**a. Describe the structure of the network. How many layers does this network have? What is**

**the purpose of each layer?**

The neural network has a total of 9 layers.

1 Input Layer, 7 Hidden Layers and 1 Output Layer

The Activation function being used here is Rectified Linear Unit (Relu) – max(0,x)

**b. What does “Loss” mean here? What is the actual loss function? You may need to consult the**

**source code, which is available on Github.**

The Loss function is a Regression with L2 norm

**c. Plot the loss over time, after letting it run for 5,000 iterations. How good does the network**

**eventually get?**

The Loss function is a Regression with L2 norm.

After waiting for the first 5000 iterations, The plot of loss function for every 500 iterations interval looks like the following plot.

A screenshot of a cell phone

Description automatically generated

From the above graph we can infer that the loss function converges to a value of around 0.0031.

**d. Can you make the network converge to a lower loss function by lowering the learning rate**

**every 1,000 iterations? (Some learning rate schedules, for example, halve the learning rate**

**every n iterations. Does this technique let the network converge to a lower training loss?)**

From the above graph given in part c, we can infer that the loss function converges to a value of around 0.0031.

Now, we reduce the learning rate for every 10000 iterations and we can notice that the algorithm converges faster than above as it is picturised in the figure below:

A close up of a map

Description automatically generated

The value of loss function converges to 0.003 and it reaches its value at around 8000 iterations. This is much faster than the case when we didn’t decrease the learning rate.

*If we decrease the learning rate as the iterations proceed, the algorithm converges faster to its final value.*

**e. Lesion study. The text box contains a small snippet of Javascript code that initializes the network. You can change the network structure by clicking the “Reload network” button, which simply evaluates the code. Let’s perform some brain surgery: Try commenting out each layer, one by one. Report some results: How many layers can you drop before the accuracy drops below a useful value? How few hidden units can you get away with before quality drops noticeably?**

Effect of the number of hidden layers on the output: The below plot indicates the loss function with iterations for various hidden layers.

**f. Try adding a few layers by copy+pasting lines in the network definition. Can you noticeably**

**increase the accuracy of the network?**

The below plot compares the performance of two neural networks with layers 9 and 12 respectively. We can notice that the loss functions are almost similar indicating that the addition of more layers will not increase the accuracy of the network. Hence, we can conclude that adding new layers to the network has no additional effect on the accuracy of the model.

A screenshot of a cell phone

Description automatically generated