The dataset contains the attribute values of Iris flowers. We will classify Iris flowers based on the length and width measurements of their sepals and petals. So, there are four features,

* Sepal length
* Sepal width
* Petal length
* Petal width

The categorized output is the corresponding species. There are three types of possible species:

* Iris setosa
* Iris virginica
* Iris versicolor

In our dataset, there are 110 examples that we have used to train our model, and 30 examples for the validation of the model. So,

* Train batch size = 110
* Test batch size = 30

We build our model using one hidden layer with 10 hidden units. We are categorizing dataset into 3 classes based on 4 attributes. So,

* Input layer size = 4
* Output layer size = 3
* Hidden layer size = 10

We can set different parameters for learning rate, number of iterations, and hidden layer size. We set different values to find the best learning rate for this model.

* Learning rate = 0.01
* Number of epochs = 500

We used Sigmoid as our activation unit which is a nonlinear activation function.

σ (WX) = 1 / (1 + e­-WX)

We draw different types of plot to understand the changes of the model over time. We used a parameter to set the frequency of drawing those plots.

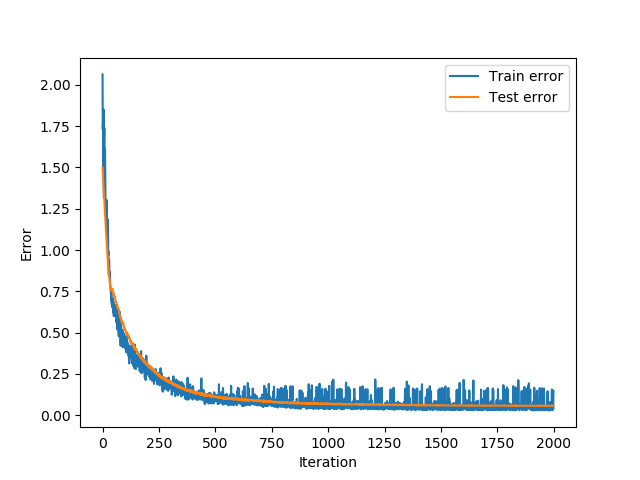
* Chart display frequency = 10

**Training and Test error changing rate along iterations:**

We had a plot to visualize the rate of changing errors along each iteration for the training and the test dataset.

***Configuration:***

* Learning rate = 0.001
* Number of epochs = 2000
* Activation unit = Relu
* Number of units in hidden layer = 10



From the figure, we can see that the training and testing errors have decreased over the number of iterations. From the graph we can conclude that the model did not overfit over the training dataset. We know there are two reason of overfitting,

* Random errors or noise
* Coincidental patterns

So, we can also conclude that the examples are also evenly distributed over the training and testing dataset, and there is no random noise or coincidental patterns.

**Histogram on activation values of hidden unit:**

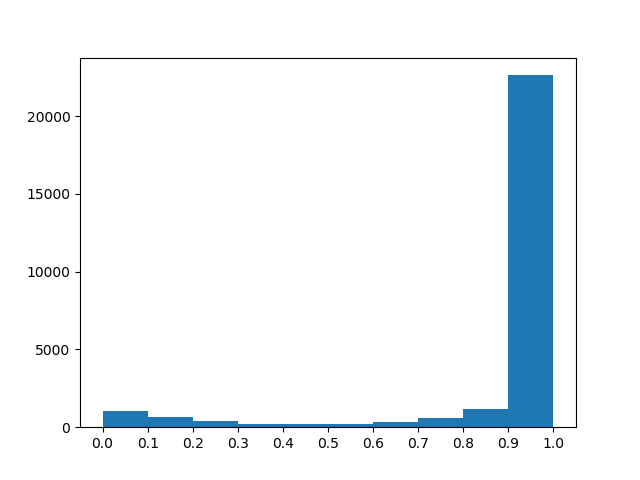
We had another plot to understand the distribution of activation values along with the iterations which are using as the inputs for the output layer.

***Configuration:***

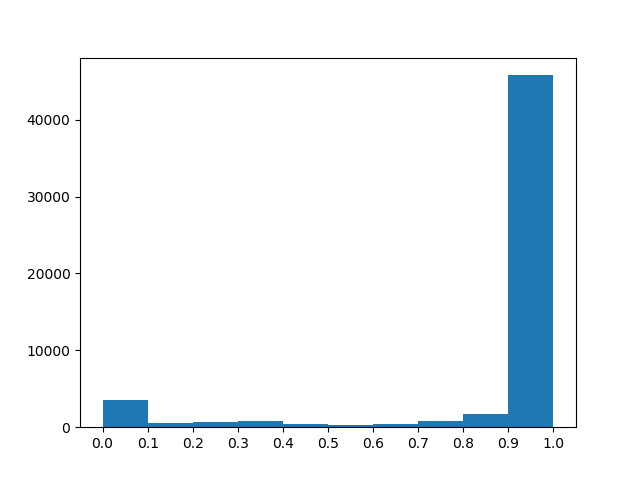
* Learning rate = 0.01
* Number of epochs = 2000
* Activation unit = Sigmoid
* Number of units in hidden layer = 10
* Chart display frequency = 25

We observed the distribution for the all activation values along iterations.

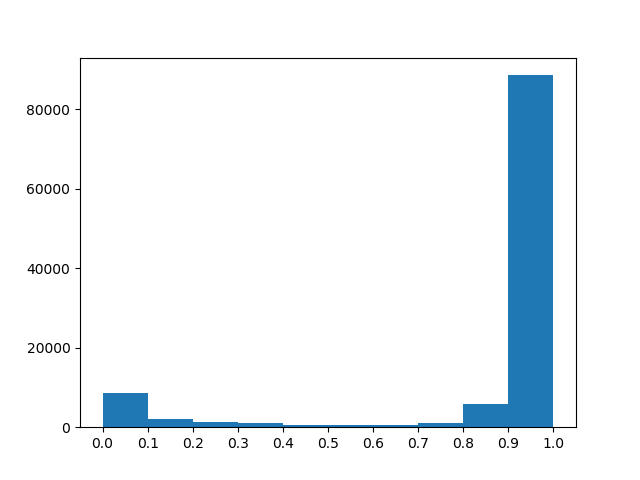
***After 25 iterations:***



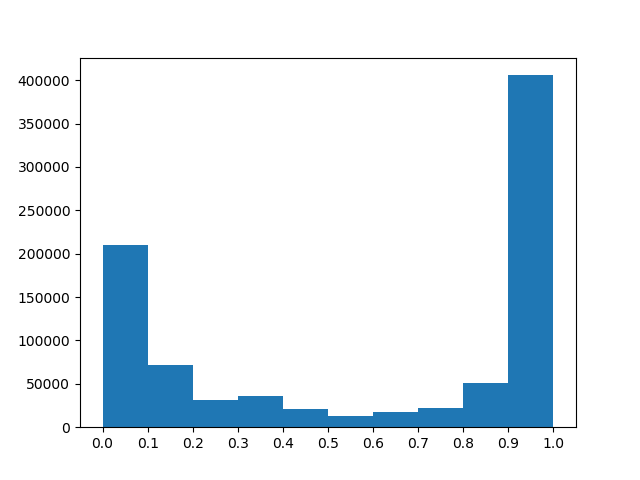
***After 50 iterations:***



***After 100 iterations:***



***Finally after 500 iterations:***



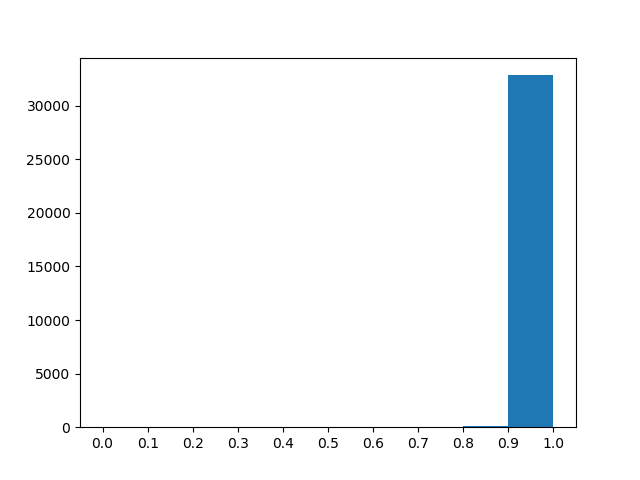
From the distribution of activation values along number of iterations, we can see that the number of lower activation values are increasing with the increase of number of iterations. We used Sigmoid as our activation unit.

Now, Lets check the distribution of activation for ***different neurons in the same layer***.

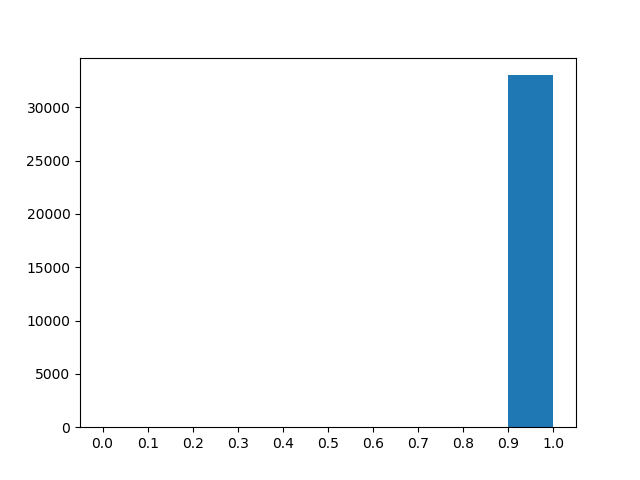
***Configuration:***

* Learning rate = 0.1
* Number of epochs = 300
* Activation unit = Sigmoid
* Number of units in hidden layer = 5

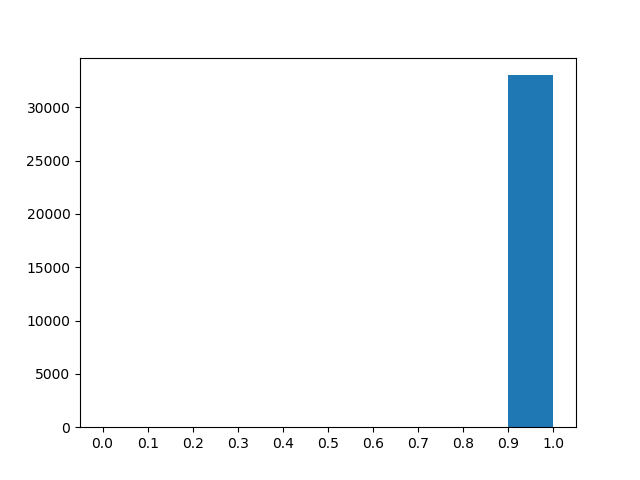
***Node 01:***



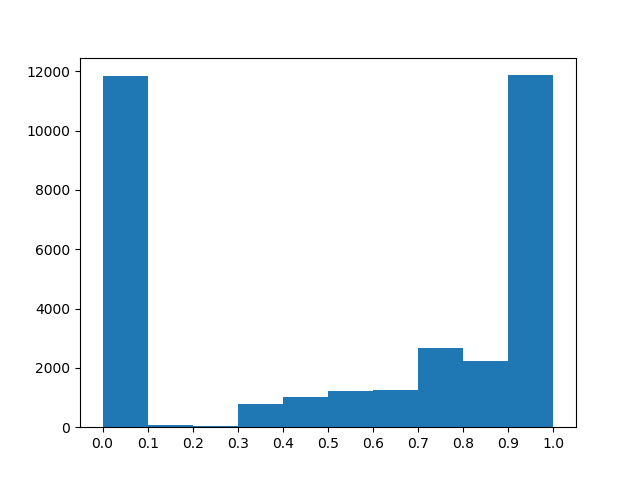
***Node 02:***



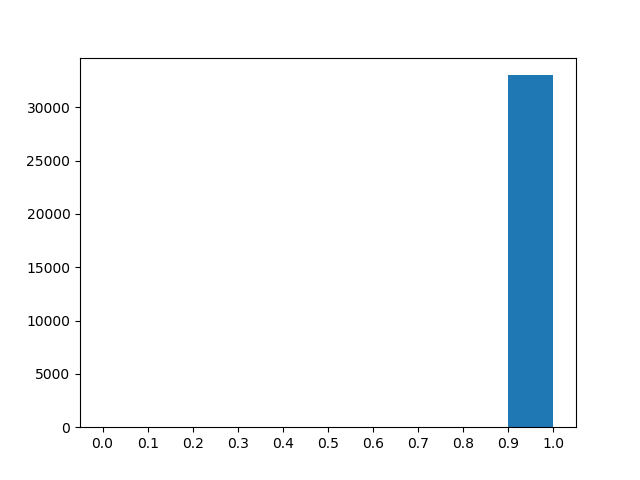
***Node 03:***



***Node 04:***



***Node 05:***



From the plots, we can see that Unit no 01, 02, 03, and 05 have almost similar distribution, but unit 04 have different distribution. So different neurons can have either similar, or different distributions.

**Weight change:**

