

Assignment 2 Report

Task 1

```
In [29]: %load_ext autoreload
          %autoreload 2
          %%run task2a.py
          %%run task2.py
          %%run task3.py
          %%run task4c.py
```

The autoreload extension is already loaded. To reload it, use:
%reload_ext autoreload

task 1a)

$$1) X \xrightarrow{w_1} z_1 \xrightarrow{f(z_1)} a_1 \xrightarrow{w_2} z_2 \xrightarrow{f(z_2)} a_2 \rightarrow C$$

δ_2

$$\delta_2 = \frac{\partial C}{\partial z_2} = \frac{\partial C}{\partial a_2} \frac{\partial a_2}{\partial z_2} = -(y - \bar{y}) \text{ (given)}$$

$$\frac{\partial C}{\partial w_1} = \frac{\partial C}{\partial z_2} \frac{\partial z_2}{\partial w_1} = \delta_2 \frac{\partial z_2}{\partial w_1}$$

$$\frac{\partial z_2}{\partial w_1} = \frac{\partial z_2}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial z_2}{\partial a_1} = w_2, \quad \frac{\partial a_1}{\partial z_1} = f'(z_1) \quad \frac{\partial z_1}{\partial w_1} = X$$

$$\frac{\partial z_2}{\partial w_1} = X f'(z_1) \sum w_2$$

$$\frac{\partial C}{\partial w_1} = X f'(z_1) \sum w_2 \delta_2$$

$$\delta_1 = f'(z_1) \sum w_2 \delta_2$$

$$\frac{\partial C}{\partial w_1} = X \delta_1 \quad X = a_0$$

$$\frac{\partial C}{\partial w_2} = a_1 \delta_2$$

$$w_1(n+1) = w_1(n) - \alpha \frac{\partial C}{\partial w_1}$$

$$= w_1(n) - \alpha X \delta_1$$

$$w_2(n+1) = w_2(n) - \alpha a_1 \delta_2$$

Task 2

```
In [2]: from task2a import mean_X
from task2a import std_X
X, Y, *_ = utils.load_full_mnist()
```

```
print("Mean:", mean_X(X))
print("Std: ", std_X(X))
```

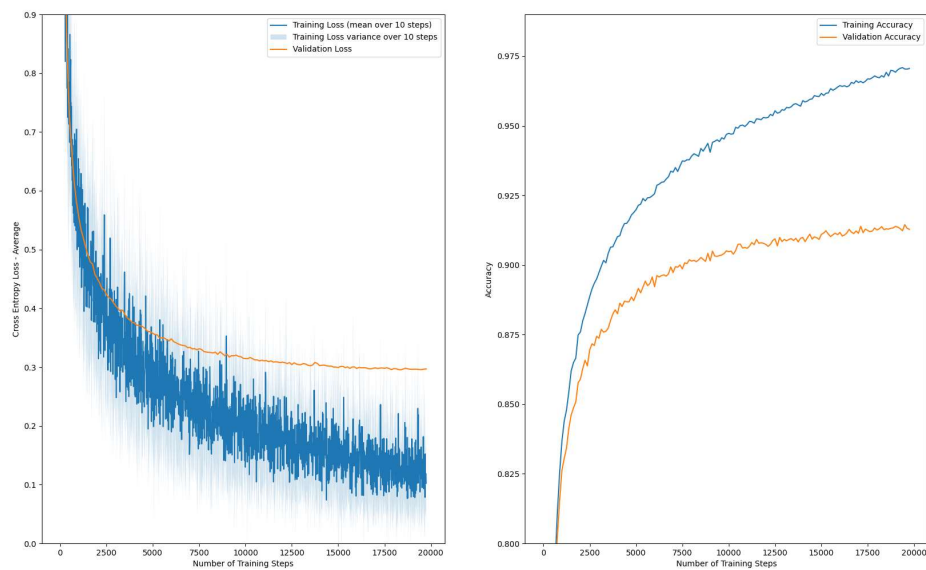
Train shape: X: (20000, 784), Y: (20000, 1)
 Validation shape: X: (10000, 784), Y: (10000, 1)
 Mean: 33.55274553571429
 Std: 78.87550070784701

Task 2a)

```
from task2a import mean_X
from task2a import std_X
X, Y, *_ = utils.load_full_mnist()
print("Mean:", mean_X(X))
print("Std: ", std_X(X))

Train shape: X: (20000, 784), Y: (20000, 1)
Validation shape: X: (10000, 784), Y: (10000, 1)
Mean: 33.55274553571429
Std: 78.87550070784701
```

Task 2c)

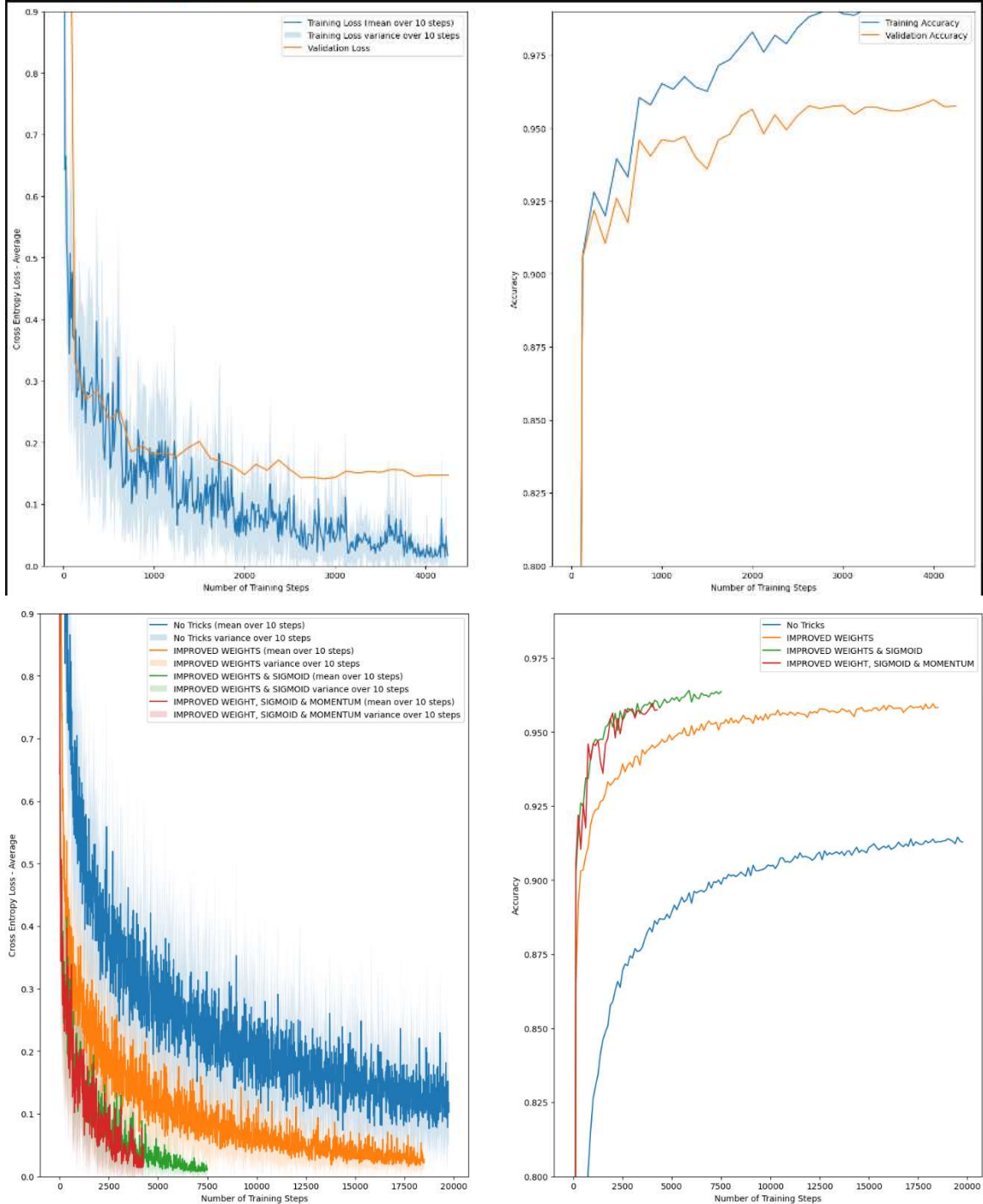


Task 2d)

$$\begin{aligned} &789 \cdot 64 + 64 \cdot 10 \\ &= 50816 \\ &50816 + 64 + 10 = 50890 \end{aligned}$$

Task 3

Final Train Cross Entropy Loss: 0.018751132431979275
 Final Validation Cross Entropy Loss: 0.14715289564348472
 Train accuracy: 0.99605
 Validation accuracy: 0.9575



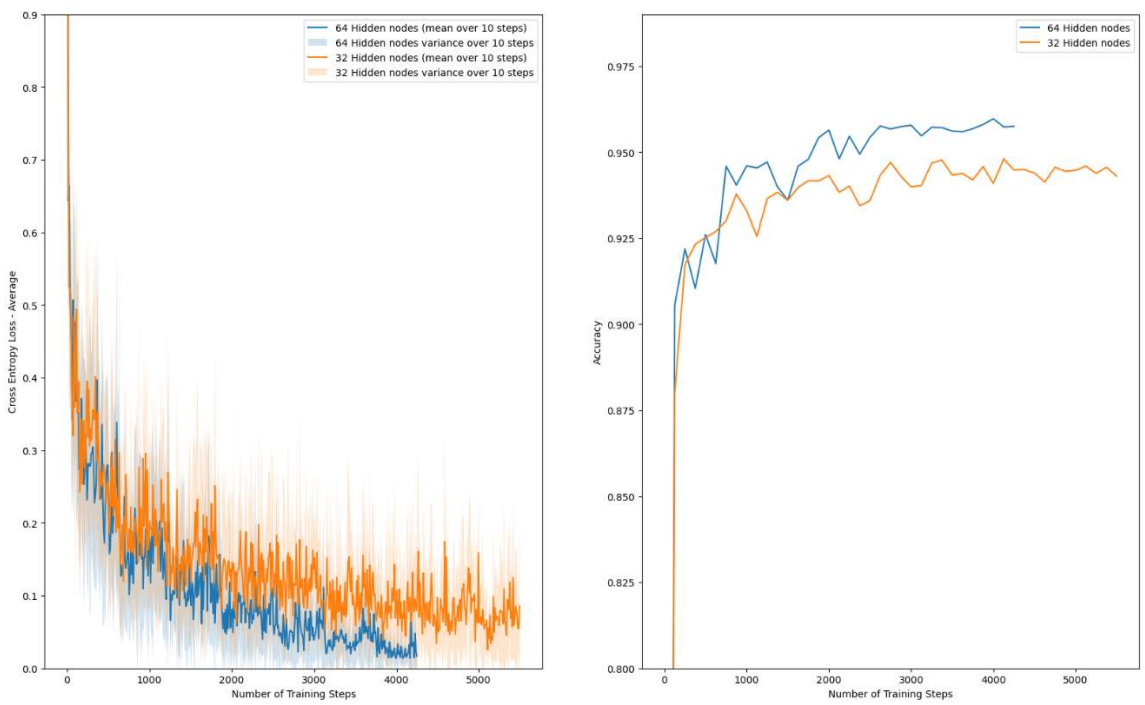
Improved weight init gave the highest step in performance but still same number of epochs

Sigmoid added a bit more performance and early stopping came into effect much earlier

Momentum had even faster convergence speed and had the fastest early stop

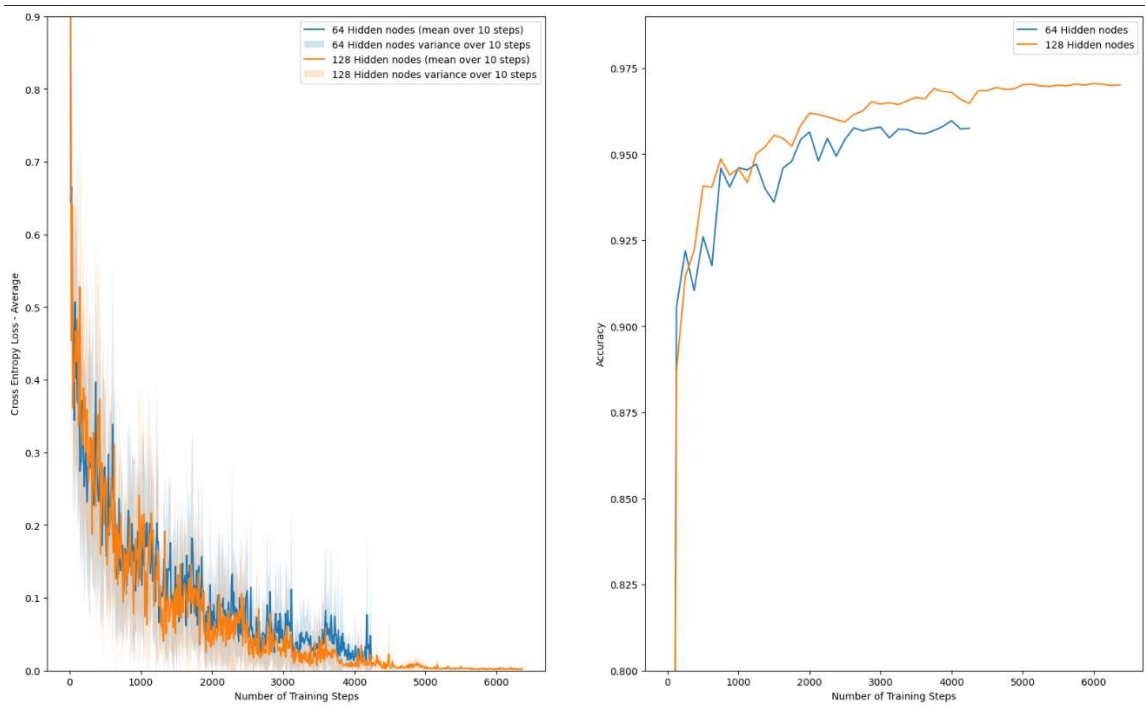
Task 4

Task 4a)



To low number of hidden nodes could make the network to simple and you loose preformance

Task 4b)



To many nodes could result in overfitting and more paramaters to train

Task 4d)

$$784 \cdot x + x \cdot x + x \cdot 10 = 50816$$

$$x^2 + 794x = 50816$$

$$x \approx 59$$

$$784 \cdot 59 + 59 \cdot 59 + 59 \cdot 10 = 50327$$

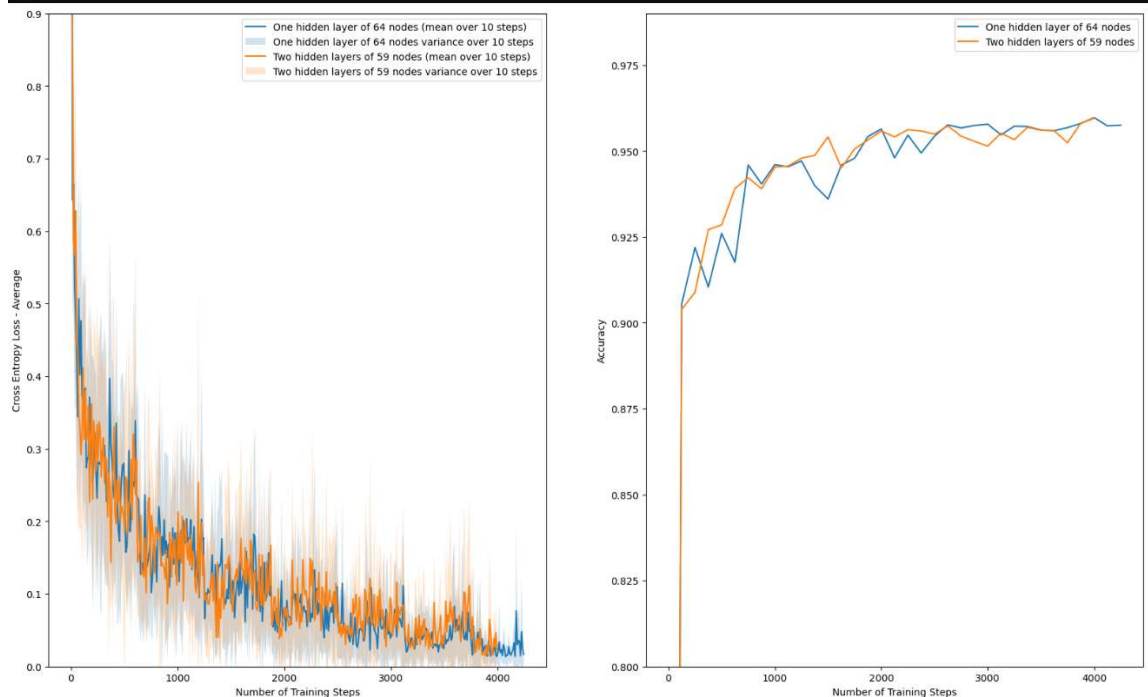
$$50327 + 58 + 58 + 10 = 50453$$

785	59	59	10		785	64	10
0	0	0	0		0	0	0
					= 50890		
= 50453							

```

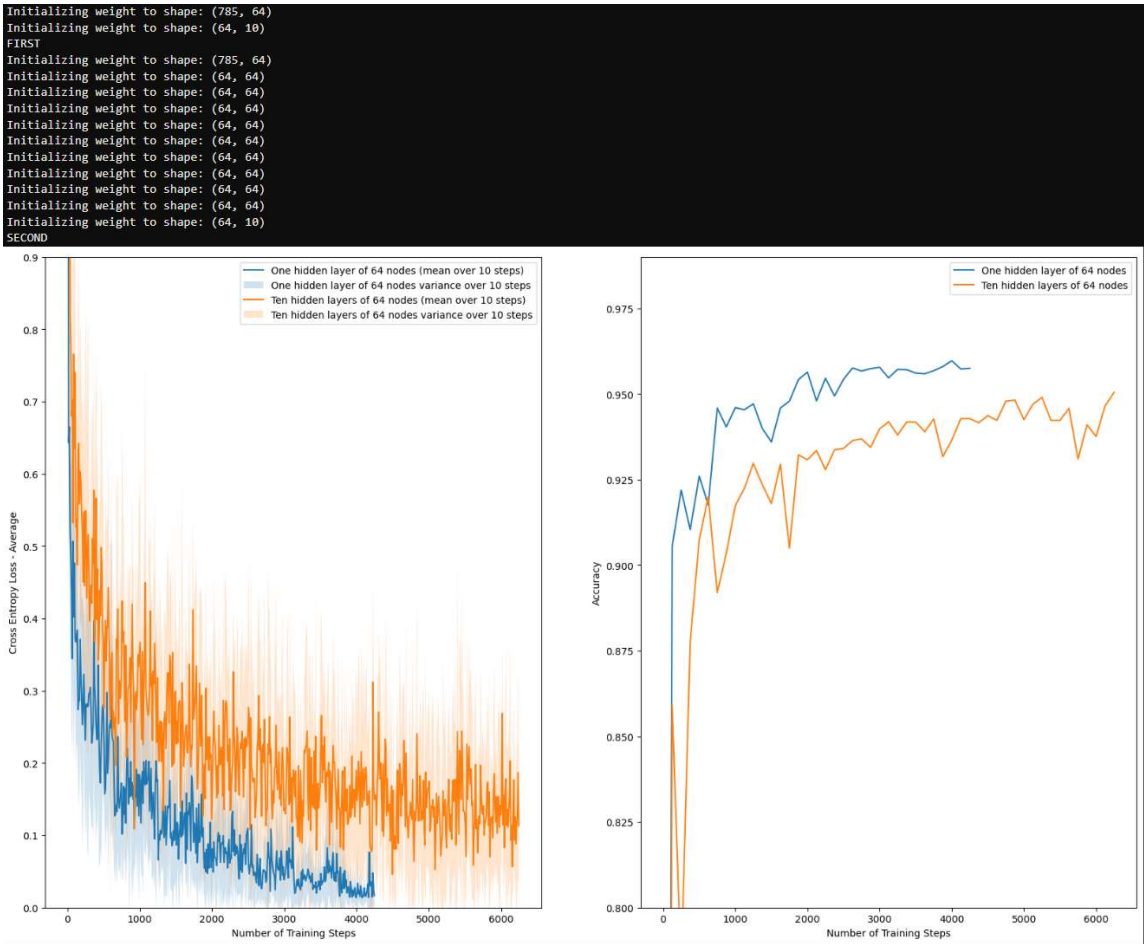
Initializing weight to shape: (785, 64)
Initializing weight to shape: (64, 10)
FIRST
Initializing weight to shape: (785, 59)
Initializing weight to shape: (59, 59)
Initializing weight to shape: (59, 10)
SECOND

```



The two networks compare well both perform equally well

Task 4e)



Task 4f)

