Deep Learning HW 1 Answer Sheet

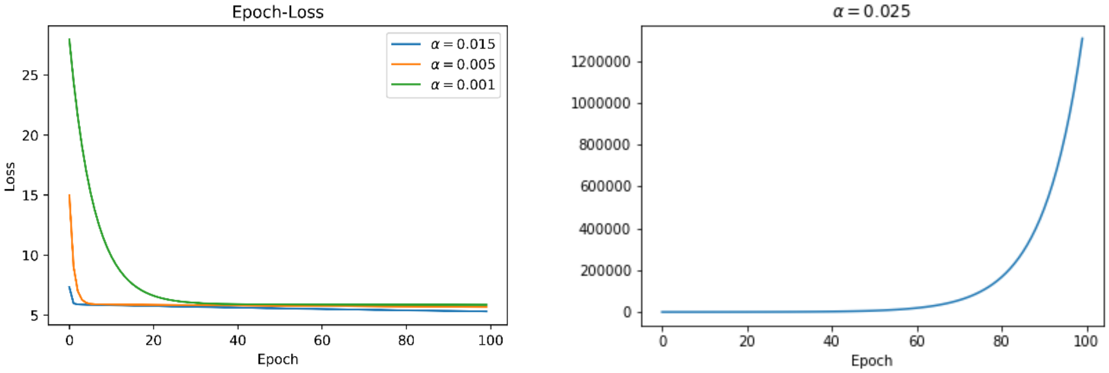
Student's Name: **Min-Chun Wu**

Problem #1: Univariate Linear Regression (25 points)

Q1: What happens when you change the step-size α?

Ans:

The following figure presents the Loss versus Epoch results when choosing different alpha but fixing the number of epochs to be 100.



For alpha being too large (as shown on the right figure for alpha = 0.025), gradient descent does not even converge.

The left figure indicates that, under the condition that the chosen alpha is going to make the loss converge, smaller alpha will make the convergence slower while larger alpha will make the convergence faster.

Q2: How many epochs did you need to converge to a reasonable solution (for any given step size)?

Ans:

As the left figure indicates, to arrive at loss with value around 5,
for alpha = 0.015, around 2 steps will suffice,
for alpha = 0.005, around 5 steps will suffice, while
for alpha = 0.001, at least 30 steps are require.

Problem #2: Polynomial Regression & Regularization (25 points)

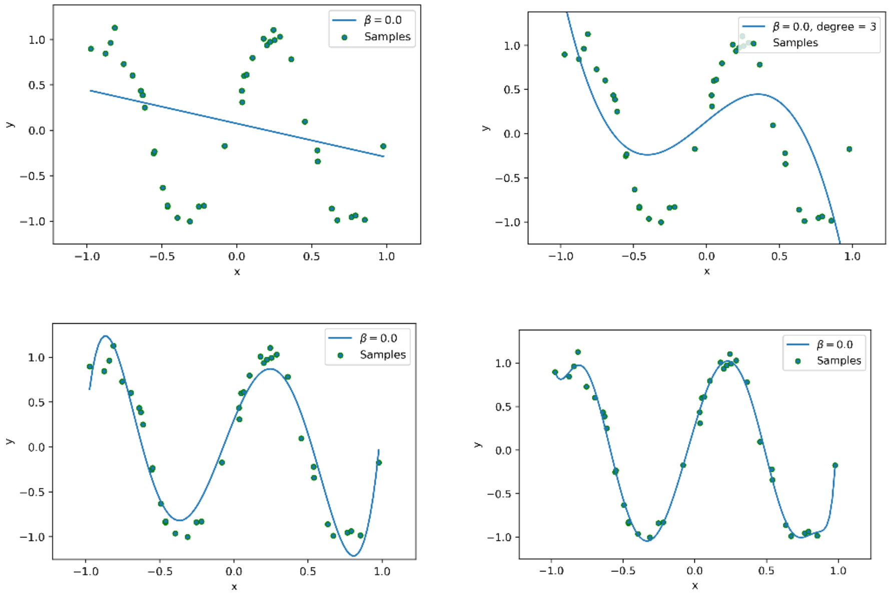
Q1:
With respect to the feature mapping,
what is a potential problem that the scheme we have designed above might create?
What is one solution to fixing any potential problems created by using this scheme (and what other problems might that solution induce)?

Ans:

One obvious problem would be what degree for the polynomial should we use.
One possible solution to it is to use validation (or cross validation) to tune this hyper-parameter.
However, this will cost quite a few examples, which is a subsequent issue especially when our data size is not very large.

Q2:
Fit a 1st order (i.e., linear), 3rd order, 7th order, 11th order, and 15th order polynomial regressor to the data in prob2_data.txt.
What do you observe as the capacity of the model is increased?
Why does this happen?

Ans:

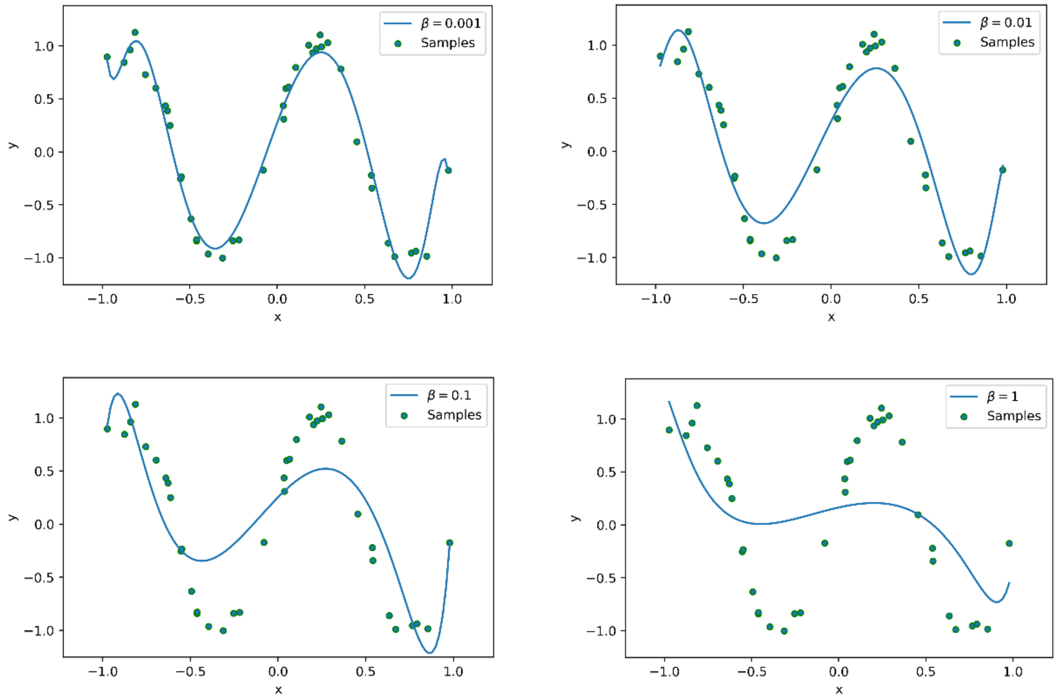


Figures above are the plotting results for degree = 1, 3, 7, 15, resp.
There are two main points to be noticed.
(1) As the model capacity increases, our final hypothesis fits the data better. The reason is quite obvious since our model has more capacity to fit the data.
(2) As the model capacity increases, it is more possible that the final hypothesis will fit the data too well, incurring “overfitting”. The reason is as the previous point.

Q3:

What do you observe as you increase the value of β? How does this interact with the general model fitting process (such as the step size α and number of epochs needed to reach convergence)?

Ans:



As β is increased, we are putting more bias towards preferring theta = (b,w) with smaller |w|. Moreover, the regularized cost becomes larger.

Hence, the number of epochs needed to reach convergence become larger.

However, the allowed maximal step size α should not change too much since if we use α larger than the un-regularized case, then the step size for the first term of our regularized cost function would be too big to gradually converge.

Q4:

(1) Comment (in your answer sheet) as to how many steps it then took with this early halting scheme to reach convergence.

(2) What might be a problem with a convergence check that compares the current cost with the previous cost (i.e., looks at the deltas between costs at time t and t − 1), especially for a more complicated model?

(3) How can we fix this?

Ans:

(1) The following report is for beta = 0, alpha = 1.6 and eps = 0.000001.

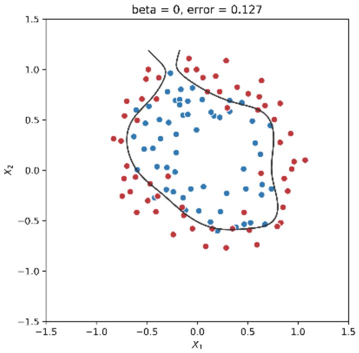
For degree = 1, 8 steps. For degree = 3, 171 steps. For degree = 7, 9513 steps. For degree = 15, 5377 steps.

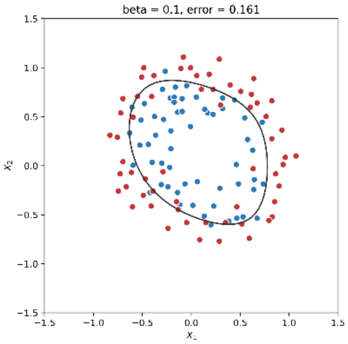
(2) If, unfortunately, we are at a point where the cost function is very “flat”, then the gradient will have a very small norm and the cost function only decreases with a very small amount. Hence, our early halting will be triggered and we are only stopped at a very flat place, not guaranteeing a global minimum or even a point close to the global minimum. This situation is called “saturating.”

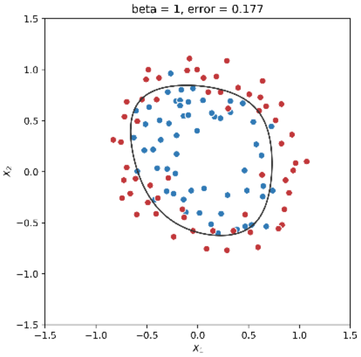
(3) To fix the saturating issue, one possible solution is to look at the log of the cost function instead of the original cost function. In this case, implement gradient descent on the log of the cost function. Since log(x) for x close to 0 is very large (in norm), we will overcome the saturating issue in this way.

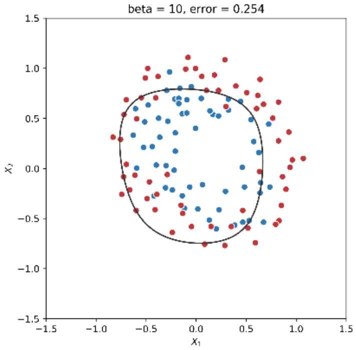
Problem #3: Multivariate Regression & Decision Boundaries (50 points)

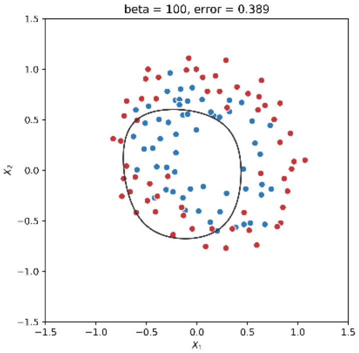
(1) Copy each of the resultant contour plots to your answer sheet and report your classification error for each scenario.
(2) Comment (in the answer sheet) how the regularization changed your model's accuracy as well as the learned decision boundary.
(3) Why might regularizing our model be a good idea if it changes what appears to be such a well-fit decision boundary?
Ans:
(1)


beta = 0, error = 0.127


beta = 0.1, error = 0.161


beta = 1, error = 0.177


beta = 10, error = 0.254


beta = 100, error = 0.389

(2) As beta goes up, the decision boundary becomes more stable but meanwhile the classification error (training error) goes up due to the model being more restrictive.

(3) There might be some amount of noise in the data, whence a final hypothesis with very low training error is not necessary a good hypothesis; it might have a very high generalization error, leading to overfitting. The reason for adding the regularization term is to reduce the generalization error. In fact, it can be proved that a bounded smooth convex learning problem is PAC learnable via the RLM (regularized loss minimization) paradigm, meaning the regularization term is a guarantee to make the problem learnable.