International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 2 (2018) pp. 1171-1176

Students' Performance Prediction Using Deep Neural Network

Bendangnuksung [1] and Dr. Prabu P [2]

[1]*Department of Computer Science, Christ University, Bengaluru, Karnataka, India.*

[1]*ORCID: 0000-0001-6319-1308*

[2]*Assistant Professor, Department of Computer Science, Christ University, Bengaluru, Karnataka, India.*

Abstract

Deep Learning and Educational data mining has gain a considerable amount of attention in this past years. In this paper, a neural network called Deep Neural Network (DNN) model is proposed that shows students which class category it belongs to . This provides knowledge to the institution so that they can offer a remedy to the potential failing students. A comparison with existing machine learning algorithm which uses the same dataset with the proposed model. The proposed deep neural network model achieved up to 84.3% accuracy and outperforms other machine learning algorithms in accuracy.

Keywords: Deep Neural Network (DNN), Deep Learning, Artificial Neural Network (ANN), Education Data Mining.

INTRODUCTION

Academic students' performance has always been a major factor in determining a student's career and the prestige of the Institutions. Education Data Mining (EDM) is a discipline which is followed to extract meaningful knowledge from an educational context. EDM applications such as model development helps to predict student performance in their academics. Thus leading researchers to dig deep into various methods in data mining to improve existing method.

The applications of Machine Learning methods to predict students' performance based on student's background and term examination performances has turn to be helpful for foreseeing the different performance in various level. Using such machine learning methods enables to timely predict the students who has a high chance of failing so that a remedy can be provided by a teacher to the student. It can even help to detect high caliber students of the institution and help him providing scholarship.

Machine Learning algorithms such as Decision Tree [10] and Naive Bayes [9] is highly used in Educational Data Mining. There is a limitation to such algorithms, as stated by Havan Agrawal [11] when input is provided in a continuous range to Bayesian classification the accuracy of the models reduces. Such classification works better with discrete data. Also stated that a Neural Network outperforms when given a continuous data.

Deep Learning is considered as the state of the art [5] tool for artificial intelligence research which applied in various applications. Deep Learning can be classified as: Deep Neural Network (DNN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Q-learning. Deep learning has been lately used for voice/sound recognition [7], Natural Language Processing [8], computer vision [6].

In this paper, we proposed a Deep Neural Network (DNN) classifier model to predict the students' performance. The proposed DNN model aims to predict students whether they fall under fail category or pass category through logistic classification analysis. Two hidden layer is implemented, first hidden has Relu activation function, second hidden layer with a Soft-Max activation function. The proposed model is effective in predicting fail students, with an estimated 85\% accuracy.

LITERATURE REVIEW

Ioannis E. Livieris, et al. [1] built an Artificial Neural Network (ANN) classifier to predict the performance of students in Mathematics. From their experiments they found that the modified spectral Perry trained artificial neural network performs better classification compared to other classifiers.

A study was conducted by S. Kotsiantis, et al. [2] which investigated in distance learning of machine learning techniques for dropout prediction of students. Important contribution was made by this study as it was a pioneer and helped to carved the path for such educational data mining. Machine learning techniques were applied in other areas, he and his team were the first people to implemented machine learning methods in an academic environment. An algorithm was fed on demographic data and several project assignment rather than class performance data to make prediction of students.

Moucary, et al. [3] applied a hybrid technique on K-Means Clustering and Artificial Neural Network for students who are pursuing higher education while adopting a new foreign language as a means of instruction and communication. Firstly, Neural Network was used to predict the student's performance and then fitting them in a particular cluster which was form using the K-Means algorithm. This clustering helped in serving a powerful tool to the instructors to identify a student capabilities during their early stages of academics.

Hongsuk, et al. [4] proposed a Deep Neural Network supervised model to estimate link based flow of traffic

conditions. A Traffic Performance Index which was used for logistic regression to distinguish between a congested traffic condition to a non-congested traffic condition. With a 3 layer model it was able to estimate the cogestion with a 99% of accuracy.

Amrieh, et al. [12] proposed a prediction model for students' performance based on data mining methods with some few features called student's behavioral features. The model was evaluated in three different classifiers; Naïve Bayesian, Artificial Neural Network and Decision tree. Random Forest, Bagging and Boosting were used as ensemble methods to improve the classifier's performance. The model achieved up to 22.1% more in accuracy compared when behavioral features were removed. It increased up to 25.8% accuracy after using the ensemble methods.

The prediction model structure [12] has features with less difference in information gain. Since machine learning algorithm generally breaks down problem and solve them individually, there is a possibility of slipping away other important features when one condition fails. This affects the overall performance of the model. In order to do so the model has to accept end-to-end features.

Such problem can be solved using a deep neural net where all features are extracted and fed to the layers at once. After feeding a neuron an activation function will check the criteria condition and activates the neuron once it pass the function. Proper activation function needs to used in every layer for activating the right neuron.

DEEP NEURAL NETWORK

Deep Learning techniques aim to learn attribute hierarchies with attribute from higher levels that is formed by the combining of other low features. This includes learning multiple methods for higher and deeper architectures. DNN is a class of multiple NN models. Model with input layers, arbitrary number of hidden layers and an output layer. The layers are made up of neurons which share similarities to human brain neurons.

A neuron is a nonlinear function that maps input vectors *{I_1 ,.....,I_n)* to an output *Y* through a weighted vector *{w_1,....,w_n) and to a function f . Also known as feed-forward.*

$$Y = f\left(\sum_{i=0}^{k} w_i \, I_1\right) = f(w^t \, I)$$

The goal of the model is to optimize the weights ***w*** such that squared loss error is reduced. This can be achieved using stochastic gradient descent (SGD). SGD iteratively update weight vector which ultimate purpose is to direct to the minimum gradient of loss function. To obtain SGD update equation:

$$w^{new} = w^{old} - n \,.(Y - t) \,.\, Y(1\text{-}Y) \,.\, I$$

An Epoch is one feed-forward and one back propagation. Each epoch helps in reducing the cost function. In deep neural network an epoch is iterated n^{th} times, updating and optimizing the gradients.

Deep Neural Network (DNN) is a deep learning architecture that allows operational models which composed of several hidden processing layers to learn various representations of data with multi-level abstraction. Deep Learning has an excellent capability to self-learn and self-adapting, making it extensively studied and have successfully used to tackle real-world complex problems.

DATASET AND DATA PREPROCESSING

Source of data for building the proposed deep neural network to predict the students' performance is obtained from https://www.kaggle.com/aljarah/xAPI-Edu-Data. It is an educational dataset collected from learning management system called kalboard 360. The dataset consists of 500 student records. It has 16 different features.

Table : I

Students Dataset		
Name	**Data Type**	**Distinct Values**
Gender	Nominal	2
Nationality	Nominal	14
Place of Birth	Nominal	14
Stages	Nominal	3
Grades	Nominal	12
SectionID	Nominal	3
Topic	Nominal	12
ParentResponsible	Nominal	2
Semester	Nominal	2
Raised hand	Numeric	0-100
Visited Resource	Numeric	0-100
Viewing Announcement	Numeric	0-100
Discussion Group	Numeric	0-100
Parent Answering	Nominal	2
Parent Satisfaction	Nominal	2
Student Absent day	Nominal	2

The dataset has three classes based on their numerical interval values.

Table II

Classes	
Interval-Values	**Class Label**
0-69	Low
70-89	Middle
90-100	High

Preprocessing of data after data collection is required to improve the quality of dataset. Data attribute selection, data cleaning, data transformation and data reduction are all part of data preprocessing. It is a part in the knowledge discovery process. The dataset contains 20 missing values in different features from 500 records. All the missing values are removed, the number of records after data cleaning is 480.

Data transformation is applied to the dataset. Nominal data type attributes Gender, Relation, Semester, ParentAnsweringSurvey, ParentSchoolSatisfacation and StudentAbsenceDays are transformed to binary data '0' and '1'. Other nominal data type attributes Nationality, PlaceofBirth, StageID, GradeID, SectionID, and Topic are transform to numerical data type.

METHODOLOGY

In this paper, we introduce a Deep Neural Network linear classifier model to predict the performance of students. This process is followed once the dataset is preprocessed: data cleaning and data transformation. The DNN model is built using python3 and tensorflow 1.3.0.

Python is a full featured for general purpose programming language. It is a mature and fast expanding platform for scientific research and numerical computing. Python host numerous open source libraries and almost all general purpose libraries for machine learning which can be further use for deep learning models. All this benefits from the python ecosystem lead to the top two libraries for numerical analysis of deep learning was developed for python language, that is Tensorflow and Theano library.

TensorFlow is an open source library for computing numerical using data flow graphs. The data flow graphs is also known as Static Computation graph. A developer has to first design the input layer and connect every input layer to the hidden layer then the same from hidden layer to output layer. The graphs are made of tensors and ops, defining all the neural networks and all mathematical calculations. The session helps to run the graph. Tensorflow comes with Graphical Processing Unit package where all the matrix calculations can be done efficiently and much faster.

Once data is preprocessed, the data is divided into two parts training and testing dataset. It is divided in the ratio 3:1 (Train/Test). In training dataset, the features and classes are split and stored in a tensorflow placeholder. Both training dataset classes records are One-hot encoded, it is a process where class variables are converted into a numerical form that will be provided to deep neural network model for effective prediction.

Table: III

One-Hot Encoding	
Classes	One-Hot Encoding format
Low	[1, 0 , 0]
Middle	[0, 1 , 0]
High	[0, 0 , 1]
Middle	[0, 1 , 0]
Low	[1, 0 , 0]

Two hidden layer is defined with 300 neurons each and epoch of 50. Next we have to construct a dataflow graph. Random initialization of weights ***w*** and bias ***b*** to every interconnected layers (input, hidden, output) . Matrix multiplication of the first hidden layer is passed to a rectified linear called as relu activation, input ***x*** as the neuron where it is connected all the neurons of the first hidden layer.

$$f(x) = max\ (x,0)$$

All the neuron in the second hidden layer which are activated by the relu activation function are computed again with the matrix multiplication with the next layer. In the 2nd hidden layer it uses a different activation function called softmax function where the matrix calculation is passed. The softmax function squashes the output into a categorical probability of distribution which tells the probability of class likely to be the output. Here ***z*** as the vector from the input layer to the output layer and ***j*** as the index of output units.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

The output is then pass to the cost function where the output is compared to the actual output. The cost function returns the error, this error is forwarded to an optimization function called Adam optimizer function in tensorflow. The optimization function updates the weights of layers so that the cost function returns a lesser error value. Once the data flow graph is built, the static computation graph should be activated to run. Computation graph can be activated using the tensorflow session. Instantiating the session and passing the data inputs to the run function. In this model an epoch of 50 is defined where the computational graph is iterated 50 times in order to give higher accuracy.

The flow of the computational graph can be visualise as shown in the Fig. 1.

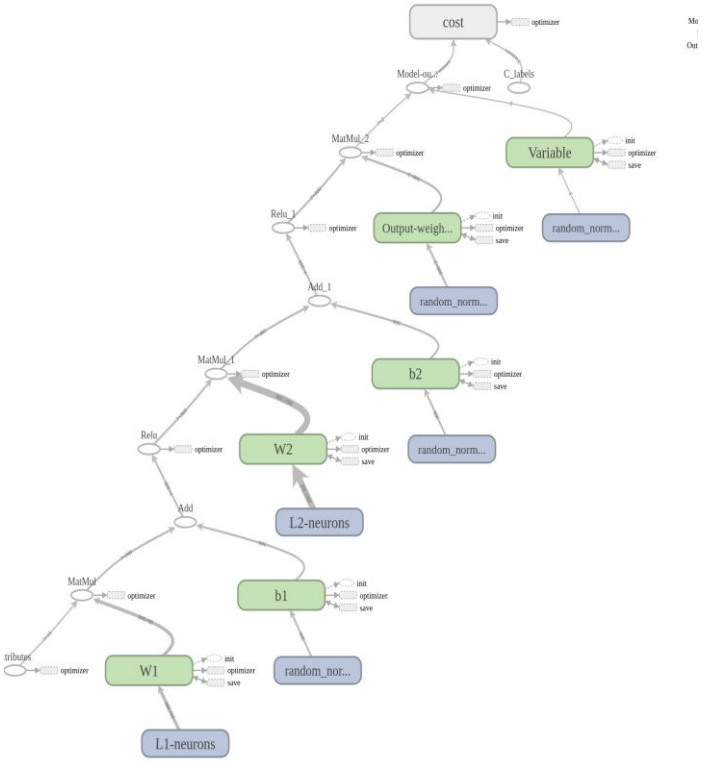

Figure 1. Graphical Computational Graph

EXPERIMENT AND RESULTS

The experiment was run on Ubuntu 16.04 operating system with the configuration of 8GB RAM and 4 Intel cores. Tools such as python3 and tensorflow was used to run the deep neural network. Tensorboard and matplotlib library was used to visualise the inner working of the model. The model took 5 minutes for executing the program.

In our experiments, we used two measures for evaluating the quality of the classifier, that is cost function and accuracy. The purpose of accuracy to achieve higher value where as the purpose of cost function is to reduce the value.

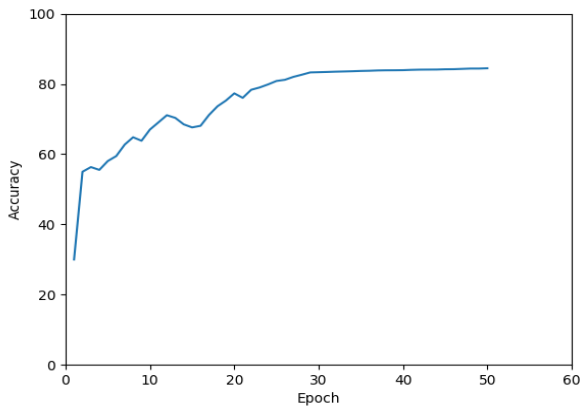

Figure 2. Accuracy

With the proposed deep neural network it was able to achieve the highest accuracy of 84.3%. The model initial accuracy is 29.8%. In the first epoch there is an increase of 20% accuracy. Once the target output does not match with the model output, it calculates the amount of error using the cost function softmax cross entropy. The adam optimizer function is then used to update the weights of the neurons such a way that the cost function reduces. Even though the dataset is very limited for a deep neural network, it still outperforms other machine learning algorithms. As dataset is limited the model should be precisely tweak in order to provide better performance. Initially four hidden layers with each of 300 neurons were install in the dataflow graph, this did not improve the model. There were few features so there was no need of many hidden layers and neurons. It was further decreased to 2 hidden layers and 100 neurons each. After 50 epoch there was slight fluctuation of accuracy, then the training was stop.

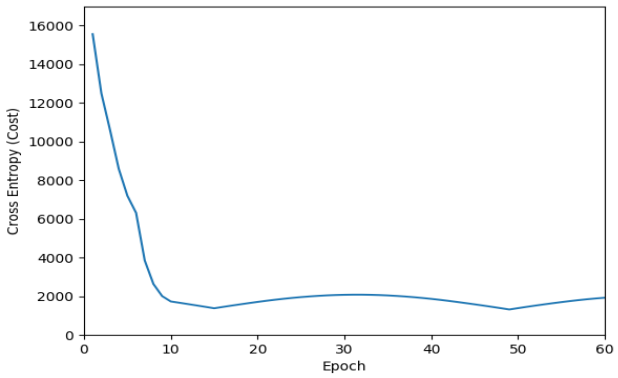

Figure 3. Cross Entropy Cost Function

The cost function error is very high initially, that is 15560. Through optimization this cost function error is reduces. The first optimization drastically reduces the cost function. To find the cost function, the softmax cross entropy is used. Over the period of iteration of feed forward and back propagation there is a high reduction of cost by the model. In the 17th iteration there is an increased of cost value due to the ineffective gradient updates but then cost starts to reduce from 34th iteration. There should always be a limit to the number of epoch because of certain limit the optimizer starts to inverse the weights thus leading to increase of cost error. In the proposed model, 50 epoch is considered as we can see after the 50th iteration there is a continuous increase of cost. Thus stopped at the 50th epoch. Analysis should be done for defining epoch. High number of epoch leads to over-fitting of model and less epoch leads to under-fitting of model. A precise understanding of data is needed before making assumption on the model.

A comparison is made to Amrieh, et al. [12] proposed model. In their paper they used three different machine learning algorithm, Decision Tree, Naive Bayes and Artificial Neural Network and the same dataset. This classification model was run in Weka tool with a cross validation of 10 folds. The final classification accuracy is considered and compared with the proposed model.

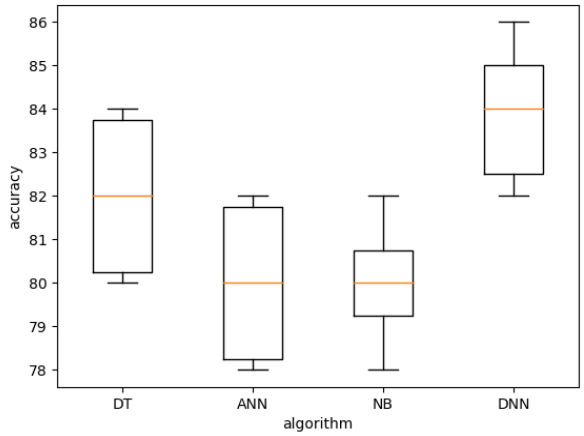

Figure 4. Algorithm Comparison

A whisker plot is used to show the spread of the accuracy scores of each cross validation of 10 folds for each algorithm in Fig. 4.

Table: IV

Classification Method Comparison	
Classifier	Accuracy
Decision Tree (J48)	82.2
Artificial Neural Netork (ANN)	80.0
Naive Bayes (NB)	80.0
Proposed Model (DNN)	84.3

As shown in the table we can see that, Deep Neural Network can outperform other machine learning classification methods even with less data but should precisely tweaking the model for better performance. Out of 120 students from testing set 19 students were wrongly classified. This model is can be reliable enough for prediction of students' performance.

CONCLUSION

A deep neural network model is proposed in this paper for predicting the students' performance. It is the first time to use a deep neural network for the education data mining and predicting of students' performance. Through the experiment we found that a DNN can perform better even with less amount of data by having deep knowledge about dataset and quality tweak on the model. The proposed model achieved an accuracy of 84.3%. With larger dataset records and features, a DNN can achieve higher accuracy and will outperform other machine learning algorithm. This model is reliable and can help to predict a student's performance and identify students who has higher chance of failing before hand to provide remedy.

ACKNOWLEDGMENT

I Bendangnuksung would like to thank my guide Dr. Prabu P and the Research Support Cell of Department of Computer Science, Christ University for giving me this opportunity to write a review on Students' Performance Prediction using Deep Neural Network.

REFERENCES

[1] Livieris, et al. (2012): Predicting students' performance using artificial neural networks 8th PanHellenic Conference with International Participation Information and Communication Technologies, pp.321-328.

[2] S. Kotsiantis, et al. (2003): Preventing student dropout in distance learning systems using machine learning techniques Applied Artificial Intelligence, 18(5), pp.411-426.

[3] Moucary, C.E., Khair, M. and Zakhem, W., 2011. Improving student's performance using data clustering and neural networks in foreign-language based higher education. The Research Bulletin of Jordan ACM, 2(3), pp 27-34

[4] Yi, Hongsuk, Jung, H. and Bae, S., 2017, February. Deep Neural Networks for traffic flow prediction. In Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on (pp. 328-331). IEEE.

[5] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. Nature, 521(7553), pp. 436-444.

[6] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014, November. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia pp. 675-678

[7] Dahl, G.E., Yu, D., Deng, L. and Acero, A., 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on audio, speech, and language processing, 20(1), pp.30-42.

[8] Collobert, R. and Weston, J., 2008, July. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning pp.160-167

[9] Koutina M, Kermanidis KL. Predicting postgraduate students' performance using machine learning techniques. InArtificial Intelligence Applications and Innovations 2011 (pp. 159-168). Springer, Berlin, Heidelberg.

[10] Saini, P. and Jain, A.K., 2013. Prediction using Classification Technique for the Students' Enrollment Process in Higher Educational Institutions. International Journal of Computer Applications, 84(14). Springer, Berlin, Heidelberg.

[11] Agrawal, H. and Mavani, H., 2015. In Student Performance Prediction using Machine Learning.

International Journal of Engineering Research and Technology.

[12] Amrieh, E.A., Hamtini, T. and Aljarah, I., 2016. Mining Educational Data to Predict Student's academic Performance using Ensemble Methods. International Journal of Database Theory and Application, 9(8), pp.119-136.

[13] Gorr, W.L., Nagin, D. and Szczypula, J., 1994. Comparative study of artificial neural network and statistical models for predicting student grade point averages. International Journal of Forecasting, 10(1), pp.17-34

[14] Li, J., Zhao, R., Huang, J.T. and Gong, Y., 2014. Learning small-size DNN with output-distribution-based criteria. In Fifteenth Annual Conference of the International Speech Communication Association

[15] Galbraith, C.S., Merrill, G.B. and Kline, D.M., 2012. Are student evaluations of teaching effectiveness valid for measuring student learning outcomes in business related classes? A neural network and Bayesian analyses. Research in Higher Education, 53(3), pp.353-374

[16] Stapel, M., Zheng, Z. and Pinkwart, N., 2016. An Ensemble Method to Predict Student Performance in an Online Math Learning Environment. In EDM (pp. 231-238).

[17] Huang, C. and Moraga, C., 2004. A diffusion-neural-network for learning from small samples. International Journal of Approximate Reasoning, 35(2), pp.137-161.

[18] Baker, R.S. and Yacef, K., 2009. The state of educational data mining in 2009: A review and future visions. JEDM-Journal of Educational Data Mining, 1(1), pp.3-17.

[19] Mythili, M.S. and Shanavas, A.M., 2014. An analysis of students' performance using classification algorithms. IOSR Journal of Computer Engineering, 16(1), pp.63-9.

[20] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks, 61, pp.85-117.

[21] Bebis, G. and Georgiopoulos, M., 1994. Feed-forward neural networks. IEEE Potentials, 13(4), pp.27-31.

[22] Tso, G.K. and Yau, K.K., 2007. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. Energy, 32(9), pp.1761-1768.

[23] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), pp.82-97.

[24] Stone, P. and Veloso, M., 2000. Multiagent systems: A survey from a machine learning perspective. Autonomous Robots, 8(3), pp.345-383.

[25] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of machine learning research, 15(1), pp.1929-1958.