An Optimized ANN Model For Predicting The Efficiency Of Perovskite Solar Cell Using MATLAB

Navya Jyothi G and Dr. Saara K
Department of Electronics and Communication Engineering, School of Engineering,
Dayananda Sagar University, Bangalore – 560068, India

Abstract-- The amalgamation of material science genome and algorithmic development has elevated the evolution of material science. Traditional methods of material discovery, development and deployment takes a long time frame. Therefore, machine learning models which primarily learns from past data helps in catering to the inherit limitations of conventional methods used in material science. Hence we demonstrate the potential of deep learning via Artificial Neural Network (ANN) which utilizes radical features to predict the efficiency of perovskite solar cell. Dataset was collected varies technical papers. The trained model then predicts the efficiency on unseen perovskite data. This paper also finds insights of challenges faced with ANN and how it could be improvised in the near future.

***Index Terms*-- Neuron, Perovskite layer, ANN, Perovskite solar cell, Power Conversion Efficiency, Over fitting.**

I. Introduction

The fruition of Perovskite Solar Cell has risen to a power conversion efficiency (PCE%) of 22.1%. Pervoskite outshines silicon prominently due to less expenditure of manufacturing energy. The anticipation of intrinsic complication of traditional methods of material discovery, manufacturing and deployment pays way towards computational simulation. Nevertheless computational simulation requires high performance computing equipment and no precise use can be made from the preceding experimental result. Henceforth, the novel course of action adopted in the material community has progressed to the employment of deep learning. High throughput of deep learning is because of the fact that it deserts the man made factors and focus mainly on the factors affecting the efficiency of Perovskite Solar Cell. Deep learning can learn either through supervised or unsupervised learning. In supervised learning the task is to map the relationship between an input and an output based on the example pair of input and ouput. In this work supervised learning is used.

II. Artificial neural network (ANN)

Artificial Neural Network finds its innovation from biological neural system but the operation and architecture has gone beyond the inspiration. Neural Network has been used since many decades now. Deep learning has its own hype cycle but at the recent times the hype has reached pinnacle because of the development of high computing power and big data era. The close collaboration of deep learning and data mining measurably surpasses the limitations of experimental observation. Neural Network maps linear or non linear relationship between inputs and targets by iteratively learning from data thereby predicts the variable of interest. ANN consists of three inescapable layers namely input, output and hidden layers. The number of hidden layers is inconstant i.e it can be declared according to the complexity of the problem. Each layer consists of neurons which accepts the inputs from the dataset, processes it and conveys the result to the output neuron. Neurons in the hidden layer acts as negotiator for dividing out the signals to the neurons in the output layer. Neurons in the input layer receives inputs from the dataset xi belongs to R [represented by x_1,x_2 and so on] and send a weighted sum to the neurons in the hidden layer. Weights [denoted as w_1, w_2 and so on] are randomly generated real numbers which are iteratively changed in the direction of reducing the error of predicted output value. The output of the hidden layer is given by where f is threshold function.

$$y_j = f\left(\sum_{i=1}^{n} w_{ij} x_i\right)$$

The output from the hidden layer are mapped onto the significant activation function in the output neuron and output is generated. The obtained output is compared with the actual target and the error is computed. ANN is model using Matlab R2016a tools. Matlab is a numeric computing environment and a programming language.

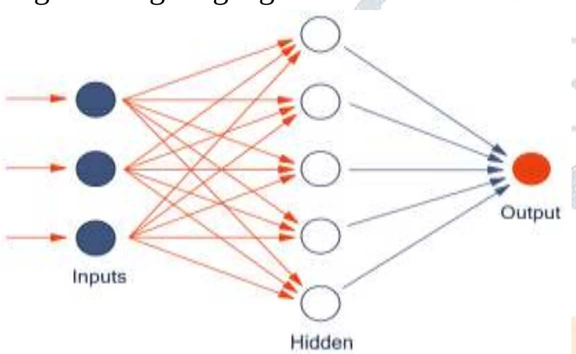

Fig 1. General Architecture of ANN

2.1 *Basic steps of deep learning*

1. *Dataset preparation*

The data is gathered from several experimental measurements and computational simulation. Dataset is preferred to be as huge as possible so that the model learns and not memorizes. The research on perovskite solar cells has its inception at the latest times therefore seven features and sixty one samples were used in the dataset. Based on the combination of these seven features the efficiency of perovskite solar cell is forecasted. There are many internal and external factors that affects the efficiency of Perovskite solar cell. But only significant variables are utilized leaving behind the redundant factors. Feature selection is a predom- inant concept which impacts the performance of an ANN. In this work parameters like perovskite layer, device structure, energy band gap, power input, iodine to bromide ratio, lead to tin ratio, formamidinium to methyl ammonium ratio. Energy band gap of the perovskite layer is chosen as the main predictor. In the active perovskite layer the photon is absorbed and excites an exiton only if the photons energy is above the energy gap of the perovskite material.

PARAMETER	MIN	MAX
Perovskite layer	**1**	**50**
Device Structure	**1**	**28**
Power Input	**100**	**100**
Energy Band Gap(eV)	**1.08**	**3.78**
Pb/Sn	**0**	**3**
FA/(FA+MA)	**0**	**1**
I/Br	**0**	**1**
PCE(%)	**0.14**	**14.6**

Table 1: Summary of input and output parameter values.

2. *Conversion of Cell array into numeric matrix*

ANN model does not accept and process input in the form of alphanumeric. Perovskite material is formulated as ABX_3. A and B are cations which could be methyl ammonium or formamidinium and X is halide. Different types of perovskite material can be built using different combinatins of AB and X. Perovskite solar cell can be inherited in inverted, planar and mesoporous type of architecture. Hence perovskite layer and device structure has be converted into numeric values before being conveyed to ANN model.

3. *Data Preprocessing*

When raw data is collected it is very noisy i.e it consists of data that cannot be comprehended. It haphazardly increases the storage space and prejudicially agitates the results. Therefore data scrubbing has to be performed on the inconsistent data using certain data preprocessing techniques. Normalization reduces redundancy of data and allows data to occupy less storage terminating in increased performance.

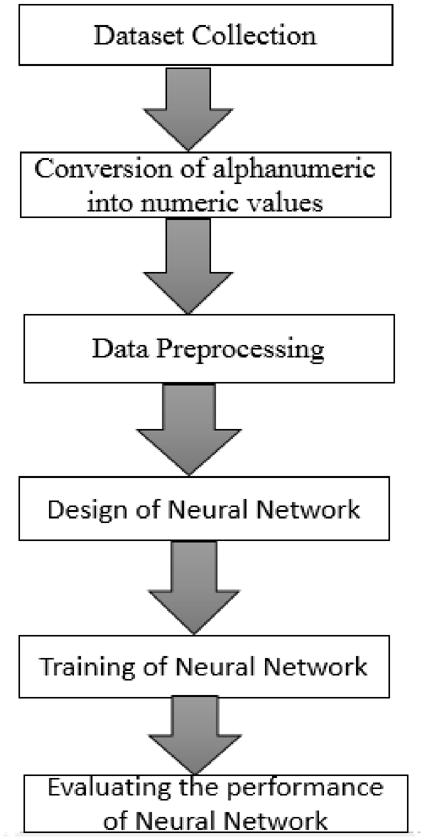

Fig 2: Flow Chart of steps of deep learning

4. Neural Network Building

It consists of simple processing elements that builds connection between inputs and targets using linear or non linear mapping. In this step the user defines the number of neurons in input, output neuron. For optimization it is tried with consecutive numbers and chosen the one which produces least root mean square error (RMSE). The learning algorithm used in this work is Levenberg-Marquart. Levenberg-Marquart has the fastest convergence. This advantage is distinguishable when accurate training is required.

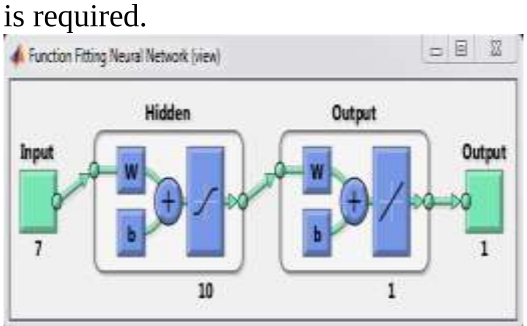

Fig 3: Network architecture of ANN model

5. Training the network

The next step is to split the dataset into training and testing set. The total dataset was partitioned into 85% of training set and 15% of testing set. During the training process arbitrary values are assigned to weights and carries on monotonously. Epoch is loop topology of the complete training set. Many epochs are required for the neural network to learn from database. Also, the weights are calibrated in every epoch.

As the iterative process of incremental calibration continues, the weights eventually gets closer to locally optimal set of values. Once the training is completed a subset of training set is given as test set just to examine if the ANN model is trained with no over fitting.

Actual Output	Predicted Output
5.48	4.54411
5.73	4.85389
4.27	5.17527
5.44	4.82865
6.4	5.46128
5.44	4.82865
1.86	3.33501
1.1	1.13393

Fig 4: Comparison table of actual output and obtained output

Root Mean Square designates the absolute fit of the model. It indicates how close the predicated value is to the actual value. Lower the value of RMSE specifies better fit.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_p - y_i)}$$

where y_p is predicted value of efficiency and y_i is actual value of efficiency. For of values in fig 4 RMSE was calculated to be 0.8589.

6. Testing the Neural Network

The performance of the trained Neural Network is evaluated in the testing stage. In this step ANN is exposed to raw data. Lower value of RMSE of testing set indicates that the neural network has generalized. But in few cases over fitting could be observed. Over fitting refers to the model that fits on the training data very well that it fails to generalize precisely.

Actual Output	Predicted Output
5.48	5.455944145
5.73	5.913519897
4.27	3.344895559
5.44	5.388937237
6.4	5.064718502
5.49	5.064718502
5.36	5.560716743
1.94	2.646191538

Fig 5: Comparison of actual output and predicted output.

From the above table it can be observed that neural network is able to generalize.

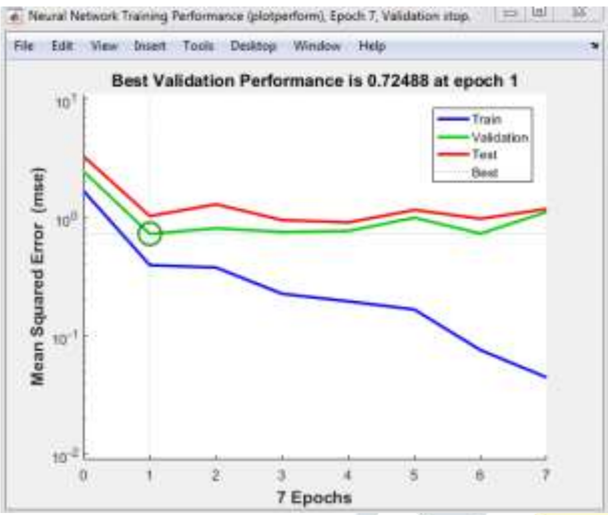

Fig 6: Validation Performance

The above figure indicates that the training continued for 6 more iterations. The curves of validation and test are almost similar. This is a good accuracy for long term prediction.

III. FUTURE SCOPE

ANN model has better accuracy for predicting the efficiency of perovskite solar cell. Over fitting and under fitting is one of the main limitation of ANN model. Cross validation tests can be used to prevent the problem of over fitting. Another limitation of ANN is that time consuming experimentation is required for determining the network structure and there's lack of elucidation of weights during the neural network building process. Techniques like principle component analysis (PCA) could be performed that reduces the multi dimensionality of the dataset. PCA could also eliminate the problem of over fitting.

IV. REFERENCES

1. Yue Liu, Tianlu Zhao, Wangwei Ju, Siqi Shi, Materials discovery and design using machine learning.
2. Shubhankar Kapoor, Ashish Singh, Utkarsh Goswami, Saswat Kumar, Gaurav Chitranshi, Design and Implementation of a Robust System for Recognizing Alphabets using Artifical Neural Network.
3. Maitha H. Al Shamisi, Ali H. Assi and Hassan A. N. Hejase, Using MATLAB to Develop Artificial Neural Network Models for Predicting Global Solar Radiation in Al Ain City – UAE
4. Ghanshyam Pilania, Prasanna V. Balachandran, Chiho Kim and Turab Lookman, Finding New Perovskite Halides via Machine Learning.
5. Marijana Zekić-Sušac, Nataša Šarlija, Sanja Pfeifer, COMBINING PCA ANALYSIS AND ARTIFICIAL NEURAL NETWORKS IN MODELLING ENTREPRENEURIAL INTENTIONS OF STUDENTS.
6. Avrim L.Blum and Ronald L.Rivest. Training a 3-Node Neural Network is NP Complete.
7. Stefano Galelli, Greer B. Humphrey, Holger R. Maier, Andrea Castelletti, Graeme C. Dandy, Matthew S. Gibbs. An evaluation framework for input variable selection algorithms for environmental data-driven models.
8. Junita Mohamad-Saleh, Brian S. Hoyle. Improved Neural Network Performance Using Principal Component Analysis on Matlab.
9. Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi and Chiho Kim. Machine learning in materials informatics: recent applications and prospects
10. Xianbo Xu and Nikhil Gupta. Artificial Neural Network Approach to Predict the Elastic Modulus from Dynamic Mechanical Analysis Results.