Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 193 (2021) 173–182

Procedia Computer Science

www.elsevier.com/locate/procedia

10th International Young Scientists Conference on Computational Science

Exploring Neural Network Layers for Knowledge Discovery

Sreenivas Sremath Tirumala[a]

[a]*Manukau Institute of Technology, Auckland, New Zealand*

Abstract

Research quest to explore artificial neural network weights to understand the underlying patterns has been a point of interest since later 1990s. Recent advances in artificial neural networks particularly with deep neural networks provided an opportunity to explore and expose weights in the individual layers for discovery, extraction and transfer of knowledge. Several experiments with transfer of weights from a particular layer to other layers are conducted to analyse how the classification is affected by moving weights into particular layer in a particular position. This paper, for the first time tries to investigate the importance of individual layers in a neural network through systematic experimental evaluation. Experiments are carried out using feed-forward deep neural network with different topologies. Three data sets namely MNIST, IRIS and a synthetic hierarchical data set are used for the experiments. Knowledge extraction and transfer of knowledge (through weights) experiments are performed with multiple strategies. The results indicate that the middle layer weights possess some underlying representation and has highest impact on classification accuracy. When the middle layer weights are transferred to an untrained deep neural network, irrespective of topology or type of data set (from the three data sets used), there is a significant improvement in classification accuracy.

Keywords: Knowledge Discovery;Knowledge Transfer Model;Deep Neural Networks

1. Introduction

The key components of an Artificial Neural Network or simply ANN is its weights. Considering the fact that ANN learns through optimisation of its weights, it is evident that the knowledge in an ANN resides in its weights. The knowledge of ANN is the expertise attained by ANN through solving a particular problem [1]. An individual weight of an ANN might not possess any significance by itself. In spite of being important or unimportant for impacting overall accuracy (considering dropout [2]) it cannot be concluded that an individual weight by itself consists of entire

* Sreenivas Sremath Tirumala.
* Corresponding author. Tel.: +64-9-9754604.
E-mail address: sreenivas.tirumala@manukua.ac.nz

10.1016/j.procs.2021.10.017

knowledge. Therefore, the weights in a particular combination(s) can be credited for the knowledge of ANNs. Hence, it can be concluded that the weights are responsible for the overall accuracy or in fact the functionality of ANNs. To understand the patterns in the weights, it is important to consider the impact of changes in the input features on the weights as a whole. There is a significant impact of individual attributes or combination of those on classification accuracy [3]. An investigation into the impact of changes in the input features on the underlying patterns in the ANN weights will provide an opportunity to explore the process of extracting knowledge from these weights (patterns / subsets). Also, analysing weight patterns will also provide an opportunity to understand how the weight patterns change through various layers. Further, the research into exploring the underlying patterns in the ANN weights will enable to expose hidden representation in the weights through which ANN is able to learn. To start with, an exploration of weights through visual representations might provide any existing visible pattern. However, the weights in one hidden layer consists of knowledge in the condensed format since all weights represents all the features with significant overlapping. An ANN with one hidden layer is trained on MNIST data set which attained a classification accuracy of 67.2%. Fig. 1 presents a simple visual representation of weights in an ANN with one hidden layer in the form of a 3D projection using t-Distributed Stochastic Embedding (t-DSE). From Fig. 1,it can be observed that no visible pattern of separation as representation are in condensed format. However, there is a notable separation of weight values which is represented with different colours. For an another prospective of understanding how the patters are represented in an ANN of more than one hidden layer termed as DNN , a 7-layered feed-forward DNN trained on MNIST data set is used. The DNN is fine-tuned to attain same classification accuracy (67.2%) that of the previous ANN. When the weights in the middle layer (4th layer) of DNN is project, a clear separation of cluster (weights) is observed as presented in Fig. 2. The Fig. 2 also presents a representation of features shown as A,B,C and D with a clear separation in spite of considerably less accuracy (67.2%). The features A and B are prominent and can be considered as strong features. The initial projections presented above provide clear understanding on the underlying representation in the weights. Therefore, it is feasible to study the weight patterns in the individual layers to understand the importance of a layer and its contribution towards overall accuracy of the DNN. The structure of the paper is organised as follows. Section 1 presents a background and focus of the research with a preliminary insight into the representation of features in neural network weights. This is followed by section 2 where the earlier research on knowledge discovery in ANNs is presented briefly. This is followed by a section 3 with the details of experiments and results. Section 4 presents a brief discussion on findings followed by conclusion as Section 5.

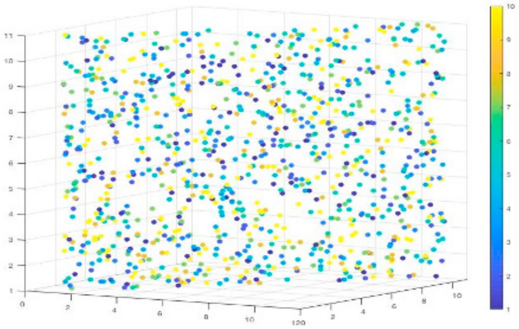

Fig. 1: Projection of weights in the hidden layer of 1-layered ANN with colours representing the attributes

2. Earlier works on Knowledge Discovery and Extraction

Knowledge discovery, extraction and transfer of knowledge has been one of the prominent topic of interest in neural network research. With recent advances in DNN, the knowledge extraction has been directly linked to the discovery of underlying representation inside the neural network weights. There are earlier ANN researchers working on knowledge extraction as late as 1990s [4, 5]. Initial works on knowledge extraction models are based on changing, often removing attributes from the input and collecting the ANN weights in the hidden layer which are attributed the better

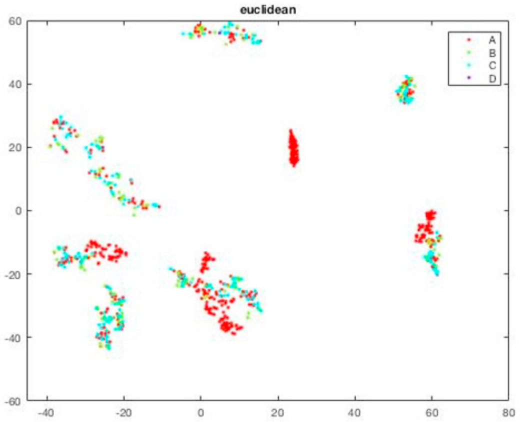

Fig. 2: Projection of weights of 4th (middle) layer of a DNN using Euclidean values. The figure also presents 4 features A, B,C and D with A and B being prominent features

classification accuracy since the weights consists are trained with only significant class based attributes in through supervised learning. A similar approach is successfully applied in 2006 with DNNs which, to some extent, successful for linearly separable data [6]. However, in the case of data with considerable overlapping of features, a ANN with only one layer fails to provide a clear pattern (a similar approach presented in section I) particular with multi-class data [7].The knowledge extraction approaches in ANNs are confined to specific problems and attained limited success in reducing training time. The recent experimental evaluation of knowledge discovery in DNNs presents no holistic view of weights in a particular layer or layers in a particular position. The task of knowledge discovery has been diluted by the importance of transfer of knowledge or transfer learning without understanding what has been transferred internally. The concept of transfer learning in DNNs termed as Deep Transfer Learning (DTL) was proposed in 2013 which concentrates on identifying transferable feature as knowledge [8]. A typical knowledge transfer involve moving weights, most of the time the entire layer or a set of layers from one DNN to another similar to a simple copying . The investigations on building a theoretical concept of understanding transferable features as transfer of knowledge is limited to understanding how and what features are transferable [9]. An early works on mere copying weights from one DNN to another was proposed in 2008 for deep convolutional networks [10] followed by a similar approach for ANNs for character recognition of latin characters using trained weights on Chinese characters [11]. There are several other works similar to the aspects of copying weights which are implemented [6, 12, 13, 14, 15, 16].

2.1. Evaluation of layers

This section presents earlier works on evaluation of neural network layers predominantly through transfer of knowledge experiments. The first attempt to understand the importance of a layer is done by Yosinski through transferring weights from a trained DNN to an untrained DNN [13]. To start with, a couple of Constitutional Neural Network $CoNN_1$ and $CoNN_2$ are trained using ImageNet . Each of CoNN is trained with equal parts of data from ImageNet. After training for considerable amount of time, the weights from $CoNN_1$ and $CoNN_2$ are transferred to new and untrained $CoNN_3$ and $CoNN_4$. The experiments from [13] concludes with a narrative that the generalisation is occurring in the first 3 layers of the CoNN. In other words, the weights in the first two layers consists of transferable features. The conclusions presented in [13], are contradicted by [15] using Deep Adaptation Networks in prestigious ICML conference in 2015. The results from [15] suggests that the generalisation tends to occur at the last few layers i.e., towards domain adaptation of specific features approaching the classifier layer. [6] concludes that the transfer of weights is possible with the trained layers and cannot be confined to a fixed layer())s. The work proposed in [6] introduced a new blocks of weights between the layers for transformation and those weights are considered as transferable weights and can be moved to new untrained networks. Transductive approach utilises weights of labelled training instances to train untrained networks to reduce training time [12]. However, the approach presented in [12] is not clear on mention what exactly is being transferred , is it a set of weights or the whole layer itself. Further, it is not clear on how it is different to using the same network as the [12] uses same data set. Apart from this there are several other deep learning based approaches where a set of layers are transferred either the initial set of layers near the input or layers near the classifier or sometime a set of weights collected in between the layers [17, 18, 19, 20]. The validation of conclusions drawn from the literature (above mentioned) is not within the scope of this research. However, it is important to highlight that the approaches presented in the literature for transfer of knowledge are based on arbitrary selection of layers or identifying weights by introducing new methods. There is no indication on what is responsible for weights values (being just numeric values)to act differently in each layer. Further, it is important to investigate how the underlying representation (or at this stage patterns) in the weights are changed over each layer after training. Further, the impact of removing a particular layer is not been considered which, to some extent, this paper will be exploring through systematic experiments.

3. Experiment Setup and Results

The task of identifying which layer has highest or least impact on the overall classification accuracy involves set of experiments with diversified strategies. For the experimental evaluation in this paper, three different data sets are used namely MNIST, IRIS and synthetic data set created with feature hierarchies used in [21] which consists of 102 samples. MNIST is a well known data set of handwritten characters (10 digits) with a training and testing samples of 60000 and 10000 respectively whereas IRIS is a benchmark data set with 150 samples and a maximum achievable classification accuracy of 99%. The experiments are carried out using Weka 3.5, Matlab 2018 and Microsoft NN framework to eliminate any software bias for randomisation of initial weight values. The hardware used for the experiments includes 4core GPU with 16 core desktop on windows 10 operating system with 512GB SSD and 16GB RAM. Since the experiments are not aimed at optimisation, open source code is used for all the experiments. A 7-layered feedforward DNN is used for all the experiments trained using Stochastic Gradient Descent (SGD) algorithm. The experiment is carried out for 500 epochs with learning rate and momentum as 0.31 and 0.38 respectively. The activation function used is Sigmoid with Softmax layers using Back Propogation (BP) algorithm. The number of hidden nodes in each layer are in the sequence of 785-1024-2048-2048-2024-1024-785 respectively. The experiment setup used by Hinton with deep belief networks, other benchmark results with DNNs and deep autoencoders inspired to choose 7-layers [1].The DNN is trained several times with different set of weights and two types of DNNs layers are locked highest accuracy as DNN$_h$, least accuracy as DNN$_l$. A colour coding is followed to represent with greenDNN$_h$ presented in Fig. 3 and DNN$_l$ with red.

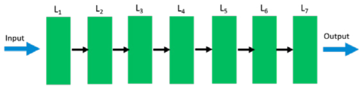

Fig. 3: Initial architecture of a seven layer DNN. The green colour represents that the weights in the layer are of DNN$_h$ (highest accuracy)

3.1. Locking and Extracting trained weights

Initial experiments are carried out to understand the impact of locking weights of one layer of trained DNN$_h$ with other weights initiated with random values. The experiments are carried out for classification without retraining the DNN. The representation of this strategy is presented in Fig. 4. A layer weights are adopted from a trained network and frozen for changes to see the impact of classification accuracy and execution time without providing any new training i.e., with random weights in the all other layers except the frozen layer. To start with the weights of the first layer of DNN$_h$ are frozen and the weights for all other layers are assigned randomly. Classification experiments are performed on this modified DNN with synthetic data set followed by freezing the weights of the second layer and assigning random weights to the other layers. This process is followed for all the experiments and the results are tabulated as Table 1.The results from the experiment are plotted in the Fig. 5 and it can be observed that the accuracy is highest when the weights of the middle layer are frozen i.e., when the middle layer weights consists of the weights of the DNN$_h$ which attained highest accuracy. To ascertain the results further, a second strategy is adopted where one layer

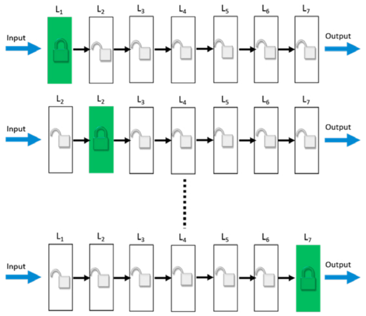

Fig. 4: The representation of experiment setup for freezing weights to identify the impact on classification accuracy.

Table 1: A comparison of original classification accuracies obtained on synthetic data set with the accuracies when a layer weights are frozen

Layer No	**Layer-wise Accuracy %**	**T-Test Results**	**Difference with Original Accuracy % (97.6)**
1	92.1	0.011	5.5
2	93.9	0.031	3.7
3	94.2	0.054	3.4
4	**96.2**	**0.021**	**1.4**
5	93.1	0.05	4.5
6	92.38	0.012	5.22
7	93.9	0.091	3.7

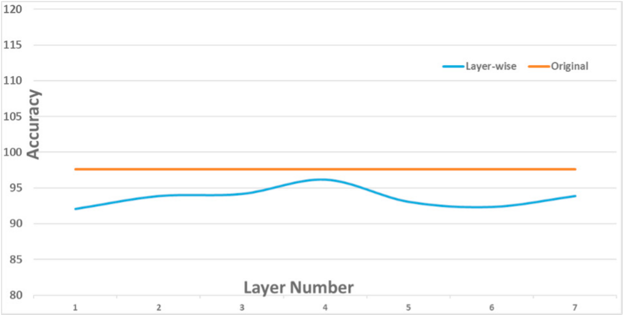

Fig. 5: A comparison between original accuracy achieved and accuracy when layer(s) is frozen (one layer at a time) for synthetic data set

from DNN$_h$ is extracted to create an ANN with 1 hidden layer and the classification experiments are conducted using this ANN using the same synthetic data set. Fig. 6 presents the results of classification accuracies and a comparison with the highest (blue line), least(black line) and set of random weights (green) and with the layer wise transfer of weights from the DNN$_h$. It is noteworthy to observe that when the middle layer weights are used in ANN, the ANN was able to achieve near best accuracy i.e., the accuracy achieved with DNN$_h$.

3.2. *Transfer of weights from Trained DNN*

The transfer of weights (referred as transfer of knowledge and sometimes transfer learning) is the process of moving weights from a trained DNN to an another DNN. However, this research is aimed at understand the importance and contribution of a layer, the experiment is designed based on transferring weights of a one layer from the trained DNN in this case DNN$_h$ (highest accuracy) to another DNN with least accuracy i.e., DNN$_h$, one layer at a time. The

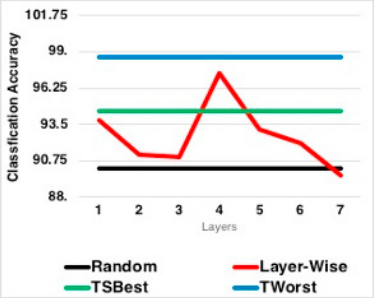

Fig. 6: Experiment Results: Comparison of classification accuracies between random weights DNN, DNNs with highest (DNN$_h$) / least accuracies (DNN$_l$) and one layer ANN formed by extracting a layer from DNN

proposed strategy is presented in Fig. 7. The experiment results when the middle layer weights are replaced is shown in Table 2 and a graph depicting the results is presented in Fig. 8. From the results it can be clearly noted that when the middle layer weights are replaced the classification accuracy is improved for all three data sets. The experiments further prove that when the middle layer is replaced, the execution time for testing is also reduced since the weights are already trained. A comparison of execution times with and without replacing the middle layer, for instance MNIST representing the original weights and MNIST(M) when the middle layer weights are replaced is presented in Fig. 9. The execution time for MNIST is 76 minutes whereas for MNIST(M) it is 41 minutes with a reduction of 35 minutes. For IRIS and SYN datasets also the impact can be clearly observed with a difference of 37 minutes for Synthetic dataset (SYN is124 and SYN(M) is 87 minutes) and 20 minutes for IRIS (IRIS 49 and IRIS(M) 29 minutes).

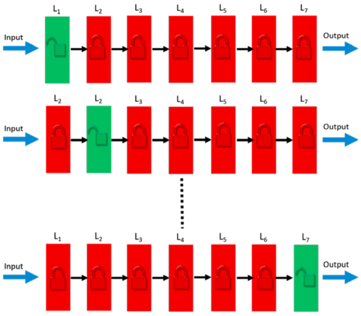

Fig. 7: Measuring classification accuracies while transferring weight values from DNN with highest (DNN$_h$) to least (DNN$_l$) classification accuracies one layer at a time

Table 2: Classification Results

data set	Original	Replacing Middle layer	T-Test
MNIST	90.2	93.8	0.002
SYN	91.4	96.5	0.0012
IRIS	89.2	98.6	0.012

3.3. Transferring weights to DNNs with different topology

In the previous sections, the transfer of weights experiments are carried out by moving weights from trained DNN to untrained DNN with the same topology. The third strategy presented in this section is to understand the impact of

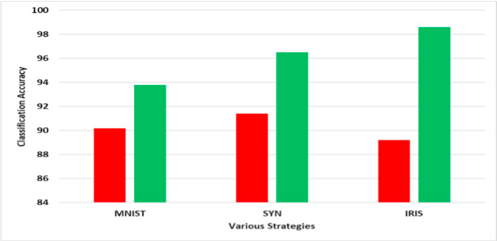

Fig. 8: Comparison of classification accuracy when middle layer weights are transferred from a trained to untrained DNN. Original accuracy is indicated in Red and Green indicates when the middle layer is replaced with trained weights.

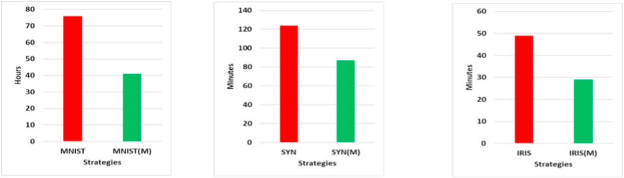

Fig. 9: Comparison of execution time when middle layer weights are transferred from a trained DNN to untrained DNN. Original value is indicated in Red and Green indicates when the middle layer is replaced with trained weights.

transferring weights from a middle layer of the 7-layered DNN weights are transferred to a 5-layer and 9-layer DNNs as presented in Fig. 10. The experiments are carried out with all three data sets i.e., IRIS, MNIST and synthetic data set. The classification results for IRIS, MNIST and synthetic data set are presented in Table 3. The experiment results indicate that the classification accuracy is considerably increased when the middle layer weights are transferred into

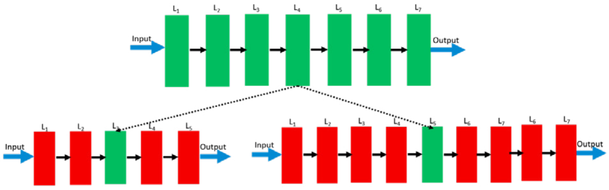

Fig. 10: Transfer of knowledge (weights) from a seven layer DNN to a five layer and a nine layer DNN

Table 3: Classification Accuracies IRIS

Dataset	DNN	Classification Accuracies			
	Topology	*Before Transfer*		*After Transfer*	
		%	T-Test	%	T -Test
IRIS	5-layered	51.2	0.037	76.5	0.002
	9-layered	43.5	0.046	71.2	0.032
MNIST	5-layered	33.5	0.022	80.2	0.006
	9-layered	45.3	0.081	76.8	0.0058
Synthetic	5-layered	44.6	0.004	85.2	0.003
	9-layered	57.4	0.006	79.7	0.008

DNNs for every topology used in the experiments. Further, the results also provide a clear indication on the significance of middle layer and the improvement of classification accuracy when the weights of middle layer (trained DNN) are transferred to untrained DNN. The improvement is irrespective of training mechanism, data set and topology. The next section presents a visualisation of weights to provide an insight on this peculiar phenomenon.

4. Discussion

The investigation for understanding the influence of a DNN layer provides an experimental evidence that the middle layer of a trained DNN possess a set of weighs that impact the classification accuracy and in turn the functionality of the DNN. Keep this in view a conjecture can be proposed that the middle layer possess important weights which now can be called as knowledge that is attained through training. At this stage though the results may not be generalised for all the data sets, it is a significant step since the experiment are performed using a variety of strategies and data sets. Fig. 11 presents a scenario of transfer of knowledge that is important and has the highest contribution for the functionality of DNN. Further, it can be noted that the knowledge is in the form of some deep representations that are hidden in the deeper layers. Majority of the knowledge transfer works particularly when weights are copied to another DNN, the knowledge that exist is getting copied unknowingly. The exploration of weights in the layers through a 3 dimensional projection is of a good help at this initial stages of the research. The projection of weight values (using t-DSE) for layer one(1), four(4) which is the middle layer and seven(7) are presented in the Fig. 12, Fig. 13 and Fig. 14 respectively. The weights are projected to observe the relative distance between the weights which indicates statistical significance (if any). The weight projects for layer 1 and layer 7 clearly provide a visual observation that the relative distance between the weights is more and the weights are a bit scattered. For the middle layer, the relative distance seems small and the weights are somewhat condensed. At this stage of the research it is a conjecture that the middle layer consists of representation (features) that are folded in the form of some 'knowledge components' which consists of the majority of important factors that guide the classier. Further, a graph representing the weight

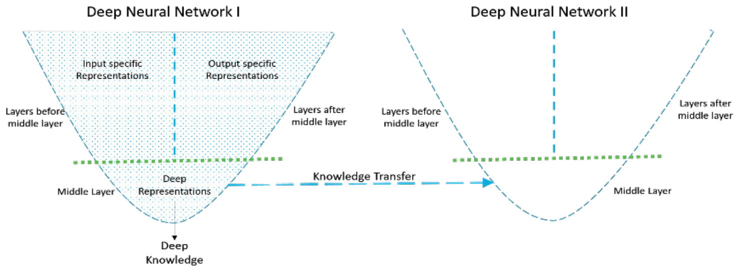
Deep Neural Network I
Deep Neural Network II
Input specific Representations
Output specific Representations
Layers before middle layer
Layers after middle layer
Layers before middle layer
Layers after middle layer
Middle Layer
Deep Representations
Knowledge Transfer
Middle Layer
Deep Knowledge

Fig. 11: The Deep Representations and Knowledge Transfer scenario: The middle layer holds the knowledge as deep representations and as such yields the highest accuracy when transferred into another DNN

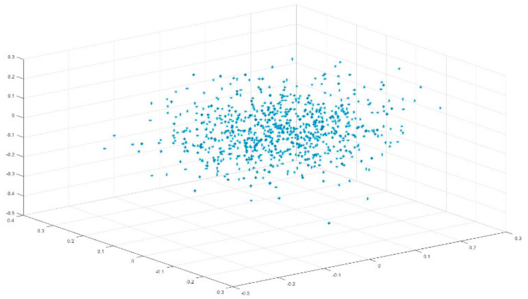

Fig. 12: Projection of weights for layer 1

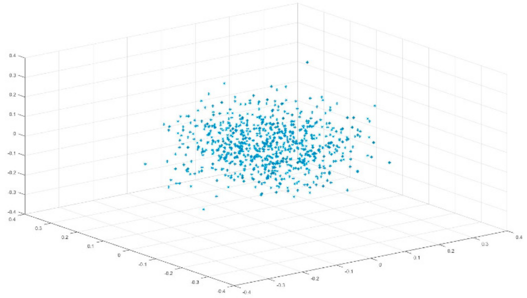

Fig. 13: Projection of weights for layer 4

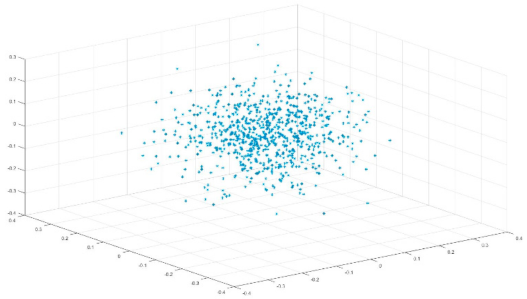

Fig. 14: Projection of weights for layer 7

variance (average variance of the weights) in each layer is presented in Fig. 15 which indicates that the middle layer has minimum variance compared to other layers. The weight variance in the layer 7 indicates that the weights at this layer have become label specific hence the value is raised up. As mentioned earlier, though conjecture at this stage is based on the experiment results, it certainly ascertains the fact that there are some hidden representation in the middle layer which influence the classification results the most. It is require to further explore the phenomenon to provide a theoretical foundation which might intern help to propose a new knowledge extraction and knowledge transfer method.

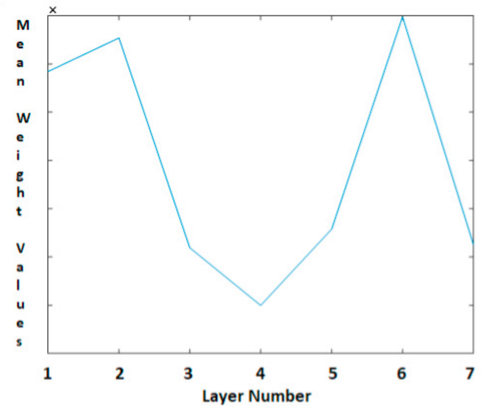

Fig. 15: The graph representing the weight variance (average variance of the weights) in each layer

5. Conclusion

This paper tries to explore the whether a particular layer has a set of weights that has more influence over the other layers in aiding the deep neural network in classification accuracy and training. Several experiments are performed using different topologies and three prominent data sets namely MNIST, IRIS and a hierarchical synthetic data set. The results indicate that the middle layer weights possess some underlying representation and has highest impact on classification accuracy. When the middle layer weights are transferred to an untrained deep neural network, irrespective of topology or data set (of three data sets used), the classification accuracy is improved. Further research with knowledge transfer (mathematical) model based on the concepts provided in this paper has been conducted which could not be included in this publication due to the limitation of conference publications, time and size.

References

[1] S. Sremath Tirumala, A component based knowledge transfer model for deep neural networks, Ph.D. thesis, Auckland University of Technology (2020).

[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[3] S. S. Tirumala, A. Narayanan, Attribute selection and classification of prostate cancer gene expression data using artificial neural networks, in: Pacific-Asia conference on knowledge discovery and data mining, Springer, 2016, pp. 26–34.

[4] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, AI magazine 17 (3) (1996) 37–37.

[5] L. Fu, Knowledge discovery based on neural networks, Communications of the ACM 42 (11) (1999) 47–50.

[6] A. V. Terekhov, G. Montone, J. K. O'Regan, Knowledge transfer in deep block-modular neural networks, in: Conference on Biomimetic and Biohybrid Systems, Springer, 2015, pp. 268–279.

[7] Z. Waszczyszyn, et al., Neural networks in the analysis and design of structures, Springer, 1999.

[8] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (8) (2013) 1798–1828.

[9] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, arXiv preprint arXiv:1411.1792 (2014).

[10] S. Gutstein, O. Fuentes, E. Freudenthal, Knowledge transfer in deep convolutional neural nets, International Journal on Artificial Intelligence Tools 17 (03) (2008) 555–567.

[11] D. C. Cireşan, U. Meier, J. Schmidhuber, Transfer learning for latin and chinese characters with deep neural networks, in: The 2012 international joint conference on neural networks (IJCNN), 2012, pp. 1–6.

[12] C. Kandaswamy, L. M. Silva, L. A. Alexandre, J. M. Santos, J. M. de Sá, Improving deep neural network performance by reusing features trained with transductive transference, in: International Conference on Artificial Neural Networks, Springer, 2014, pp. 265–272.

[13] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Advances in Neural Information Processing Systems, 2014, pp. 3320–3328.

[14] C. Kandaswamy, L. M. Silva, L. A. Alexandre, J. M. Santos, J. M. Sá, Artificial Neural Networks and Machine Learning – ICANN 2014: 24th International Conference on Artificial Neural Networks, Hamburg, Germany, September 15-19, 2014. Proceedings, Springer International Publishing, Cham, 2014, Ch. Improving Deep Neural Network Performance by Reusing Features Trained with Transductive Transference, pp. 265–272. doi:10.1007/978-3-319-11179-7_34.

[15] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: D. Blei, F. Bach (Eds.), Proceedings of the 32nd International Conference on Machine Learning (ICML-15), JMLR Workshop and Conference Proceedings, 2015, pp. 97–105. URL http://jmlr.org/proceedings/papers/v37/long15.pdf

[16] S. S. Tirumala, A deep autoencoder-based knowledge transfer approach, in: Proceedings of International Conference on Computational Intelligence and Data Engineering, Springer, 2018, pp. 277–284.

[17] D. Han, Q. Liu, W. Fan, A new image classification method using cnn transfer learning and web data augmentation, Expert Systems with Applications 95 (2018) 43–56.

[18] A. Khatami, M. Babaie, H. R. Tizhoosh, A. Khosravi, T. Nguyen, S. Nahavandi, A sequential search-space shrinking using cnn transfer learning and a radon projection pool for medical image retrieval, Expert Systems with Applications 100 (2018) 224–233.

[19] A. X. Wang, C. Tran, N. Desai, D. Lobell, S. Ermon, Deep transfer learning for crop yield prediction with remote sensing data, in: Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies, 2018, pp. 1–5.

[20] S. Latif, R. Rana, S. Younis, J. Qadir, J. Epps, Cross corpus speech emotion classification-an effective transfer learning technique, arXiv preprint arXiv:1801.06353 (2018).

[21] S. S. Tirumala, A. Narayanan, Hierarchical data classification using deep neural networks, in: International Conference on Neural Information Processing, Springer, 2015, pp. 492–500.