Q 1.1

$$
\begin{aligned}
&\text{Softmax}(x+c) \\
&= \left[\frac{e^{x_i+c}}{\sum_j e^{x_j+c}}\right]_i \\
&= \left[\frac{e^c e^{x_i}}{\sum_j e^c e^{x_j}}\right]_i \\
&= \left[\frac{e^c e^{x_i}}{e^c \sum_j e^{x_j}}\right]_i \\
&= \left[\frac{e^{x_i}}{\sum_j e^{x_j}}\right]_i \\
&= \text{softmax}(x)
\end{aligned}
$$

Therefore, softmax is invariant to translation

If $c=0$, $e^{x_i+c} = e^{x_i} \in (0, +\infty)$

If $c=-\max x_i$ $e^{x_i+c} \in [0, 1]$

Setting $c=-\max x_i$ would help avoiding overflow

Q1.2

- The range of each element is $(0, 1]$

  The sum of all elements is 1

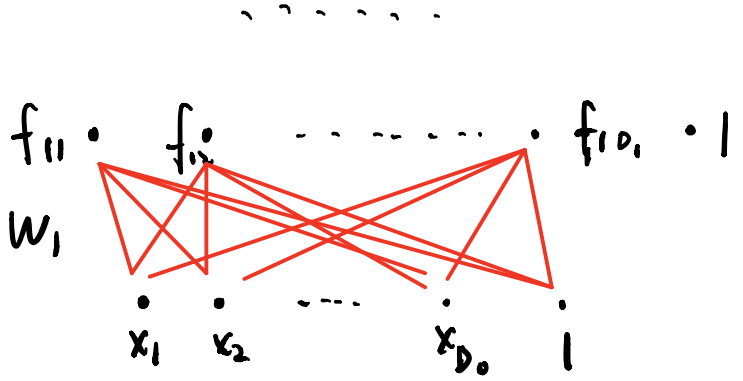
- Softmax takes an arbitary real valued x and turns it into a probability distribution of the values in x

- $S_i = e^{x_i}$ $\Rightarrow$ Calculate the exponential value of each element in vector x

  $S = \sum S_i$ $\Rightarrow$ Calculate the sum of all exponential values

  $\text{softmax}(x_i) = \frac{1}{S} S_i$ $\Rightarrow$ Calculate the probability distribution

Q 1.3



$x \in \mathbb{R}^{D_0 \times 1}$ $f_1 = \mathbb{R}^{D_1 \times 1}$, $f_2 = \mathbb{R}^{D_2 \times 1}$ ...

$W_1 = \mathbb{R}^{D_0 \times D_1}$ $W_2 = \mathbb{R}^{D_1 \times D_2}$ ...

$b_1 = \mathbb{R}^{D_1}$, $b_2 = \mathbb{R}^{D_2}$ ...

$f_1 = W_1^T x + b_1$

$f_2 = W_2^T f_1 + b_2$

$\Rightarrow f_2 = W_2^T W_1^T x + W_2^T b_1 + b_2$

$\Rightarrow f_3 = W_3^T W_2^T W_1^T x + W_3^T W_2^T b_1 + W_3^T b_2 + b_3$

$\Rightarrow f_n = \left[\prod_{i=1}^{n} W_n^T\right] x + \left[\sum_{j=1}^{n} \left(\prod_{k=j+1}^{n} W_k^T\right) b_j\right]$

$= w' x + b'$

Therefore, multi-layer neural network without non-linear function are equivalent to linear regression

Q 1.4

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma'(x) = -\frac{1}{(1+e^{-x})^2}(-e^{-x})$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

$$= \frac{e^{-x}+1}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2}$$

$$= \sigma(x) - \sigma^2(x)$$

Q 1.5

$$y_j = \sum_{i=1}^{d} w_{ij} x_i + b_j$$

$$\Rightarrow \frac{\partial y_j}{\partial w_{ij}} = x_i \qquad \frac{\partial y_j}{\partial x_i} = w_{ij} \qquad \frac{\partial y_j}{\partial b_j} = 1$$

$$\frac{\partial J}{\partial w} = \begin{bmatrix} \frac{\partial J}{\partial w_{11}} & \cdots & \frac{\partial J}{\partial w_{1k}} \\ \vdots & & \\ \frac{\partial J}{\partial w_{d1}} & \cdots & \frac{\partial J}{\partial w_{dk}} \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial y_1} x_1 & \cdots & \frac{\partial J}{\partial y_k} x_1 \\ \vdots & & \\ \frac{\partial J}{\partial y_1} x_d & \cdots & \frac{\partial J}{\partial y_k} x_d \end{bmatrix}$$

$$= \begin{bmatrix} \delta_1 x_1 & \cdots & \delta_k x_1 \\ \vdots & & \\ \delta_1 x_d & \cdots & \delta_k x_d \end{bmatrix} = X \delta^T \in \mathbb{R}^{d \times k}$$

$$\frac{\partial J}{\partial x} = \begin{bmatrix} \frac{\partial J}{\partial x_1} \\ \vdots \\ \frac{\partial J}{\partial x_d} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{k} \frac{\partial J}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_1} \\ \vdots \\ \sum_{j=1}^{k} \frac{\partial J}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_d} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{k} \delta_j w_{1j} \\ \vdots \\ \sum_{j=1}^{k} \delta_j w_{dj} \end{bmatrix}$$

$$= W \delta \in \mathbb{R}^{d \times 1}$$

$$\frac{\partial J}{\partial b} = \begin{bmatrix} \frac{\partial J}{\partial b_1} \\ \vdots \\ \frac{\partial J}{\partial b_k} \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial y_1} \cdot \frac{\partial y_1}{\partial b_1} \\ \vdots \\ \frac{\partial J}{\partial y_k} \cdot \frac{\partial y_k}{\partial b_k} \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial y_1} \\ \vdots \\ \frac{\partial J}{\partial y_k} \end{bmatrix} = \delta \in \mathbb{R}^{k \times 1}$$

Q1.6

1. $x \to f_1 \to \sigma(f_1) \to f_2(\sigma(f_1)) \to (f_2(\sigma(f_1))) \cdots \to y$

$$\frac{\partial y}{\partial x} = \frac{\partial \sigma(f_n)}{\partial f_n} \cdot \frac{\partial f_n}{\partial \sigma(f_{n-1})} \cdot \frac{\partial \sigma(f_{n-1})}{\partial f_{n-1}} \cdot \frac{\partial f_{n-1}}{\partial \sigma(f_{n-2})} \cdots$$

$$= \prod_{i=1}^{n} \frac{\partial \sigma(f_i)}{\partial f_i} \cdot \prod_{j=2}^{n} \frac{\partial f_j}{\partial \sigma(f_{j-1})} \cdot \frac{\partial f_1}{\partial x}$$

$$= \prod_{i=1}^{n} (f_i)(1-\sigma(f_i)) \cdot \prod_{j=2}^{n} \frac{\partial f_j}{\partial \sigma(f_{j-1})} \cdot \frac{\partial f_1}{\partial x}$$

Since $(f_i)(1-\sigma(f_i)) \in (0, 1)$

When $n \to \infty$ $\prod_{i=1}^{n} (f_i)(1-\sigma(f_i)) \to 0$

$$\frac{\partial y}{\partial x} \to 0$$

Therefore, when using sigmoid activation, it might lead to a "vanish gradient" problem if it is used for many layers

2. $\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$

$= 1 - \frac{2e^{-2x}}{1 + e^{-2x}}$

$= 1 - \frac{2}{e^{2x} + 1} \quad \in (-1, 1)$

$S(x) \in (0, 1)$

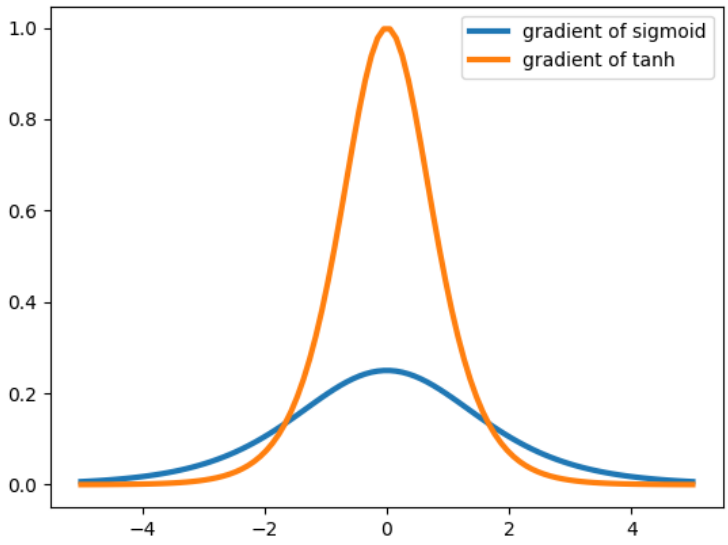
The output range is symmetric for tanh, thus avoid bias in the gradients.

Also, $\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh^2(x)$

$\frac{\partial G(x)}{\partial x} = G(x) - G^2(x)$

tanh has stronger gradients than sigmoid, so it is less likely to have "gradient vanish" and has higher convergence speed through training

3.



tanh has stronger gradient than sigmoid around value of zero, therefore tanh is less likely to have vanishing gradient problem

4. $\tanh(x) = 2\sigma(2x) - 1$

Q 2.1.1

If all parameters in the network are initialized to zero, then all neuron will output zero through the forward propogation. Through back propogation, the same output will lead to the same gradients for all parameters. Therefore, all parameters will get updated exactly same and the parameters will always be same through the trainning process. It is impossible to obtain the global optimum in this way.

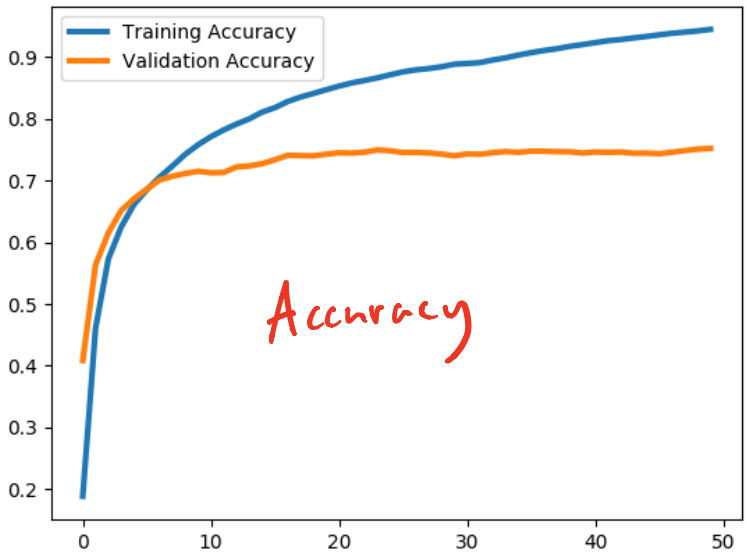
Q 2.1.3

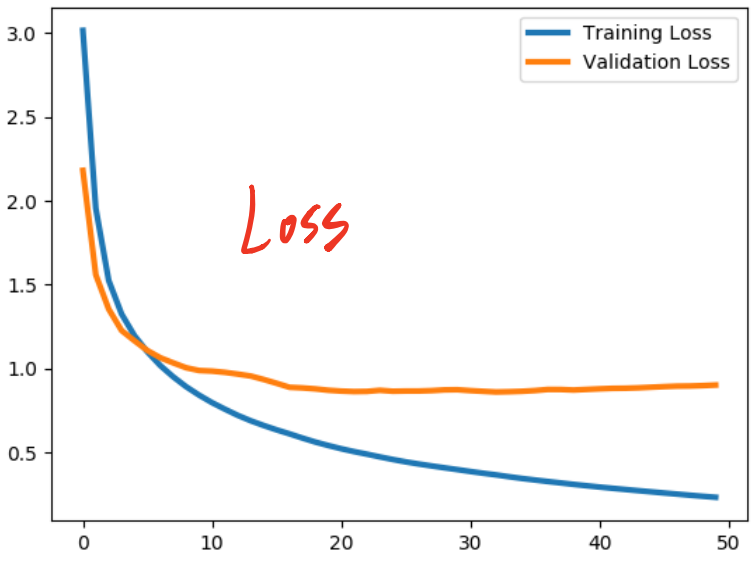
Random initialization break the symmetry of the network thus increase the chance of the network to reach the global optimum.

By scaling the initialization depending on layer size, it makes the variance of the parameters' gradients be same for all layers. Then the output value through forward propagation and the gradients through back propagation won't be too large or too small, thus avoid the gradient explosion or vanishing problem.

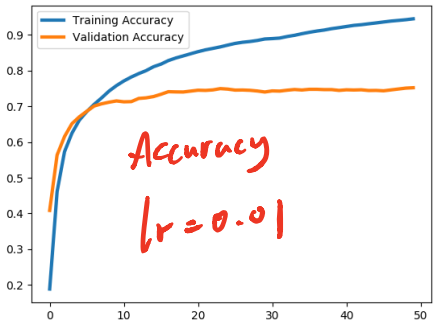
Q3.1.1

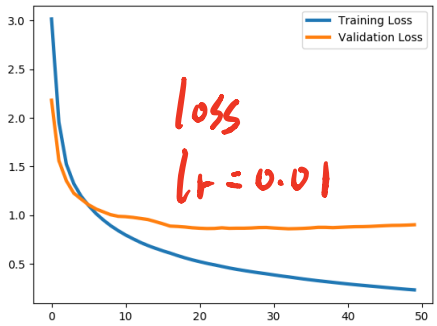
learning rate : 0.01 Validation Accuracy : 76.6%

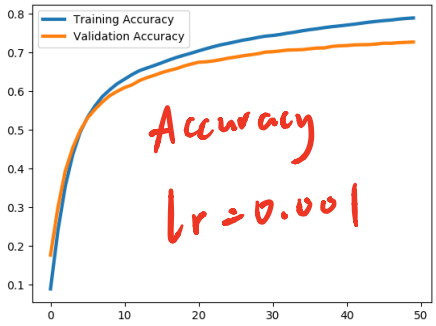


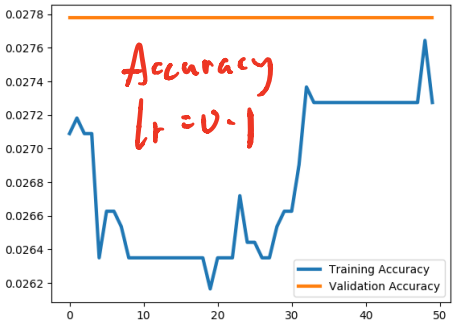


Q3.1.2 Final test accuracy: 77.0%



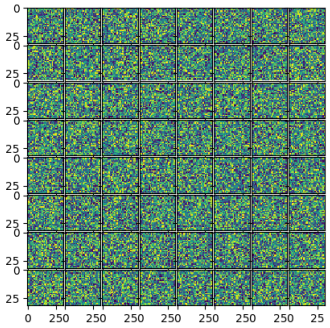




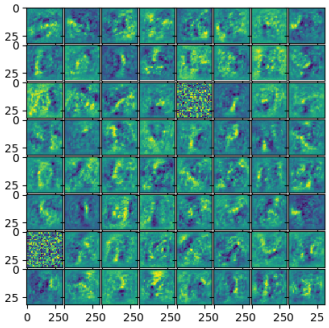


The best learning rate is 0.01. With the best learning rate, the network converge to high accuracy and low loss fastly. With a smaller learning rate 0.01, the converge speed of the network is lower than the best learning rate. With a larger learning rate 0.1, the gradients update at each each iteration is too large, the model failed to reach local optimum and thus learned nothing.

Q 3.1.3



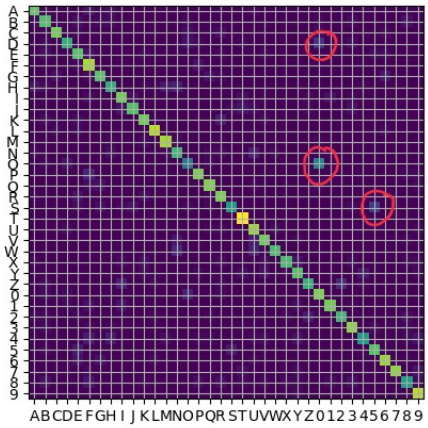
Weights of Initialization



Weights after Training

The initial weights before training are basically random noisy. The weights after training clearly show some patterns which means that the network is able to learning some meaningful things through training.

Q3.1.4



Confusion Matrix

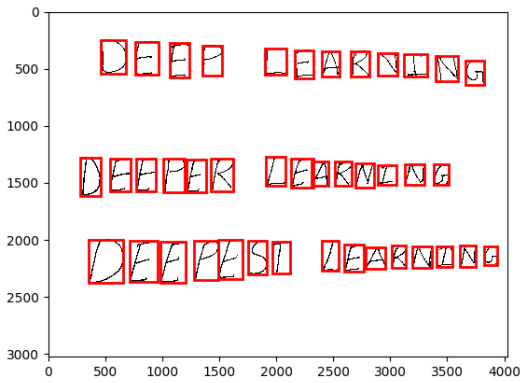
The top confusing pairs of classes are pairs (O, 0), (S, 5), (D, 0). The characters with in the pair looks very similar. It is even difficult for human to distinguish them in hand written given an only single image.

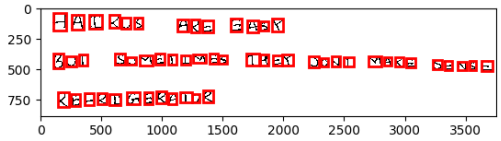
Q 4.1

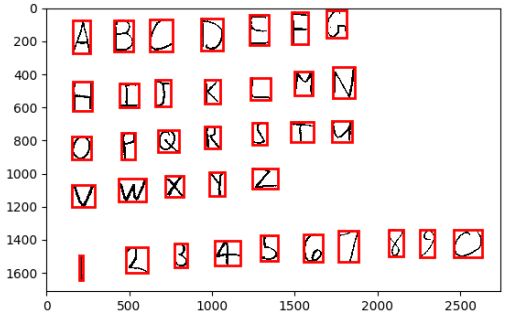
Assumption 1: Two characters should be isolated clearly from each other.
Potential negative example:

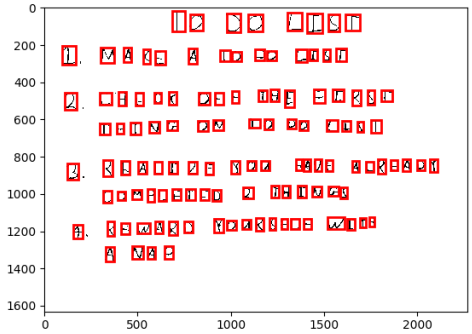
Assumption 2: The characters within the same image should be in similar size.
Potential negative example:

Q4.3









Q4.4

Deep:

```
D E E P L E A R M I N G
D E E P E R L E A R K I N G
D E E P 8 S P L E A R Q I N 6
```

Haiku:

```
H A I K U S A R H H A G Y
B W T S D M E T I M E G T H E T D Q W T M A K 6 B H N G E
R E F R I G E R A T Q R
```

letters

```
Q B C D B F G
H I J K L M N
Q P Q R S T U
V W X Y Z
1 Z 3 G S 6 7 8 9 J
```

list:

```
T 0 D Q L I S T
I N A K 6 A T D Z B L I S T
2 L H F C K J F E T H 8 F I R S T
T H I N G B N T Q D Q L I S T
3 R L A L I Z E Y O U H A V E 2 L R 6 A D T
6 O M P L F T L D J T H I N G S
Q R F W A R D Y O W R G E L F W E T H
A N A P
```

Q5

```
--------------------------------------------------
D E E P L E A R N I N G
D E E P E R L E A R N I N G
D E E P E S T L E A R N I N G
--------------------------------------------------
T O D O L I S T
I M A K E A T D D Q L I S T
2 C H E C K D F F T H E F I R S T
T H I N G D N T D D Q C I 5 T
3 R B A L I Z E Y D U H A V E A L R E A D Y
C D M P L E T E D Z T H I N G S
4 R E W A R D Y D U R S E L F W I T H
A N A P
--------------------------------------------------
A B C D E F G
H I J K L M N
O P Q R S T U
V W X Y Z
1 Z 3 4 S G 7 8 g Q
--------------------------------------------------
H A I K D 5 A R E E A 5 Y
B U T S Q M E T I M E 5 T H E Y D D N T M A K E S E N 5 E
R E F R I G E R A T D R
--------------------------------------------------
```

Deep

list

letters

haiku