Chase Jacobs

Neural Network – Week 8

1. Source code

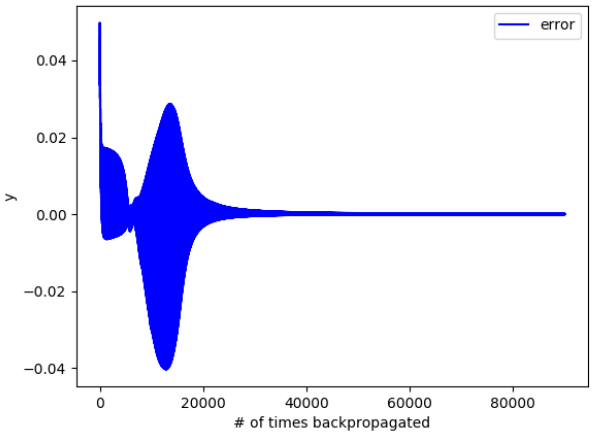
https://github.com/ChaseTJacobs/cs450/tree/master/week8

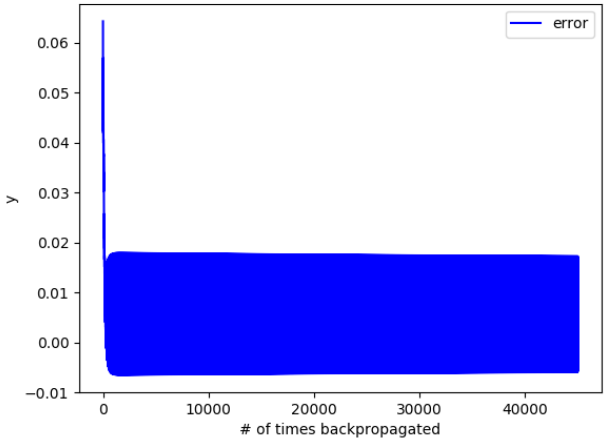
2. Approach

I chose the implementation option. For each node in the network I created a class. In each node is a "value" variable, which represents "a" in the equations, an "error" variable, and a "isBiasNode" Boolean variable. There is also a weights array, which correspond to the weights from that node to the nodes in the *previous* layer. Then the layers were represented by a list of Nodes, and the whole network was a list of layers.

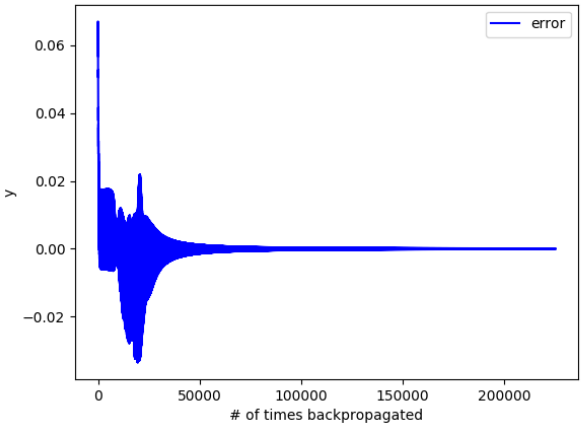
3. Challenges

The hardest part was figuring out what would work for storing the network. I went through a few iterations before I landed on what I have now. Then it was just a case of programming out the feed-forward and back-propagation methods. The back-propagation was probably the hardest thing to program due to the hidden nodes needing so much information from other nodes in order to calculate the error.





4. Results



Above are three different graphs of errors from the iris dataset. The top left graph has 2000 epochs with 3 hidden layers. The top right graph has only 1000 epochs with 5 hidden layers. The bottom graph has 5000 epochs with 4 hidden layers. The top left and bottom runs resulted in a very high percentage (~96%) each time, while the middle obviously didn't have enough epochs (or had too many hidden layers for the number of epochs) to get good results. The iris dataset was split into training and testing sets similar to previous assignments that used it. The testing set was used to calculate the success rate of the neural network.

Note: I included the result of the 1000 epochs graph in order to show what happens when you don't have enough epochs.

Here are the exact builds of each network for each of the runs:
Top left: 2000 epochs with 3 hidden layers. Each layer had 2, 4 and 3 nodes respectively.
Top right: 1000 epochs with 5 hidden layers. Each layer had 1, 2, 3, 4, 5 nodes respectively.
Bottom: 5000 epochs with 4 hidden layers. Each layer had 8, 8, 8, 2 nodes respectively.

The other dataset I used was the very simple one we did in class with the "is raining" or "isn't raining" possible outputs. The network built from this data set was very successful in telling whether or not it was raining based on the input.

5. Above and Beyond

N/A

6. Assessment
   D) Meets requirements.

Please state which category you feel best describes your assignment and give a 1-2 sentence justification for your choice:

I spent a lot of time making the neural network learn, but I don't think I went above and beyond, unless you count how beautiful my graphs are as above and beyond material. These are generated using matplotlib.pyplot and will be generated every time you run the program. They will save the folder the python program is in, so if you run my code, you'll have a few PNGs in that folder now.