NON ITERATIVE NEURAL NETWORKS CLASSIFICATION AND REGRESSION

Poornima M Chabbi (2021488990)
University of Texas at Dallas
Dallas, USA
pxc190007@utdallas.edu

Shivvrat Arya (2021459028)
University of Texas at Dallas
Dallas, USA
sxa180157@utdallas.edu

ABSTRACT

This paper presents Extreme Learning Machine Neural Network and Random Vector Functional Link Neural Networks which use non iterative methods. These methods use matrix inversion to find unknown weights between hidden layer and output layer. Extreme learning machines are feed-forward non iterative neural networks in which the hidden nodes are random projections of inputs which can be calculated by multiplying inputs by random values (random weight values) giving the unknown weights between hidden and output layer. Random Vector Functional Link neural network are feed-forward non iterative neural networks in which the hidden layer and the input layer is merged and then the matrix inversion method is applied on it in order the unknown weights between hidden and output layer. Both the algorithms performance was compared with other classification and regression algorithms such as KNN Algorithm, Decision Tree algorithm, Logistic Regression and Naive Bayes algorithm. The accuracy obtained from both the algorithms was close to the state of art and time taken was very less when compared to other algorithms.

Index Terms-

Extreme Learning Machine Neural Network, Random Vector Functional Link Neural Networks, pseudo-inversion method.

1 Introduction

In this paper, we use the method of pseudo inversion for non-iterative neural network based classification and regression. The pseudo inversion method is used to find the unknown weights between hidden layer and the output layer. It is useful when there are large number of unknowns and this method consumes less time to find the unknowns. Extreme Learning Machine Neural Network is a feed-forward neural network that follows the above method. The weights between input layer and hidden layer are random projection of inputs which can be obtained by multiplying inputs with random values. These weights don't change for a given model and a dataset. Using these weights and the values of the output nodes, we can obtain the unknown weights between hidden and output layer.Random Vector Functional Link Neural Networks is a feed-forward neural network which also uses the pseudo inversion method. In this type of neural network, the hidden nodes are merged with the input nodes forming one single layer. Pseudo inversion method is used to obtain weights between this layer and the output layer.

2 Problem Definition and Algorithm

2.1 Problem Statement

To design a method for classification and regression using Random Vector Functional Link and Extreme Learning Machine neural network architecture for a given dataset and evaluating using predefined evaluation metrics.

2.2 Algorithm Definition

Pseudo-inversion method

This method is used to obtain the inverse of a given matrix. Consider matrix X, the pseudo inverse of matrix X is:

$$x^{+} = (x^{T} \bullet x)^{-1} x^{T} \tag{1}$$

Extreme Learning Machine Neural Network

The weights between the input layer and hidden layer are random assignments and the weights matrix H is obtained. These weights obtained remain constant for a given dataset. The weights W between the hidden layer and the output layer can be obtained from dot product of pseudo inversion of H and output label Y. That is:

$$W = (H^{T} \bullet H)^{-1} H^{T} \bullet Y \tag{2}$$

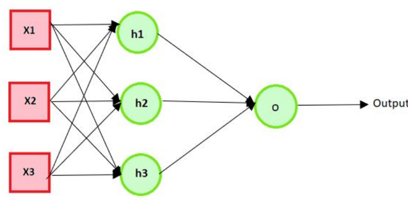

Figure 1: Extreme Learning Machine neural network, x1, x2 and x3 are input nodes, h1, h2 and h3 are hidden nodes and o is output.

Random Vector Functional Link Neural Networks

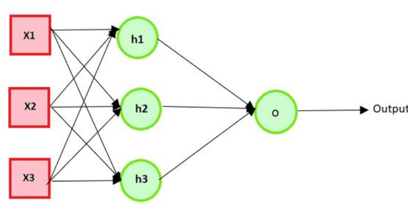

Figure 2: Random Vector Functional Link, x1, x2 and x3 are input nodes, h1, h2 and h3 are hidden nodes and o is output.

The weights between the input layer and the hidden layer are random assignments and weights matrix H is obtained. Now, the input nodes are concatenated to the hidden nodes forming a new hidden layer. Now, the weights between the new hidden layer and the output layer is obtained from dot product of pseudo inversion of H and output label Y. That is:

$$W = (H^{T} \bullet H)^{-1} H^{T} \bullet Y \tag{3}$$

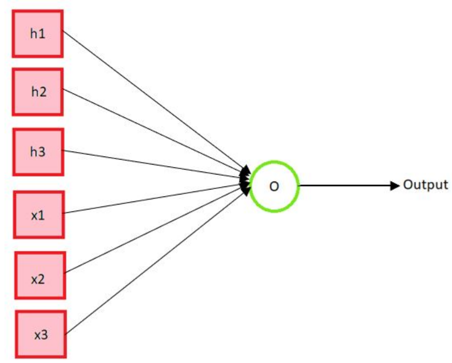

Figure 3: Random Vector Functional Link, h1, h2, h3, x1, x2 and x3 are merged to form a new layer and o is output node.

3 Experimental Evaluation

3.1 Methodology

The following datasets were used for evaluation of the algorithms:

Datasets for classification:

- SONAR dataset[5]
- Banknote Authentication dataset[5]
- Ionosphere dataset[6]
- HTRU dataset[2]

Datasets for regression:

- Airfoil Self-Noise dataset[5]
- Abalone dataset[5]
- Wine_quality dataset[3]

The following metrics are used for evaluation of the algorithms:

The evaluation metrics used for classification:

1. Accuracy: it is the portion of true results among the total number of cases tested.
   $accuracy = \frac{(TruePositive+TrueNegative)}{(FalsePositive+FalseNegative+TruePositive+TrueNegative)}$
2. Precision: it is the portion of results that were predicted to be positive and are positive.
   $Precision = \frac{(TruePositive)}{(FalsePositive+TruePositive)}$
3. Recall: the portion of actual positive that was correctly classified.
   $Recall = \frac{(TruePositive)}{(FalseNegative+TruePositive)}$
4. F1 score: It is the mean of precision and recall. Also, F1 score is a number between 0 and 1.
   $F1 = 2\frac{(precision*recall)}{(precision+recall)}$

The evaluation metrics used for regression:

1. Mean squared Error: Consider a dataset with n data points on all variables, vector of n predictions is generated from it. Let the observed values of variables being predicted be vector yi and $\overline{yi}$ be the predicted values, then MSE of the predictor is:
$MSE = \frac{1}{n}\sum_{1}^{n}(yi - \overline{yi})^2$
2. Mean Absolute Error: Consider a dataset with n data points and (X,Y) be a pair exhibiting the same phenomenon. Then mean absolute error will be:
$MAE = \frac{\sum_{1}^{n}|yi-xi|}{n}$

3.2 Result

The results of classification on the above datasets:

Sonar	**Accuracy**	**Precision**	**Recall**	**F1_Score**	Threshold Value	Activation Function
RVFL	0.9523809524	1	0.9	0.947368421	0.3	tanh
ELM	0.9523809524	1	0.9	0.947368421	0.3	tanh
KNN	0.7857142857	0.8823529412	0.68181818	0.7692307692	-	-
Decision Tree	0.7380952381	0.8235294118	0.63636364	0.7179487179	-	-
Logistic Regression	0.8095238095	0.9375	0.68181818	0.7894736842	-	-
NaiveBayes	0.7142857143	0.7083333333	0.77272727	0.7391304348	-	-

Table 1: Comparision of evaluation metrics for all the algorithms for sonar dataset

Banknote Authentication	**Accuracy**	**Precision**	**Recall**	**F1_Score**	Threshold Value	Activation Function
RVFL	0.9963636364	0.992	1	0.9959839357	0.5	tanh
ELM	0.9963636364	0.992	1	0.9959839357	0.5	tanh
KNN	1	1	1	1	-	-
Decision Tree	0.978182	0.97561	0.97561	0.97561	-	-
Logistic Regression	0.814545	0.86	0.699187	0.7713	-	-
NaiveBayes	0.989091	0.991803	0.98374	0.987755	-	-

Table 2: Comparision of evaluation metrics for all the algorithms for banknote dataset

Ionosphere	**Accuracy**	**Precision**	**Recall**	**F1_Score**	Threshold Value	Activation Function
RVFL	0.90140845	0.91228070	0.9629629	0.93693693	0.6	tanh
ELM	0.94366197	0.94642857	0.9814814	0.96363636	0.6	tanh
KNN	0.943662	0.923077	1	0.96	-	-
Decision Tree	0.929577	0.921569	0.979167	0.949495	-	-
Logistic Regression	0.943662	0.958333	0.958333	0.958333	-	-
NaiveBayes	0.971831	0.979167	0.979167	0.979167	-	-

Table 3: Comparison of evaluation metrics for all the algorithms for ionosphere dataset

HTRU	Accuracy	Precision	Recall	F1_Score	Threshold Value	Activation Function
RVFL	0.97541899}	0.8584615385	0.861111	0.8597842835	0.3	tanh
ELM	0.97262569	0.9632653061	0.726153	0.8280701754	0.3	tanh
KNN	0.970391	0.894558	0.778107	0.832278	-	-
Decision Tree	0.979888	0.934641	0.846154	0.888199	-	-
Logistic Regression	0.942179	0.642082	0.87574	0.740926	-	-
NaiveBayes	0.967877	0.834835	0.822485	0.828614	-	-

Table 4: Comparison of evaluation metrics for all the algorithms for HTRU dataset

The results of regression on the above datasets:

airfoil_self_noise	MSE	MAE
RVFL	26.93218598	3.697649876
ELM	21.83814594	3.650994113
Decision Trees	6.287081757	1.85210299
SVM	43.16346265	5.205110557
Linear Regression	24.65444383	3.884846094
Randon Forest	3.321738251	1.393496013

Table 5: Comparison of evaluation metrics for all the algorithms for airfoil dataset

abalone	MSE	MAE
RVFL	5.019985573	1.583022702
ELM	5.631542024	1.617825786
Decision Trees	6.005048	1.683615
SVM	6.177883	1.713278
Linear Regression	6.814436	1.722998
Randon Forest	10.08373	2.200957

Table 6: Comparison of evaluation metrics for all the algorithms for abalone dataset

wine_quality	MSE	MAE
RVFL	1.066437856	0.6245330424
ELM	1.372869122	0.6385503607
Decision Trees	0.608642	0.612474
SVM	0.440622	0.480102
Linear Regression	0.605131	0.559476
Randon Forest	0.652041	0.460204

Table 7: Comparison of evaluation metrics for all the algorithms for wine_quality dataset

3.3 Discussion

The performance of Extreme Learning Machine algorithm and Random Vector Functional Link algorithm was compared with the performance of KNN Algorithm, Decision Tree algorithm, Logistic Regression and Naive Bayes algorithm on various datasets. Time taken by both Extreme Learning Machine algorithm and Random Vector Functional Link algorithm was very less when compared to that of other algorithms. For classification the accuracy obtained from Extreme Learning Machine algorithm and Random Vector Functional Link algorithm was better than other algorithms.For regression, the mean squared error and mean absolute error was comparable with the other algorithms.

4 Related work

[1] In this paper, the multi-layers Extreme Machine Learning (EML)/ Random Vector Functional Link Network (RVFL) on four datasets of handwritten characters. The impact of the architecture handwritten characters and the robustness of the distribution of the results across different runs. It also shows that a maximum combination rule provides an accuracy of 98.03% for Bangla, 95.97% for Arabic digits, 96.30% for Oriya digits and 98.64% for Devnagari. The accuracy improves with the increasing in the size of the hidden layers and state-of-the-art is reached. But after the hidden layers reaches a certain size the performance doesn't improve.[1]

[2] This paper proposes multi-classifier extreme learning machine (MT-ELM) which is generalization for multiple tasks. This emerging neural network architecture offers fast learning. This architecture does not reconfigure the network to performs multiple classification tasks. The MNIST dataset is used to validate this design, where the accuracy of the model for classifying numbers in the MNIST dataset was 91.7% and for categorizing number parity it was 90.35%. The power dissipated from this design which is synthesized on a TSMC-65nm technology node was 13.6 mW for a network with 12 output neurons and 80 hidden neurons.[7]

[3] This paper presents an algorithm to detect the wind power ramp in certain forecasting horizon. To predict the wind power, ramp random vector functional link (RVFL) network is employed. The real-world wind power data is used to evaluate the forecasting method. This method performs better than the Artificial Neural Network. And it has comparable performance with Support Vector Machine and Random Forest. The time taken for training and testing is in favour of the RVFL neural network. The results of this model are compared by non parametric Wilcoxon signed rank test and show if there exists significant difference between filters. The best results were obtained from the combination of RVFL classifier with F-score approach. This model has higher accuracy and is most efficient that the actual feature space.[4]

5 Future work

• The weights between the input nodes and the hidden nodes are randomly assigned. Cross validation can be done and the model that best fits the given dataset can be obtained.

• It can be used further for multi-label classification.

• Other inversion techniques can be used on these algorithms and their performance can be evaluated.

6 Conclusion

Extreme Learning Machine algorithm and Random Vector Functional Link algorithms use the pseudo-inversion method for obtaining the weights between the hidden nodes and the output nodes. In this paper, the performance of both the algorithms was compared with KNN Algorithm, Decision Tree algorithm, Logistic Regression and Naive Bayes algorithm using predefined performance metrics. It was observed that the time taken by both the algorithms was very less compared to other algorithms. Since non-iterative pseudo-inversion method is used for obtaining the weights between the hidden nodes and the output node, the algorithms perform faster than other algorithms that use iterative method. This is because other algorithms use the iterative method which take a long time to converge whereas pseudo-inversion method is one-step method. For both the algorithms the highest accuracy of 99.63% was obtained on Banknote Authentication dataset when compared to other algorithms. The Mean squared error of 5.01 and Mean absolute error of 1.58 was obtained on abalone dataset. This method can be used for classification in applications with large labeled instances.

References

[1] Hubert Cecotti. "Deep Random Vector Functional Link Network for handwritten character recognition". In: *2016 International Joint Conference on Neural Networks (IJCNN)* (2016), pp. 3628–3633.

[2] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

[3] Yeşim Er and Ayten Atasoy. "The Classification of White Wine and Red Wine According to Their Physicochemical Qualities". In: *International Journal of Intelligent Systems and Applications in Engineering* (Dec. 2016), pp. 23–23. DOI: `10.18201/ijisae.265954`.

[4] F. Li, T. Lung, and C. Yeh. “Comparison of filter approaches based on RVFL classifier”. In: *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 4. Aug. 2010, pp. 1520–1524. DOI: 10.1109/FSKD.2010.5569333.

[5] Ryan A. Rossi and Nesreen K. Ahmed. “The Network Data Repository with Interactive Graph Analytics and Visualization”. In: *AAAI*. 2015. URL: http://networkrepository.com.

[6] V G Sigillito et al. “Classification of radar returns from the ionosphere using neural networks”. In: *Johns Hopkins APL Tech. Dig* vol. 10 (1989). in, pp. 262–266.

[7] A. M. Zyarah and D. Kudithipudi. “Extreme learning machine as a generalizable classification engine”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. May 2017, pp. 3371–3376. DOI: 10.1109/IJCNN.2017.7966279.