Tampere University

Joni Keinänen

SMALL-WORLD ARTIFICIAL NEURAL NETWORKS FOR CLASSIFICATION

Bachelor's thesis
Faculty of Information Technology and Communication Sciences
May 2020

ABSTRACT

Joni Keinänen: Small-world artificial neural network for classification
Bachelor's thesis
Tampere University
Bachelor's Degree Programme in Electrical Engineering
May 2020

Training deep artificial neural networks requires a lot of computational power and time. This is partly because of huge number of trained parameters that exists in these networks, and some of them are well-known to be redundant after training. Traditionally artificial neural networks have very rigid layer structure which does not let information go through network very efficiently.

Many of our world's networks are classified as small-world networks, meaning that their nodes are connected to each other by tiny distances. This specific connectivity structure enables enhanced information flow though network. This study focuses on how small-world topology affects artificial neural networks performance on classification tasks. To achieve this, an artificial neural network is modeled as graph and connections within the network is rewired to create a small-world artificial neural network.

The small-world neural network is compared to regular neural network with zero rewiring. Experiments include total of six different artificial neural networks. One with dropout regularization, one with weight regularization, one without any regularization methods and their small-world counterparts. Every network has same number of neurons and connections within layers to keep results comparable. The results show that small-world networks achieved higher classification accuracy and higher convergence speed during training.

Keywords: small-world network, deep-learning, regularization

PREFACE

This thesis concludes my journey of bachelor's degree at Tampere University. I would like to thank my instructors Joni Pajarinen and Azwirman Gusrialdi for providing me an interesting topic and for all support and advice throughout the project. I would also like to acknowledge CSC – IT Center for Science, Finland, for computational resources.

Tampere 27.5.2020

Joni Keinänen

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

FFANN	feed forward artificial neural network
LSTM	long short term memory
ReLU	rectified linear unit
C	clustering coefficient
D_{global}	global efficiency
D_{local}	local efficiency
L	average path length
L1	lasso regularization
L2	ridge regression

1. INTRODUCTION

The artificial neural network is a computational tool vaguely inspired by the biological neural networks. It has been shown to be a powerful tool for various learning tasks. Deep artificial neural networks require a lot of data and computational power to train due to huge number of parameters that needs to be learned. Such networks are also prone to overfitting because of their capability to model rare dependencies in the training data.

A small-world network is a network whom nodes are connected to each other by tiny distances. Many of our worlds huge networks have small-world properties, for example the neural network of the worm Caenorhabditis [1] elegans or the social network of humans [2]. Small-world networks have been shown to be more efficient of exchanging information over the network [3]. Since the small-world networks have been shown to be prevalent in our world and efficient in solving different problem their structure could improve performance of the artificial neural networks. The goal of this research is to study how artificial neural networks with small-world topology perform in classification tasks. To achieve this total of six different artificial neural networks are designed, one with dropout regularization, one with weight regularization, one without any regularization methods and corresponding networks with small-world topology.

Chapter 2 covers theory behind the regularization methods and small-world graphs as well as related work done in area of small-world artificial neural networks. Chapter 3 introduces used methods to design a small-world artificial neural network. Chapter 4 shows experiments and results and chapter 5 contains the conclusion.

2. BACKGROUND AND RELATED WORK

This chapter contains more information about dropout, weight regularization and graph theory and related work about small-world networks and small-world artificial neural networks.

2.1 Dropout

Dropout is widely used regularization method in many state-of-the-art neural networks. It is a technique used to prevent neural networks from overfitting. Dropout simply refers to dropping out nodes in a neural network during the training phase. Dropping out means ignoring the node and all connections directly related to it as presented in figure 2.1. The nodes that are dropped out are chosen randomly with the probability *p*.

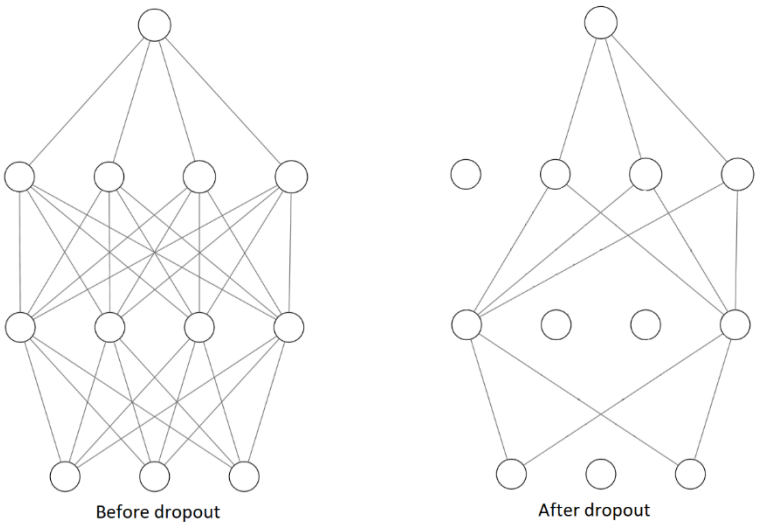

Figure 2.1. *Dropout neural network model. In left is a fully connected regular neural network. In right is the network after dropout is implemented*

Dropping out random units during training leads to more robust network and better generalization after the training [4]. Dropout only affects the training phase and does not have any effect to the networks structure otherwise.

2.2 Weight regularization

Weight regularization is another technique that is used to prevent neural networks from overfitting. It tackles the problem by keeping weights in the neural network small. Large weights in the network might lead in a situation where small changes in the input leads

to large changes in the output thus making the network unstable [5]. Weight regularization methods add a penalty term to the network's loss function.

There are two main approaches to calculate the penalty term: *L1* regularization which is also known as lasso regularization and *L2* regularization which is also known as ridge regression. The *L1* loss uses the sum of the absolute values of the network's weights and the *L2* loss uses the sum of the squared values of the network's weights. [6] The networks loss function is defined as:

$$Loss = Error(y, \hat{y}) \tag{1}$$

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} |w| \tag{2}$$

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w^2 \tag{3}$$

where λ is the regularization parameter which can be manually tuned, *w* is a weight of single connection in the network and error(y, $\hat{y}$) describes a loss between true value *y* and predicted value $\hat{y}$. Equation 1 defines a loss function with no regularization, equation 2 describes a loss function with *L1* regularization and equation 2 describes a loss function with *L2* regularization. In equations 1 and 2 the later term is penalty that is added by regularization method. The main difference between *L1* loss and *L2* loss is that *L1* loss tends to shrink the less important feature's coefficients to zero resulting to more weights with 0.0 value.

2.3 Graph theory

A graph is a mathematical structure that is used to model relations between objects. It consists a set of nodes and a set of edges. The nodes represent objects and the edges describes the relations between a pair of nodes. A directed graph is a special type of graph where every edge has an orientation. The edges in a directed graph are one-way and the direction is described with an arrow. [7] A directed graph with five nodes and six edges is presented in figure 2.2.

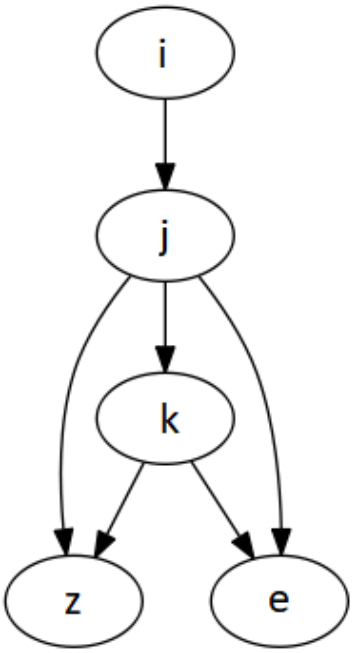

Figure 2.2. *A directed graph with five nodes and six edges*

The neighborhood of a node *j* in graph is a set of nodes that are connected to node *j* and a set of edges between those nodes. In a directed graph node have in-neighbors which include nodes that have edge to the node and out-neighbors which include nodes that have edge from the node. [7] For example, in figure 2.2 node *k* have in-neighbor *j* and out-neighbors *z* and *e*.

2.4 Small-world networks

The idea of small-world networks originated from a research paper from Watts and Strogatz [1] where they observed several complex networks such as the neural network of the worm Caenorhabditis elegans and the power grid of the western United States. Connection topology of these networks could not be classified as a completely random nor a completely regular so Watts and Strogatz introduced a new category called a small-world network. Since then small-world networks have been widely studied and it has been shown that small-world networks have enhanced signal propagation speed, synchronization, and information-flow though the network [3][2][8].

Small-world networks have two structural properties: high clustering coefficient and low average path length. Clustering coefficient measures the amount of highly connected cliques in a network and average path length measures the average path length within network's nodes. [1] In other words clustering coefficient measures local connectivity in

a network and average pathlength measures global connectivity in a network. Nodes among a small-world network are both locally and globally densely connected.

Small-world network can be generated from a regular network by following so called Watts-Strogatz model [1]. The idea of the Watts-Strogatz model is to take a regular network and add randomness to its connections. The Watts-Strogatz model is presented in figure 2.3.

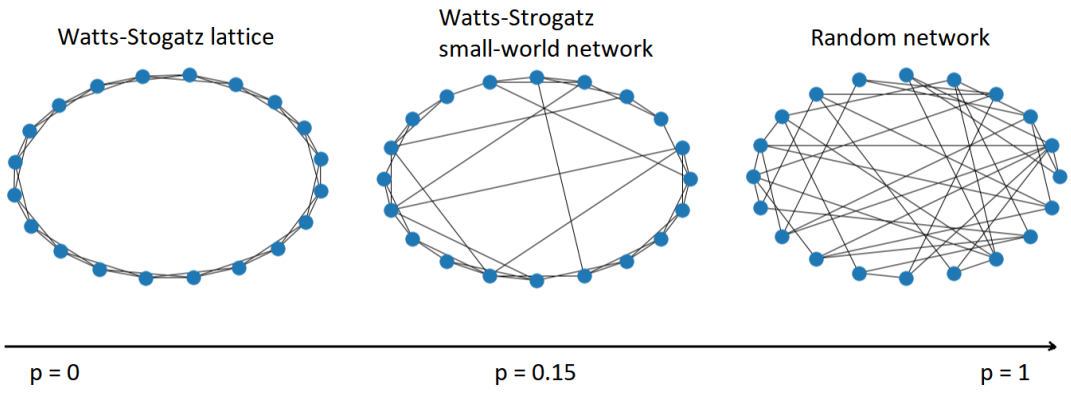

Figure 2.3. *Watts-Strogatz model*

The algorithm to produce Watts-Strogatz graph starts with regular N-dimensional ring lattice where every node is connected to its K-neighbors. Then a random subset (p%) of all connections are rewired to another random edges.

2.5 Small-world artificial neural networks

There is limited number of studies in the literature of the effects of a small-world network topology in artificial neural networks.

Simard et al. [10] compared a small-world artificial neural network to a regular feed forward artificial neural network (FFANN). The authors trained the networks with random binary input and output patterns and performed multiple experiments. They found out that in six out of seven of those experiments a small-world network topology reduced learning error and made learning faster.

Erkaymaz et al. [11] studied how a small-world network topology impacts performance of FFANN in real life problems. They implemented two experiments, estimating the thermal performances of solar air collectors, and predicting modulus of rupture values of oriented strand board. In both problems FFANN with a small-world topology was able to outperform conventional FFANN. This was followed with publications by Erkaymaz and

Mahmut [12] and by Erkaymaz et al. [13] where the authors explored the effects of small-world network topology and different methods of constructing small-world artificial neural networks in the area of diabetes diagnosis. They were able to show that a small-world FFANN reached better classification accuracy than equivalent traditional FFANN.

Javaheripi et al. [14] studied impact of small-world topology in a convolutional neural network. They trained convolutional neural networks for two different image classification tasks and compared results between small-world convolutional network, DenseNet [15] and ResNet [16]. The small-world solution achieved substantially faster convergence speed during training than two other networks.

Gray et al. [17] investigated a small-world network structure with the long short term memory (LSTM) networks. Their results show that deep small-world LSTMs are more efficient during training than fully connected dense LSTMs.

All in all, studies in literature show that small-world topology in neural networks can improve training and testing performance compared to dense networks with equal number of parameters.

3. METHODS

This chapter describes how to define small-world network and presents the algorithm used to make small-world networks.

3.1 Measurements for small-world network

In the original small-world network paper [1] Watts and Strogatz proposed that networks 'small-worldness' could be determined with two properties: clustering coefficient (C) and average path length (L). However, since FFANNs have unconnected neurons within the same layer neither clustering coefficient nor average path length can be calculated.

Authors in [3] proposed that 'small-worldness' of the network could be determined by how efficiently it transports information. They introduced two parameters: the local efficiency (D_{Local}) and the global efficiency (D_{Global}), which corresponds to 1/C and L respectively. Thus, the network exhibits a small-world property when both parameters D_{Global} and D_{Local} are small meaning that small-world networks are very efficient in global and local communication.

The global efficiency of a network is defined as:

$$D_{Global} = \frac{1}{\frac{1}{N(N-1)}\sum_{i \neq j \in N} \frac{1}{d_{ij}}} \tag{4}$$

where N is the number of nodes in the network and d_{ij} is shortest path length between two nodes i and j. The local efficiency of a network is defined as:

$$D_{local} = \frac{1}{\frac{1}{N}\sum_{x \in N} E(G_x)} \tag{5}$$

$$E(G_x) = \frac{1}{N_x(N_x-1)} \sum_{m \neq n \in N} \frac{1}{d_{mn}} \tag{6}$$

where N_x is the number of out-neighbors and in-neighbors for node x, and d_{mn} is the shortest path between nodes n and m after node x is removed from the network. [10][11]

3.2 Model

3.2.1 Rewiring algorithm

There are numerous ways to construct a small-world networks. The Watts-Strogatz rewiring algorithm presented in chapter two is used in this study. The algorithm starts with regular FFANN modeled as a graph, where each node is connected to all nodes within the next layer. The rewiring is performed by visiting each of the edges in the network once and rewiring the connection with probability *p*. Rewiring is done by removing the original connection between neurons (*i, j*) and randomly finding a new goal node (*k*). Neuron *k* is selected such as it is not an in-neighbor or out-neighbor to node *i* and there are no duplicated connections between nodes. Then connection is formed between nodes *i* and *k*. Figure 3.1 demonstrates the rewiring policy.

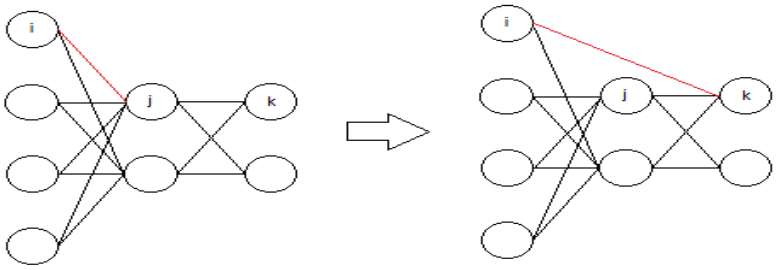

Figure 3.1. *Rewiring policy*

The rewiring process does not change the number of connections in the network thus it does not affect at the number of parameters in the FFANN.

3.2.2 Architecture for small-world neural network

In order to generate a small-world network from given artificial neural network the network is first modeled as a directed graph representation. Then the connections within graph are rewired with different probabilities $p \in [0,1]$ and efficiencies D_{Global} and D_{Local} are computed for each generated graph. Figure 3.2 represents the efficiency values versus different rewiring probability in the network that consists 6 hidden layers with 64 neurons on each hidden layer and output layer with 11 neurons.

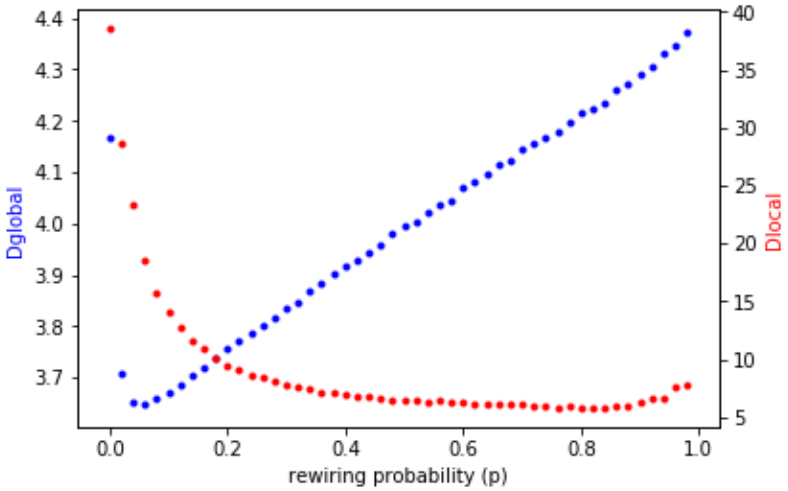

Figure 3.2. *Network efficiency values with different rewiring probabilities*

4. EXPERIMENTS

This chapter presents the datasets that are used to perform experiments and results of three classification problems.

4.1 Datasets

Experiments are conducted on three different classification tasks taken from UCI machine learning repository [18]: Dataset for Sensorless drive diagnosis, Avila dataset and letter recognition dataset. Sensorless drive diagnosis dataset consists measurements of electric current drive signals. There are 11 different classes and total of 48 different attributes are used to predict state of the motor.

Avila dataset attributes has been extracted from pictures of the Avila bible. The classification task is to associate each pattern to a copyist. Total of 10 attributes are used to predict a class and the dataset consist 12 classes.

Letter recognition dataset comprise statistical information about different letters. The objective is to identify a capital letter this information is gathered from. 16 different attributes are extracted from the letters and there are 26 different classes in the dataset.

4.2 Performance measurements

In order to compare the performance of a small-world neural network to a regular fully connected neural network and how different regularization methods affect performance total of six networks are trained: a fully connected neural network, a fully connected neural network with dropout regularization, a fully connected neural network with weight regularization, a small-world neural network, a small-world neural network with dropout regularization and a small-world neural network with weight regularization. Each of these networks have same number of layers and connections within neurons to keep results comparable. To eliminate effect of randomness, each network is trained 10 times and the plots show an estimate of the central tendency and error bands showing a confidence interval.

4.2.1 Sensorless drive diagnosis dataset

For sensorless drive diagnosis data neural networks used consists of seven layers. The layers are an input layer with 48 neurons, five hidden layers with 64 neurons each and

an output layer with 11 neurons. In the dropout networks dropout percentage is 0.3 after the input layer and 0.1 after each hidden layer. Both *L1* and *L2* regularizations are used in the weight regularization networks with values 10^{-7} and 10^{-6} respectively. ReLU activation function is used in the hidden layers and softmax in the output layer. The learning rate is set at 0.01 for the weight regularization networks, 0.001 for the dropout networks and 0.0001 for the networks without any regularization methods. All networks are trained with Adam optimizer [19] and small-world models are created by using rewiring probability of 0.7. Neural networks are trained with 1755 samples and evaluation is performed with 1000 samples.

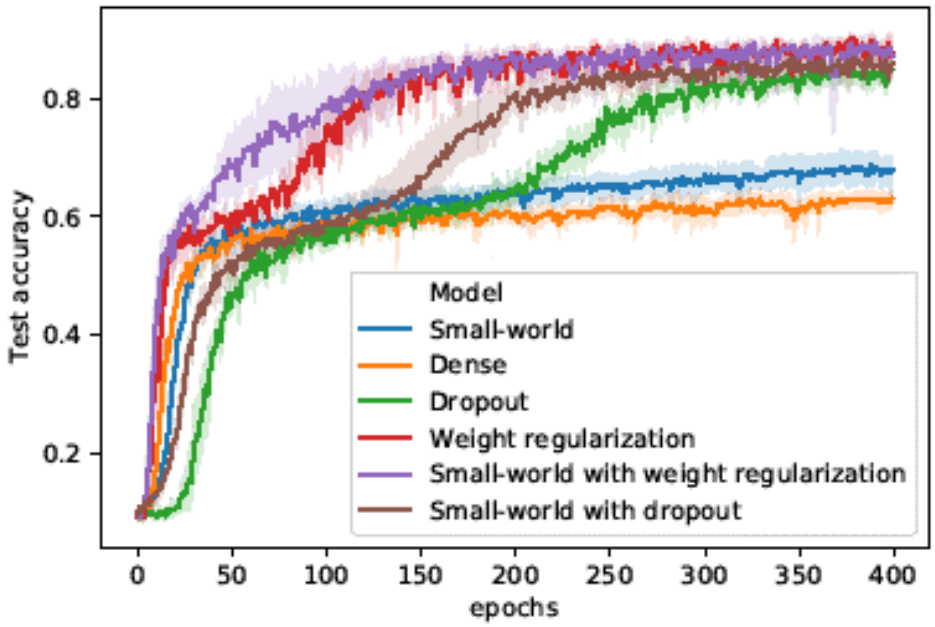

Figure 4.1. *Testing accuracy for sensorless drive diagnosis dataset*

Figure 4.1 represents the accuracy of different networks during training. Accuracy is measured from the test samples that network has not seen during training. Each network is trained with batch size of 128. As can be seen, small-world topology improves networks classification accuracy if regularization methods are not used. Small-world topology does not have big effect in accuracy when either dropout or weight regularization is used. However small-world topology seems to improve every networks convergence speed thus making networks learn faster.

4.2.2 Letter recognition dataset

For letter recognition task neural networks used consists of seven layers. An input layer with 16 neurons, 5 hidden layers with 16 neurons in each and an output layer with 26 neurons. Dropout percentage is set at 0.1 after input layer and every hidden layer. L1 and L2 regularizations are used with values 10^{-6} and 10^{-7} respectively. ReLU activation function is used for the hidden layers and softmax for the output layer. Learning rate is set at 0.01 in the weight regularization networks, 0.001 in the dropout networks and 0.0001 in the networks without any regularization methods. The Adam optimizer is used for training. The networks are trained with 2000 samples and evaluated with 1000 samples. Figure 4.2 presents training history for this dataset.

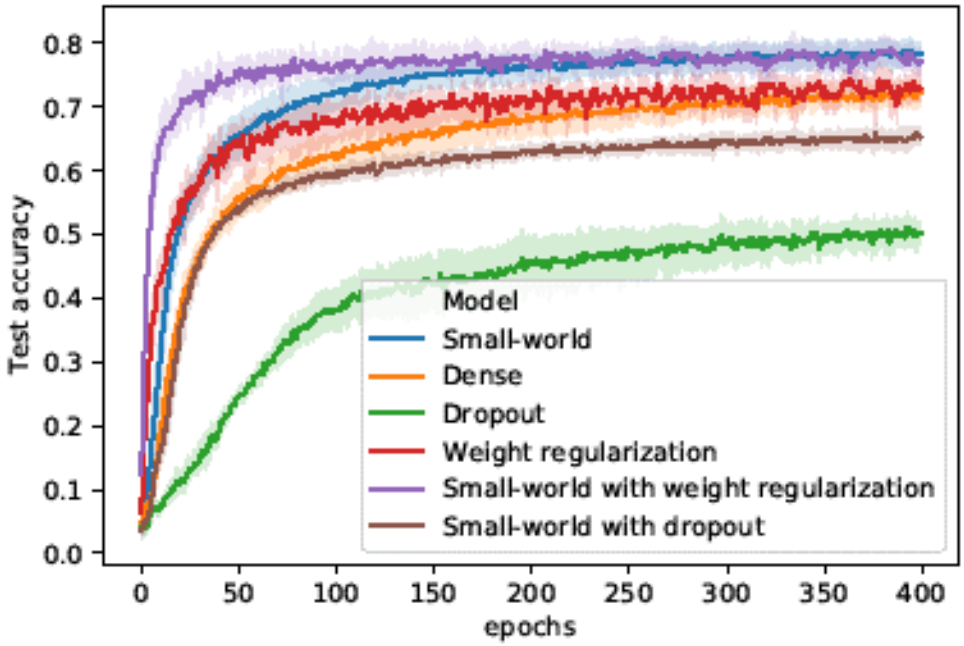

Figure 4.2. *Testing accuracy for letter recognition dataset*

In this experiment small-world topology improves final classification accuracy for all models. Although difference between models without regularization is barely noticeable. Small-world topology also improves convergence speed in the dropout and the weight regularization models. Small-world topology does not seem to have effect in convergence speed in the network with no regularization.

4.2.3 Avila dataset

For Avila dataset networks used consists of 7 layers. An input layer with 10 neurons, 5 hidden layers with 16 neurons and an output layer with 12 neurons. Dropout percentage is set on 0.1 after the input layer and all the hidden layers. L1 and L2 regularizations are used with values 10^{-6} and 10^{-7}. ReLU activation function is used for the hidden layers and softmax for the output layer. Learning rate is set at 0.01 in the weight regularization networks, 0.001 in the dropout networks and 0.0001 in the networks without any regularization methods. Adam optimizer is used for the training. The Networks are trained with 2085 samples and evaluated with 1000 samples. Figure 4.3 presents training history for Avila dataset.

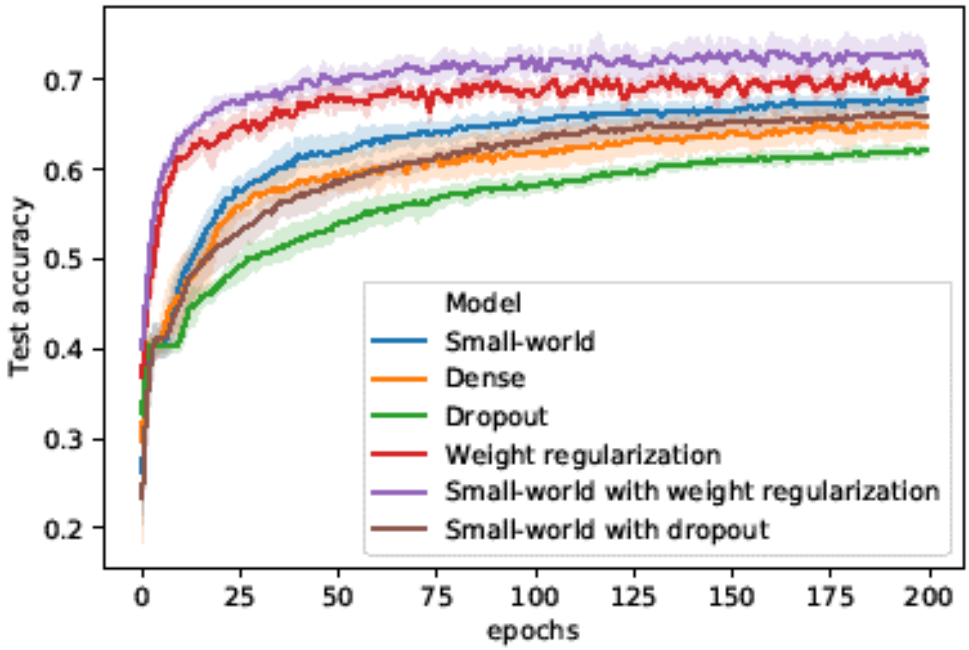

Figure 4.3. *Testing accuracy for Avila dataset*

Similarly, to letter recognition experiment a small-world topology improves classification accuracy in all models. Small-world topology also improves the convergence speed of all models in this experiment.

5. CONCLUSION

The goal of this thesis was to study how small-world topology affects performance of an artificial neural network. The proposed approach is based on previous studies within the field of small-world networks. A small-world artificial network is constructed from a regular artificial network by rewiring the connections within neurons. Rewiring is based on Watts-Strogatz model [1] and the goal is to find the optimal network structure. After rewiring the final network is both locally and globally efficient.

Networks are evaluated with three different classification tasks. The experiments show that small-world topology improves convergence speed of all networks in all three experiments. In two out of three experiments small-world topology also consistently improved networks classification accuracy. To sum it up, small-world topology consistently improved networks performance in classification tasks.

REFERENCES

[1] Duncan J. Watts, and Steven H. Strogatz. “Collective Dynamics of ‘small-World’ Networks.” Nature 393.6684 (1998): pp. 440–442.

[2] Steven H. Strogatz. “Exploring Complex Networks.” Nature 410.6825 (2001): pp. 268–276.

[3] V. Latora and M. Marchiori. “Efficient behavior of smallworld networks.” Physical review letters, 87(19):198701, 2001.

[4] Srivastava N. Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” Journal of Machine Learning Research 15 (2014): pp.1929–1958.

[5] Reed, Russell D., and Robert J. Marks. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. Cambridge, Mass: MIT Press, 1999.

[6] Ng, Andrew. “Feature Selection, L1 Vs. L2 Regularization, and Rotational Invariance.” Proceedings of the Twenty-First International Conference on Machine Learning. Vol. 69. ACM, 2004.

[7] Graph Theory. Laxmi Publications Pvt Ltd, 2018.

[8] M. Barahona and L. M. Pecora. Synchronization in smallworld systems. Physical review letters, 89(5):054101, 2002.

[9] Hong Dawei and Shushuang Man. “Signal Propagation in Small-World Biological Networks with Weak Noise.” Journal Of Theoretical Biology 262.2 (2010): pp. 370–380.

[10] Simard D, L Nadeau, and H Kröger. “Fastest Learning in Small-World Neural Networks.” Physics Letters A 336.1 (2005): 8–15.

[11] Erkaymaz, Okan & Ozer, Mahmut & Yumuşak, Nejat. (2014). Impact of Small-World topology on the performance of a feed- forward artificial neural network based on two different real-life problems. TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES. 22. 10.3906/elk-1202-89.

[12] Erkaymaz, Okan & Ozer, Mahmut. (2016). Impact of small-world network topology on the conventional artificial neural network for the diagnosis of diabetes. Chaos, Solitons & Fractals. 83. 178-185. 10.1016/j.chaos.2015.11.029.

[13] Erkaymaz, Okan, Mahmut Ozer, and Matjaž Perc. “Performance of Small-World Feedforward Neural Networks for the Diagnosis of Diabetes.” Applied Mathematics and Computation 311 (2017): pp. 22–28.

[14] Javaheripi, Mojan, and Farinaz Koushanfar. “SWNet: Small-World Neural Networks and Rapid Convergence.” arXiv.org (2019): https://arxiv.org/abs/1904.04862

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In CVPR, volume 1, page 3, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[17] S. Gray, A. Radford and P. Kingman. “GPU Kernels for Block-Sparse Weights” https://d4mucfpksywv.cloudfront.net/blocksparse/blocksparsepaper.pdf, 2016

[18] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Sciense.

[19] Kingma, Diederik, and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” arXiv.org (2017): https://arxiv.org/abs/1412.6980.