

Distributed deep Autoencoder for recommendation system

¹Ravi Kumar R.R.S, ²Apparao Giduturi, ³Anuradha Sesetti

¹Research Scholar, Dept. of CSE, GIT, GITAM (Deemed to be University), Vishakhapatnam, India.

²Professor, Dept. of CSE, GIT, GITAM (Deemed to be University), Vishakhapatnam, India.

³Assistant Professor, Dept. of CSE, GIT, GITAM (Deemed to be University), Vishakhapatnam, India.

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Recommender systems are one of the prominent area often the researchers are attracting to apply the concepts of Deep neural networks. Many of them introduced the problems facing by users like information load, Sparsity, Top-K recommendations, rating predictions and user feedback and giving best suitable solutions. Many machine learning and matrix factorization techniques are used to solve the problems but problems are linear modeling. Now in this paper, we propose distributed auto encoder with optimization to capture non-linear relationships between users and items to give predictions for missing values, giving top-k recommendations. We did several experiments on movielens 100k, movielens20M as compared with base line methods. Our proposed model performance outperforms the other existing models in terms of RMSE, MAE, HitRatio and Novelty evaluation metrics.

Keywords: Deep learning, Neural networks, Optimization, Recommendation systems, Stoplist.

Introduction: The rapid processing of vast information in World Wide Web, the users always trying to catch the required data from relevant information Ferreira et al.[1]. To address these issues the researchers applying machine and deep learning techniques to find required information for users among large data sets. Now a day, the required information to users is in the form of music, products, articles and news etc. For getting relevant items users are facing the problem of information load. Recommender systems are playing most interesting role to solve the issues of users and several organizations. It is a technique used to give required items to users based on several considerations like past history and behavior.

Recommender systems are used to give solution to information load problem by using the information search, filtering and retrieval methods to suggest required and behavior based items to users Ferreira et al. [1]. Later the companies using machine learning techniques like different types of collaborative and factorization model to decision making and giving personalized recommendations. Vito et al.[2]. Cremonesi et al.[13],Deshpande et al.[14],Ma et al.[15]

Deep Learning is the strengthening area for recommendation systems research. Vito et al.[2]. Several deep learning techniques are applied in collaborative filtering are Convolutional neural networks, Recurrent neural networks and deep neural networks and suggested to use distributed optimization techniques for minimizing computational cost. Jindal et al.[3]. More concentration done on the performance of the recommendation systems but F.Yuan et al[4] explain the issues of data contamination solved by Autoencoder. In our proposed work we are introduced the Distributed Autoencoder for recommendation systems. The contributions listed as follows.

1. We propose the distributed Autoencoder a capture non-linear data and to predict ratings . 2. We proposed distributed Autoencoder with optimization to give accurate top-k recommendations.

3. Evaluate the model using various evaluation metrics like RMSE, MAE, HitRatio and Novelty.

The organization of the paper as following: Section 2 is discussing about the related work. Section 3 explains the proposed system. Section 4 discusses the experimental results and Section 5 conclude the paper.

2. Related Work.

In this section we describe some of the base Machine Learning KSRK et al. [10] methods for our work.

Collaborative Filtering: This model successfully applied and suited for recommendation systems research and study. Many variants of collaborative filtering are developed are user based, item based and Hybrid systems. These techniques are using similarity metrics to do predictions and recommendations and consider user-item data. Dong et al. [7].

Matrix Factorization: It is widely used method doing prediction tasks using factors. It considers the implicit feedback as compared with collaborative filtering models. Several models like PCA, LDA and Probabilistic matrix factorization are developed. They are used to consider user-item matrix and find latent features. Meshal, Jianlin et al.[6]

Neural Networks: The flavors of artificial neural models are implemented in many works. Some of the works suitable for recommendation systems are collaborative filtering using RBM. It is not learning well up to consideration on user tastes and feedbacks. Meshal, Jianlin et al.[6].

Autoencoder: It is a deep neural network model contains three layers (Input layer, Hidden layer and output layer). Here the combination of input and hidden layer is encoder. The hidden and output layer combination is decoder. It can be used for recommendation systems. Ferreira et al. [1].

3. Proposed system.

In this we introduce the architecture of distributed deep Autoencoder for recommendation system. Sergeev et al.[8]. In addition we present process like Auto encoder data preprocessing, Build model, Test generator, Customized

loss for non linearities,Fit generator,Optimization and Top-K item recommendations. First we draw the Architecture of the Distributed deep auto encoder for recommendations in fig.1.We are training data by using standard data set our general models giving better results but we are facing a problem of data growth in future. By considering these issues we prepared distributed architecture for Autoencoder .In this architecture we have used the three autoencoders in parallel and the data is divided in to chunks and flow to the each encoder. All Autoencoder having the capabilities of data loading, masking and shuffling, build model, Train generator, Test generator,Custom loss, Fit generator. Next we have the Evaluation process. Chuduri et al[11], chinta et al[12] ,Ferreira et al. [1].

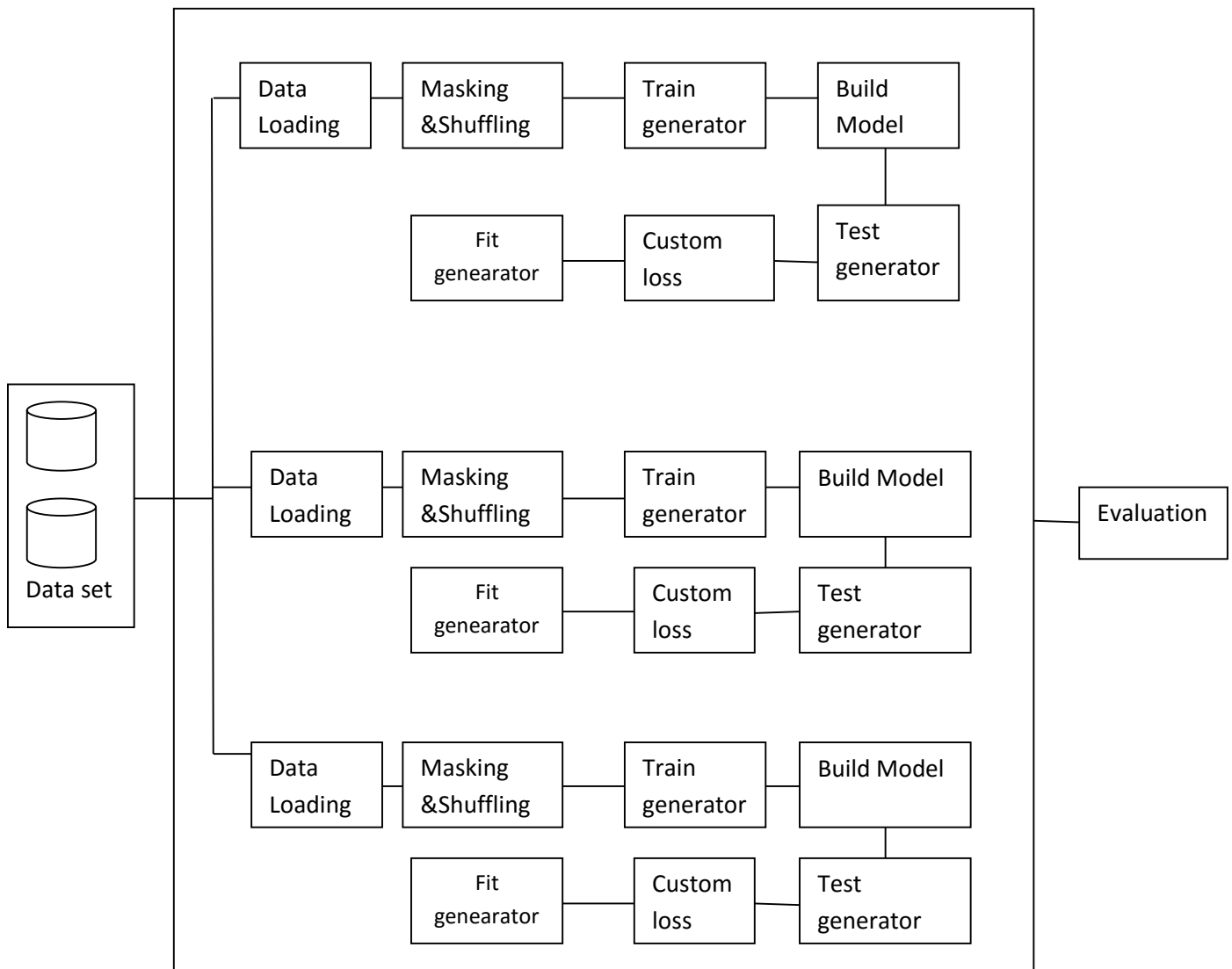


Fig 1: Architecture of Distributed Deep Autoencoder

Data set: The data we are using in our system are standard recommendation data sets of GroupLens are movielens 100k and movielens 20M.Maxwell et al.[9].

Data loading: We are loading our data set files like ratings.csv etc. Next we can divide our data into test data and train data.

Masking and Shuffling: The mask function to find which data values are missing and which are not missing following the rating conditions. Shuffling is for make copies of every matrix and gives to train generator.

Train generator: At working time it shuffle the data and same data order followed at test time. Each user represented as individual row in training and testing. The index of row contains user id it copied and passed during training. Here the data is divided into the number of batches and check the sum of data in all batches equal to length of the input data. Always check last batch is smaller than other batches. Set the upper limit to be minimum of plus one size of batch and length of actual input data. We put the condition for out of bound array. Next we take the present batch , mask values and rating values then follow the same process for all batches. Calculate the Average value for all ratings. Next find the cost and add noise to input data and forward to auto encoder to predict missing ratings..Finally returns the input and target data.

Build Model: Here we are preparing the dense layer for connecting the remaining layers of the autoencoder. It is used to connect input layer to hidden layer and hidden layer to output layer. The dropout rate after the input to input layer informs us to missing ratings. Here we reconstructing and predicting missing ratings. We can use L2 regulation as drop out regulator and we use some hidden activation functions. Next we find the sum of all differences to the sum of squares. Finally we calculate MAE.

Test generator: It works with both test and train data. Here we recall the training data and find out which on needs predictions. We use testing to predict unseen ratings so we are not using any shuffling activity here. Batch wise we process the data and find missing and subtract from input. Here we are using different various optimizers for improving results. Next we report the regularized loss by adding RMSE to L2 penalty. The current MSE is always larger than previous RMSE.

Custom Loss: Here we introduced the loss function using mask. Here we find out any square difference from actual rating.

Fit Generator : It will work with both training and testing generators for training and validation data. Based on that it check any loss and calculate MSE.

Evaluation: Finally we evaluate our model using MSE, SSE, HitRatio and Novelty metrics.

In next, we present the Algorithm for distributed auto encoder.

Algorithm1: Distributed Autoencoder.

Input: Set of users $U=\{1,2,\dots,u\}$, set of items $I=\{1,2,\dots,i\}$

Output: Prediction of ratings, Top-K recommendations

1. Store the data and split the data
2. For each AutoEncoder
 - 2.1. Load data
 - 2.2. Perform mask operation for finding empty values.
 - 2.3. Train the data using train generator.
 - 2.3.1. Perform shuffle operation.
 - 2.3.2. Check the equality of sum of batch size and length of input data
 - 2.3.3. Add noise
 - 2.3.4. Return inputs and Targets.
 - 2.4. Build Model
 - 2.4.1. Prepare dense layer.
 - 2.4.2. Check dropout rate and L2 regulator
 - 2.4.3. Calculate MAE
 - 2.5. Train data using training generator.
 - 2.5.1. Optimization
 - 2.5.2. Calculate regularization loss.
 - 2.5.3. Calculate RMSE with L2 Penalty.
 - 2.6. Find Custom loss
 - 2.7. Fit Model using train data and test data
 - 2.8. Prediction of ratings.
 - 2.9. Generate top-K recommendation.

4. Experimental Results: The following table presents the results of evaluation metrics of Distributed auto encoder.

Methods	MovieLens 100K				MovieLens 20M			
	RMSE	Novelty	MAE	HR	RMSE	Novelty	MAE	HR
Auto Encoder	1.8253	512.45	1.1222	0.0075	1.8285	514.45	1.5411	0.0080
Distributed Autoencoder	1.4385	557.83	0.9235	0.0110	1.4385	586.83	1.0442	0.0120

As per the results mentioned in the above table the distributed autoencoder showing better results than Autoencoder following the metrics are RMSE, Novelty, MAE and Hit Ratio. The distributed autoencoder shows lower RMSE, MAE and Higher Novelty, HitRatio as compare with Autoencoder. The following plots represents our experimental results graphically.

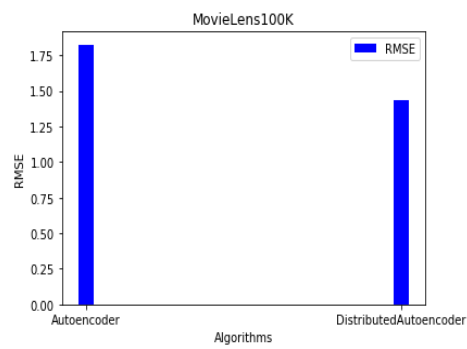


Fig.2: Comparison of RMSE on MovieLens100K

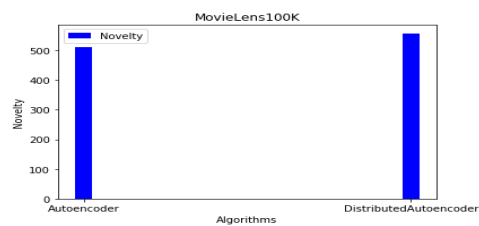


Fig3. Comparison of Novelty on MovieLens 100K



Fig.4. Comparison of MAE on MovieLens 100K



Fig.5. Comparison of Hit Ratio on MovieLens 100K



Fig.6. Comparison of RMSE on MovieLens 20M

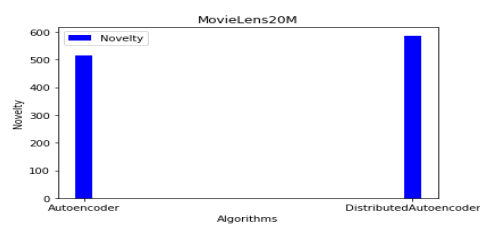


Fig.7. Comparison of Novelty on MovieLens 20M



Fig.8. Comparison of MAE on MovieLens 20M

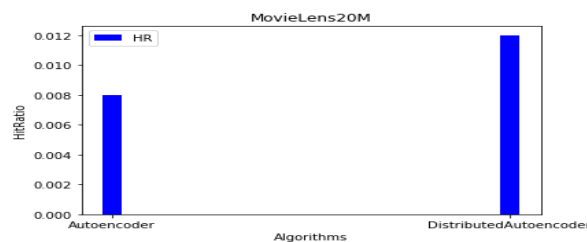


Fig.9. Comparison of Hit Ratio on MovieLens 20M

5. Conclusion and Futurework: The aim of good recommendation system is to give accurate recommendations. Several machine learning and deep neural learning methods are used in this scenario. In this paper, we developed and evaluated a recommendation system based on the deep learning and called it as Distributed autoencoder. The experimental results shows our system is better as compared with Autoencoder. It works effectively on large scale and distributed data sets and giving accurate recommendations. In future, we develop recommendation models on data bridges using cognitive methods.

References:

- [1] Diana Ferreira, Sofia Silva, Antonio Abelha and Jose Machado “Recommendation System Using Autoencoders” MDPI, 2020, pp.1–17.
- [2] Vito Bellini , Tommaso Di Noia , Eugenio Di Sciascio ,And Angelo Schiavone, Semantics-Aware Autoencoder, IEEE Access ,2019, 166122–166137.
- [3] Rajni Jindal, Kirti Jain, A Review On Recommendation Systems Using Deep Learning, IJSTR, 2019, 2978-2985.
- [4] Feng Yuan, Lina Yao, and Boualem Benatallah, Adversarial Collaborative Auto-encoder for Top-N Recommendation, IEEE, 2019.
- [5] Hyeungill Lee, Jungwoo Lee, Scalable deep learning-based recommendation systems, ICT Express, 2019, 84–88.
- [6] Meshal Alfarhood And Jianlin Cheng, CATA++: A Collaborative Dual Attentive Autoencoder Method for Recommending. Scientific Articles, IEEE Access, Vol.8, 2020, pp. 183633- 183648.
- [7] Bingbing Dong, Yi Zhu, Lei Li, And Xindong Wu, Hybrid Collaborative Recommendation via Dual-Autoencoder, IEEE Access, 2020, pp. 46030- 46040.
- [8] Alexander Sergeev, Mike Del Balso, Horovod: fast and easy distributed deep learning in TensorFlow, Cornell University, 2018, pp. 1- 10.
- [9] F. Maxwell Harper and Joseph A. Konstan. 2015. “The MovieLens Datasets: History and Context”. ACM Transactions on Interactive Intelligent Systems (TiiS) , December 2015.
- [10] K.S.R.K.Sarma and M.Ussenaiah, “ Texture Classification using Advanced Texton Texture Matrix” IJAST Vol. 29, No. 3s, (2020), pp. 729 – 744.
- [11] Raghavendra K Chunduri, Aswani Kumar Cherukuri.” Scalable Formal Concept Analysis algorithm for large datasets using Spark”, JAIHC, 2018.
- [12] Chinta Venkata Murali Krishna, Dr .G. Appa Rao.” Acquiring the user’s opinion by using a generalized Context-aware Recommender System for real-world applications”. IJET, 2018.
- [13] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: Proceedings of the Fourth ACM Conference on Recommender Systems, Association for Computing Machinery, New York, NY, USA, 2010, pp. 39–46,
- [14] M. Deshpande, G. Karypis, Item-based top-*n* recommendation algorithms, ACM Trans. Inf. Syst. 22 (2004) 143–177.
- [15] C. Ma, P. Kang, X. Liu, Hierarchical gating networks for sequential recommendation. arXiv preprint arXiv:1906.09217, 2019.