A Literature Review And Comparative Analysis of Deep Belief Networks

Haress, Ashraf

Artificial Intelligence Major,

Faculty of Informatics & Computer Sciences

The British University in Egypt

El-Sherouk, Egypt

[ashraf196280@bue.edu.eg](mailto:ashraf196280@bue.edu.eg)

# Abstract

This paper re-introduces the Deep Belief Network (DBN) model by stating its history, previous work that leads up to its inception, its mechanism, and its applications. The paper then compares different DBN models on a specific application, in order to identify the more suitable model for said application.

*Keywords:* *Deep Belief Networks, Restricted Boltzmann Machines, Explanation, Comparative Analysis.*

# Introduction

In 1983, Hinton & Sejnowski proposed Boltzmann (BM) & Restricted Boltzmann Machine (RBM) [1]. A Boltzmann Machine is a network model where the nodes are connected in a full mesh topology [2]. It is used to learn important aspects of an unknown probability distribution based on samples from this distribution [3]. Since there are edges (where n is the number of nodes) which is inefficient, the Restricted Boltzmann Machine (RBM) was proposed which excludes edges on the same layer in the network. Figure 1 demonstrates the difference between BM and RBM model.

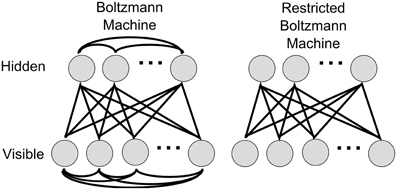
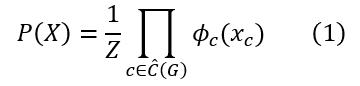


Fig. 1. BM vs RBM regarding connectivity [4]

BMs and RBMs have the same 2 layers: a visible layer, where each neuron corresponds to a component of an observation (i.e. input), and a hidden layer, where each neuron represents the dependency between the components of an observation. Hidden layers are also considered to be non-linear feature detectors [3].

The structure of BM and RBM is an undirected graphical model (UGM), which is also referred as Markov Network, or Markov Random Field (MRF). An MRF is a type of probabilistic graphical model (PGM). The concept of PGM relates to graph and probability theories. In MRF, the random variables represent nodes, and the undirected edges represent the variables which interact (correlate) in some way [5]. Therefore, one can calculate the joint probability of events of a random variable “X” as:



And by assuming we have an undirected graph G = (X, E) where X is the set of nodes and E is the set of undirected edges, then a clique “c” can be defined as a subset of X in which all the nodes of that subset are pairwise connected [6]. For example, in Figure 2, both {x1, x2} and {x1, x2, x3} are cliques of the set of cliques “C”, but only the latter subset is a maximal clique (which means no node can be added to the latter subset such that it will still remain a clique). Therefore, the latter subset is included in the set of maximal cliques “Ĉ” related to graph G (expressed as Ĉ(G)).

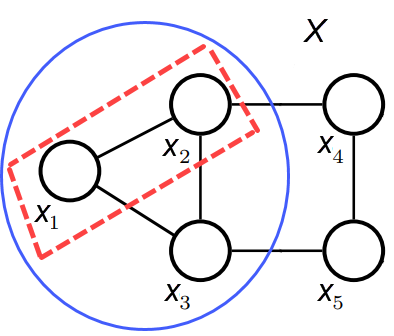
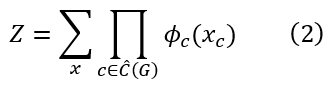
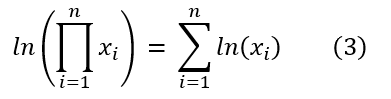


Fig. 2. Clique vs maximal clique. Adapted from [6]

Continuing with the explanation of equation (1), is the potential function applied on each maximal clique that includes an event x such that . This potential function does not output the conditional probabilities of events included in a maximal clique , but rather it outputs a non-negative value which represents a “compatibility score” of how much the events of maximal clique correlate. Moreover, these non-negative values do not have a specific range. Therefore, they have to be normalised in order to be in the range [0, 1], and as such (1) can represent a valid probability distribution model. For that normalization to occur, the product of the potential function is divided by — called the partition function, or the normalising constant — which is the sum of the potential values calculated for maximal cliques of the entire network (i.e. for all events in all of . Succinctly [6]:

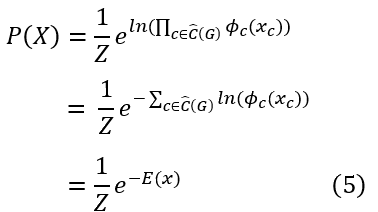


Moreover, since the output of the potential function isn’t restricted to a specific range, the product of such potential functions could be a very large value. Recalling the product property: 

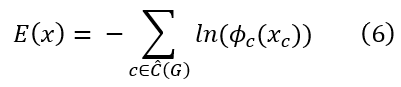
And the exponent raised to rule:



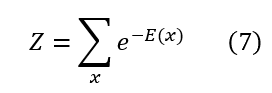
These properties could be utilized to change (1) to:



Such that:



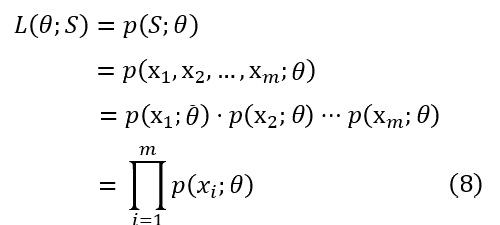
Where is the energy function. Noting that the negative sign was included as the sum of values is negative. Thus, equation (5) represents the probability distribution of MRF, which is also called Gibbs distribution [6]. In addition, we can use (3) and (4) to change (2) into:



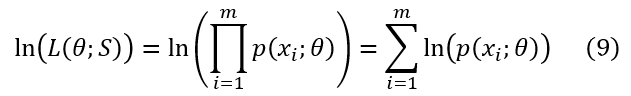
Before explaining the RBM training process, it is important to note that the energy function does not have to be as stated in (6), rather it depends on the structure of the model and how we want represent correlation of data.

.**State that before we start talking about the RBM training process, the energy function is different for the RBM model than that of MRF such that INSERT EQUATION**.

Now, to understand the RBM training process, MRF unsupervised learning must be understood. Unsupervised learning means learning an unknown distribution based on sample data. Let be the parameters of the energy function that we want to learn in order to model a distribution on the examples of the sample data that resembles the data’s distribution, and let the training data and assumed to be independent and identically distributed (i.i.d.). To estimate the parameters , Maximum Liklihood Estimation (MLE) is used. In the context of MRFs, MLE means finding that will maximize the probability of under ; the MRF distribution [6]. If we denote the likelihood function, which is the joint probability density function of , as , then we have:



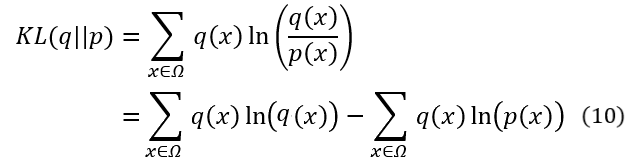
Where the semicolon “;” indicates that the symbols that appear after it are parameters of the probability distribution[[1]](#footnote-1). However, following the Gibbs distribution for MRF, it is generally **not possible** to find analytically using gradients [6] for (7) and its log-likelihood equivalent which is given by:



Therefore:

1. Gradient ascent is applied using Constrastive Divergence (CD).
2. CD uses Gibbs sampling to choose the examples (observations) that the CD will train upon from the data set.

Regarding the former point, to understand CD, we have to first discuss Kullback–Leibler (KL) divergence. if we calculate the distance (i.e. difference) between the unknown distribution of the data and the distribution of the MRF given a finite space , then one can say that minimizing that distance is equivalent to maximizing the likelihood of which is given by in (7). The KL divergence is given by:



Therefore, approximating the expectation of to resemble the expectation of by training samples from results in a higher log-likelihood of . That is why maximizing the log-likelihood corresponds to minimizng the KL divergence.

Now, Contrastive Divergence is defined

.**add Z=epower part above and change all numbers, continue with pdf until KL and updating theta, and stop before “Log-likelihood gradient of MRFs with latent variables” then check that , then how pot. Func differs and we’re using energy**.

we have Z which denotes the sum of probabilities of all nodes (x) in X,

To illustrate MN with an example, suppose you In 2006, Hinton tried to revive neural networks after it came to a halt due to the “vanishing gradient” problem, where he proposed a greedy learning method for RBM: Train every layer of the network using RBM training [5]. In other words, he treated the neural network as stacked layers of RBM, where this interpretation of the network is called the Deep Belief Network (DBN). DBN allowed good initialization of weights (using RBM training) which prevented the vanishing gradient problem.

# Literature Review

The DBN model is used in various applications including solar radiation estimation [7], time-series problems such as draught forecasting [8], speech-recognition problems such as interpreting human commands to execute program tasks [9], and image classification problems such as land cover classification [10]. This paper will focus on the image classification problem. Specifically, various DBN models trained on the MNIST handwritten-digits dataset will be discussed, and compared.

## MNIST Dataset

## Models’ Architectures

## Expiremental Environments

## Parameter Settings

## Expiremental Results

### Interpretation of the Results (Comparitive Analysis)

# Conclusion

[**ASK DR/TA**: Do I just re-state the abstract, and re-state the model that got better results?]

# References

[1] G. Hinton and T. Sejnowski, *Optimal perceptual inference*. 1983, p. 453.

[2] A. Patel and R. K. Rama, *An overview of Boltzmann Machine and its special class*. 2020. doi: 10.13140/RG.2.2.28630.88641.

[3] A. Fischer and C. Igel, *An Introduction to Restricted Boltzmann Machines*. 2012, p. 36. doi: 10.1007/978-3-642-33275-3\_2.

[4] P. O’Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, “Real-Time Classification and Sensor Fusion with a Spiking Deep Belief Network,” *Front. Neurosci.*, vol. 7, p. 178, Oct. 2013, doi: 10.3389/fnins.2013.00178.

[5] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, “Restricted Boltzmann Machine and Deep Belief Network: Tutorial and Survey.” arXiv, Aug. 05, 2022. Accessed: Oct. 20, 2022. [Online]. Available: http://arxiv.org/abs/2107.12521

[6] A. Fischer and C. Igel, “Training restricted Boltzmann machines: An introduction,” *Pattern Recognit.*, vol. 47, no. 1, pp. 25–39, Jan. 2014, doi: 10.1016/j.patcog.2013.05.025.

[7] H. Zang *et al.*, “Application of functional deep belief network for estimating daily global solar radiation: A case study in China,” *Energy*, vol. 191, p. 116502, Jan. 2020, doi: 10.1016/j.energy.2019.116502.

[8] N. A. Agana and A. Homaifar, “EMD-Based Predictive Deep Belief Network for Time Series Prediction: An Application to Drought Forecasting,” *Hydrology*, vol. 5, no. 1, Art. no. 1, Mar. 2018, doi: 10.3390/hydrology5010018.

[9] R. Sarikaya, G. E. Hinton, and A. Deoras, “Application of Deep Belief Networks for Natural Language Understanding,” *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 4, pp. 778–784, Apr. 2014, doi: 10.1109/TASLP.2014.2303296.

[10] B. Ayhan and C. Kwan, “Application of Deep Belief Network to Land Cover Classification Using Hyperspectral Images,” in *Advances in Neural Networks - ISNN 2017*, Cham, 2017, pp. 269–276. doi: 10.1007/978-3-319-59072-1\_32.

1. The first term of (7) is asking for the likelihood of the parameters θ while the data *S* are present, but the second term is asking for the probability of observing *S* while the parameters θ are present. [↑](#footnote-ref-1)