Assignment 2 Report

Task 1

task 1a)

$$\frac{\partial C}{\partial w_{ji}} = \frac{\partial C}{\partial z_j}\frac{\partial z_j}{\partial w_{ji}}$$
$$= \delta_j \frac{\partial z_j}{\partial w_{ji}}$$

$$\frac{\partial z_j}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} w_{j'}^T \cdot x \qquad = \begin{cases} 0, & j' \neq j \\ x_i, & j' = j \end{cases}$$

$$\frac{\partial C}{\partial w_{ji}} = \delta_j \cdot x_i$$

Inserted in the the equation for w_{ji} we get:

$$w_{ji} := w_{ji} - \alpha \delta_j x_i$$

Using the chain rule we can decompose δ_j and sum over all the nodes in layer k:

$$\delta_j = \frac{\partial C}{\partial z_j} = \sum_k \frac{\partial C}{\partial z_k}\frac{\partial z_k}{\partial a_j}\frac{\partial a_j}{\partial z_j}$$

Know that $f'(z_j) = \frac{\partial a_j}{\partial z_j}$, $\delta_k = \frac{\partial C}{\partial z_k}$:

$$\delta_j = f'(z_j)\sum_k \delta_k \frac{\partial z_k}{\partial a_j}$$

$$\frac{\partial z_k}{\partial a_j} = \frac{\partial}{a_j}\sum_j{}' w_{kj'}a_{j'} + b_{j'}$$
$$= \begin{cases} 0, & j' \neq j \\ w_{kj}, & j' = j \end{cases}$$

Which leads to:

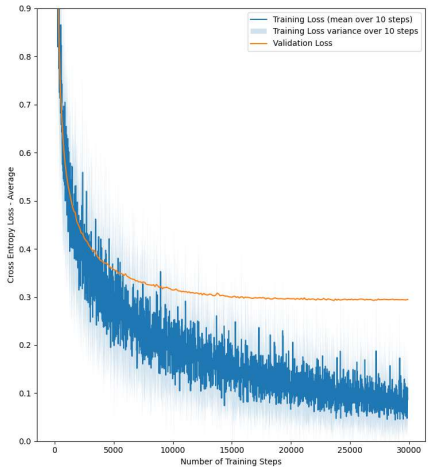
$$\delta_j = f'(z_j)\sum_k \delta_k w_{kj}$$

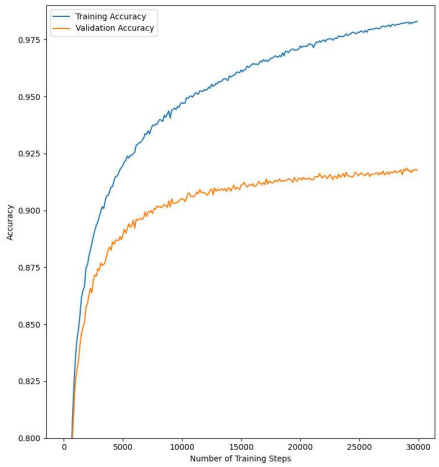
Task 2

Task 2a)

$\mu = 33.55, \sigma = 78.88.$

Task 2c)



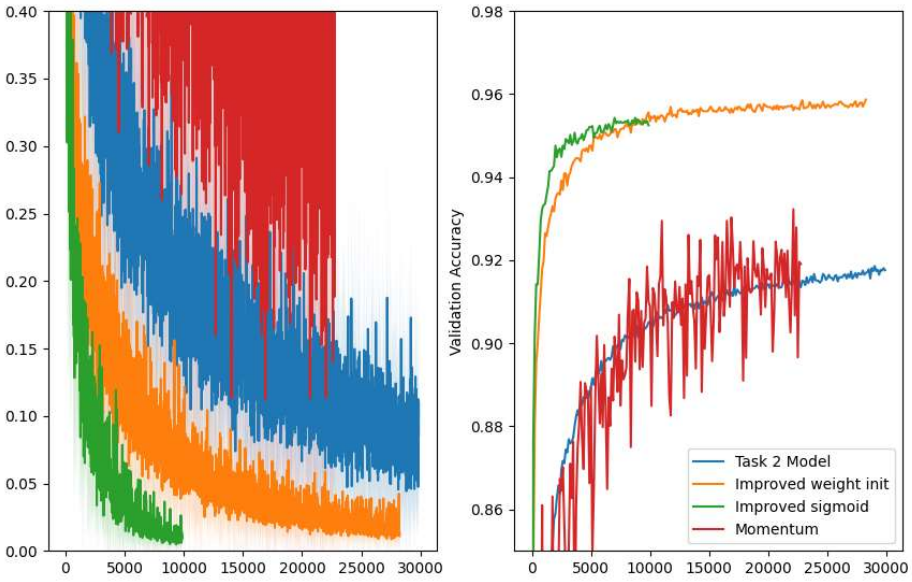


Task 2d)

Parameters = number of weights + number of biases
For the hidden layer: 784 * 64 weights + 64 biases = 50240 parameters
For the output layer: 64 * 10 weights = 640 parameters
Total = 50880 parameters

Task 3

Plot of all the changes implemented below. The performance increases with the implementation of improved input weights and improved sigmoid function, but something is obviously wrong with the way I have implemented momentum. I note that the validation accuracy drastically improves with the implementation of improved input weights. The introduction of an improved sigmoid function has no effect on the validation accuracy, but causes the model to early stop at a much earlier stage.



Task 4

Task 4a)

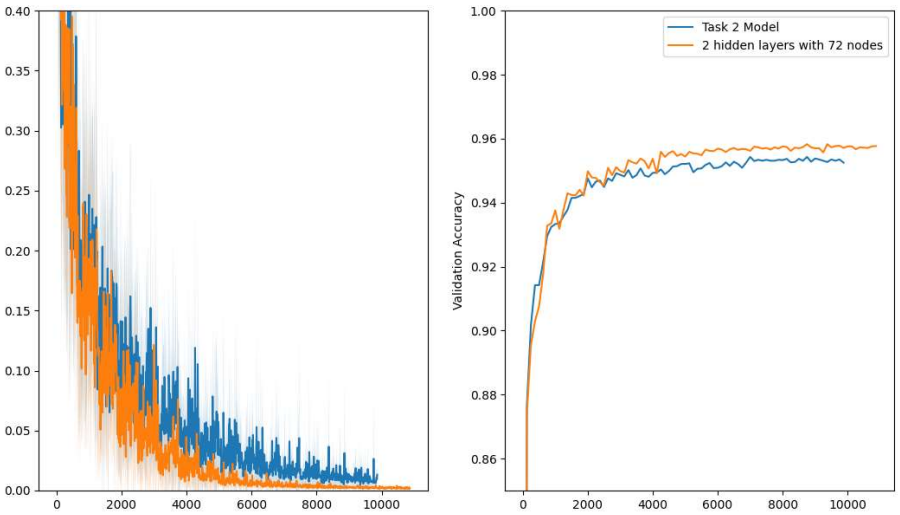
FILL IN ANSWER.

Task 4b)

FILL IN ANSWER

Task 4d)

Choose to implement two hidden layers with 72 nodes. The number of parameters is then 62 424. The training accuracy is marginally improved by this, but the model takes longer to early stop.



Task 4e)

The training accuracy actually gets worse. I suspect this might have to do with the gradient "vanishing", i.e. the gradient gets smaller and smaller for each layer and the impact on the "deepest" layers is then in practice zero.