Multilayer Perceptron for letter image recognition

Objective:

Train a multi layer perceptron on letter image recognition for given train and test data sets and capture the performance results by varying the number of neurons in hidden layers, loss function used and stopping criteria of algorithm.

Each stimulus was converted into 17 numerical features which include class lablel information as per details shown below.

Attribute Information:

1.	letter	capital letter	(26 values from A to Z)
2.	x-box	horizontal position of box	(integer)
3.	y-box	vertical position of box	(integer)
4.	width	width of box	(integer)
5.	high	height of box	(integer)
6.	onpix	total # on pixels	(integer)
7.	x-bar	mean x of on pixels in box	(integer)
8.	y-bar	mean y of on pixels in box	(integer)
9.	x2bar	mean x variance	(integer)
10.	y2bar	mean y variance	(integer)
11.	xybar	mean x y correlation	(integer)
12.	x2ybr	mean of x * x * y	(integer)
13.	xy2br	mean of x * y * y	(integer)
14.	x-ege	mean edge count left to right	(integer)
15.	xegvy	correlation of x-ege with y	(integer)
16.	y-ege	mean edge count bottom to top	(integer)
17.	yegvx	correlation of y-ege with x	(integer)

Multi layer percepron is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes as shown in Fig.a. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.

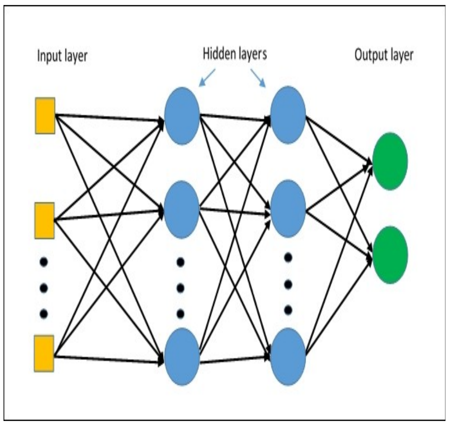


Fig.a

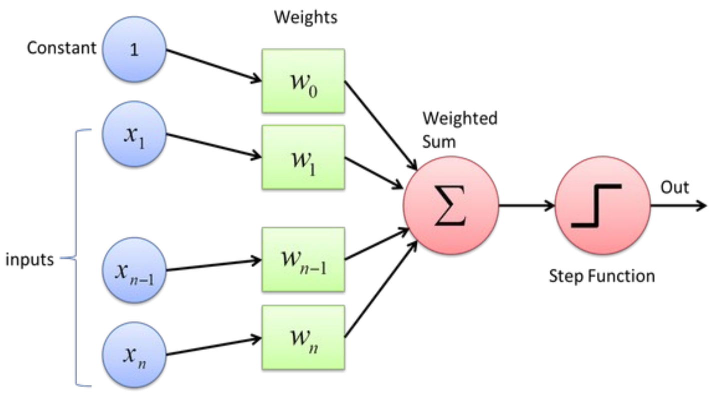


Fig.b

Input is multi-dimensional (i.e. input can be a vector) where input x = (x1, x2, .., xn)
Input nodes (or units) are connected (typically fully) to a node (or multiple nodes) in the next layer.

A neuron in the next layer takes a weighted sum of all its inputs as shown in Fig.b
Activation function used in assignment is Sigmoid function.
A sigmoid function is a mathematical having a characteristic "S"-shaped curve or sigmoid curve. Often, sigmoid function refers to the special case of the logistic function and defined by the formula

Backpropagation is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network. In the context of learning, backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss funtion. This technique is also sometimes called backward propagation of errors, because the error is calculated at the output and distributed back through the network layers.

Details of **loss functions** used in learning process:

1. **Cross Entropy Error -**
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.
In binary classification, where the number of classes M equals 2, cross-entropy can be calculated as: $-(y\log(p)+(1-y)\log(1-p))$
For M > 2, (i.e. multiclass classification), we calculate a separate loss for each class label per observation and sum the result. $-\sum y\log(p)$
Note: y and p are desired and actual outputs.
2. **Sum of squares deviation** - sum of squared errors of prediction (SSE), is the sum of the squares of errors (deviations predicted from actual values of data). It is a measure of the discrepancy between the data and an estimation model.
E is comupted as

$$E = \sum_p E_p = \sum_p \sum_i (t_{ip} - o_{ip})^2$$

where index p ranges over the set of input patterns, i ranges over the set of output units, and E_p represents the error on pattern p.

<u>**Stopping Criteria used**</u>: In implementation, 2 different stopping criteria are used to stop learning.
1. Stop after a certain number of runs through all the training data (each run through all the training data is called an epoch); Here we used 100 epochs as the limit.
2. Back-propagation attempts to reduce the errors between the output of the network and the desired result. During this process, weight adjustment is made and for the change to the weight wij from node i to node j:

$$\Delta w_{ij} = \epsilon \delta_{ip} i_{ip}$$

and W(new) = W(old) + Δwij
Training is stopped when ||ΔW|| < ε where ||ΔW|| is the Euclidean norm of ΔW and ε = 0.01
We also use learning rate = 0.001 in the test.

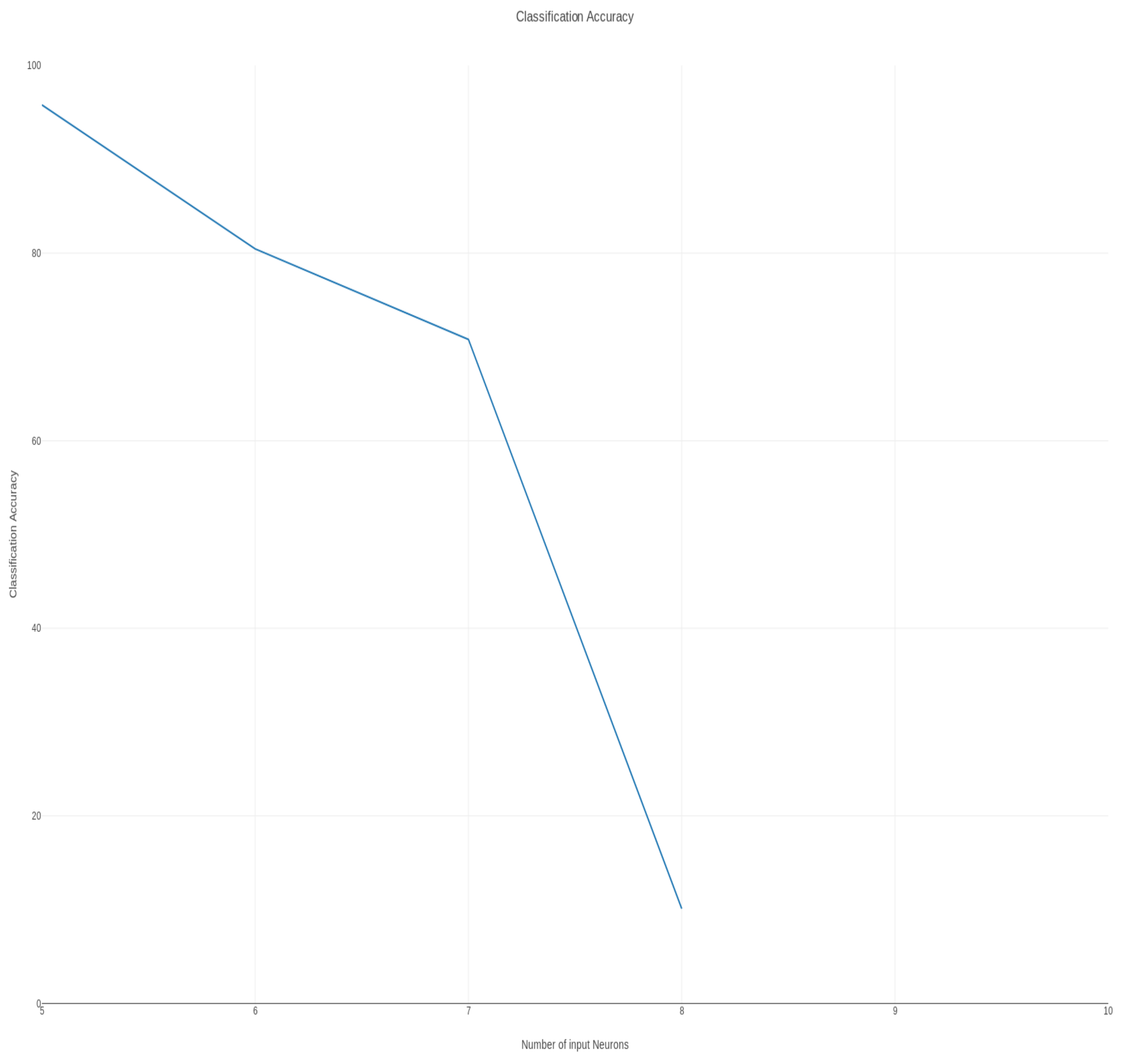
<u>**Observations:**</u>

MLP is trained by varying the number of neurons in hidden layer from 5 to 8 and train and test data given is used with different loss functions and stopping crieria is detailed below. A function to draw a graph of classification accuracy against number of input neurons is developed to demonstrate the results. Classfication accuracy is measured based on number of correct and incorrect results using train data and test data samples. For each of these samples

varying n ,i.e, 5 to 8 are used as per problem specification.
N Vs Classification accuracy is plotted for following inputs and details are provided below.
X-axis – n value and y-axis – Classification accuracy
1. Inputs used : Train data, sum of squares deviation loss function and stopping criteria |ΔW|| < ε

Results of given input neurons is summarized below and graph of the same is also shown below.

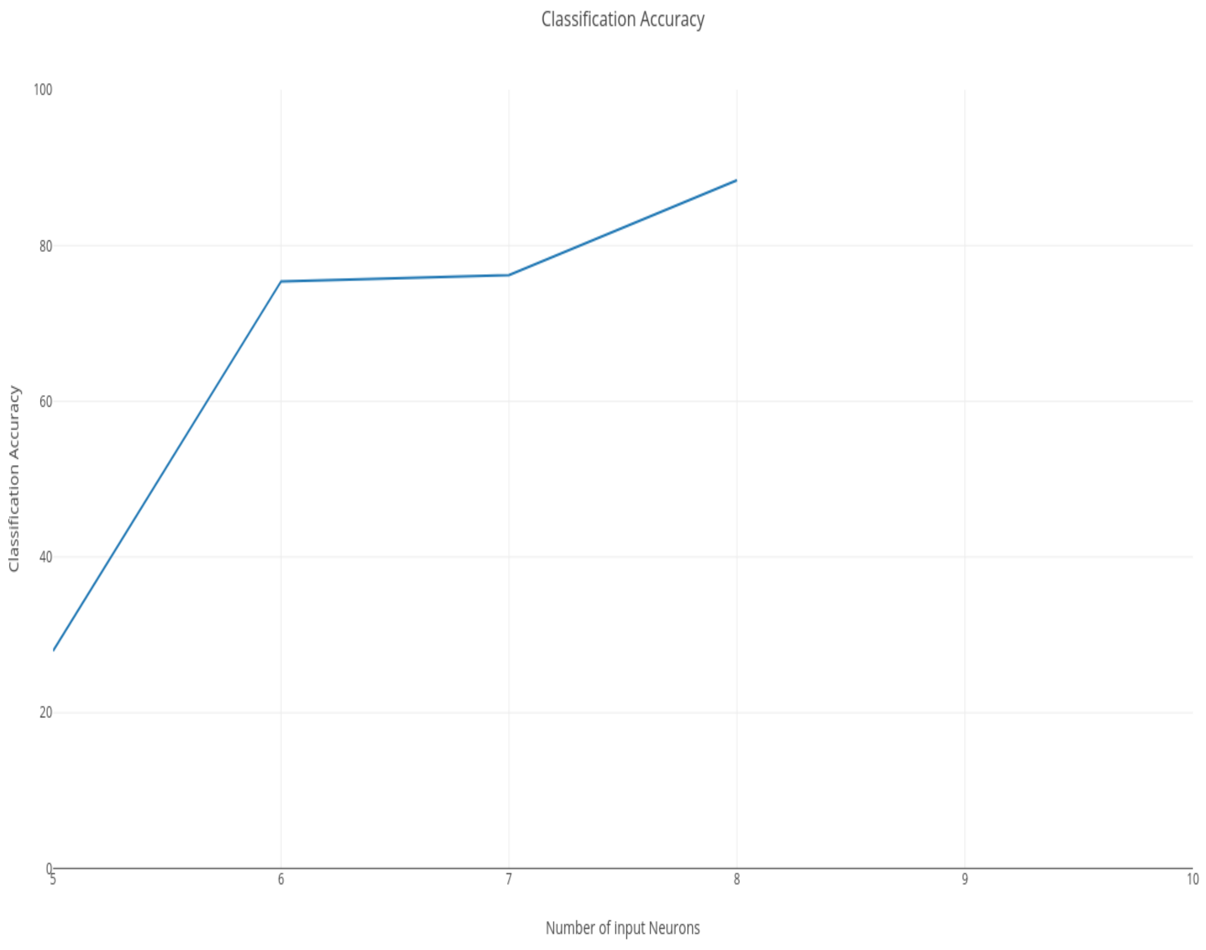
Number of input neurons	Classification Accuracy
5	95.80
6	80.46
7	70.80
8	10.10



2.Inputs used : Train data, cross entropy loss function and stopping criteria |ΔW|| < ε

Results of given input neurons is summarized below and graph of the same is also shown below.

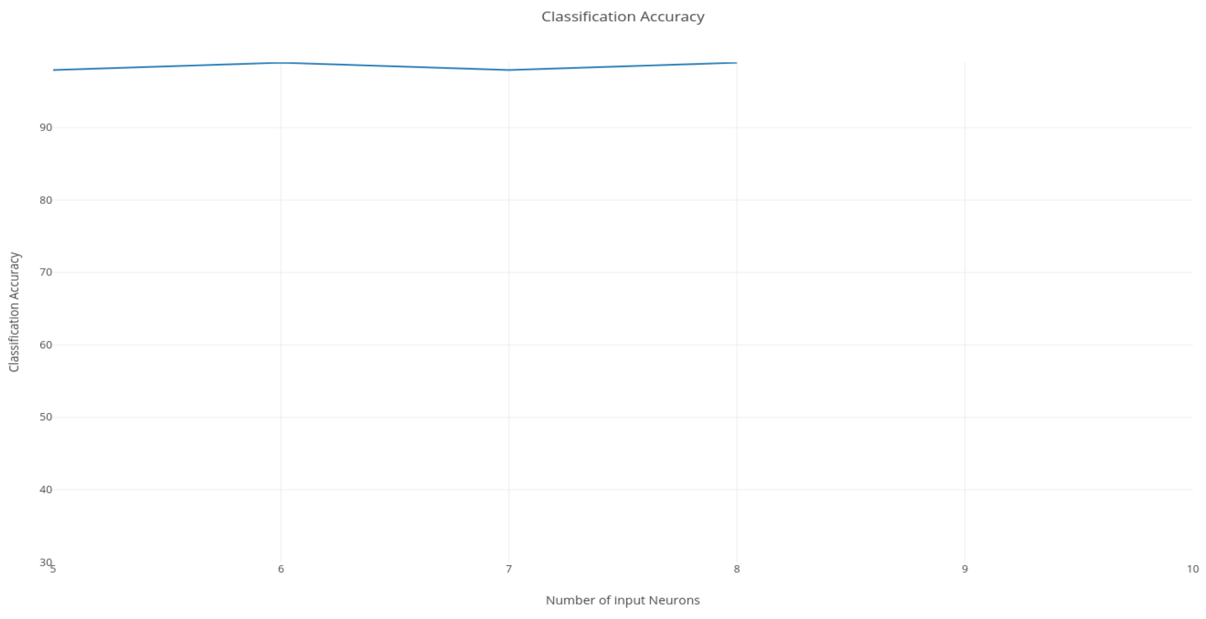
Number of input neurons	Classification Accuracy
5	27.93
6	75.41
7	76.22
8	88.40



3. Inputs used: Train data, cross entropy loss function and number of epochs = 100 as stopping criteria

Results of given input neurons is summarized below and graph of the same is also shown below.

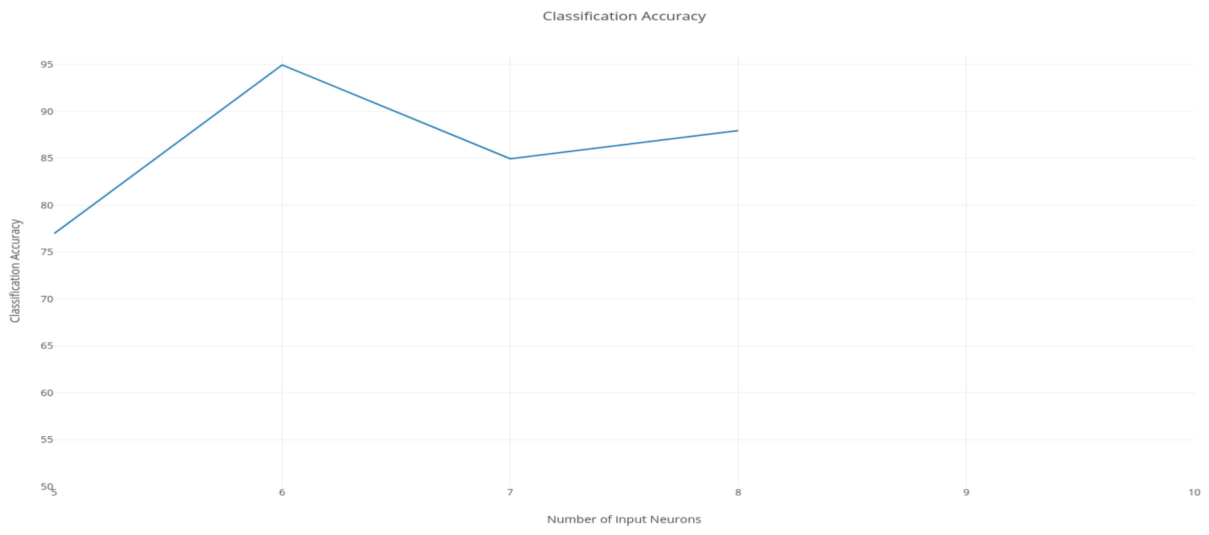
Number of input neurons	Classification Accuracy
5	97.96
6	98.96
7	97.96
8	98.96



4. Inputs used: Train data, sum of squares loss function and number of epochs = 100 as stopping criteria.

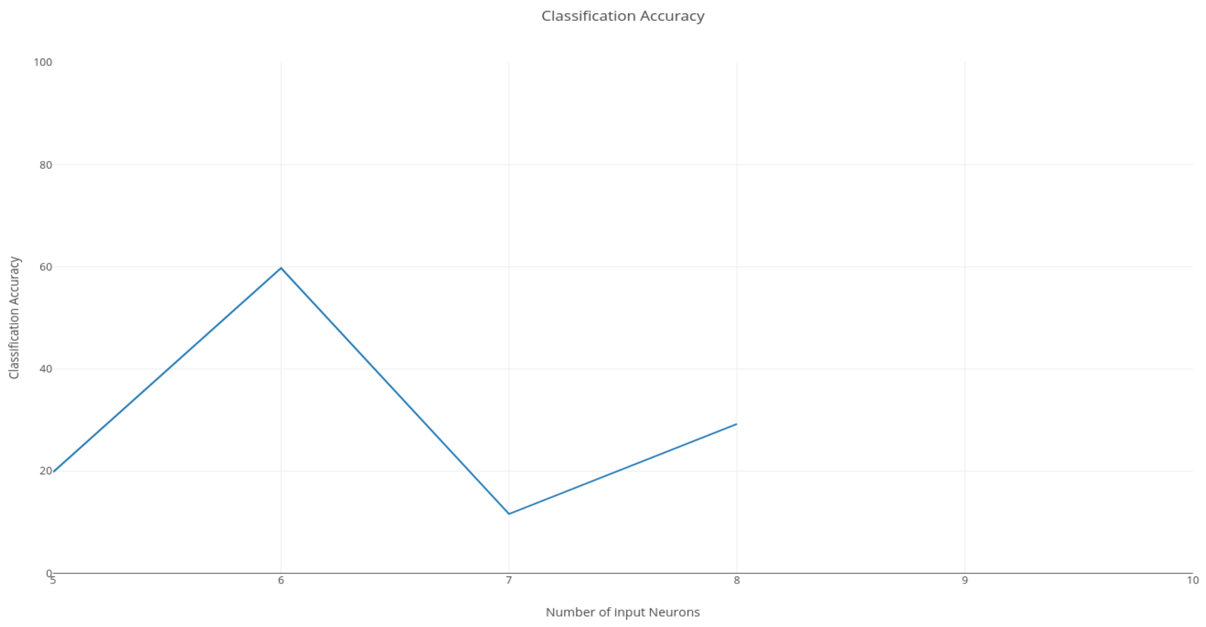
Results of given input neurons is summarized below and graph of the same is also shown below.

Number of input neurons	Classification Accuracy
5	76.97
6	94.96
7	84.96
8	89.96



5. Inputs used : test data, cross entropy loss function and stopping criteria |ΔW|| < ε
Results of given input neurons is summarized below and graph of the same is also shown below.

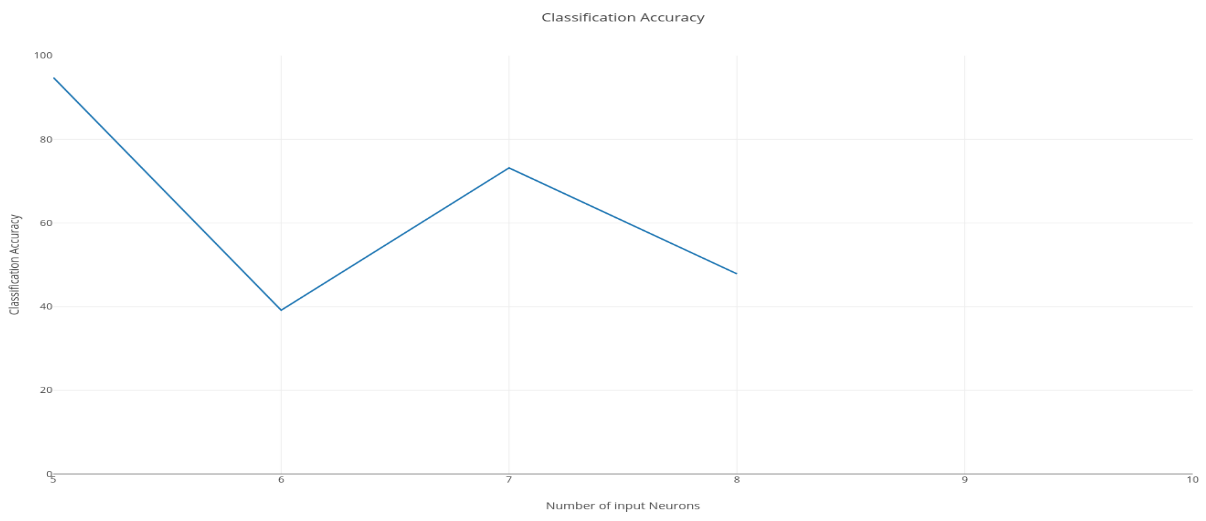
Number of input neurons	Classification Accuracy
5	19.82
6	59.76
7	11.61
8	29.23



6. Inputs used : test data, sum of squares loss function and stopping criteria |ΔW|| < ε.

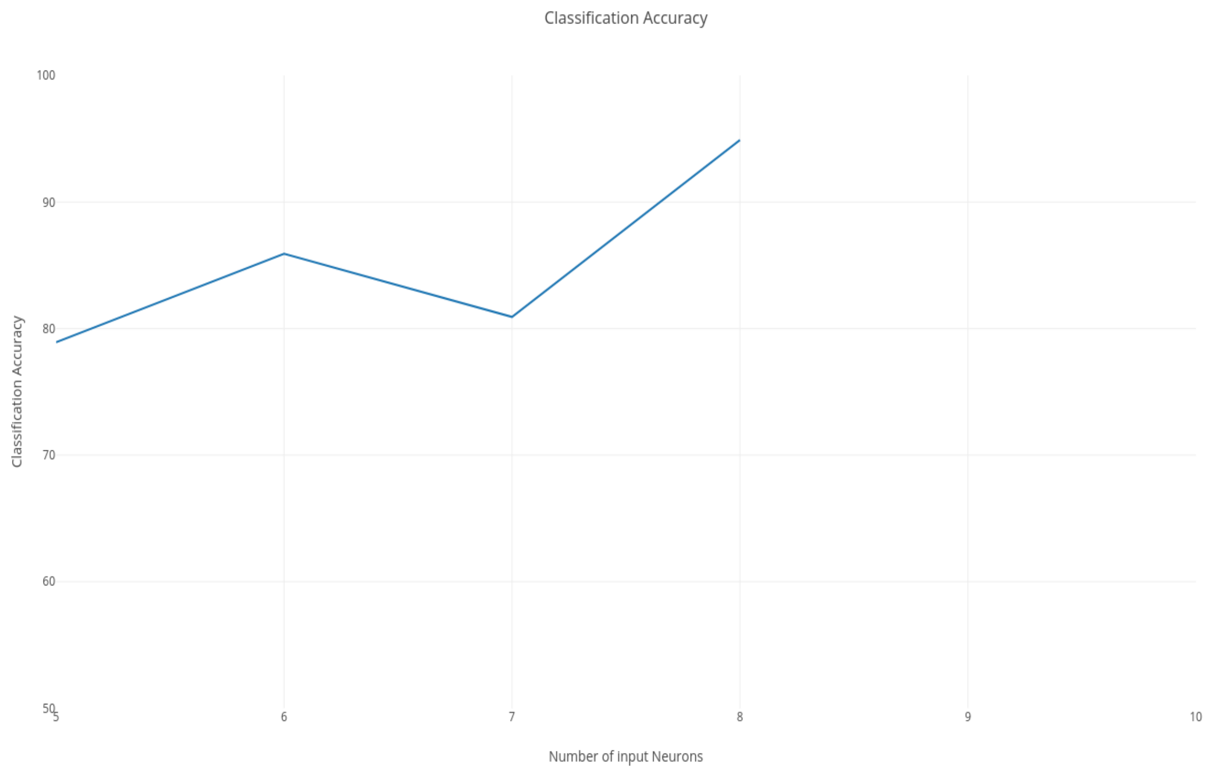
Results of given input neurons is summarized below and graph of the same is also shown below.

Number of input neurons	Classification Accuracy
5	94.79
6	39.14
7	73.17
8	47.85



7. Inputs used: test data, cross entropy loss function and number of epochs = 100 as stopping criteria
Results of given input neurons is summarized below and graph of the same is also shown below.

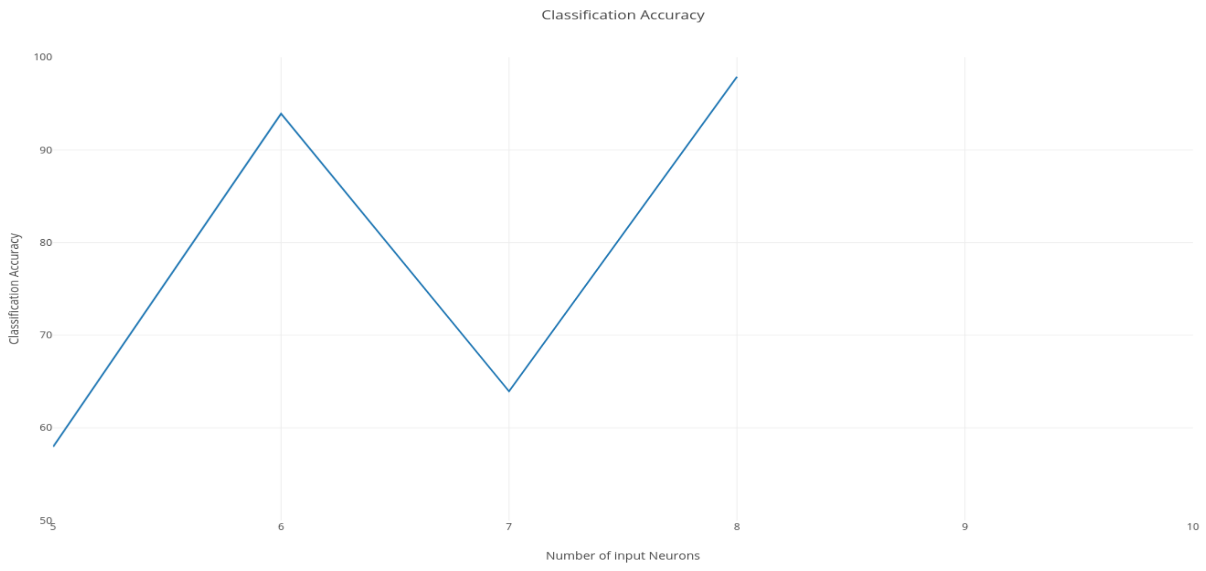
Number of input neurons	Classification Accuracy
5	78.92
6	85.91
7	80.92
8	94.90



8. Inputs used: test data, sum of squares loss function and number of epochs = 100 as stopping criteria

Results of given input neurons is summarized below and graph of the same is also shown below.

Number of input neurons	Classification Accuracy
5	57.94
6	93.91
7	63.94
8	97.90



Inferences:
In general, learning rate, number of inputs, weight adjustment and number of epochs are used to tune the performance of network.
1. Increase in number of input perceptrons improves classification to some extent. Beyond certain value, increase in input perceptrons doesn’t help to improve performance.
2.Accuracy improves with large data sets.
3. Cross entropy loss funcion is preferred to improve accuracy for multi-class problems.
4. Accuracy improves and becomes stable with slow learning rate.
5. Cross entropy combined with softmax output (preferred for multi class problems) improves prediction and this accuracy levels.
6. Random weight selection, class distribution of data in train/test data provided, loss functions and stop functions used influenced results.