Assignment 2

<u>Task 1</u>

a)

$$\frac{dC}{dw_{ji}} = \sum_k \left(\frac{dC}{d\hat{y}} \cdot \frac{d\hat{y}}{dz_k} \cdot \frac{dz_k}{da_j} \cdot \frac{da_j}{dz_j} \cdot \frac{dz_j}{dw_j} \right) = \underbrace{\sum_k \Big(\overbrace{\frac{dC}{d\hat{y}} \cdot \frac{d\hat{y}}{dz_k}}^{\delta_k} \cdot \frac{dz_k}{da_j} \Big) \cdot \frac{da_j}{dz_j}} \cdot \frac{dz_j}{dw_j} = \underline{\delta_j \cdot \frac{dz_j}{dw_j} = \delta_j \cdot x}$$

$$\delta_j = \delta_k \cdot \frac{dz_k}{da_j} \cdot \frac{da_j}{dz_j} = \delta_k \cdot \sum_k w_{kj} \cdot f_a'(z_j)$$

$$w_{ji} = w_{ji} - \alpha \frac{dC}{dw_{ji}} = w_{ji} - \alpha \delta_j x$$

b)

$$W^J = I \times J \qquad (1 \times I \cdot I \times J = \overset{\text{hidden}}{1 \times J})$$

$$X = 1 \times I$$

$$\delta^J = 1 \times J \qquad \left(\delta = \text{error} = \frac{dC}{dz_j}\right)$$

$$W^J = W^J - X^T \delta^J$$

$$W^k = J \times K \qquad (\overset{\text{input}}{1 \times J} \cdot J \times K = 1 \times K)$$

$$Z^k = 1 \times J \cdot J \times K = 1 \times J$$

$$\partial^k = 1 \times K$$

$$W^k = W^k - \alpha Z^{k^T} \cdot \partial^K$$

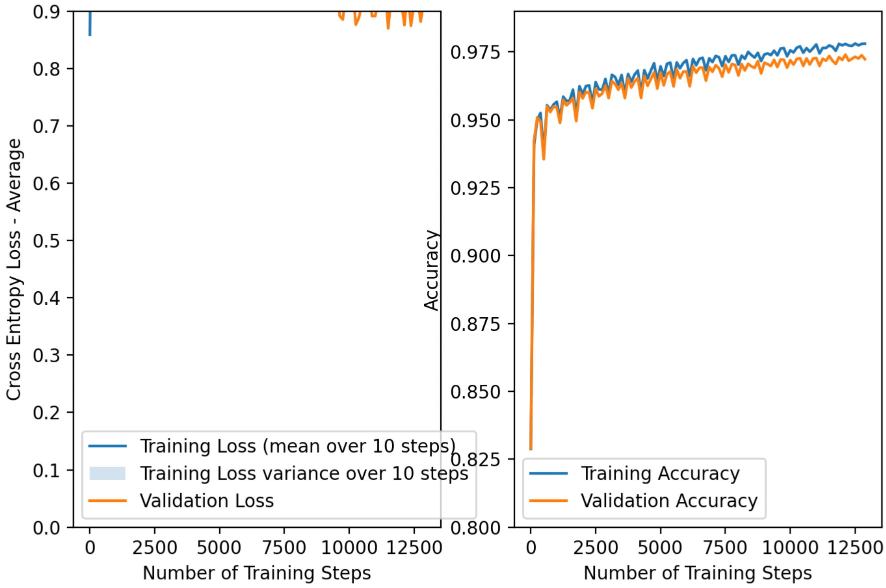
<u>Task 2</u>

a)

Mean: 33.55274553571429

Standard devation: 78.87550070784701

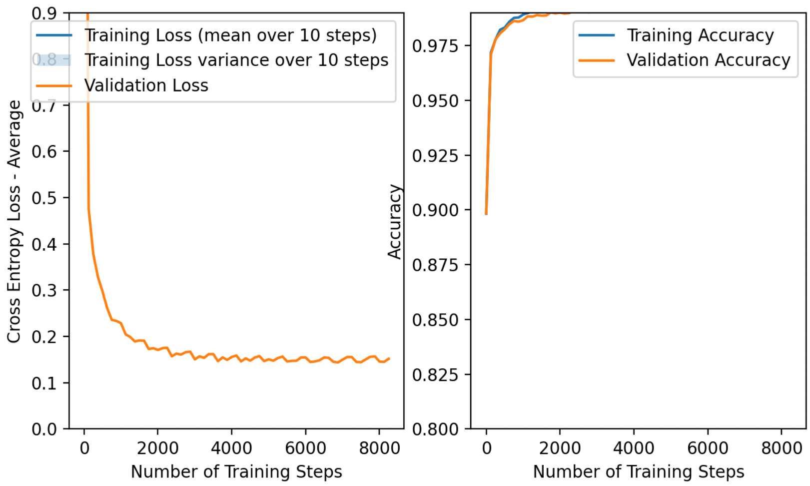
c)



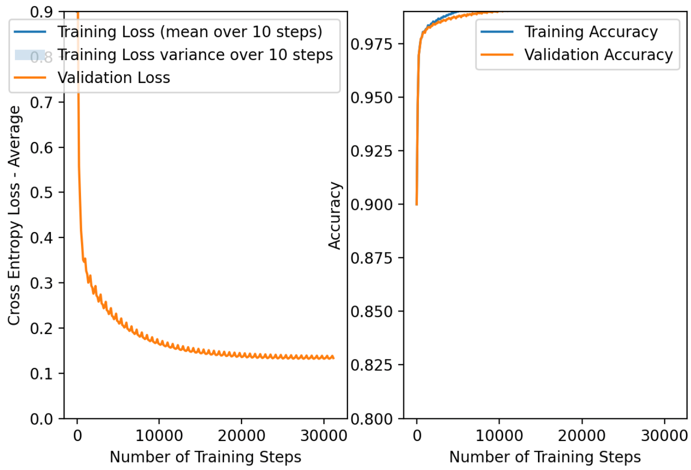
d)
There are 784 * 64 + 64 + 64 * 10 + 10 parameters in the network

<u>Task 3</u>

- use_improved_weight_init=True:

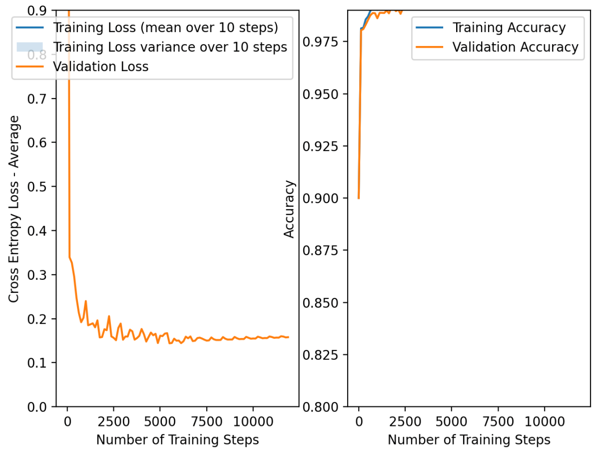


-
    - With relu it generalizes very fast

- 

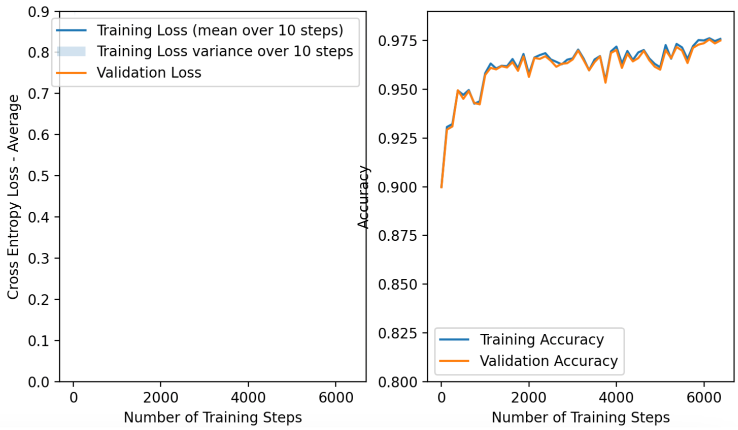
    - We can see that with the He weight initialization the model learns faster than with the random weight initialization, when relu is used

- Use_improved_weight_init=True, use_momentum=True, lr=0.02

- 

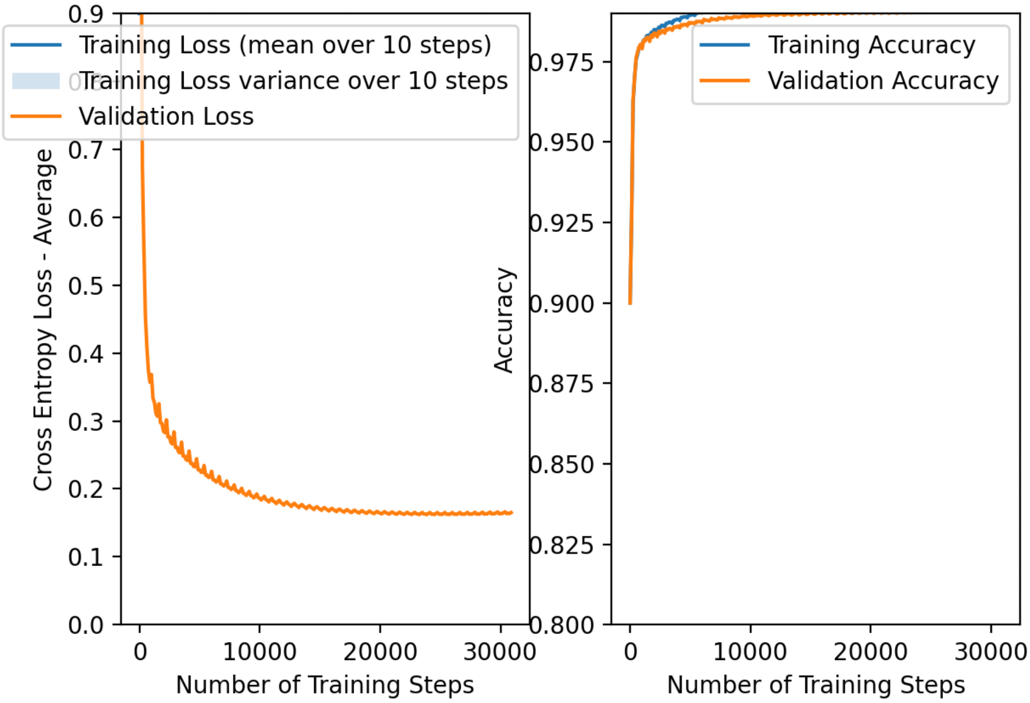
    - We see that with momentum the gradient steps in more drastic directions, which is expected as the gradients can become bigger if the previous gradient has the same direction of the current gradient, which can lead to what is in practice a increasing learning rate (compounding effect)

- Use_imporved_sigmoid=True

- 

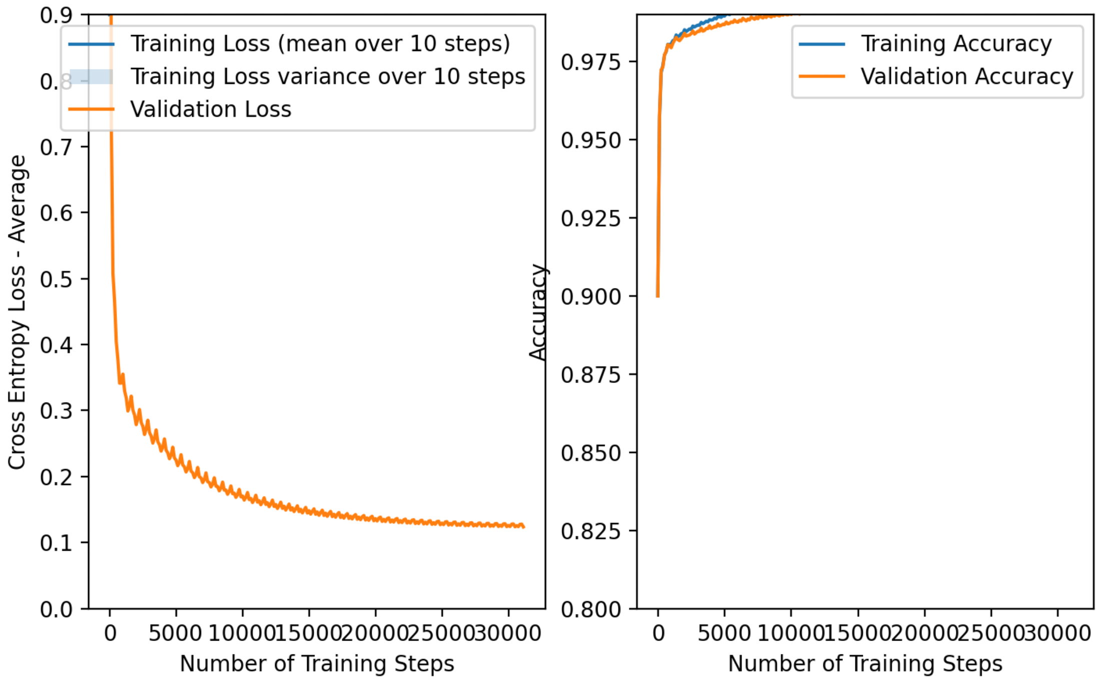
Task 4

- Hidden-layer-units=32:



-
    - We see that the model Validation accuracy is slightly worse when there are fewer neurons in the hidden layers. This is likely because the model capacity is not large enough to model that true data's distribution. It has a high bias.

- Hidden-layer-units=128:

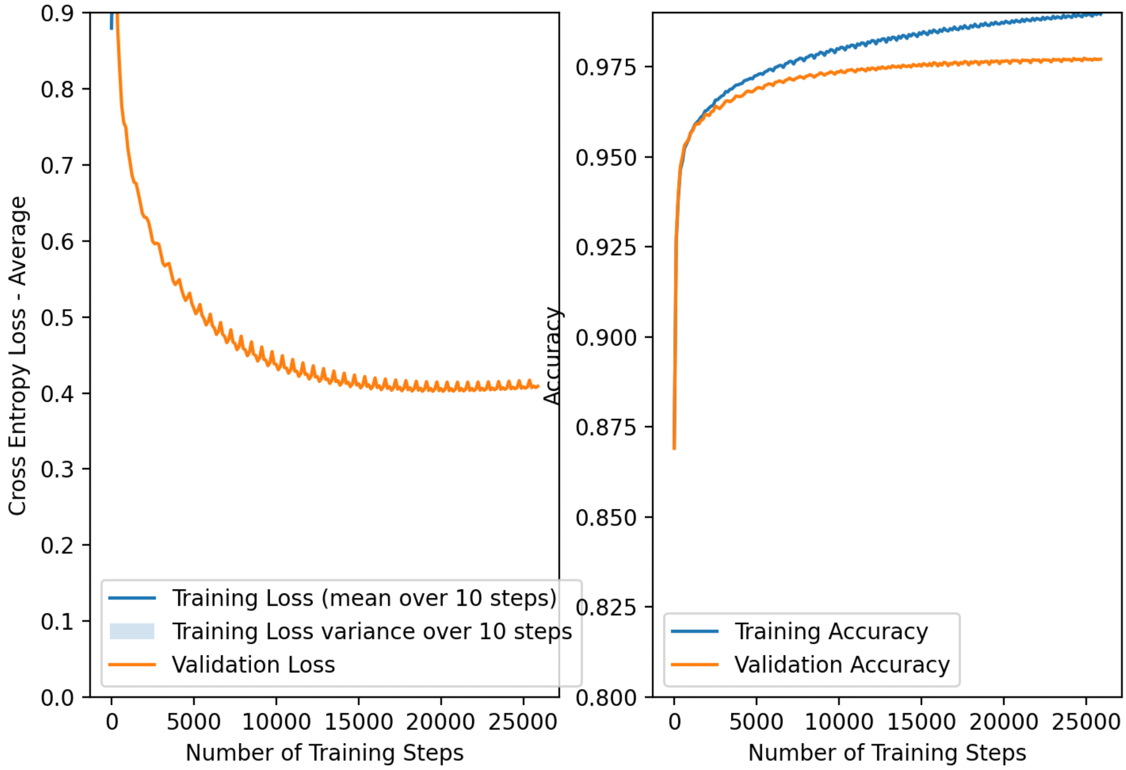


-

- We see that the valditation loss slightly improved compared to the network with 32 hidden units. This is likely because the increased capacity in the model makes it more capable of learning the general patters of the true data distribution.

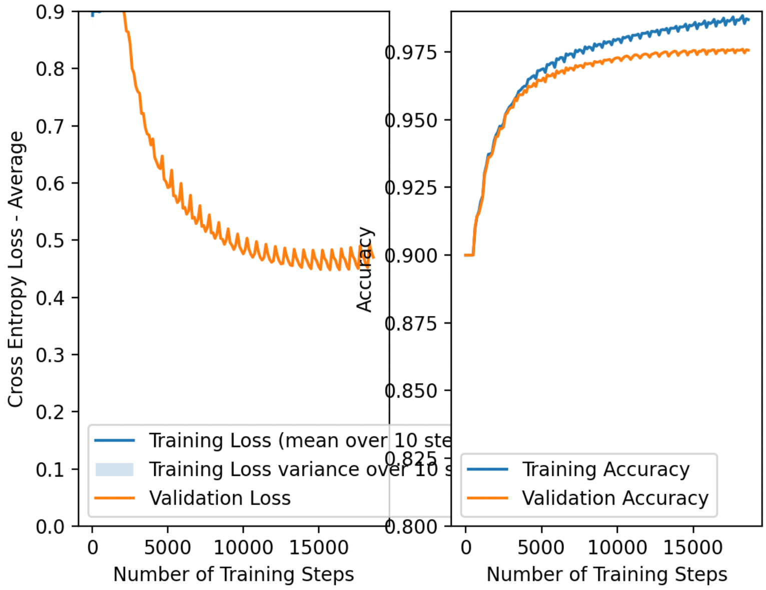
d)

- Network with 2 hidden layers:



-
  - As the epochs increase the gap between the training accuracy and the validation accuracy increases. The model likely is overfitting from on the data from it's increased capacity of having multiple layers. This means it is modelling more of the noise on the traning data, which is not a part of the true data's distribution
- Number of params = (784*64 weights + 64 biases) + (64*64 weights + 64 biases) + (64*10 weights + 10 biases) = 55050

- Network with 10 hidden layers:



- ○
  - ▪ We also see a tendency for the network to overfit here.
- ○ Number of params = 785*64 + 64 + (64*64 + 64)*9 + 64*10 + 64 = 88448 parameters