ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse57932020.pdf
https://doi.org/10.30534/ijatcse/2020/57932020

Classification of Dataset Using Deep Belief Networks Clustering Method

Rotimi-Williams Bello[1], Zidiegha Seiyaboh[2], Daniel A. Olubummo[3], Abdullah Zawawi Talib[4]
[1,4]School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau, Pinang, Malaysia, [1]sirbrw@yahoo.com, [4]azht@usm.my
[2]Institute of Research in Applicable Computing (IRAC), University of Bedfordshire, LU1 3JU United Kingdom, seiyabohzidiegha@gmail.com
[3]Department of Computer & Information Systems, Robert Morris University, Moon-Township, Pennsylvania, USA, adanielolu@yahoo.co.uk

ABSTRACT

Dataset in large collection involves considerable handling in its analysis especially when it is being employed in classification problems that involve big data. Due to the technology development, the manner and approach in which this dataset is being manipulated for classification purposes differ not only in one respect but in many respects with different uncorrelated results which sometimes make prediction inaccurate. By definition, classification is the act of arranging objects into classes or categories of the same type; these objects can be huge or otherwise, and to manually classify them will be a herculean task. The basic reason for classification is to punctiliously predict the class for each case in the dataset using class label. Notable classification, clustering and regression methods are support vector machines, neural networks, random forest, k-nearest neighbor and decision trees. The conventional clustering method that is widely employed to classification problems cannot handle the weight associated problem which characterized the transmission of neurons from layer to layer within the network. Employed in this work for the classification and clustering resolution is deep belief networks clustering method. The neural network architecture and loss function popularly employ in deep learning are considered for transforming the input data to clustering-friendly feature representation.

Key words : Classification, Clustering, Deep learning, Deep belief networks, Dataset.

1. INTRODUCTION

Lately, applications of deep belief networks (DBNs) have gained great acceptance in various fields due to the advancements in the underlying structure of their layer-by-layer method of learning [1], [2]. The evidence is substantiated by their applications in various areas of machine learning such as image characterization [3], speech identification and recognition [4], [5], information mining and retrieval [6], [7], [8], and natural language processing and understanding [9]. There are so many classification models related to DBNs that are employed for different tasks, they are 3D object recognition [10], recognition of hand-written characters [2], [11], modeling of data that is captured on motion [12], [13], and alphabet transliteration [14]. DBNs, being probabilistic generative models comprise of combined and trained restricted Boltzmann machines (RBMs) with layer of visible variable and various layers of hidden variables. DBNs training approach is by greedy unsupervised approach with a back-propagation fine-tune procedure among the layers excluding the visible layer and the output layer for effective and optimal classification tasks performance. Although normal networks such as convolutional neural networks have modeling ability, their networks are not as deep as the networks of deep belief networks which make the networks to have greater modeling ability than the rest. There are two approaches to training DBNs models, namely the generative models approach and the discriminative models approach. Back-propagation method is employed for the DBNs discriminative training whereby to get the over-fitting of the data reduced, a multilayer feed-forward neural network is initialized by DBNs using the features produced in each layer when there is fewer training sample for supervised classification [9], [14], [15].

According to [3] and [4], DBNs as probabilistic generative models comprise of variables, hidden in the various layers of the networks. These hidden variables are otherwise known as the DBNs' hidden units; the various layers of the networks have connections with one another and not with each layer's unit. The upper two layers of DBNs do not have direct links with the remaining layers, while on the contrary the bottom layers have direct links.

Unsupervised networks of trained RBMs are arranged into stacks to form DBNs; each hidden layer of RBMs' sub-network serves as a visible layer for the next hidden layer. The benefit in deep architecture is that each layer in the network learns the complex features of the input data more

than the layers before it. With initial weight initialization, DBNs and RBMs could be employed as feature extraction methods. It is expected of DBNs to perform better than the conventional neural network considering their connection weights initialization rather than the random weights initialization of neural network. For input reconstruction, each layer in DBNs relies on contrastive divergence method which boosts the network's performance.

The remainder of this work is as follows: Described in section 2 are preliminaries and taxonomy. Using of DBNs for clustering and classification is explained in section 3. Presented in section 4 are the results and discussion. Section 5 concludes the work with future work.

2. PRELIMINARIES AND TAXONOMY

The network architectural terminologies needed for understanding deep clustering are briefly discussed in this section, namely neural network which includes feed-forward and fully-connected neural network, convolutional neural network, deep belief networks, and loss functions.

2.1 Application of neural network architecture for deep clustering

Many literatures have recorded the effectiveness of some trained neural network architectures in learning to extract and represent features newly. The following are few of the popular neural networks that deep learning utilizes in extracting and representing features.

2.1.1 Feed-forward and Fully-connected Neural Networks

A feed-forward neural network (FNN) as an opposite of recurrent neural network is an artificial neural network that has its nodes connection non-cyclical instead, from the input nodes, via the hidden nodes to the output nodes, the information moves in only one direction without forming any network loops as shown in Figure 1. Moreover, fully-connected network (FCN) is made up of multiple layers of neurons, whereby each neuron in the network is connected to every network neuron preceding its layer with individual connection having its own weight. FCN, also known as multi-layer perceptron (MLP) is entirely all-purpose pattern of connection which does not assume features in the data instead, its use is necessary in a supervised learning scenario where there is provision for labels. Nonetheless, when it comes to clustering, it is necessary to have a good initialization of network's parameters to guard against trivial solution as a result of all data points being mapped out to tight clusters due to naif FCN which can result to a little value of clustering loss [16].

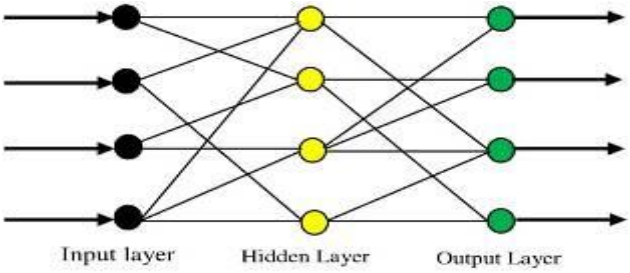

Figure 1: Feed-forward Network

2.1.2 Convolutional Neural Network

The inspiration that led to the introduction of convolutional neural networks (CNNs) [17] in classification process is not far from biological process whereby the pattern in which neurons are connected is motivated by the visual cortex structure of the animal. Similarly, in convolutional neural network, each neuron in the network layer is not fully connected to every previous neuron preceding its layer but is only connected to a few neurons (depends on kernel size) in the layer preceding its layer, and the same set of weights is employed for each neuron meaning that individual connection does not have its own weight as illustrated in Figure 2. The application of CNNs to image datasets helps in vicinage and shift-invariance of feature extraction. It does not require any initialization for it to be directly trained with a particular clustering loss, but the clustering performance can be boosted with an excellent initialization.

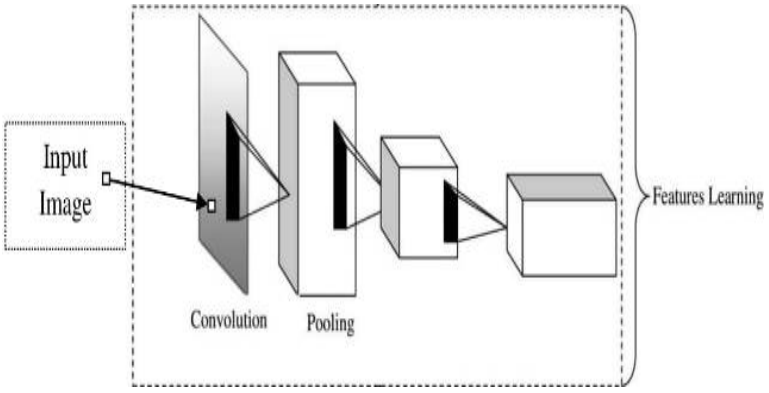

Figure 2: Convolutional Neural Network

2.1.3 Deep Belief Networks

Deep belief networks (DBNs) [18] comprise of stacks of trained RBMs [19] forming a generative graphical model with the ability to learn the feature extraction of the input data and even represent the input data in a deep hierarchical manner as shown in Figure 3. DBNs are different from other networks because of the manner in which the weights are initialized. The training of the networks using unsupervised greedy layer-wise method is adopted in training DBNs having RBMs as the building blocks for each concerned layer. After this, the necessary DBNs' parameters are fine-tuned taking into consideration some loss function. Employing back-propagation with greedy layer-wise weight initialization

makes DBNs perform better than employing back-propagation with random weight initialization as found in the conventional neural network.

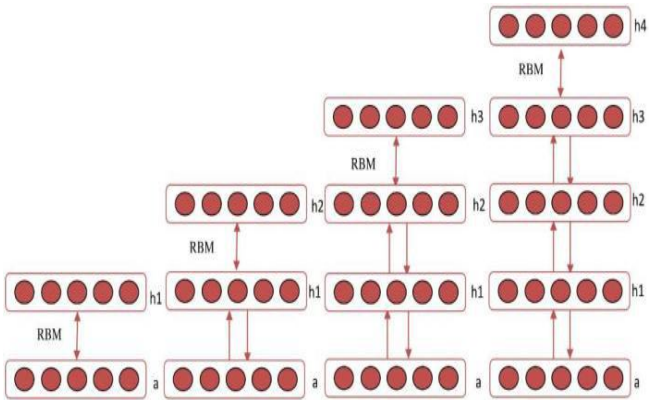

Figure 3: Stacked Restricted Boltzmann Machines-based Deep Belief Networks [1]

2.2 Clustering Loss functions

Clustering lost functions are used in machine learning and deep leaning to guide and evaluate how well the networks learn and represent the input data features in a clustering-friendly manner. Principal clustering loss and auxiliary clustering loss are the most major kinds of clustering loss functions. These are discussed in sub-section 2.2.1 and sub-section 2.2.2

2.2.1 Principal Clustering Loss

This kind of clustering loss functions includes the cluster centroids of the datasets and the cluster assignments of the datasets. Otherwise stated, the clusters can be got directly after the clustering loss has guided in the network training. This kind of clustering loss has the following as options, namely k-means loss [16], [20], and agglomerative (Hierarchical) clustering loss [21], [22], cluster classification loss [23], cluster assignment hardening and so on.

2.2.2 Auxiliary Clustering Loss

This second category of clustering loss functions exclusively takes care of guiding the network for more feasible representation learning for clustering, although it is difficult if not impossible for it to generate clusters directly. The implication of this is that, to obtain the clusters, any deep clustering methods with simply auxiliary clustering loss need running a clustering method after the network training.

Locality-preserving loss [24] as one of the many auxiliary clustering losses employed in deep learning enforces the network for the preservation of the cluster's locality. Group sparsity loss [24] on the other hand is one of the many auxiliary clustering losses that inspired by spectral clustering where exploitation of block diagonal similarity matrix is done for the purpose of representation learning and so on.

3.USING DBNs FOR CLUSTERING AND CLASSIFICATION

Classifying datasets using deep belief networks clustering method is the main focus of this study. Image dataset of approximately 1000 cow objects and 4 classes were provided for the classification and clustering process. 70% of the dataset was employed for the training, 10% for validation and 20% for testing. The RBMs employed were earlier built using binary probabilistic units for the network layers which include both the input layer and the hidden layers. Training on neurons and valued data that is continuous is much slower compare to the training with binary inputs; this means that if the training process with binary inputs is pretty slow, then, it would have been impracticable training on continuous input. Addition of noise to sigmoid units is part of the past work carried out on continuous valued data in RBMs. In clustering the input data, the input has been scaled to 0 and 1 interval by the DBNs approach this work employed. Four layers of RBMs formed the DBNs; the first layer which serves the purpose of data entry is regarded as the layer that is visible, followed by the layer that is hidden which becomes visible to the next RBMs hidden layer. The final hidden layer of the stacked RBMs represents the output of the DBNs which is otherwise known as the class label unit of the DBNs.

By employing Gibbs' method, the first layer of the RBMs is trained with 1000 iterations thereafter passing the output to the next RBMs layer for training with the same number of iteration. The diagrammatical representation of this process is shown in Figure 3. The systematic process of DBNs in handling clustering and classification task as shown in Figure 3 is sum up as follows: the first layer which represents the visible layer of the first stack of RBMs accepts the input data via its visible nodes and transfers the received input in a modeled fashion to the next layer which is the hidden layer within the first stack of RBMs to complete the first round of the RBMs stack process. Subsequently, all the hidden layers of the remaining stacks of RBMs become visible to each other in order to enable the passing of the modeled input from one layer to another until it gets to the final layer, which is the class label unit with one output unit for classification. To get to the last layer, the DBNs are discriminately fine-tuned by employing a back-propagation algorithm through which the probability of the class labels is computed.

Mathematically, as previously explained, the rule for upgrading the weights of the layer that is visible to layer that is hidden is given by

$$\Delta w_{ij} = \varepsilon \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \right) \quad (1)$$

In view of the fact that there are no hidden layer-to-hidden layer connections, it is not difficult to compute $\langle v_i h_j \rangle_{data}$.

Nonetheless, computing $\langle v_i h_j \rangle_{model}$ is not computationally cheap. Contrastive divergence method is employed to make the computation more rapidly fast and easy in such a way that

$<v_i h_j>_{recon}$ had to substitute for the $<v_i h_j>_{model}$. Then, the upgrade rule becomes

$$\Delta w_{ij} = \varepsilon \left(<v_i h_j>_{data} - <v_i h_j>_{recon}\right) \quad (2)$$

To compute $<v_i h_j>_{recon}$ training vector is employed for the initialization of the units that are visible. Afterward, using equation (3), the hidden units are calculated. Afterward, using equation (4), the visible units v_i are recalculated, and thereafter renamed as recon as in equation (2). Lastly, using the reconstructed visible states, the states of the hidden units are computed.

$$P(h_j = 1|v) = \sigma\left(\Sigma v_i w_{ij} + a_j\right) \quad (3)$$

Where σ is the sigmoid function $\sigma(x) = (1 + \exp(-x))^{-1}$. For a given hidden vector *h*, an unbiased state of visible data unit *i* is obtained by

$$P(h_j = 1|v) = \sigma\left(b_i + \Sigma\, h_j w_{ij}\right) \quad (4)$$

Table 1: Algorithm of the Deep Belief Networks Classifier

Initializing gn to be equal to 1000
Initializing ε to be equal to 0.1
Reading input range of [0,1] into [NI] and [NF]
Scaling input range of [0,1] into [NI] and [NF]
Selecting the features that are discriminating
Initializing n to be equal to 4
Initializing the number of units that are hidden for each RBM
Initializing W randomly
Naming W
Clustering the output
Assigning a class label
Running objects in the testing dataset
According to the cluster range, the object's class is defined by the output of the DBNs layer.

gn = Number of Gibbs methods
ε = Epsilon value
NI = Number of inputs
NF = Number of features
W = Weights of DBNs
N= Number of RBMs

4 RESULTS AND DISCUSSION

Application of DBNs has been carried out on 4 classes of 1000 cow objects for classification purposes. 70% of the available dataset was employed for training, 10% for validation and 20% for testing. The DBNs output was divided into four distinct clusters, each having two classes with the following intervals: first cluster [0, 0.246], second cluster [0.246, 0.87], third cluster [0.87, 1], fourth cluster [1, 1.02]. Table 1 is the algorithm of the deep belief network classifier that shows the steps involved in the network execution of the classification task. Table 2 is the experimental results of the test carried out on the trained DBNs having the architecture of 500 nodes for each layer of the hidden layers. 20% of the available dataset was employed for the testing and the range that each object's output lies in was found. Based on the resulted range, each object class was determined and the result compared with associated class label. The classification result of testing using 20% of the dataset yielded 92.3%.

Table 2: Experimental Results

Hidden Layer Topology	Pre-training Learning Rate	Pre-training Epoch	Fine-tuning Learning Rate	Fine-tuning Epoch	Validation Error	Test Error
500	0.01	100	0.1	100	29.2	25.8
500	0.01	150	0.1	100	26.4	29.6
500	0.01	150	0.1	100	25.8	25.4
500	0.01	300	0.1	200	23.6	23.2

5 CONCLUSION

For a better clustering result, it is normal to join clustering algorithms and deep learning together. Moreover, considering the fact that deep clustering is usually employed in a lot of realistic applications due to its feature extraction ability, in this study, we have elucidated on how clustering of data and classification of data could be achieved by unsupervised learning of DBNs. Test carried out on cow dataset as a continuous dataset supported this approach, although with continuous dataset posing a challenge which was overcome using scale interval of 0 and 1. As it is widely known that reliable results have be produced in dimensionality reduction using undirected deep architecture, it shows that dimensionality reduction could be achieved with unsupervised learning of DBNs. Also, the taxonomy of deep clustering presented in this study comprehensively describes the importance and unimportance of deep clustering algorithms.

Although getting the networks and clustering models optimized together drastically improve the performance of the cluster, there is little theoretical evidence explicating the practicality. More exploitation of the theoretical aspect of deep clustering will anchor future work in this field.

REFERENCES

[1] Bello, R. W., Talib, A. Z., Mohamed, A. S. A., Olubummo, D. A., & Otobo, F. N. (2020). **Image-based Individual Cow Recognition using Body Patterns.** International Journal of Advanced Computer Science and Applications, 11(3), 92-98. https://doi.org/10.14569/IJACSA.2020.0110311

[2] Hinton, G. E. (2002). **Training products of experts by minimizing contrastive divergence.** Neural computation, 14(8), 1771-1800.

[3] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). **A fast learning algorithm for deep belief nets.** Neural computation, 18(7), 1527-1554. https://doi.org/10.1162/neco.2006.18.7.1527

[4] Durga Indira, N., Sony, K., Kiran, G. D. S., Lavanya, K. & Sashank, D. P. (2020). **Compressive sensing for speech signals.** International Journal of Advanced Trends in Computer Science and Engineering, 9 (2), 1147-1150. https://doi.org/10.30534/ijatcse/2020/38922020

[5] Zhang, X. L., & Wang, D. (2016). **A deep ensemble learning method for monaural speech separation.** IEEE/ACM transactions on audio, speech, and language processing, 24(5), 967-977. https://doi.org/10.1109/TASLP.2016.2536478

[6] Vasudevan, S. K., Vamsee Krishna Kiran, M., Sini Raj, P., & Thangavelu, S. (2020). **AI approach with increased accuracy to extract the tabular content from pdf and image files.** International Journal of Advanced Trends in Computer Science and Engineering, 9 (2), 1013-1019. https://doi.org/10.30534/ijatcse/2020/18922020

[7] Sutedja, I., Heryadi, Y., Wulandhari, L.A., & Abbas, B. (2020). **Imbalanced data classification using auxiliary classifier generative adversarial networks.** International Journal of Advanced Trends in Computer Science and Engineering, 9 (2), 1068-1075. https://doi.org/10.30534/ijatcse/2020/26922020

[8] Salakhutdinov, R., & Hinton, G. (2009). **Semantic hashing.** International Journal of Approximate Reasoning, 50(7), 969-978.

[9] Welling, M., Rosen-Zvi, M., & Hinton, G. E. (2005). **Exponential family harmoniums with an application to information retrieval.** In Advances in neural information processing systems (pp. 1481-1488).

[10] Hinton, G. E., & Salakhutdinov, R. R. (2006). **Reducing the dimensionality of data with neural networks.** Science, 313(5786), 504-507. https://doi.org/10.1126/science.1127647

[11] Nair, V., & Hinton, G. E. (2009). **3D object recognition with deep belief nets.** In Advances in neural information processing systems (pp. 1339-1347).

[12] Taylor, G. W., & Hinton, G. E. (2009, June). **Factored conditional restricted Boltzmann machines for modeling motion style.** In Proceedings of the 26th annual international conference on machine learning (pp. 1025-1032).

[13] Deselaers, T., Hasan, S., Bender, O., & Ney, H. (2009, March). **A deep learning approach to machine transliteration.** In Proceedings of the Fourth Workshop on Statistical Machine Translation (pp. 233-241). Association for Computational Linguistics. https://doi.org/10.3115/1626431.1626476

[14] Mohamed, A. R., Hinton, G., & Penn, G. (2012, March). **Understanding how deep belief networks perform acoustic modelling.** In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4273-4276). IEEE.

[15] Mohamed, A. R., Dahl, G. E., & Hinton, G. (2011). **Acoustic modeling using deep belief networks.** IEEE transactions on audio, speech, and language processing, 20(1), 14-22. https://doi.org/10.1109/TASL.2011.2109382

[16] Yang, B., Fu, X., Sidiropoulos, N. D., & Hong, M. (2017, August). **Towards k-means-friendly spaces: Simultaneous deep learning and clustering.** In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3861-3870). JMLR. org.

[17] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). **Imagenet classification with deep convolutional neural networks.** In Advances in neural information processing systems (pp. 1097-1105).

[18] Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009, June). **Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.** In Proceedings of the 26th annual international conference on machine learning (pp. 609-616). https://doi.org/10.1145/1553374.1553453

[19] Hinton, G. E. (2012). **A practical guide to training restricted Boltzmann machines.** In Neural networks: Tricks of the trade (pp. 599-619). Springer, Berlin, Heidelberg.

[20] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). **Data clustering: a review.** ACM computing surveys (CSUR), 31(3), 264-323.4 https://doi.org/10.1145/331499.331504

[21] Beeferman, D., & Berger, A. (2000, August). **Agglomerative clustering of a search engine query log.** In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 407-416). https://doi.org/10.1145/347090.347176

[22] Yang, J., Parikh, D., & Batra, D. (2016). **Joint unsupervised learning of deep representations and image clusters.** In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5147-5156). https://doi.org/10.1109/CVPR.2016.556

[23] Hsu, C. C., & Lin, C. W. (2017). **Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data.** IEEE Transactions on Multimedia, 20(2), 421-429.

[24] Huang, P., Huang, Y., Wang, W., & Wang, L. (2014, August). **Deep embedding network for clustering.** In 2014 22nd International conference on pattern recognition (pp. 1532-1537). IEEE.