Assignment 8 (Reference lecture: 08/30)

Due: 09/07 2:00 pm

Submission Instruction:

Step 1: A pdf file submitted on Gradescope. This file should contain the code and output. The easiest way to generate this file is to first do screenshots(Snipping Tool on Windows, Shift-Command-4 on Mac), then paste the screenshots on a word file and transform it into a pdf file. You can also directly transform an ipython notebook to pdf format: File -> Print Preview -> right click and Print -> change Print Destination to "Save as Pdf". You are welcome to share your proven method on piazza.

Step 2: Code file submission to either the email address(cse190na4ai@gmail.com) or your private repository. This file should match the pdf version in step 1 and will serve as a reference if we find some problems with the pdf file submitted.

Let $n = 10,\ t = 25$.

Using gradient descent scheme presented on week4 slides perform learning of neural nets to perform regression of this high dimensional data

- $NN1(n, m, n)$ (one hidden layer neural net with *linear* activations having n input/output neurons and m hidden layer neurons (this is a parameter in the code) no need to have bias b vector
  presented on slide 58

- $NN2(n, m, n)$ (one hidden layer neural net with *sigmoidal* activation function). No need to have bias b vector
  presented on slide 58

Let $X =$ np.load('assignment8_X.npy') ($t * n$ dimensional matrix) ;
Y = np.load('assignment8_Y.npy') ($t * n$ dimensional matrix) be matrices storing training pairs (i-th row of X and Y is the i-th training pair (X_i, Y_i).

Set initial weight matrices using W1 $=$ 0.1 * np.random.randn(m,n),
W2 $=$ 0.1 * np.random.randn(n,m).

To minimize the *Mean Square Loss*
$L(W) = \frac{1}{t}\sum_{i=1}^{t}\left\|f_W(X_i) - Y_i\right\|_2^2$ perform computing the gradients of Mean Square Loss function as presented on week4 slides, in particular compute the gradients of the neural net using backprop algorithm.

Perform 500 steps of gradient descent using $\alpha = 0.01$ to train the nets.
Using three sizes of networks $m = 10, 25, 50$.

Output

- Final loss value,
- Plot of loss values along the gradient descent path,
- (final W2) * (final W1)

<u>Partial credit rubric:</u>

Total: 10 pts

- NN1, relatively low final loss on each of the three nets corresponding to m = 10, 25, 50 **(4.5 pts)**, correct plot and correctly output (final W2) * (final W1) **(0.5 pt)**
- NN2, relatively low final loss on each of the three nets corresponding to m = 10, 25, 50 **(4.5 pts)**, correct plot and correctly output (final W2) * (final W1) **(0.5 pt)**

<u>Reminder on assignment grades:</u>

ALL assignments are graded on a 10 pts scale, BUT they will be weighted according to their difficulties in calculating the total score for all assignments. In particular, later assignments (like this one) are more difficult than earlier ones, thus will be weighted more.