Theoretical part: Question 1

First denote $f(x,w)_i = z_i$ From the chain rule we get

$$\frac{\partial L(\hat{y},y)}{\partial w} = \sum_i \frac{\partial L(\hat{y},y)}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial w}$$

$$\frac{\partial \hat{y}_i}{\partial w} = \sum_k \frac{\partial \hat{y}_i}{\partial z_k}\frac{\partial z_k}{\partial w}$$

and since

$$\hat{y}_i = softmax(x)_i = \frac{e^{f(x,w)_i}}{\sum_j e^{f(x,w)_j}} = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

we get

$$\frac{\partial \hat{y}_i}{\partial z_k} = \begin{cases} \frac{e^{z_i}}{\sum_j e^{z_j}}\left(1 - \frac{e^{z_i}}{\sum_j e^{z_j}}\right) = \hat{y}_i(1-\hat{y}_i) & , \text{i = k} \\ -\frac{e^{z_i}}{\sum_j e^{z_j}}\frac{e^{z_k}}{\sum_j e^{z_j}} = -\hat{y}_i\hat{y}_k & , \text{i} \neq \text{k} \end{cases}$$

$$\frac{\partial \hat{y}_i}{\partial w} = \sum_{k\neq i} -\hat{y}_i\hat{y}_k\frac{\partial z_k}{\partial w} + \hat{y}_i(1-\hat{y}_i)\frac{\partial z_i}{\partial w} = \sum_k -\hat{y}_i\hat{y}_k\frac{\partial z_k}{\partial w} + \hat{y}_i\frac{\partial z_i}{\partial w}$$

$$\frac{\partial z_i}{\partial w} = \frac{\partial f(x,w)_i}{\partial w}$$

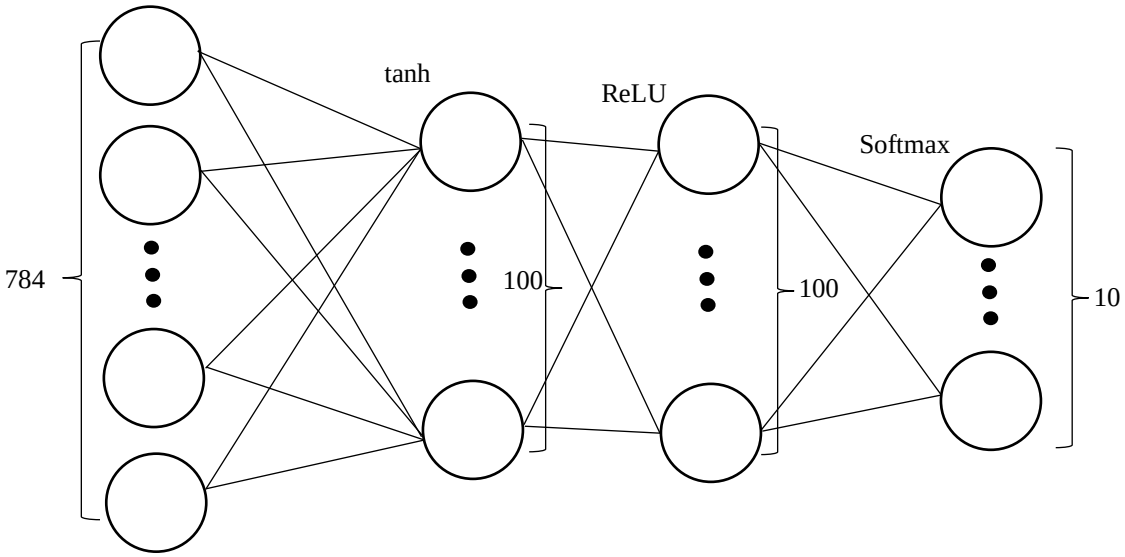
and so we get

$$\frac{\partial L(\hat{y},y)}{\partial w} = \sum_i \left(\sum_k -\hat{y}_i\hat{y}_k\frac{\partial z_k}{\partial w} + \hat{y}_i\frac{\partial z_i}{\partial w}\right)\frac{\partial L(\hat{y},y)}{\partial \hat{y}_i}$$

Practical part:

Question 1:

Network architecture:



Hyperparameters:

In the ADAM implementation the hyperparameters were $\beta_1 = .9, \beta_2 = .999, \epsilon = 10^{-8}$

In the training process, I set the batch_size parameter to 100

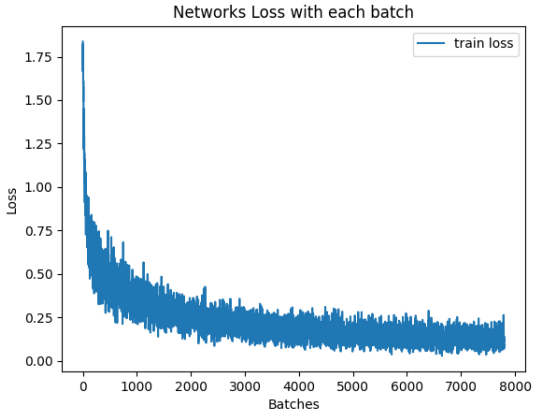
Optimization:

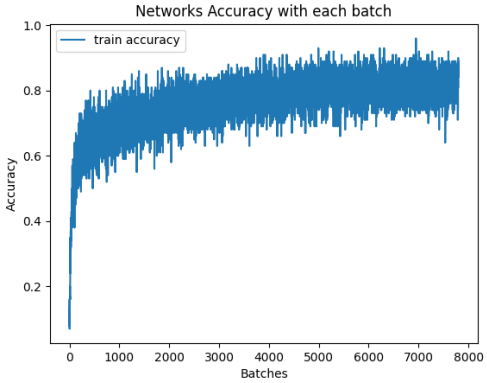
Implemented the ADAM optimizer in the backward function

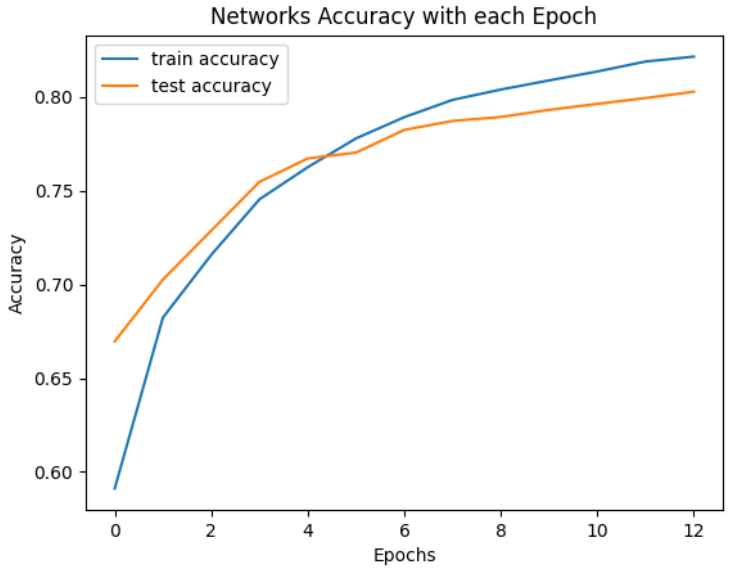
Calculated the gradients w.r.t. to each weight matrix, bias and updated the weights and biases accordingly

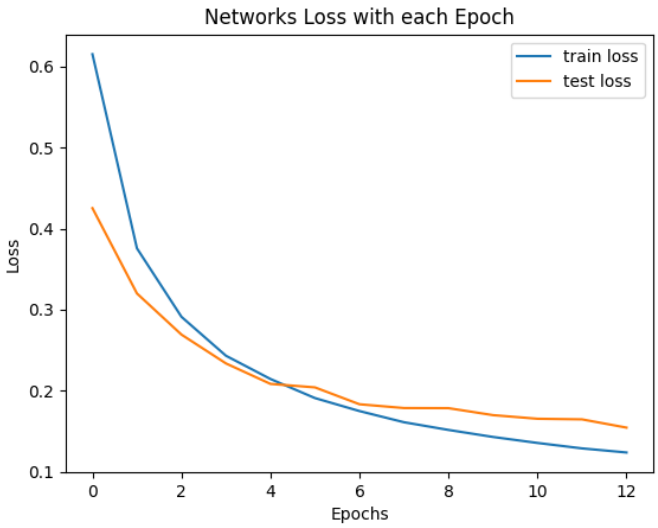
Short summary & conclusions:

Although I calculated the gradient of softmax in Question 1 (Theoretical part) the training process seemed to be negatively affected from the gradient of the softmax and the network loss seemed to improve better when the softmax gradient weren't multiplied

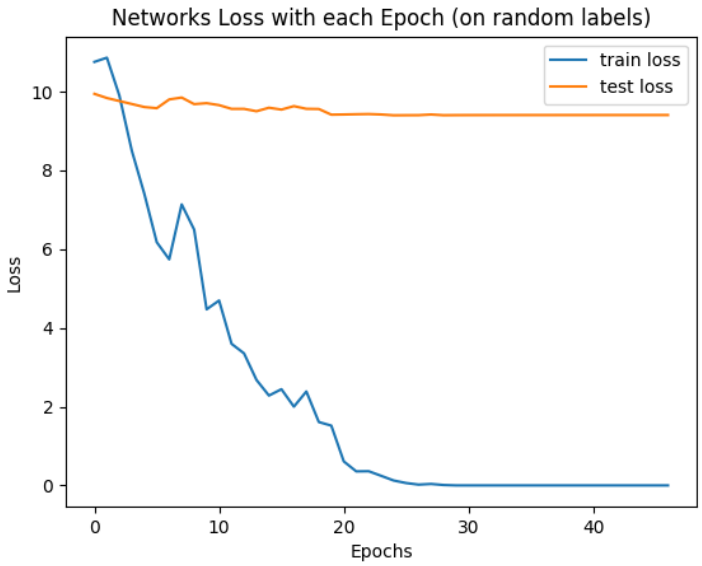

Networks Loss with each batch
train loss
Loss
Batches


Networks Accuracy with each batch
train accuracy
Accuracy
Batches


Networks Accuracy with each Epoch
train accuracy
test accuracy
Accuracy
Epochs


Networks Loss with each Epoch
train loss
test loss
Loss
Epochs

Question 2:



although the loss on the training set is improving and almost 0, the loss on the test set is almost the same as the initial loss.

And that's due to the face that the network attempted to learn the labels that's being generated at random.

And of course we can't learn the "toss of a coin".