CMSC 352 Machine Learning **Due: 2.Nov.2020**
HW #5: Back-propagation Error Checking

When implementing the gradient-based learning algorithm for a deep-learning system, one frequently encounters difficulty getting every detail right. This is problematic, since the system will appear to learn, and may even learn fairly difficult problems, even though there are errors in the implementation of the algorithm.

An indispensable tool is gradient checking. Gradient check requires you to estimate derivatives by perturbing single parameters and then finding the change in the loss function. When estimating the derivative of the loss function with respect to a weight w, for example, modify w by a small constant amount (*h*) and then compute the difference to get a numerical estimate of the partial derivative:

$$\frac{\partial L(w)}{\partial w} = \frac{L(w+h) - L(w-h)}{(2h)}$$

You must look here to read a discussion of gradient checking and description of how to evaluate the relative error.

Directions

Starting with the bprop.py code I provide, complete the bprop and errCheck methods. This program implements a network with three layers, the last of which has logistic units and the sum-squared loss function.

First, I recommend you do your best to implement vanilla backpropagation with no momentum or other additions. You must implement code that can run a three layer network, in which each layer is of arbitrary size. Verify that your network can learn XOR sometimes.

- Try to employ Numpy matrix operations for your computations.
- Provide comments in your code.
- Limit lines to 100 characters.

Now implement error checking. Have your program print the RelError for the three weights matrices as indicated in errCheck. Use error checking with the four patterns of the XOR network. You should use gradient error checking to uncover any bugs in your backpropagation implementation.

Deliverables

1. The bprop.py code. Your submitted code should run the XOR problem using a network having two inputs, two hidden layers, each having two units, and a single output unit. Your submitted program should conduct gradient checking for each pattern, in order, exactly one time. For each iteration, it should output the iteration number followed by the relative error for w3, w2, then w1. It should not print anything else!
2. A one-page PDF report that describes the relative error you observed on XOR and your assessment of the correctness of your implementation.