National Institute of Technology Karnataka, Surathkal

Photovoltaic Array Fault Identification Using ANN with Improvised Learning Techniques

Submitted by:

Aditya S Gourishetty (171EE103)

Dwijesh Athrey K (171EE116)

M Nikhil Bharath (171EE223)

Supervisor: Dr. Debashisha Jena

Department of Electrical and Electronics Engineering

Contents

1 Declaration **2**

2 Acknowledgement **3**

3 Abstract **4**

4 Introduction **5**
4.1 Photovoltaic systems 5
4.2 Faults in Photovoltaic systems 6
4.3 Neural Networks 8
4.4 Adam Optimizer and Cyclical learning Rate 10

5 Experimentation and results **14**
5.1 Simulation and Data Generation 14
5.2 Data Preprocessing 14
5.3 Training and Testing the Model 15
5.4 Results and Accuracy 15

6 References **17**

1 Declaration

We hereby declare that the project entitled "Photovoltaic Array Fault Identification Using ANN with Improvised Learning Techniques" was carried out by us during the V Semester of the academic year 2019-2020. We declare that this is our original work and has been completed successfully according to the direction of our guide Dr.Debashisha Jena and as per the specifications of NITK Surathkal.

Place:
Date:

Aditya S Gourishetty (171EE103)

Dwijesh Athrey K (171EE116)

M Nikhil Bharath (171EE223)

2 Acknowledgement

Firstly we would like to thank Dr. Debashisha Jena, Dept. of Electrical and Electronics for his prolonged support which was crucial for driving this project to its completion. We are also immensely grateful to Dr.Debashisha Jena for giving his time for providing his profound insights on our initial and intermediate results and accordingly suggesting changes to our methods to get better results without which our project could not have arrived at this stage.

We would also like to thank Mr. Reddiprasad Reddivari for providing us with appropriate guidance in acquiring the required knowledge on Photovoltaic Systems which was essential for this project. We are also thankful to Electrical and Electronics Department, National Institute of Technology Karnataka for providing us the platform to learn and acquire the relevant skills and for giving the opportunity to be a part of such projects. Lastly, we thank our families and friends for providing us with constant support throughout the course of the project.

3 Abstract

Photovoltaic (PV) systems are exposed to harsh and unforgiving conditions in their environment, which can lead to development of different types of faults in the system. Usually, these faults are detected and identified manually. This process can be prone to errors and takes a lot of time. This project describes a machine learning approach using artificial neural networks (ANN) that can presently detect and identify 3 different types of fault by using data collected from the simulation of a PV system and can be extended for other faults based on the availability of data.

4 Introduction

Electrical energy is the most commonly used form of energy in the world, given the convenience with which it can be converted into other different forms of energy as per requirements. Since electricity was first generated commercially in 1870, the demand for it and consequently the generation, has only been growing exponentially. Conventionally, coal, oil and natural gas have been used to generate electricity in thermal power plants and even today, these account for close to two-thirds of the total electrical energy generated in the world.

However, there has been a great shift towards renewable energy sources such as wind, hydro and solar energy, especially in the last 15-20 years, as there is greater realisation that these resources are more sustainable. Generation of power using solar energy is a field that has been identified as having a lot of potential, given the amount of solar energy the Earth receives, decrease in costs and advancement of the semiconductor device technology used in photovoltaic (PV) systems. This has lead to a lot of research in the area of PV systems.

Due to the highly complex structure of these systems and the harsh environment they are subjected to, faults can develop in these systems, which lead to power losses. Hence, the systems need to be monitored for faults at all times.

4.1 Photovoltaic systems

The word 'photovoltaic' is a combination of the Greek word 'phōs' which means 'light' and the word Volta from the name of the Italian physicist Alessandro Volta who is credited with the invention of the electric battery, which is also named as the unit for electric potential. As the name suggests, Photovoltaic systems are the systems dealing with the conversion of light energy or photons into conveniently utilizable electrical energy. Photovoltaic systems are basically a set of photovoltaic cells or PV cells electrically connected in a specific topology along with some power electronic devices such as inverters to facilitate the conversion of solar energy into electrical energy to be used for various applications. The basic principle of converting solar energy into electrical energy which is practiced in solar cells is that the photons from the light are used to excite the electrons in the semiconductor material which when connected in a suitable topology with other cells can generate considerable electric potential or voltage to be used to drive current to the load.
There are various topologies in which the PV cells are arranged in combination with the inverters. The term 'string' is extensively used while referring to the topologies of PV systems. A 'string' in the domain of PV systems is a set of PV of PV cells electrically connected in series. As mentioned in [4] some of the topologies can be centralised topology where all the strings in the PV system are connected in parallel which are in turn connected to a central inverter, string topology where there each string is connected to a separate inverter, AC Module topology where each PV Module has its own inverter and multistring topology which is a combination of central and string topology.

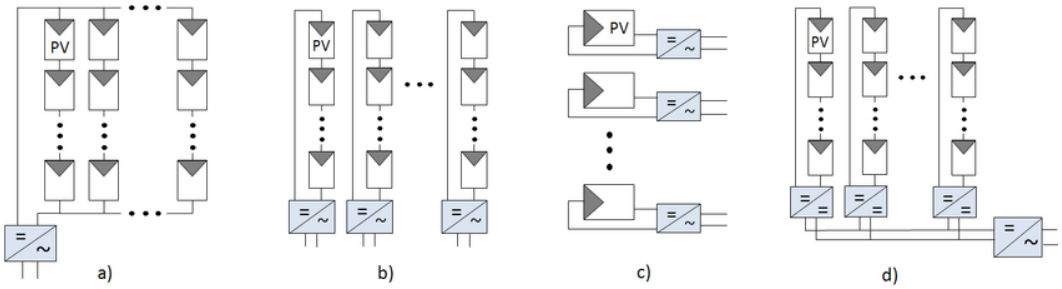

Figure 1: PV System topologies[4] a) Central b)String c)AC Module d)Multistring Source:[6]

PV cells have the I-V characteristics (current vs voltage characteristics) is similar to the reverse bias characteristics of a diode. For a PV cell the a unique I-V curve is observed for a given set of Irradiance and temperature. As the Irradiance is increased the curve is shifted upwards due to generation of more current for the same voltage because of more photons falling on the PV cell to excite more electrons as shown in figure 2a. The rise in temperature causes reduction of band gap due to which the electrons get excited at lower voltage and therefore the cut off voltage is reduced which is signified by a leftward shift of the I-V curve as shown in figure 2b.

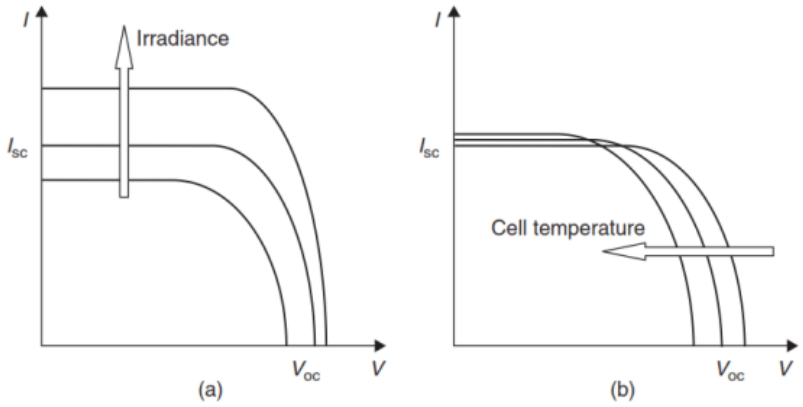

Figure 2: a)Effect of Irradiance b)Effect of Temperature Source:[5]

From the I-V curve of a solar cell one can obtain a P-V curve which is basically (I x V) vs V curve. As the current is fairly constant with increase in voltage until the cut off voltage, the maximum power point occurs at the knee of the I-V curve as shown in the figure 3 . For maximum power output, the PV systems are therefore operated at this operating point of current and voltage using various methods commonly known as MPPT or Maximum Power Point Tracking Techniques. We can also observe that with a rise in Irradiance there is better MPP and with a rise in temperature we observe a worse MPP.

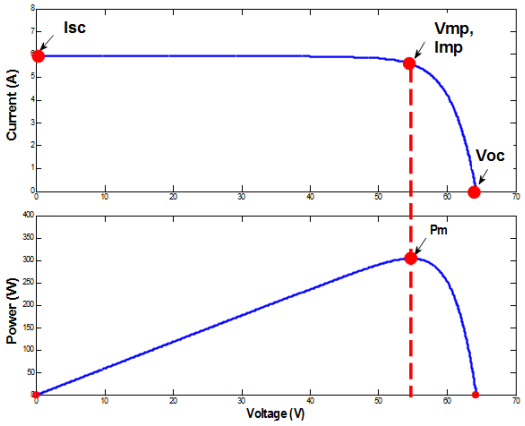

Figure 3: MPP in I-V and P-V curves

4.2 Faults in Photovoltaic systems

In [7], the detailed classification and comprehensive study of the various faults observed in PV systems is done. Due to the ease of simulation, for this project we have considered three of the faults which are partial shading, bypass diode short circuit and hotspot.

Partial Shading

Partial shading occurs when there is non-uniform irradiance on the PV module. For example two cases are taken in [7] as shown in figure , where in one case there is an irradiance of 1000 W/m2 for all the cells except 3 of them in a string which have an irradiance of 800 W/m2, and case two where there is one more set of PV cells with an Irradiance of 500 W/m2. When there is partial shading as explained in [7] there is emergence of multiple steps in the I-V curve as shown in figure 4 which is equal to the number of cells with a lower irradiance which in turn causes the emergence of multiple peaks in the P-V curve. This is not desirable as it might cause problems in MPPT. Also, there is an overall reduction of Pmax.

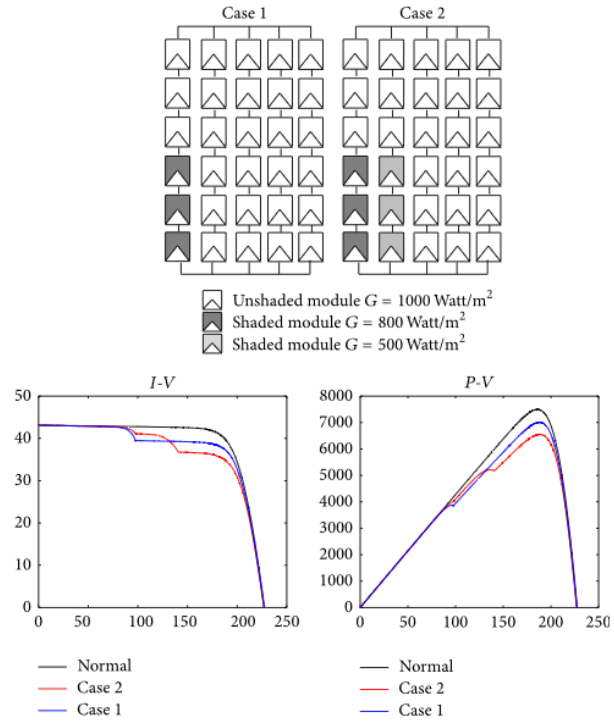

Figure 4: Partial Shading Source:[7]

Bypass diode short circuit

This fault occurs when one of the PV cell the PV module is short circuited due to the short circuit of the bypass diode. The Pmax and Voc is reduced while the current remains about the same as shown in the figure 5. The explanation of this fault is given also given in [7].

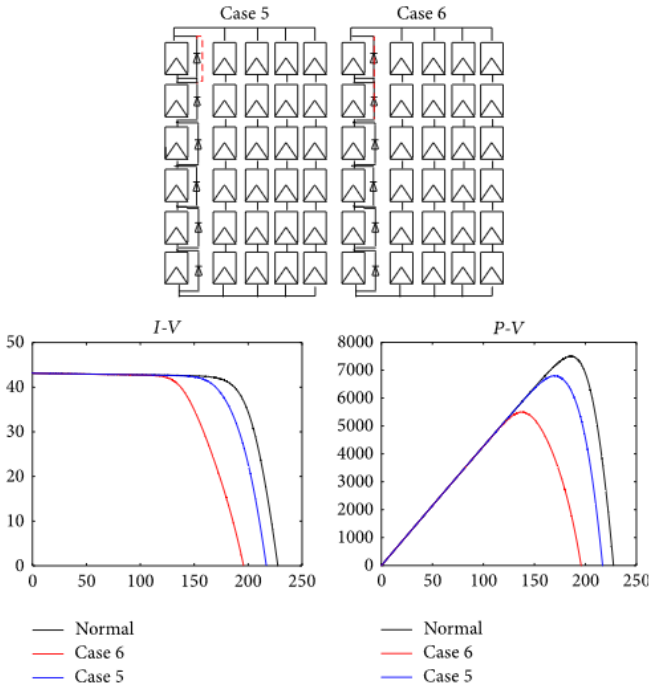

Figure 5: Short Circuit Fault Source:[7]

Hotspot

Hotspot faults occur when there is a significant rise in temperature at specific cells caused by manufacturing defects or external damage. Due to rise in temperature there might be a slight decrease in the cut off voltage in the I-V curve resulting in a overall decrease in Pmax but more than the performance issues of PV module there will be permanent physical damage due to this fault and therefore is undesirable.

4.3 Neural Networks

A basic neural network consists of neurons or nodes across different layers in its model that are used in making predictions, classifying things, etc. There exists a great deal of similarity between the architecture of the human brain and the modelling and functioning of neural network. As human brain learns from its past experiences, a neural network model learns by itself from its past values through the process of training. In simple words, we initially provide the neural network with a set of input and its corresponding output. Now, the objective of the neural network lies in intently matching the output values over the course of training. The cost function of the neural network calculates the error between the expected output and the predicted output. The figure shown below is a sample of a simple neural network.

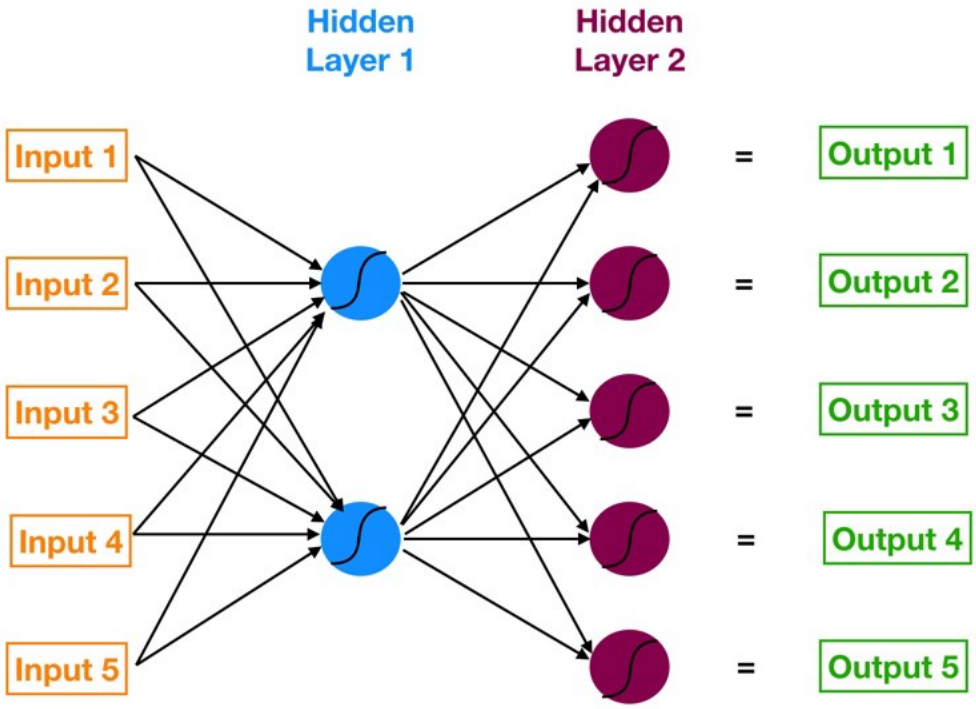

Figure 6

It consists of an input layer (orange) and a couple of hidden layers, out of which one being the output layer (green) essentially used in the prediction of the model. All the nodes present in the different layers of the network are interconnected as shown by the arrows. Each arrow carries a certain value known as weight which acts as a multiplier from the node it emerges. Inside each node is where all the computations take place as it sums the inputs from different nodes multiplied with its corresponding weights. The calculated value is then passed to an activation function which decides if the given value must go further or in other words if the neuron has been activated or not.

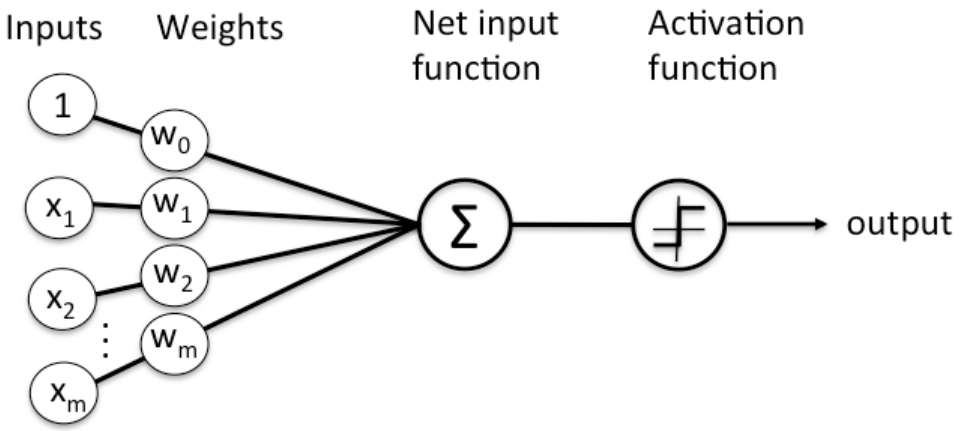

Figure 7

The learning is primarily done in two steps - broadly known as forward propagation and backward propagation.

Forward propagation:
The given input propagates forward to the units in each hidden layer, finally producing the output with the help of activation functions. If In1 is the input of the neural network and W1, W2, W3, W4 are its weights then the output units of the first layer is given by:

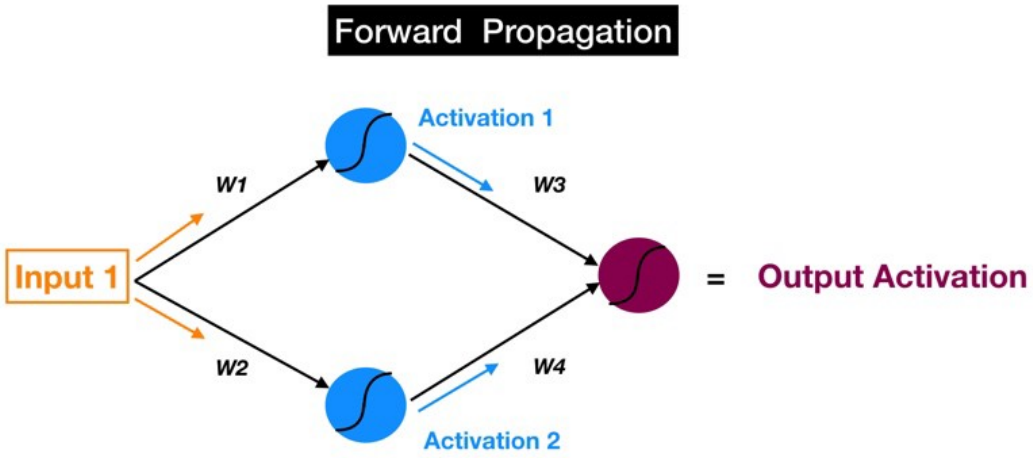

Figure 8: Forward Propagation

h11 = W1*In1 + Bias_Neuron1
h12= W2*In1 + Bias_Neuron1
A2 = Neuron 1 Activation = Sigmoid(H11)
A3 = Neuron 2 Activation = Sigmoid(H12)
Similarly,
h21= W3*A2 +W4*A3+ Bias_Neuron2
Predicted output = Sigmoid(h21)
Here sigmoid is used as the activation function but in reality there are numerous activation functions that can be used. Sigmoid, Tanh, Rectified Linear Unit (ReLU), Leaky relu are some of the widely used activation functions.

Backward propagation:
The backward propagation algorithm traverses backward in the neural network computing the gradient of the cost function with respect to the weights whilst updating the weights in the neural network.

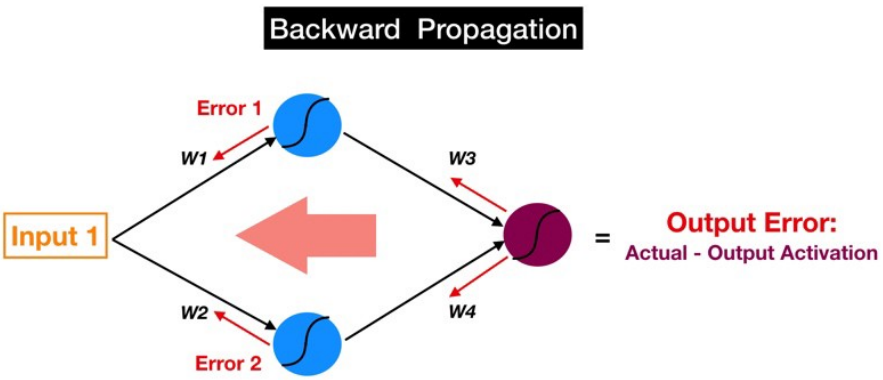

Figure 9: Back Propagation

The weights W1, W2, W3, W4 are updated in the fashion as shown below
W4=W4 - alpha * d(Cost Function)/dW4
W3=W3 - alpha * d(Cost Function)/dW3
W2=W2 - alpha * d(Cost Function)/dW2
W1=W1 - alpha * d(Cost Function)/dW1

Here alpha denotes the learning rate. This process of learning is popularly known as gradient descent which is one of the most basic optimizing algorithms. Other optimizing algorithms such as Adam optimizer and Cyclic

learning rate algorithms are discussed in the subsequent chapters.

This process of forward and backward propagation repeats itself over the given number of iterations until the cost function is minimized to a certain value and the predicted output approximates the expected output.

4.4 Adam Optimizer and Cyclical learning Rate

Adam optimizer
An adaptive learning algorithm is a type of algorithm whose parameters change itself over time during its course of operation inorder to optimize the cost function effectively. Adam optimizer, firstly introduced in the year 2014 and presented at the ICLR 2015 conference has gained immense popularity as one of the finest adaptive learning algorithms since then. It comes under the umbrella of gradient descent algorithms and uses the combination of RMSprop and Stochastic gradient descent with momentum. In SGD, one random example is selected out of the many examples for each iteration and is optimized against selecting all the examples in the normal gradient algorithm.

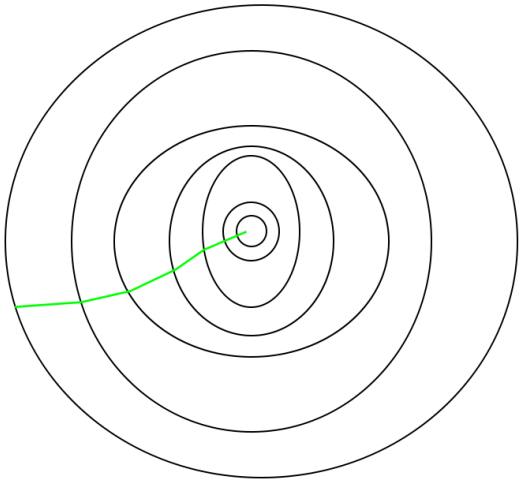

Figure 10

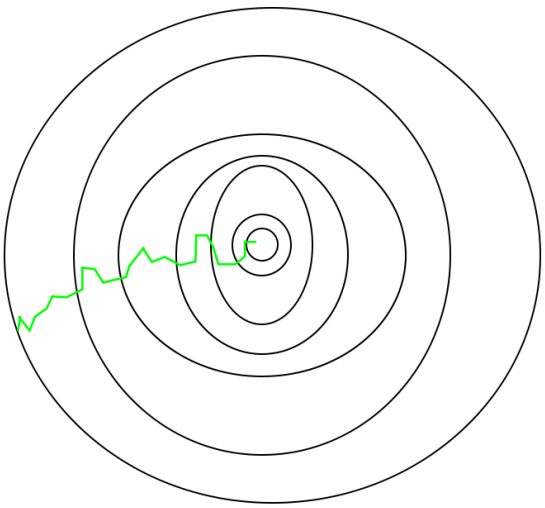

Figure 11

From the figure it is evident that SGD is noisier than the normal gradient descent and hence takes more number of iterations to achieve minimal but on the whole it is much more effective than using the normal gradient descent algorithm.

Gradient descent with momentum converges to the minima much faster than the normal gradient. The process of momentum is very similar to that of normal gradient descent except for the fact that it restricts the

movement of the function in one direction which leads to faster convergence.

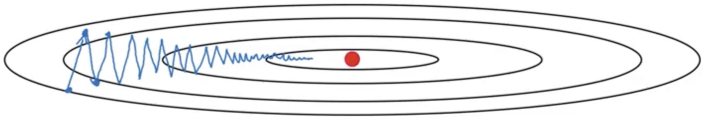

Figure 12

Here it restricts the oscillations in the y axis. The formula for the momentum is as shown below

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw$$

$$v_{db} = \beta \cdot v_{dw} + (1 - \beta) \cdot db$$

$$W = W - \alpha \cdot v_{dw}$$

$$b = b - \alpha \cdot v_{db}$$

Figure 13

RMSprop is a modified version of the gradient descent with momentum as it takes the root mean square of gradients into account as shown in the formula below.

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2$$

$$v_{db} = \beta \cdot v_{dw} + (1 - \beta) \cdot db^2$$

$$W = W - \alpha \cdot \frac{dw}{\sqrt{v_{dw}} + \epsilon}$$

$$b = b - \alpha \cdot \frac{db}{\sqrt{v_{db}} + \epsilon}$$

Figure 14

Adam just combines all the above optimization algorithms. Each parameter in adam's optimizer has different learning rate. It uses first and second moment of gradient to compute its learning rate for each parameter.

In TensorFlow: learning_rate=0.001, beta1=0.9, beta2=0.999, epsilon=1e-08.

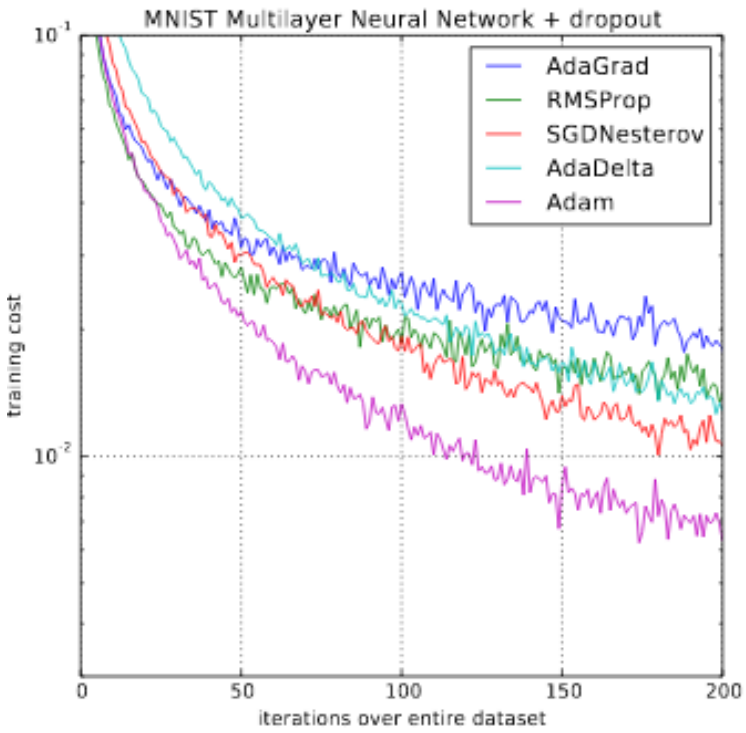

Figure 15

Cyclical learning rate

It is widely accepted that the learning rate is the most important hyperparameter for training any neural network effectively and quickly. The learning rate is used to control the amount by which the weights are adjusted in order to move towards the point of minimum cost.

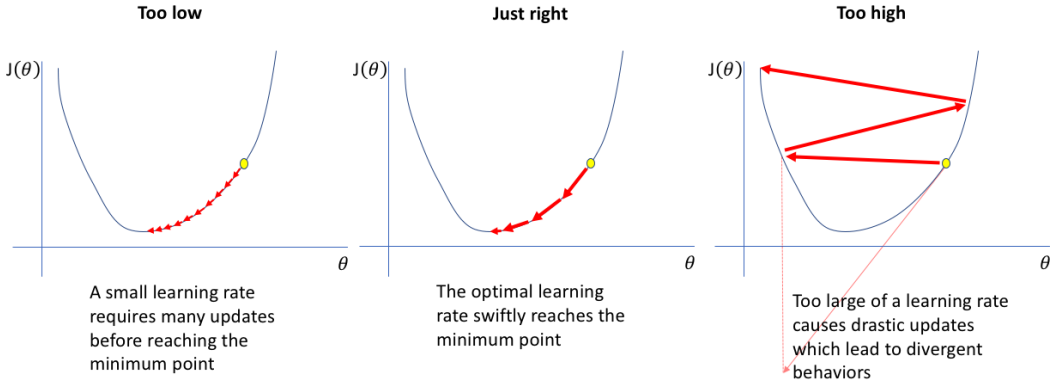

Figure 16: Effect of learning rate on gradient descent Source : Jeremy Jordan

The optimum learning rate generally has to be found experimentally with a trial-and-error method. However, Leslie N Smith, in his 2015 paper [3], stated that varying the learning rate cyclically within a set range was found to be beneficial overall and led to optimal accuracy in classification even without tuning the learning rate.

Working:

Varying the learning rate in bound range ensures that while the learning rate is never so high that the cost diverges from the minima, it also ensures that the periods of higher learning rate mean we do not get stuck at saddle points and local minima.

The learning rates can be varied in different methods within the range, such as triangular, parabolic or sinusoidal. However, it was found by Smith that all of them led to equivalent results. This led to the adoption of the triangular window, as it is the simplest function and implements the idea effectively.

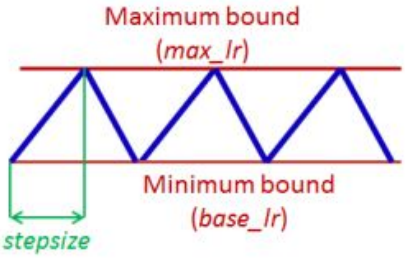

Figure 17: Triangular window for cyclical learning rate Source:[3]

Variations of the triangular window such as triangular2, in which the maximum learning rate is halved for each cycle, and exponential range, where the maximum learning rate is reduced exponentially for each cycle, can also be used.

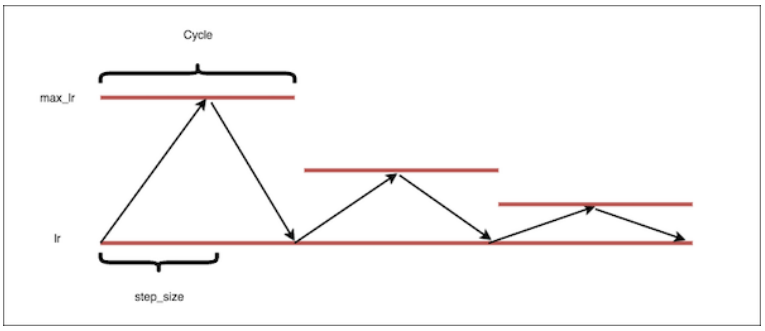

Figure 18: Traingular2

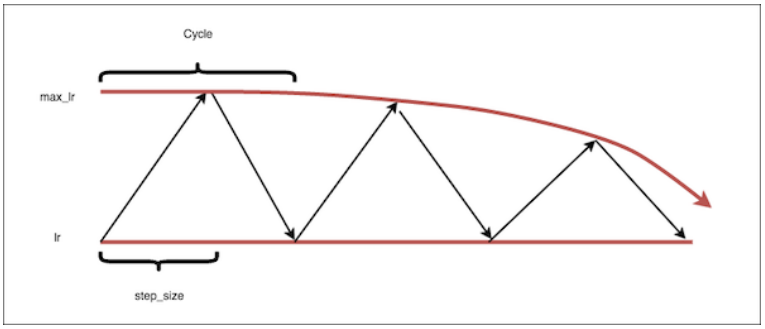

Figure 19: Exponential Range

5 Experimentation and results

5.1 Simulation and Data Generation

For the generation of data, we have used a MATLAB Simulink Model having a centralized PV Module topology with a set of three strings each consisting of three PV cells. The PV cells used while designing the Model are Simulink's inbuilt PV Array Blocks which take Irradiance in W/m2 and Temperature in degrees centigrade as inputs and can be connected in a circuit. The schematic of the Model is shown in the figure.

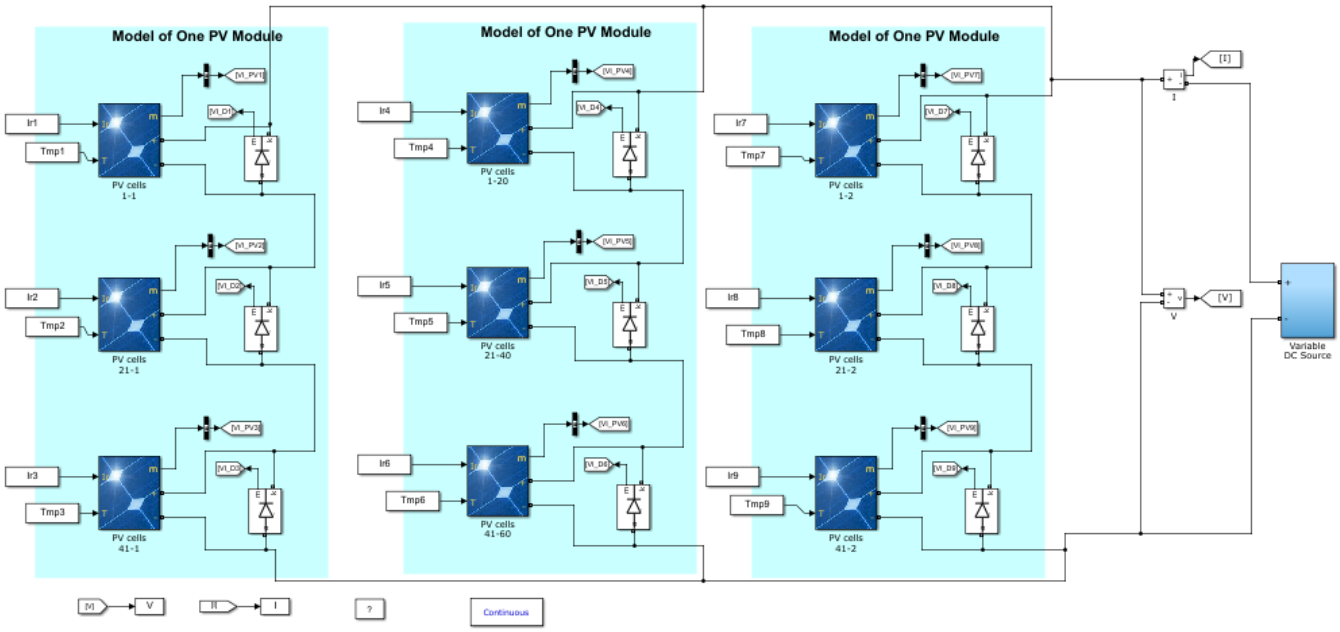

Figure 20: Simulink Model

For generating data for multiple values of Irradiance and Temperature we have used "from workspace" blocks as inputs for Irradiance and Temperature of each PV array block while feeding the values of irradiance with variables Ir1-Ir9 and Tmp1-Tmp9 for temperature through running ".m" script files where the model is run multiple times for different faults with different inputs and the corresponding outputs are stored each time in a ".txt" file to be used later for training the neural network. A snapshot of one of the script files is shown in figure.

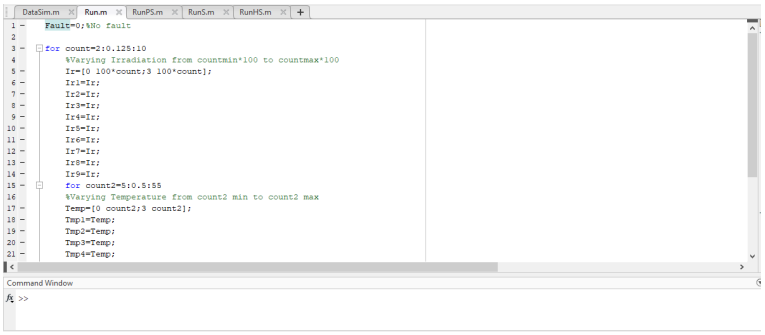

Figure 21: Matlab Script file code for Data Generation

The data was generated seperately for training and testing such that both of the datasets are overlapping with each other but not coinciding i.e. none of the data points in testing dataset is present in training dataset and visa-versa but the range of the data is nearly same. For generating the data we have varied the Irradiance for each PV Module from 300 W/m2 to 1000 W/m2 and temperature is varied for each set of Irradiances from 5 degrees Centigrade to 50 degrees Cntigrade with different step sizes for Training and Testing data such that there is no coincidence for different faults accordingly.

5.2 Data Preprocessing

The MATLAB scripts generate .txt files containing the data for different faults. The data from each .txt file is imported into a spreadsheet.Different sheets are made for training and testing data. Headers are entered for all the columns. The data is then normalised by using the following formula:

$$z = \frac{x \text{ - } \min(x)}{\max(x) \text{ - } \min(x)}$$

Figure 22: Normalization by feature scaling

5.3 Training and Testing the Model

The artificial neural network model used for our training consists of an input layer and 3 hidden layers, out of which one being the output layer. There are 21 parameters that are being given as inputs which include Pmax, Vmax, Imax, temperature and irradiance of the 9 panels. The two hidden layers in between is made up of 15 nodes each and use sigmoid as their activation function. The output layer consists of 4 nodes, with each of them corresponding to a fault. The activation function for the output layer is also sigmoid function.

Fault 0: 1 0 0 0 - Normal condition
Fault 1: 0 1 0 0 - Partial shading
Fault 2: 0 0 1 0 - Short circuit
Fault 3: 0 0 0 1 - Hotspot

The output is then predicted by returning the index of the maximum value in the array of 4 elements which will then correspond to the given fault. The MATLAB Simulink model generated a total of 20384 examples for all the faults. Now the training process involves iterating the generated examples until the cost function is made less than a certain value. This is done separately in three different optimization algorithms namely normal gradient descent, Adam optimizer and Cyclic learning rate algorithm.

The testing data consists of 4619 examples so as to be compatible with the rule of 80:20 - 80% being the training data and 20% being the testing data.

5.4 Results and Accuracy

Initially, the neural network is trained using normal gradient descent algorithm with learning rate set to 0.06. The model gave an accuracy of 96.62% in testing, with cost function reducing to 0.0598 after being trained for over 51000 epochs

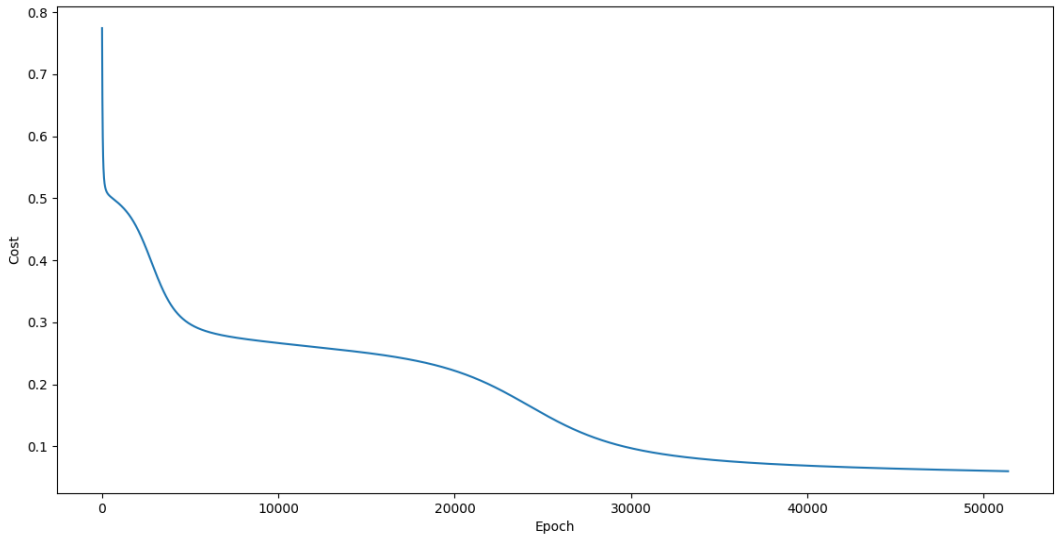

Figure 23: Gradient Descent

Then, the neural network was trained using ADAM optimiser. The accuracy obtained using this model was 99.95% in testing, with cost function reducing to a value of 0.0004 after being trained for close to 9400 epochs.

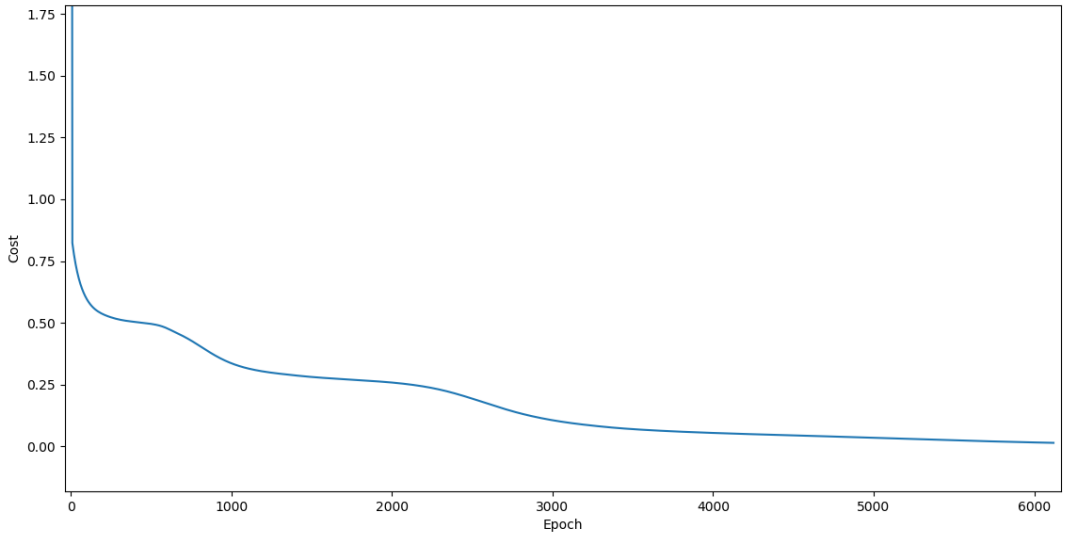

Figure 24: Adam Optimizer

Finally, the neural network was trained using cyclical learning rate algorithm. The learning rate bounds were set to 0.01 and 0.1 and the window used was triangular2. The accuracy obtained using this model was 99.956% during testing, with cost function reducing to a value of 0.012 after being trained for 845 epochs

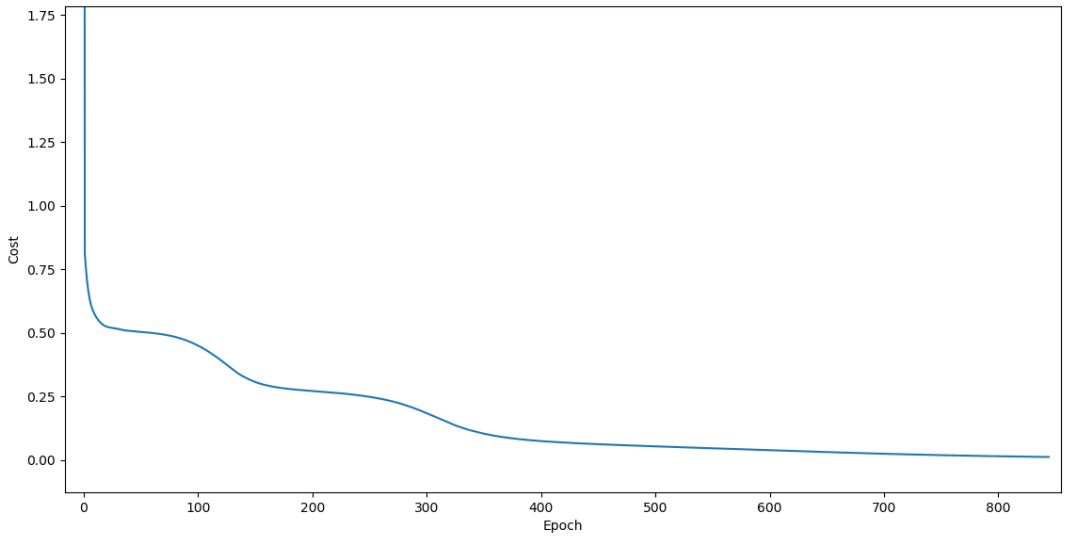

Figure 25: Cyclic Learning Rate

Conclusion

Using the neural network models described in the above sections, the identification of the presence of 3 different types of faults was demonstrated. The models had reasonably high accuracy, with notable improvement when ADAM optimiser and cyclical learning rates were used. The improvement is not only with respect to accuracy but also with respect to the time and the number of epochs that are needed for training. The cyclical learning rate algorithm is the most efficient in this scenario.

These models can be extended to detect and identify other types of faults in PV systems provided the datasets for those faults are available.

6 References

1. Lian Lian Jiang , Douglas L. Maskell "Automatic fault detection and diagnosis for photovoltaic systems using combined artificial neural network and analytical based methods"

2. Sunil Rao, Andreas Spanias and Cihan Tepedelenlioglu "Solar Array Fault Detection using Neural Networks"

3. Leslie N. Smith "Cyclical Learning Rates for Training Neural Networks"

4. N.D Marks, T.J. Summers2 and R.E. "Betz3Photovoltaic Power Systems: A Review ofTopologies, Converters and Controls"

5. A. F. Almarshoud "Technical and Economic Performance of 1MW Grid-connected PV system in Saudi Arabia"

6. Suresh B. Sadineni, Jonathan D. Realmuto and Robert F. Boehm "An Integrated Performance Monitoring and Solar Tracking System for Utility Scale PV Plants"

7. M. Sabbaghpur Arani and M. A. Hejazi "The Comprehensive Study of Electrical Faults in PV Arrays"