

Neural Network Architecture:

No of Hidden Layers = 1
Nodes in Input layer = 784 (No of features)
Nodes in Output layer = 10 (No of labels)
Nodes in Hidden layer = 88 ($\sqrt{784 \times 10}$)

Activation Functions used:

At Hidden layer – sigmoid

At Output layer - softmax (for multiclass)

No of layers effect Performance:

0 layers – Only capable of representing linearly separable decisions

1 layer – Can approximate any function that contains a continuous mapping

2 layers – Any arbitrary decision boundary to arbitrary accuracy. Can approximate any smooth mapping to any accuracy

It is suggested for most practical problems, only 1 hidden layer is enough.
So I have chosen 1 hidden layer

No of neurons in hidden layer:

There is no specific formula to get optimized number of neurons.

But there are some thumb rules.

I have used the Masters Rule which says that For a three layer network with n input and m output neurons, the hidden layer would have $\sqrt{n \times m}$ neurons.