Assignment 1 – Classifier Neural Networks

1. Introduction

In this paper, I will discuss the implementation of a simple neural network that is used for classification. Specifically, I will not be focusing on the performance of the network but rather, I will perform multiple experiments to understand how the number of hidden nodes and a change in the learning rate can affect the activation of the hidden nodes.

My hypothesis is that an increased amount of hidden nodes will lead to a smaller ratio of node being activated compared to a smaller amount of hidden layers, as it has been argued that being a certain amount of hidden layers, it will have a diminishing effect of increasing performance. In addition, I hypothesize that a lower learning rate will help to achieve a better distinction between hidden node activations, clearly defining what nodes are being activated by what parts of the input data.

2. Model & Experiment Design

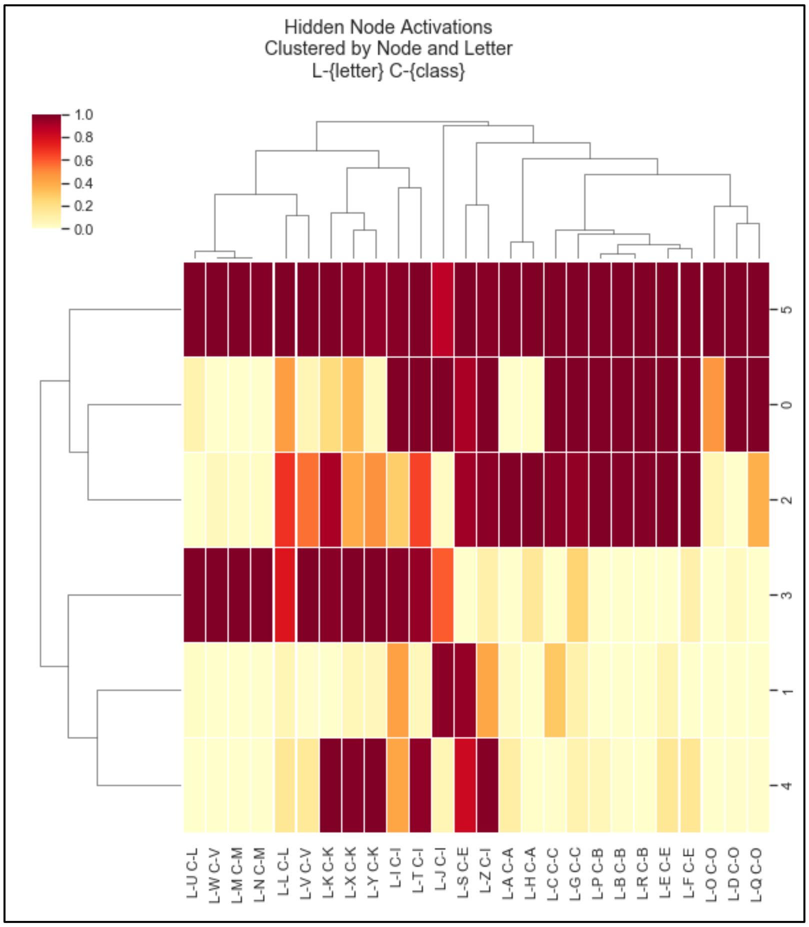
A simple neural network (input-hidden-output) architecture utilizing the backpropagation algorithm is implemented for this experiment. The neural network contains 81 input nodes, corresponding to having a 9x9 element input grid, and 9 output nodes. There are 26 potential classes, which corresponds to each letter in the alphabet, however we are only interested in 9 classes. These 9 classes are determined by how similar the letters are in terms of line/angle construction (i.e. class 4 containing "E" and "F"). The input data is structured to be a 9x9 grid, consisting of 0 (white space/pixel) and 1 (the angle of the letter in the pixel). In addition, a noise parameter = 0.10 is implemented into the model make the data appear "more realistic" as in the real world, we cannot expect to have perfect, well defined data to work with.

There will be a baseline model (81 input nodes, 9 output nodes, 6 hidden layers, learning rate α = 1.0), and experiments involving two factors – number of hidden nodes (2, 12), α (0.5, 1.5)) – will be done to see how they affect the hidden node activations, if at all.

For the backpropagation algorithm portion of the neural network, α=1.0, η=0.5, the maximum number of iterations was capped at 5000, and ε = 0.01.

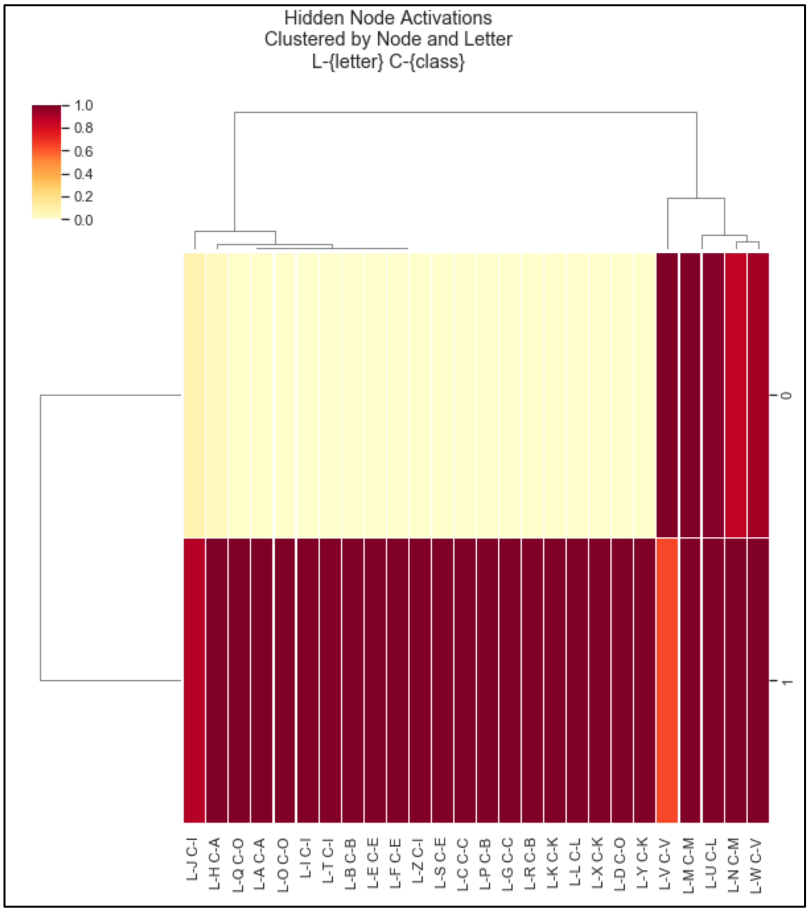
3. Hidden Node Activations – Number of Hidden Nodes

Like mentioned earlier, the baseline model will be used as a control to see how hidden node activations change when the specified factors are changed. The following figure shows the activation clustered by node and letter:

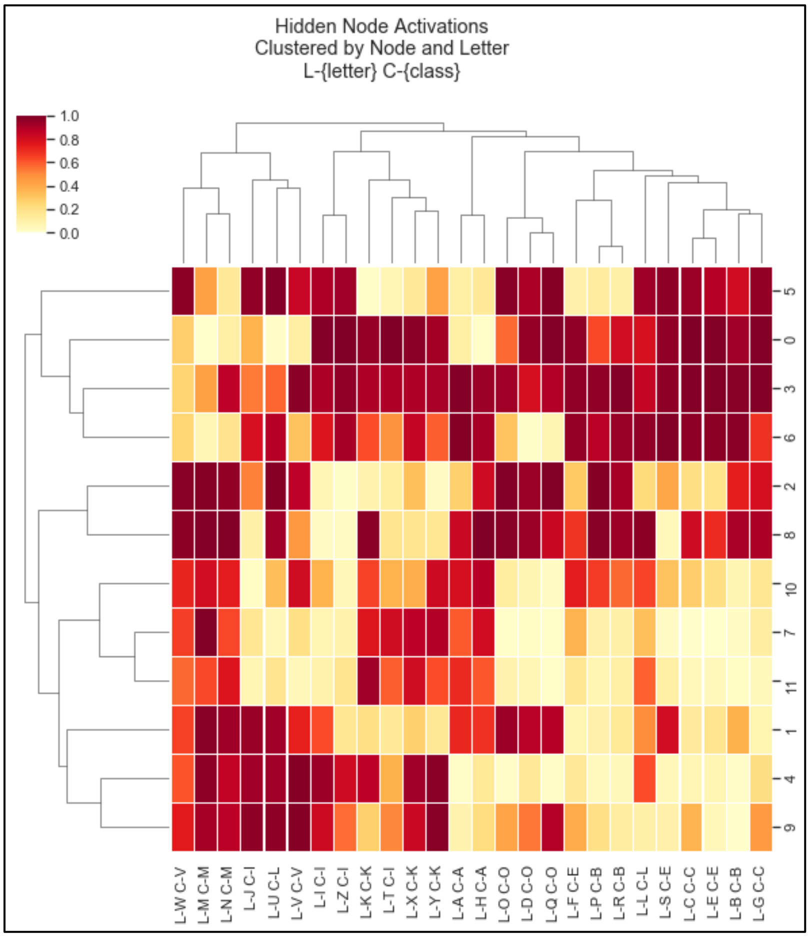


There are two major groups of nodes: Group 1: (1, 3, 4) and Group 2: (0, 2, 5). Here, we can see that node 5 is being activated in all letter clusters, which is not very meaningful. On the opposite end, node 1, and the majority of Group 1 is deactivated for the most part, with the exception of being activated by mostly class I. This can suggest that node 1 is inclined to be activated where there is a vertical line down the middle of the input grid. We can also see that for node 0, it looks to be activated by letters and classes with rounded letters such as “C”, “B”, “O”, and possibly “S” and “J”, while being deactivated by letters containing straight or diagonal lines, such as “A”, “M” and “V”.

Next, the baseline model was adjusted to having only two (2) hidden nodes as opposed to the original 6 to see how the nodes would activate. The following shows that not much can be derived as node 1 was activated for all letters and classes and with node 0 only activating for partial classes of V and M (potentially letters with top left-bottom right diagonals):



This finding made it easy to realize that increasing the number of hidden nodes would show activations of a more granular level. For the next test, the baseline model's hidden nodes increased to twelve (12), doubling the initial number of hidden nodes. The following displays the activations:



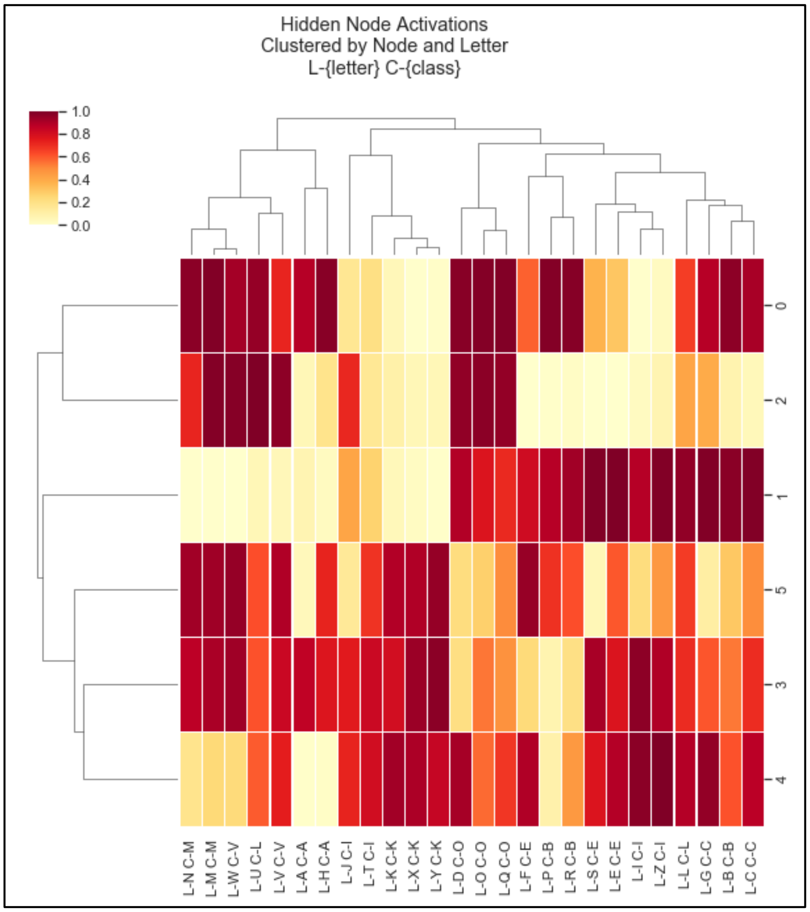
With a higher level of detail, the dendrogram can help us assess the relationships between nodes and letters. There are roughly three groups of hidden nodes that share relationships: Group 1 - (1, 4, 9, 11, 7, 10), Group 2 - (2, 8), and Group 3 - (0, 3, 5, 6). For the Group 1, it appears that the nodes are being activated when there are vertical or diagonal lines (such as "K"), while Groups 2 and 3 appears to have nodes activating when there are horizonal lines (such as "E") or when the letters are rounded (such as "B").

Comparing to the baseline model, we can determine better relational groups when the hidden nodes are increased. Even with an increased number of nodes, the majority of them are activated from a 0.6-1.0 magnitude for the class characteristics they are inclined to activate to.

4. Hidden Node Activations – Hyperparameter α

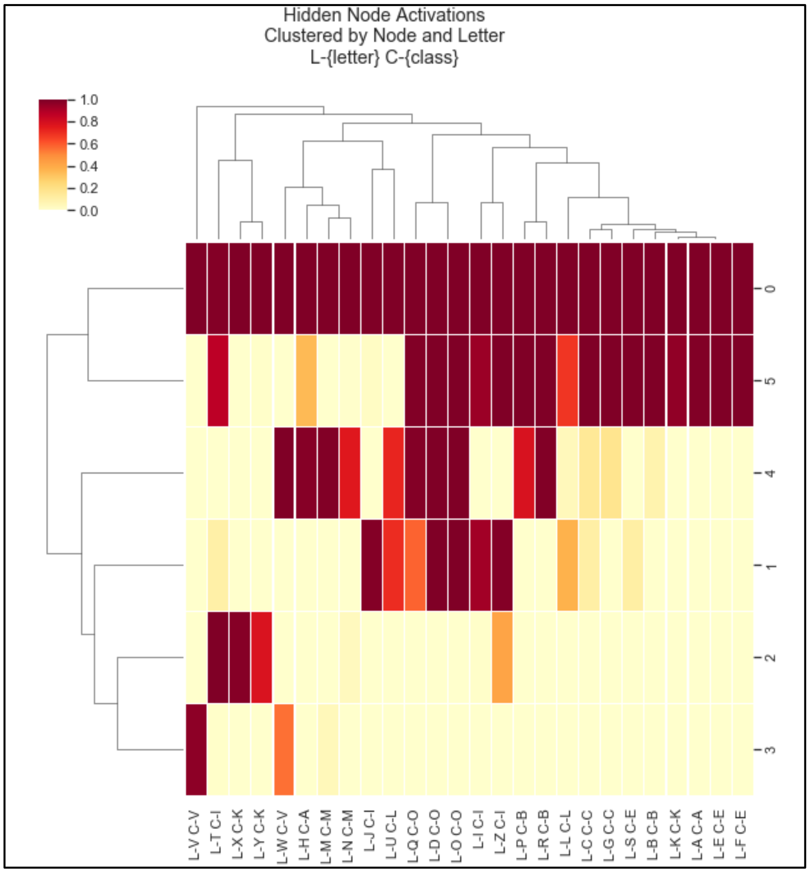
Now that the effects of adding hidden nodes to the baseline model can show different interpretations of hidden node activations to varying degrees, we will now explore what can be interpreted by the model when we change the fixed learning rate α.

First, the baseline model was adjusted to have α = 0.5:



Compared to the baseline model, it looks as if the node activations almost "inversed" from activation to deactivation and vice-versa. For example, in the baseline model, nodes 1 and 4 were deactivated for classes C and E, however when α = 0.5, the nodes were activated to a high magnitude. The baseline model for node 5 was activated to a high magnitude for every class, however the new decreased alpha parameter deactivated some of the classes, and then shifting the magnitude of activation down for the rest. In addition, the letter classes groupings remained quite similar to when α=1.0.

Now, let's look at the hidden node activations when α = 1.5:



This dendrogram looks a lot more similar to the baseline model, possibly suggesting that where α=0.5 converged by the end of training led to very different results compared to when α=1.0 and α=1.5. Compared to the baseline model, instead of node 5 being activated for all classes, looks as if class 0 is being activated for all classes, and nodes 2 and 3 are deactivated for the majority of the classes.

5. Conclusion

The results from the set of experiments proved that my hypotheses were incorrect for the most part. My hypothesis on the increased number of hidden nodes leading to a smaller ratio of nodes being activated was wrong as the increased amount of hidden nodes actually led to a better representation of how nodes were being activated by having more well defined class groups and hidden node groups. The most surprising result was actually how the learning rate affected the representation of hidden node activations. From the baseline (and increased learning rate) to the decreased learning rate of 0.5, the node activations almost inversed completely, which is fascinating and something I'd like to look into deeper if more time was allotted. This set of experiments was a very good exercise of learning how hyperparameters affect the model's hidden node activations.