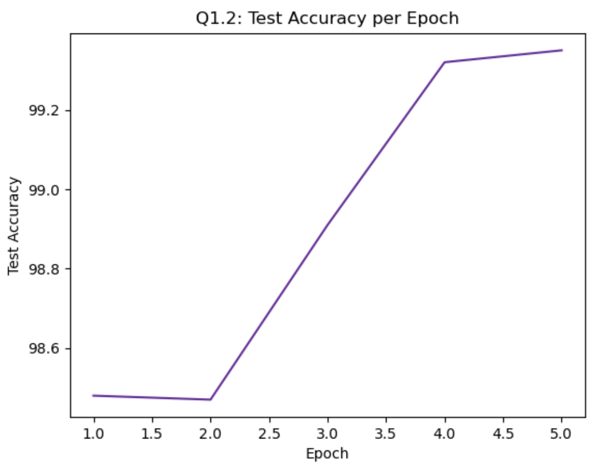
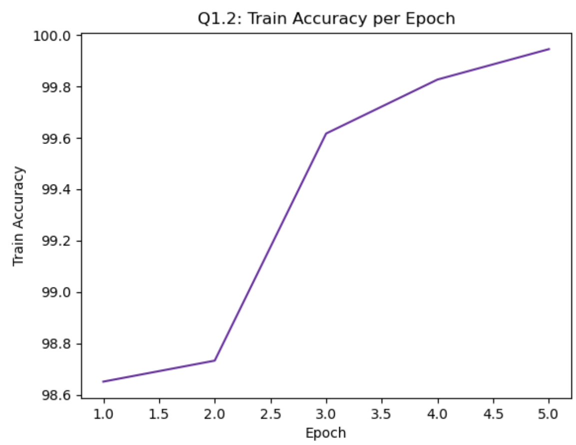
CS680 – Assignment 3

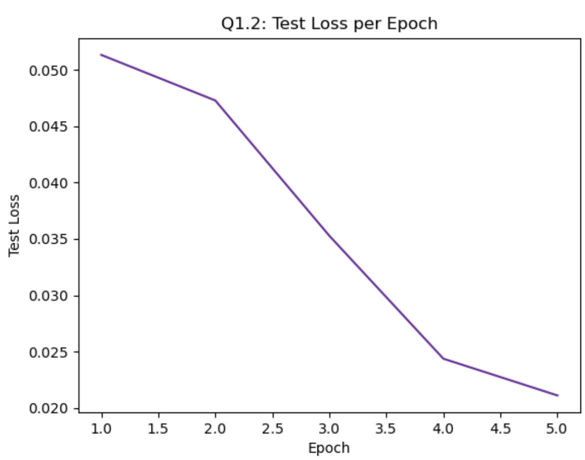
Ali ElSaid (20745892)

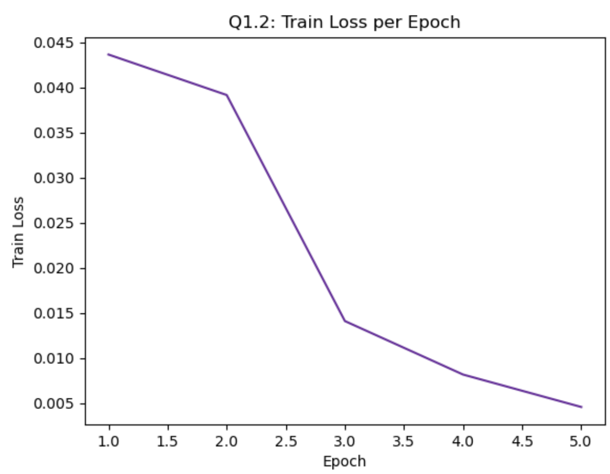
Q1.1: (implementation in vgg11.py)

Q1.2:



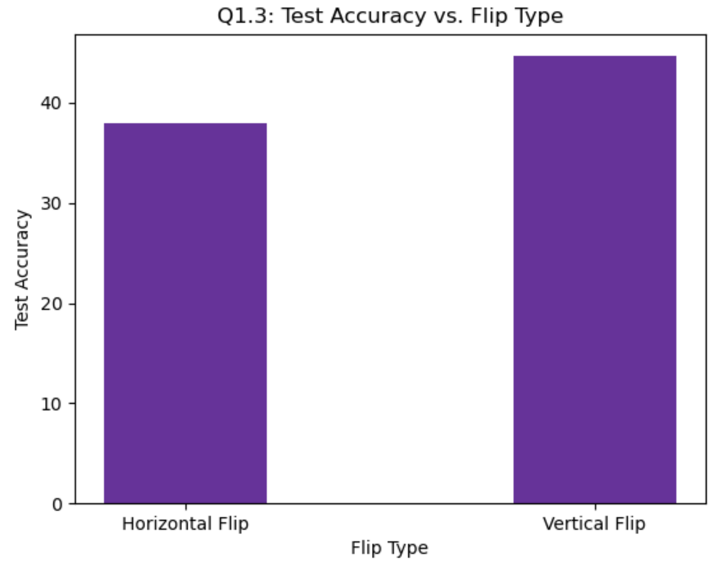






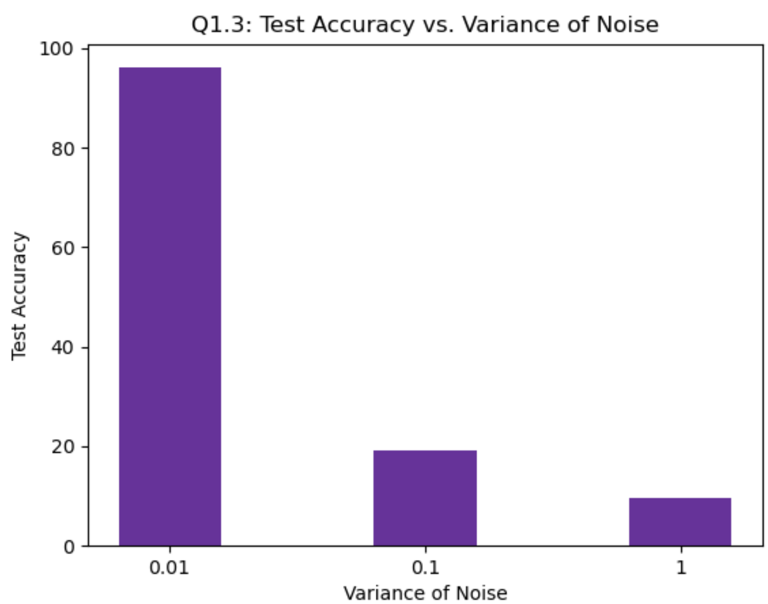
Q1.3:

e)



The flips drastically decreased the test accuracy, which is expected since the model has been trained to detect only upright images.

f)

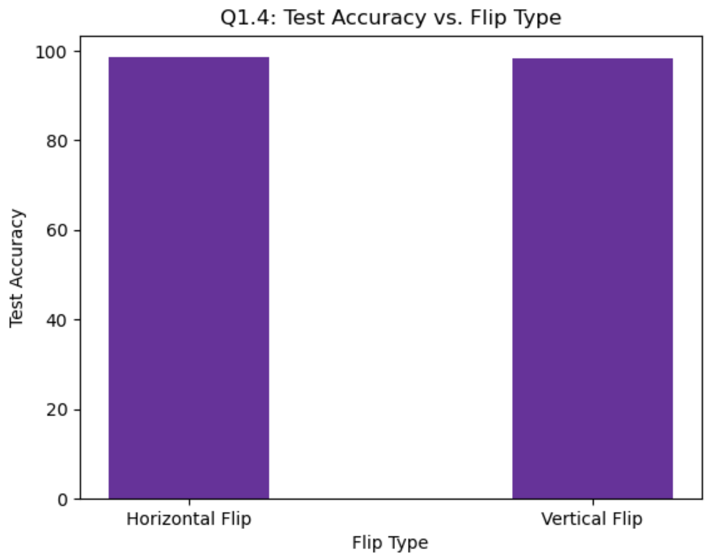


The gaussian noise affected the test accuracy proportional to how aggressive it was (variance) which is expected since the higher the variance the more the images deviate from what the model trained on, so it's expected that it performs worse on completely unseen data.

Q1.4:

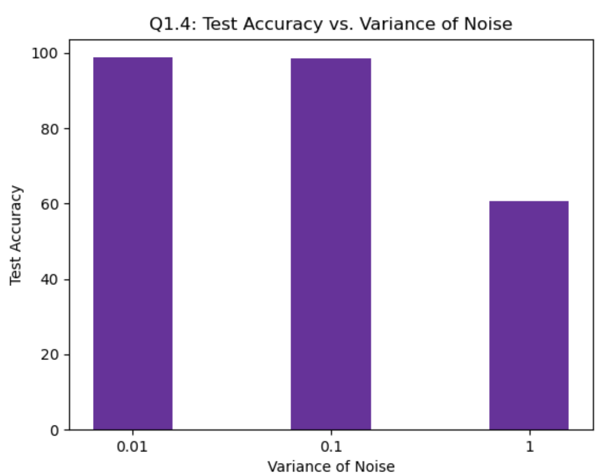
I quadrupled the training data by augmenting using 3 different methods: (horizontal flip, vertical flip, gaussian noise with 0.01 variance). I applied each transformation to the data and concatenated the transformed version of the base training data to my existing data. So, in the end I have 4 sub datasets (base, horizontally flipped, vertically flipped, added gaussian noise with 0.01 variance) and train on all of them shuffled.

e)



We can see that after the augmentation, the performance of the model on the flipped images has drastically gone up. Which is expected since now the model has trained on similar images.

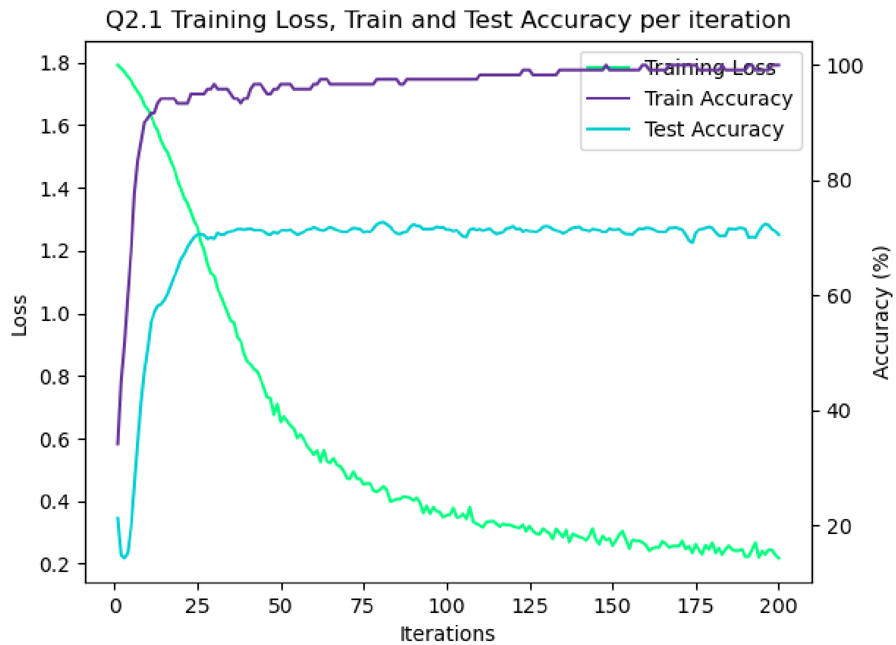
f)

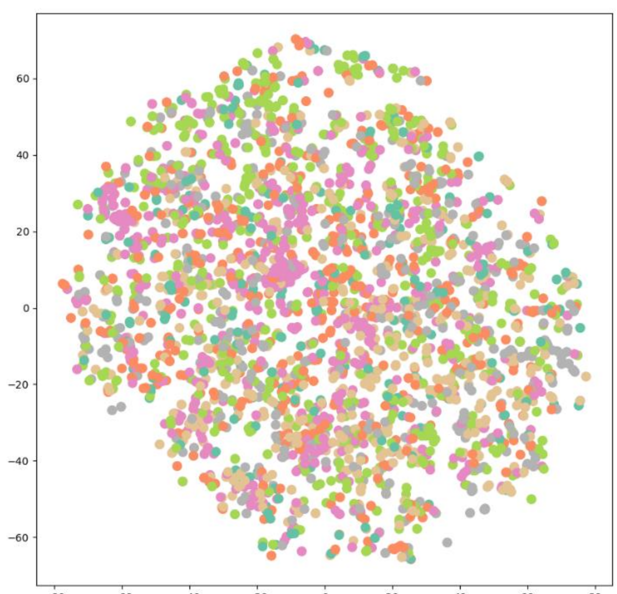


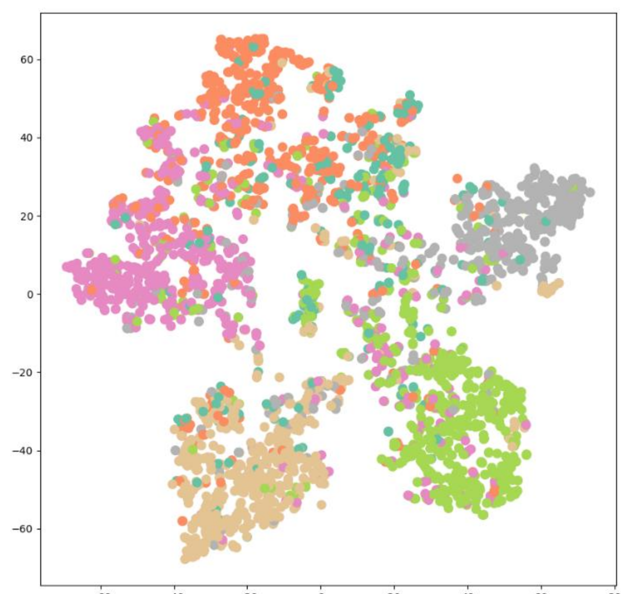
We can see that after the augmentation, the performance of the model on the noisy test set has also increased across the board for the same reason. The model now sees similar images in training. In specific, the accuracy for the 0.1 variance has gone almost to 100% since that is the one trained on. And training on that also helped the model detect some of the more extreme cases that have 1 variance, therefore their performance increased too.

Q2.1:

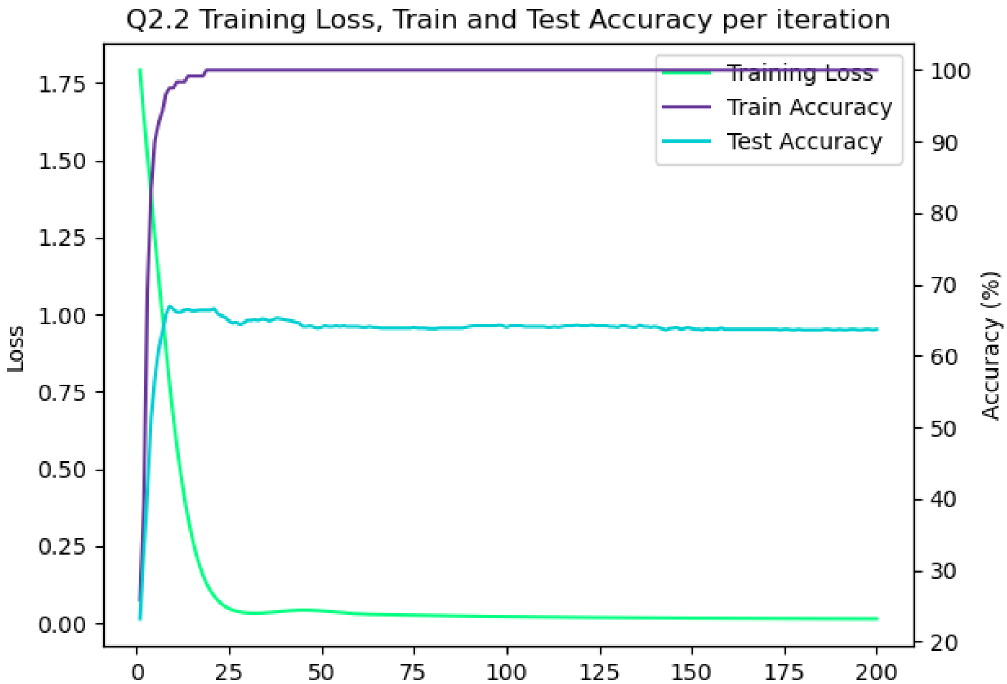
Note: In the areas where we should plot the classification error, I plotted the classification accuracy (1 – classification error) so that the curves of the loss and accuracy be visually distinct, i.e. going in different directions.

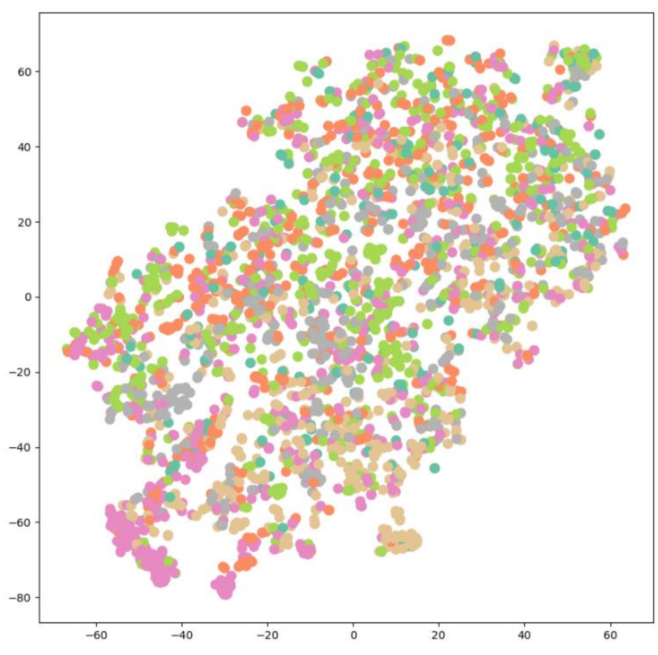


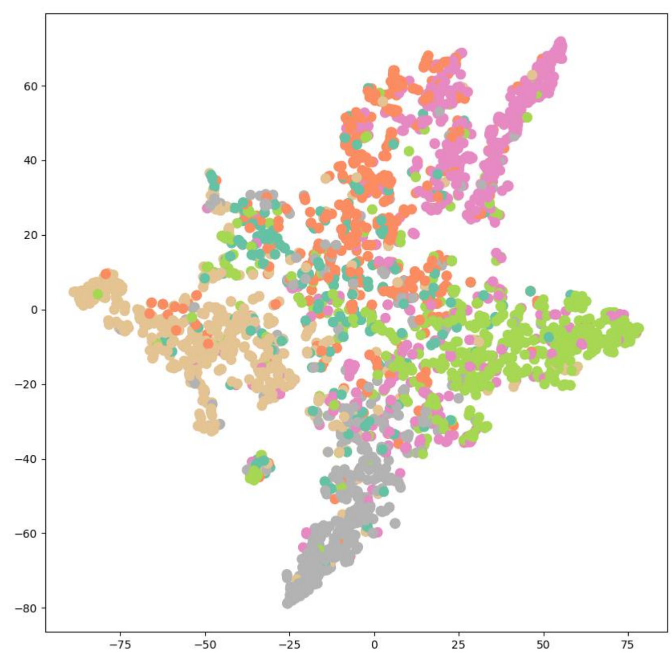




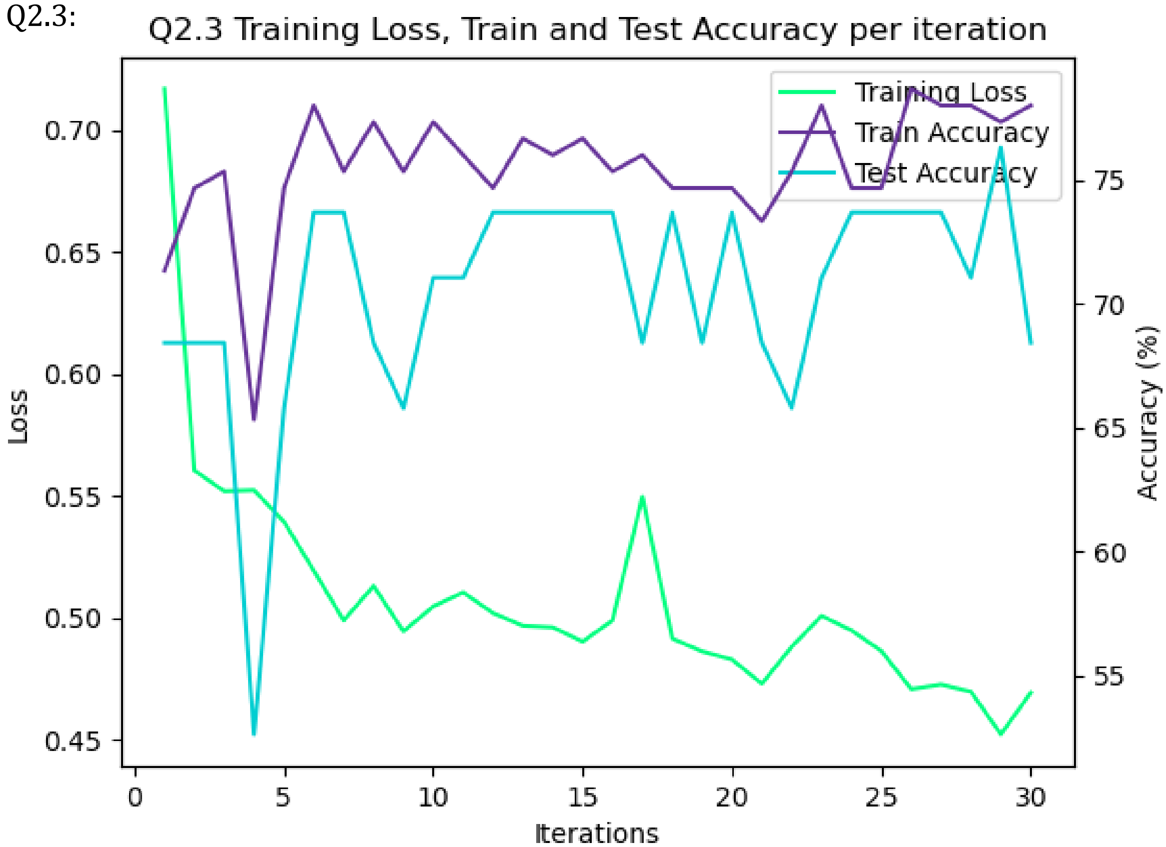
Q2.2:


Q2.2 Training Loss, Train and Test Accuracy per iteration
Training Loss
Train Accuracy
Test Accuracy
Loss
Accuracy (%)



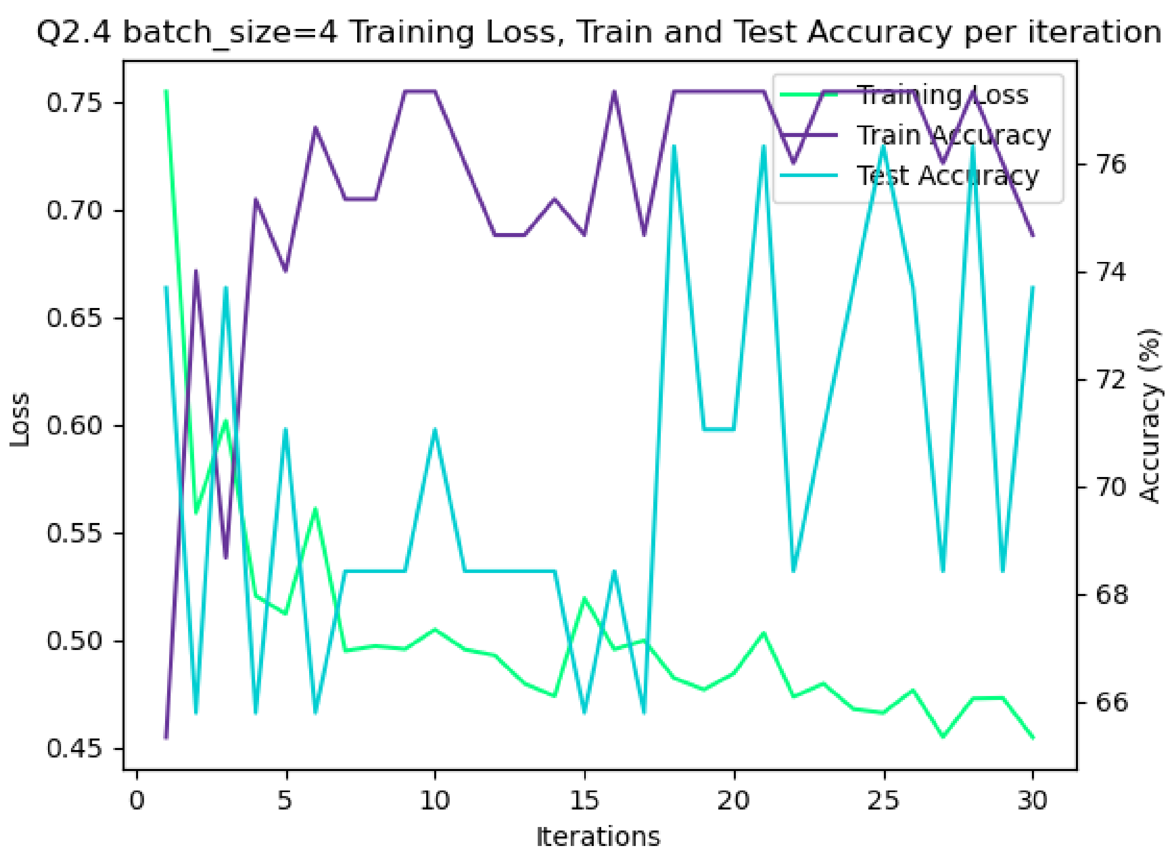


Q2.3:

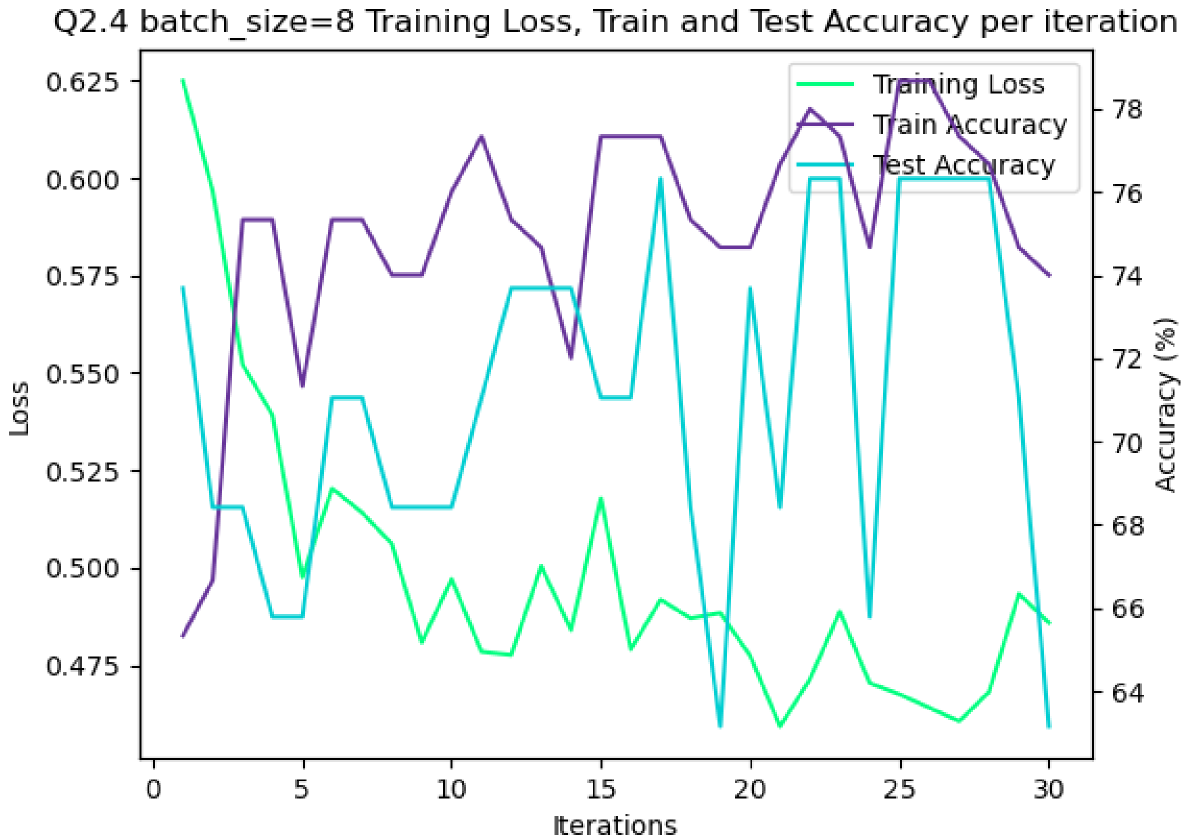


Q2.3 Training Loss, Train and Test Accuracy per iteration
Training Loss
Train Accuracy
Test Accuracy
Loss
Accuracy (%)
Iterations
0.45
0.50
0.55
0.60
0.65
0.70
55
60
65
70
75
0
5
10
15
20
25
30

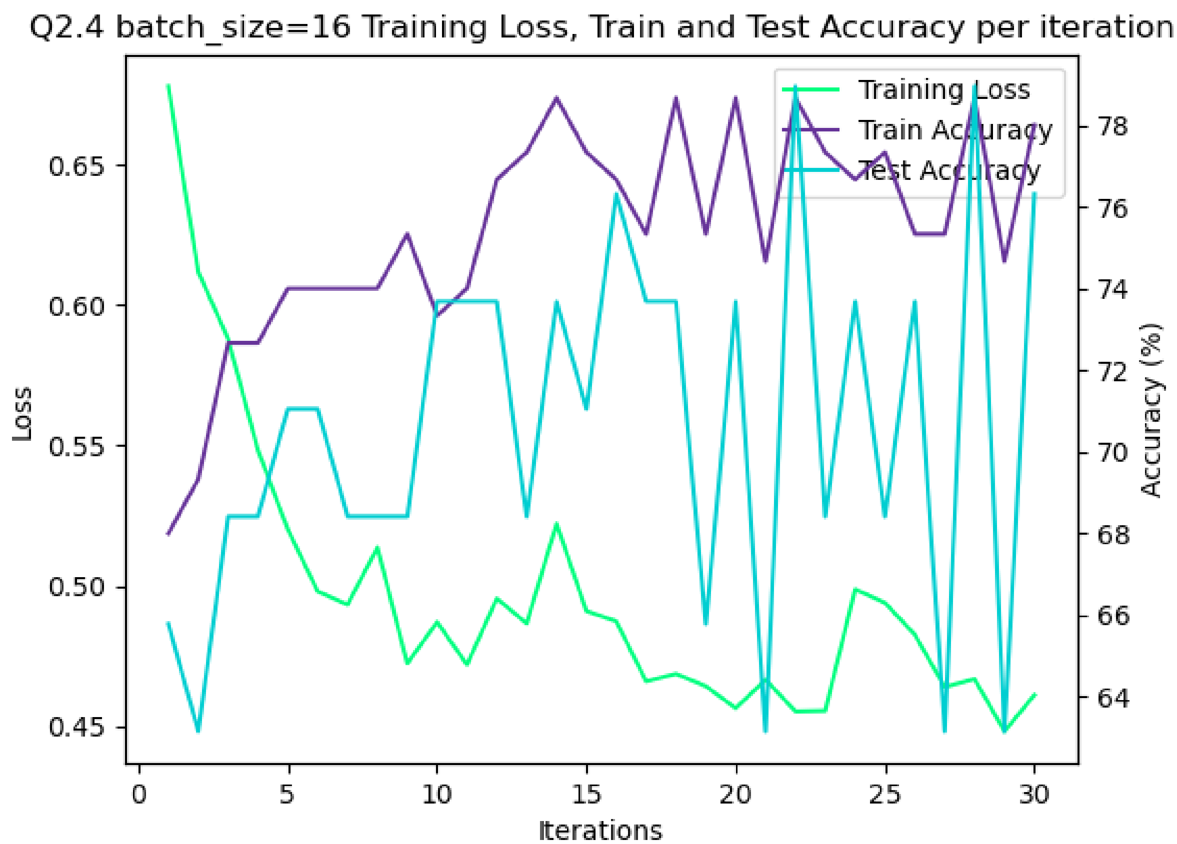
Q2.4:



Q2.4 batch_size=4 Training Loss, Train and Test Accuracy per iteration
Training Loss
Train Accuracy
Test Accuracy
Loss
Accuracy (%)
Iterations
0.45
0.50
0.55
0.60
0.65
0.70
0.75
66
68
70
72
74
76
0
5
10
15
20
25
30



Q2.4 batch_size=8 Training Loss, Train and Test Accuracy per iteration
Training Loss
Train Accuracy
Test Accuracy
Loss
Accuracy (%)
Iterations
0.625
0.600
0.575
0.550
0.525
0.500
0.475
78
76
74
72
70
68
66
64
0
5
10
15
20
25
30



Q2.4 batch_size=16 Training Loss, Train and Test Accuracy per iteration
Training Loss
Train Accuracy
Test Accuracy
Loss
Accuracy (%)
Iterations
0.65
0.60
0.55
0.50
0.45
78
76
74
72
70
68
66
64
0
5
10
15
20
25
30

Q3.1:

Q3.1:

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} E\left[(y_i - w^T \tilde{x}_i)^2\right]$$

- $w^T \tilde{x}_i = w^T(x_i + \epsilon_i) = w^T x_i + \epsilon_i$

- Since $Var(x) = E(x^2) - E(x)^2 \rightarrow E(x^2) = Var(x) + E(x)^2$

So, $E\left[(y_i - w^T \tilde{x}_i)^2\right] = E\left[(y_i - w^T x_i - w^T \epsilon_i)^2\right]$

$$= Var\left[y_i - w^T x_i - w^T \epsilon_i\right] + E\left[y_i - w^T x_i - w^T \epsilon_i\right]^2$$

$$= \underbrace{var\left[y_i - w^T x_i\right]}_{\downarrow \text{ since } y_i, w^T, x_i \text{ are fixed}} + \underbrace{var(w^T \epsilon_i)}_{\downarrow} + \underbrace{\left(y_i - w^T x_i - w^T \underbrace{E[\epsilon_i]}_{0}\right)^2}_{\downarrow}$$

$$0 \quad + \quad w^T \lambda I w \quad + \quad (y_i - w^T x_i)^2$$

$$= (y_i - w^T x_i)^2 + \lambda \|w\|_2^2$$

since $\downarrow$

- $\underset{1\times d}{w^T} \underset{d\times 1}{w} = \langle w \cdot w \rangle = \sum_i w_i^2 = \|w\|_2^2$

$\therefore$ the problem becomes:

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} E\left[(y_i - w^T \tilde{x}_i)^2\right] = \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \|w\|_2^2$$

which is Ridge Regression

Q3.2:

Q3.2:

$$p\,\epsilon_{ji} \sim \text{Bernoulli}(p) \rightarrow \begin{cases} 1, & p \\ 0, & 1-p \end{cases}$$

$$\rightarrow \epsilon_{ji} \rightarrow \begin{cases} 1/p, & p \\ 0, & 1-p \end{cases}$$

- $$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} E\left[(y_i - w^T \tilde{x}_i)^2\right]$$

$$\tilde{x}_i = x_i \odot \epsilon_i$$

- $$E\left[(y_i - w^T \tilde{x}_i)^2\right] = E\left[(y_i - w^T(x_i \odot \epsilon_i))^2\right]$$

$$w^T(x_i \odot \epsilon_i) = \sum_{j=1}^{d} w_j x_{ji} \epsilon_{ji}$$

$$= \text{Var}\left(y_i - \sum_{j=1}^{d} w_j x_{ji} \epsilon_{ji}\right) + E\left[y_i - \sum_{j=1}^{d} w_j x_{ji} \epsilon_{ji}\right]^2$$

[since y_i's and x_i's are fixed]

$$= \text{Var}(y_i) + \text{Var}\left(\sum_{j=1}^{d} w_j x_{ji} \epsilon_{ji}\right) + \left(y_i - \sum_{j=1}^{d} w_j x_{ji} \underbrace{E[\epsilon_{ji}]}_{=1=\frac{1}{p}\cdot p}\right)^2$$

[since ϵ_{ji}'s are indep. for diff. j's]

$$= 0 + \sum_{j=1}^{d} w_j^2 x_{ji}^2 \underbrace{\text{Var}(\epsilon_{ji})}_{\lambda} + \left(y_i - \sum_{j=1}^{d} w_j x_{ji}\right)^2$$

$$= \lambda \sum_{j=1}^{d} w_j^2 x_{ji}^2 + \left(y_i - \sum_{j=1}^{d} w_j x_{ji}\right)^2$$

$$= \lambda \|w^T x_i\|_2^2 + (y_i - w^T x_i)^2$$

$\therefore$ The original problem becomes

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} E\left[(y_i - w^T \tilde{x}_i)^2\right] = \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \underbrace{\lambda \|w^T x_i\|_2^2}$$

regularization component that depends on x