COSC 4086 Machine Learning Hw 5 - 60 pts.

This homework is over the ANN section from Ch.2, including the material not in the text that was presented in class and lab.

1] (20 pts.) Use the code in the program **29_MLP_XOR_2hidden.py** (with 2 nodes in the hidden layer) as a starting point. We found that, for certain combinations of the random starting weights, the algorithm "gets stuck" in a local minimum, and it never reaches a correct solution to the classification problem.

A. Wrap the entire algorithm in a big loop and count how many times it finds a solution and how many times it gets stuck. It is OK to assume that, after 100,000 unsuccessful iterations, the algorithm is stuck. Run the big loop at least 1000 times, and estimate the probability of getting stuck.

B. Unlock the biases for all nodes (hidden and output), and allow them to change according to gradient descent. Repeat the exercise from part A. Has the probability changed? Discuss.

2] (10 pts.) Use the code in the program **28_MLP_XOR_3hidden_rate_biases.py** (with 3 nodes in the hidden layer) as a starting point. Modify it to add another hidden node, for a total of 4 hidden nodes.

Include only screenshots of the parts of the code that you changed, and of the relevant output.

3] (10 pts.) Compare the programs with 3 and 4 hidden nodes by running each 250 times and calculating the average number of iterations taken to converge.

Include only screenshots of the newly-developed code that you used to collect the statistics, and of the relevant output.

- Hint: Store the 250 numbers of iterations in a list, or a Numpy array.

What do you notice?

Try to explain the result.

4] (10 pts.) Complexity: Number of operations per iteration.

Use the code in the program **28_MLP_XOR_3hidden_rate_biases.py** (with 3 nodes in the hidden layer) as a starting point. **Remove all the calculations involving biases**.

Count how many arithmetic operations (+, -, *, /) are required in the functions **fwd( )** and **bak( )**, counted together. Assume that a call to **exp( )** counts as only one operation.

⇩

5] (10 pts.) Complexity - continued.

Generalize the count from the previous problem. The ANN has **i** input nodes and **h** nodes in the hidden layer. Express your result as a function of **i** and **h**.

If the operations are done on a sequential CPU, and each operation takes 1 ns, how long would it take to train a network whose inputs are 100x100 pixel images, having 200 hidden nodes, for 10,000 iterations? Use the appropriate unit of time :)