Mengke Li
Email: mengkel@clemson.edu
Course: CPSC 8430 - Deep Learning
Instructor: Dr. Luo

Homework - #1
ID: C28207748
Term: Spring 2023
Due Date: 11^{th} Feb, 2023

Problem 1-1: Simulate a Function

1 Describe the models you use, including the number of parameters (at least two models) and the function you use.

2 In one chart, plot the training loss of all models.

3 In one graph, plot the predicted function curve of all models and the ground-truth function curve.

4 Comment on your results. Use more than two models in all previous questions. (bonus) Use more than one function. (bonus)

Solution:

1 In this training, three models are used with 7 (model1), 4(model2), 2(model3), hidden layers in each model. The activation function is tanh, and the optimizer is SDG, while the loss function is MSE loss.

2 The training losses of all models are shown in the upper panel of Fig.1

3 The predicted functions of all models are shown in the lower panel of Fig.1

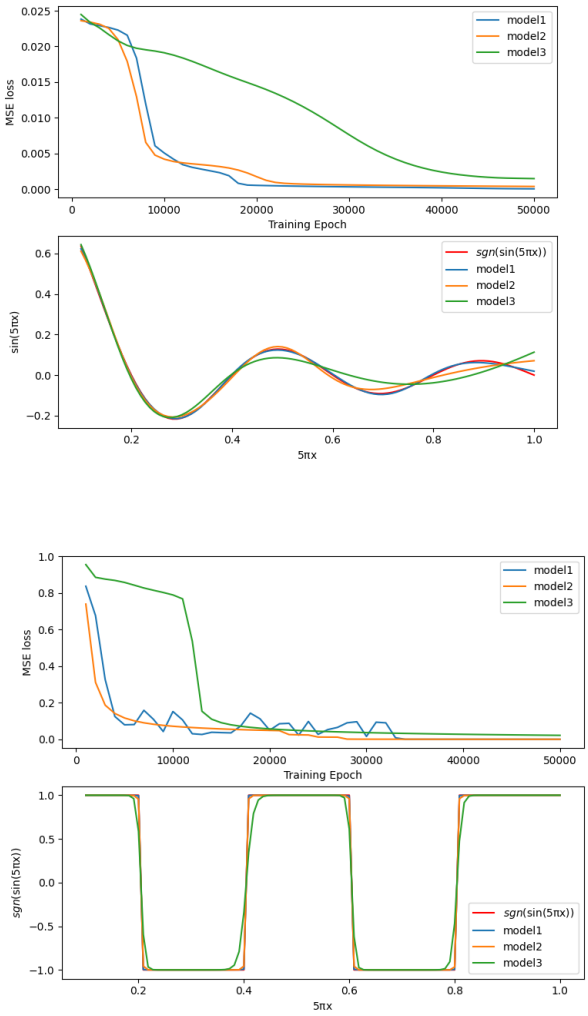

Figure 1: The left figure show the function $sin(5\pi x)$ vs. $5\pi x$. The right figure shows the function panel $SIGN(sin(5\pi x))$ vs. $5\pi x$. In each figure, the upper panel presents the $MSEloss$ as a function of training epoch. The lower panel present the fitting to ground truth data.

4 Those plots make sense. From the loss curves, we can see that the more complex the neutron network is, the better the results are. Model three is the simplest one with only one hidden layer, the training loss decreases slowly, while model1 and model2 perform better with more hidden layers. The same phenomenon can be seen in the prediction curve. Model1 amd model2 have a better fit than model3.

Problem 1-2: Train on Actual Tasks

1 Describe the models you use and the task you chose.

2 In one chart, plot the training loss of all models.

3 In one chart, plot the training accuracy.

4 Comment on your results. Use more than two models in all previous questions. (bonus) Train on more than one task. (bonus)

Solution:

1 In this training, the training and testing data are downloaded from MNIST data set. Two CNN layers and one fully connected are used with MaxPooling method. In my jupyter notebook file, two different network structures are shown.

2 The training losses of all models are shown in the left panel of Fig.2

3 The model accuracy of all models are shown in the right panel of Fig.2

4 Those plots make sense. Two different model with different structures will give different training results. Because the network structures are similar, the loss and accuracy are comparable. Both of those two model show loss decreasing and accuracy increasing as the training process goes.

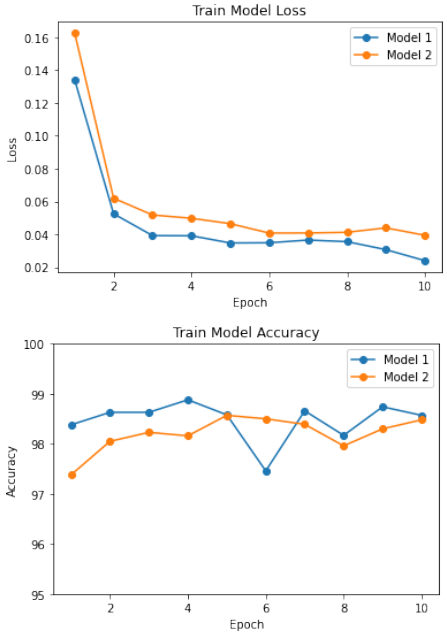

Figure 2: The left figure show the training loss as a function of training epoch. The right figure shows the accuracy as a function of epoch

---

Problem 2-1: Visualize the optimization process

1 Describe your experiment settings. (The cycle you record the model parameters, optimizer, dimension reduction method, etc)

2 Train the model for 8 times, selecting the parameters of any one layer and whole model and plot them on the figures separately.

3 Comment on your result.

Solution:

1 Model description:

- 3 Dense layers with dimension 748*15*30*10
- Activation function: relu
- Optimizer: Adam
- Loss: Cross Entropy Loss

2 The two important components for eight runs are shown is Fig.3. The left one demonstrate the important components for the weights on the first layer; while the right one presents the reduced dimension for the whole model.

3 From Fig.3 we can see that using PCA can successively reduce the dimension and get the important to features that describe the important weights.

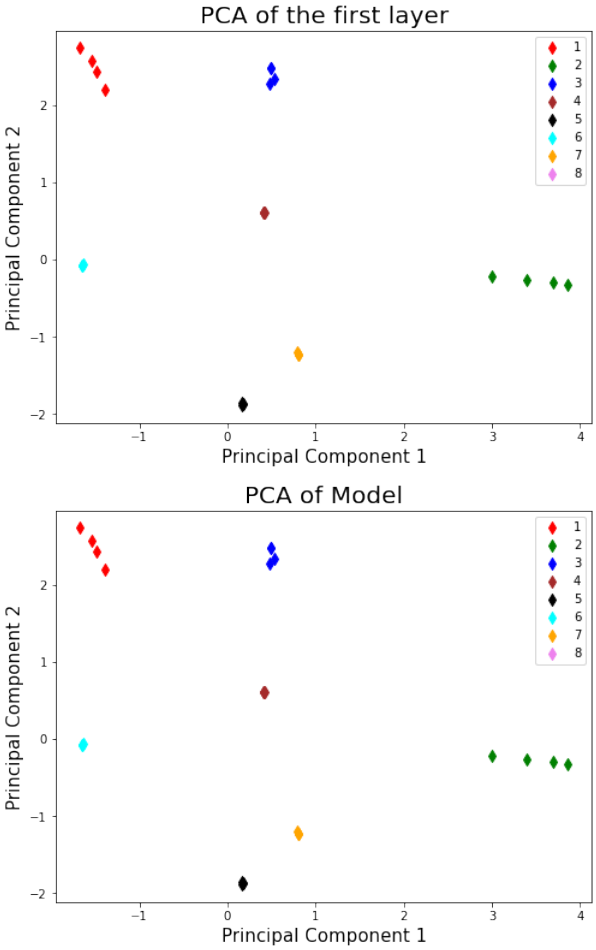

Figure 3: The left figure shows the reduced dimension for the first layer, while the right figure shows the reduced dimension for the whole model.

Problem 2-2: Observe gradient norm during training and comments on gradient being almost zero

1 Plot one figure which contain gradient norm to iterations and the loss to iterations.

2 What happens when gradient is almost zero?

3 State how you get the weight which gradient norm is zero and how you define the minimal ratio.

4 Comment your result.

Solution:

1 The loss vs. epoch and gradient norm vs. epoch plots are shown in Fig.4

2 If the gradient is almost zero, that means in the training process the loss function is very close to its local or global minimum. Therefore, the trained model is reaching the best model.

3 If the gradient norm is close to zero, the weight is almost the same, because the learning process is updating its weights based on the learning rate and the gradient. If gradient is zero, δW is also close to zero. The minimal ratio can be computed by getting the second derivative of the loss function. If the minimal ratio is positive and greater than zero, that means the model is getting to the best training point.

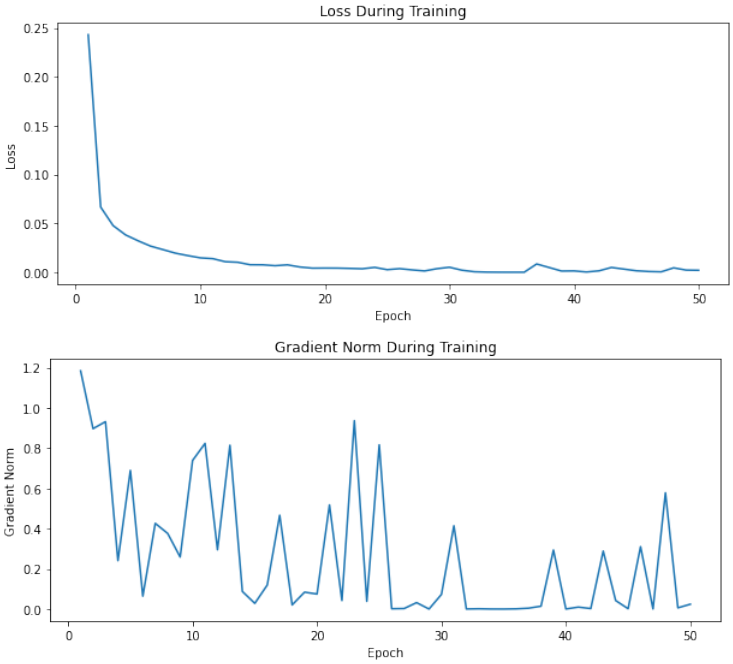

Figure 4: The upper figure shows the loss vs. epoch, while the lower panel demonstrate the gradient norm vs. epoch.

The same process for training a function $sin(5\pi x)$ is shown in Fig.**??**

Problem 3-1: Can network fit random labels?

1 Describe your settings of the experiments. (e.g. which task, learning rate, optimizer)

2 Plot the figure of the relationship between training and testing, loss and epochs.

Solution:

1
- 2D convolution layers with dimension 1*16*32 with kernal size 5 with 2D Max Pooling
- 2D Dense layer (32*5*5)*50*10
- Activation function: relu
- Optimizer: Adam
- Loss: Cross Entropy Loss

2 The training, testing losses vs. epochs are shown in Fig.5. From this plot we can see that though the training loss is decreasing as the learning process goes, the testing loss climes up. Which means the model is over-fitting the training set, and the predictive power is weak. This infers that this CNN model cannot fit with random label.

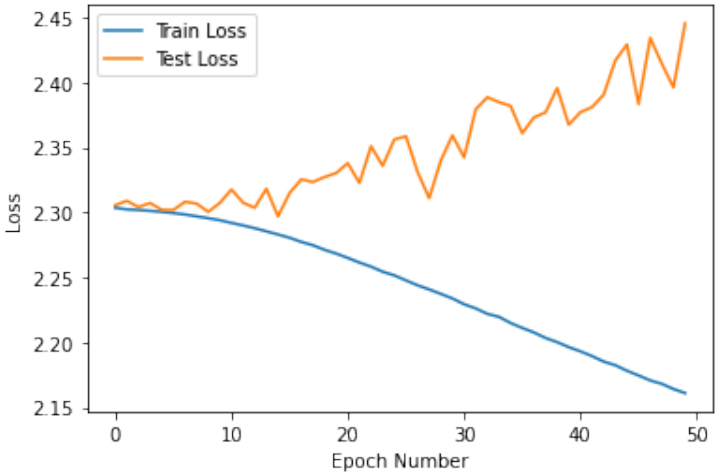

Figure 5: The training and testing losses as a function of epoch number

Problem 3-2: Number of parameters v.s. Generalization

1 Describe your settings of the experiments. (e.g. which task, the 10 or more structures you choose)

2 Plot the figures of both training and testing, loss and accuracy to the number of parameters.

3 Comment your result.

Solution:

1
- Trained on MNIST data set
- 2D Dense layer 745*5*30*10, similar structre for other 9 models
- Activation function: relu
- Optimizer: Adam
- Loss: Cross Entropy Loss

2 The training, testing losses vs. epochs are shown in Fig.6. From the left plot we can see that when the model parameters are increasing, the training and testing loss have a decreasing trend, while the accuracy data has an increasing trend. The models tested here are only 10 models, if more models are trained, this trend will be more obvious.

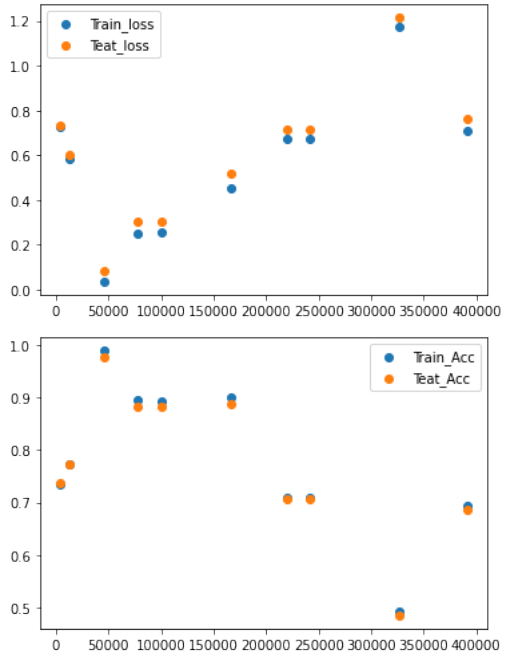

Figure 6: The left figure shows the training and testing losses as a function of training parameters. The right figure shows the training and testing accuracy as a function of model parameters.

Problem 3-3: Flatness v.s. Generalization

1 Describe the settings of the experiments (e.g. which task, what training approaches)

2 Plot the figures of both training and testing, loss and accuracy to the number of interpolation ratio.

3 Plot the figures of both training and testing, loss and accuracy, sensitivity to your chosen variable.

Solution:

1
- Trained on MNIST data set
- 2D Dense layer 745*5*30*10, similar structre for other 9 models
- Activation function: relu
- Optimizer: Adam
- Loss: Cross Entropy Loss

2 The training, testing losses vs. epochs for different batch sizes are shown in Fig.7. We can clearly see that the batch size will affect the training and testing loss. Similarly for the training and testing accuracy. Even though the deviation is not significant, to get a better performance, we need to test on different batch size to get the best model.

The sensitivity evolution as a function of batch size is plotted in Fig.9. The sensitivity peaks at batch size around 4500, which means there is a best choice for the training batch size.

The interpolation accuracy plot is shown in Fig.10.

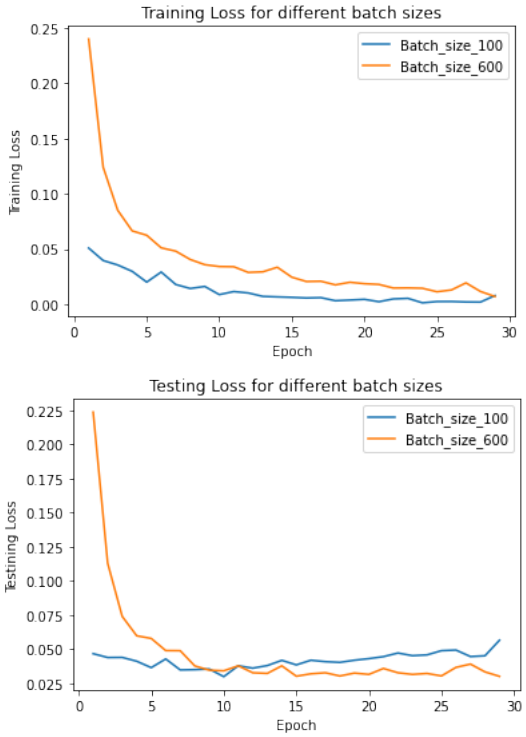

Figure 7: The figure shows the training and testing losses as a function of training epoch with different batch size.

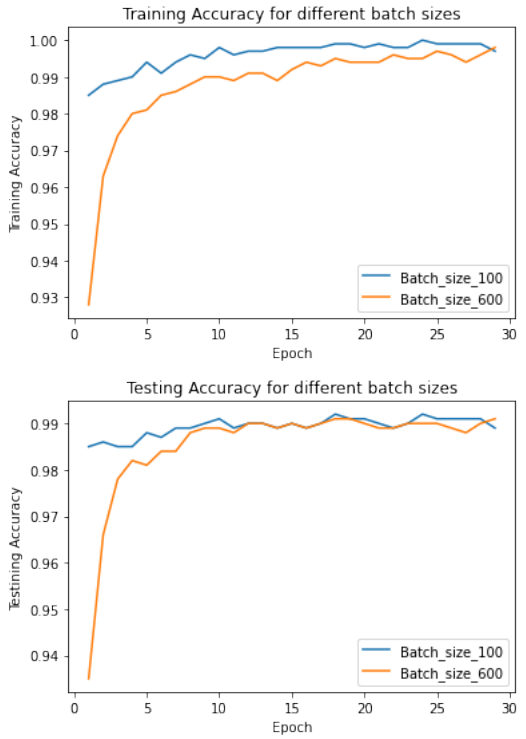

Figure 8: The figure shows the training and testing accuracy as a function of training epoch with different batch size.

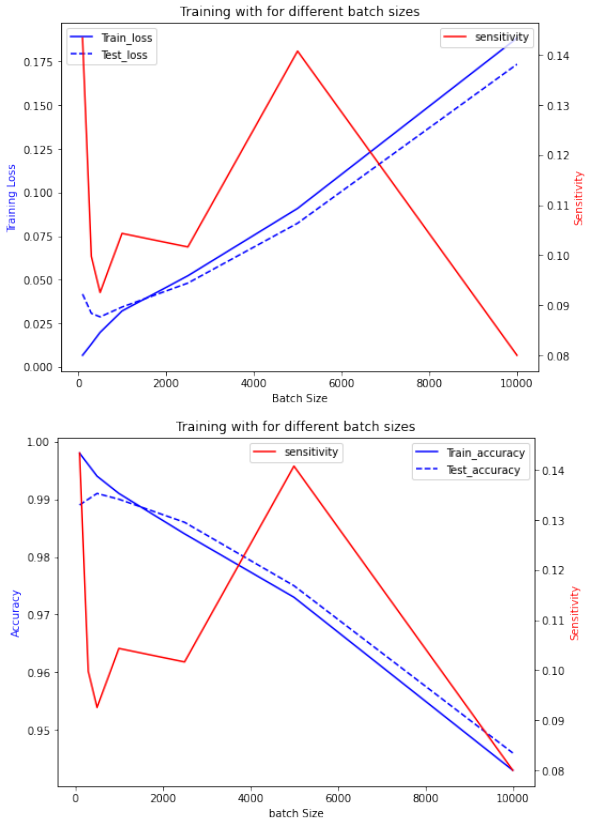

Figure 9: The figure shows the training and testing accuracy as a function of training epoch with different batch size.

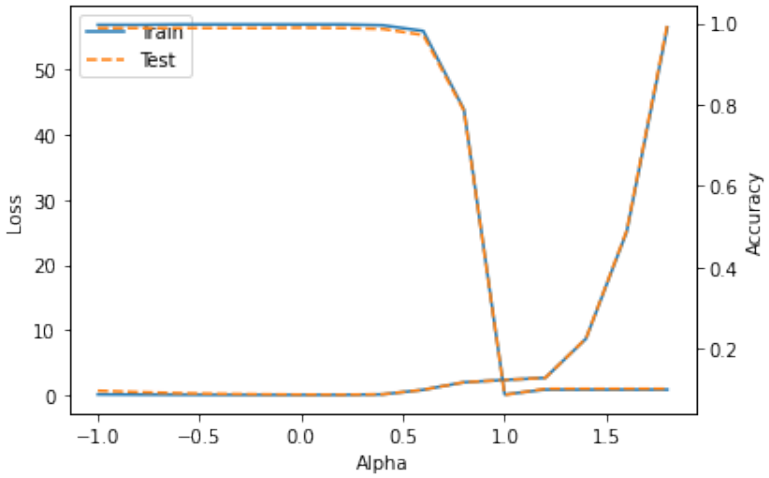

Figure 10: The figure shows accuracy of interpolation ratio as a function of α