Assignment 3: Designing a Neural Network for classification

Introduction

This Report summarizes the Neural Network Architecture designed by our team and highlight the methods and functions employed in the network. It also discusses the results and efficiency of the network. The report also gives insight into the dataset and describe its various statistical properties.

About Dataset

The Dataset is fetched from **“UCI Machine Learning Repository”,** and is affiliated to the healthcare domain. The Dataset label is **“Heart Disease Data Set”.** This Dataset is published by **“Cleveland Clinic Foundation”**

Source

Creators:

- Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
- University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
- University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
- V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

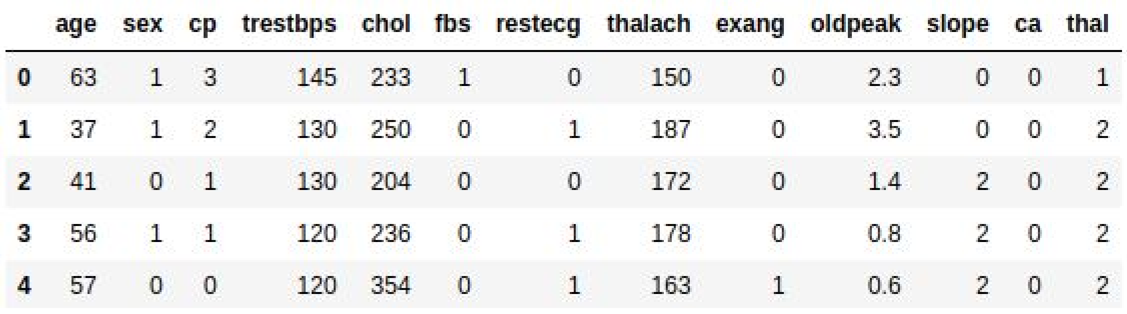
Donor: David W. Aha (aha '@' ics.uci.edu)

Description

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. The database is further modified to fit into standard binary classification by concentrating on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0). The Database comprises of 303 unique samples.

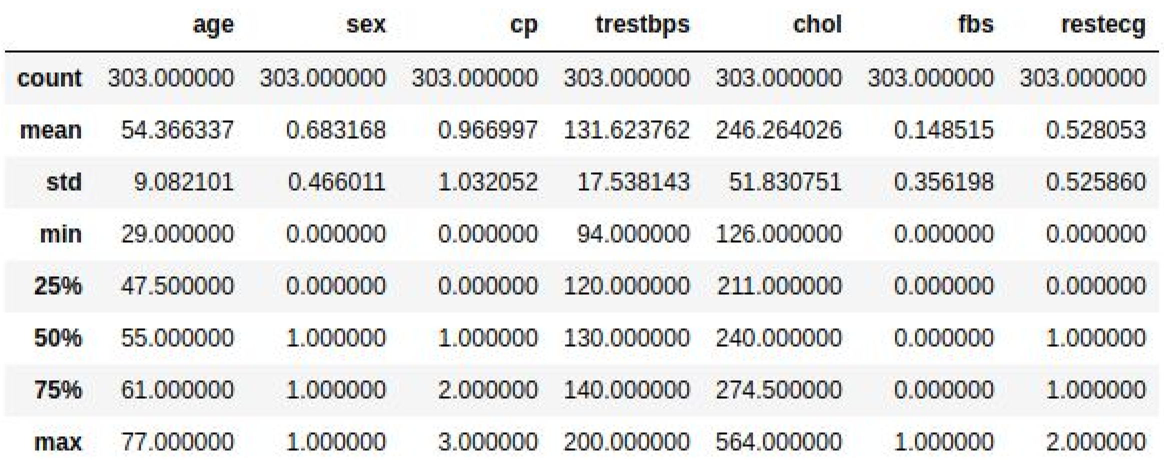
Attribute Information

1) Age
2) Sex
3) Chest pain type (4 values)
4) Resting blood pressure
5) Serum cholesterol in mg/dl
6) Fasting blood sugar > 120 mg/dl
7) Resting electrocardiographic results (values 0,1,2)
8) Maximum heart rate achieved
9) Exercise induced angina
10) Oldpeak = ST depression induced by exercise relative to rest
11) The slope of the peak exercise ST segment
12) Number of major vessels (0-3) colored by fluoroscopy
13) thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
14) Target Value (values 0,1)

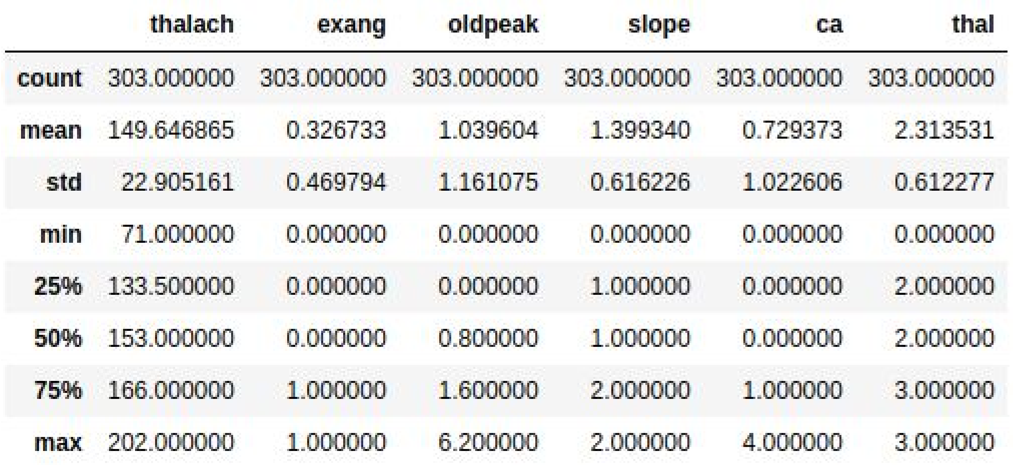


	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2

Statistical Properties

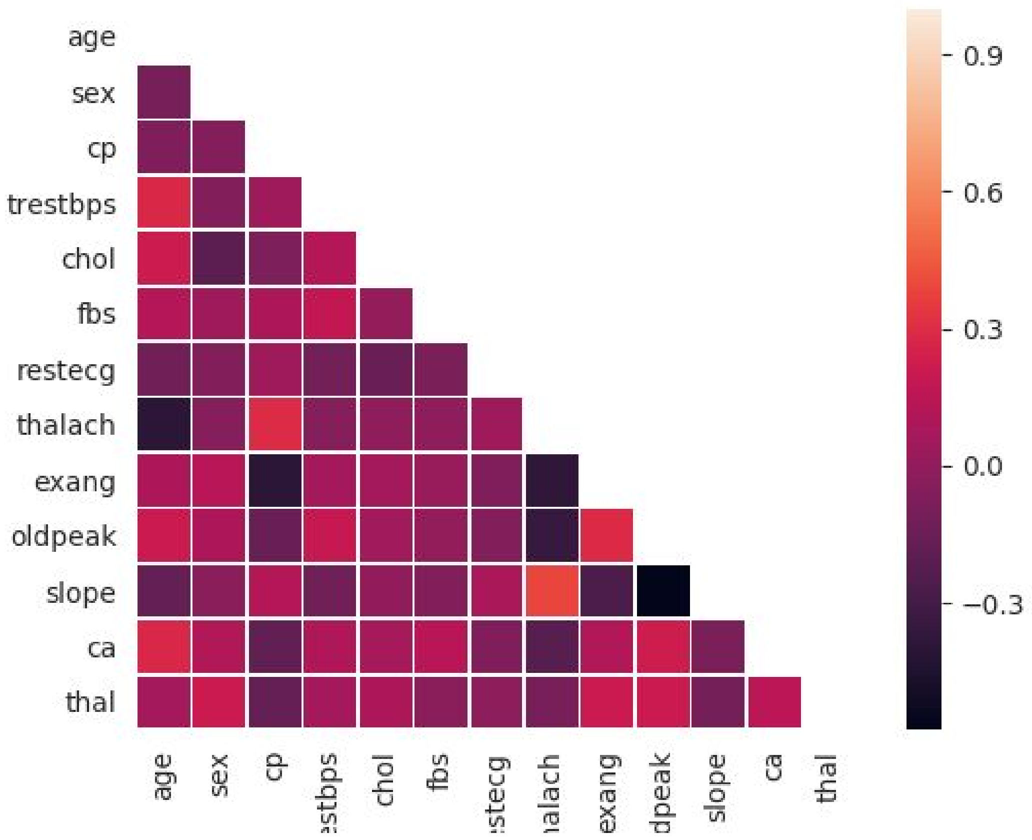


	age	sex	cp	trestbps	chol	fbs	restecg
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000



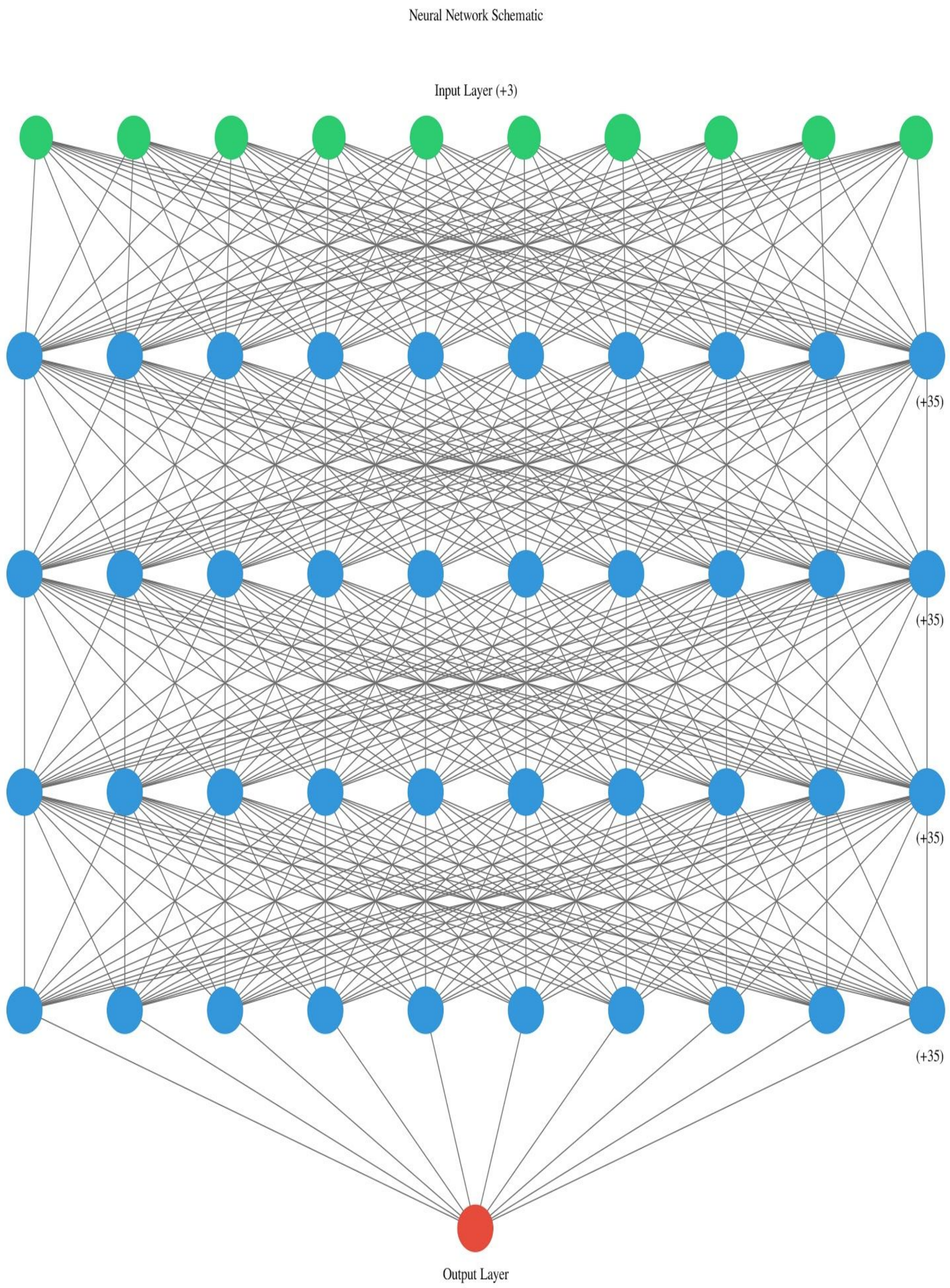
	thalach	exang	oldpeak	slope	ca	thal
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531
std	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277
min	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000
50%	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000
75%	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000
max	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000

Correlation Heatmap



Architecture

The Neural Network consists of **5 layers** with **4 hidden layers** and **1 output layer**. The hidden layer each have **45 neurons** and the output layer has single neuron. The connection between different layers of network is of standard Deep Neural Network i.e each neuron in the preceding layer is connected to every other neuron in next successive layer.



Neural Network Schematic
Input Layer (+3)
(+35)
(+35)
(+35)
(+35)
Output Layer

Methods and Functions

The Neural Network designed for binary classification so to attain maximum accuracy appropriate **activation functions** and **loss function** are used and tested to squeeze as much performance from the network.

Loss Function : Cross Binary Entropy

$$-(y\log(p) + (1-y)\log(1-p))$$

Cross Binary Entropy loss is ideal and best suited for classification problems. Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0. As the predicted probability approaches 1, log loss slowly decreases. As the predicted probability decreases, however, the log loss increases rapidly. Log loss penalizes both types of errors, but especially those predictions that are confident and wrong.

Hidden Layer Activation Function : ReLU

$$RELU(x) = \begin{cases} 0 \ if \ x < 0 \\ x \ if \ x >= 0 \end{cases}$$

A rectified linear unit has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. ReLUs' machinery is more like a real neuron in your body. ReLU activations are the simplest non-linear activation function you can use, obviously. When you get the input is positive, the derivative is just 1, so there isn't the squeezing effect you meet on backpropagated errors from the sigmoid function. Research has shown that ReLUs result in much faster training for large networks.

Output Layer Activation Function : Sigmoid

$$f(x) = sigmoid(x) = \frac{1}{1 + e^{-x}}$$

The sigmoid function has been widely used in machine learning intro materials, especially for the logistic regression and some basic neural network implementations. The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output.Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

The function is differentiable.That means, we can find the slope of the sigmoid curve at any two points. The function is monotonic but function’s derivative is not. The logistic sigmoid function can cause a neural network to get stuck at the training time.

Optimizer Function : Simple Gradient Descent

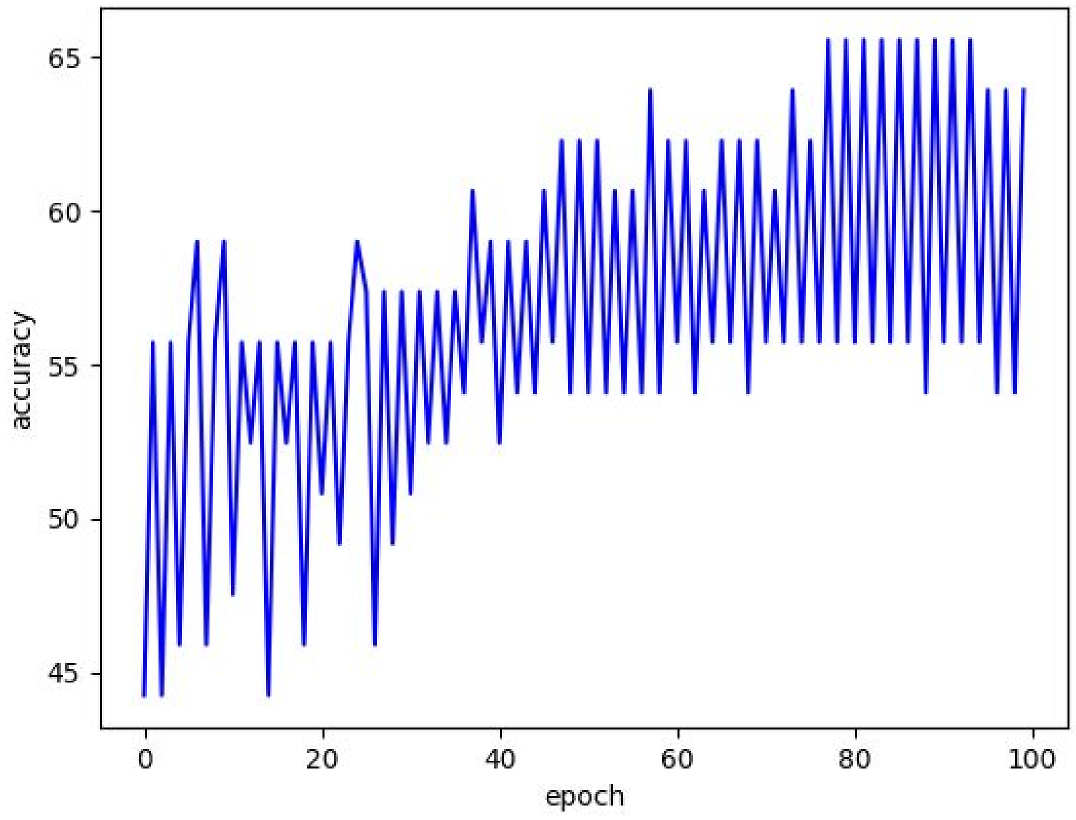
$$\text{Repeat until convergence } \{$$

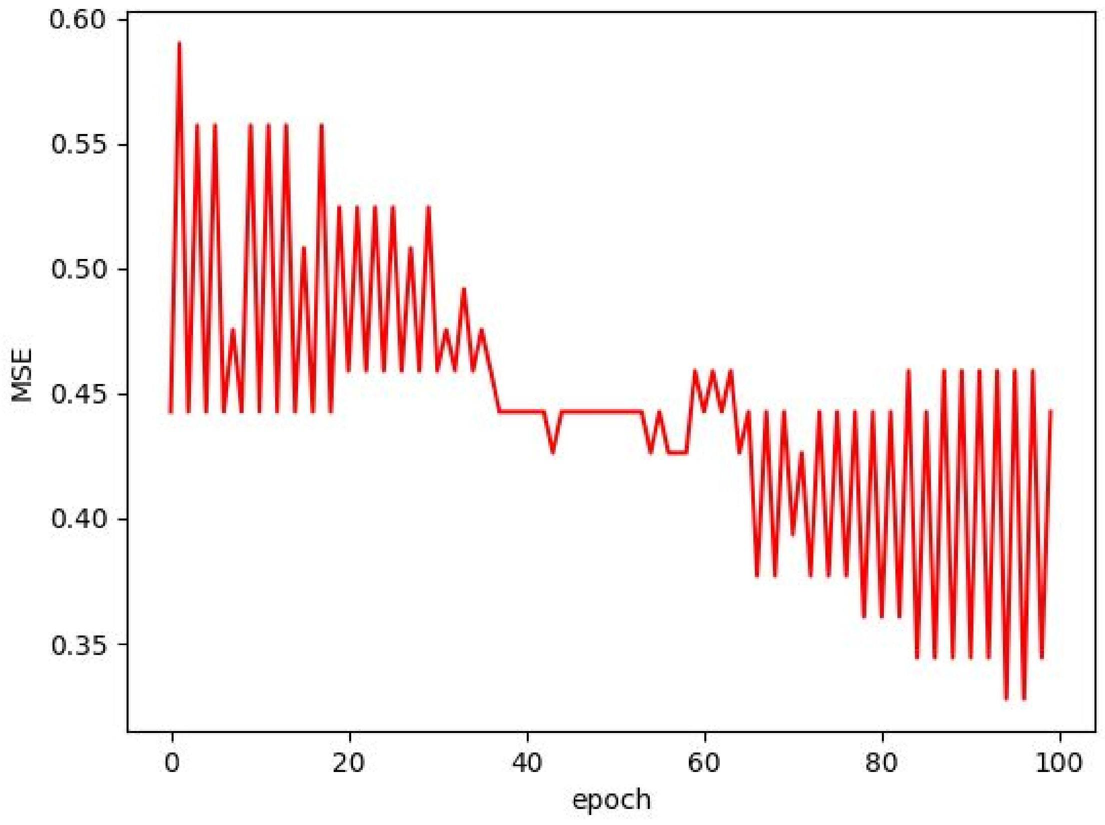
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

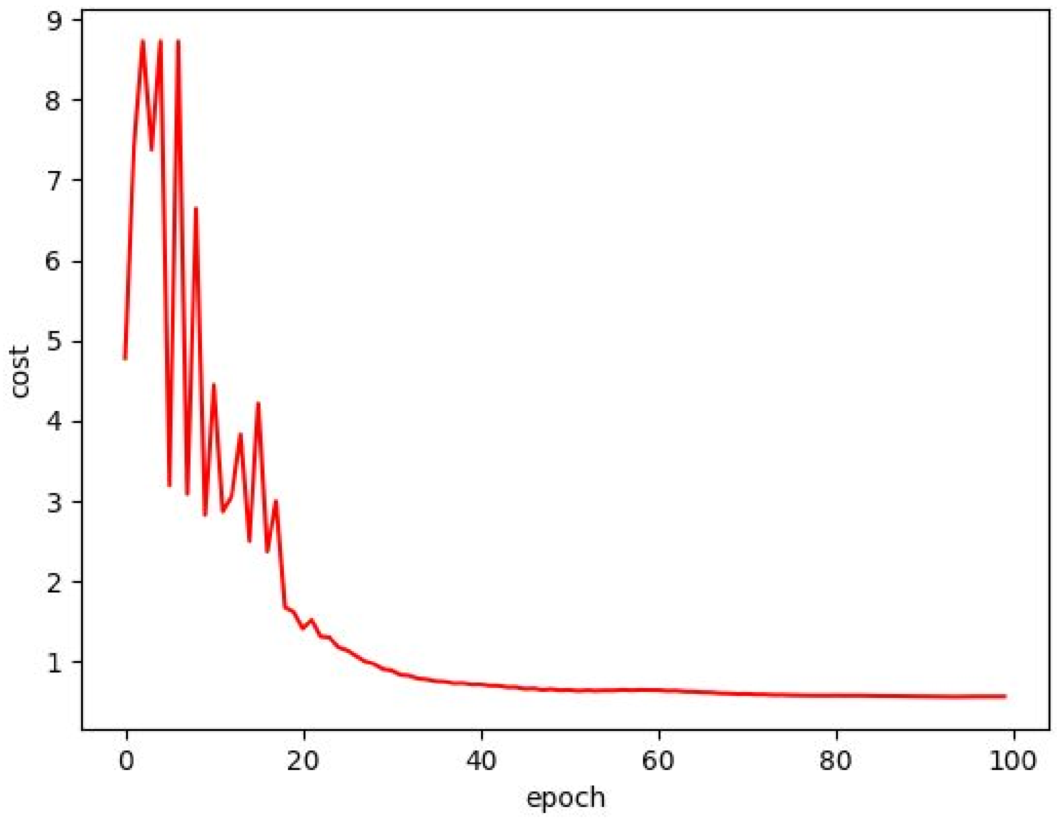
$$\}$$

Result and Conclusion

We ran the network for **100 epochs** with **learning rate 0.001** and dataset has been divided into train and test with **split 0.8**. The accuracy , mean square error and cost function are as follows :-




0.60
0.55
0.50
0.45
0.40
0.35
MSE
0
20
40
60
80
100
epoch



Final Accuracy : 81.96721

Final MSE : 0.36065573770491804