Future Computing and Informatics Journal

Volume 5 | Issue 2 (2020) Article 3

2020

Multilayer Perceptron with Auto encoder enabled Deep Learning model for Recommender Systems

subhashini narayan
VIT University, subanarayan@gmail.com

Future Computing and Informatics Journal

Volume 5 | Issue 2 (2020) Article 3

2020

Multilayer Perceptron with Auto encoder enabled Deep Learning model for Recommender Systems

subhashini narayan
VIT University, subanarayan@gmail.com

Future Computing and Informatics Journal

Homepage: https://digitalcommons.aaru.edu.jo/fcij/

doi: http://Doi.org/10.54623/fue.fcij.5.2.3

Multilayer Perceptron with Auto encoder enabled Deep Learning model for Recommender Systems

subhashini narayan

VIT University, vellore, Tamil Nadu, India

subanarayan@gmail.com

Abstract

In this modern world of ever-increasing one-click purchases, movie bookings, music, health-care, fashion, the need for recommendations have increased the more. Google, Netflix, Spotify, Amazon and other tech giants use recommendations to customize and tailor their search engines to suit the user's interests. Many of the existing systems are based on older algorithms which although have decent accuracies, require large training and testing datasets and with the emergence of deep learning, the accuracy of algorithms has further improved, and error rates have reduced due to the use of multiple layers. The need for large datasets has declined as well. This research article propose a recommendation system based on deep learning models such as multilayer perceptron that would provide a slightly more efficient and accurate recommendations.

Keywords: Recommendation system, deep learning, multilayer perceptron.

1. Introduction

To study, understand, analyse and improve the accuracy of recommendations by using deep learning algorithms rather than the use of any common machine learning algorithm and towards the end of the research proposal, be able to have one algorithm that stands out from the rest in terms of its accuracy. One of the first objectives is to research about the currently used algorithms in earlier researches and find out what flaws can be overcome with the use of deep learning, over the traditional machine learning algorithms. The background research of about thirty papers mainly gives us an idea of where the need to start and gives us a better understanding about algorithms and recommender systems as whole since this is a relatively upcoming field in information technology.

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

Over the past twenty years, rise in the use of recommendations in various fields of ecommerce, travel, movies, music, finance and so on. Take an example of movies, instead of sitting and looking at tons of movies which are there online, it would make it much easier for you if an engine were to look at your past history of watching movies, your earlier selections and suggest you movies based on your ratings and ratings of others. This way it would select learn about top rated movies that you have rated earlier and then look at similar people who have interests similar to yours and suggest you some of the movies they have rated as top movies. But if you look at the use of algorithms, most of the sites use older machine learning algorithms which have certain flaws that can be improved upon. Thus, over the past five years, the emerging field of deep learning comes into view. Deep learning is an improvement over machine learning algorithms in the sense that it learns intricates details between the data. It iterates over the data many more times trying to find better relationships that can be used to provide better Recommendations. Since it is still a relatively emerging field, this research proposal aims to work on using deep learning techniques.

Now among the existing recommendation systems, the more commonly used algorithm is the collaborative filtering algorithm which does recommendations based on prior data of your ratings as well as others. But recent newer algorithms which use deep learning, such as a multilayer perceptron or an autoencoder, which when applied to collaborative filtering, should be able to learn better and provide more accurate recommendations.

Also, the datasets undergo the following procedures such as clean the dataset, delete duplicates, and feature select the ones we need. Data cannot be directly used since most datasets are just crawled off the net and may have a lot of null values.

Having null values will negatively impact results since the relation between the attributes would be more difficult to learn and hence we need to clean and pre-process the data according to the need. First, this proposal aims to analyse the accuracy of collaborative filtering with a supervised machine learning algorithm. Collaborative filtering filters the data depending on your ratings, and then we test the algorithm on multiple datasets, evaluate and see its accuracy and see where improvement is needed. This would use a very basic algorithm is a very just to work on it and to compare other algorithms against it. Next, prefers to use a supervised deep learning model for collaborative filtering, finally test it on multiple datasets, and evaluate to test its accuracy against the previously mentioned machine learning algorithm.

Overall objective of this proposal is to implement collaborative filtering with an unsupervised deep learning algorithm where the recommendations given do not depend on any test data, we once again apply it to a dataset and we try to analyse the accuracy of this algorithm. The main aim here is to see whether this will give us a better result than the previous algorithms and hence the use of a third algorithm. This algorithm will be a more upcoming algorithm which is said to have very promising results. And last but not the leastm we will be evaluating these algorithms with various common evaluation metrics and then comparing these algorithms with the help of these metrics and providing a detailed summary and visualized result with the help of bar graphs and thus see its feasibility and efficiency and summarize the findings at the end of the research proposal.

1.1. Motivation

Machine learning is a subset of artificial intelligence that is widely used in most recommendation systems across the globe such as Netflix, Amazon and other platforms. Whereas deep learning, which is a subset of machine learning itself.

Published by Arab Journals Platform, 2020

The main difference between machine learning and deep learning is that deep learning involves multiple convolutional layers which basically help understand the intrinsic behaviour of the known variables rather than find just find a relationship between the seen differences. This is done so as to improve the accuracy of predictions given by the system. We intend to work on a machine learning algorithm and a couple of deep learning algorithms and finally compare the outcomes to each other.

Recommendation systems on the other hand are being used more widely these days to help analyse user behaviour and make predictions based on previous choices whether it is in the field of ecommerce, movie booking, music applications or even tourism. As the need increases, the need for constant improvement arises. Humans thrive on evolution and thus the same way, algorithms that predict these recommendations need to be constantly improved as well. There arose the motivation to research on better algorithms than the ones currently used by Netflix, Spotify, Amazon, Apple Music or Lenskart. These ecommerce, music and movie websites use it more frequently than most other sites. Currently the commonly used algorithm is collaborative filtering using various machine learning algorithms which can further be improved upon.

In 2009, Netflix had opened up a challenge called Netflix prize which paved way for many developers to come up with many different algorithms to help improve their recommendations. Netflix had offered to pay the best algorithm, $1M and a developer did manage to get the prize money but Netflix never used the algorithm because the engineering costs even though the algorithm could increase their recommendation accuracy by 10%. Thus the other problem to be kept in mind is to cut down on engineering costs while building an algorithm.

Hence we were curious to try it on the own with the help of existing algorithms and making slight modifications to see whether it would give us the kind of results we were hoping for. If it works well, then it could be later for a better time to develop further on these algorithms and make it more efficient.

Using a deep learning model along with collaborative filtering for a recommender system, should significantly improve the accuracy but that is open to research, which is why we have chosen to implement and test three algorithms each hopefully proving significantly better the previous ones. Although there have been researches on deep learning, we intend to research on less used deep learning algorithms and hope to prove that a deep learning algorithm, with multiple convolutional nets would increase the efficiency of such systems.

One of the other things we need to note is that most recommendation engines are not generalizable; they are usually specific to a certain dataset or a couple. We intend to analyse algorithms to see if they would hopefully work on multiple datasets, given that we are able to find datasets in the format we require. Whether the collaborative filtering models are generalizable, that can only be seen towards the end of this research proposal. If not, we will have to try it on many other datasets and modify them one at a time and then build on a more generalizable model if time permits.

The motivation here is to understand the currently existing algorithms from research and see where the fault lies. Especially since we will be focusing on collaborative filtering with machine learning and deep learning, we will look mainly at those papers that implement them, see their salient features and see whether the algorithms we have implemented can outperform them.

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

1.2. Background

Chen H. C. And A. L. P. Chen [1] have presented a music recommender with user profiling and music group based on content based filtering and also collaborative filtering approaches. Content based filtering systems usually recommend items that are quite similar to previous items to users, which causes over personalization recommendations. Collaborative filtering has problems such as cold start, low scalability and sparsity.

Krstic M. and M. Bjelica [2] have proposed a TV program recommender where PCA is used for feature extraction. The ELM neural network algorithm and single layer network with one hidden node are used to classify the programs based on watching habits with limited dataset and had an improved accuracy of about 92.5% as compared to other algorithms. But this is designed just for a TV program recommendation and also, contextual post-filtering did not perform as well as the system did on contextual pre-filtering.

Dong A. H. et al. [3] have discussed an apparel recommendation system that uses the data entered by the user to make rules and these rules help identify the products that a user might be interested in buying. It uses a blackboard structure and a task tree to store the apparel color, skin color and such attributes are used to identify clothes that the user might like. This recommendation system is a rule-based identification of which apparel comes closest to the data entered by the user. This does not utilize any deep learning as such and is more of association rule data mining and thus requires a lot of pre-trained data.

Shoja B. M. and N. Tabrizi [4] have discussed an algorithm to analyse customer reviews for e-commerce and then generate recommendations. First, LDA is applied to get a dictionary of attributes. Multiplayers are used as encoder and decoder to reconstruct input data to output without loss of dimensions.

This is to remove problem of sparsity and then collaborative filtering is used to get rankings. For processing they have used the NLP package nltk as well. Here deep learning has only been used to encode and has not been used as the predictive model.

Mao Q. et al. [5] have developed a recommender for costumes based on expert systems. Here, a knowledge base is made and then production rules are developed from it. Index adding algorithm and blackboard algorithm are used in this paper. There is no use of machine learning here, only data mining with the help of a knowledge base and rules.

Yuan N. J. et al. [6] have suggested a recommender system for drivers of teaxis and people who want to take a taxi. Here again, GPS trajectories have been used to check locations and other factors such as weather and past drop-off behaviors. This requires extensive data and the IVMM algorithm is old. Collaborative filtering is used here and as mentioned earlier, it has a cold start and sparsity problems which has been ignored here.

Chen P. et al. [7] have provided a cab and carpool recommendation system to reduce mileage and wastage of fuel. Here they have used trajectory using the requests of users and users in the car to calculate the dispersion distance. The trajectory also finds areas where there have been a lot of requests and clustering is used to select the optimum passengers. Here the GPS coordinates lead to errors of data and distance calculation. And this does not use any deep learning techniques but rather just uses big data and clustering.

Ravi L. and S. Vairavasundaram [8] have proposed a location recommender based on social pertinent trust walker (SPTW) algorithm using location based social network (LBSN).

The uniqueness of this algorithm is that it provides a recommendation for a group of users by checking the POI of all the users in the group.

Published by Arab Journals Platform, 2020

A score is given from the similarity matrix and then ranked and displayed to the users. The drawback here is that it only checks their interests and does not check the feasibility of traveling to the location for all the users and weather conditions etc.

Hao J. et al. [9] have suggested a probability based hybrid user model that uses collaborative filtering recommendation system. The dataset that has been taken is Movie Lens 100k. Their results give a lower mean absolute error and higher coverage rate prediction than dgim. But this hasn't been tested on other datasets and also, the ability of fault tolerance needs to be promoted when data density is high.

Wu D. and J. Lu [10] have proposed a fuzzy tree based recommendation system where the fuzzy tree groups similarities and divides them into sub categories. The learning activity is used to calculate matching similarity to learner's requirement and predict rating. This has been done in the e-learning domain with a self-made dataset and also, the accuracy seems to be quite low.

Aher S. B. and L. M. R. J. Lobo [11] have combined K-means clustering and Apriori algorithm for course recommendation by first clustering the courses into various clusters and then applying apriori algorithm on each cluster. It has a good overall accuracy in recommending courses but other combinations of machine learning algorithms can be used to gain better accuracy.

Xing S. et al. [12] have published their research on content aware PoI system where CNN captures the user's sentiment, preference and PoI property vectors which are in the review content and use it to evaluate user's PoI rating. This is tested on location based social network dataset. Their model shows a better recommendation because of a higher accuracy and better recall rate. The training though, takes a lot of time being a loosely coupled model and also does not predict user's preference transition over PoIs but rather only uses geographical data. Also, this domain is not within the scope of study.Liu S. and X. Meng [13] have provided a recommendation algorithm for location-based information by constructing a region-based location graph which also takes care of the cold start problem. They use the short-distance and long-range movements to make business information recommendations. The algorithm they have used is a modification of the PageRank algorithm. It provides a higher hit ratio for cold users.

Bahramian Z. et al. [14] have discussed a system for tour planning which uses case based reasoning as to filter the knowledge base. SBR stores the past data and its ratings. Then a multilayer perceptron is used to compute the ratings of all possible tours. The input to the ANN is the history of visited tours and all the related ratings. Process is iterated till a good tour is recommended to the user. It uses feedback so overcomes the cold start problem. Here again, a tour is not just about the past ratings, some users also look at the budget, weather, mood and how crowded it is. So this can be improved upon.

Wang Y. et al. [15] have proposed a sentiment based movie recommender system on Spark combining both collaborative filtering and content-based methods and have suggested a hybrid methodology. This provides increased efficiency and high timeliness compared to both the methods alone. Euclidean distance is used to measure the similarity between users by using the ratings given by 2 users. Then the Chinese segmentation is used to segment the words, then analyzed for sentiments using Spark and then rated and recommended to the user. This technique has been proposed only in movie recommendations and needs to be checked with other aspects and domains.

Yang D. et al. [16] have discussed a cnn-based system which is also time-aware because many times, while recommending, the time and location are neglected. Their neural network algorithm takes time context and history into consideration.

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

The mean squared error of the algorithm is less than a normal CNN and hence is a good algorithm. But requires a large dataset if the accuracy needs to be improved further and to have a highly stable algorithm.

Also, time divisions as weekdays and weekends are not greatly contributing factors for a movie selection.

Shao Y. and C. Wang [17] have presented a semantic model with HIBoosting which has weighted links that show the association between expressions. This boosts the gradient of features to gain more information by associations. This paper has addressed cold start and sparsity and has a higher accuracy compared to KNN, SVD, SVD++. This algorithm has been verified only on movie recommendations.

Guo F. and Q. Lu [18] have studied the context of information and have applied the distribution cognition theory to extract user's preferences in movie rating. It provides a better result than tcf and ccf. Bur here the dependencies haven't been taken into account. Also, this does not have any machine learning implementation.

Hao T. et al. [19] has proposed a recommendation system to recommend photo poses where CNN is used to extract the features of the photos, salient region detection and histogram are used for storing the local data. It calculates similarity between user's photo and professional photos. Their precision is 91.2, but this does not demonstrate the use of CNN for learning but rather only for feature extraction.

Deng F. et al. [20] have suggested a movie recommender based mainly on the visual appeal of its poster. Here they have used the editorial features, user-generated features and the visual features. Their baseline algorithm is KNN and by importing these features into it, calling it hybrid feature-based KNN. The algorithm works less accurately on small datasets and also, posters of all movies are not always available online.

Shrivastava K. et al. [21] have developed a model to extract emotions from text where a CNN model is used to identify emotions in text sequentially based on the previous sentence.

It uses two 2D Convolutional layers, two pooling layers, one fully connected layer and an attention mechanism layer to extract the features and classify the emotions into 6 classes. Their accuracy is about 80% which is significantly higher than RF and LSTM. But the dataset is imbalanced and hence training model biases towards the class which has a large number of instances. Minor classes seem to be ignored and the annotators aren't always in agreement with each other. This has to be improved or better annotators need to be used. Wang Z. et al. [22] have presented a news topics recommendation system using the TF-IDF, RAKE and keyword scoring algorithms. The keywords are scored according to frequency of appearance over time. This is based on a keyword extraction and it does not use any machine learning algorithm as such. This paper could have used Natural Language Processing which would have made it more efficient and accurate in providing better recommendations. Also, this does not check the context or sentiments of the words in the topics, for example, the word 'Java' could both mean a place or a computer language and proper nouns are not taken into consideration as well.

Li Z. et al. [23] have used a multi-view GCN for predicting links in a graph with matrix completion. It uses the Adam algorithm for learning. It has been tried on 6 datasets and seems to have a higher accuracy than the traditional algorithms used for link prediction such as TOP, SVD++, MC etc. However link prediction is not a domain we are looking at.

Tan Q. and Liu F. [24] have proposed an e-commerce, movie and music recommendation system which uses an attention-based behaviour-aware neural network. The attention mechanism of the algorithm mimics the attention of the human brain. This algorithm com- bines self-attention and the Bi-gated recurrent unit. It does embedding of the items, then learns the short-term choices of the user with self-attention and later learns the long-term choices using Bi-GRU.

Published by Arab Journals Platform, 2020

It has a high accuracy in all the domains they have tested. In e- commerce, sometimes the price changes also attract the customers and this is not taken into consideration in this model.

Ramzan B. et al. [25] have built a system for hotel recommendations with collaborative filtering and sentiment analysis based on people's opinions. It utilizes a NoSQL Cassandra database in the Hadoop framework. Feature extraction is done with NLP using the NLTK package in python and then polarity detection is done to assign values to it based on whether it's a positive, neutral or negative review. Finally it classifies and recommends it to the user. It shows good results and also reduced time taken for recommendations compared to conventional approaches.

Manogaran G. et al. [26] have proposed a Multiple Kernel Learning with Adaptive Neuro-Fuzzy Inference System, a deep learning method for heart disease diagnosis. MKL is utilized to distinguish the parameters between people who have heart disease and those who do not. Fuzzy membership values are used in analyzing BP and cholesterol. This algorithm has provided a higher specificity compared to other algorithms such as LS with SVM, PCA with ANFIS etc. Here only two features have been used which is the drawback. More features can be introduced to make the system a more reliable one.

Deng X. and F. Huangfu [27] have discussed a collaborative variational deep learning method for healthcare recommendation and stochastic gradient variational bayes. The datasets they have tested them out are on CiteULike-a, CiteULike-t, GHC. It recommends doctor as the end result of the testing. One place of improvement would be to incorporate knowledge graph to obtain more side information of the users to make better recommendations. Su J. H. et al. [28] have presented an algorithm of their own called uMender which is a combination of both content based recommendation system and context based. They have done their testing on the Amazon reviews dataset. Their final results show a precision of 0.62, a recall of 0.6 and F-measure of 0.61. To improve they could adopt perceptual patterns to enhance content-based retrieval.

Chang J. H. et al. [29] have implemented a hotel recommender system which uses social media date to improve recommendations. Here they have used LDA for feature selection and applied the Supplementary Information Transformation (SITA) algorithm on Yelp and Twitter datasets. Their precision at the end 0.97 and MRR was 0.15 at sensitive 20. The data they have used is homogeneous in nature and hence their performance can be enhanced by applying more heterogeneous social media data.

Fan J. et al. [30] have suggested a personalized image recommender on three different levels such as topic-based recommendation, context-based recommendation and intention- based recommendation. They test it on the Flickr images and have got a precision of 85.8% and a recall value of 86.3%. But since this paper only deals with images from Flickr, it would be good to try this algorithm out on different and varied datasets and see if they get the same precision on those as well.

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

1.1. Table for literature

Domain used	Number of papers	Application on datasets	Algorithms used
Music and TV Channels	3	Personal datasets – MIDI files, Channel and ratings	Collaborative filtering with K-means, Context-based neural net, PCA, ELM, Attention-BNN
E-commerce	4	Personal clothes dataset, Amazon reviews	Collaborative filtering with rule-based engine, NLP, LDA, uMender
Travel	3	Beijing road network data, taxicabs data, Foursquare LBSN data	Collaborative filtering with IVMM, Clustering, SPTW
E-learning	2	Personal datasets of courses and subjects	Fuzzy, K-means, Apriori
Location Recommender	3	Foursquare LBSN data, YELP dataset, Tehran tour dataset	CNN, PageRank, Context based system, case-based reasoning
Movies	7	MovieLens-100k, Douban dataset, MovieLens-1m, CiaoDVD, LastFM, Moviepilot-mp.mood, hetrec2011-movielens-2k, CiteSeer, Flixter, Jester, Cora	Collaborative filtering, Collaborative filtering with content based, Time-aware CNN, HIBoosting, Distributive cognition theory, HFB-KNN, MV-GN Graph
Image Recommender	2	Personal dataset from Flickr, Weibo and Foursquare	CNN, topic-based, context-based, intention-based
Text & News Article Recommender	2	Charmed TV show dialogues, Personal newspaper headline dataset	CNN, RAKE and NLP
Hotel Recommender	2	Trip Advisor, Expedia, YELP, Twitter dataset	NLP, Collaborative Filtering with sentiment analysis, LDA, SITA
Healthcare	2	Heart disease dataset, CiteULike-a, CiteULike-t, GHC	Multiple Kernel Learning with ANFIS, Collaborative Variational Deep Learning, Stochastic Gradient Variational Bayes

Published by Arab Journals Platform, 2020

2. Proposed methodology

The research paper aims to build a deep learning-based recommender system where we will be using three algorithms, one machine learning algorithm, a single layer perceptron which uses matrix factorization with a single neural net and two deep learning algorithms, multilayer perceptron which uses matrix factorization and multiple neural nets, and an autoencoder algorithm which is unsupervised compared to the other two algorithms. Thus the separate goals of the research proposal would be researching and doing a literature survey, selecting the datasets, exploring the data to see if it suits the needs, cleaning the data, selecting the required features and then we move on to the application and implementation phase where we choose a few algorithms that we want to work on, modify them as per the use of datasets and to get better results. By the end of the research proposal we hope to see deep learning model working efficiently that a machine learning algorithm and giving good results.

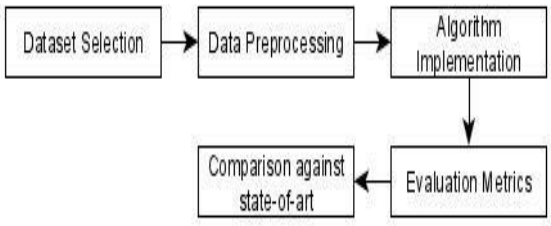

Figure 1:2.1 Overall architecture

We have decided to test the algorithms on multiple datasets, namely the MovieLens-100k dataset [31] which has been used very widely in many machine learning based papers and the Amazon Grocery and Gourmet Food dataset which has been used just twice across two papers by researchers in the past since it is a very recent dataset. Although it remains to be seen if these algorithms can be applied on any and every kind of dataset. This is further explained in the paper and the limitations we will face with certain datasets. Another thing we need to note here is that some data will have null values and missing values and sometimes a few duplicates as well, which is why we will be doing the data cleaning phase. Also feature selection is helpful since it is at times some attributes do not really have much to do with the result of the unknown variable and hence it is okay to remove them completely. The first step after selecting data, is data pre-processing which involves cleaning and feature selection, we will be doing with Weka Tool.

Within that the first step is the removal of duplicates and the filter we will use for it is called Remove duplicates itself, which scans through the data, finds duplicate rows and remove them. The next step is feature selection where we will use two methods namely the Cfc subset evaluator along with the Best first search and Correlation attribute evaluator with the Ranker search method. The Cfc subset evaluator basically evaluates the importance of a certain subset of attributes by looking at them individually and calculating the predictive ability of each attribute or feature as well as the degree of redundancy in between these attributes. Subsets of attributes that are very highly correlated to the class attribute but also having low intercorrelation are generally chosen. The search algorithm for this, uses breadth first for searching. [32].

Correlation attribute evaluator uses the Pearson correlation coefficient which evaluates the closeness of linear relation between two attributes. It checks the bond of linear associa- tivity. The Pearson correlation coefficient is given as follows: [33]

$$\rho(i,j) = \frac{1}{m-1}\sum_{k=1}^{m}\left(\frac{M_{ik} - \text{avg}(i)}{\text{std}(i)}\right)\left(\frac{M_{jk} - \text{avg}(j)}{\text{std}(j)}\right)$$

Diagram 1: Pearson correlation coefficient (1)

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

$$L_1(i,j) = \sum_{k=1}^{m} |M_{ik} - M_{jk}|.$$

Figure 2: Manhattan distance (2)

Where ρ(i,j) is the Pearson coefficient between Mik and Mjk which are the Manhattan distances between rows (i,k) and (j,k) respectively.

Manhattan distance makes the correlation calculation in a computer easier, and the Manhattan distance between two rows i and j is:

The searching used along with correlation attribute evaluation is Ranker which ranks the attributes or features in the order of their coefficient values.

After that we apply the algorithms, namely collaborative filtering along with matrix factorization using single layer perceptron, collaborative filtering with deep matrix factorization using multilayer perceptron and multilayer autoencoder. The difference that we have be- tween applying a machine learning algorithm and a deep learning based one is that machine learning uses a single convolutional net whereas a deep model implements multiple nets so what we will be doing here is mainly taking a single layer perceptron, which is a machine learning algorithm and making it into a deep model but applying multiple nets, dense and dropout layers which hopefully yield better results. This mainly owes to the fact that multi-ple hidden layers help identify other relationships between the known variables rather than just the ones visible to the eye and that way it tends to yield better recommendations for recommender systems.

The main goal of the research proposal is to analyse and compare the accuracies of these three algorithms on datasets to see if a deep learning model is better than a machine learning model as well as to identify if an unsupervised model will work as good as a supervised one. Towards the end we will also have an analysis and visualization of the results so as to discuss the outcome and what the future work that can be done. This research proposal as a whole is an analysis of one machine learning algorithm and multiple deep learning algorithms whereby we will test it all against various evaluation metrics then compare these algorithms against the outcome of other machine learning, state-of-art algorithms.

Thus the main objective here is to be able to show that the algorithms are better than the current ones in use in certain ecommerce platforms and that their sales or recommendations could probably be made better with such algorithms that use deep learning.

3. Technical specification

The system we are working on runs a Windows 10 64-bit operating system with an Intel i5 processor and 6 GB RAM for GPU processing, a higher RAM is available, is preferred for faster and more efficient results. The operating system does not matter but however the processor and RAM capacity does matter. Training of epochs of an algorithm would be significantly faster if the RAM were about 12 GB but since we do not have that kind of equipment at hand, we will not be able to do so. For the coding of the algorithms, we will be using Miniconda 3, Jupyter Notebook with Python 3 with TensorFlow, Keras and Scikit-learn packages Tensorflow is a package in python which is widely used for every machine learning algorithm and python provides a lot of in-built functions that would make it slightly easier to build algorithms. Keras and Tensorflow have many in-built algorithms and evaluation metrics that ease the work. We will not be using any hardware since the aim here is to build a recommender engine and does not involve any hardware as such. For preprocessing, analysis and

Published by Arab Journals Platform, 2020

visualization we will be using the Weka Tool and MS Excel. The Weka tool is a tool that has various preprocessing techniques such as feature selecting, reduction of dimensionality, instance selection, and also has various machine leanring algorithm in-built which we will not be using since we have to modify these algorithms according to the need and also since Weka does not have any deep learning algorithms. Excel will be used to give visualization or graphs of the tables towards the end of the paper.

4. Design approach and details

The architecture involves selecting a dataset and then pre-processing the data such as data cleaning, attribute and instance selection which will be done with the help of Weka Tool and then the algorithms will be applied after which their metrics will be evaluated and compared against state of art.

4.1. Detailed design approach and methods

The first step involves data pre-processing where we use the Weka tool and remove du- plicate rows of data, and we use CfcSubsetEval and Correlation attribute Eval as the main evaluation

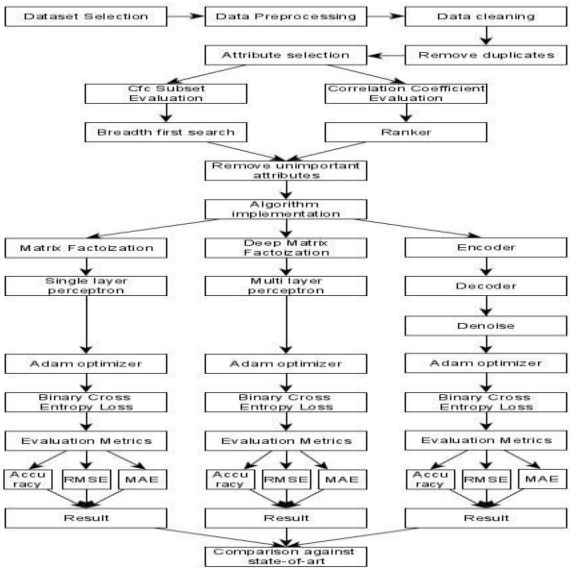

Figure 3: 4.1 Detailed design architecture

methods in Weka. The Cfc subset evaluator basically evaluates the importance of a certain subset of attributes uses breadth first search to individually calculate the predictive ability of each attribute or feature. Subsets of attributes that are very highly correlated to the class attribute but also having low intercorrelation are generally chosen. Correlation attribute evaluator uses the Pearson correlation coefficient which evaluates the closeness of linear relation between two attributes and Ranker ranks this in the order of most closeness.

Most recommender system papers have used collaborative filtering for recommendations. Amazon and Netflix currently use this to recommend products. But the usual collaborative filtering techniques do not learn if there are any non-linear connections between the attributes in the data set. Linear collaborative filtering is the act of filtering or making evaluations of

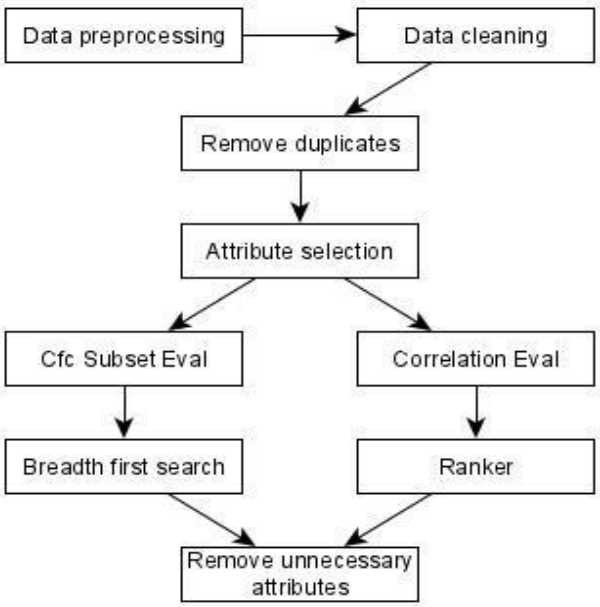

Figure 4: 4.2 Data preprocessing

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

items based on the opinions of other people. The mainly used classes of collaborative filtering algorithms are user-based and item-based nearest neighbourr, graph-based algorithms, rule-mining algorithms and convolutional neural networks. [34] Collaborative filtering withdeep matrix factorization, we hope will give better results.

Here we propose a similar collaborative filtering structure but with a feed forward convolutional neural network for matrix factorization which learns the non-linear characteristics between attributes. Both the single layer perceptron and the multilayer perceptron use ma- trix factorization but the main difference being the matrix factorization used in multilayer perceptron is a deep algorithm which learn other characteristics compared to just a normal dot product. It is detailed below.

Matrix factorization is a simple idea that tries to learn connections between the known values in order to impute the missing values with a dot product of the factors. But the problem is, matrix factorization is a linear method, meaning that if there are complicated non-linear interactions going on in the data set, a simple dot product may not be able to handle it well.

Deep matrix factorization involves the replacement of the dot product with a neural network that is trained jointly with the factors. This makes the model more powerful because a neural network can model important non-linear combinations of factors to make better predictions. There also is deep semi-matrix factorization which learns a small set of hierarchical structure of the features, with every layer learning the representation suitable for clustering as per the various attributes [35] while deep matrix factorization projects the users and items into a separate latent structured space [36]. Below is the architecture of a deep matrix factorization.

As you can see, in Deep Matrix Factorization is an algorithm that embeds the user ratings and the users to provide the rating index whereas in traditional deep matrix factorization, the users and ratings are only used separately to get the rating index.

Once the matrix factorization is done, we fit it into a single layer perceptron which effectively means a single hidden layer. A perceptron's objective is to find whether the input entered giving a output of true or false which effectively means that it can only give a 0 or 1 as an output. Thus, in this recommendation systems, we only try to predict the outcome of whether a user will select the item or not. The only difference in a multilayer perceptron is that the number of hidden layers in it will be more depending on the amount of efficiency as per requirement. The other differences of how a multilayer perceptron works and how we have modified it for use within the system is mentioned later in the paper.

The other algorithm we will be using is an autoencoder. Now the difference of an autoencoder is that it is an artificial neural network that is unsupervised. An autoencoder is a combination of an encoder and a decoder which first encodes the data and then builds the data back into a form that is as close to the original after reducing noise as much as possible. The working of it is explained later in the paper.

Here comes the evaluation of the results where we will use three evaluation metrics, one loss function and an optimizer and then will come comparison between the algorithms.

4.2. Constraints

Application of the algorithms on two datasets may not verify its generalizability, which would require further testing and research on multiple datasets to verify. Also, the efficiency and accuracy of the algorithms used in the current research is based on the hardware that

Published by Arab Journals Platform, 2020

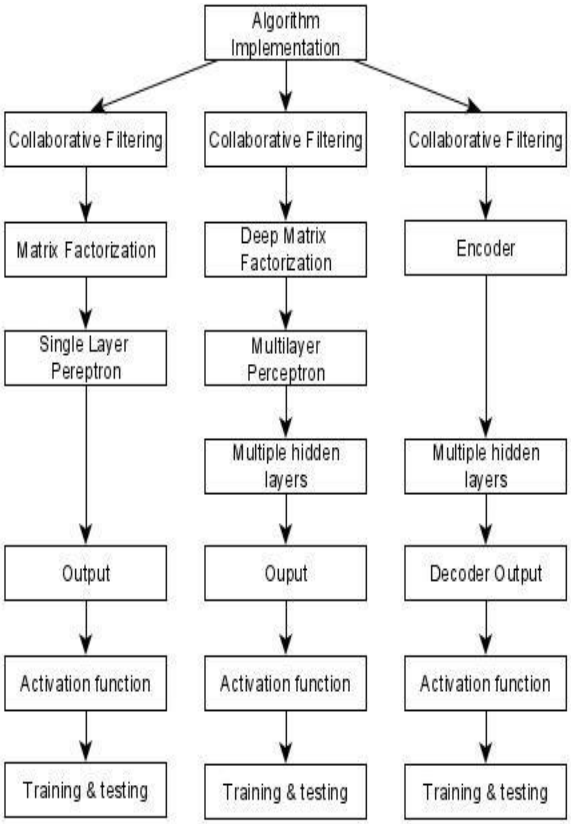

Figure 5: 4.3 Algorithms implemented

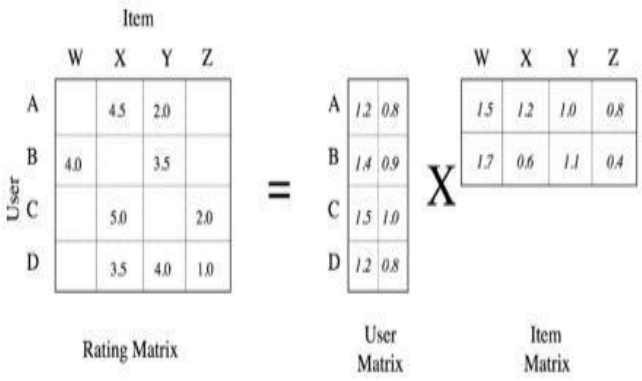

Figure 6: Linear matrix factorization

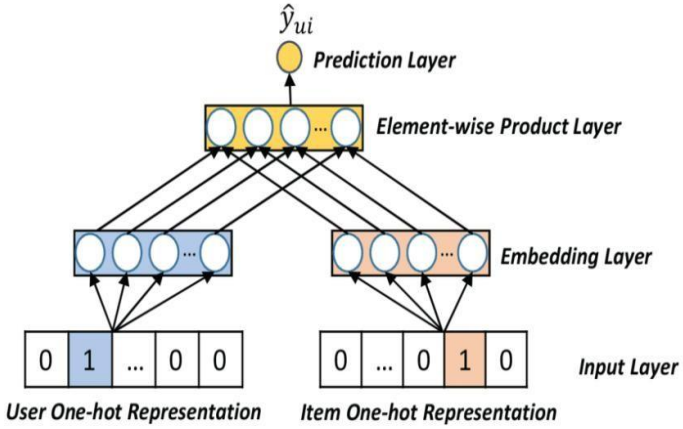

Figure 7: 4.5 Deep Matrix Factorization Architecture

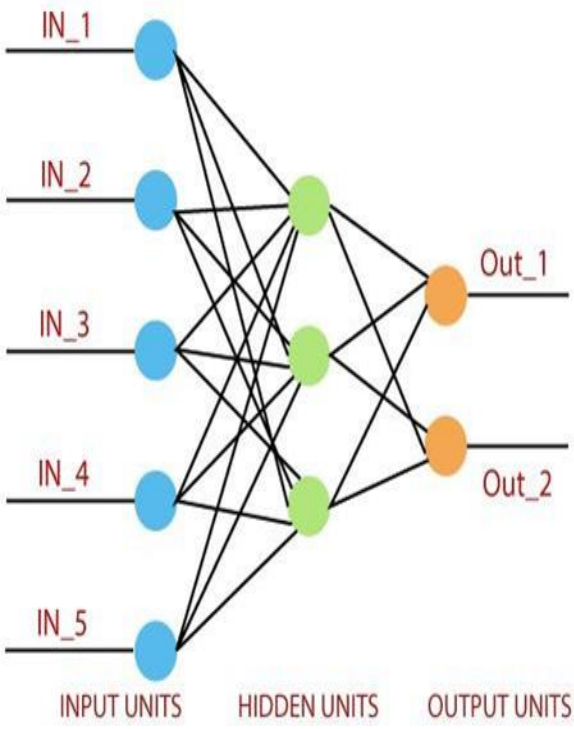

Figure 8: 4.6 A single-layer perceptron

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

we will be using which is a 6 GB RAM computer. A higher RAM memory would ensure the processing happens faster and gives better results. Another constraint that we have found that is that certain datasets available online are only available in certain file formats and hence data pre-processing on json files are difficult. Thus, certain algorithms may not be able to use these datasets efficiently. This is later explained in the paper.

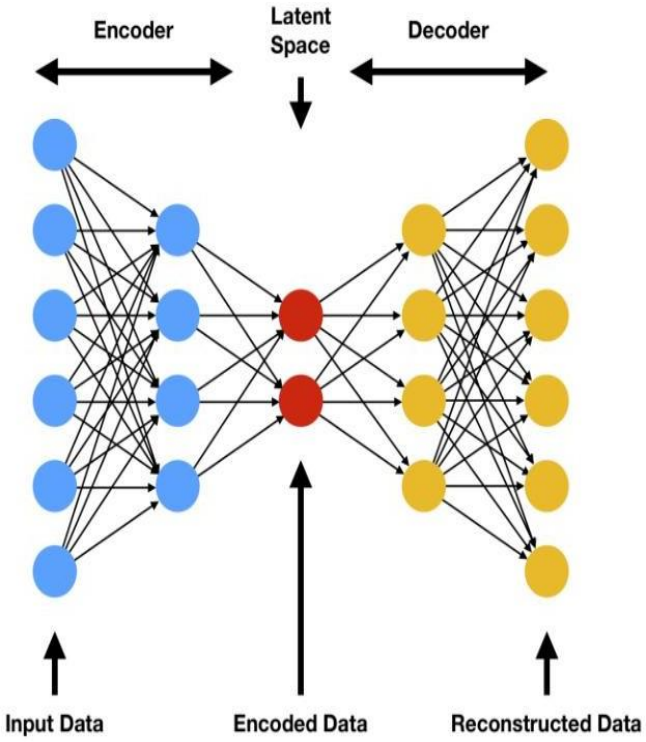

Figure 9: 4.7 An autoencoder encoding and decoding

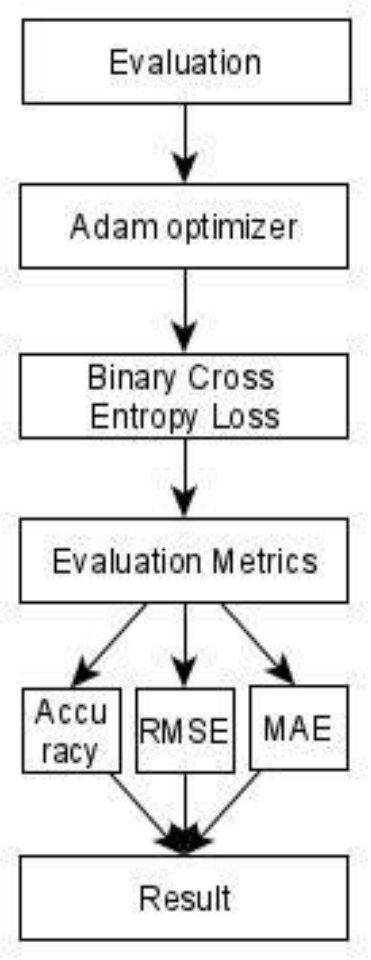

Figure 10: 4.8 Evaluation flow

5. Dataset election

The datasets we have chosen are the Movie Lens 100k dataset which has 943 users, 1682 movies and 100,000 ratings which is available on https://grouplens.org/datasets/movielens/100k/and the Amazon Grocery and Gourmet Food dataset with 127496 users, 41320 items and 1,143,860 ratings available on

https://nijianmo.github.io/amazon/index.html.
We will be using these datasets to verify the algorithm experimentally. The Amazon Grocery and Gourmet food dataset is relatively unused but only available in the json format. We will try to work with that as much as possible.

Published by Arab Journals Platform, 2020

Attributes of movie dataset (3 files):

Ratings file attributes – user id, movie id, Rating, timestamp.
Users file attributes – user id, gender, age, occupation, and zip-code.
Movies file attributes – movie id, title, genres
Attributes of Grocery and Gourmet Dataset (1 file): Overall (rating), verified (yes or no), review time, reviewer id, asin (item ID), review, summary, reviewer name, vote, style, unix review time.

5.1. Data cleaning

Collectively this proposal chose a 3 separate files as the datasets , The proposal use Weka Tool for clean each dataset of its missing values and inconsistencies There were around 119 missing values in the User.csv file, for which we have used replace Missing Values which replaces the missing values with the modes from the data. In the movies.csv file as we can see the dataset is only 97% unique hence we need to remove the duplicate instances. From 3944 instances it has come down to 3883 instances now. Thus we successfully have done a bit of cleaning of the data and the view after the removal of duplicates is shown below. This was necessary to ensure a better results in the recommendation since duplicates and missing values can negatively impact recommendations while training and testing.

6. Proposed algorithms

Three algorithms have been implemented here, namely Collaborative Filtering with Ma- trix Factorization Single Layer, Collaborative Filtering with Deep Feed Forward Multilayer Perceptron and Collaborative Filtering with an Autoencoder and then a denoised autoen- coder.

7. Single layer perceptron gives a 0 or 1 output from the inputs entered

A perceptron has mainly four important aspects which are an input layer, bias for the layer and weights for each of the neurons in the input layer, an input sum of it and an activation function. The input sum is a sum of the products of each weight and its input and finally summed up with the bias and then an activation function which you could say, standardizes the value to either 0 or 1 and thus giving the final value of the perceptron. The

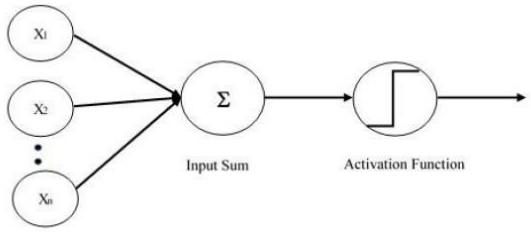

Figure 11: Working of single layer Perceptron

formula for a perceptron is given right below. Then a likelihood estimator ort a similarity quotient will estimate how close to user selections the output may be. (3)

$$y_j = \varphi\left(\sum_{i=1}^{n} \omega_{ij}\, x_i + b_j\right)$$

Figure 12:

7.1. Deep Feed Forward Multilayer Perceptron

An MLP is usually seen as a logistic regressor in which the input is first modified using a known non-linear transformation. This transformation puts the data into a place where it becomes separable as linear. The middle layer here is referred to as a hidden layer. One hidden layer is enought to make an MLP a universal approximator.

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

But if you have many hidden layers can help learn complex representations and more non-linearities in the data. Use of multiple hidden layers makes it deep learning. Also, in addition to the classic dense layers, we now also have layers such as dropout, convolutional, pooling, and recurrent layers. Pooling layers are utilized to reduce the dimensions of the outputted feature maps from convolutional laters. So basically it reduces the number of parameters to be learnt and the magnitude of computation done in the neural network. Dense layers are usually used

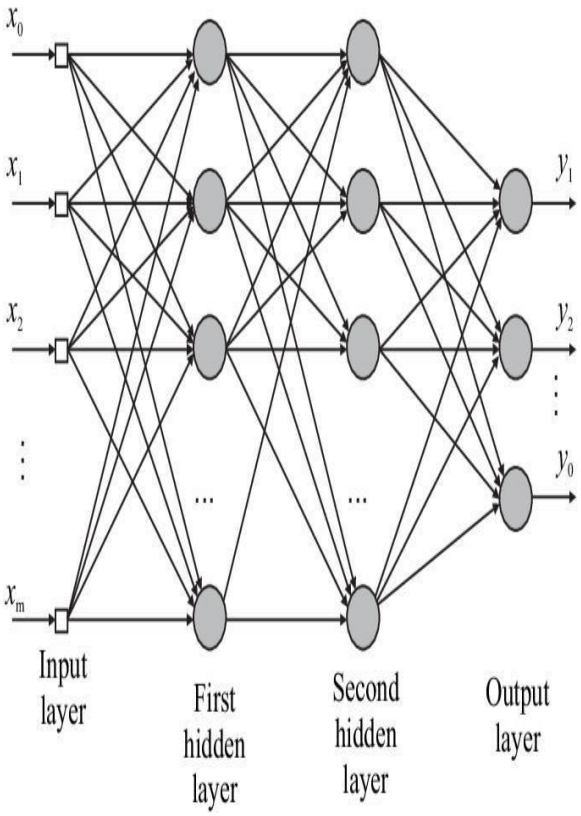

Figure 13: Multilayer Perceptron architecture

Along with these other known types. The mathematical definition of a single hidden layer multilayer perceptron is a function:

Table 2:

$$f: R^D \longrightarrow R^L \quad (4)$$

where D is the length of input x and L is the length of output f(x) such that in a matrix notation: $f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x)))$ with bias vectors b(1), b(2) weight matrices W(1) and W(2) and activation functions G and s.

Such models are named as feedforward because data flows through the function being evaluated from the input x, through the intermediate functions and computations used to define f, and then at the end, to the output y.

7.2. Batch Normalization

Deep artificial neural networks are not that easy to train, in fact it is quite difficult since the input from the previous layers can suddenly change after weight updates. Batch normalization is one such technique used to standardize the inputs to whichever network is being used, applied to ether the activations of a previous layer or to inputs directly. Batch normalization speeds up training, in some cases by halving the epochs or maybe even better, and provides some regularization, reducing the generalization error by a great margin.

7.3 Dropout layers

Dropout is a good technique used to prevent a model from overfitting. How dropout works is by randomly setting the outgoing edges of hidden units (neurons that make up the hidden layers) to 0 at each update of the training phase. A standard fully-connected neural net layer is also called a dense layer.

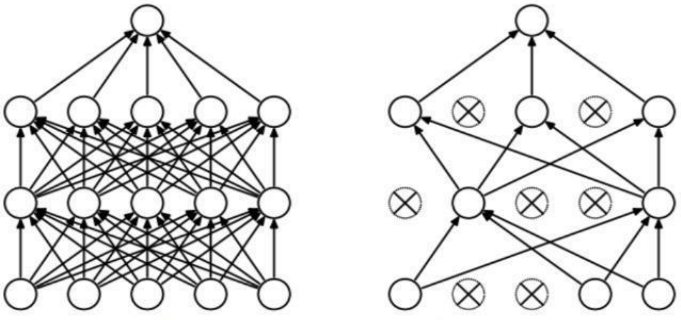

Figure 14 :6.8 Dropout in nets

7.4 Activation function

An activation function is a non-linear type of transformation that we do on top of the input before transmitting it to the next hidden layer or send it as final output. Here we have used sigmoid and reLU activation functions.

Published by Arab Journals Platform, 2020

The biggest advantage that sigmoid has over step and linear function is that it is non-linear. This is an important feature of the sigmoid function. This basically means that when there are multiple neurons having sigmoid function as their activation function – the entire output will also be non-linear. The function ranges from 0-1 and has somewhat of an S shape. The ReLU function stands for Rectified linear unit. It is actually the most widely used activation function. The major advantage of the usage of this ReLU function over other activation functions is that it does not activate all the neurons at the same time. This would be more understood if you look at the ReLU function where we can see that if the input is negative it will convert it to zero and thus the neuron will not get activated.

7.5 Regularization

Regularization is a widely used technique which does minor modifications of the learning algorithm so that the model does generalization better. This greatly improves the model's performance on the unseen or unknown data as well. A model of regression that usually utilizes L1 regularization technique is called Lasso Regression and any such model that uses L2 is called a Ridge Regresssor.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Cost function Cost function

Figure 15

Ridge regression adds a term "squared magnitude" of coefficient as penalty term to the original loss function. Lasso Regression (Least Absolute Shrinkage and Selection Operator) adds another term "absolute value of magnitude" of coefficient as penalty term to the original loss function. Here we have used L2 regularizer.

7.6 Working of a Multilayer Autoencoder

An autoencoder has two part:

Encoder: The encoder compresses the entered input into a latent-space matrix type of representation. Its representation cam be a function h=f(x).

Decoder: The decoder rebuilds the input from the latent-space matrix type of represen- tation.

Its representation can be a function r=g(h). Working of a multilayer auto encoder The function r=g(h) means that the decoder will try to decode the latent type of representation into a form that is as close as possible to the entered input. The main idea of an autoencoder is to remove the noise and also reduce dimensions of data, this way it can learn the data much easier and faster. In the study we will be using a multilayer autoencoder which has 5 convolutional layers, the first is the input layer with 512 input neurons and then 3 hidden

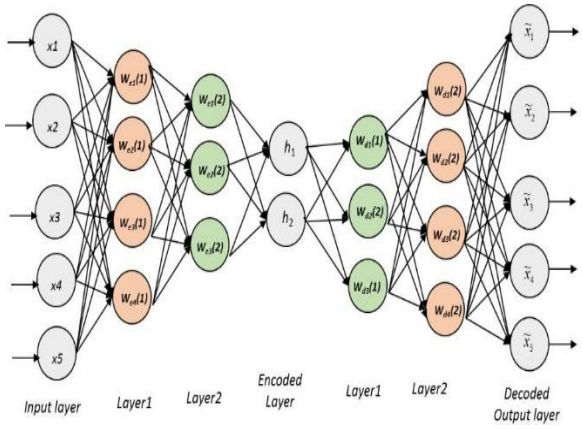

Figure 16: 6.11 Multilayer autoencoder architecture

layers with 256, 128 and 256 neurons each and finally the output layer once again with 512 neurons. Here again we will be using dropout and dense layers, activation function, regularization and optimizers. Instead of implementation of a penalty to the loss function, we implement another autoencoder that learns more useful information by modifying the reconstruction error term of the loss function.

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

This is achieved by adding some noise of the input data and making the autoencoder learn to remove it. By doing this, the encoder will extract the most important features and learn a more very robust representation of the data.

8. Result & discussion

8.1. Optimizers, loss and metrics

8.1.1. Optimizer: Adam

Adam is a reliable adaptive algorithm for optimizing the learning rate and this was designed specifically only for training and testing deep neural networks. It utilizes something called squared gradients to scale or improve the learning rate like RMSprop and it takes advantage of the momientum by using a moving average of the gradient instead of using the gradient itself like an stochastic gradient with momentum.

8.1.2. Loss function: Binary cross entropy

This is also named as Sigmoid Cross-Entropy loss. It is the combination of sigmoid activation and a Cross-Entropy loss. Unlike other losses like softmax loss, this is independent for each vector component (class), which means that the loss computed for every convolutional neural network output vector component is not at all affected by the other component's values. This is why it is mainly used for multi-label classification, where the insight for an element belonging to a certain class should not influence the decision for another class in any case.

8.1.3. Evaluation Metrics

The evaluationmetrics we have used here are: mean_absolute_error, root_mean_squared_error, binary_accuracy and val_loss. Distribution of movies:

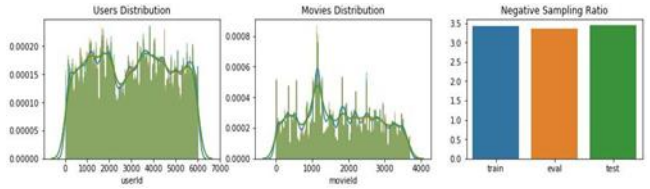

Figure 17

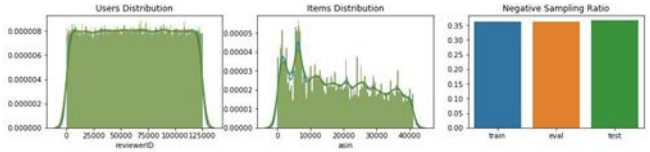

Figure 18

8.2. Comparison of algorithms

Since we have applied different algorithms, let us now look at the results and the comparisons of the algorithms with the different metrics we have identified earlier. First let us look at the Movie Dataset.

The above table clearly identify that the autoencoder and denoised autoencoder have outperformed the SLP and MLP algorithms that we have implemented. Even though MLP is a deep learning algorithm, we believe it is not as efficient as an autoencoder as we have

Table 3: Movie dataset algorithm comparison

Algorithm/Metric	Loss	Binary accuracy	RMSE	MAE
SLP	0.4565	0.7991	0.3730	0.2843
MLP	0.4439	0.7858	0.3765	0.2901
Autoencoder	2.8987	0.9429	0.7150	0.5759
Denoised autoencoder	1.8527	0.9501	0.5567	0.4144

Published by Arab Journals Platform, 2020

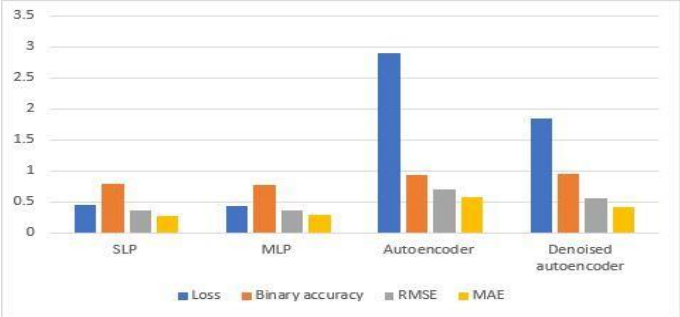

Figure 19: Fig.7.1 Movie dataset algorithm comparison

seen from the findings. The MLP can further be tested with different dropout, with more training epochs, and more hidden layers to get a better value but it still would seem less performing as compared to the autoencoder we have implemented.

But the other drawback we can notice in this experiment is that the autoencoder has given a poor value of RMSE and MAE as compared to MLP and SLP which means it has not fit the model as good as we may have liked. Now this may be due to the dataset we have used, still remains to be understood. Drawbacks need to be identified and rectified.

In grocery dataset, the limitation in terms of applying algorithms on it because the Amazon dataset are only available in the json format and there is no json to csv converter that could convert it. The main necessity for that is, for the ID column, the data is nominal, which means each product has an ID is alphanumeric and hence an index for the data cannot be made which is necessary to make a matrix where row number is the item number and the column number is the movie number and the rating is put in the field. Then a pivot table is made and split into various testing groups.

Algorithm/Metric	Loss	Binary accuracy	RMSE	MAE
SLP	0.5876	0.7564	0.4175	0.3522
MLP	0.5351	0.7657	0.4077	0.3268

Table 4: Grocery dataset algorithm comparison

From this table we can see that for the grocery dataset, the multilayer perceptron has worked better than the single layer perceptron but still has not provided a great result. It remains to be seen if other datasets may perform better on this algorithm.

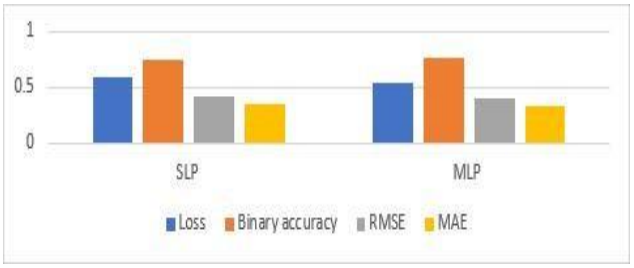

Figure 20: 7.2 Grocery dataset algorithm comparisons

As far as the experiment is concerned, let us compare the movie dataset with state of the art machine learning algorithms with SVD, SVD++ and KNN values that have already been found out by earlier researchers on the same movie-lens datasets.

Table 5: State-of-art movie dataset algorithm comparison

Algorithm/Metric	RMSE	MAE
SVD	0.934	0.737
SVD++	0.92	0.722
KNN	0.98	0.774
SLP	0.3730	0.2843
MLP	0.3765	0.2901
Autoencoder	0.7150	0.5759
Denoised autoencoder	0.5567	0.4144

Fig. 7.3 State-of-art movie dataset algorithm comparison

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3

In all the models outperform the classic machine learning algorithms in terms of their RMSE and MAE values. And thus it is fair to say that deep learning algorithms have done better than the traditional machine learning algorithms. But as mentioned earlier there is a lot of improvement and have to continue with researching and developing and working on other datasets. As far as the current research is concerned we have received satisfactory results.

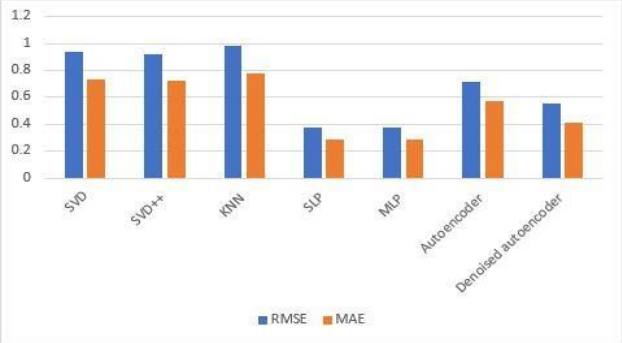

Figure 21:

9. Conclusion

Through the experiment were able to analyse and prove the accuracy of deep learning algorithms over traditional machine learning algorithms and the use of these algorithms in providing recommendations would significantly improve the quality of recommendations provided by the system. Towards the end of this research proposal we were able to see that autoencoders, especially the autoencoder that was trained to remove noise was outperforming all the other algorithms. The accuracy of the encoders, RMSE and MAE values were promising but the binary cross entropy loss was a little high meaning there is still room for improvement. This will be further researched and worked upon to see where changes can be made to reduce the loss. Furthermore, we have seen that in the case of two datasets, MLP, was effective in terms of its RMSE and MAE values but may tend to be less accurate at times.

But the fact that it returned similar results for both datasets shows us that it is a very generalized model and can be applied to any dataset.

Looking at the auto encoder, need to test it on other datasets that are available to prove its generalisability. As of since we have only tested it on one dataset, this cannot be proven. This will done by furture work. But what we can come to a conclusion is that a multilayer autoencoder has worked far better than the other models we have implemented and also is better when we compared it against other machine learning algorithms are in use right now. A the end of the experiment , hope to get even better results when tested on other datasets and other algorithms that may come up in the near future.

References

[1] H. C. Chen, A. L. Chen, A music recommendation system based on music and user grouping, Journal of Intelligent Information Systems 24 (2-3) (2005) 113–132.

[2] M. Krstic, M. Bjelica, Context-aware personalized program guide based on neural network, IEEE Transactions on Consumer Electronics 58 (4) (2012) 1301–1306. doi:10.1109/tce.2012.6414999.
URL https://dx.doi.org/10.1109/tce.2012.6414999

[3] A. H. Dong, D. Shan, Z. Ruan, L. Y. Zhou, F. Zuo (2013).

[4] B. M. Shoja, N. Tabrizi, Customer Reviews Analysis With Deep Neural Networks for E-Commerce Recommender Systems, IEEE Access 7 (2019) 119121–119130. doi:10.1109/access.2019.2937518.
URL https://dx.doi.org/10.1109/access.2019.2937518

[5] Q. Mao, A. Dong, Q. Miao, L. Pan, Intelligent Costume Recommendation System Based on Expert System, Journal of Shanghai Jiaotong University (Science) 23 (2) (2018) 227–234. doi:10.1007/s12204-018-1933-x.
URL https://dx.doi.org/10.1007/s12204-018-1933-x

[6] N. J. Yuan, Y. Zheng, L. Zhang, X. Xie, T-finder: A recommender system for finding passengers and vacant taxis, IEEE Transactions on knowledge and data engineering (2012).

[7] P. Chen, H. Lv, S. Gao, Q. Niu, S. Xia, A Real-Time Taxicab Recommendation System Using Big Trajectories Data, Wireless Communications and Mobile Computing 2017 (2017) 1–18. doi:10.1155/2017/5414930.URL
https://dx.doi.org/10.1155/2017/5414930

Published by Arab Journals Platform, 2020

[8] L. Ravi, S. Vairavasundaram (2016).
[9] J. Hao, Y. Yan, G. Wang, L. Gong, B. Zhao (2016).
[10] D. Wu, J. Lu, G. Zhang, A fuzzy tree matching-based personalized e-learning recommender system, IEEE transactions on fuzzy systems 23 (2015) 2412–2426.
[11] S. B. Aher, L. M. R. J. Lobo, Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data (2013). doi:10.1016/j.knosys.2013.04.015.
URL https://dx.doi.org/10.1016/j.knosys.2013.04.015
[12] S. Xing, Q. Wang, X. Zhao, T. Li, Content-aware point-of-interest recommendation based on convolutional neural network, Applied Intelligence 49 (3) (2019) 858–871.
[13] S. Liu, X. Meng (2015).
[14] Z. Bahramian, R. A. Abbaspour, C. Claramunt (2017).
[15] Y. Wang, M. Wang, W. Xu, A sentiment-enhanced hybrid recommender system for movie recommendation: A big data analytics framework, Wireless Communications and Mobile Computing (2018).
[16] D. Yang, J. Zhang, S. Wang, X. Zhang, A Time-Aware CNN-Based Personalized Recommender System, Complexity (2019).
[17] Y. Shao, C. Wang, HIBoosting: A Recommender System Based on a Gradient Boosting Machine, IEEE Access 7 (2019) 171013–171022.
[18] F. Guo, Q. Lu (2015).
[19] T. Hao, Q. Wang, D. Wu, J. S. Sun, Adaptive recommendation for photo pose via deep learning, Multimedia Tools and Applications 77 (17) (2018) 22173–22184.
[20] F. Deng, P. Ren, Z. Qin, G. Huang, Z. Qin (2018).
[21] K. Shrivastava, S. Kumar, D. K. Jain (2019).
[22] Z. Wang, K. Hahn, Y. Kim, S. Song, J. M. Seo (2018).
[23] Z. Li, Z. Liu, J. Huang, G. Tang, Y. Duan, Z. Zhang, Y. Yang, MV-GCN: Multi-View Graph Convolutional Networks for Link Prediction, IEEE Access 7 (2019) 176317–176328.
[24] Q. Tan, F. A. Liu (2019).
[25] B. Ramzan, I. S. Bajwa, N. Jamil, R. U. Amin, S. Ramzan, F. Mirza, N. Sarwar (2019).
[26] G. Manogaran, R. Varatharajan, M. K. Priyan, Hybrid Recommendation System for Heart Disease Diagnosis based on Multiple Kernel Learning with Adaptive Neuro-Fuzzy Inference System, Multimedia Tools and Applications 77 (4) (2018) 4379–4399. doi:10.1007/s11042-017-5515-y.
URL https://dx.doi.org/10.1007/s11042-017-5515-y
[27] X. Deng, F. Huangfu, Collaborative Variational Deep Learning for Healthcare Recommendation, IEEE Access 7 (2019) 55679–55688. doi:10.1109/access.2019.2913468.
URL https://dx.doi.org/10.1109/access.2019.2913468
[28] J. H. Su, H. H. Yeh, S. Y. Philip, V. S. Tseng, Music recommendation using content and context information mining, IEEE Intelligent Systems 25 (1) (2010) 16–26.
[29] J. H. Chang, C. E. Tsai, J. H. Chiang, Using heterogeneous social media as auxiliary information to improve hotel recommendation performance, IEEE Access 6 (2018) 42647–42660.
[30] J. Fan, D. A. Keim, Y. Gao, H. Luo, Z. Li, JustClick: Personalized Image Recommendation via Exploratory Search From Large-Scale Flickr Images, IEEE Transactions on Circuits and Systems for Video Technology 19 (2) (2009) 273–288. doi:10.1109/tcsvt.2008.2009258.
URL https://dx.doi.org/10.1109/tcsvt.2008.2009258
[31] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, Acm transactions on interactive intelligent systems (tiis) 5 (2015) 1–19.
[32] S. Gnanambal, M. Thangaraj, V. T. Meenatchi, V. Gayathri, Classification Algorithms with Attribute Selection: an evaluation study using WEKA, International Journal of Advanced Networking and Ap- plications 9 (6) (2018) 3640–3644.
[33] D. J. Chang, A. H. Desoky, M. Ouyang, E. C. Rouchka, Compute pairwise manhattan distance and pearson correlation coefficient of data points with gpu, 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing (2009) 501–506.
[34] J. B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The adaptive web, Springer, 2007, pp. 291–324.
[35] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B. W. Schuller, A Deep Matrix Factorization Method for Learning Attribute Representations, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (3) (2017) 417–429. doi:10.1109/tpami.2016.2554555.
URL https://dx.doi.org/10.1109/tpami.2016.2554555
[36] H. J. Xue, X. Dai, J. Zhang, S. Huang, J. Chen, Deep Matrix Factorization Models for Recommender Systems, IJCAI (2017) 3203–3209.

https://digitalcommons.aaru.edu.jo/fcij/vol5/iss2/3
DOI: http://doi.org/10.54623/fue.fcij.5.2.3