Samarth Kalluraya

Assignment 3

EM 531

Problem 1)

(i)

$$y(x) = 2 * \sin(2\pi x) + \sin(8\pi x) + 0.5 * \sin(16\pi x)$$

This equation has the highest sin term of 16πx. Thus, in order to best fit the cure we need to include the term sin 16πx in our feature. Since our features are a Fourier series where the last term is sin(Mπx), M must have a minimum value of 16 to correctly fit the data. If M is less than 16 then the predicted curve would not match the curves generated due to the presence of the sin(16πx) term in the equation.

Problem 1)

(ii)

The plots for different learning rates and batch sizes are shown below. The first graph plots the model prediction (red line), exact solution (yellow line) and the training data set (blue points). The second graph plots the loss vs iterations of the Adam optimization

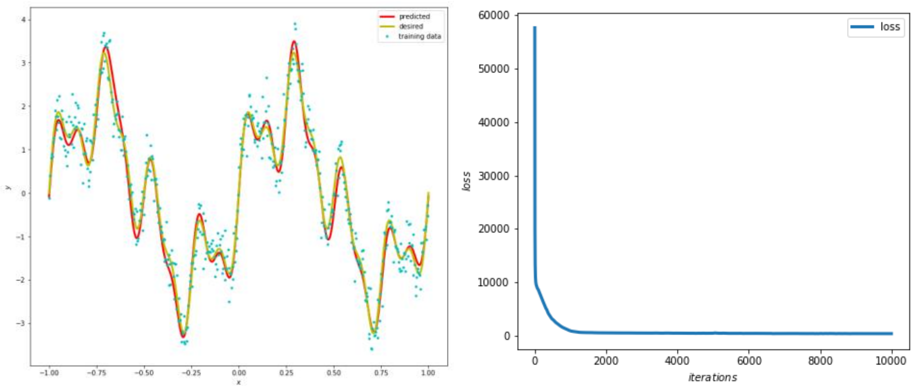
A guide to the plots is given in this table:

Figure	Learning rate	Batch size	Loss
1a	0.001	1	132.79
1b	0.001	64	120.90
1c	0.001	500	109.58
2a	0.0001	1	4220.64
2b	0.0001	64	3045.53
2c	0.0001	500	2885.91
3a	0.4	1	1234.14
3b	0.4	64	657.28
3c	0.4	500	616.53

1. **Learning rate 0.001**

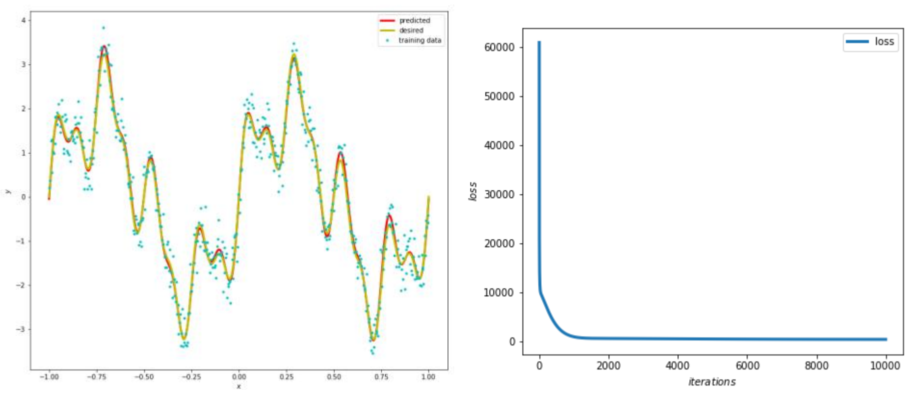
1a) Batch size 1

Final loss: 132.798456136484



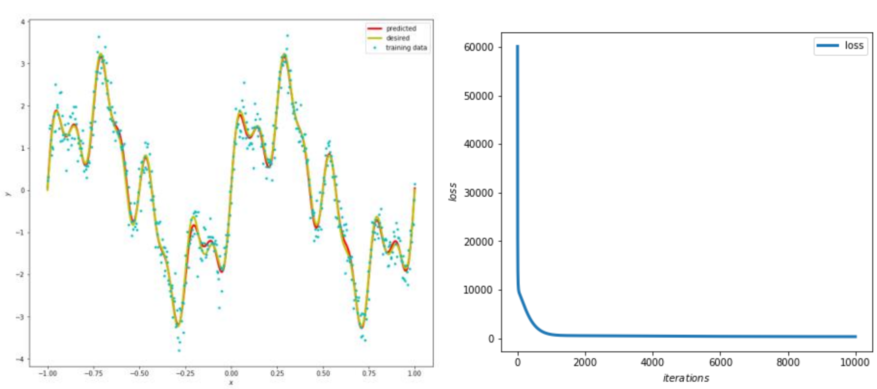
1b) Batch size 64

Final loss: 120.90401274412875 loss



1c) Batch size 500

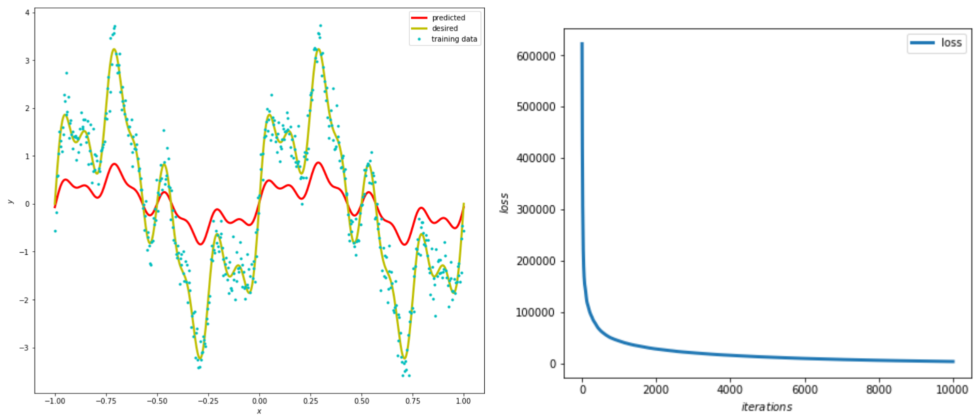
Final loss: 109.5841658654189 loss



2. Learning rate 0.0001

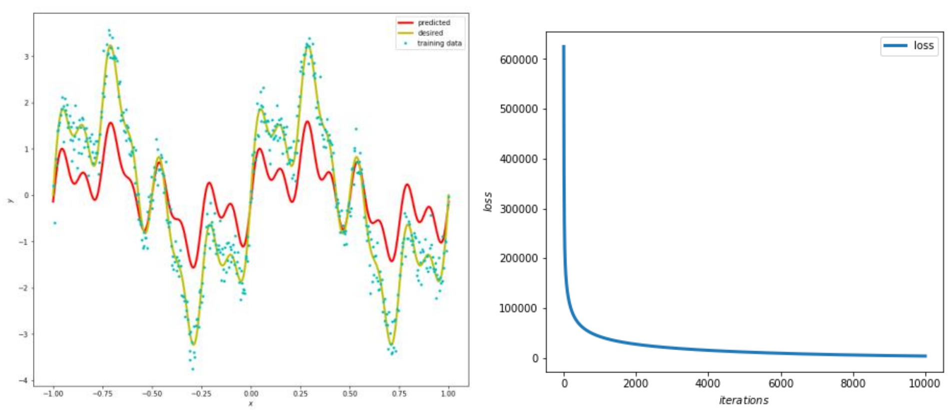
2a) Batch size 1

Final loss: 4220.642665640096 loss



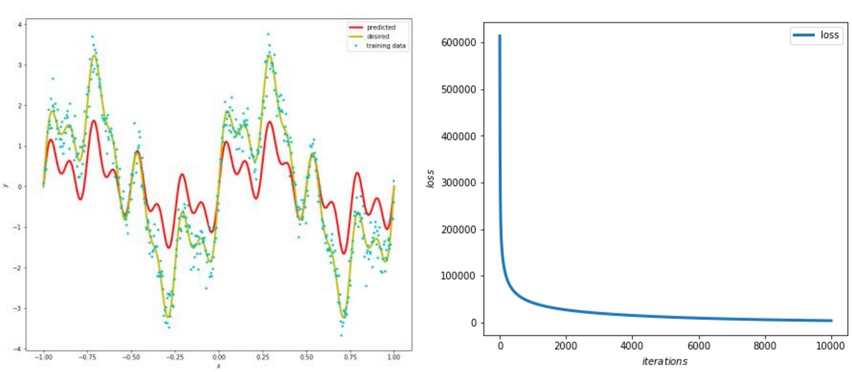
2b) Batch size 64

Final loss: 3045.5346623900177 loss



2c) Batch size 500

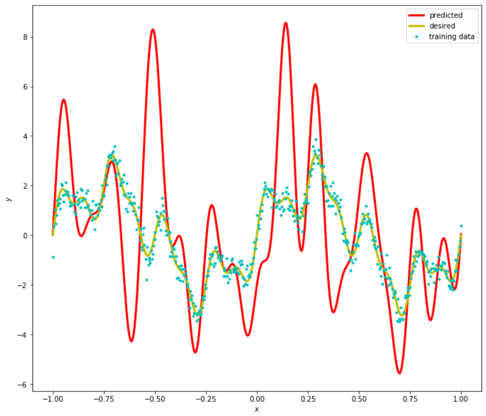
Final loss: 2885.9142530924737 loss

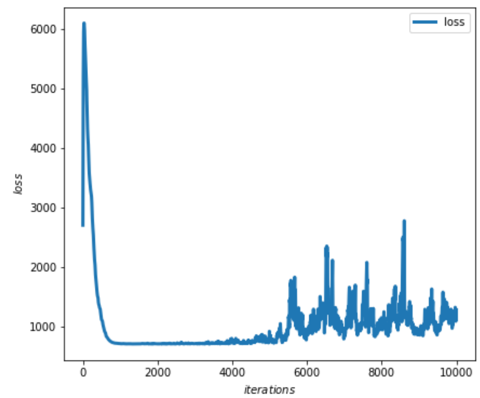


3. Learning rate 0.4

3a) Batch size 1

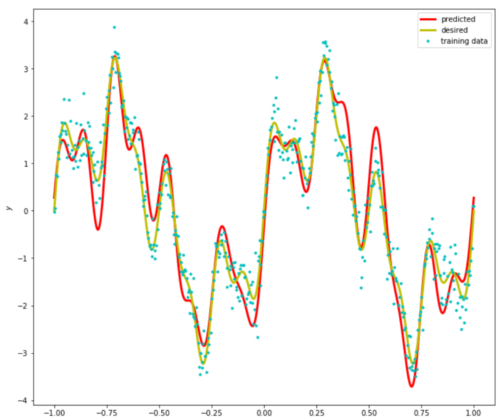
Final loss: 1234.1485749075655 loss

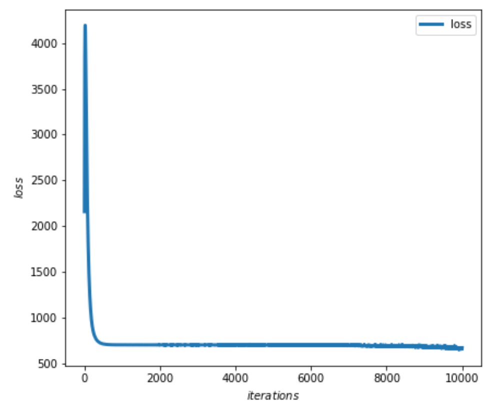




3b) Batch size 64

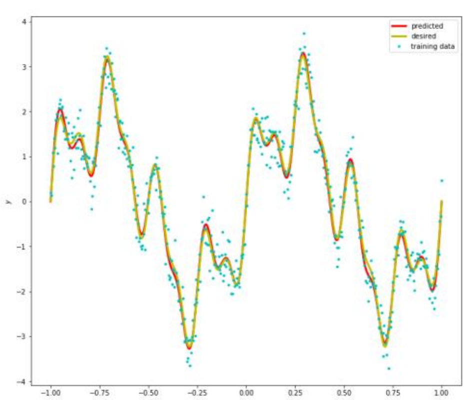
Final loss: 657.2833377144029 loss

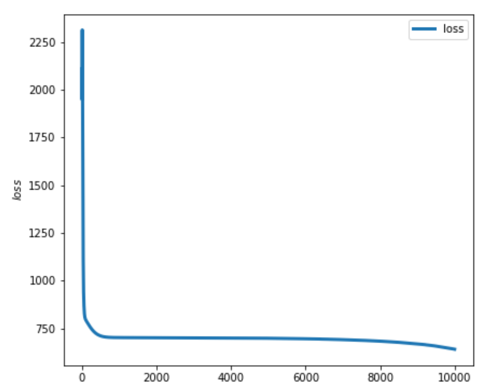




3c) Batch size 500

Final loss: 616.53156454813077 loss



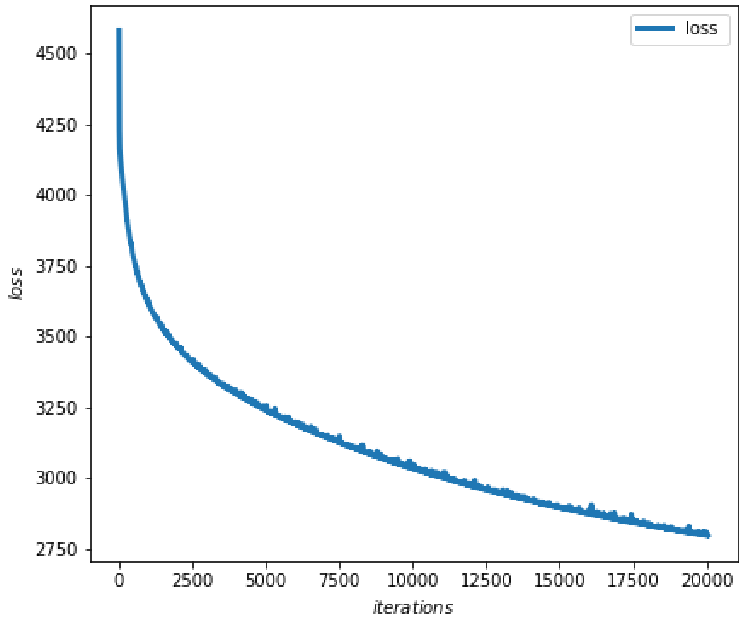


We can clearly summarize the following observations:

- We can see that initially as we have set all the parameters to 0, the loss is very high. The gradient descent algorithm then brings down the loss rapidly till it starts to stabilize at a certain value. The value of the loss is dependent on the batch size as well as the learning rate.
- Learning rate:
  a) The learning rate determines the step size that the gradient descent takes in every iteration. The when the learning rate is very small (in our case 10^{-4}) the gradient descent takes a large number of iterations to bring down the loss. Thus, in the 10,000 iterations, the loss does not decrease that much and hence the parameters are not tuned to their optimum value. Thus, we can see in that the prediction is very different from the desired values.
  b) When the learning rate is very large (in our case 0.4), the gradient descent takes large steps and hence is not able to converge on to stable values. This is clearly visible in the figure 3a where the values of loss oscillates towards the end.

- Batch size:
  a) We can see that as we make the batch size smaller, for the same number of iterations and the same learning rate, the loss increases. This is because as we reduce the batch size the training model makes use of lesser number of data points and hence it would not be as finely tuned as the parameters of the larger batch size.

Problem 2)



Binary cross-entropy loss vs iterations

Loss value at last iteration = 2798.2639

Confusion matrix => $\begin{bmatrix} 902 & 321 \\ 410 & 1700 \end{bmatrix}$

Classification Accuracy = 78.067%