International Journal of Emerging Trends in Science and Technology
8 (2) December 2022, 23-27
http://www.publishingindia.com/ijetst

Efficient ANN Topologies for Economic Load Dispatch – An Experiment

Umapathi Krishnamoorthy
Associate Professor, Department of Electrical and Electronics Engineering, M.Kumarasamy College of Engineering, Karur, Tamil Nadu, India.

Abstract --A proven truth is that most complex non-linear problems can be effectively solved by using Artificial Neural Networks. And hence this paper addresses the economic load dispatch in the power distribution sector of electrical engineering. There are 'N' number of quantities that affect the load dispatch and thus its optimization demands laborious training to achieve good results. This paper studies the prediction accuracy of three different ANN topologies on a 3-source 5-bus system. Topology used includes Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) and Deep Feed Forward (DFF) networks and their corresponding results were portrayed.

Keywords --Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Deep Feed Forward (DFF), Recurrent Neural Network (RNN), Economic Load Dispatch (ELD), Artificial Neural Network, Power System.

I. INTRODUCTION

Artificial Intelligence imparts human like behavior and Artificial Neural Network (ANN) are designed to function like human nervous system. Artificial Neural Networks have proven to be efficient in studying complex non-linear problems and solving real-world problems which were difficult to solve optimally using conventional approaches based on mathematical theories.

ANN is formed from artificial neurons which hold the values of the data provided and then multiplied with some randomly initialized values which are known as weights. Each neuron from the input layer is multiplied with the randomly initialized vector of weightsw(i,j), where i denotes the neuron multiplied with the weight and j denoted the neuron which is finally produced as the byproduct of multiplication and then the activation function is applied to the multiplied result and the obtained result is the neuron of the first hidden layer. The first hidden layer neurons get through the same procedure to develop the neurons of the second hidden layer and finally, the last hidden layers make the neurons of the output layer which holds the predicted value of the model. The predicted value is then compared with the actual value and the loss function is decided for the proposed problem which gets reduced afterevery training iteration till it reaches a certain accepted value and then the testing is done.

Economic Load Dispatch ELD, is a complex non-linear problem of optimal distribution of generated power from the given generators in accordance with their maximum and minimum power limits and the balance of the power equation. The solution of the ELD problem is affected by numerous parameters involved such as the generation constraints, the reactance and the resistance of lines involved, and the losses that occurred in the operation of a grid system. Problems come in a variety of datasets, some are easy to capture in certain equations, some are sequential where there is a hidden sequence that needs to be predicted for future use. Numerous topologies are proposed in ANN for solving a variety of problems encountered[1] . Some of the most effective proposed network topologies for numeric data are LSTM (Long Short-Term Memory)[2], GRU (Gated Recurrent Unit)[3] and DFF (Deep Feed Forward).

DFF is a complex implementation of simple Feed Forward in which hidden layers are added for better fitting on the training data for the power system. Hidden layers are added for the improvement in the predictions.

GRU is the advanced implementation of standard Recurrent Neural Network, RNN [4] where two gates are applied for fixing the problem of vanishing gradients. Update gate is used to retrieve old sequence information and reset gate is applied to discard irrelevant information takes more time for training as it is a more complex model as compared to the standard RNN which is not equipped with intelligent gates to control the flow of information.

LSTM[2, 4] is another advancement done in standard RNN with the applications of 3 gates which gives the LSTM cell tight control over the flow of information. The three gates are named the forget gate, input gate, and output gate. The forget gate is to decide which information is relevant to keep from the prior steps and the input gate is to decide which information is relevant to add to the memory from the current step and the last output gate is to decide what will be the next hidden state. Together all three make a strong network for capturing hidden sequences with even better accuracy than the GRU. However, GRU being less complex takes less time than the LSTM and hence used in relatively small datasets where fast predictions are preferred over accurate ones.

In this paper, a 3-generator, 5- bus power system is considered along with its generating capacity constraints. Three different topologies of neural networks (DFF, LSTM, and GRU) are implemented on the power system for the optimum solution of the economic load dispatch problem. Their results are compared on the mean squared error loss function and number of epochs to converge the training loss for each model. The training time taken for each epoch is about 4 times in LSTM and GRU in comparison with DFF. Results are plotted for all three topologies to validate the bestsuited topology on the tabular dataset of the proposed ELD problem. Learning curves are plotted for the comparison of the learning of DFF, GRU, and

LSTM on G1, G2, and G3 by studying the overlapping of predicted values and the actual values graphs of all three generators named G1, G2, and G3.

II. PROBLEM DEFINITION

Economic Load Dispatch is the optimal distribution of the generation power to the generating units present in the power system while satisfying the load power demands[5] .

A. Datasetfor the Neural Network

The proposed problem is based on the dataset of 5 buses and 3 generator systems in which loads on 5 buses are fed as an input and optimal generation power for all three generators as an output. The dataset consists of 4204 data points which are split in the ratio of 80-20,80 percent data is for training and the rest 20 percent is for testing. The dataset contains varying instances of load distribution on all the buses. These variations will help to build a more robust and precise ANN model for the economic load dispatch problem on the described power system.

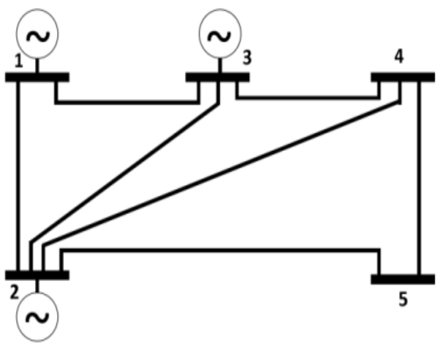

Figure 1. 3-Generator 5-Bus System under consideration

Table 1. Bus and Line data assumed for the problem under study

Bus Code	Assumed Voltage Magnitude (p.u.).	Assumed Angle Degree	Line Code	Line Impedances		Line Charging Y/2 (p.u.)
				R(p.u)	X(p.u.)	
1	1.06	0.0	1-2	0.02	0.06	0.030
2	1.045	0.0	1-3	0.08	0.24	0.025
3	1.03	0.0	2-3	0.06	0.18	0.020
4	1.00	0.0	2-4	0.06	0.18	0.020
5	1.00	0.0	2-5	0.04	0.12	0.015
-	-	-	3-4	0.01	0.03	0.010
-	-	-	4-5	0.08	0.24	0.025

B. *Balancing of Power*

The total power generated must be equal to the submission of total power demanded and the overall transmission losses involved which can be described in the equation:

$$P_G = P_D + P_L \tag{1}$$

$$P_L \leq P_G \leq P_D \tag{2}$$

P_G: Generation Power ;P_D: Total Demand Power ; P_L: Transmission Losses

Transmission line losses are given by,

$$P_L = \sum_{i=1}^{ng} \sum_{j=1}^{ng} P_i B_{ij} P_j + \sum_{i=1}^{ng} B_{oi} P_i + B_{00} \tag{3}$$

C. *GENERATOR LIMITS*

All the three generators named G1, G2, and G3 have their limits on which they can operate without the occurrence of any fault in the system. Limits are provided as: G1:10-85 MW G2:10-80 MW G3:10-70 MW.

III. ANN TOPOLOGIES IMPLEMENTED

ANN comes in a number of topologies to deal with a variety of problems available. Some neurons are simply feed-forward, some are having the feedback structure to adhere to the sequence generated. Three topologies are summarized below for better understanding:

A. *Deep Feed Forward (DFF):*

Deep Feed-Forward is the cumulative developed form of multiple hidden layers in a simple feed-forward neural network. They were developed in the early '90s. They are having more than one hidden layer which makes them different from the simple feed-forward networks which further makes them a better fit in learning core patterns in data for acceptable accuracy. Insights of applied DFF are elaborated below:

B. *DFF Model*

In this paper, one model is applied to each generator. So, 3 models of DFF are trained on the developed dataset. A summary of each model applied is provided below:

No of units in input layer: 5 neuron cells each containing the given load on the assigned bus .

No of hidden layers added: 6

No of neurons on each hidden layer: 50

Loss function: Mean Squared Error

Epochs: 500

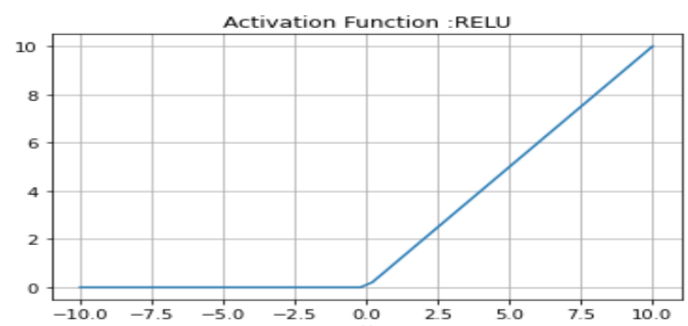

Figure 2. ReLU Activation Function.

Activation function used: ReLU [10] ReLU can be described as the Rectified Linear Activation Function[11] which is the piecewise linear function which outputs input directly if the input given is positive, otherwise, it gives 0 as the output and thus solved the problem of vanishing gradient which occurs in the case of other exponential activation functions The mathematical equation of ReLU :

$$Y = \max(0, x) \tag{4}$$

Optimizer applied: Adam [12] Adam optimizer can be described as the replacement algorithm for the optimization in place of stochastic gradient descent for the training of deep learning models.

C. Long Short-Term Memory (LSTM)

LSTM is the advanced form of standard recurrent networks which can learn the order dependencies and the sequence present in the problem for providing better predictions than the standard model. It uses 3 gates for controlling the flow of information named as forget gate, input gate, and output gate. It is more complex, thus takes more time to train in comparison with the standard model but provides much better predictions.

a) LSTM MODEL APPLIED

In this paper, one model is applied on each generator. So, 3 models of LSTM are trained on the developed dataset. A summary of each model applied is provided below:

No of units in input layer: 5 neuron cells each containing the given load on the assigned bus

No of hidden layers added: 6

No of neurons on each hidden layer: 40

Loss function: Mean Squared Error

Epochs: 500

Activation function used: Tanh and Sigmoid

Tanh activation function [10] is used in the regulation of the values passed in the LSTM network. This function ensures that the values only remain between 1 and -1 or particularly it normalizes the values passed inside the network in the range of [-1,1].

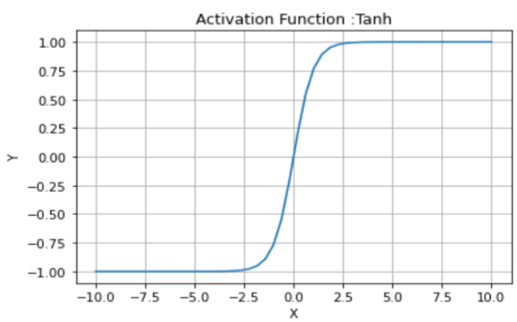

Figure 3.a. Tan activation function

Sigmoid [10] is the same as the tan in core functionality as it also ensures that the value remains in a particular range which is [0,1].These activation functions are useful in the case of dominant values which are so large that make other values insignificant and lead to overfitting.Optimizer applied: Adam LSTM gates: Input gate, Output gate and Forget gate Forget gate [13] (f_t) is the deciding gate whether the information has to be passed on or leave behind.The information coming from the hidden state and the current state is first passed through the sigmoid activation (σ_g) and then if the output is nearer to 0, information is irrelevant and if nearer to 1, information is relevant enough to pass on.

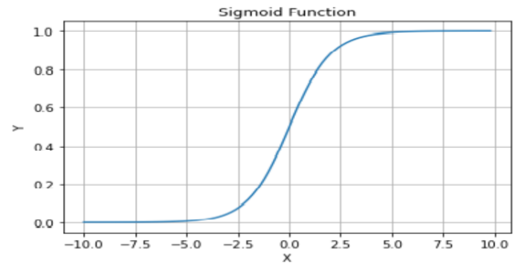

Figure 3.b. Sigmoid activation function

$$f_t = \sigma_g(W_f[h_{t-1}, x_t] + b_f \quad (5)$$

Input gate [13] (i_t) is useful for updating the cell state. The previous hidden state and the current input state are passed through the sigmoid function which decides which values are important to pass on further, then the previous hidden state and the current state are also passed through the tanh function which squishes the values in the range [-1,1] for regulation purposes. Finally, tanh and the sigmoid output get multiplied and the sigmoid output decides the relevant information from the available tanh output.

$$i_t = \sigma_g(W_i[h_{t-1}, x_t] + b_i \quad (6)$$

Cell state is the state of the passed information which gets updated after each training session. For the update process, the previous cell state gets point-wise multiplied to the forget vector from which some values may be dropped in case of multiplication with the values which are nearer to 0 and then the point-wise addition of output from the input gate and the output of the point-wise multiplication gives the new relevant cell state. Output gate[13] (σ_t) is the deciding entity for the next hidden state. The previous hidden state and the current state are first passed through the sigmoid function and the new derived cell state is passed through the tanh function. Finally, the multiplication of the sigmoid output and the tanh output makes the content of the new hidden state which is used for predictions.

$$o_t = \sigma_g(W_o[h_{t-1}, x_t] + b_o \quad (7)$$

$$c_t = tanh_c(W_c X_t + U_c h_{t-1} + b_c \quad (8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c_t \quad (9)$$

$$h_t = O_t \odot \sigma_g(c_t) \quad (10)$$

D. Gated Recurrent Unit (GRU)

GRU was a simplified modification introduced in 2014 to the LSTM. GRU is able to solve the vanishing gradient problem arising in RNN using only two gates- update and reset gate[14] .These gates are responsible for passing the previous information to the current state using sigmoid and tanh activation functions[15] .

a) Gru Model Applied

In this paper, one model is applied on each generator. So, 3 models of GRU are trained on the developed dataset A summary of each model applied is provided below:

No of units in input layer: 5 neuron cells each containing the given load on the assigned bus
No of hidden layers added: 6
No of neurons on each hidden layer: 50
Activation function used: Tanh and Sigmoid
Optimizer applied: Adam
Loss function: Mean Squared Error
Epochs: 500
GRU gates: Update gate and Reset gate

Update gate[16] (z_t) is responsible to keep only the relevant part of the previous information for the problem. It uses the sigmoid function to calculate the relevancy of previous information [13] .

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (11)$$

Reset gate[16] (rt) forgets the irrelevant part of the previous information. The sigmoid function helps in measuring the relevancy of the information on a scale of 0 to 1.

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (12)$$

The current memory (h_t) state is calculated using the tanh activation function on the results from reset and update gates. Finally, the results from the update and reset gates are multipliedwith the previous and current memory state to calculate the current unit and pass it on for future predictions.

$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (13)$$

$$h_t = \tanh(z_t \odot h_{t-1} + (1 - z_t) \odot h'_t) \quad (14)$$

IV. SIMULATION RESULTS

All the three topologies were applied to the proposed dataset and then training was started with a high loss which decreases with each iteration. The learning curve and comparison curve between prediction and actual values were studied for all three generating units for each ANN architecture. Finally, comparison curves were plotted for the final comparison of all three topologies. We used the Keras library to build and run all the sequential ANN models [17].

A. Performance Comparison

Training curves of all architectures are studied for all three generators named G1, G2, G3. All three architectures have converged to acceptable accuracy in a lesser number of iterations. DFF has performed slightly better than GRU and LSTM on all three generators irrespective of the fact that the other two architectures are modern and complex.

Figure 4. MSE loss on test dataset for the 3 models

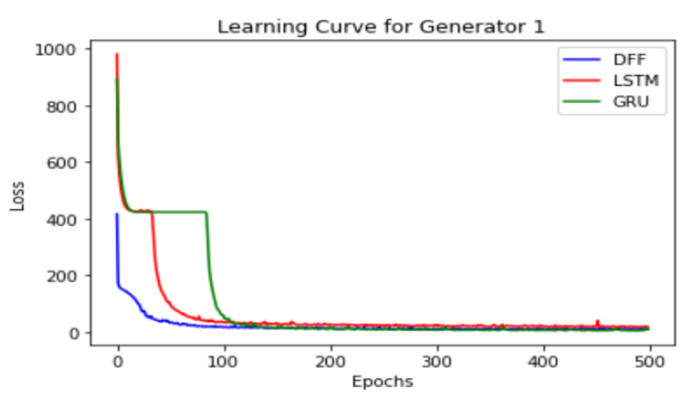

Figure 5. Loss Vs Epoch for Generator1

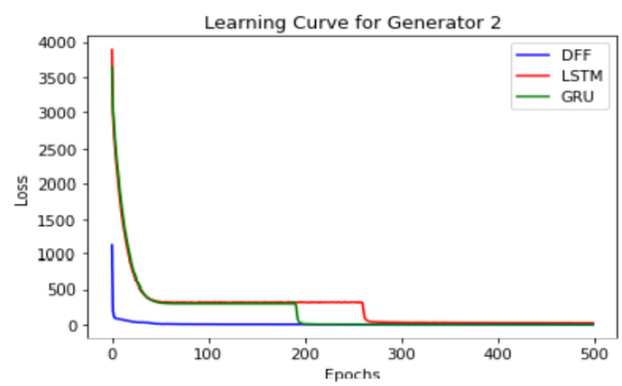

Figure 6. Loss Vs Epoch for Generator2

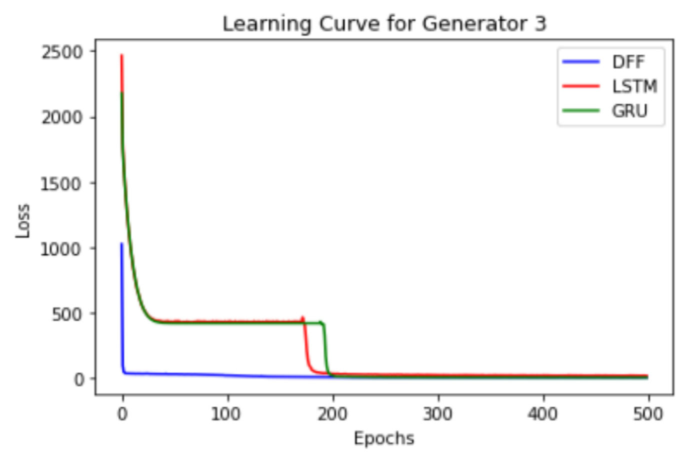

Figure 7. Loss Vs Epoch for Generator3

Table 2. Performance comparison of the experiment

ANN Model	MSE loss values on Test Data.		
	Generator 1	Generator 2	Generator 3
DFF	13.6678	2.8227	3.8296
LSTM	15.2760	3.1170	5.1399
GRU	16.6401	2.8131	3.4041

V. CONCLUSION

This paper presents Deep Feed Forward as the best-suited topology for a non-linear problem of Economic Load Dispatch in comparison to the other two proposed topologies named as Gated Recruitment Unit and Long Short-Term Memory. In DFF, due to less complexity in the structure, both time and learning are optimized to acceptable accuracy. GRU and LSTM have performed well in learning but the time taken is approximately 4 times as compared to DFF and testing loss is slightly greater than the DFF. The performance of GRU and LSTM is somewhat comparable or equal in learning and testing. These results have demonstrated the feasibility of DFF over LSTM and GRU in the case of the non-sequential problem of Economic Load Dispatch.

REFERENCES

[1] P. D. Wasserman and T. Schwartz, "Neural networks. II. What are they and why is everybody so interested in them now?," in IEEE Expert, vol. 3, no. 1, pp. 10-15, Spring 1988, doi: 10.1109/64.2091.

[2] Hochreiter, Sepp &Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[3] Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

[4] Hu, Xiaolin& Wang, Jun. (2006). A Recurrent Neural Network for Solving Nonconvex Optimization Problems. 4522 - 4528. 10.1109/IJCNN.2006.247077.

[5] HadiSadat(1999), "Power System Analysis", Tata McGraw Hills, International Edition

[6] Alnaib, Ibrahim. (2020). IEEE-BUS SYSTEM DATA.

[7] Surekha Deshmukh(2004), Feed-Forward Neural Network Based Economic Load Dispatch, National Power System Conference

[8] Yalcinoz, Tankut& Short, M.J.. (1998). Neural Network Approach for Solving Economic Dispatch Problem with Transmission Capacity Constraints. Power Systems, IEEE Transactions on. 13. 307 - 313. 10.1109/59.667341.

[9] Alexei Botchkarev(2019), Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics:PropertiesandTypology,doi:10.28945/4184

[10] Nwankpa, Chigozie&Ijomah, Winifred &Gachagan, Anthony & Marshall, Stephen. (2020). Activation Functions: Comparison of trends in Practice and Research for Deep Learning.

[11]Liu,Danqing.(2017, november 30). A Practical Guide to ReLU. Medium. https://medium.com/@danqing/a-practicalguide-to-relu-b83ca804f1f7

[12] Kingma, Diederik& Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[13]Phi.Michael(2018,september24).Illustrated Guide to LSTM's and GRU's: A step by step explanation.towardsdatascience.https://towardsdatascience.com/ illustrat ed-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[14] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv preprint arXiv:1412.3555, 2014.

[15]Kostadinov,Simeon, Understanding GRU Networks.towards data science.https://towardsdatascience.com/understanding-gru-networks2ef37df6c9be

[16] Heck, Joel & Salem, Fathi. (2017). Simplified minimal gated unit variations for recurrent neural networks. 1593-1596. 10.1109/MWSCAS.2017.8053242.

[17] Chollet, F. Keras: Theano-based deep learning library. (2015) Code: https://github.com/fchollet. Documentation: http://keras.io.