happel et al. [15] found optimum configurations of a neural network by introducing modularity to the network. Modularity can be used in algorithms like modular principal component analysis [16], which create independent sub regions in imagery to perform PCA, which provides better recognition than using a whole PCA transform. Tononi et al. [17] found highly redundant connections in neural networks and proposed optimizing the configurations of the network using modularity to improve recognition and reduce computational time. Liu et al. [18] used the concepts from convolutional neural networks by utilizing receptive fields in deep belief networks, which provided high recognition accuracies across several different databases.

By incorporating these different concepts of polynomial neural networks (polynomial weighting), deep belief networks (multilayer architecture), and convolutional neural networks (modularity), we propose a new architecture, the hierarchical autoassociative polynomial network, which inherits the strengths of each concept for better recognition ability.

The distinctive contributions of this work are as follows:

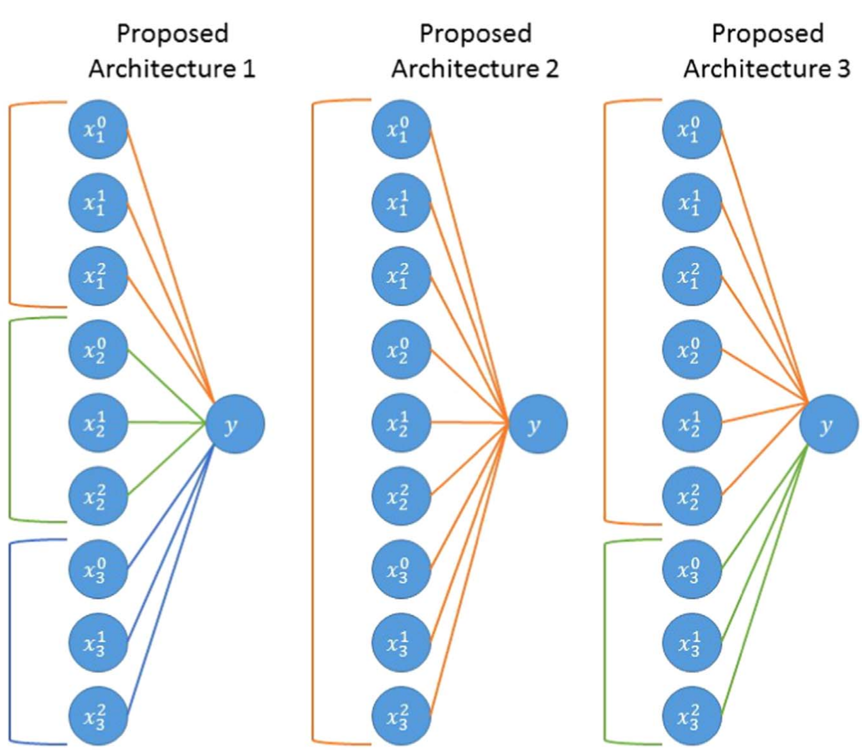

Fig. 1. Different proposed architectures. Using backpropagation can define different relationships, depending on the grouping decided in the different architectures.