EBN-Net: A Thermodynamical Approach to Power Estimation Using Energy-Based Multi-layer Perceptron Networks

Koustav Dutta, Rajarshi Pal, and Rajendra Prasad

Abstract Energy forecasting at the generation site in the smart grid furnishes extensive precious data, which further facilitates unprecedented options in smart power applications in emerging smart cities. At power generation facilities excess generated power requires energy storage, and standby power generation results in huge economic losses, whereas under the generation of power lead to fluctuations at the distribution side. In this paper, the power estimation at the combined cycle power plant has been done taking into consideration the various factors affecting the process with the help of multi-layer perceptron networks, but the prime focus of this paper is the presentation and explanation of the working of multi-layer perceptron networks in a way that has never been done before as per the literature survey conducted by us. The core working and the propagation of weights and related processes in the neural networks have been explained with the help of second law of thermodynamics and kinetic theory of gases taking into consideration the works of eminent scientists like Maxwell, Gibbs and Boltzmann. The explanations and mechanisms of the working of the various layers of the multi-layer perceptron network with the viewpoint of thermodynamics presented in this paper opens a new dimension in the interpretation of the working of neural networks which ultimately optimizes the ways of thinking and explainable of the different neurons and synapses of the neural network. The proposed algorithm helps to improve the forecasting with Multi-Layer Perceptron Networks (MLPN) essentially to predict power generation output using the minimum number of input variables. The results, which can be considered highly satisfactory, demonstrate the MLPN's prediction accuracy with a normalized root mean square error for all conditions of less than 5% and with practically no deviation. We demonstrate how beneficial matching of two already proven techniques can bring about spectacular results in energy generation prediction.

Keywords Energy-based model · Multi-layer perceptron network · Neural networks · Power estimation · Combined cycle power plants · Artificial neural networks · Boltzmann distribution · Entropy

K. Dutta · R. Pal · R. Prasad (✉)
KIIT Deemed to be University, Bhubaneswar, India
e-mail: rprasadfet@kiit.ac.in

S. K. Sabut et al. (eds.), *Proceedings of International Conference on Communication, Circuits, and Systems*, Lecture Notes in Electrical Engineering 728,
https://doi.org/10.1007/978-981-33-4866-0_50

1 Introduction

Multi-Layer Perceptron Networks (MLPNs) are non-linear and non-parametric data fitting methods, which can capture non-linear dependencies between parameters far better than the conventional linear models do [1]. For a set of input and output data, the corresponding tuning parameters (weights) of MLPN should be set appropriately, which is referred to as training methods [2]. In this case, the inputs are historical time series lags, and the single output is the one step ahead forecasted value. In this study, a three layer (one input layer, one output layer and one hidden layer) feed-forward MLPN is used, and the weight parameters are calculated by scaled conjugate gradient backpropagation algorithm in an iterative procedure [3]. For the sake of preventing the overfitting over the data, a percentage of data is reserved for validation. If the value of error does not decrease for validation data in a predefined number of consecutive iterations, then the termination criterion will be satisfied, and the training will stop. Eventually, the sigmoid transfer function is used in hidden layer neurons, and the linear transfer function is used in output neuron [4]. This paper focuses on the development and application of deep artificial neural networks for accurate estimation of the power generated from a combined cycle power plant taking into consideration various ambient variables. A Combined Cycle Power Plant (CCPP) is composed of steam turbines (ST), gas turbines (GT) and heat recovery steam generators as a single unit. In a CCPP, the electricity is produced by gas and steam turbines, which are included in the same cycle and is transferred from one turbine to another. The vacuum collected from the system has an effect on the seam turbine and the other three of the ambient variables affect the GT performance.

2 Proposed Algorithm

The paper deals with the problem of inaccurate and adulterated estimation of power generated from a Combined Cycle Power Plant (CCPP). This paper thus brings about a phenomenal work by estimating the power generated in an optimized and precise manner. The paper incorporates the dataset containing 9568 data points collected from a combined cycle power plant over 6 years (2006–2011) when the power plant was set to work with a full load [5]. Features consist of hourly average ambient variables such as temperature (T) in the range 1.81 °C and 37.11 °C, Ambient Pressure (AP) in the range 992.89–1033.30 millibar, Relative Humidity (RH) in the range 25.56–100.16% and Exhaust Vacuum (V) in the range 25.36–81.56 cm Hg to predict the net hourly electrical energy output (EP) of the plant. The architecture of the MLPN network is as follows:

1. The Multi-Layer Perceptron Networks (MLPN) developed in the paper consists of four input neurons in the input layer, i.e. Temperature, pressure, humidity and vacuum which plays the important role of estimation of the power generated from a CCPP.

2. The input layer is connected with the first hidden layer with the incorporation of various weights and biases. The first hidden layer consists of 6 neurons. In this process, the detailed features, inter-correlation and relationship among the different variables are learnt, and feature maps are developed. The weights used help in learning the importance and dependence of each of the variables used in the input layer, and the biases play a role as an adjustment factor, thus involving delay in the learning representation. The activation function used in this first hidden layer is Rectified Linear Unit (ReLU) [6] [ReLU: max $(0, x)$] which helps in extracting the non-linear features from the data. Since ReLU is zero for all negative inputs, it is likely for any given unit to not activate at all.
3. Next, the neurons of the first hidden layer are connected densely with the neurons of the second hidden layer, which consists of 6 neurons, thus forming a stacked representation of the deep neural network. Again, the weights and biases pass through the interconnection in between the two layers, thus helping to extract and understand the detailed inter-relationships in between the various factors which help in the learnt representation of the input vectors in the neural network. The activation function used in this layer is a Leaky ReLU [ReLU: max $(0.01, x)$] which doesn't get affected by the problem of the negative values of the input vectors, thus preventing the process of overfitting of the neural network. The design of the stacked multi-layer perceptron networks is given in Fig. 1.

This beauty of the paper lies in the fact that this paper brings about an interpretation and explanation of a neural network mechanism with the help of laws of thermodynamics and kinetic theory of gases taking into consideration the works of eminent scientists like Maxwell, Gibbs and Boltzmann. The explanations and mechanisms of the working of the various layers of the multi-layer perceptron network with the view point of thermodynamics presented in this paper opens a new dimension in the interpretation of the working of neural networks which ultimately optimizes the ways of thinking and explanables of the different neurons and synapses of the neural network.

The explanation of the multi-layer perceptron networks [7] used in the paper is done in the form of an Energy-Based Model (EBM) which is solely based upon the attainment of minimum energy state by gas molecules in surrounding space and is based upon the concept of kinetic theory of gases [8] and principle of thermodynamics

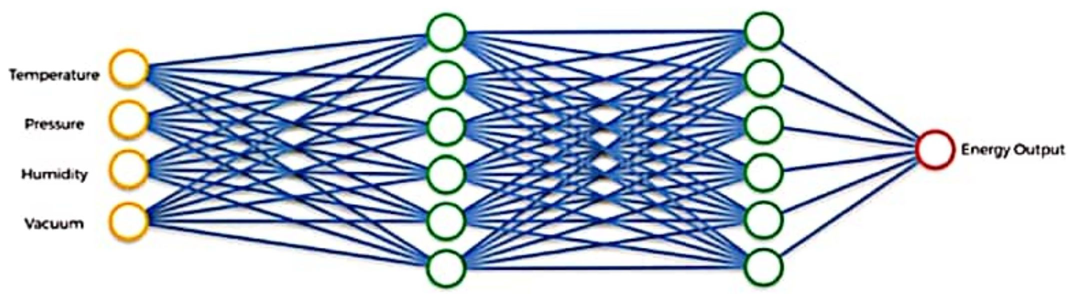


Fig. 1 Architecture of multi-layer perceptron network

[9] of all particles in the universe. In fact, the concept is based upon the fact of attaining zero entropy by all the particles in the universe by minimum energy state attainment. The concept can be explained by the Boltzmann distribution [10] for gaseous particles but is applicable for any particle present in any space in the universe, similarly, the neurons present in the neural network also attain the state of minimum energy in the network by optimization of the learnable parameters used in the neural network with the help of optimization functions and thus reaching the state of minimum energy which is mathematically analogous to point of Global Minima, i.e. Cost function (Loss function for complete training sample) becomes zero. Thus, the concept of the world and particles present in the universe can all be regarded as a neural network all of which are in the process of attainment of minimum entropy, thus reaching minimum energy state.

The equation used in the interpretation of the multi-layer perceptron networks as an Energy-Based Model (EBM) is explainable by creating an analogy and is given by the Boltzmann distribution in Eq. 1:

$$p_i = \left[\frac{e^{-\varepsilon_i/kT}}{\sum_{j=1}^{M} e^{-\varepsilon_j/kT}} \right] \tag{1}$$

where p_i is the probability of a system being in a certain state, ε_i is the energy of the system at state i, k is the Boltzmann constant (1.380649), and T is the temperature of the system.

The Boltzmann distribution tells that the probability of a system remaining in the state i is very low if the energy of the system is high, i.e. there is much randomness or excitation present in the system (The intensity of Brownian Motion [11] of the molecules of the system is high); thus, probability of a system being in a certain state is inversely proportional to the energy of the system in that state.

The excitation or randomness of the molecules of the system is also explained in terms of entropy of the molecules of the system. The velocity of the molecules of the system undergoing Brownian motion due to increase in temperature (entropy also increases) also increases to a large extent, and thus, the probability of the molecules of the system to remain in a particular state also decreases which can be shown by the Maxwell–Boltzmann distribution curve in Fig. 2.

In case of a multi-layer perceptron network, the weights and biases of the neurons or synapses dictate the system to remain in the lowest energy state, where the energy equation depends upon the update of the weights via the optimization process which is given by Eq. 2:

$$E(v, h) = -\sum_{i}^{n} a_i v_i - \sum_{j}^{m} b_i h_i - \sum_{i}^{n} \sum_{j}^{m} v_i w_{i,j} h_j \tag{2}$$

where a_i, b_j are the biases, v_i are the visible nodes, h_j are the hidden nodes, and $W_{i,j}$ are the weights between v_i and h_j.

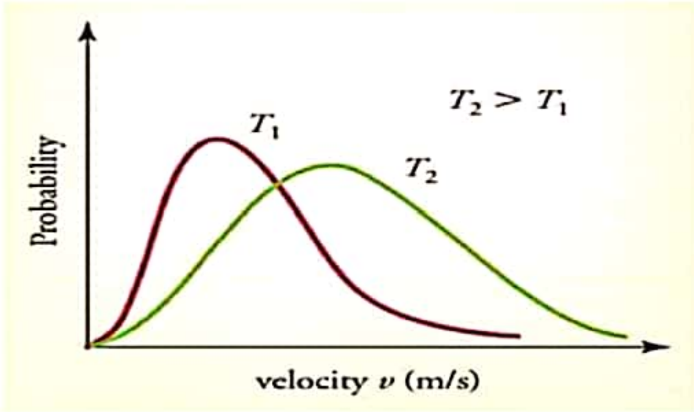


Fig. 2 Maxwell–Boltzmann probability distribution curve

The process of update of parameters causes the change in energy of the system. and thus, the system slowly reaches the lowest energy state which can be shown by the Fig. 3:

The entire architecture of the Multi-Layer Perceptron Networks (MLPN) with the thermodynamical approach of interpretation used for estimation of the power generated has been developed with complex differential equations, linear algebraic representations and the concept of graph theory intertwining with the underlying equations of derivatives and vector algebra. The mathematical equations and developments used in the paper is in alignment with the thermodynamic approach presented in the paper and suffices the interpretation and explanation in all possible ways. The development of the multi-layer perceptron networks is shown in Fig. 4.

where

W_{ij}^{i}	Weights used in each of the connections in between the neurons in the ith layer
X_i	Input vectors incoming into the multi-layer perceptron networks
f_{ij}	Each of the neurons in the hidden layers
O_{ij}	Output vectors outgoing from each of the neurons of the layers
$\hat{y}$	The final predicted output neuron

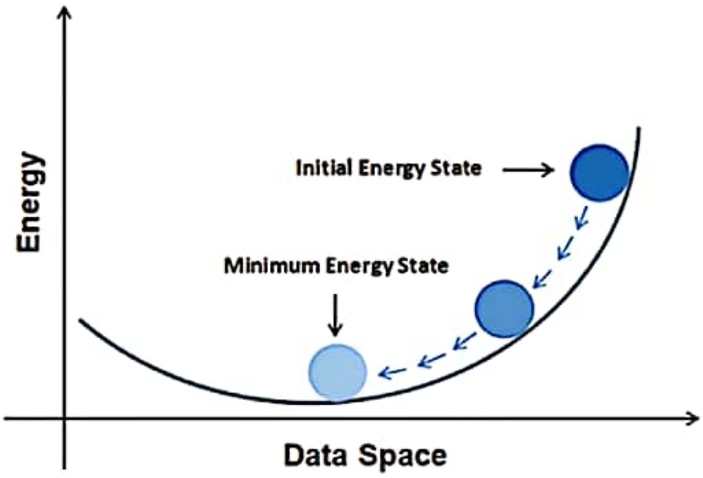


Fig. 3 Energy estimation curve

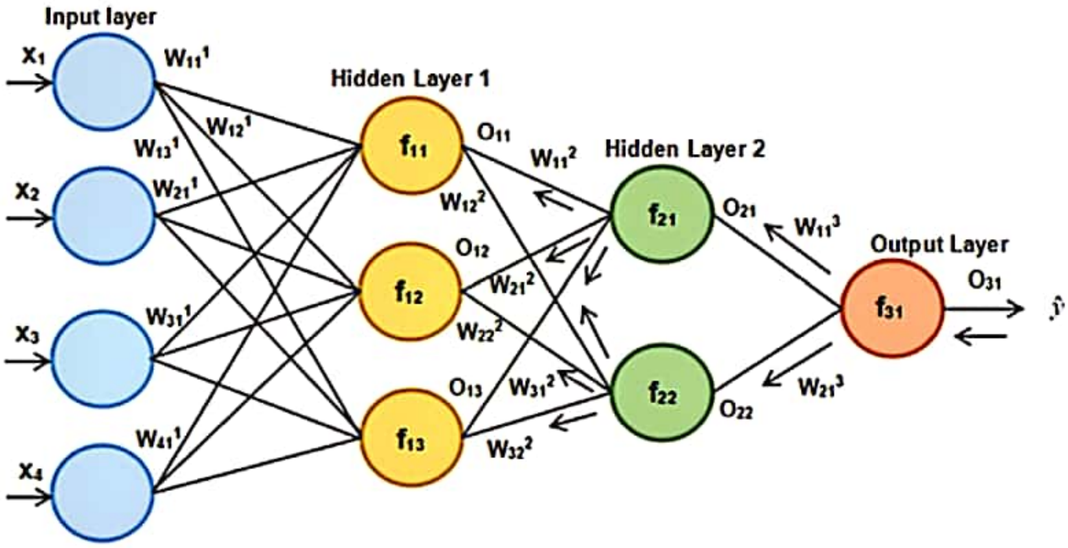


Fig. 4 Detailed architecture and working of multi-layer perceptron network

y Original output (used in the training samples of the MLPN)
$\rightarrow$ Represents the direction in which the optimization is done in the MLPN.

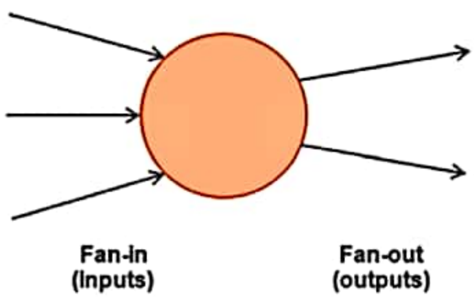
In the architecture of the above neural network, the weight initialization technique used is He-Normal initialization as well as He-Uniform initialization [11]. Let us consider the neuron as shown in Fig. 5.

In the above figure, fan-in represents the input vectors incoming into the neuron and fan-out the output vectors outgoing from the neuron. In this method of He-Initialization, the weights are initialized keeping in mind the size of the previous layer which helps in attaining a global minimum of the cost function faster and more efficiently. The weights are still random but differ in range depending on the size of the previous layer of neurons. This provides a controlled initialization hence the faster and more efficient gradient descent [12].

The equation of the He-Normal initialization is given in Eq. 3 which incorporates a normal or Gaussian distribution.

$$W_{ij} \approx N(0, \sigma), \text{ where } \sigma = \sqrt{\frac{2}{\text{fan} - \text{in}}} \tag{3}$$

Fig. 5 Diagram of a neuron



The equation of the He-Uniform initialization is given in Eq. 4 which incorporates a uniform distribution.

$$W_{ij} \approx u\left[-\sqrt{\frac{\sigma}{\text{fan}-\text{in}}}, \sqrt{\frac{\sigma}{\text{fan}-\text{in}}}\right] \tag{4}$$

The working of the neural network by incorporation of weights and biases with the addition of the activation function can be explained in Figs. 6 and 7.

In the process of getting an accurate estimation of the generated power, the neural network has to minimize the lost function (difference between the predicted and original output) during the training procedure of the neural network by updating the various learnable parameters used in the multi-layer perceptron networks like the weights and biases. This process of optimizing the learnable parameters during the training procedure of the MLPN is done with the help of optimization function. The optimization function used in this paper is Adaptive Momentum Optimizer (Adam) [13] which basically follows a backpropagation algorithm, i.e. it goes back into the neural network like a feedback mechanism and thus updates the parameters which in turn goes on minimizing the loss function and reaching a precise and accurate estimation of the output. The cost function used in the neural network is a mean-squared error loss [14].

The process of optimization can also be explained by the fact that while updating the parameters in the network via the chain rule derivative functions, the optimizer reaches the global minima or lowest energy state of the gradient descent curve [15] (analogically of the energy curve).

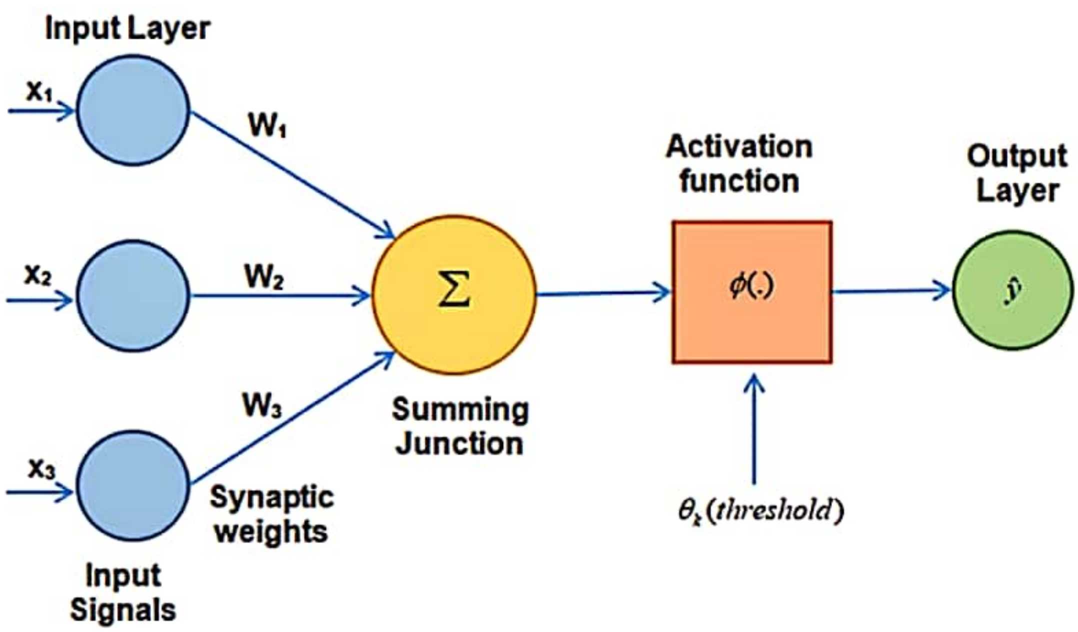
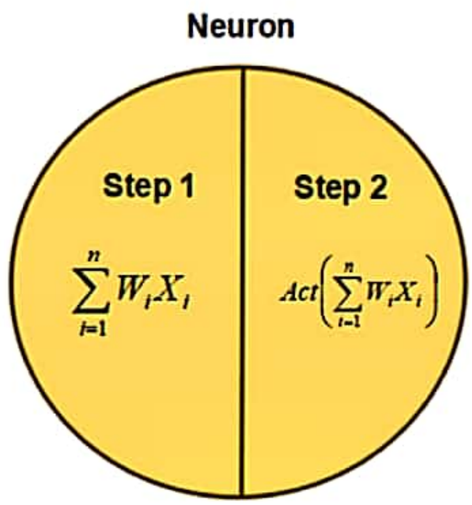


Fig. 6 Working of multi-layer perceptron network

Fig. 7 Detailed mechanism inside perceptron (neuron)



The learning rate value used in this paper is 0.001. Thus, basically, the learning rate parameter helps to decide the rate at which the derivative of the loss function of the neural network will reach zero and attain the global minima position.

The optimization and training of the neural network are done in batches for efficient computation and quick learning. The batch size used in the MLPN is 16, and the number of epochs used in the training process are 150. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. The gradient descent curve is shown in Fig. 8.

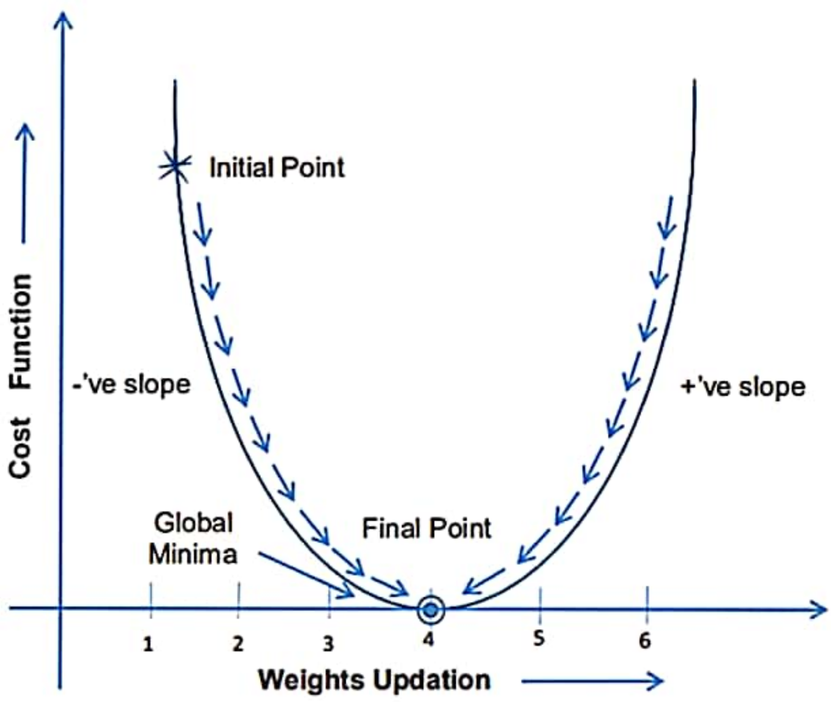


Fig. 8 Gradient descent algorithmic curve

From the figure above, the continuous process of update of the parameters, thus reaching the global minima point is clearly shown. In this way, the neural network (basically the cost function) reaches the lowest energy state having lowest entropy and randomness, and thus, the probability of remaining in this very state is maximum which is in alignment with Maxwell–Boltzmann distribution curve interpretation. The update of the weight parameter in the neural network is given in Eq. 5:

$$W_{\text{new}} = \left(W_{\text{old}} - \eta \frac{\partial L}{\partial W_{\text{old}}}\right) \tag{5}$$

where

W_{old} Initial weight of the neural network;
W_{new} Updated weight due to optimization.
L Cost function of the neural network;
η Learning rate.

The equation of the Cost Function is given in Eq. 6:

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}(y - \hat{y})^2 \tag{6}$$

where

N number of training samples;
y original output;
$\hat{y}$ predicted output.

The entire process of the backpropagation algorithm and continued chain rule process for the update of the parameters via the optimization function as shown in the equations below with respect to the architecture of the multi-layer perceptron networks shown in Fig. 1.

$$W^3_{11\text{new}} = W^3_{11\text{old}} - \eta \frac{\partial L}{\partial W^3_{11\text{old}}}$$

$$\frac{\partial L}{\partial W^3_{11}} = \left(\frac{\partial L}{\partial O_{31}} \cdot \frac{\partial O_{31}}{\partial W^3_{11}}\right); \left(\frac{\partial L}{\partial W^3_{21}}\right) = \left(\frac{\partial L}{\partial O_{31}} \cdot \frac{\partial O_{31}}{\partial W^3_{21}}\right)$$

$$W^2_{11\text{new}} = W^2_{\Delta\Delta\text{old}} - \eta \frac{\partial L}{\partial W^2_{11}}$$

$$\left(\frac{\partial L}{\partial W^2_{11}}\right) = \left(\frac{\partial L}{\partial O_{31}} \cdot \frac{\partial O_{31}}{\partial O_{21}} \cdot \frac{\partial O_{21}}{\partial W^2_{11}}\right) + \left(\frac{\partial L}{\partial O_{31}} \cdot \frac{\partial O_{31}}{\partial O_{22}} \cdot \frac{\partial O_{22}}{\partial W^2_{11}}\right)$$

The application of the process of contrastive divergence approach which is specifically used in Gibbs sampling process (special case of Markov-Chain Monte-Carlo

Table 1 Comparison of accuracies of different network optimization techniques

Optimization technique or algorithm	Accuracy score (%)
Adaptive momentum	98.93
Adaptive gradient	96.23
Adaptive delta	92.45
RMSProp	91.34
Stochastic gradient descent with momentum	87.67

Technique) can also be done in this case due to the interpretation of the multi-layer perceptron networks as an energy-based model, and thus, due to the application of contrastive divergence, the learning process of the algorithm during training phase can be optimized to a large extent, and the point of minimum energy or global minima can be reached very fast and efficiently with more accurate and proper updated learnable parameters in order to predict or estimate the output with highest level of accuracy.

3 Results and Conclusion

Accuracy is used as a metric in order to analyze the performance of the multi-layer perceptron networks for estimation of the power generated in the CCPP. A comparative analysis report of accuracies achieved with the help of various network optimization technique is given in Table 1.

Therefore, from the table, it is deduced that the Adaptive Momentum (Adam) optimization algorithm which is actually a combination of Stochastic Gradient Descent with Momentum (SGDM), and RMS prop algorithms help in the best optimization of weights and parameters in order to achieve an accuracy of 97.93%, which is by far the most robust and efficient algorithm to achieve such a 'State-of-Art' accuracy in the process of estimation of the power generated in the CCPP.

References

1. H. Nascimento Camelo, P.S. Lucio, J.B.V.L. Junior, P.C.M. de Carvalho, A hybrid model based on time series models and neural networks for forecasting wind speed in the Brazilian northeast region. Sustain. Energy Technol. Assess. **28**, 65–72 (2018)
2. G. Sadeghi, S. Nazari, M. Ameri, F. Shama, Energy and energy evaluation of the evacuated tube solar collector using Cu_2O/water nanofluid utilizing MLPN methods. Sustain. Energy Technol. Assess. **37**, 100578 (2020)
3. C. Khadse, M. Chaudhari, V. Borghate, Conjugate gradient back-propagation based artificial neural network for real-time power quality assessment. Int. J. Electr. Power Energy Syst. **82**, 197–206 (2016)

4. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. J. Machine Learn. Res. **15**(1), 1929–1958 (2014)
5. H. Kaya, P. Tüfekci, F.S. Gürgen, Local and global learning methods for predicting power of a combined gas & steam turbine, in *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE*, pp. 13–18 (2012)
6. K. Hara, D. Saito, H. Shouno, Analysis of function of rectified linear unit used in deep learning, in *2015 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2015), pp. 1–8
7. P. Tüfekci, Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. Int. J. Electr. Power Energy Syst. **60**, 126–140 (2014)
8. E. Wild, On Boltzmann's equation in the kinetic theory of gases, in *Mathematical Proceedings of the Cambridge Philosophical Society* (Vol. 47, No. 3, pp. 602–609). Cambridge University Press: A. Kleidon, R.D. Lorenz (eds.), *Non-equilibrium Thermodynamics and the Production of Entropy: Life, Earth, and Beyond* (Springer Science & Business Media, 2004)
9. J.S. Rowlinson, The Maxwell-Boltzmann distribution. Mol. Phys. **103**(21–23), 2821–2828 (2005)
10. G.E. Uhlenbeck, L.S. Ornstein, On the theory of the Brownian motion. Phys. Rev. **36**(5), 823 (1930)
11. T.A. Sai, H.H. Lee, Weight initialization on neural network for neuro PID controller—case study, in *2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT)* (IEEE, 2018), pp. 1–4
12. R. Lenka, K. Dutta, A. Khandual, S.R. Nayak, Bio-medical image processing: medical image analysis for malaria with deep learning, in *Examining Fractal Image Processing and Analysis* (IGI Global, 2020), pp. 158–169
13. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization (2014). arXiv preprint arXiv: 1412.6980
14. C.M. Theobald, Generalizations of mean square error applied to ridge regression. J. Roy. Stat. Soc.: Ser. B (Methodol.) **36**(1), 103–106 (1974)
15. L. Bottou, Large-scale machine learning with stochastic gradient descent, in *Proceedings of COMPSTAT'2010* (Physica-Verlag HD, 2010), pp. 177–186