Hierarchical data classification using Deep Neural Networks

Sreenivas Sremath Tirumala and A. Narayanan

AUT University, Auckland, New Zealand ssremath@aut.ac.nz

Abstract. Deep Neural Networks (DNNs) is becoming an increasingly interesting, valuable and efficient machine learning paradigm with implementations in natural language processing, image recognition and hand-written character recognition. Application of deep architectures is increasing in domains that contain feature hierarchies i.e., features from higher levels of the hierarchy formed by the composition of lower level features. However it is not clear about the efficiency of DNNs in classifying the hierarchical data which is the focus of this paper. This study is organized into two parts. Firstly, a taxonomic hierarchical data is generated and a DNN is trained to classify the organisms into various species depending on the characteristics. The second step involves testing the ability of DNNs to identity whether two given organisms are related or not. The experimental results show that the accuracy of the results is reduced with the increase in 'depth'. Further, a better performance was achieved when every hidden layers has same number of nodes compared with the experiment where each hidden layer has different number of nodes.

1 Introduction

Artificial Neural Networks (ANN) became 'once again' the point of focus due to the success of 'Deep Learning'. Since the advances in the research of Support Vector Machines (SVM), most of the researchers turned toward Support Vector Machines (SVMs) and other machine learning paradigms. Recent work by Nitish and Hinton has addressed the problem of over-fitting, which is considered as a major drawback of neural networks [1]. The concept of DNNs was proposed in 1989 as Convolutional Neural Networks (CNN) without using the word 'Deep'. Back Propagation (BP) was used to train CNNs and was proven less successful due to limitations of BP. After the introduction of new greedy layer-wise training followed by supervised training of the entire network, ANNs once again came into lime light in the form of DNNs [2]. The learning mechanism of DNNs is called Deep Learning and proved to be successful over SVM based systems [3]. Since then DNNs became increasingly successful with applications in natural language processing [4] [5], image recognition [6] [7] [8], visual recognition [9], computer vision [10] [11], text mining [12] and hand-written character recognition [13]. Corporate giants like Apple, Google and Microsoft are using Deep Learning principles for their services whereas Facebook and Twitter have invested in the research for understanding the features of social interactions.

Application of deep architectures is increasing in the domains that contain feature hierarchies i.e., features from higher levels of the hierarchy formed by the composition of lower level features [14]. Despite its reported success, it is still not clear what the limits of deep learning are. DNNs are quite successful in the case of flat classification and the capability of DNN for hierarchical data classification is not been explored. The aim of this paper is to undertake some exploratory analysis and evaluation of deep architectures using synthetic data known to contain hierarchical features and to evaluate the architecture to identify how exactly to reconstruct the knowledge contained in these hierarchical features. For ANNs with multiple layers, it is important to understand whether an effective conjugation occurs between two (hidden) layers [15]. As a study, we tried to understand this by experimenting with equal number of nodes for hidden layers versus unequal number of nodes.

A synthetic data set is generated with 6 classes (species) of organisms. To measure the hierarchical nature of the data set, the Cophenetic correlation coefficient was calculated from the plotted dendrogram of the data set which is found to be 0.9934 which is considered to be efficient. For the first experiment of classifying the organisms into various species based on characteristics, experiments are conducted with two strategies, varying the depth and changing the number of nodes for every hidden layer. The results show that varying the depth has proven effective in both the cases. Further, the topology with same number of nodes in the hidden layers has proven to be better than having different number of nodes. The detailed observations are presented in experimental results section.

The paper is organized as follows. Section II introduces various types of Deep Architectures. Section III briefly explain about data representation and an in detail explanation about the synthetic data used for the research. Experimental results are presented as Section IV followed by Conclusion and future direction in Section V.

2 Deep Neural Networks

Number of hidden layers of an Artificial Neural Network (ANN) constitutes its depth. If the number of hidden layers is more than 1, such MNNs architecture is said to be 'Deep' and the ANNs as DNN [16]. Feed-forward ANNs with more than one hidden layer units that makes it more efficient than a normal ANNs [17]. Theoretical studies also support the statement that DNNs have the advantage of more efficient representation compared with shallow networks and with less number of hidden units [18]. DNNs, being a simple form of deep architecture implementation uses BP algorithm for training [19] and weights are updated using stochastic gradient descent as

$$\Delta w_{ij}(t+1) = \Delta w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}} \tag{1}$$

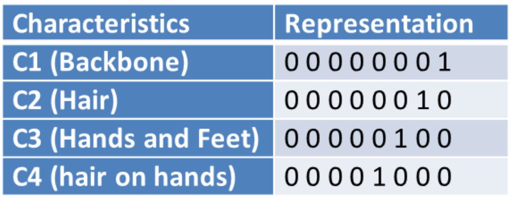
where η represents the learning rate, C is the cost function associated and w_{ij} represents weight.

For large volumes of training data, the process of training DNNs is done in two steps. Firstly, the data is small sized data sets followed the batch-wise training process [4]. With this, there will be an increase in number of parameters to be optimized with which the entire training more complex.

Convolutional Neural Networks or ConvNets are deep architecture based networks, a type of feed-forward ANNs that perform feature extractions by applying convolution and sub sampling. CNN was proposed by Fukushima as Neocognitron. [20] and extended by LeCun [6]. Latest advancement of simplifying the learning process of Neocognitron resulted in the research of artificial vision by Fukushima [21] [22]. Deep Belief Network (DBN) is another type of DNNs based on MLP model with greedy layer-wise training proposed by G.E. Hinton [23]. DBNs have multiple interconnected hidden layers where each layer acts as an input to the next layer without lateral connection between the nodes present in that layer. DBN uses probabilistic logic nodes and uses activation function. Stacked auto-encoders was proposed by Yosgu Bengio by implementing encoding and decoding mechanism using ANNs. The main aim of auto-encoders is to reproduce the input [24]. Initially both encoder and decoder networks are assigned with random weights and trained by observing the discrepancy between original data. The error is back propagated through the decoder network followed by encoder network. The training procedure is similar to DBNs. Stacked De-noising auto-encoder algorithm was proposed in 2010 with which the performance gap between RBM based and auto-encoder based deep architectures was narrowed [25].

3 Data Representation

Binary number representation enables to generate large data set using just two digits 0 and 1. Connectionist methods of data representation can be categorized into specific (localist) or spread out (distributed). Common definition of localist representation in the correct context is, in localist representation each neuron or unit is associated with a single characteristic and each characteristic is represented by one and only one neuron or unit [26]. Localist representation is simple, easy to code and understand. However, localist representation cannot be used for componential structure based data. In distributed representation a single concept is represented by a combination of neurons or units and each neuron or unit can be a part of multiple representations [27–29]. Therefore, in a distributed representation, an isolated neuron has no meaning or cannot be interpreted and existence of neuron has meaning only when it is present in a group. Distributed representation is quite efficient and best suitable for gradient based learning. With binary encoding, n neurons can produce 2^n patterns when distributed representation is used where as the number is very limited in case of localist representation.

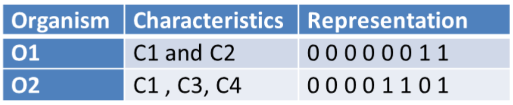


Characteristics	Representation
C1 (Backbone)	0 0 0 0 0 0 0 1
C2 (Hair)	0 0 0 0 0 0 1 0
C3 (Hands and Feet)	0 0 0 0 0 1 0 0
C4 (hair on hands)	0 0 0 0 1 0 0 0

Fig. 1. Characteristics representation

A character, a localist representation is a unique feature like having backbone, hair etc., which determines the uniqueness of the organism. However, the organism has multiple characteristics, which is represented as within 8 bits. The organism that has a backbone is coded as C1 with last bit a 1 and is represented as 0 0 0 0 0 0 0 1. Similarly, other characteristics may be represented as shown in Fig.1.

As mentioned earlier, the categorization of organisms into taxa is based on the characteristics they possess. Since each characteristic of an organism is represented in bits, organisms with multiple characteristics are represented as a combination of binary bits. For instance, organism O1 has backbone and hair which are C1 - 00000001 and C2 - 00000010 as presented in Fig.1. So, the characteristics of organism O1 is represented as 00000011 with combined characteristics as shown in Fig.2. Similarly organism O2 has backbone, hair, and hair on the hands (C1, C2 and C4) which is represented as 00001011.

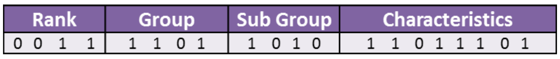


Organism	Characteristics	Representation
O1	C1 and C2	0 0 0 0 0 0 1 1
O2	C1 , C3, C4	0 0 0 0 1 1 0 1

Fig. 2. Multiple Characteristics Representation

A Particular combination of characteristics of the organism determines its Sub-Group. The second level of hierarchy is formed by grouping the combination Sub-Groups as a single Group. For instance, the combinations of 4 bits of sub-groups are coded in group as a combination of bits. We enforce coarse coding paradigm for representing organism as each neuron or unit is part of multiple representation. Coarse coding is a type of distributed representation where a pattern of individual units with different combinations are used for higher representation [27]. The individual units in coarse code have no property or the property is inaccurate. However, pooling them together in a combination constitute a meaningful representation [30].

We represent the organism as a stream of binary data of 20 bits categorized into Rank, Group, Sub-Group and characteristics with four bits each for Rank,



Rank	Group	Sub Group	Characteristics
0 0 1 1	1 1 0 1	1 0 1 0	1 1 0 1 1 1 0 1

Fig. 3. Binary Representation of Organism

Group and Sub-Group and rest of eight bits for characteristics as shown in Fig.3. Selecting 20 as the size of the representation is justified as it can produce 2^{20} different combinations. Consider the following representation of an organism 0 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1. The first 4 bits represent rank, followed by 4 bits each for Group and Sub-Group respectively and the last 8 bits for characteristics. The Taxonomic Rank is determined by the shared characteristics, Group and Sub-Group. Further, with this rank, the hierarchy of the organism can also be determined.

Hierarchical Data can be defined as data units with hierarchical based inter relations among them. Multi-classification problems can be solved using hierarchical classification by pre-arranging the data into hierarchy. Most of the real world problems has hierarchical data. A taxonomic data is Taxa based hierarchical data to represent groups of organisms organized by species name or rank for easy and efficient management of data as well as retrieval. A Hierarchical tree is constructed from the synthetic data of the organisms and Fig.4represents its dendrogram.

The Cophenetic correlation coefficient determines the efficiency of hierarchical structure by determining similarity of the data between two values by calculating the distance between a pair of un modelled data within the dendrogram [31]. To determine the efficiency of the hierarchical data, Cophenetic correlation coefficient is calculated and the typical value for this is around above 0.8 and values above 0.95 are considered as more efficient [32]. The Cophenetic correlation coefficient calculated from the dendrogram (refer to Fig.4) is 0.9934. This values highlights that the synthetic data is efficiently structured with considerable accuracy.

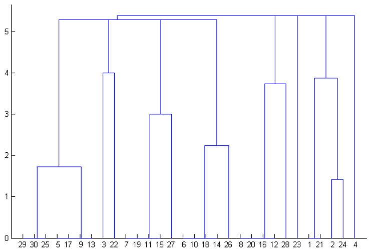


Fig. 4. Hierarchical structure of the Data

4 Experimental Results and Discussion

This experimental study is divided into two parts. Firstly, a taxonomic hierarchical data is generated and a DNN is trained to classify the species depending on the characteristics. In the second experiment, a second set of data is used to identify whether the given two organisms are related or not. For type 1 experiment, a 5-layer neural network topology with three hidden layers as shown in Fig.**??**. Firstly, we used 30, 40, 50 nodes for experiment and then changed it to 30, 30, 30 to determine the influence of symmetric and asymmetric node count. BP is used for training with learning rate and momentum fixed at 0.3 and 0.1 respectively. Auto-encoder style of layer wise training is adopted for the experiments. Block Data Division is adopted for dividing the data set. For all the experiments, data set is divided randomly with first 60% for training next 10% for validation and final 30% for testing which is considered to be sufficient.

The experiment is repeated for a 7-layer neural network with 4 hidden layers with 30,30,30,30 nodes and 30,40,50,60 nodes for type 1 and type 2 experiments respectively

For first set of experiments, 20 inputs representing the 20 bits of the organism is used . 6 outputs determine the species of the organism. Total number of samples used for this experiment is 90 with each one belonging to one of the six different species. The experiment is run with 100 epochs for 10 times and the results obtained are presented as Table 1. Confusion matrix, error histogram and performance graph for each experiment for experiments 1A, 1B, 1C and 1D are presented in the appendices A, B, C and D respectively.

EXP No.	**TRAINING**	**VALIDATION**	**TESTING**	**ALL**
1A	100%	100%	100%	100%
1B	100%	100%	81.5%	94.4%
1C	100%	100%	92.6%	97.8%
1D	13%	55.6%	14.8%	17.8%

Table 1. Results of Experiment - I: Confusion Matrix values

The second set of experiments is carried out to identify whether two organisms are related or not. For example, Tiger is related to Cat since they form the same species whereas Rat is not related to cat. The parameters used for these experiments are same as Experiment - I. The input in this case is a 40 bit binary numbers fed to the network resulting in either '0' for not related or '1' if related. 60 data samples are used for this experiment and the results are shown in Table 2. Results with confusion matrix, error histogram and performance graph for each experiment are presented as Appendices E, F, G, H for the experiments 2A, 2B, 2C and 2D respectively.

The first experiment 1A in which the hidden nodes are 30,30,30 has showed 100% results for training, validation and testing whereas when the number of

EXP No.	TRAINING	VALIDATION	TESTING	ALL
2A	100%	100%	88.9%	96.7%
2B	100%	100%	100%	100%
2C	100%	100%	83.3%	95.0%
2D	100%	100%	93.4%	98.3%

Table 2. Results of Experiment - II: Confusion Matrix values

nodes in the hidden layers are changed to 30,40,50 there has been a variation in the testing results which is 81.5% constituting the overall results as 84.5% as shown in Table 1. However, when the depth of the neural network is increased to 4 the confusion matrix showed a little variation for same number of hidden nodes experiment (1C) whereas the results of the experiment with different number of hidden nodes (1D) showed a drastic fall in the accuracy rate with 17.8% as overall percentage. For experiment 1A, the best validation performance is 0.0003588 at epoch 10 where as for 1B it is 0.00081271 at epoch 12. From the confusion matrix (refer Appendix-B), it is evident that the classification error has occurred for 5 species with 3 of class 5 been classified as class 4 due to similarity in most of their characteristics. The performance difference between experiment 1A (3 hidden layers equal nodes) and 1C (4 hidden layer equal nodes) is 2.2% in the favour of 1A which may be ignored. However, the difference between 1B (3 hidden layers and different number of nodes) and 1D (4 hidden layer different number nodes) is 76.6% in the favour of 1C the reason being the inefficient combination of number of hidden nodes and the depth of the network. On the other hand if we analyse the significance of same number of nodes and different number of nodes with depth being same, the difference between 1A and 1B is 5.6% in favour of 1A and 1C and 1D is 80% in favour of 1C.

The results of the second set of experiments for identifying whether two organisms are related or not are quite different compared to that of the first experiment. In first experiment better results are achieved with topology having same number of hidden nodes. In this experiment, better results are achieved by the topology with different number of hidden nodes. The experimental results are illustrated in detail as appendices D, E, F, and G for experiments 2A, 2B, 2C and 2D respectively. The difference between overall accuracy for experiments with 3 hidden layers, 2A (same number of nodes) and 2B (different number of nodes) is 3.3% in favour of 2B. In case of experiments with 4 hidden layers, experiment 2D (different number of nodes) is 3.3% more accurate than 2C(same number of nodes). When the performance difference is analysed in terms of depth, the topology with 3 hidden layers (2A and 2B)has better performance than the 4 hidden layered topology (2c and 2D) with an average difference of 5.6% and 6.4% respectively.

5 Conclusion and Future Work

The aim of this paper is to identify the efficiency of DNNs in classifying hierarchical data as well as the influence of 'depth', the symmetry of number of hidden nodes. A hierarchical data set is generated with 6 classes (species) of organisms. The Cophenetic correlation coefficient value of 0.9934 confirms the hierarchical nature of the data set. A set of experiments are conducted by varying the depth and changing number of hidden nodes. The first sets of experiments are to classify the species and second set to identify the relationship between the species. The experimental results show that the 'depth' has negative effects on the accuracy of the results especially in the case of classifying hierarchical data. Interestingly, the experiments with same number of hidden nodes have better results compared with that of different number of hidden nodes.

However, the conclusions are based on the synthetic data set generated and with only two types of topologies with 3 and 4 numbers of hidden layers. It will be interesting to observe the results with high volume data set and increasing the number of hidden layers. Experiments do need to be conducted, decreasing the number of hidden nodes. Another direction of study could be the identification of species with limited number of inputs. It is also not clear as how to extracting knowledge of hierarchical features in a human intelligible way from deep learning architectures. Further, Knowledge extraction from deep neural networks so as to reconstruct the hierarchy could be one more possible direction.

References

1. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
2. J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. Published online 2014; based on TR arXiv:1404.7828 [cs.NE].
3. K. P. Murphy, *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012.
4. G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.
5. L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 8599–8603, May 2013.
6. Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 253–256, May 2010.
7. J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *In NIPS*, 2012.

8. S. Gao, L. Duan, and I. Tsang, "Defeatnet – a deep conventional image representation for image classification," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.
9. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.
10. Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
11. C. Xiong, L. Liu, X. Zhao, S. Yan, and T. Kim, "Convolutional fusion network for face verification in the wild," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.
12. D. Hingu, D. Shah, and S. S. Udmale, "Automatic text summarization of wikipedia articles," in *Communication, Information Computing Technology (ICCICT), 2015 International Conference on*, pp. 1–4, Jan 2015.
13. A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," 2009.
14. L. Wang, T. Liu, G. Wang, K. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *Image Processing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.
15. J.-C. Ban and C.-H. Chang, "The learning problem of multi-layer neural networks," *Neural Networks*, vol. 46, no. 0, pp. 116 – 123, 2013.
16. C.-H. Chang, "Deep and shallow architecture of multilayer neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, 2015.
17. G. Tesauro, "Practical issues in temporal difference learning," in *Machine Learning*, pp. 257–277, 1992.
18. Y. Bengio and O. Delalleau, "Shallow vs. deep sum-product networks," in *In Advances in Neural Information Processing Systems 24 (NIPS11), 2011. Xavier Glorot, Antoire Bordes, and Yoshua Bengio. Deep*, 2011.
19. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Neurocomputing: Foundations of research," in *Neurocomputing: Foundations of Research* (J. A. Anderson and E. Rosenfeld, eds.), ch. Learning Representations by Back-propagating Errors, pp. 696–699, Cambridge, MA, USA: MIT Press, 1988.
20. K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.
21. K. Fukushima, "Artificial vision by multi-layered neural networks: Neocognitron and its advances," *Neural Networks*, vol. 37, pp. 103–119, 2013.
22. K. Fukushima, "Training multi-layered neural network Neocognitron," *Neural Networks*, vol. 40, pp. 18–31, 2013.
23. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, 2006.
24. Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, U. D. Montral, and M. Qubec, "Greedy layer-wise training of deep networks," in *In NIPS*, MIT Press, 2007.
25. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.
26. H. B. Barlow, "Single units and sensation: a neuron doctrine for perceptual psychology?," *Perception*, vol. 1, no. 4, pp. 371–394, 1972.

27. G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," in *Distributed Representations* (D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, eds.), pp. 77–109, Cambridge, MA, USA: MIT Press, 1986.
28. T. V. GELDER, "Defining distributed representation," *Connection Science*, vol. 4, no. 3-4, pp. 175–191, 1992.
29. S. Sanjeevi and P. Bhattacharya, "A connectionist model for predicate logic reasoning using coarse-coded distributed representations," in *Knowledge-Based Intelligent Information and Engineering Systems* (R. Khosla, R. Howlett, and L. Jain, eds.), vol. 3682 of *Lecture Notes in Computer Science*, pp. 732–738, Springer Berlin Heidelberg, 2005.
30. M. R. W. Dawson, P. M. Boechler, and M. Valsangkar-Smyth, "Representing space in a pdp network: Coarse allocentric coding can mediate metric and nonmetric spatial judgements.," *Spatial Cognition and Computation*, vol. 2, no. 3, pp. 181–218, 2000.
31. R. R. Sokal and F. J. Rohlf, "The comparison of dendrograms by objective methods," *Taxon*, vol. 11, no. 2, pp. 33–40, 1962.
32. F. J. Rohlf and D. R. Fisher, "Tests for hierarchical structure in random data sets," *Systematic Biology*, vol. 17, no. 4, pp. 407–412, 1968.