International Journal of Research in Engineering, Science and Management
Volume 6, Issue 12, December 2023
https://www.ijresm.com | ISSN (Online): 2581-5792

Recommendation System for Learning Resources Using Collaborative Learning of Ensemble Neural Networks

Nency Sharma[1*], Anand Rajavat[2]
[1]M.Tech. Scholar, Department of Computer Science and Engineering, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India
[2]Director, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India

***Abstract*: With the increase in e-learning applications and its allied benefits, improving e-learning resources based on the requirements of the users has become a challenging undertaking due to the large number of resources. Different people prefer different learning resources especially based on their age, nation, neighborhood and requirements. There is no specific method for fitting make the recommendation system which can help users get to the best resources inside the least measure of time. However, collaborative learning has come up as an effective technique in machine learning based approaches for the design of recommendation systems. In this paper, a collaborative learning-based approach has been proposed for ensemble neural networks (ENNs) to design a recommendation system for learning resources. The proposed approach uses the resilient back propagation-based way to deal with train the ensemble neural network. It has been shown that the proposed approach outperforms previously existing techniques both in terms of error and iterations which indicates higher precision and lesser time which is a basic aspect in real time e-learning recommendation system applications.**

***Keywords*: e-learning, learning resource recommendation system, ensemble neural network, collaborative learning, mean square error.**

1. Introduction

E-learning has come to the forefront today due to the access to online experts, large number of resources and convenience of the targeted audience. Today, online learning resources are gaining huge popularity such as:

- Coursera
- Udemy
- NPTEL
- MIT.OCW etc.
- Byjus etc.

With several learning resources available online, a recommendation system can recommend appropriate resources based on the student's need. Automatic multimedia learning resources recommendation has become an increasingly relevant problem: it allows students to discover new learning resources that match their needs, and enables the e-learning system to target the learning resources to the right students. Automatic multimedia learning resources recommendation has become an increasingly relevant problem, it allows students to discover new learning resources that match their tastes, and enables the e-learning system to target the learning resources to the right students. The task of recommendation algorithms in e-learning systems is to give student a personalized and suitable learning service [1]. The learning resources are more and more diversified recent years, and it could be audio, video, pictures, text, and so on. In the past decades, a multitude of recommendation algorithms has been developed. They can be divided into two groups: history data-based recommendation (HDBR) methods and content-based recommendation (CBR) methods. The HDBR methods have been widely researched for recommendation systems. These methods only rely on the user's history data without requiring the details of such resources. Collaborative Filtering (CF) is one of the most distinguished approaches [2]. CF methods can be classified into two types:

1. Neighborhood-based method
2. Model-based methods

Model-based methods use L2 norm to normalize the solution [3]. HDBR methods require extensive historical data, which is difficult to obtain from the e-learning system and to achieve a decent performance, and they always suffer from the "cold start" problem. Thus, CBR methods may be a better choice for learning resources recommendation in e-learning systems. However, CBR methods characterize each user and item. Today learning resources come in various formats: audio, video, pictures, text, etc, and there is a huge amount of implicit information in learning resources that can be difficult to obtain, such as knowledge point, complexity, and prepared knowledge. Hence, content-based recommendation algorithm for learning resources is difficult to construct. The salient features of the approach employing collaborative learning and ensemble neural networks is described in the subsequent sections.

2. Collaborative Learning in Neural Networks

The training of deep neural networks, one must confront the challenges of general nonconvex optimization problems. Local gradient descent methods that most deep learning systems rely

*Corresponding author: nencysharma007.69@gmail.com on, such as variants of stochastic gradient descent (SGD), have no guarantee that the optimization algorithm will converge to a global minimum. It is known that an ensemble of multiple instances of a target neural network trained with different random seeds generally yields better predictions than a single trained instance. However, an ensemble may be computationally expensive at inference time. To keep the exact same computational complexity for inference, several of the common training techniques have been developed by adding additional networks in the training graph to boost accuracy without affecting the inference graph, including auxiliary training [4], multi-task learning [5], and knowledge distillation [6]. Auxiliary training is introduced to improve the convergence of deep networks by adding auxiliary classifiers connected to certain intermediate layers [7]. However, auxiliary classifiers require specific new designs for their network structures in addition to the target network. Furthermore, it is found late that auxiliary classifiers do not result in obvious improved convergence or accuracy. Multi-task learning is an approach to learn multiple related tasks simultaneously so that knowledge obtained from each task can be reused by the others [8]. However, it is not useful for a single task use case. Knowledge distillation is introduced to facilitate training for the neural network for a smaller transfer learning model. Artificial Intelligence and Machine Learning (AI &ML) are preferred techniques for analyzing large and complex data. Generally, artificial neural networks (ANN) are used for the implementation of artificial intelligence practically. The architecture of artificial intelligence can be practically implemented by designing artificial neural networks. The mathematical conversion of the ANN can be done by analyzing the biological structure of ANN. In the above example, the enunciated properties of the ANN that have been emphasized upon are:

1) Strength to process information in parallel way.
2) Learning and adapting weights
3) Searching for patterned sets in complex models of data.

To see how the ANN really works, a mathematical model has been devised here, to indicate the functions mathematically [7]. Here it is to be noted that the inputs of information parallel goes on into the input layer as specified whereas the end result analysis is marked from the output layer.

The feature of parallel acceptance and processing of data by the neural network serves a vital role. This ensures efficient and quicker mode of operation by the neural network. Also adding to it, the power to learn and adapt flexibly by the neural network aids in processing of data at a faster speed. These great features and attributes make the ANN self-dependent without requiring much intervention from humans. The output of the neural networks can be given by:

$$Y = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{X_i}.\mathrm{W_i} \quad + \quad \theta_i \qquad (1)$$

Here,
Y represents output
X represents inputs
W represents weights
Ө represents Bias

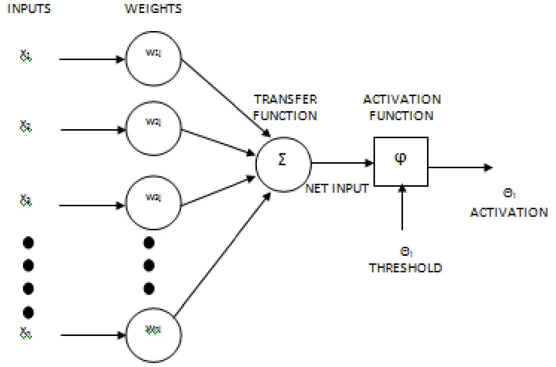


Fig. 1. Mathematical model of neural network

Preparing of ANN is of significant importance before it tends to be used to predict the outcome of the information inputs. Neural Networks can be used for a variety of different purposes, for example, pattern recognition in large and complex information pattern sets wherein the calculation of parameters would be extremely overwhelming for conventional factual techniques. The weights or the equivalents of experiences are evaluated and updated based on the information patterns which are fed to the neural networks for preparing. The framework of collaborative learning comprises of three significant parts: the generation of a populace of classifier heads in the preparation chart, the plan of the learning objective, and enhancement.

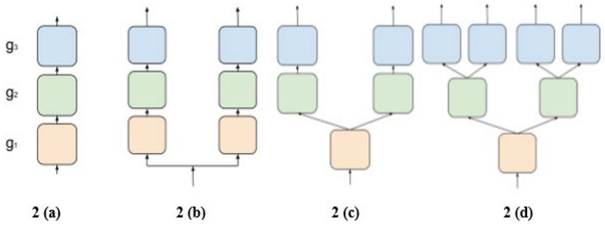


Fig. 2. (a) Target Network, (b) Multiple Instances, (c) Simple ILR Sharing, (d) Hierarchical ILRs haring

The figure above depicts the sub-categories of preparing. Like helper preparing [8], we add several new classifier heads into the first network chart during preparing time. At inference time, just the first network is kept and all added parts are discarded. Unlike helper preparing, each classifier head here has an identical network to the first one in terms of chart structure. This approach leads to advantages over helper preparing in terms of engineering effort minimization. To start with, it does not require to design extra networks for the helper classifiers. Second, the structure symmetry for all heads does not require extra different weights associated with misfortune functions to well balance injected backpropagation error flows, because an equal weight for each head's objective is ideal for preparing. Mathematically:

Assuming the target network to be trained in figure 2(a) is given by:

$$\boldsymbol{z} = g(x, \theta) \qquad (2)$$

Here,
g is determined by the graph architecture
θ represents the network parameters.

The term g can also be represented as the cascade of the following sub-nets, given mathematically by:

$$g(x,\theta) = g3(g2(g1(x1,\theta1),\theta2),\theta3) \qquad (3)$$

The cascade of the network is often termed as Ensemble Neural Network (ENN).

$$\text{Here,}\theta = [\theta1,\theta2,\theta3] \qquad (4)$$

In general, it is observed that that the preparation memory size is generally relative to the number of layers/operations. With the multi-instance pattern, the number of parameters in the whole preparation chart is relative to the number of heads. Clearly, ILR sharing can relatively reduce the memory utilization and speed up preparing, compared to multiple instances without sharing.

3. Resilient Back Propagation

The Back propagation (BP) neural network is the most famous among all the neural network applications. It enjoys the benefits of yielding high order exactness. However, viable applications are hard to be satisfied because of the problems of slow learning and the likelihood of being trapped into a nearby least especially when the size of the network is large. These problems are due to the way that the learning of BP neural network is mechanical and elementary. Numerous researchers have worked to overcome these problems, especially the nearby convergence [9]. Multilayer networks normally use sigmoid transfer functions in the hidden layers. These functions are often called "crushing" functions, because they compress an infinite info range into a finite result range. Sigmoid functions are characterized by the way that their slopes approach zero, as the info gets large. This causes a problem when you use steepest descent (gradient decent/back propagation) to prepare a multilayer network with sigmoid functions, because the gradient can have a tiny magnitude and, therefore, cause little changes in the weights and biases, even however the weights and biases are a long way from their ideal values.

The purpose of the resilient propagation (RPROP) preparing calculation is to eliminate the limits of these magnitudes of the partial derivatives. Just the indication of the derivative can determine the direction of the weight update; the magnitude of the derivative affects the weight update. Another most troublesome aspect of the back propagation learning was picking the correct preparation parameters. Resilient propagation does have preparing parameters, yet it is extremely rare that they need to be changed from their default values. This makes resilient propagation a very easy method for utilizing a preparation calculation. It likewise has the nice property that it requires just a modest increase in memory requirements. Also, resilient propagation is considerably more efficient than back propagation.

Resilient propagation, so, RPROP is one of the fastest preparation calculations available. The RPROP calculation simply refers to the direction of the gradient. It is a supervised learning method. It works much the same way to back propagation, except that the weight updates are done in a different manner. In back propagation the change in weight is calculated with the magnitude of the partial derivative:

$$\Delta w_{i,j}(t) = \alpha x_i(t).\delta_j(t) \qquad (5)$$

Here,
$\alpha$is the learning rate
$x_i$is the propagating to ith neuron at time ‘t’
$\delta_j$is the corresponding error gradient

This is however different in resilient back propagation. Resilient propagation, on the other hand, calculates an individual delta Δij, for each connection, which determines the size of the weight update. The following learning rule is applied to calculate delta:

$$\Delta_{i,j}^{t}= \begin{cases} \eta^{+}.\Delta_{i,j}^{(t-1)}\ if\ \frac{\partial E^{t-1}}{\partial w_{i,j}}.\frac{\partial E^{t}}{\partial w_{i,j}} > 0 \\ \eta^{-}.\Delta_{i,j}^{(t-1)}\ if\ \frac{\partial E^{t-1}}{\partial w_{i,j}}.\frac{\partial E^{t}}{\partial w_{i,j}} < 0 \\ \quad \Delta_{i,j}^{(t-1)}\ otherwise \end{cases} \qquad (6)$$

Here,
The update-value Δij evolves during the learning process based on the sign of the error gradient of the previous iteration $\frac{\partial E^{t-1}}{\partial w_{i,j}}$and the error gradient of the current iteration$\frac{\partial E^{t}}{\partial w_{i,j}}$.

Assuming the derivative retains its sign, the update value is somewhat increased by the element η+ to accelerate convergence in shallow regions. η+, is a steady normally with a value of 1.2. In the event that the derivative is 0 then we don't change the update-value. Once the update-value is calculated for each weight, the weight-update is then calculated. There are two rules to keep to calculate the weight-update. That's what the main rule is in the event that the current derivative and the previous derivative retain their signs, the accompanying equation is used to calculate the weight-update.

$$\Delta w_{i,j}^{t} = \begin{cases} -\Delta_{i,j}^{t}\ if\ \frac{\partial E^{t}}{\partial w_{i,j}} > 0 \\ +\Delta_{i,j}^{t}\ if\ \frac{\partial E^{t}}{\partial w_{i,j}} < 0 \\ \ 0\ otherwise \end{cases} \qquad (7)$$

The weight is updated as:

$$w_{i,j}^{t+1} = w_{i,j}^{t} + \Delta w_{i,j}^{t} \qquad (8)$$

If the current derivative is a positive value meaning the previous value was also a positive value (increasing error), then the weight is decreased by the update value.

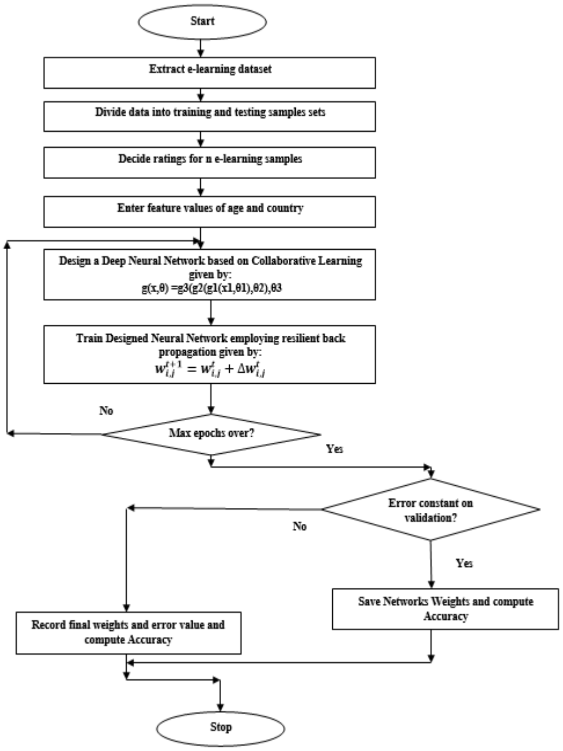

Start
Extract e-learning dataset
Divide data into training and testing samples sets
Decide ratings for n e-learning samples
Enter feature values of age and country
Design a Deep Neural Network based on Collaborative Learning given by:
g(x,θ) =g3(g2(g1(x1,θ1),θ2),θ3
Train Designed Neural Network employing resilient back propagation given by:
$w_{i,j}^{t+1} = w_{i,j}^{t} + \Delta w_{i,j}^{t}$
No
Max epochs over?
Yes
Error constant on validation?
No
Yes
Save Networks Weights and compute Accuracy
Record final weights and error value and compute Accuracy
Stop

Fig. 3. Flowchart of proposed system

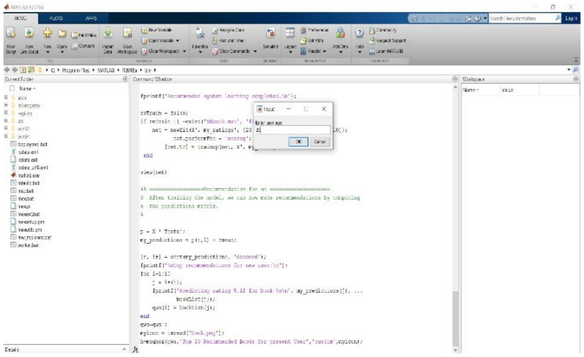
In general, it has been proved that the choice of this parameter isn't basic in any way. Even for a lot larger or smaller of this value, quick convergence is achieved. Δmax, is the greatest value a delta update, Δij, can have. This value is set to 50. Δmin, is the base value a delta update, Δij, can have. This is set to a very low positive value, 1e-6. The η-was given a value of 0.5. η-value is used as a reducing factor when the derivative has changed sign. This is normally a major leap, likely missing the base. Since it isn't known by how much the base was missed, it is a decent guess to halve the update-value by utilizing η-= 0.5. Then again, η+ must be large enough for quick development. However, in the event that it is too large a value, learning process can be disturbed. η+ was chosen as 1.2.

4. Results

The results are cited sequentially. To verify the effectiveness of the proposed recommendation algorithm, we have done several offline experiments on the Book-Crossing data set is collected by Cai-Nicolas Ziegler from the Book-Crossing virtual book community in 2004. According to the Book-Crossing data set, we supply the brief introductions of the books from Amazon In the data set, the rating scores marked by users are between 0 and 10. The higher the score, the higher the favorability.

The figure 4 depicts the command line screenshot to enter the age by the user which would be used as a parameter or feature for training the collaborative ensemble neural network.

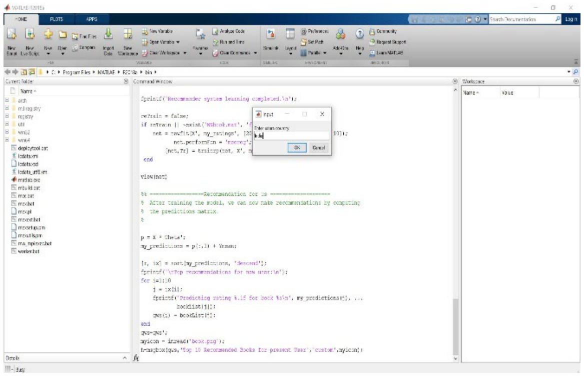
The figure 5 depicts the command line screenshot to enter the country by the user which would be used as a parameter or feature for training the collaborative ensemble neural network.



```
fprintf('Recommender system learning completed.\n');

reTrain = false;
if reTrain || ~exist('NNbook.mat', 'fi
    net = newfit(X', my_ratings', [20                    10]);
          net.performFcn = 'msereg';
       [net,tr] = trainrp(net, X', my_
 end

view(net)

%% =================Recommendation for us ==================
%  After training the model, we can now make recommendations by computing
%  the predictions matrix.
%

p = X * Theta';
my_predictions = p(:,1) + Ymean;

[r, ix] = sort(my_predictions, 'descend');
fprintf('\nTop recommendations for new user:\n');
for i=1:10
    j = ix(i);
    fprintf('Predicting rating %.1f for book %s\n', my_predictions(j), ...
            bookList{j});
    qws{i} = bookList{j};
end
qws=qws';
myicon = imread('book.png');
h=msgbox(qws,'Top 10 Recommended Books for present User','custom',myicon);
```

Enter user age
26

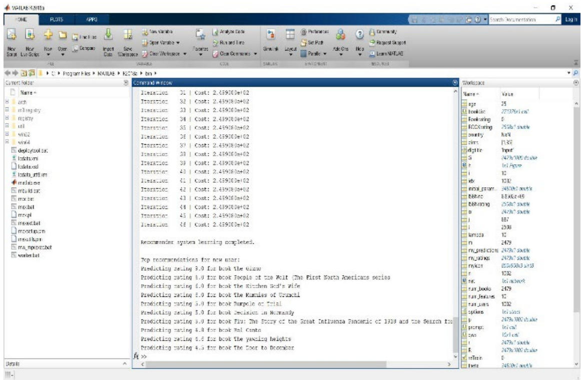
Fig. 4. Command line screenshot to enter age



```
fprintf('Recommender system learning completed.\n');

reTrain = false;
if reTrain || ~exist('NNbook.mat', 'f
    net = newfit(X', my_ratings', [20                    10]);
          net.performFcn = 'msereg';
       [net,tr] = trainrp(net, X', m
 end

view(net)

%% =================Recommendation for us ==================
%  After training the model, we can now make recommendations by computing
%  the predictions matrix.
%

p = X * Theta';
my_predictions = p(:,1) + Ymean;

[r, ix] = sort(my_predictions, 'descend');
fprintf('\nTop recommendations for new user:\n');
for i=1:10
    j = ix(i);
    fprintf('Predicting rating %.1f for book %s\n', my_predictions(j), ...
            bookList{j});
    qws(i) = bookList{j};
end
qws=qws';
myicon = imread('book.png');
h=msgbox(qws,'Top 10 Recommended Books for present User','custom',myicon);
```

Enter users country
India

Fig. 5. Command line screenshot to enter country



```
Iteration    31 | Cost: 2.459080e+02
Iteration    32 | Cost: 2.459080e+02
Iteration    33 | Cost: 2.459080e+02
Iteration    34 | Cost: 2.459080e+02
Iteration    35 | Cost: 2.459080e+02
Iteration    36 | Cost: 2.459080e+02
Iteration    37 | Cost: 2.459080e+02
Iteration    38 | Cost: 2.459080e+02
Iteration    39 | Cost: 2.459080e+02
Iteration    40 | Cost: 2.459080e+02
Iteration    41 | Cost: 2.459080e+02
Iteration    42 | Cost: 2.459080e+02
Iteration    43 | Cost: 2.459080e+02
Iteration    44 | Cost: 2.459080e+02
Iteration    45 | Cost: 2.459080e+02
Iteration    46 | Cost: 2.459080e+02

Recommender system learning completed.

Top recommendations for new user:
Predicting rating 9.0 for book The Client
Predicting rating 6.0 for book People of the Wolf (The First North Americans series
Predicting rating 6.0 for book The Kitchen God's Wife
Predicting rating 5.0 for book The Mummies of Urumchi
Predicting rating 5.0 for book Rumpole on Trial
Predicting rating 5.0 for book Decision in Normandy
Predicting rating 5.0 for book Flu: The Story of the Great Influenza Pandemic of 1918 and the Search fo
Predicting rating 4.8 for book Bel Canto
Predicting rating 4.6 for book The yawning heights
Predicting rating 4.5 for book The Door to December
>>
```

Fig. 6. Initialization of collaborative learning

The figure 6 depicts the implementation of collaborative learning based on input parameters.

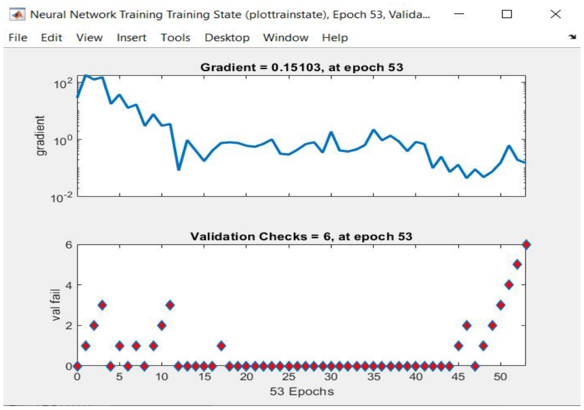
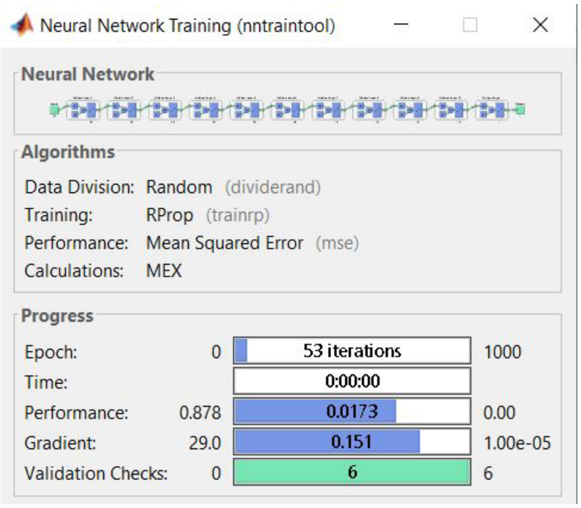


Fig. 7. Training states

The figure above depicts the training states of the ensemble neural network designed for recommendation. The gradient and the validation checks are shown.



Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
Training: RProp (trainrp)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	53 iterations	1000
Time:		0:00:00	
Performance:	0.878	0.0173	0.00
Gradient:	29.0	0.151	1.00e-05
Validation Checks:	0	6	6

Fig. 8. Training parameters

The figure 8 depicts the training parameters of the designed neural network. The neural architecture with a 10 hidden layer deep learning is shown.

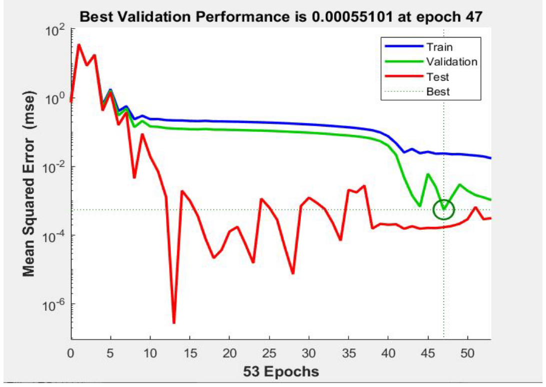


Fig. 9. Mean Square Error variation

The figure 9 depicts the variation of the mean square error as a function of iterations.

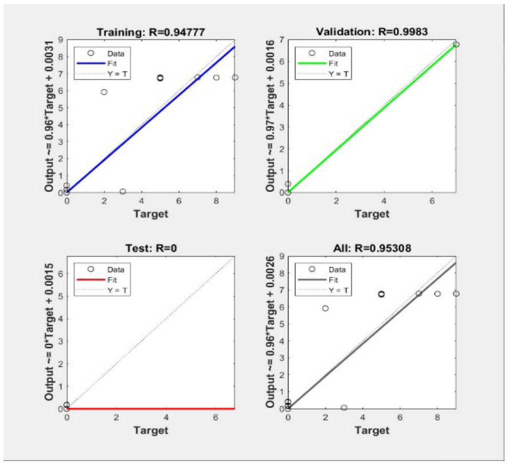


Fig. 10. Regression analysis

The figure 10 represents the regression of the proposed system which is 0.95 (approx.) for the overall average.

A comparison with previous work [1] clearly indicates that the proposed system outperforms the previously existing system in terms of mean square error and iterations. A comparative analysis ensues.

Table 1
Comparative analysis

Parameter Value	Previous Work	Proposed Work
MSE	3.3841	0.0173
Iterations to stability	400	53
Neural Network Category	CNN	Ensemble with Resilient Back Propagation

5. Conclusion

It tends to be concluded from the previous conversations that with more candidates picking e-learning applications due to its specific advantages, it has become necessary to design an optimized recommendation system for learning resources. One of the significant challenges is however the plethora of resources to choose from. The proposed work embodies the use of the age, nation and personal interests of the users to design the recommendation system. The technique used is the ensemble neural network with collaborative learning. The preparation calculation used is the resilient back propagation. It has been shown that the proposed approach outperforms the previously existing systems in terms of error, exactness and time complexity as a function of preparing iterations.

References

[1] Jiangbo Shu, Xiaoxuan Shen, Hai Liu, Baolin Yi, Zhaoli Zhang, "A content-based recommendation algorithm for learning resources", Springer 2018.

[2] Moreno, M.N., Segrera, S., López, V.F., Muñoz, M.D., Sánchez, Á.L., "Web mining-based framework for solving usual problems in recommender systems. A case study for movies′ recommendation," Neurocomputing 176, 72-80 (2016).

[3] Kohavi, R., "A study of cross-validation and bootstrap for accuracy estimation and model selection," In: proceeding International Joint Conference on Artificial Intelligence, pp 1137–1143 (1995).

[4] Mao, K., Chen, G., Hu, Y., Zhang, L., "Music recommendation using graph-based quality model," Signal Process. 120, 806–813 (2016).

[5] Horsburgh, B., Craw, S., Massie, S., "Learning pseudo-tags to augment sparse tagging in hybrid music recommender systems," Artif. Intell. 219, 25–39 (2015).

[6] Wang, Y., Shang, W., "Personalized news recommendation based on consumers' click behavior," in: Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference, 2015, pp. 634–638.

[7] Shi, B., Ifrim, G., Hurley N, "Learning-to-Rank for Real-Time High-Precision Hashtag Recommendation for Streaming News," in: Proceedings of the 25th International Conference on World Wide Web, 2016. International World Wide Web Conferences Steering Committee, pp. 1191–1202.

[8] Zhao, X., Yuan, J., Wang, M., Li, G., Hong, R., Li, Z., Chua, T. S., "Video recommendation over multiple information sources," Multimed. System 19(1), 3–15 (2013).

[9] Pyo, S., Kim, E., Kim, M., "Automatic and personalized recommendation of TV program contents using sequential pattern mining for smart TV user interaction," Multimed. System 19(6), 527–542 (2013).

[10] Kaššák, O., Kompan, M., Bieliková, M., "Personalized hybrid recommendation for group of users: top-N multimedia recommender," Inf. Process. Manag. 52(3), 459–477 (2016).

[11] Zhao, W., Wu, R., Liu, H., “Paper recommendation based on the knowledge gap between a researcher’s background knowledge and research target,” Inf. Process. Manag. 52(5), 976–988 (2016).

[12] Polatidis, N., Georgiadis, C.K., “A multi-level collaborative filtering method that improves recommendations,” Expert Systems with Applications 48 (2016): 100-110.

[13] Koren, Y., Bell, R., “Advances in collaborative filtering,” in Recommender systems handbook, pp. 77–118. Springer, Heidelberg (2015).

[14] Salah, A., Rogovschi, N., Nadif, M., “A dynamic collaborative filtering system via a weighted clustering approach,” Neuro-computing 175, 206–215 (2016).

[15] Pan, W., Xia, S., Liu, Z., Peng, X., Ming, Z., “Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks,” Inf. Sci. 332, 84–93 (2016).

[16] Bell, R.M., Koren, Y., “Scalable collaborative filtering with jointly derived neighborhood interpolation weights,” in: Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference, 2007, pp. 43–52.

[17] Ignatov, D.I., Nikolenko, S.I., Abaev, T., Poelmans, J., “Online recommender system for radio station hosting based on information fusion and adaptive tag-aware profiling,” Expert System Appl. 55, 546–558 (2016).