SEMINAR REPORT

ON

BOLTZMANN MACHINE

A report submitted in partial fulfilment of the requirement for the award of

The degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted By:-

Ayush Singh Rawat

1501051120

2015-2019

Department of Information Technology

DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

2018-2019

ABSTRACT
We have information, however we don't have a clue what are the characteristic highlights of the information that would enable us to separate it from the rest. At first we begin constructing these highlights by discovering what these are and what might be the qualities comparing to these. When these inborn highlights are discovered, it is then used to figure how the first information would look like given these highlights. Calculated Regression calculation is being utilized here to demonstrate a showing how Boltzmann machine builds the expectation precision. Test results demonstrates a bounce of 89% to 93% precision in forecast and a better correlation for better feature extraction.

1. INTRODUCTION

A Boltzmann Machine is a system of symmetrically associated, neuron like units that settle on stochastic choices about whether to be on or off. Boltzmann machines have a basic learning calculation that enables them to find fascinating highlights with regards to datasets made out of twofold vectors. The learning calculation is exceptionally moderate in systems with numerous layers of highlight finders, however it very well may be made a lot quicker by learning one layer of highlight identifiers at once. Boltzmann machines are utilized to take care of two very unique computational issues. For a hunt issue, the loads on the associations are fixed and are utilized to speak to the cost capacity of an improvement issue. The stochastic elements of a Boltzmann machine at that point enable it to test double state vectors that speak to great answers for the enhancement issue. For a learning issue, the Boltzmann machine is appeared set of double information vectors and it must discover loads on the associations with the goal that the information vectors are great answers for the enhancement issue characterized by those loads.

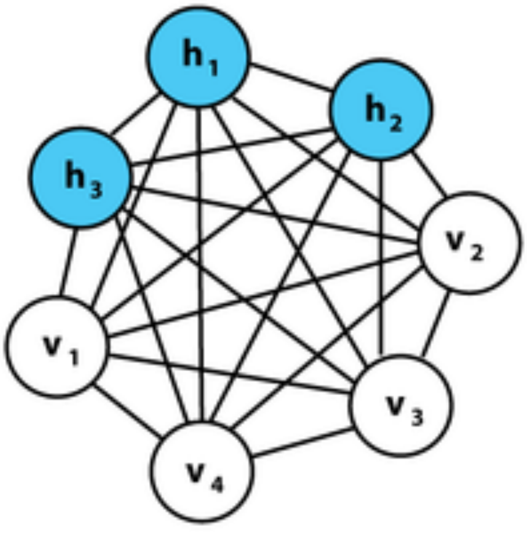

Figure 1(a): A Boltzmann machine

Collaborative Filtering is a method used by recommender systems to make predictions about an interest of an specific user by collecting taste or preferences information from many other users. The technique of Collaborative Filtering has the underlying assumption that if a user A has the same taste or opinion on an issue as the person B, A is more likely to have B's opinion on a different issue. Figure 2 shows how boltzmann machine implements collaborative filtering

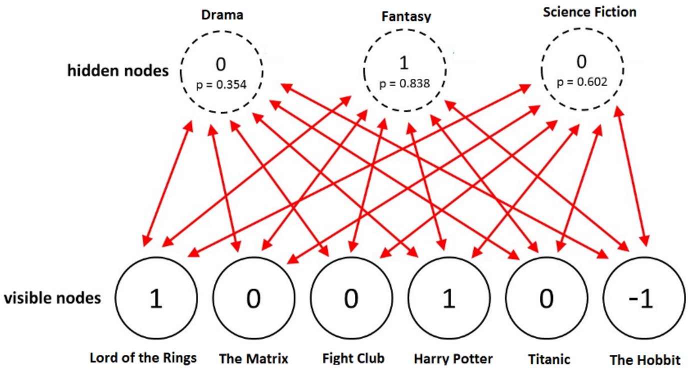

Figure 1(b): Boltzmann machine for collaborative filtering

2. IMPLEMENTATION

2.1 Training Algorithm:

The preparation of the Restricted Boltzmann Machine contrasts from the preparation of ordinary neural systems by means of stochastic slope drop.

The Two principle Training steps are:

Gibbs Sampling- The initial segment of the preparation is called Gibbs Sampling. Given an information vector v we use p(h|v) for expectation of the feature extraction h. Knowing the concealed qualities we use p(v|h) :

$$p(v_i = 1 \mid \mathrm{h}) = \frac{1}{1+e(-(\mathrm{a}_i + w_i h_i))} = \tilde{\mathrm{O}}\ (\mathrm{a}_i + \sum \mathrm{h}_i \mathrm{w}_{ii})$$

Equation 1: Feature extraction

for forecast of new info esteems v. This procedure is rehashed k times. After k cycles, we acquire another info vector v_k which was reproduced from unique information esteems v_0.

$$p(h_i = 1 \mid \mathrm{v}) = \frac{1}{1+e(-(\mathrm{b}_j + W_j v_i))} = \sigma\,(\mathrm{b}_j + \sum_i v_i w_{ij})$$

Equation 2: Sigmoid Neuron

Contrastive Divergence step- The update of the weight grid occurs amid the Contrastive Divergence step. Vectors v_0 and v_k are utilized to ascertain the enactment probabilities for shrouded esteems h_0 and h_k :

$$p(v_i = 1 \mid \mathrm{h}) = \frac{1}{1+e(-(\mathrm{a}_i + w_i h_i))} = \tilde{\mathrm{O}}\ (\mathrm{a}_i + \sum \mathrm{h}_i \mathrm{w}_{ii})$$

Equation 3: Initial Divergence

The contrast between the external results of those probabilities with information vectors v_0 and v_k results in the refreshed grid :

$$\Delta W = \mathrm{v_O}) \otimes P(\mathrm{h_O} \,|\, \mathrm{v_O}) - \mathrm{v_k} \otimes P(\mathrm{h_k} \,|\, \mathrm{v_k}) - \mathrm{v_k})$$

Equation 4: Updated Divergence

Utilizing the update lattice the new loads can be determined with angle rising, given by:

$$W_{new} = W_{old} + \triangle W$$

Equation 5: Updating Lattice

2.2 **Collaborative Filtering-** Where i represents movies, u is users, n the set of neighbours used n the algorithm, Expl score is between 0 and 1, if recommended then 1 and if not able to then 0, Explanation comes from the fact that they have included the neighbors of the user and rating for the movie. This value is also fed along with the movie ratings to predict the ratings of the movie for users. So, this model is known to outperform the previous methods in different cases.

$$Expl.\,Score(u,i) = \frac{\sum_{x \in N_k(u)} r_{x,i}}{\sum_{N_k(u)} R_{max}}$$

Equation 6: Expl. Score for predictive analysis

2.3 DESCRIPTION OF DATASET

The MNIST database contains 60,000 preparing pictures and 10,000 testing pictures taken from American Census Bureau representatives and American secondary school understudies. Figure 2 demonstrates the dataset and assessment is done on Spyder IDE with python 3.6.4.

Figure 2(a): MNIST dataset for digits

Table 1: Description of dataset for first 5 rows

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel77
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

DESCRIPTION OF DATASET FOR COLLABORATIVE FILTERING

The dataset is from MovieLens which consists of 1,00,000 ratings, on a scale of 1 to 5, for 1700 movies and 1000 users.

Ratings have been normalized between 0 and 1 for RBM input and the training and test data has been divided in 90:10 ratio.

3. RESULTS

3.1 Output without using Boltzmann machine

LOGISTIC REGRESSION ON ORIGINAL DATASET

	precision	recall	f1-score	support
0	0.94	0.96	0.95	196
1	0.94	0.97	0.95	245
2	0.89	0.90	0.90	197
3	0.88	0.84	0.86	202
4	0.90	0.93	0.91	193
5	0.85	0.75	0.80	183
6	0.91	0.93	0.92	194
7	0.90	0.90	0.90	212
8	0.85	0.83	0.84	186
9	0.81	0.84	0.83	192
avg / total	0.89	0.89	0.89	2000

Figure 3(a): Applying Logistic regression

3.2 Output with using Boltzmann machine

RBM + LOGISTIC REGRESSION ON ORIGINAL DATASET

	precision	recall	f1-score	support
0	0.95	0.98	0.97	196
1	0.97	0.96	0.97	245
2	0.92	0.95	0.94	197
3	0.93	0.91	0.92	202
4	0.92	0.95	0.94	193
5	0.95	0.86	0.90	183
6	0.95	0.95	0.95	194
7	0.93	0.91	0.92	212
8	0.91	0.90	0.91	186
9	0.86	0.90	0.88	192
avg / total	0.93	0.93	0.93	2000

Figure 3(b): Accuracy with RGB+ Logistic regression

3.3. Root Mean Squared Error in Collaborative featuring

The RGB model here outperforms other models for f<20 with testing root mean squared loss less than 1 and the results can be seen in figure 3(c).

```
epoch_nr: 0,  train_loss: 1.169, test_loss: 1.020
epoch_nr: 10, train_loss: 0.936, test_loss: 0.959
epoch_nr: 20, train_loss: 0.889, test_loss: 0.931
epoch_nr: 30, train_loss: 0.873, test_loss: 0.923
epoch_nr: 40, train_loss: 0.859, test_loss: 0.925
epoch_nr: 50, train_loss: 0.844, test_loss: 0.929
```

Figure 3(c): RMSE Training loss in RGB

From Figure 3(d) we can observed that RBM outperforms other approaches in terms of explainability. For a test user, the top-3 rated movies and their genres by the user, in addition to the top-3 recommended movies.

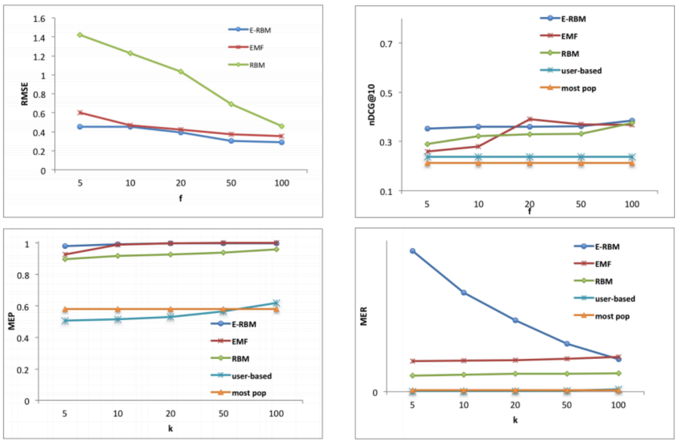

Figure 3(d): Other models compared to RGB in RMSE in collaborative filtering

4. FUTURE EXTENSION

An alternative way of using an RBM is to treat this learning as a pretraining stage and find better feature extracting and then using the squared error objective function, is then used to finetune the weights for optimal reconstruction of each user's ratings. However, overfitting becomes an issue and more careful model regularization is required.

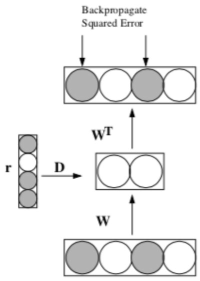

Figure 4(a):

5. CONCLUSION AND FUTURE SCOPE

Results demonstrate the feasibility of the neuromorphic related alterations performed for RBM execution. These outcomes additionally exhibit that Contrastive Divergence, which was initially created in the Machine Learning Gibbs testing system, gives tasteful outcomes for neural inspecting. Additionally, changes made for the weight refreshes in the neuromorphic structure to diminish state stockpiling don't appear to have critical effect on execution.
It has been appeared for Machine Learning that mistake rates for grouping errands can be additionally diminished by stacking RBMs and doing learning at each pair of layers independently. These machines are called Deep Belief Networks and, in view of the neural inspecting results acquired for RBM, it might be conceivable to utilize a comparative way to deal with actualize DBNs in a neuromorphic system. As found in Figure 4 we got a bounce of 4% with better component extraction.

REFERENCES

[1] Hinton, G.E., Sejnowski, T.J., and Ackley, D.H., "Boltzmann Machines:

Constraint Satisfaction Networks that Learn", Technical Report CMU-CS-84-119, May 1984.

[2] Mead, C., "Analog VLSI and Neural Systems". Reading, MA: Addison-Wesley, 1989.

[3] Aarts, E.H.L. and Korst, J.H.M., "Boltzmann Machines and Their Applications", PARLE Parallel Architectures and Languages, Europe, 1987.

[4] Ackley, D.H., Hinton, G.E., and Sejnowski, T.J., "A Learning Algorithm for Boltzmann Machines". Cognitive 9, pp. 147-169, 1985.

[5] Smolensky, P., "Information processing in dynamical systems: Foundations of harmony theory". Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. Cambridge, MA: MIT Press/Bradford Books, pp. 194-281, 1986.

[6] Hinton, G.E., "Training products of experts by minimizing contrastive divergence". Neural Computation, 14(8), pp. 1771-1800, August 2002.

[7] Buesing, L., Bill, J., Nessler, B., and Maass, W., "Neural Dynamics as Sampling: A Model for Stochastic Computation in Recurrent Networks of Spiking Neurons", PLOS Computational Biology, Volume 7, Issue 11, November 2011.

[8] Hinton, G.E., Osindero, S., and Teh, Y.W., "A fast learning algorithm for deep belief nets", Neural Computation, 18:1527-1554, 2006.

[9] Osindero, S. and Hinton, G.E., "Modeling image patches with a directed hierarchy of Markov random fields", NIPS 20, Cambridge, MA, 2008.

[10] Abdollahi, Behnoush and Nasraoui, Olfa. Explainable matrix factorization for collaborative filtering. In Proceedings of the 25th International Conference Companion on World Wide Web, pp. 5–6. International World Wide Web Conferences Steering Committee, 2016.