Homework: Neural Networks
(Fully Connected)

Problem 1: Two-class classification with neural networks (40 points)

Use a multi-layer fully connected neural network architecture to perform nonlinear classification on the provided dataset (eggs.csv). Here, we choose the network to have four hidden layers with ten units in each layer, and the tanh activation. We then tune the parameters of this model by minimizing an associated two-class Softmax cost via gradient descent.

Provided below is a sample visualizing of the nonlinear decision boundary learned in the top row of Figure along with the dataset itself. You need to produce a scattered plot like the one on the right panel of Figure 1, but it is optional to plot the (black) decision boundary. In addition, produce a cost history plot and an accuracy history plot like the example given in Lecture Slide 54.

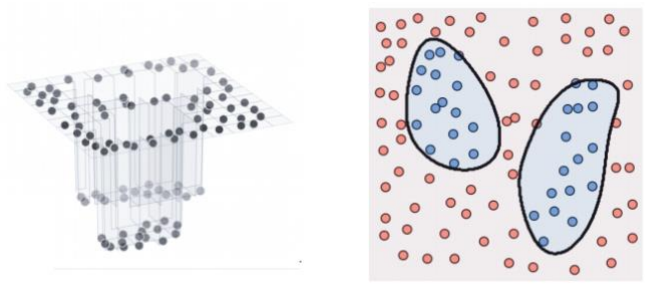

Figure 1

Use the Python code in Lecture Slides to create and initialize neural network architecture, and use autograd to compute the gradients.

Problem 2: Number of weights to learn in a neural network (20 points)

(a) Find the total number Q of tunable parameters in a general L-hidden-layer neural network, in terms of variables expressed in the layer sizes below

```
layer_sizes = [N, U_1, ..., U_L, C]
```

(b) Based on your answer in part (a), explain how the input dimension N and number of data points P each contributes to Q. How is this different from what you saw with kernel methods in the previous chapter?

Problem 3: Early stopping cross-validation (40 points)

Experiment:
This experiment illustrates the early stopping procedure using a simple non-linear regression dataset (split into 2/3 training and 1/3 validation), and a three-hidden-layer neural network with ten units per layer, and with tanh activation. Three different steps from a single run of gradient descent (for a total of 10, 000 steps)

is illustrated in Figure 2, one per each column, with the resulting fit at each step shown over the original (first row), training (second row), and validation data (third row). Stopping the gradient descent early after taking (around) 2000 steps provides, for this training-validation split of the original data, a fine nonlinear model for the entire dataset.

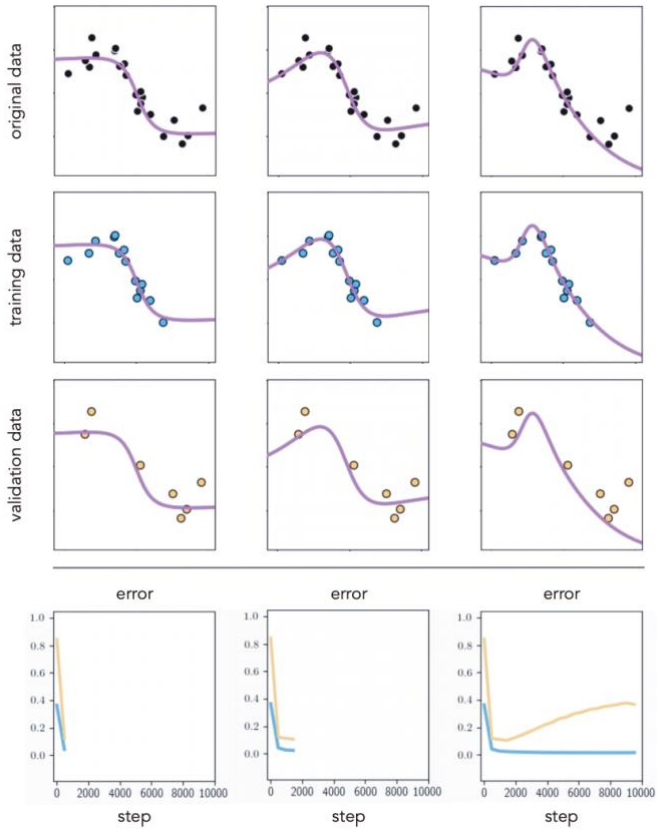

Figure 2

Problem:
Implement the above experiment with the above a) training-validation split b) network architecture and c) training configuration on the provided dataset (noisy_sin_sample.csv). You need only to produce the top and bottom plots in the **last** column, i.e., the top right (on original data) and the bottom right (the error curve till the last step).

Bonus question (10 pts):
If you also correctly produce the plots of rows 1-3 of the **middle** column (you may or may not combine the three plots into a single one, with legend), and the error history (the rightmost plot in row 4), you will gain 10 bonus points. In other words, the full marks of this homework become **110 points** if you choose to do so. (Of course, answering this question is optional.)