International Journal of Communication and Information Technology

E-ISSN: 2707-6628
P-ISSN: 2707-661X
www.computersciencejournals.com/ijcit
IJCIT 2022; 3(2): 14-17
Received: 11-05-2022
Accepted: 16-06-2022

Kama HN
Department of Industrial Mathematics, Admiralty University of Nigeria, Delta State Nigeria

Mankilik IM
Department of Industrial Mathematics, Admiralty University of Nigeria, Delta State Nigeria

Corresponding Author:
Kama HN
Department of Industrial Mathematics, Admiralty University of Nigeria, Delta State Nigeria

Artificial neural networks: Prediction for restricted Boltzmann machines

Kama HN and Mankilik IM

DOI: https://doi.org/10.33545/2707661X.2022.v3.i2a.50

Abstract
Artificial Neural Network (ANN) is the branch of Artificial Intelligence (AI) that is inspired by the architecture of the human brain. A type of recurrent ANN known as Restricted Boltzmann Machines (RBMs) are probabilistic graphical models that can be interpreted two-layered network of stochastic units with undirected connections between pairs of units in the two layers. RBMs are used specifically as a generative model. The result obtained from Neural Network Model shows 0.000373 errors with 88 steps. Prediction using neural network shows 0.9928202080, 0.3335543925 and 0.9775153014 while Converting probabilities into binary classes setting threshold level 0.5 result shows that the predicted results are 1, 0, and 1.

Keywords: Artificial intelligence, restricted Boltzmann machines, model, probability, neural networks, contrastive divergence learning

Introduction
Restricted Boltzmann Machines is the neural network that belongs to the energy-based model. It is a probabilistic, unsupervised, generative deep machine learning algorithm. During the past decade, the restricted Boltzmann machine (RBM) has received much attention as building blocks for deep belief networks (Hinton and Salakhutdinov, 2006; Bengio, 2009) [7, 19]. The variants and extensions of the RBM have been applied in a wide range of pattern recognition problems, such as handwriting recognition (Hinton and Salakhutdinov, 2006) [7], document processing (Dahl *et al*., 2012; Srivastava *et al*., 2013) [14, 15], and collaborative filtering (Salakhutdinov *et al*., 2007) [5]. Despite great successes, there still lacks an efficient algorithm for training RBMs. The existing algorithms aim to maximize the log-likelihood function of the RBM using a gradient-based method, while the true gradient of the log-likelihood function is intractable. Hinton *et al*. (2006) [7] proposed the so-called Contrastive Divergence (CD) algorithm to train RBMs, where the log-likelihood gradient is approximated based on a short run of Markov chain Monte Carlo (MCMC). Due to the approximation errors, CD does not necessarily converge to the maximum likelihood estimate (MLE) of the parameters as noted by Carreiera-Perpi˜nán and Hinton (2005) [8] and Bengio and Delalleau (2009) [19]. Fischer and Igel (2010) [1] observed that the approximation errors can even lead to a distortion of the learning process; that is, after some iterations the likelihood can start to diverge in the sense that the model systematically get worse if the run of MCMC is not long enough. To address the issue of convergence, some variants of CD have been proposed with a general strategy to obtain better approximation of the log-likelihood gradient by sampling from a Markov chain with a greater mixing rate. These variants include persistent CD (Tieleman, 2009) [11], fast persistent CD (Tieleman and Hinton, 2009) [11], tempered transitions (Salakhutdinov, 2009) [16], and parallel tempering (Desjardins *et al*., 2010; Cho *et al*., 2010) [10, 13]. The majority of these variants, as noted by Schulz *et al*. (2010) [17], include a number of hyper parameters in addition to the more popular heuristics of weight-decay, momentum, and learning rate schedules. However, because exact evaluation of the log-likelihood function is impractical for even a middle-sized RBM, it is unclear how to set the hyper parameters and which heuristic to select.

Methodology
Restricted Boltzmann Machine: A very useful tool for deep learning applications is the restricted Boltzmann machine (RBM), which is a two-layer (or two-group) Boltzmann machine with m visible units

$v_i\ (i = 1, 2, \ldots, m)$ and n hidden units $h_j\ (j = 1, 2, \ldots, n)$ where both v_i and h_j are binary states. The visible unit i has a bias α_i and the hidden unit j has a bias β_j while the weight connecting them is denoted by w_{ij}. The restriction is that their neuron units connecting visible and hidden units form a biparte graph (see Figure 1.3 below), while no connection with the same group (visible or hidden) is allowed.
Using the notations and configuration given by Hinton (2010) [18], a RBM can be represented by a pair (v, h) where $v = (v_1, v_2, \ldots, v_m)^T$ and $h = (h_1, h_2, \ldots, h_n)^T$. The energy of the system can be calculated by

$$\rho(v,h) = \frac{1}{Z} e^{-E(v,h)} \quad (1)$$

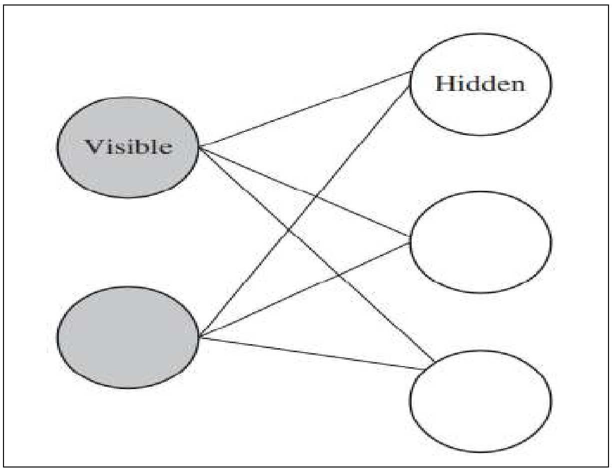


Fig 3: Restricted Boltzmann machine with visible and hidden units.

The probability of a network associated with every possible pair of a visible vector v and a hidden vector h is assumed to obey the Boltzmann distribution

$$\rho(v,h) = \frac{1}{Z} e^{-E(v,h)}, \quad (2)$$

where Z is a normalization constant, also called partition function, which is essentially the summation over all possible configurations. That is,

$$Z = \sum_{v,h} e^{-E(v,h)}. \quad (3)$$

The marginal probability of a network associated with v can be calculated by sum over all possible hidden vectors in Equ. (40), so we have

$$\rho(v) = \frac{1}{Z} \sum_h e^{-E(v,h)}, \quad (4)$$

The essential idea of using RBM for training over a set of data (such as images) is to adjust the weights and bias values so that a training image can maximize its associated network probability (thus minimizing its corresponding energy). For a training set, the maximization of the joint probability $\rho(v)$ is equivalent to the maximization of the expected log probability log $\rho(v)$. Since

$$\frac{\partial \log \rho(v)}{\partial w_{ij}} = < v_i h_j >_{data} - < v_i h_j >_{RRM\ model} \quad (5)$$

we can calculate the adjustments in weights by using the stochastic gradient method

$$\Delta w_{ij} = \eta\left(< v_i h_j >_{data} - < v_i h_j >_{RRM\ model}\right) \quad (6)$$

Where $< v_i h_j >$ means the expectation over the associated distributions. It is worth pointing out that the stochastic gradient ascent (in contrast to the SGD) is used. The individual activation probabilities for visible and hidden units are Sigmoid function $(x) = 1/(1 + e^{-x})$. That is,

$$\rho(v_i = 1|h) = S\left(\alpha_i + \sum_j w_{ij} h_j\right) \quad (7)$$

And

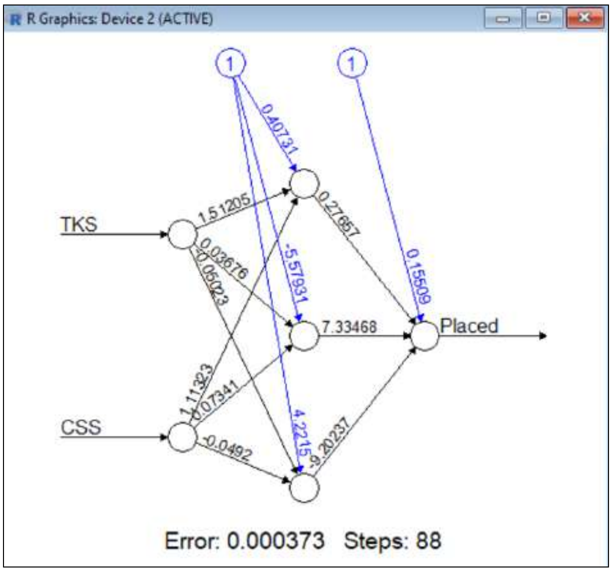
$$\rho(h_j = 1|v) = S\left(\beta_i + \sum_j w_{ij} v_j\right). \quad (8)$$

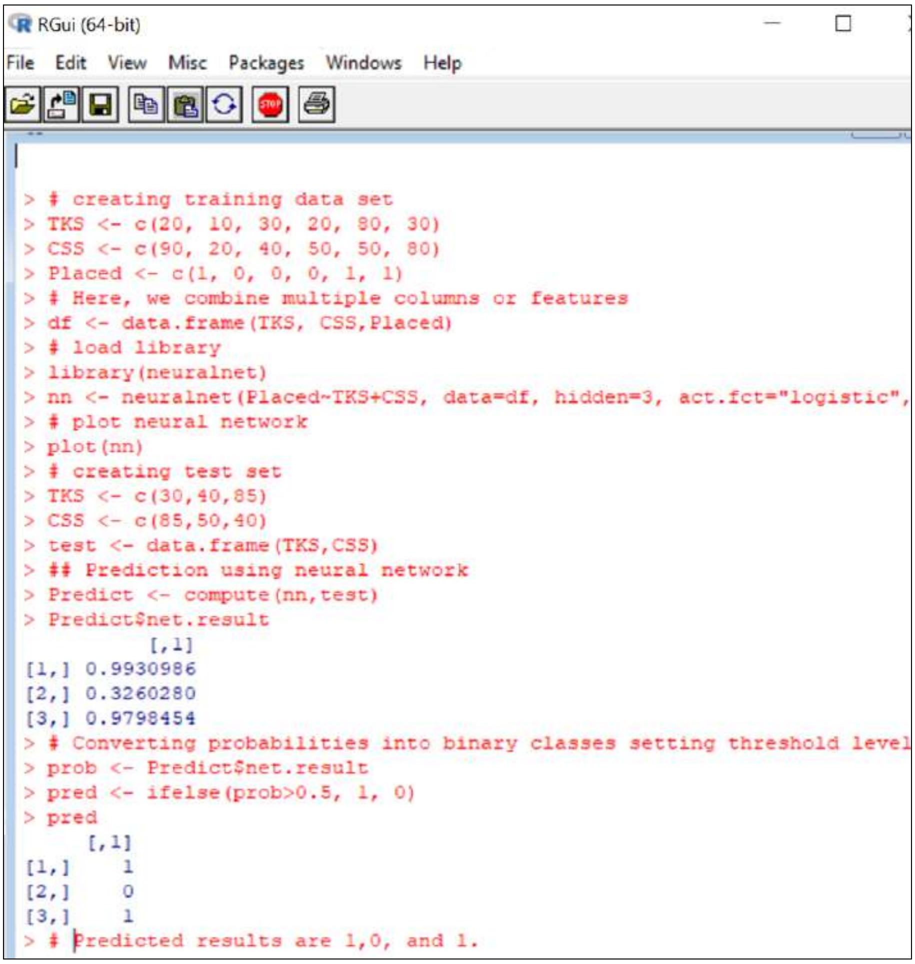
The RBMs form the essential part of deep belief networks with stacked RBM layers. There are many good software packages for ANNs, and there are dozens of good books fully dedicated to theory and implementations. We used r programming for the data analysis below.

Data Analysis and Result

Technical Knowledge Score	Communication skills score	Student place
20	90	Placed
10	20	Not Place
30	40	Not Place
20	50	Not Place
80	50	Place
30	80	Place

Sources: Data camp





RGui (64-bit)

File Edit View Misc Packages Windows Help

```
> # creating training data set
> TKS <- c(20, 10, 30, 20, 80, 30)
> CSS <- c(90, 20, 40, 50, 50, 80)
> Placed <- c(1, 0, 0, 0, 1, 1)
> # Here, we combine multiple columns or features
> df <- data.frame(TKS, CSS,Placed)
> # load library
> library(neuralnet)
> nn <- neuralnet(Placed~TKS+CSS, data=df, hidden=3, act.fct="logistic",
> # plot neural network
> plot(nn)
> # creating test set
> TKS <- c(30,40,85)
> CSS <- c(85,50,40)
> test <- data.frame(TKS,CSS)
> ## Prediction using neural network
> Predict <- compute(nn,test)
> Predict$net.result
          [,1]
[1,] 0.9930986
[2,] 0.3260280
[3,] 0.9798454
> # Converting probabilities into binary classes setting threshold level
> prob <- Predict$net.result
> pred <- ifelse(prob>0.5, 1, 0)
> pred
     [,1]
[1,]    1
[2,]    0
[3,]    1
> # Predicted results are 1,0, and 1.
```

Conclusion

The RBMs form the essential part of deep belief networks with stacked RBM layers. There are many good software packages for ANNs, and there are dozens of good books fully dedicated to theory and implementations. Therefore, we will not provide any code here. Neural Network Model shows 0.000373 errors with 88 steps. Prediction using neural network shows 0.9928202080, 0.3335543925 and 0.9775153014 while Converting probabilities into binary classes setting threshold level 0.5 result shows that the predicted results are 1, 0, and 1.

References

1. Changhau I. Loss Functions in Artificial Neural Networks. Github.io online mates; c2017. https://isaacchanghau.github.io/post/loss_functions/ (accessed 26 April 2018).
2. Fischer A, Igel C. Training restricted Boltzmann machines: an introduction. Pattern Recognition. 2014;47(1):25-39.
3. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504-507.
4. Bengio Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning. 2009;2(1):1-121.
5. Salakhutdinov RR, Mnih A, Hinton GE. Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th International Conference on Machine Learning (ICML); c2007. p. 791-798.
6. Hinton GE, Osindero S, Teh W. A fast learning algorithm for deep belief nets. Neural Computation. 2006;18(7):1527-1554.
7. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504-507.
8. Carreira-Perpi˜nán MA, Hinton GE. On contrastive divergence learning. In The 10th International Workshop on Artificial Intelligence and Statistics (AISTATS); c2005. p. 59-66.
9. Sam SO, Sewe S, Kimathi G, Wainaina M. Antecedents of patients Covid-19 management outcomes. International Journal of Statistics and Applied Mathematics. 2021;6:109-17.
10. Fischer A, Igel C. Empirical analysis of the divergence of Gibbs sampling based learning algorithms for restricted Boltzmann machines. In International Conference on Artificial Neural Networks (ICANN); c2010. p. 208-217.
11. Tieleman T, Hinton G. Using fast weights to improve persistent contrastive divergence. In Proceedings of the 26th Annual International Conference on Machine Learning; c2009. p. 1033-1040.
12. Desjardins G, Coutville A, Bengio Y, Vincent P, Dellaleau O. Parallel tempering for training of restricted Boltzmann machines. In Journal of Machine Learning Research Workshop and Conference Proceedings. 2010;9:145-152.
13. Cho K, Raiko T, Ilin A. Parallel tempering is efficient for learning restricted Boltzmann machines. In Proceedings of International Joint Conference on Neural Networks (IJCNN); c2010. p. 3246-3253.
14. Dahl GE, Tao S, Thompson IM. Lactation Biology Symposium: Effects of photoperiod on mammary gland

development and lactation. Journal of Animal Science. 2012 Mar 1;90(3):755-60.
15. Srivastava A, Brewer AK, Mauser-Bunschoten EP, Key NS, Kitchen S, Llinas A, Ludlam CA, Mahlangu JN, Mulder K, Poon MC, Street A. Guidelines for the management of hemophilia. Haemophilia. 2013 Jan;19(1):e1-47.
16. Salakhutdinov R, Hinton G. Semantic hashing. International Journal of Approximate Reasoning. 2009 Jul 1;50(7):969-78.
17. Schulz KF, Altman DG, Moher D. CONSORT 2010 statement: updated guidelines for reporting parallel group randomised trials. Journal of Pharmacology and pharmacotherapeutics. 2010 Dec;1(2):100-7.
18. Hinton AL. Introduction: Toward an anthropology of transitional justice. InTransitional Justice; c2010 May 26. p. 1-22. Rutgers University Press. https://www.datacamp.com/tutorial/neural-network-models-r
19. Bengio Y, Delalleau O. Justifying and generalizing contrastive divergence. Neural Computation. 2009;21(6):1601-1621.