Different Types of Fault Detection of IEEE-9 Bus System using Neural Network

Md. Junaed Al-Hossain
North Carolina State University

Mst Shamima Hossain
Daffodil International University

Adrita Anika
Bangladesh University of Engineering and Technology

Tanney Chowdhury
Bangladesh University of Engineering and Technology

Project Report on

Different Types of Fault Detection of IEEE 9-Bus System using Neural Network

Group No.: 04

Course No.: EEE 306
Course Name: Power System I Laboratory

Submitted to:

ZABIR AHMED
SHOILIE CHAKMA

DEPARTMENT OF EEE
BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY

Submitted by:

TANNEY CHOWDHURY (1406123)
BIDYA DEBNATH (1406124)
ADRITA ANIKA (1406125)
SAJIBUL ISLAM (1406126)
MUAHTASIB SARKAR (1406127)
MST. SHAMIMA HOSSAIN (1406128)
MD. JUNAED-AL-HOSSAIN (1406129)
SK. HASIBUL ALAM (1406130)
ANANNA ISLAM (1306052)

Forwarding Letter

Date: *12th July, 2017*

Zabir Ahmed
Shoilie Chakma

Department of EEE
Bangladesh University of Engineering & Technology

Subject**: Submission of Project Report on "*Different Types of Fault Detection of IEEE-9 Bus System using Neural Network*"**

Dear Sir,

We are glad to present our project report entitled "Different Types of Fault Detection of IEEE-Bus System using Neural Network". We are grateful to you for giving us the opportunity to work on one of the major power section fault detection using a basic form of Artificial Intelligence commonly named "Neural Network". While working on this project we have not only developed our knowledge on fault analysis but also become familiar with Neural Network mechanisms. The project has enlarged our vision to apply our project for the salvation of the problem of fault detection of the over-all power system of our country.

We have tried our best to accomplish our project within given time. As we were new to Neural Network technic we have faced some limitations and could not fully perform the way we desired .But we hope to go farther with our project with filling up all our limitations in future .Yet we believe that our project report will help the readers to understand our works and views regarding the project.

Thank you for your consideration.

Sincerely---

Students of Power System Laboratory,
Group 4.

Contents

1. Introduction

Bangladesh has a power system of around 460 Buses supplying around 15,000 MW power (based on installed capacity) to meet the demand of day to day life of people of Bangladesh. The proper chain of generation and supply can be greatly hampered if different types of fault both symmetrical and unsymmetrical fault occurs due to various reason such as lighting, tree contact, transformer failure ,insulation damage etc. If such kind of fault occurs in power systems of hundreds of buses it is very difficult to recognize which type of fault has occurred and on which particular bus without actual inspection on the spot. After an immediate fault the repairing companies often have to inspect the fault sight in person and then take measures of repair .This causes a great time delay and brings about a great obstacle on production and domestic sections which can hardly go without continuous supply of power.

Many experienced power system engineers has been successful to predict the fault without being present on sight based on their experienced .This particular phenomenon inspired us to develop a system that stores data of different fault currents and voltages and using that database we can predict the probability of faults like whether the fault is Line to Ground (LG) or Line to Line (LL) etc. We have used "Neural Network" a form of artificial intelligence that can analyze provided data , build up a form of function and provide output corresponding to given input .We will provide fault voltage and fault current inputs and Neural Network will provide the probability of which fault may occur.

With the help of Neural Network on fault detection, the faulted bus and type of fault can be predicted and the repair companies can prepare themselves to take necessary measures to repair the faulted bus as repair mechanisms of each type of fault are different.

We have confined our project works on Line-Ground Fault (LG), Line-Line Fault (LL), Double Line-Ground Fault (LLG), and Triple Line-Ground Fault (LLLG) using sequence components of fault voltages as produced results have shown much accuracy unlike other corresponding inputs. Further development of the Neural Network and the database structure fault detection mechanism can be greatly developed.

2. Overview of Symmetrical Components

2.1. Fortescue's Theorem

An unbalanced system of *n* related phasors can be resolved into *n* systems of balanced phasors called the 'symmetrical components' of the original phasors. The *n* phasors of each set of components are equal in length, and the angles between adjacent phasors of the set are equal. In a three-phase system which is normally balanced, unbalanced fault conditions cause unbalanced currents and voltages. If the currents and voltages are related by constant impedances, the system is linear and the principle of superposition applies. The voltage response of the linear system to the unbalanced currents can be determined by considering the separate responses of the individual elements to the symmetrical components of the currents.

According to Fortescue's theorem, three unbalanced phasors of a three-phase system can be resolved into three balanced systems of phasors. The balanced set of components are:

1. *Positive-sequence components* consisting of three phasors equal in magnitude, displaced from each other by 120° in phase, and having the same phase sequence as the original phasors,
2. *Negative-sequence components* consisting of three phasors equal in magnitude, displaced from each other by 120° in phase, and having the phase sequence opposite to that of the original phasors,
3. *Zero-sequence components* consisting of three phasors equal in magnitude, and with zero phase displacement from each other.

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} V_a^{(0)} \\ V_a^{(1)} \\ V_a^{(2)} \end{bmatrix} = \mathrm{A} \begin{bmatrix} V_a^{(0)} \\ V_a^{(1)} \\ V_a^{(2)} \end{bmatrix}$$

Pre-multiplying both sides by $\mathbf{A}^{-1}$

$$\begin{bmatrix} V_a^{(0)} \\ V_a^{(1)} \\ V_a^{(2)} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

For currents, the equations are the same.

2.2. Introduction to Faults

There are different types of faults. Some of them are discussed below:

1) Single line to ground fault
2) Line-to-line fault
3) Double line to ground fault
4) Three phase fault
5) Sliding fault
6) Bolted fault etc.

The flow of current from each line into the fault is indicated by arrows shown on the diagram beside hypothetical stubs connected to each line at the fault location. Appropriate connections of the stubs represent the various types of fault.

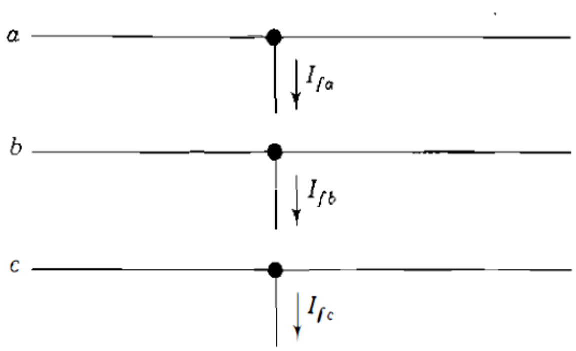


Figure 1: Connection diagram for hypothetical stubs

2.3. Description of the Faults

2.3.1. Single Line to Ground Fault: The single line-to-ground fault, the most common type, is caused by lightning or by conductors making contact with grounded structures. For a single line-to-ground fault through impedance Zf the hypothetical stubs on the three lines are connected, as shown in Figure, where phase "a" is the one on which the fault occurs.

Here,

$I_{fb} = 0, \quad I_{fc} = 0, \quad V_{ka} = Z_f I_{fa}$

$$\begin{bmatrix} I_a^{(0)} \\ I_a^{(1)} \\ I_a^{(2)} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}\begin{bmatrix} I_a \\ 0 \\ 0 \end{bmatrix}$$

$$I_a^{(0)} = I_a^{(1)} = I_a^{(2)} = \frac{1}{3} I_a$$

$$V_a^{(0)} = -I_a^{(0)} Z_0$$

$$V_a^{(1)} = E_a - I_a^{(1)} Z_1$$

$$V_a^{(2)} = -I_a^{(2)} Z_2$$

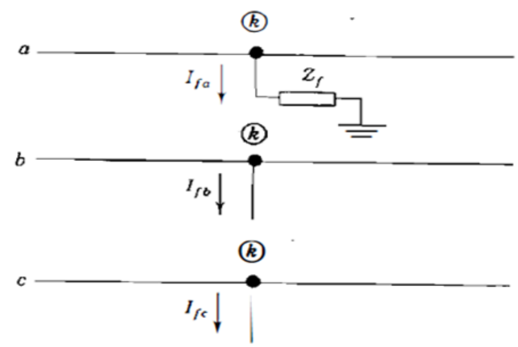


Figure 2: Single Line to Ground Fault

$$V_a = V_a^{(0)} + V_a^{(1)} + V_a^{(2)}$$

$$0 = -I_a^{(0)} Z_0 + E_a - I_a^{(1)} Z_1 - I_a^{(2)} Z_2$$

$$= E_a - I_a^{(1)} (Z_0 + Z_1 + Z_2)$$

$$I_a^{(1)} = \frac{E_a}{Z_0 + Z_1 + Z_2}$$

Here, Z_f is assumed to be zero. It should be noted that about 70 to 80 percent of transmission line faults is single line to ground fault.

2.3.2. Line to Line Fault: To represent a line-to-line fault through impedance Z_f, the hypothetical stubs on the three lines at the fault are connected, as shown in Figure. Bus k is again the fault point P, and without any loss of generality, the line-to-line fault is regarded as being on phases "b" and "c ".The following relations must be satisfied at the fault point,

$I_{fa} = 0, \qquad I_{fb} = -I_{fc}\ , \qquad V_{kb} - V_{kc} = I_{fb}Z_f$

Conditions at the fault:

$$V_b = V_c, \quad I_a = 0, \quad I_b = -I_c$$

$$V_a^{(0)} = -I_a^{(0)} Z_0$$

$$V_a^{(1)} = E_a - I_a^{(1)} Z_1$$

$$V_a^{(2)} = -I_a^{(2)} Z_2$$

$$\begin{bmatrix} I_a^{(0)} \\ I_a^{(1)} \\ I_a^{(2)} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}\begin{bmatrix} 0 \\ I_b \\ -I_b \end{bmatrix}$$

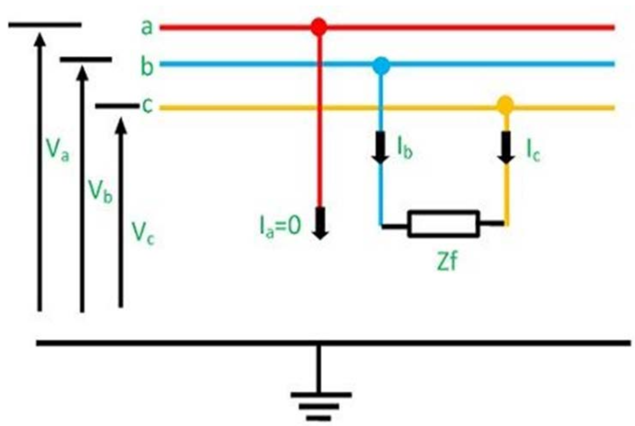


Figure 3: Line to Line Fault

$$I_a^{(0)} = 0$$

$$I_a^{(1)} = -\ I_a^{(2)}$$

$$\begin{bmatrix} V_a^{(0)} \\ V_a^{(1)} \\ V_a^{(2)} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a & a \end{bmatrix}\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

$$V_a^{(1)} = V_a^{(2)}$$

$$E_a - I_a^{(1)} Z_1 = -I_a^{(2)} Z_2$$

$$E_a - I_a^{(1)} Z_1 = I_a^{(1)} Z_2$$

$$I_a^{(1)} = \frac{E_a}{Z_1 + Z_2}$$

2.3.3. Double Line to Ground Fault: For a double line-to-ground fault the hypothetical stubs are connected as shown in Figure. Again, the fault is taken to be on phases "b" and "c", and the relations now existing at the fault bus k are,

$$I_{fa} = 0$$

$$V_{kb} = V_{kc} = (I_{fb} + I_{fc})\, Z_f$$

$$I_{fa}^{(0)} = (I_{fb} + I_{fc})/\ 3$$

$$V_{kb} = V_{kc} = 3Z_f\, I_{fa}^{(0)}$$

$$\begin{bmatrix} V_{ka}^{(0)} \\ V_{ka}^{(1)} \\ V_{ka}^{(2)} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}\begin{bmatrix} V_{ka} \\ V_{kb} \\ V_{Kb} \end{bmatrix}$$

$$V_{ka}^{(1)} = V_{ka}^{(2)}$$

$$3V_{ka}^{(0)} = V_{ka} + 2V_{kb} = (V_{ka}^{(0)} + V_{ka}^{(1)} + V_{ka}^{(2)}) + 2(3Z_f I_{fa}^{(0)})$$

$$V_{ka}^{(1)} = V_{ka}^{(0)} - 3Z_f I_{fa}^{(0)}$$

$$V_{ka}^{(1)} = V_{ka}^{(2)} = V_{ka}^{(0)} - 3Z_f\, I_{fa}^{(0)}$$

$$I_{fa} = 0; \quad I_{fa}^{(0)} + I_{fa}^{(1)} + I_{fa}^{(2)} = 0$$

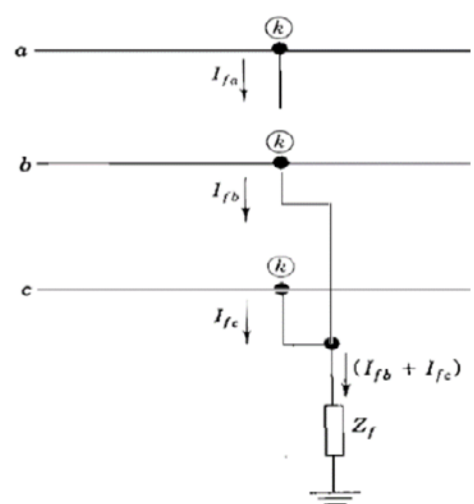


Figure 4: Double Line to Ground Fault

Here,

$$I_{fa}^{(1)} = \frac{V_f}{Z_{kk}^{(1)} + \left[\dfrac{Z_{kk}^{(2)}\left(Z_{kk}^{(0)} + 3Z_f\right)}{Z_{kk}^{(2)} + Z_{kk}^{(1)} + 3Z_f} \right]}$$

$$I_{fa}^{(2)} = -I_{fa}^{(1)} \left[\frac{Z_{kk}^{(0)} + 3Z_f}{Z_{kk}^{(2)} + Z_{kk}^{(0)} + 3Z_f} \right]$$

$$I_{fa}^{(0)} = -I_{fa}^{(1)} \left[\frac{Z_{kk}^{(2)}}{Z_{kk}^{(2)} + Z_{kk}^{(0)} + 3Z_f} \right]$$

2.3.4. Three Phase Fault: A balanced system remains symmetrical after the occurrence of a three phase fault having the same impedance between each line and a common point. Only positive-sequence currents flow. With the fault impedance Zf equal in all phases,, we simply add impedance Zf to the usual (positive-sequence) Thevenin equivalent circuit of the system at the fault bus k and calculate the fault current from the equation:

$$I_{fa}^{(1)} = \frac{V_f}{Z_{kk}^{(1)} + Z_f}$$

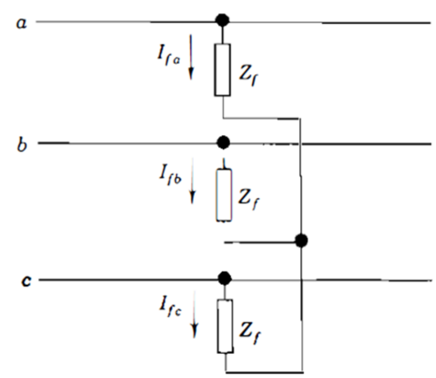


Figure 5: Three Phase Fault

It should be noted that, about 5 percent of total transmission faults is three phase fault.

2.3.5. Sliding Fault: It is the type of fault that occurs at a percentage distance of the transmission line. It is actually the four types of faults discussed earlier. In accordance with distance, just the types of fault and their combinations are changed.

2.3.6. Bolted Fault: When the value of Z_f is zero, the four types of faults are called bolted fault. In this fact, the current becomes huge. This type of fault seldom happens.

3. Artificial Neural Networking (ANN)

ANN implementation is used to design the best ANN configuration. The process of ANN implementation starts from data collection and ends with the Performance evaluation of ANN. Percentages of classification accuracy and mean square error are used to represent the performance of ANN in terms of accuracy to predict the status of the IEEE 9- bus system.

3.1. Generation of Training Data

Input data sets for ANN training are generated from Newton-Raphson load flow analysis by varying both real and reactive loads at all the buses randomly in the range of 5% to 100% and -10% to -100% of their base case value. In data collection, the input data are divided into train data, validation data and test data. NR load flow analysis is conducted at all steps and corresponding voltage at each bus for different faults are calculated. The PSAF was used as a computing tool. Collection of these data constitutes the training data set.

3.2. ANN Structure

A multi-layered feed-forward neural network has been proved suitable for most power system problems. Figure1 shows the architecture of the feed forward neural network. The architecture of the ANN used in this paper consists of an input layer, a hidden layer and an output layer. The input layer has 6 neurons since the number of variables in the input neural network is 6 (voltage magnitude and angles of the three sequence components). The number of hidden neuron in hidden layer is 100. The more hidden neurons are used to train the neural network, the more computational time will be consumed. In the target layer, the neural network has 36 output vectors since we have 36 output classes.

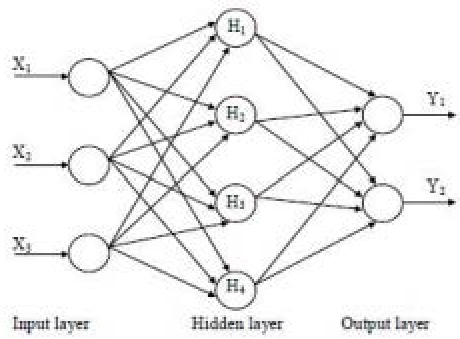


Figure 6: Architecture of feed forward neural network

3.3. ANN Training

Training process of neural network helps to identify the topology of neural network and it's interconnected Weights. The training speed depends on the speed factor such as the learning rule, the transfer function of neurons. In the training process of the neural network, a set of network inputs and target outputs are required. And also it requires enough information in order to simulate a good prediction of power system status during training process. The weights and biases of the network are iteratively adjusted to minimize the network performance function during training process itself. The feed forward back propagation neural network can be trained with different training algorithms. The most commonly used training algorithm for multi-layer feed forward network is back propagation (BP) algorithm, which is a gradient descent algorithm.

(Since this method is too slow, some high performance algorithms like conjugate gradient algorithms, Quasi-Newton algorithms, Levenberg-Marquardt (LM) algorithm are developed to train network which converges faster than BP algorithm. In the present work, LM training technique is used due to its faster training and good convergence .And this algorithm is suitable for medium sized neural network. And this algorithm combines steepest descent algorithm and the Gauss–Newton algorithm.)

In neural network, over fitting is also known as overtraining where further training will not result in better generalization. The error of validation set is periodically monitored during the training process. The training error usually decreases as the iteration grows, so does the validation error. When the overtraining starts to occur, the validation error typically tends to increase. Therefore, it would be useful and time saving to stop the training after the validation has increased for some specified number of iteration. The whole ANN process can be depicted as shown below in Figure 7:

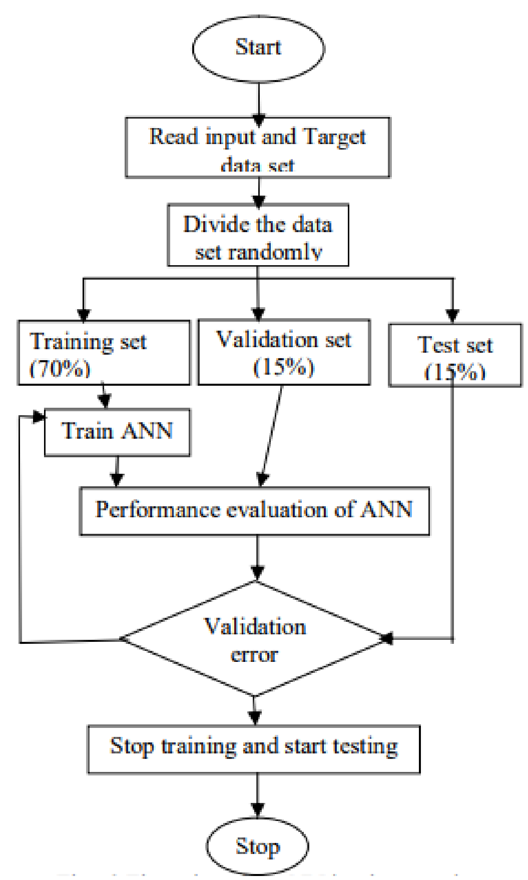


Figure 7: Flow chart for ANN implementation

3.4. Performance of ANN

The performance of the feed forward back propagation neural network is evaluated by the percentages of Classification Accuracy (CA) and mean squared error. The percentages of CA for the neural network are calculated by using the following equation:

$$CA\ (\%) = \frac{number\ of\ cases\ classified\ accurately}{total\ number\ of\ cases} \times 100\%$$

The neural network configuration which gives highest accuracy of prediction and lowest value of mean square error is considered as the best performance configuration.

4. IEEE 9-Bus System

IEEE bus systems are used by researchers to implement new ideas and concepts. The system consists of loads, transmission lines, and generators as shown in Figure 3.

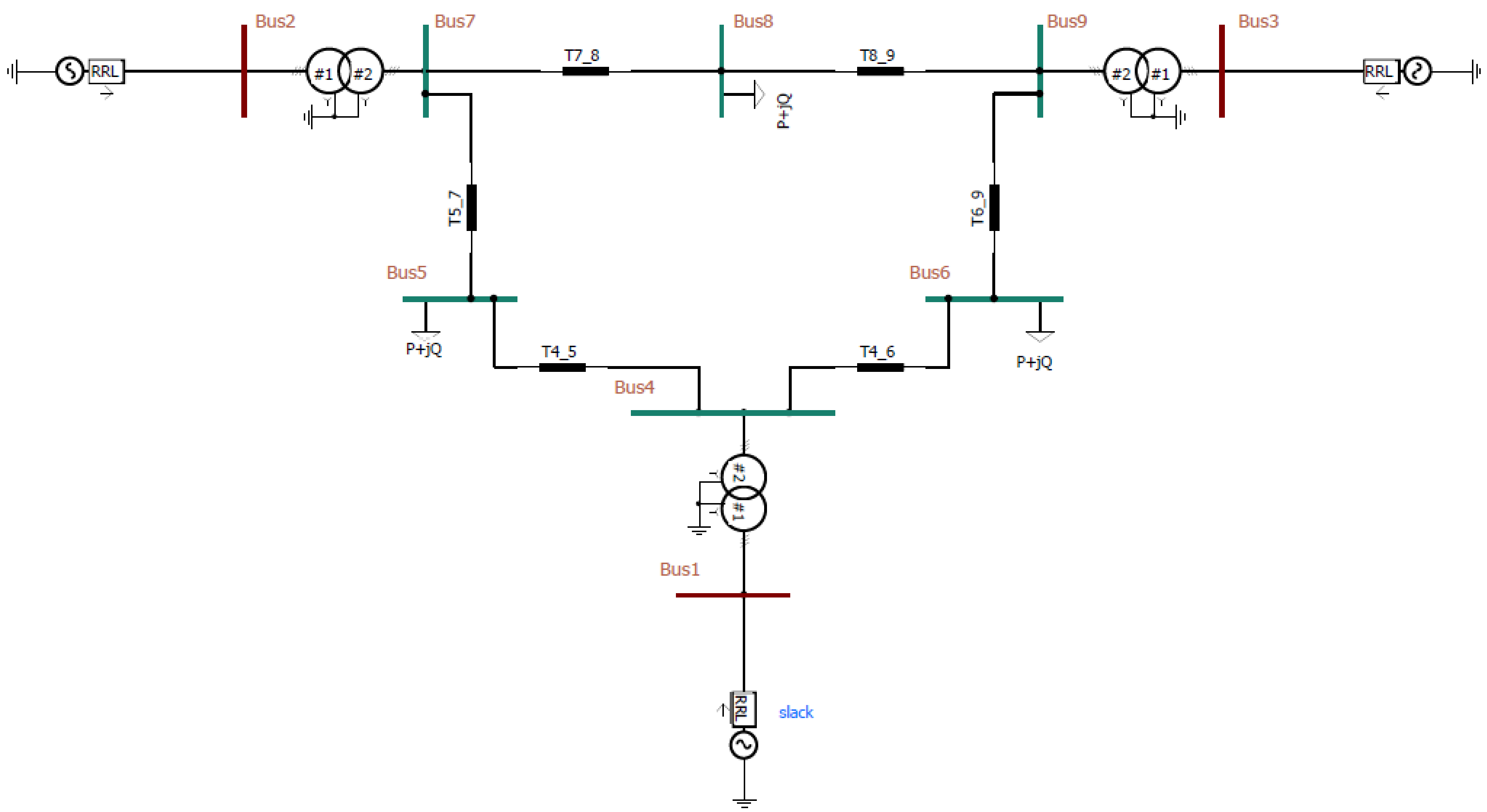


Figure 8: PSACD model of IEEE 9-bus system

4.1. Data

Each machine (generator) is represented as a voltage source where its source impedance is set arbitrarily as 1 Ohm. Table 1 summarizes per unitized terminal conditions of each source, with 100 [MVA] base.

Table 1 - Terminal conditions of IEEE 9-bus system

Bus	V [kV]	δ [deg]	P [pu]	Q [pu]
1	17.1600	0.0000	0.7163	0.2791
2	18.4500	9.3507	1.6300	0.0490
3	14.1450	5.1420	0.8500	-0.1145

Transmission lines are modelled using the Bergeron model. Table 2 summarizes the transmission line parameters.

Table 2 - Transmission line characteristics of IEEE 9-bus system

Line		R [pu/m]	X [pu/m]	B [pu/m]
From Bus	To Bus			
4	5	0.0100	0.0680	0.1760
4	6	0.0170	0.0920	0.1580
5	7	0.0320	0.1610	0.3060
6	9	0.0390	0.1738	0.3580
7	8	0.0085	0.0576	0.1490
8	9	0.0119	0.1008	0.2090

Loads are modelled as a constant PQ load with parameters as shown in Table 3.

Table 3 - Load characteristics of IEEE 9-bus system

Bus	P [pu]	Q [pu]
5	1.25	0.50
6	0.90	0.30
8	1.00	0.35

4.2. Validation

The PSCAD model was validated against the PSS/E power flow values from [1]. Table 4 depicts the line and source power flow comparison.

Table 4 - Source and line power flow comparison of IEEE 9-bus system

Bus		PSS/E		PSCAD	
		P [pu]	Q [pu]	P [pu]	Q [pu]
1		0.716	0.279	0.7152	0.2761
2		1.630	0.049	1.6320	0.0454
3		0.850	-0.114	0.8512	-0.1170
From Bus	**To Bus**				
4	5	0.433	0.235	0.4322	0.2334
4	6	0.283	0.013	0.2830	0.0115
5	7	0.842	-0.104	0.8430	-0.1041
6	9	0.633	-0.178	0.6340	-0.1810
7	8	0.788	-0.008	0.7892	-0.0089
8	9	0.217	0.023	0.2172	0.0229

5. PSAF

The Power Systems Analysis Framework (PSAF) is a suite of modular analysis programs with a common database and network editing facility. The suite includes programs for Load Flow, Short Circuit, Harmonic and Transient Stability analyses of electrical networks.

The highly interactive graphical interface makes it easy to draw the network One-Line Diagram on the screen and define the parameters of its components. As an alternative to the graphic method of entering the network description, PSAF allows one to enter data in tables. In this case, the One-Line Diagram is drawn automatically as one works.

5.1. Analysis

All analysis modules include the following three basic dialog boxes:

- **"Study" dialog box:** to set the study parameters such as MVA tolerance, number of iterations, System frequency, etc.
- **"Solve" dialog box:** to solve the network with the respective calculation engine for Power Flow, AC Contingency, Short Circuit, Motor Starting, Transient Stability or Harmonic Analysis.
- **"Unit" dialog box:** to set unit for each parameters in solution reports.

To create a new PSAF file and do simulations, the following steps need to be followed:

1. A new folder in any drive of PC was created.
2. The software PSAF 2.81 was opened. From ‘Database’ menu, ‘New Blank Database Directory’ was selected to show the path of the folder created in step 1.
3. From ‘File’ menu, ‘Preferences’ was selected to open ‘Preference settings’ window. Under ‘Directory’ tab, ‘DATABASE’ under ‘Item’ was selected and ‘Browse’ was clicked to show the ‘Path’ of the folder created in step 1.The same was done for ‘NETWORK’ under ‘Item’.
4. From ‘File’ menu, ‘New Study’ was selected. In ‘Loading Network Data’ window, ‘Use tabular format’ was selected for network data presentation and ‘OK’ was clicked. Next ‘creating a new network’ was selected in ‘New Study Dialog’ window and ‘Default 1 bus’ in ‘New Network Dialog’ window.
5. The network was created by using tabular format.

5.2. Fault Analysis

1. Using the information given in the figure, the network shown in figure was created.
2. From ‘Analysis’ menu, ‘Select Solver’ was clicked. ‘Fault Analysis ANSI’ in ‘Engine Selection Dialog’ window was selected.
3. For analyzing fault on bus:
    a) From ‘Analysis’ menu, ‘Study’ was selected. In ‘Short Circuit ANSI Study Dialog’ window under the ‘Fault Analysis Parameters’ tab, select ‘Fault Selected Bus(es)’ under ‘Shunt Fault’. Select ‘LLL’, ‘LG’, ‘LL’, and ‘LLG’ under ‘Fault type’. ‘Ok’ was clicked.
    b) From ‘Analysis’ menu, ‘Solve’ was selected. In ‘Solving for Fault Analysis’ window, B1, B2, B3, B4, B5, B6 under ‘Select Bus(es)’ was selected. ‘Solve!’ was clicked.
4. For analyzing sliding fault on lines:
    a) From ‘Analysis’ menu, ‘Study’ was selected. In ‘Short Circuit ANSI Study Dialog’ window under the ‘Fault Analysis Parameters’ tab, ‘Sliding fault on selected Line’ was selected and the fault location (as % of line length) under ‘Shunt Fault’ was specified. ‘LLL’, ‘LG’, ‘LL’, ‘LLG’ under ‘Fault type’ was selected. ‘Ok’ was clicked.
    b) From ‘Analysis’ menu, ‘Solve’ was selected. In ‘Solving for Fault Analysis’ window, L14, L16, L32, and L52 under ‘Select Line(s) or Cable(s)’ was selected. ‘Solve!’ was clicked.
5. From ‘Report’ menu, ‘Show Text Report’ was selected. Reports appeared on the bottom part of the window.
6. The same procedure was repeated for 216 times by varying the loads between 5% to 100% range.

5.3. Short Circuit (ANSI Version)

CYMFAULT - ANSI offers two methods for performing fault analysis:

- ANSI follows the American National Standards Institute standards for circuit breaker application C37.010 (symmetrical current basis), C37.5 (total current basis) and C37.13 (low voltage circuit breakers):
  - It calculates four specific duty types, according to the standards, and applies multipliers to the calculated currents to account for asymmetry (DC component).
  - It adjusts the reactance of motors according to their size and speed to account for the fact that their contributions to faults decay with time.
  - It does not permit the inclusion of pre-fault load current.
  - It does allow the use of load flow solution voltages.
  - It does not permit an impedance in the fault itself. Only "bolted" faults are permitted.
- Conventional (non-ANSI) does not follow the ANSI standards:
  - It does not adjust motor reactance, but does allow to use transient impedances of generators to calculate the current a few cycles after fault inception.
  - It allows to include pre-fault load current and a fault impedance.
  - It also can compute voltages and currents due to series faults (open circuits).

5.4. Units for Reports (Short Circuit)

It may be chosen to report the voltages, currents and impedances as phase quantities and/or in terms of sequence components. In the Labels it may be chosen which phase(s) or which sequence(s) to display. Labels can show results for more than one phase (ABC) and more than one sequence (zero, positive, negative).

It may also be chosen to report values in polar form (magnitude and phase angle) or in rectangular form (real and imaginary parts). We have used values in per unit.

6. Procedure

We have chosen IEEE 9 bus system for our project and considered four types of faults i.e. LLL, LG, LL and LLG. We trained our network using voltage magnitude and angles of the three phase components of each 9 bus of the 9-bus system. So we have 36 classes to classify (4 types of fault at each 9 bus). For generating data, we used PSAF CYME software. We have done 216 times fault analysis. To make our input matrix suitable for the analysis, MATLAB has been used. The code is provided below:

```
%%
clear all
close all
clc
%%
load rawVoltageData.mat
%%

iMax = size(faultVoltages, 1);
iDel = 324;
nTrain = size(faultVoltages, 1)/9;
D = size(faultVoltages, 2)*9;
C = 36;

voltageData = zeros(nTrain, D);
labels = zeros(nTrain, C);
y = zeros(nTrain, 1);

for i = 1:iDel:iMax
    iStart = i;
    iEnd = i + iDel - 1;
    subMatFaultVoltages = faultVoltages(iStart:iEnd, :);

    iLLL = 1:4:324;
    iLG = iLLL + 1;
    iLL = iLLL + 2;
    iLLG = iLLL + 3;

    % B = reshape(A', 6, 10)'
    iStart = (iStart - 1)/9 + 1;
    voltageData(iStart:(iStart + 8), :) = reshape(subMatFaultVoltages(iLLL,
:)', 54, 9)';
    y(iStart:(iStart + 8)) = 1:9;

    voltageData((iStart+9):(iStart + 17), :) =
reshape(subMatFaultVoltages(iLG, :)', 54, 9)';
    y((iStart+9):(iStart + 17)) = 10:18;

    voltageData((iStart+18):(iStart + 26), :) =
reshape(subMatFaultVoltages(iLL, :)', 54, 9)';
    y((iStart+18):(iStart + 26)) = 19:27;

    voltageData((iStart+27):(iStart + 35), :) =
reshape(subMatFaultVoltages(iLLG, :)', 54, 9)';
    y((iStart+27):(iStart + 35)) = 28:36;
end
linearIndices = sub2ind(size(labels), (1:nTrain)', y);
labels(linearIndices) = 1;
```

The input matrices are named as 'voltgeData' and 'labels'. The "voltageData" is a 7776*54 matrix. The 54 columns represent 6 data (magnitude and angle of each sequence) i.e. 54 (6 x 9) for 9 buses. The number of rows are 7776 (i.e. 216 x 36) since we have done 216 times analysis for 36 classes. The 'labels' matrix is a 7776 x 36 matrix because of the same reason. The labels are from 1 to 36 where a particular number indicates a particular fault at a particular bus. It is mentioned below:

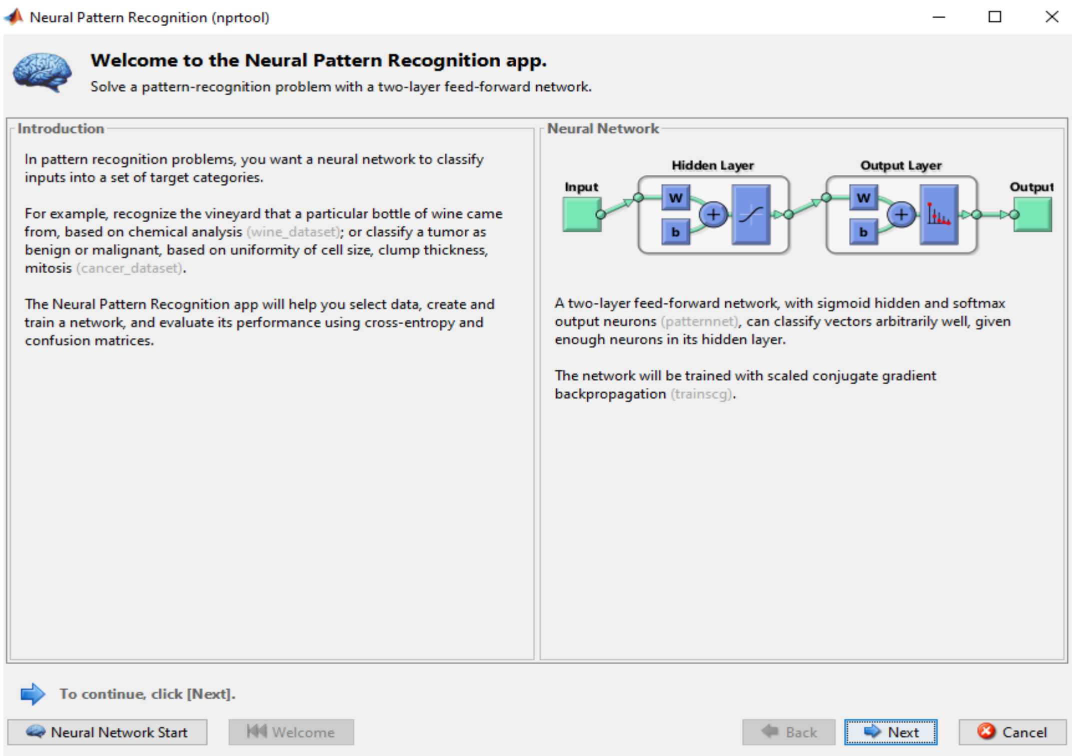
Class no.	Faulted Bus no. and Type of Fault
1	B1 LLL
2	B2 LLL
3	B3 LLL
4	B4 LLL
5	B5 LLL
6	B6 LLL
7	B7 LLL
8	B8 LLL
9	B9 LLL
10	B1 LG
11	B2 LG
12	B3 LG
13	B4 LG
14	B5 LG
15	B6 LG
16	B7 LG
17	B8 LG
18	B9 LG
19	B1 LL

20	B2 LL
21	B3 LL
22	B4 LL
23	B5 LL
24	B6 LL
25	B7 LL
26	B8 LL
27	B9 LL
28	B1 LLG
29	B2 LLG
30	B3 LLG
31	B4 LLG
32	B5 LLG
33	B6 LLG
34	B7 LLG
35	B8 LLG
36	B9 LLG

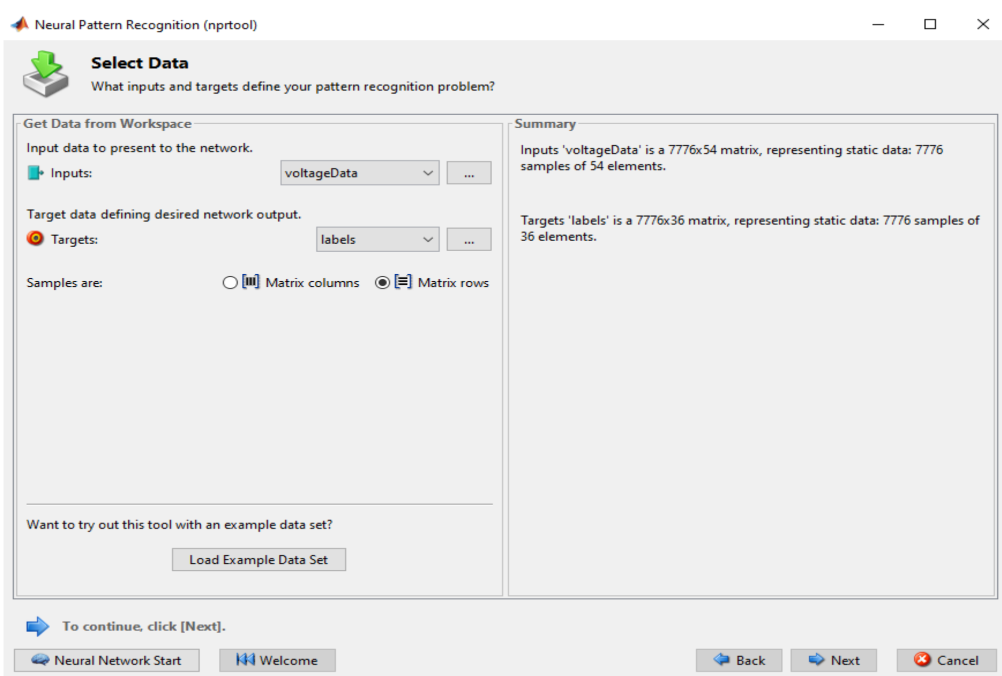
After formation of the input matrix, we have used the MATLAB app Neural Networking Pattern recognition for the analysis. Firstly input data are selected for the analysis. 70% of the data are for training, 15% for validation and 15% for testing. We have chosen 100 as the number of hidden layers. We have done the analysis for several number of layers and considered it as a good selection for higher accuracy.

We have done the analysis in MATLAB using its app function “Neural Network Pattern Recognition”.

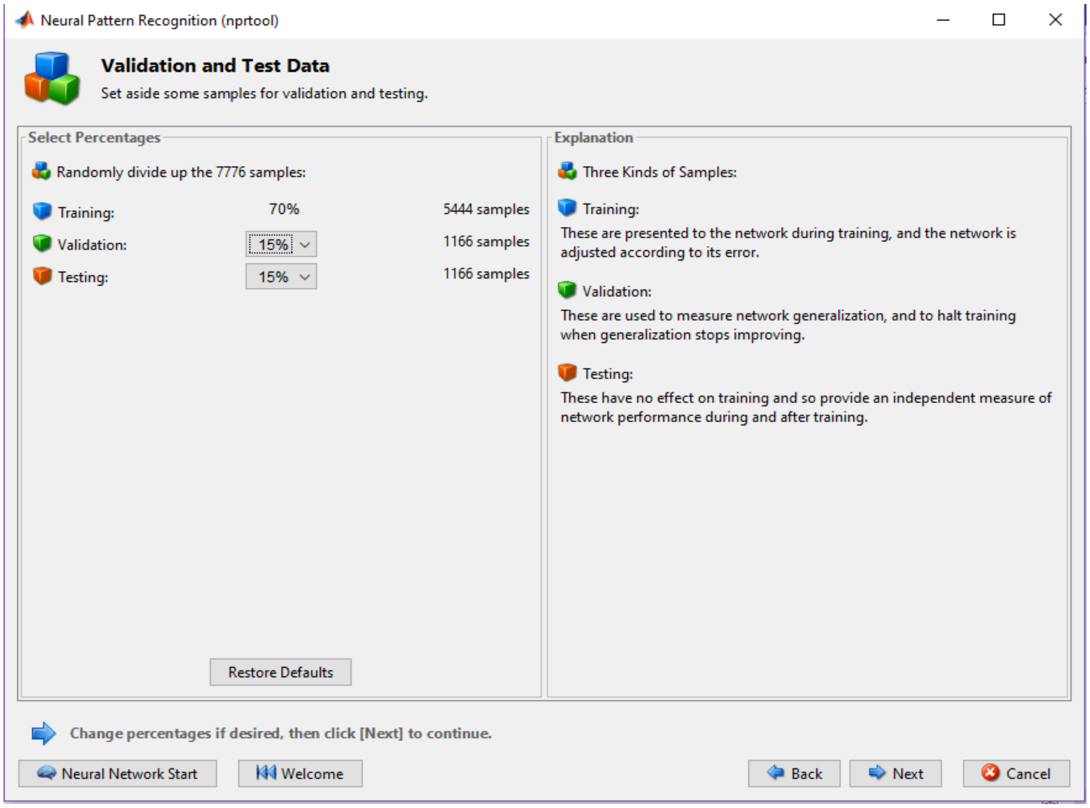
It solves problem with a two layer feed-forward system.



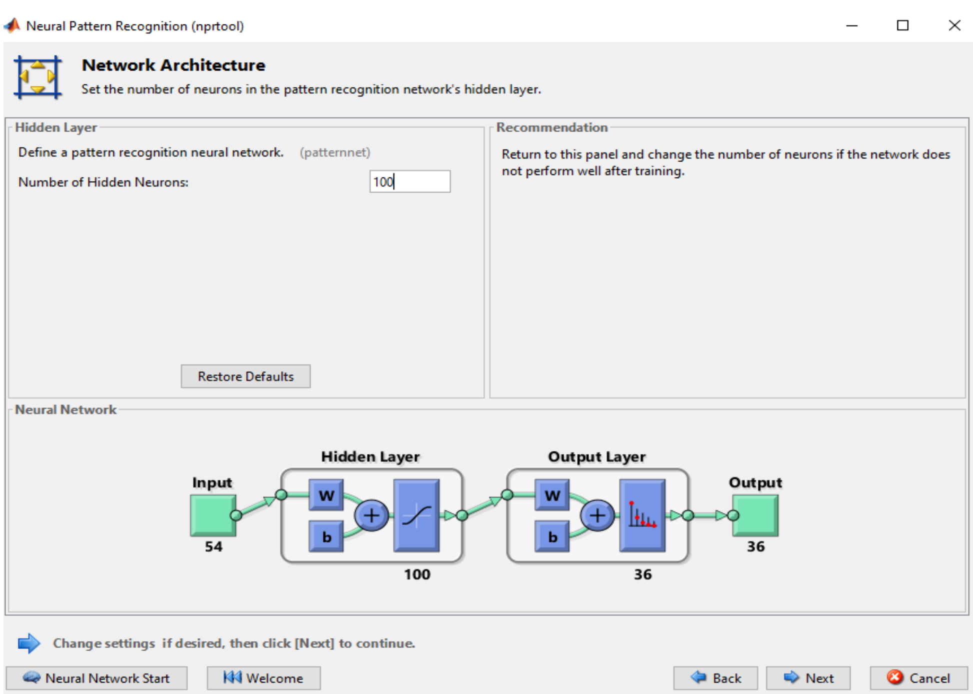
Then input matrix and label matrices are selected.



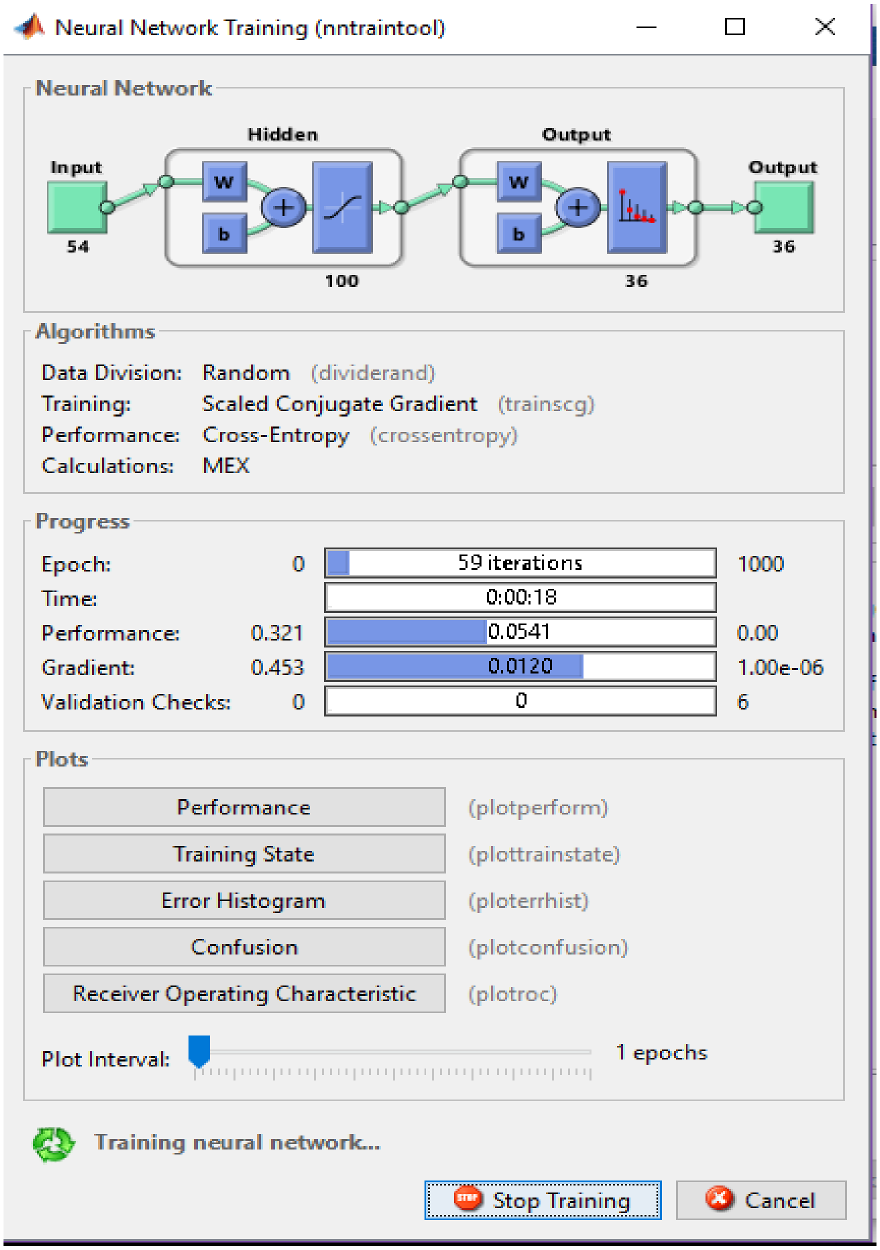
From the samples, 70% are selected as training samples and 15% for validation and 15% for testing.



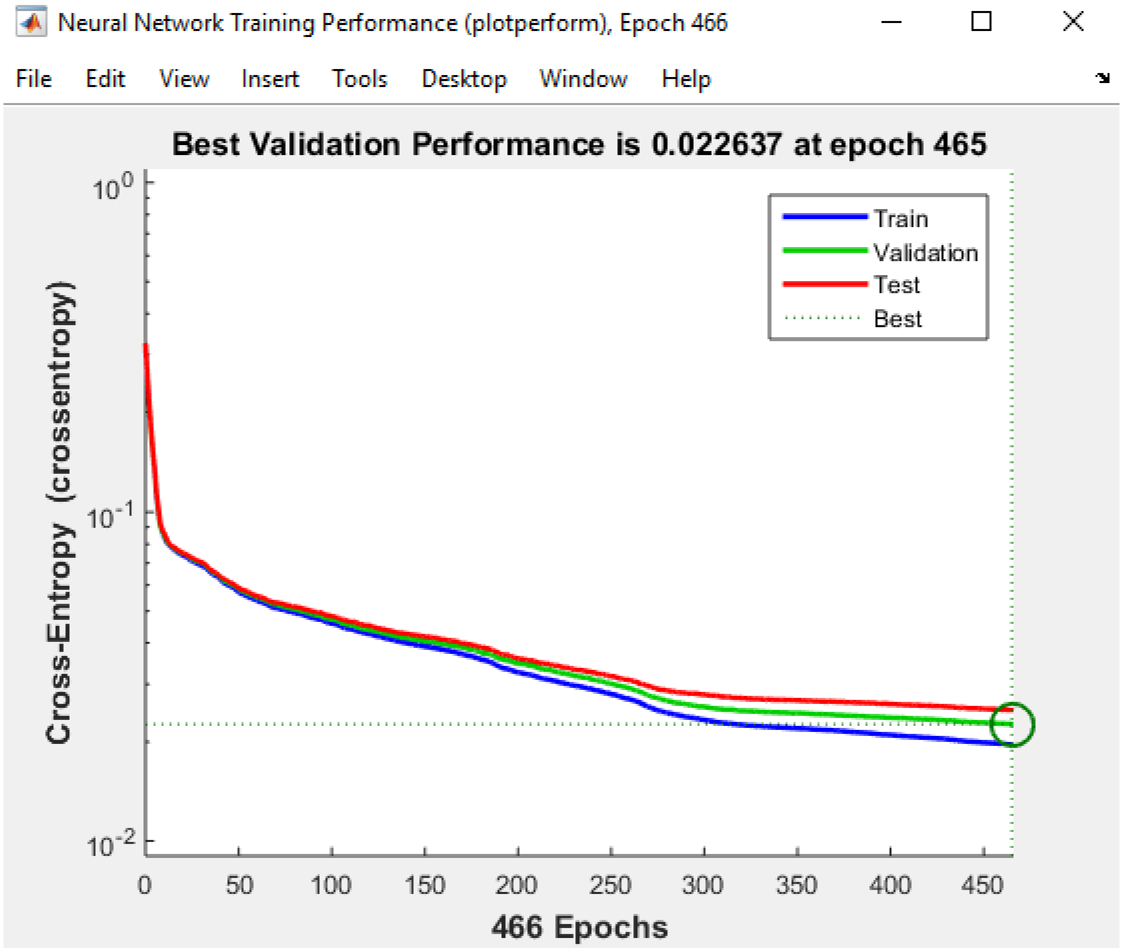
We chose 100 as the number of hidden layers.



Then we trained the network.



While training, we can see the performance plot and see that cross-entropy error is being reduced with increasing number of epochs.



Thus our analysis is done.

7. Performance

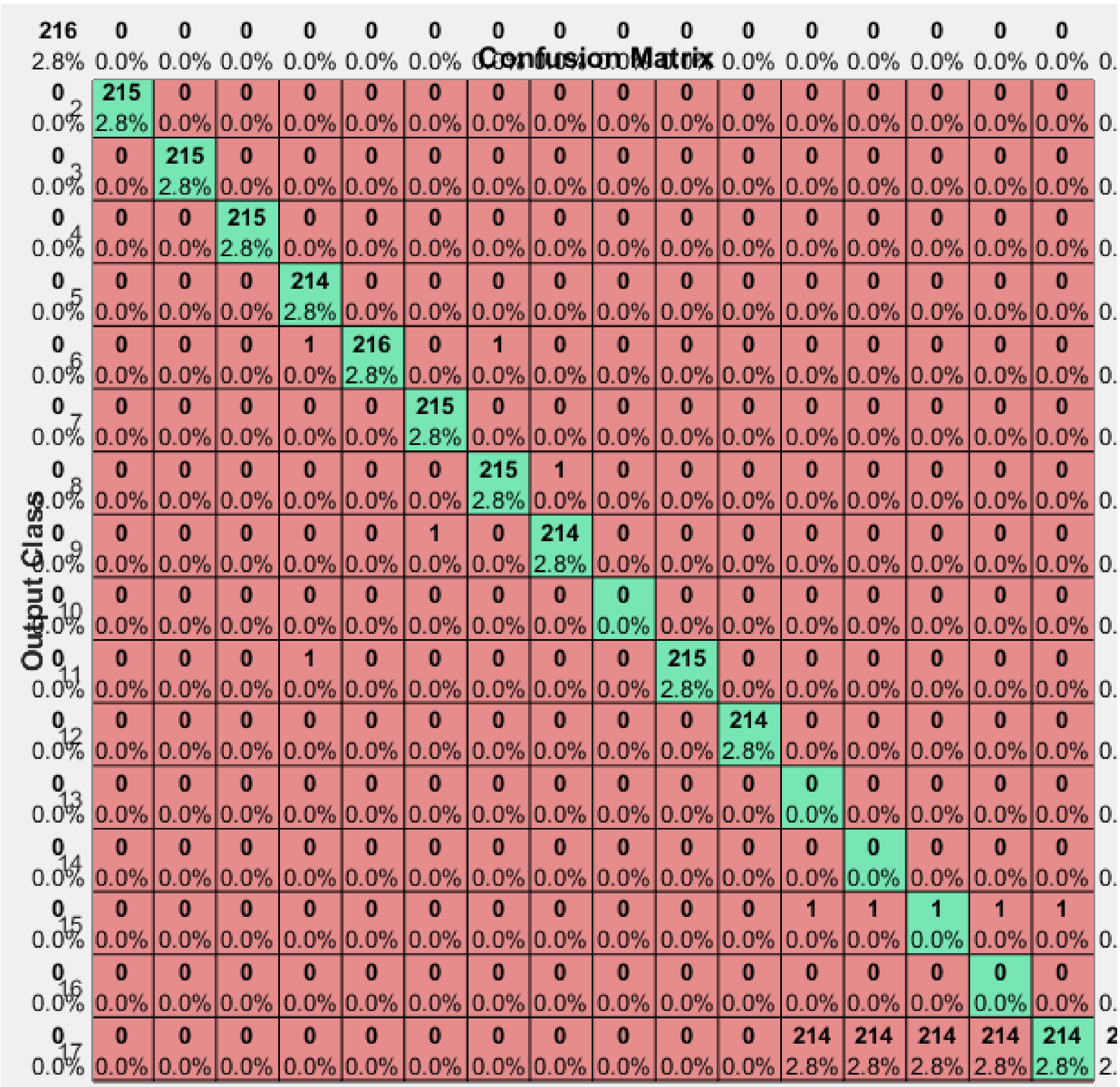
Results

	Samples	CE	%E
Training:	5444	7.34262e-0	20.09551e-0
Validation:	1166	20.76016e-0	18.18181e-0
Testing:	1166	20.77096e-0	20.24013e-0

The last column shows the result of the analysis. We can see that testing data has approximately 80% accuracy. It slightly differs for various analysis since the analysis is started form different arbitrary points at different time. Additional data can also be tested.

8. Limitation

We have 36 classes in our project. From the Confusion matrix, we can see it cannot classify the 14 to 18 no classes (i.e. LG fault in B4, B5, B6, B7, B8 buses) properly.



We think this is because the voltage magnitude and angle at the buses for LG faults in the corresponding buses (B4, B5, B6, B7, and B8) are equal or nearly equal. (PSAF voltage report for 2 buses are attached with). That is why our network could not classify this properly.

8.1. Scope of Improvement

We have chosen the voltage magnitude and angles of sequence components for each bus as our input data. If extra features (such as current magnitude and angles etc.) are included, the accuracy would be increased.

9. Voltage Report

B4	B1	16.5	LLL	0	0	0.4316	0.5841	0	0
B4	B1	16.5	LG	0	0	1	0	0	0
B4	B1	16.5	LL	0	0	0.7158	0.1761	0.2842	-0.444
B4	B1	16.5	LLG	0	0	0.7158	0.1761	0.2842	-0.444
B4	B2	18	LLL	0	0	0.6655	-0.8696	0	0
B4	B2	18	LG	0	0	1	0	0	0
B4	B2	18	LL	0	0	0.8327	-0.344	0.1674	1.7119
B4	B2	18	LLG	0	0	0.8327	-0.344	0.1674	1.7119
B4	B3	13.8	LLL	0	0	0.493	-1.1856	0	0
B4	B3	13.8	LG	0	0	1	0	0	0
B4	B3	13.8	LL	0	0	0.7465	-0.3915	0.2536	1.1522
B4	B3	13.8	LLG	0	0	0.7465	-0.3915	0.2536	1.1522
B4	B4	230	LLL	0	0	0	0	0	0
B4	B4	230	LG	1	180	1	0	0	0
B4	B4	230	LL	0	0	0.5	0	0.5	0
B4	B4	230	LLG	0.5	0	0.5	0	0.5	0
B4	B5	230	LLL	0	0	0.1722	-4.7632	0	0
B4	B5	230	LG	1	180	1	0	0	0
B4	B5	230	LL	0	0	0.5859	-0.7042	0.4143	0.9959
B4	B5	230	LLG	0.5	0	0.5859	-0.7042	0.4143	0.9959
B4	B6	230	LLL	0	0	0.1081	-5.0443	0	0
B4	B6	230	LG	1	180	1	0	0	0
B4	B6	230	LL	0	0	0.5538	-0.4966	0.4462	0.6164
B4	B6	230	LLG	0.5	0	0.5538	-0.4966	0.4462	0.6164
B4	B7	230	LLL	0	0	0.3114	-4.993	0	0
B4	B7	230	LG	1	180	1	0	0	0
B4	B7	230	LL	0	0	0.6552	-1.1806	0.3452	2.2415
B4	B7	230	LLG	0.5	0	0.6552	-1.1806	0.3452	2.2415
B4	B8	230	LLL	0	0	0.2504	-4.9955	0	0
B4	B8	230	LG	1	180	1	0	0	0
B4	B8	230	LL	0	0	0.6248	-0.9996	0.3755	1.6636
B4	B8	230	LLG	0.5	0	0.6248	-0.9996	0.3755	1.6636
B4	B9	230	LLL	0	0	0.2245	-5.213	0	0
B4	B9	230	LG	1	180	1	0	0	0

B4	B9	230	LL	0	0	0.6119	-0.9552	0.3883	1.5051
B4	B9	230	LLG	0.5	0	0.6119	-0.9552	0.3883	1.5051
B5	B1	16.5	LLL	0	0	0.5954	-1.0874	0	0
B5	B1	16.5	LG	0	0	1	0	0	0
B5	B1	16.5	LL	0	0	0.7977	-0.4022	0.2024	1.5853
B5	B1	16.5	LLG	0	0	0.7977	-0.4022	0.2024	1.5853
B5	B2	18	LLL	0	0	0.6321	-0.6798	0	0
B5	B2	18	LG	0	0	1	0	0	0
B5	B2	18	LL	0	0	0.8161	-0.2668	0.184	1.1834
B5	B2	18	LLG	0	0	0.8161	-0.2668	0.184	1.1834
B5	B3	13.8	LLL	0	0	0.5822	-1.7224	0	0
B5	B3	13.8	LG	0	0	1	0	0	0
B5	B3	13.8	LL	0	0	0.791	-0.6302	0.2092	2.3834
B5	B3	13.8	LLG	0	0	0.791	-0.6302	0.2092	2.3834
B5	B4	230	LLL	0	0	0.2893	-5.0172	0	0
B5	B4	230	LG	1	180	1	0	0	0
B5	B4	230	LL	0	0	0.6442	-1.1207	0.3561	2.0276
B5	B4	230	LLG	0.5	0	0.6442	-1.1207	0.3561	2.0276
B5	B5	230	LLL	0	0	0	0	0	0
B5	B5	230	LG	1	180	1	0	0	0
B5	B5	230	LL	0	0	0.5	0	0.5	0
B5	B5	230	LLG	0.5	0	0.5	0	0.5	0
B5	B6	230	LLL	0	0	0.3221	-4.915	0	0
B5	B6	230	LG	1	180	1	0	0	0
B5	B6	230	LL	0	0	0.6606	-1.197	0.3398	2.3275
B5	B6	230	LLG	0.5	0	0.6606	-1.197	0.3398	2.3275
B5	B7	230	LLL	0	0	0.2425	-5.3249	0	0
B5	B7	230	LG	1	180	1	0	0	0
B5	B7	230	LL	0	0	0.6208	-1.043	0.3795	1.7064
B5	B7	230	LLG	0.5	0	0.6208	-1.043	0.3795	1.7064
B5	B8	230	LLL	0	0	0.3032	-4.9575	0	0
B5	B8	230	LG	1	180	1	0	0	0
B5	B8	230	LL	0	0	0.6512	-1.1528	0.3492	2.1498
B5	B8	230	LLG	0.5	0	0.6512	-1.1528	0.3492	2.1498
B5	B9	230	LLL	0	0	0.3614	-4.8729	0	0
B5	B9	230	LG	1	180	1	0	0	0
B5	B9	230	LL	0	0	0.6802	-1.2973	0.3203	2.7557
B5	B9	230	LLG	0.5	0	0.6802	-1.2973	0.3203	2.7557

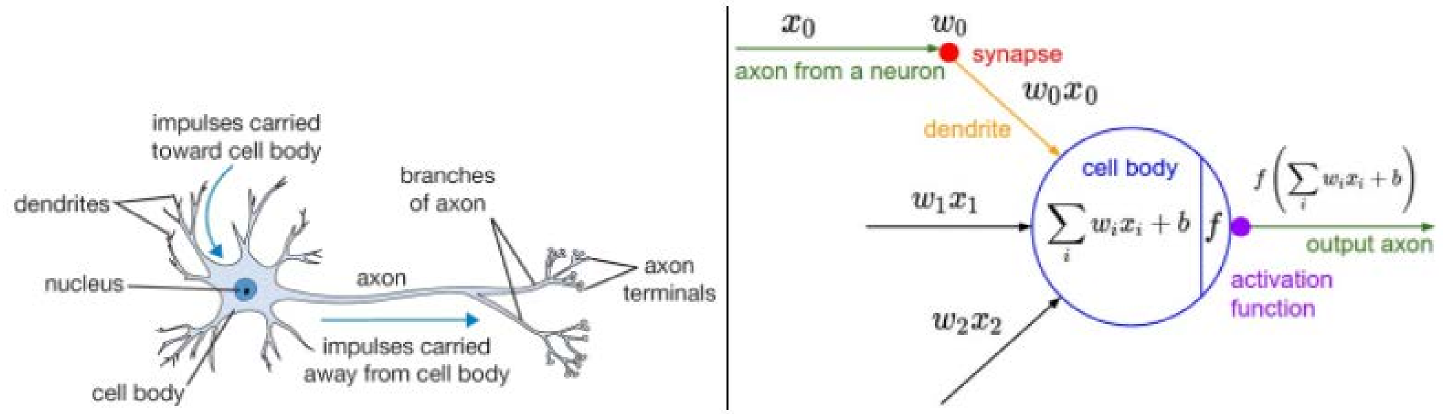
10. Discussion:

- A Multilayered feed forward system is used to identify faults in the IEEE-9 Bus system. The number of hidden layer is kept 100 which gives higher accuracy for this case, As the more number of hidden layers take more time to train.
- The more data to train, the more accuracy. We have done 216 times fault analysis to generate training data using CYME PSAF software as we did not have any existing real data to use as training data. We produced data of different sequence voltages for Line-Ground Fault (LG), Line-Line Fault (LL), Double Line-Ground Fault (LLG), and Triple Line-Ground Fault (LLLG).
- The data had to be rearranged in a 7776 x 54 matrix. Rows representing 216 times analysis for 36 classes and columns for 6 data (voltage magnitude and phase angle) for each bus.
- As changes in values of sequence currents were not suitable enough to use as training data and the percentage of accuracy was poor we did not use the sequence current values.
- At the time of training, it is not desirable to over train the system. When the overtraining starts to occur, the validation error tends to increase. So iteration is stopped at that point.
- A gradient descent algorithm, known as the back propagation algorithm is used as training algorithm. Backpropagation is a method to calculate the gradient of the loss function with respect to the weights in an artificial neural network. Backpropagation uses the error values to calculate the gradient of the loss function. In the second phase, this gradient is fed to the optimization method, which in turn uses it to update the weights, in an attempt to minimize the loss function.
- Conjugate gradient algorithms, Quasi-Newton algorithms, Levenberg-Marquardt (LM) algorithm could converge data faster than BP method, but BP method was sufficient for our purpose.
- Percentage of accuracy is near 83%. Though it changes with each time of training, the values are close.
- This Programme can be used for any bus system given the voltage magnitude and angles, which will be helpful for the providing companies get notified and solve the problem as soon as possible.
- If we had enough data for current and voltage values for faults due to various reasons, Such as lighting, tree falling on line etc. we would have been able to tell not only type of fault but also reason of fault.

11. Appendices

11.1. Appendix A

The basic computational unit of the brain is a neuron. Approximately 86 billion neurons can be found in the human nervous system and they are connected with approximately 10^{14} to 10^{15} synapses. The diagram below shows a cartoon drawing of a biological neuron (left) and a common mathematical model (right). Each neuron receives input signals from its dendrites and produces output signals along its (single) axon. The axon eventually branches out and connects via synapses to dendrites of other neurons. In the computational model of a neuron, the signals that travel along the axons interact multiplicatively with the dendrites of the other neuron based on the synaptic strength at that synapse. The idea is that the synaptic strengths are learnable and control the strength of influence (and its direction: excitatory (positive weight) or inhibitory (negative weight)) of one neuron on another. In the basic model, the dendrites carry the signal to the cell body where they all get summed. If the final sum is above a certain threshold, the neuron can fire, sending a spike along its axon. In the computational model, we assume that the



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

precise timings of the spikes do not matter, and that only the frequency of the firing communicates information. Based on this rate code interpretation, we model the firing rate of the neuron with an activation function which represents the frequency of the spikes along the axon. Historically, a common choice of activation function is the sigmoid function, since it takes a real-valued input (the signal strength after the sum) and squashes it to range between 0 and 1.

11.2. Appendix B

Every activation function (or non-linearity) takes a single number and performs a certain fixed mathematical operation on it. “tanh” and “sigmoid” are discussed below:

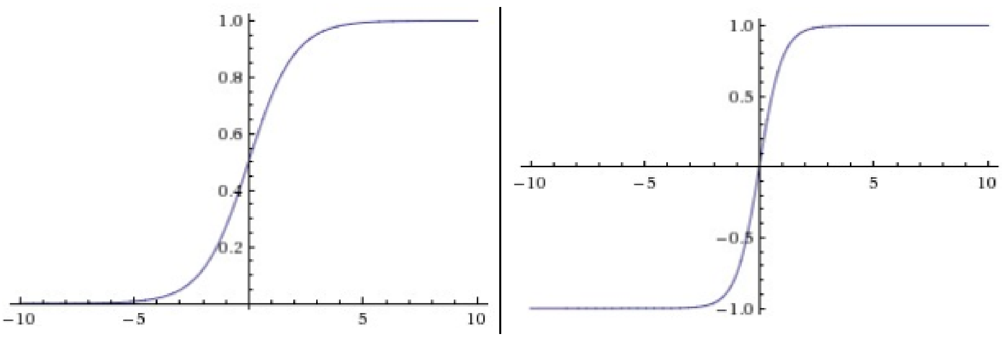


Figure 9:
Left: Sigmoid non-linearity squashes real numbers to range between [0, 1]
Right: tanh non-linearity squashes real numbers to range between [-1, 1].

- **Sigmoid**: The sigmoid non-linearity has is shown in the image above on the left. As alluded to in the previous section, it takes a real-valued number and “squashes” it into range between 0 and 1. In particular, large negative numbers become 0 and large positive numbers become 1. The sigmoid function has seen frequent use historically since it has a nice interpretation as the firing rate of a neuron: from not firing at all (0) to fully-saturated firing at an assumed maximum frequency. In practice, the sigmoid non-linearity has recently fallen out of favor and it is rarely ever used. It has two major drawbacks:
- **tanh:** The “tanh” non-linearity is shown on the image above on the right. It squashes a real-valued number to the range [-1, 1]. Like the sigmoid neuron, its activations saturate, but unlike the sigmoid neuron its output is zero-centered.

11.3. Appendix C

Cross Entropy Loss and Gradient Descent Algorithm: Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

Gradient descent is also known as steepest descent. Gradient descent is a popular method in the field of machine learning because part of the process of machine learning is to find the highest accuracy, or to minimize the error rate, given a set of training data. Gradient descent is used to find the minimum error by minimizing a "cost" function.

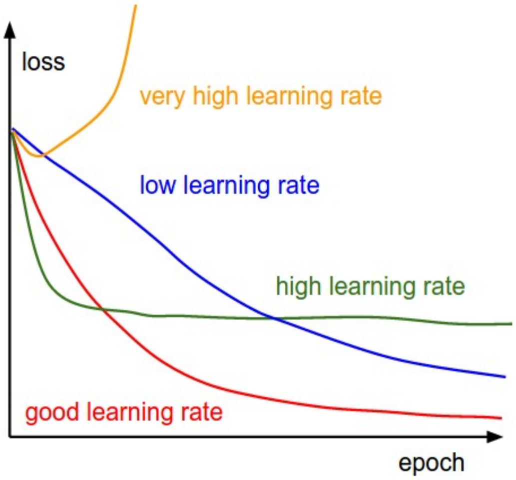


Figure 10: Loss vs Epoch count graph

11.4. Appendix D

Over Fitting: Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A model that has been over fit has poor predictive performance, as it overreacts to minor fluctuations in the training data.

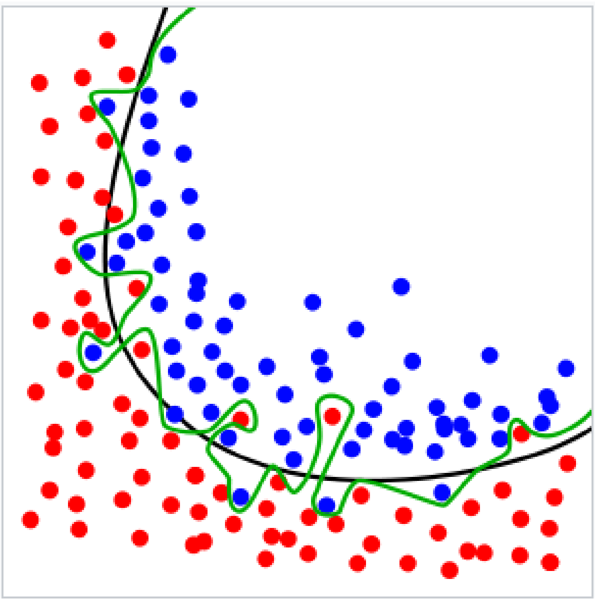


Figure 11: The green line represents an over fitted model and the black line represents a regularized model.

11.5. Appendix E

Acronym	Meaning
ANN	Artificial Neural Network
ANSI	American National Standards Institute
BP	Back Propagation
CA	Classification Accuracy
IEEE	Institute of Electrical and Electronics Engineers
LG	Line to Ground
LL	Line to Line
LLG	Double Line to Ground
LLL	Triple Line
LLLG	Triple Line to Ground
LM	Levenberg-Marquardt
PSAF	Power System Analysis Framework
PSCAD	Power System Computer Aided Design
PU	Per Unit