Q1.

~~Base~~ Q1

$$\frac{P(y_i=1 \mid x_i; w)}{P(y_i=0 \mid x_i; w)} > 1$$

because $\therefore P(y_i=1 \mid x_i; w) = \frac{1}{1+e^{-w^T x_i}}$

$\therefore P(y_i=0 \mid x_i; w) = \frac{e^{-w^T x_i}}{1+e^{-w^T x_i}}$

$$e^{w^T x} > 1$$

$$w^T x > 0$$

then

$$b + \sum_{i=1}^{d} w_i x_i > 0 \quad \text{because} \quad b + w^T x > 1 \text{ and } w^T x > 0$$

Q2.

$$\frac{P(y=1)\prod_{i=1}^{d} P(x_i \mid y=1)}{P(x_1, \dots, x_d)}$$

$P(y=1 \mid x_1 = x_1, x_2 = x_2) > P(y=1 \mid x_1 = x_1)$ (2)

(1) ↳ $\frac{P(y=1)\prod_{i=1}^{2} P(x_i \mid y=1)}{P(x_1, x_2)} = \frac{P(y=1) \cdot P(x_1 \mid y=1) \cdot P(x_2 \mid y=1)}{P(x_1, x_2)}$

Here, we can assume

$P(x_1 \mid y) = P(x_2 \mid y)$ ∵ x_2 is exactly copy

$$= \frac{2 \cdot P(y=1) \cdot P(x_1 \mid y=1)}{P(x_1, x_2)}$$

(2) $P(y=1 \mid x_1 = x_1) = \frac{P(y=1) \cdot P(x_1 \mid y=1)}{P(x_1)}$

$\frac{2 \cdot P(y=1) \cdot P(x_1 \mid y=1)}{P(x_1, x_2)}$ this equation shows that

x_2 does not exist Because the model does not include x_2

Such that

$$= \frac{2 \cdot P(y=1) \cdot P(x_1 \mid y=1)}{P(x_1)} = 2P(y=1 \mid x_1 = x)$$

Hence $P(y=1 \mid x_1 = x_1, x_2 = x_2) > P(y=1 \mid x_1 = x_1)$

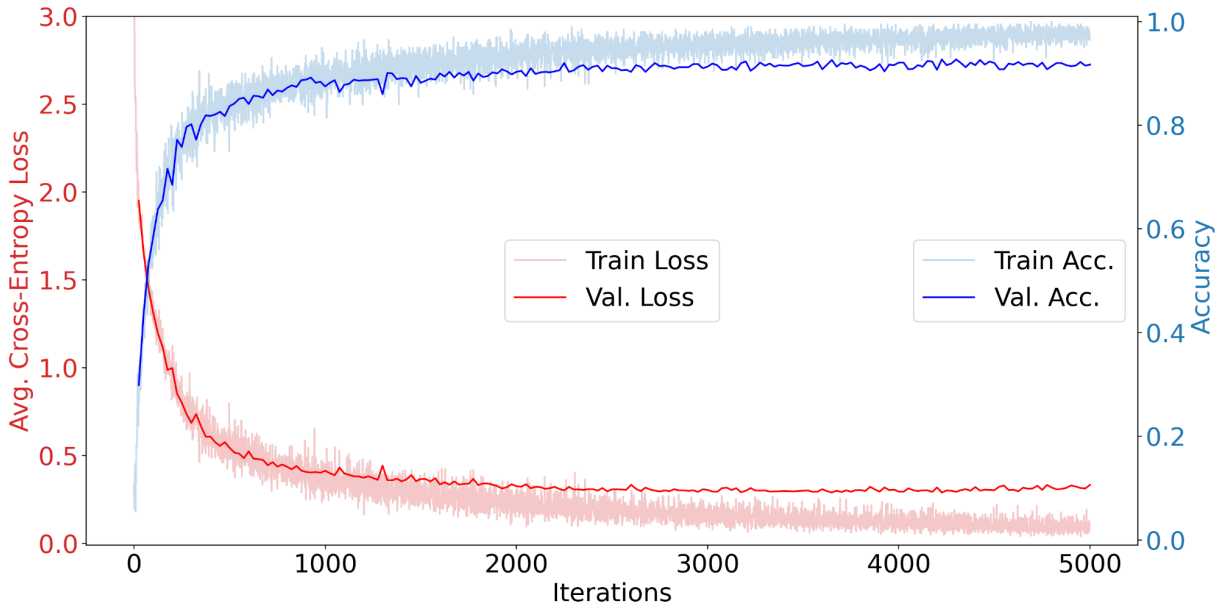
Q3.

```
# dL/dx = (dL/d_out) * (d_out/d_in)
# where: d_out/d_in = w.T
# From Eq. 24
grad_input = np.dot(grad, np.transpose(self.weights))

# dL/dW = [(dL/dW).T * (input)].T
# dL/dB = dL/d_out + 0
self.grad_weights = np.transpose(np.dot(np.transpose(grad), self.input))
self.grad_bias = np.sum(grad, axis = 0)

return grad_input
```

Q4.

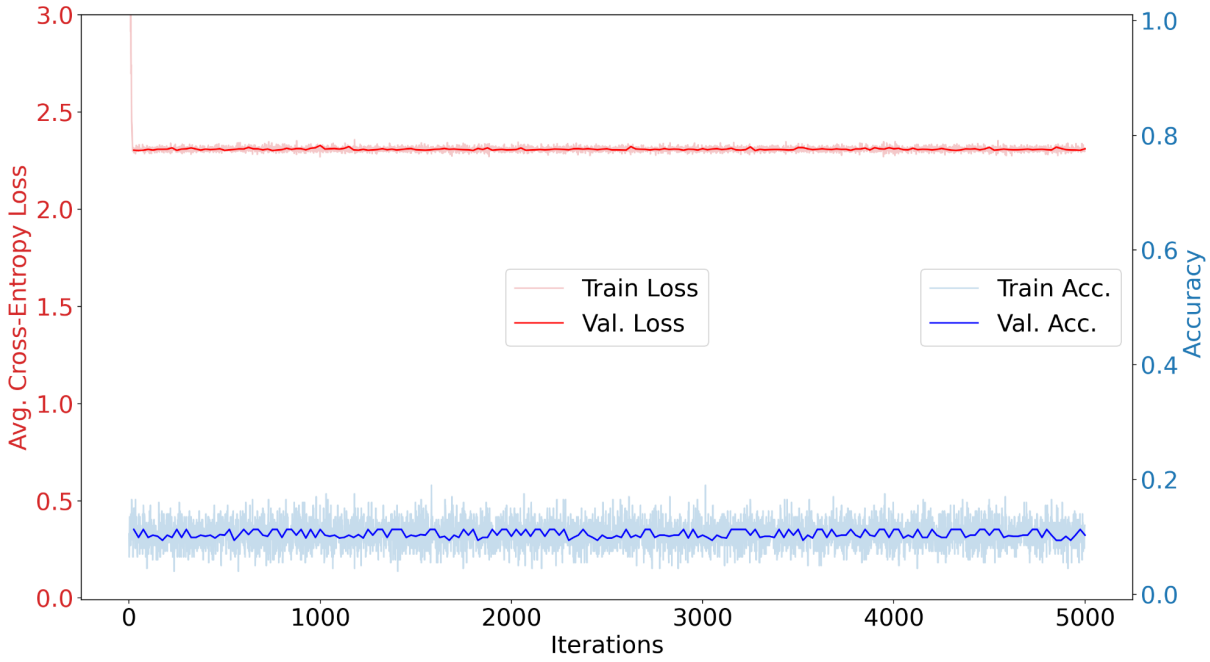
1. Step size of 0.0001 (leave default values for other hyperparameters)

```
# GLOBAL PARAMETERS FOR STOCHASTIC GRADIENT DESCENT
np.random.seed(102)
step_size = 0.0001
batch_size = 200
max_epochs = 200
```



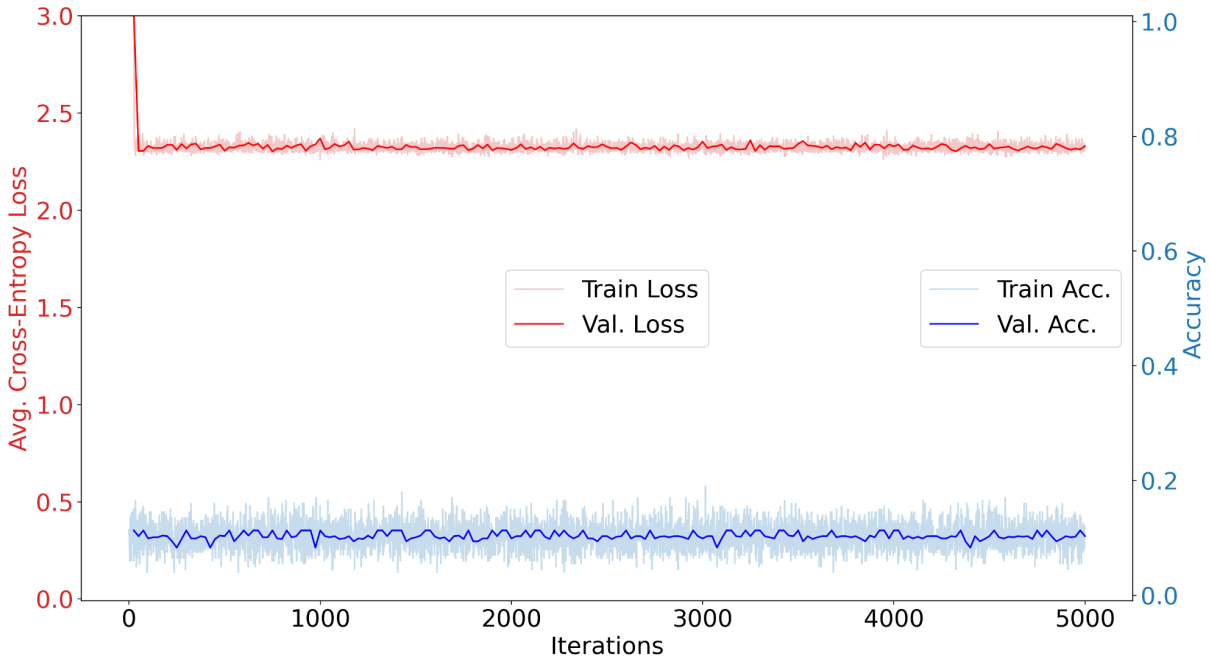
2. Step size of 5 (leave default values for other hyperparameters)

```
# GLOBAL PARAMETERS FOR STOCHASTIC GRADIENT DESCENT
np.random.seed(102)
step_size = 5
batch_size = 200
max_epochs = 200
```



3. Step size of 10 (leave default values for other hyperparameters)

```
# GLOBAL PARAMETERS FOR STOCHASTIC GRADIENT DESCENT
np.random.seed(102)
step_size = 10
batch_size = 200
max_epochs = 200
```



a) Compare and contrast the learning curves with your curve using the default parameters. What do you observe in terms of smoothness, shape, and what performance they reach?

The step size is the slope of the graph that increases rate when calculating step by step.
Such that when step size gets smaller (0.0001) the graph shows smoothly and curve shape on the plot. However, the performance of the accuracy is not getting better than default step size (0.01)
When the step size gets bigger (5 or 10) the slope changes rapidly, so that the value and train graph seem to parallel each other (Actual graph is not parallel). Because the loss values stay in high statements which means that loss rate is unstable.Also, the accuracy of the program is not good when step size is bigger than default value.
As a result, the default step size (0.01) is the best step size (learning rate) on this program)

b) For (a), what would you expect to happen if the max epochs were increased?

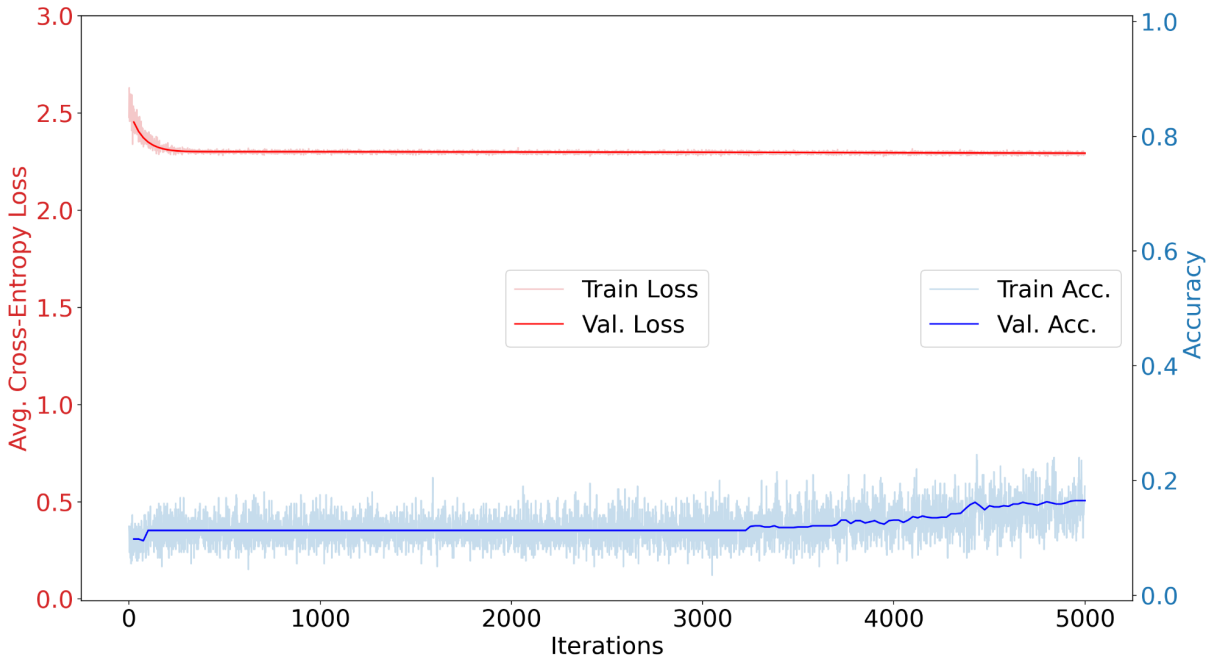
When increasing epochs the training accuracy rate is increasing.

Q5.
I ReLU's and Vanishing Gradients [3pts]. Modify the hyperparameters to run the following experiments:

1. 5-layer with Sigmoid Activation (leave default values for other hyperparameters)

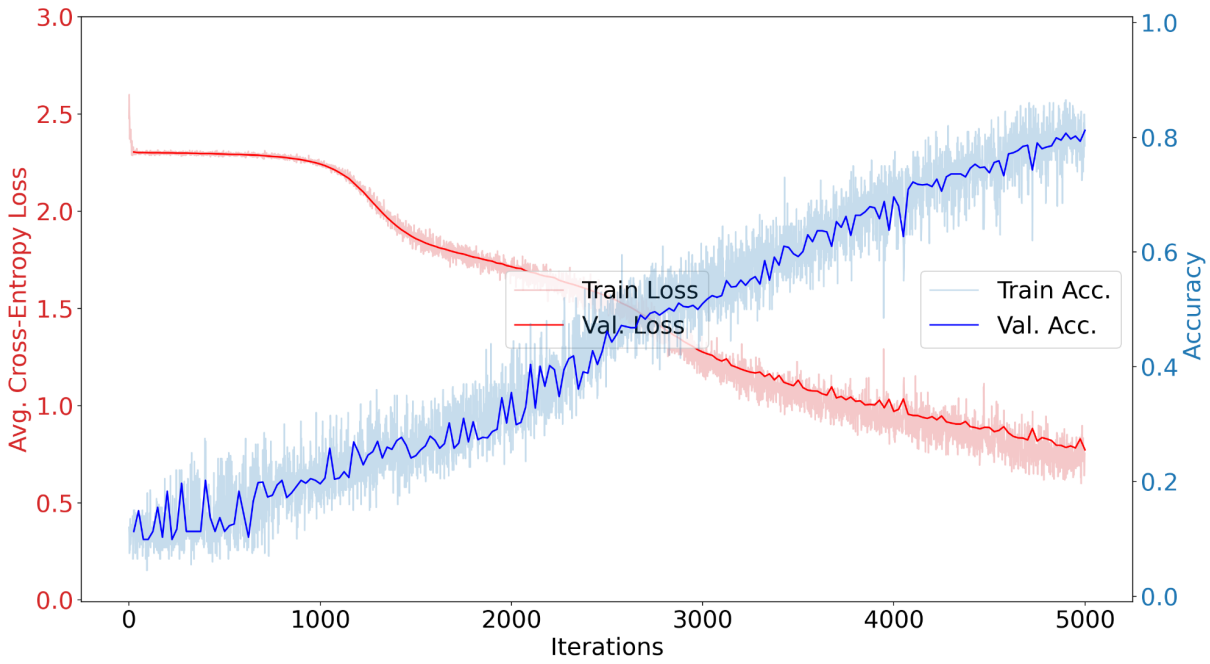
```
activation = "ReLU" if False else "Sigmoid"
```

```
# GLOBAL PARAMETERS FOR STOCHASTIC GRADIENT DESCENT
np.random.seed(102)
step_size = 0.01
batch_size = 200
max_epochs = 200
```



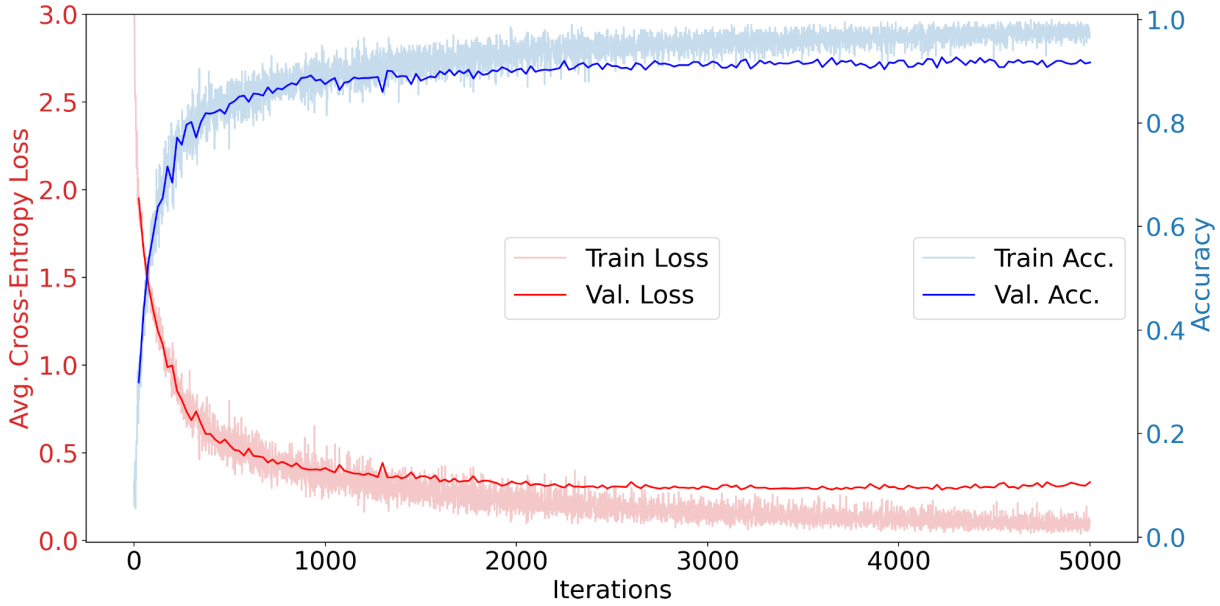
2. 5-layer with Sigmoid Activation with 0.1 step size (leave default values for other hyperparameters)

```
# GLOBAL PARAMETERS FOR STOCHASTIC GRADIENT DESCENT
np.random.seed(102)
step_size = 0.1
batch_size = 200
max_epochs = 200
```



3. 5-layer with ReLU Activation (leave default values for other hyperparameters)

```
activation = "ReLU" if True else "Sigmoid"
```



Include these plot in your report and answer the following questions:

a) Compare and contrast the learning curves you observe and the curve for the default parameters in terms of smoothness, shape, and what performance they reach. Do you notice any differences in the relationship between the train and validation curves in each plot?

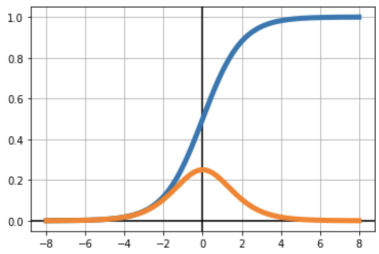
When the Sigmoid function is activated in the activation function, the shape (1) graph shows that each line (train and validation curves) is not crossed with each other when parameters are default values. However, in (2) the lines (train and validation loss, accuracy curves) are crossed each other. The performance of the program was simulation graph (2) is more accurate than simulation graph(1). The slope of the graph on the (3) each line is crossed and the shape shows stability and the performance of the accuracy rate shows the highest rate.

b) If you observed increasing the learning rate in (2) improves over (1), why might that be?

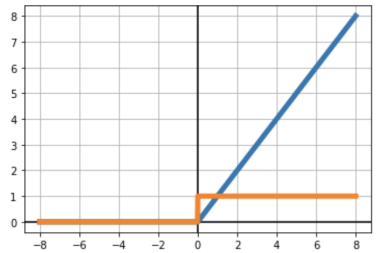
Gradient weight's values and gradient bias's values are very small, so the changes of the loss value are not very affected. Therefore, as the step size increases, the next step loss value has a very small difference between the current loss value. As a result, when derivative of the activation function on the back propagation, it will affect the gradient weight and gradient bias by the activation function's status. Sigmoid functions exist between 0 and 1, when the derivative results exist on very small ranges that make gradient weight and gradient bias getting smaller.

c) If (3) outperformed (1), why might that be? Consider the derivative of the sigmoid and ReLU functions.

Sigmoid functions exist between 0 and 1, so the input values are very small.



However, the ReLU function gets the input directly in output, if input is positive value and when derivative of input value during the back propagation. The result is not getting very smaller. Therefore, the ReLU function can make enough changes of the loss value with small step sizes.



Reference: Slide 8 in NueralNetworks

Q6. Using the default hyperparameters, set the random seed to 5 different values and report the *validation accuracies* you observe after training. What impact does this randomness have on the certainty of your conclusions in the previous questions?

10

```
# GLOBAL PARAMETERS
np.random.seed(10)
step_size = 0.01
batch_size = 200
max_epochs = 200
```

```
2021-05-16 17:07:15 INFO     [Epoch 185]   Loss:    0.1001     Train Acc:     97.34%      Val Acc:      91.9%
2021-05-16 17:07:15 INFO     [Epoch 186]   Loss:    0.1007     Train Acc:     97.54%      Val Acc:      91.9%
2021-05-16 17:07:15 INFO     [Epoch 187]   Loss:   0.09827     Train Acc:      97.4%      Val Acc:      92.7%
2021-05-16 17:07:15 INFO     [Epoch 188]   Loss:    0.1005     Train Acc:     97.22%      Val Acc:      92.3%
2021-05-16 17:07:15 INFO     [Epoch 189]   Loss:    0.1041     Train Acc:     97.14%      Val Acc:      92.0%
2021-05-16 17:07:15 INFO     [Epoch 190]   Loss:   0.09931     Train Acc:     97.14%      Val Acc:      91.8%
2021-05-16 17:07:15 INFO     [Epoch 191]   Loss:   0.09561     Train Acc:     97.42%      Val Acc:      91.2%
2021-05-16 17:07:15 INFO     [Epoch 192]   Loss:   0.09408     Train Acc:     97.66%      Val Acc:      92.4%
2021-05-16 17:07:15 INFO     [Epoch 193]   Loss:    0.0918     Train Acc:      97.7%      Val Acc:      91.7%
2021-05-16 17:07:15 INFO     [Epoch 194]   Loss:    0.0942     Train Acc:     97.32%      Val Acc:      91.7%
2021-05-16 17:07:15 INFO     [Epoch 195]   Loss:   0.09407     Train Acc:     97.44%      Val Acc:      92.6%
2021-05-16 17:07:15 INFO     [Epoch 196]   Loss:   0.09419     Train Acc:     97.52%      Val Acc:      92.1%
2021-05-16 17:07:15 INFO     [Epoch 197]   Loss:    0.1015     Train Acc:     97.18%      Val Acc:      92.2%
2021-05-16 17:07:15 INFO     [Epoch 198]   Loss:   0.09075     Train Acc:      97.7%      Val Acc:      92.5%
2021-05-16 17:07:15 INFO     [Epoch 199]   Loss:   0.09429     Train Acc:     97.38%      Val Acc:      92.1%
```

50

```
# GLOBAL PARAMETERS
np.random.seed(50)
step_size = 0.01
batch_size = 200
max_epochs = 200
```

```
2021-05-16 17:08:04 INFO     [Epoch 185]   Loss:    0.1333     Train Acc:     96.14%      Val Acc:      91.1%
2021-05-16 17:08:05 INFO     [Epoch 186]   Loss:    0.1296     Train Acc:      96.6%      Val Acc:      91.2%
2021-05-16 17:08:05 INFO     [Epoch 187]   Loss:    0.1296     Train Acc:      96.3%      Val Acc:      90.9%
2021-05-16 17:08:05 INFO     [Epoch 188]   Loss:    0.1275     Train Acc:     96.52%      Val Acc:      91.5%
2021-05-16 17:08:05 INFO     [Epoch 189]   Loss:     0.127     Train Acc:     96.58%      Val Acc:      90.8%
2021-05-16 17:08:05 INFO     [Epoch 190]   Loss:    0.1317     Train Acc:     96.48%      Val Acc:      90.8%
2021-05-16 17:08:05 INFO     [Epoch 191]   Loss:    0.1343     Train Acc:     96.46%      Val Acc:      90.5%
2021-05-16 17:08:05 INFO     [Epoch 192]   Loss:    0.1311     Train Acc:     96.22%      Val Acc:      91.4%
2021-05-16 17:08:05 INFO     [Epoch 193]   Loss:    0.1219     Train Acc:     96.68%      Val Acc:      91.4%
2021-05-16 17:08:05 INFO     [Epoch 194]   Loss:    0.1193     Train Acc:     96.84%      Val Acc:      90.6%
2021-05-16 17:08:05 INFO     [Epoch 195]   Loss:    0.1255     Train Acc:     96.46%      Val Acc:      90.8%
2021-05-16 17:08:05 INFO     [Epoch 196]   Loss:    0.1246     Train Acc:      96.6%      Val Acc:      90.9%
2021-05-16 17:08:05 INFO     [Epoch 197]   Loss:    0.1226     Train Acc:     96.76%      Val Acc:      90.6%
2021-05-16 17:08:05 INFO     [Epoch 198]   Loss:    0.1203     Train Acc:     96.88%      Val Acc:      90.7%
2021-05-16 17:08:05 INFO     [Epoch 199]   Loss:      0.12     Train Acc:     96.76%      Val Acc:      90.5%
```

100

```
# GLOBAL PARAMETERS F
np.random.seed(100)
step_size = 0.01
batch_size = 200
max_epochs = 200
```

```
2021-05-16 17:08:47 INFO     [Epoch 184]   Loss:     0.1111     Train Acc:     96.96%     Val Acc:     91.8%
2021-05-16 17:08:47 INFO     [Epoch 185]   Loss:     0.1172     Train Acc:     96.86%     Val Acc:     91.8%
2021-05-16 17:08:47 INFO     [Epoch 186]   Loss:     0.1114     Train Acc:     97.06%     Val Acc:     92.0%
2021-05-16 17:08:47 INFO     [Epoch 187]   Loss:     0.1177     Train Acc:     96.66%     Val Acc:     91.5%
2021-05-16 17:08:47 INFO     [Epoch 188]   Loss:     0.1106     Train Acc:     97.04%     Val Acc:     92.4%
2021-05-16 17:08:47 INFO     [Epoch 189]   Loss:     0.1138     Train Acc:     97.06%     Val Acc:     91.3%
2021-05-16 17:08:47 INFO     [Epoch 190]   Loss:     0.1113     Train Acc:     97.04%     Val Acc:     91.2%
2021-05-16 17:08:47 INFO     [Epoch 191]   Loss:     0.1068     Train Acc:      97.1%     Val Acc:     91.4%
2021-05-16 17:08:47 INFO     [Epoch 192]   Loss:     0.1047     Train Acc:      97.2%     Val Acc:     91.1%
2021-05-16 17:08:47 INFO     [Epoch 193]   Loss:     0.1096     Train Acc:     97.22%     Val Acc:     90.5%
2021-05-16 17:08:47 INFO     [Epoch 194]   Loss:     0.1194     Train Acc:     96.68%     Val Acc:     91.6%
2021-05-16 17:08:47 INFO     [Epoch 195]   Loss:     0.1054     Train Acc:     97.18%     Val Acc:     91.5%
2021-05-16 17:08:47 INFO     [Epoch 196]   Loss:     0.1136     Train Acc:     96.74%     Val Acc:     92.2%
2021-05-16 17:08:47 INFO     [Epoch 197]   Loss:     0.1059     Train Acc:     97.24%     Val Acc:     90.7%
2021-05-16 17:08:47 INFO     [Epoch 198]   Loss:     0.1043     Train Acc:     97.14%     Val Acc:     92.1%
2021-05-16 17:08:47 INFO     [Epoch 199]   Loss:     0.1038     Train Acc:     97.48%     Val Acc:     92.0%
```

200

```
# GLOBAL PARAMETERS F
np.random.seed(200)
step_size = 0.01
batch_size = 200
max_epochs = 200
```

```
2021-05-16 17:09:43 INFO     [Epoch 185]   Loss:     0.1345     Train Acc:      96.4%     Val Acc:     91.1%
2021-05-16 17:09:43 INFO     [Epoch 186]   Loss:      0.137     Train Acc:     96.42%     Val Acc:     91.8%
2021-05-16 17:09:43 INFO     [Epoch 187]   Loss:     0.1349     Train Acc:     96.42%     Val Acc:     91.7%
2021-05-16 17:09:43 INFO     [Epoch 188]   Loss:     0.1346     Train Acc:     96.28%     Val Acc:     91.6%
2021-05-16 17:09:43 INFO     [Epoch 189]   Loss:     0.1328     Train Acc:     96.24%     Val Acc:     91.6%
2021-05-16 17:09:43 INFO     [Epoch 190]   Loss:      0.132     Train Acc:     96.48%     Val Acc:     91.7%
2021-05-16 17:09:43 INFO     [Epoch 191]   Loss:     0.1334     Train Acc:     96.26%     Val Acc:     92.0%
2021-05-16 17:09:43 INFO     [Epoch 192]   Loss:     0.1334     Train Acc:     96.38%     Val Acc:     91.6%
2021-05-16 17:09:43 INFO     [Epoch 193]   Loss:     0.1288     Train Acc:     96.52%     Val Acc:     91.9%
2021-05-16 17:09:43 INFO     [Epoch 194]   Loss:     0.1307     Train Acc:     96.42%     Val Acc:     91.9%
2021-05-16 17:09:43 INFO     [Epoch 195]   Loss:     0.1271     Train Acc:     96.58%     Val Acc:     91.8%
2021-05-16 17:09:43 INFO     [Epoch 196]   Loss:     0.1286     Train Acc:     96.54%     Val Acc:     91.6%
2021-05-16 17:09:43 INFO     [Epoch 197]   Loss:     0.1319     Train Acc:     96.62%     Val Acc:     91.8%
2021-05-16 17:09:43 INFO     [Epoch 198]   Loss:     0.1242     Train Acc:     96.82%     Val Acc:     91.6%
2021-05-16 17:09:43 INFO     [Epoch 199]   Loss:     0.1245     Train Acc:     96.78%     Val Acc:     91.9%
```

1000

```
# GLOBAL PARAMETERS F
np.random.seed(1000)
step_size = 0.01
batch_size = 200
max_epochs = 200
```

```
2021-05-16 17:10:38 INFO     [Epoch 184]   Loss:     0.123     Train Acc:     98.82%      Val Acc:      92.5%
2021-05-16 17:10:39 INFO     [Epoch 185]   Loss:     0.121     Train Acc:     96.94%      Val Acc:      92.2%
2021-05-16 17:10:39 INFO     [Epoch 186]   Loss:    0.1204     Train Acc:     96.66%      Val Acc:      91.2%
2021-05-16 17:10:39 INFO     [Epoch 187]   Loss:    0.1212     Train Acc:     96.66%      Val Acc:      91.9%
2021-05-16 17:10:39 INFO     [Epoch 188]   Loss:    0.1202     Train Acc:     96.74%      Val Acc:      92.3%
2021-05-16 17:10:39 INFO     [Epoch 189]   Loss:    0.1217     Train Acc:     96.56%      Val Acc:      92.4%
2021-05-16 17:10:39 INFO     [Epoch 190]   Loss:    0.1225     Train Acc:     96.82%      Val Acc:      91.4%
2021-05-16 17:10:39 INFO     [Epoch 191]   Loss:    0.1217     Train Acc:     96.62%      Val Acc:      90.9%
2021-05-16 17:10:39 INFO     [Epoch 192]   Loss:     0.118     Train Acc:     97.06%      Val Acc:      91.9%
2021-05-16 17:10:39 INFO     [Epoch 193]   Loss:    0.1206     Train Acc:     96.86%      Val Acc:      91.2%
2021-05-16 17:10:39 INFO     [Epoch 194]   Loss:    0.1199     Train Acc:     96.66%      Val Acc:      91.8%
2021-05-16 17:10:39 INFO     [Epoch 195]   Loss:    0.1188     Train Acc:     96.96%      Val Acc:      92.3%
2021-05-16 17:10:39 INFO     [Epoch 196]   Loss:    0.1156     Train Acc:     96.84%      Val Acc:      91.3%
2021-05-16 17:10:39 INFO     [Epoch 197]   Loss:     0.114     Train Acc:      96.8%      Val Acc:      92.5%
2021-05-16 17:10:39 INFO     [Epoch 198]   Loss:    0.1129     Train Acc:     96.94%      Val Acc:      91.5%
2021-05-16 17:10:39 INFO     [Epoch 199]   Loss:    0.1097     Train Acc:     97.18%      Val Acc:      92.1%
```

- The result does not have big changes by randomness (random seed)
- So my conclusion from the previous question is certainly.

As a result, the conclusion from the previous question does not have a big impact by randomness. Such that my conclusion from the previous question is certainly.

Q8. Kaggle submit

See the kaggle submit

3. Debriefing (required in your report)

1. Approximately how many hours did you spend on this assignment? 23 hours?

2. Would you rate it as easy, moderate, or difficult? Difficult

3. Did you work on it mostly alone or did you discuss the problems with others? Alone

4. How deeply do you feel you understand the material it covers (0%–100%)? 80%

5. Any other comments? Hard to understand the Gradient Descent part of the code, tracking was a bit hard for me.