Kenny Ostyn Hanson

16-720

HW 5

Q1.1 Prove that softmax is invariant to translation

$$\text{softmax}(x_i) = \text{softmax}(x+c) \quad \forall c \in \mathbb{R}$$

$$\text{softmax}(x+c) = \left(\frac{e^{x_i+c}}{\sum_j e^{x_j+c}} \right)_i$$

$$= \left(\frac{e^{x_i} e^{c}}{\sum_j e^{x_j} e^{c}} \right)_i$$

$$= \left(\frac{e^{x_i}}{\sum_j e^{x_j}} \right)_i \equiv \text{softmax}(x_i) \quad \text{(Proven)}$$

softmax(x) is invariant to translation

For $c=0$, $e^{x_i+c} = e^{x_i} : (0 \to +\infty)$

For $c = -\max x_i$ $e^{x_i+c} : (0 \to 1)$

Using $c = -\max x_i$ will simplify the process, avoiding overflow.

Q1.2 Softmax → 3 step process

- Range for each element : 0 - 1
  Sum for all elements : 1

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

- ① $S_i = e^{x_i}$ → Calculate the exponential value of each element
  ② $S = \sum S_i$ → Calculate the sum of exponential values
  ③ $\text{softmax}(x_i) = \frac{1}{S} S_i$ → Calculate the probability distribution of the values.

Q1.4 Derive gradient of sigmoid function

Show that gradient $\rightarrow \sigma(x)$ (without accessing x)

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

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

$$= \boxed{\frac{1}{e^{x}(1+e^{-x})^{2}}} \quad \text{(Derived)}$$

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

$$\boxed{\therefore \sigma'(x) = \sigma(x) - \sigma^{2}(x)} \quad \text{(shown)}$$

Q1.5 Show in getting $\frac{dy}{dw}$, $\frac{dJ}{dx}$ & $\frac{dJ}{db}$ ⇝ In scalars & reform matrix

$y = Wx + b$ or $y_i = \sum_{j=1}^{d} x_j W_{ij} + b_i$

$\frac{dJ}{dy} = \delta \in \mathbb{R}^{k \times 1}$ $\quad W \in \mathbb{R}^{k \times d}$ $\quad x \in \mathbb{R}^{d \times 1}$ $\quad b \in \mathbb{R}^{k \times 1}$

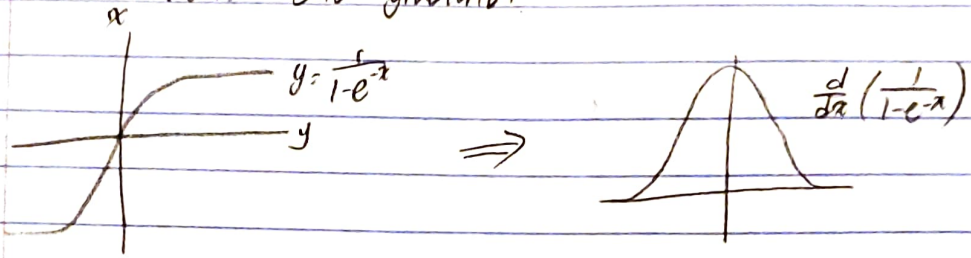
$\frac{dy_i}{dw_{ij}} = x_j \Rightarrow \frac{dJ}{dw} = \frac{dJ}{dy} \cdot \frac{dy}{dw} = \underbrace{\delta x_j}_{\substack{k \times 1 \quad 1 \times d \\ d \times k \text{ or } k \times d}} = \begin{bmatrix} \delta_1 x_1 & \cdots & \delta_k x_1 \\ \vdots & & \\ \delta_1 x_d & & \delta_k x_d \end{bmatrix} \in \mathbb{R}^{d \times k}$

$\frac{dJ}{dx} = \frac{dJ}{dy} \cdot \frac{dy}{dx} = \underbrace{\delta \cdot W_{ij}}_{\substack{k \times 1 \quad k \times d \\ d \times 1}} = \begin{bmatrix} \sum_{j=1}^{k} \delta_j W_{1j} \\ \vdots \\ \sum_{j=1}^{k} \delta_j W_{dj} \end{bmatrix} \in \mathbb{R}^{d \times 1}$

$\frac{dJ}{db} = \frac{dJ}{dy} \cdot \frac{dy}{db} = \underset{k \times 1}{\delta} \cdot \underset{k \times k}{1} = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_k \end{bmatrix} \in \mathbb{R}^{k \times 1}$

↓ const

Q1.6 1. The sigmoid activation function poses to have the "vanishing gradient" problem that can be explained through the plot. As more layers use it, the graph plateaus indicating an eventual zero gradient.



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

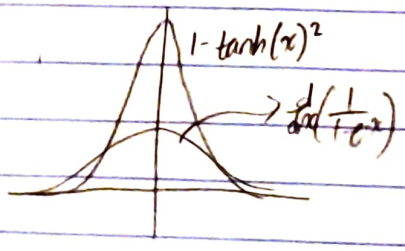
$= 1 - \frac{2e^{-2x}}{1+e^{-2x}}$

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

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

Sigmoid $= \frac{1}{1-e^{-x}} \in (0, 1)$

Tanh has a symmetric output until -1 unlike sigmoid. This can help eliminate bias in the gradient. Also, tanh is less likely to have the "vanishing gradient" problem because of its more significant gradient.

3. 

tanh has higher derivative values than sigmoid, which reduces the vanishing problem chance.

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

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

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

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

$= -(1 - 2\sigma(2x))$

$= \frac{1}{2\sigma(2x)-1}$

1. HELLO → good

HELLO → bad

HELLO ← HELLO

2. HELLO → good

HELLO → bad

Q1.1

Q2.1.1 Network Initialization

A network initialized to all zeros will result the nerurons to ouptput zero, with forward propagation. In addition, the back propagation will result to the same gradients for each test train. With specific layer weights and biases, the parameters will be the same throughout the training process and the optimum will not be obtained.

Q 2.1.3 Network Initialization

Random numbers initialization is desired, because it breaks the network symmetry. There is a faster chance for the network to reach global optimum without the symmetry.

Scaling the initialization depending layer size is done to make the output value (forward propagation) and gradients (back propagation) not too big or too small by making the variance of the parameter's gradients constant for every layer. This will prevent the vanishing problem or gradient explosion.

Q 3.1

Best accuracy of 75.6 % is achieved with batch size of 48 and learning size of 0.01, along with the given 40 epochs & hidden_size of 64.

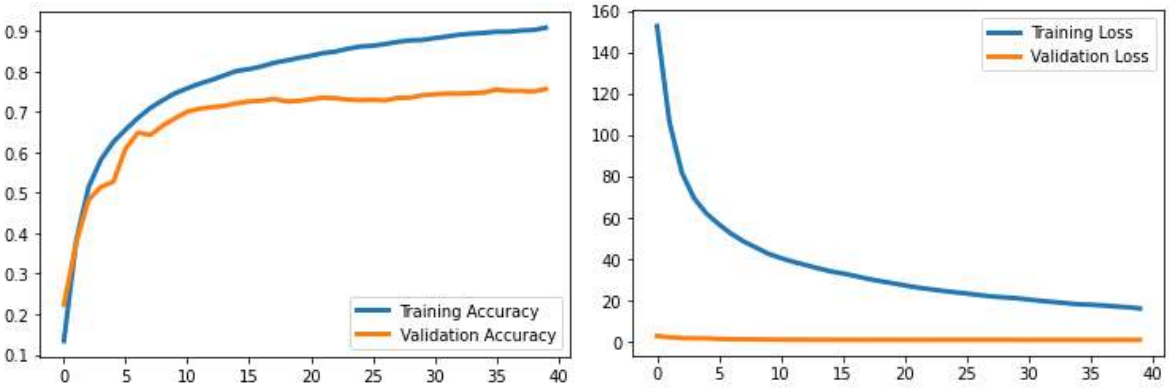


Figure 1. Best learning rate of 75.6% using 0.01 learning size.

Q 3.2

Best accuracy of 75.6 % is achieved with batch size of 48 and learning size of 0.01, along with the given 40 epochs & hidden_size of 64. The plots are shown below. Further figures show the loss and accuracy using learning size of 0.1 and 0.001 respectively. A lower learning rate of 0.001, the time needed to reach the desired accuracy becomes longer, making the final accuracy at the same epoch value to be slightly lower. With a larger learning rate, the gradient updates at a much larger rate to obtain the local optimum. In other words, the gradient difference is too big that it skips notable learning points.

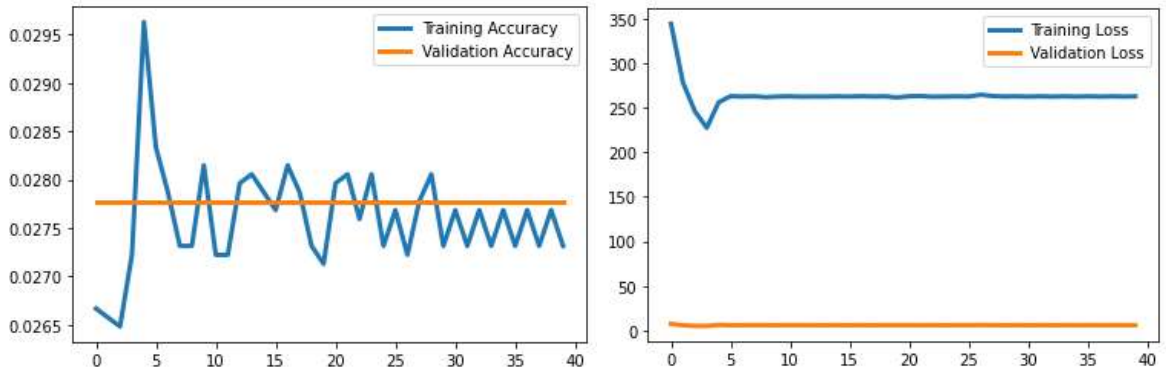


Figure 2. Learning rate of 2.78% using 0.1 learning size.

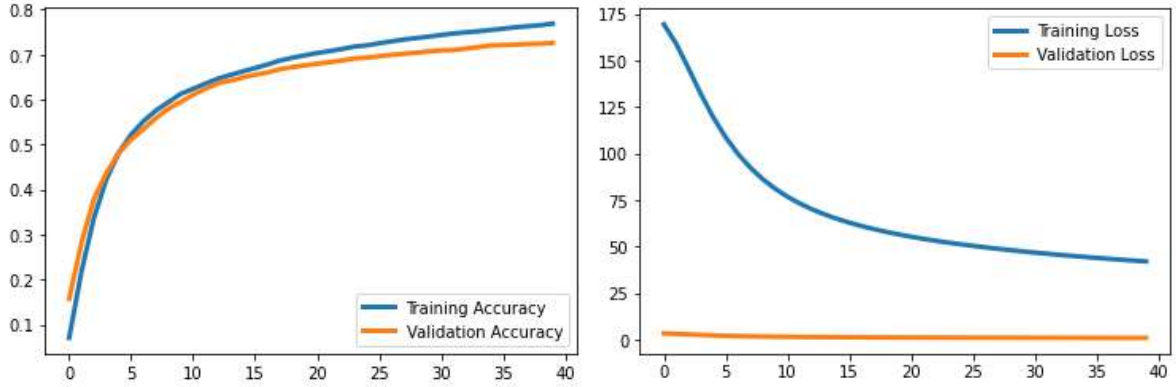


Figure 3. Learning rate of 72.58% using 0.001 learning size.

Q 3.3

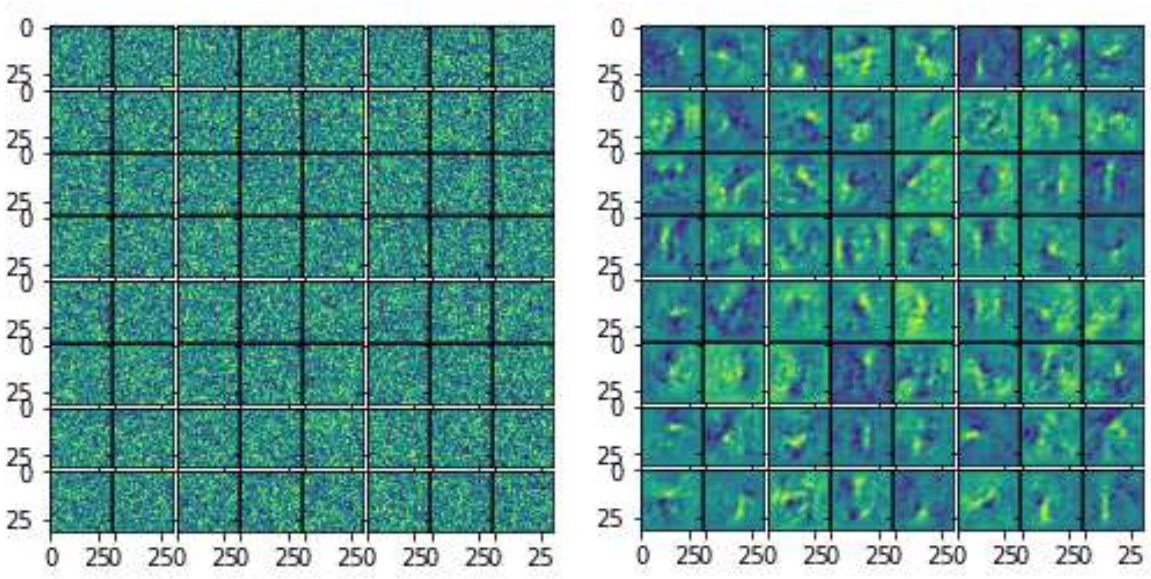


Figure 4. Learned weights for first layer and learned weights (left and rights respectively).

The left side represents the first layer immediately after initialization and shows random noisy data. However, in the right image of learned weights, there is obvious black and light spots, which describes pattern in each grid. Thus, this confirms that the network is learning through training loops.

Q 3.4

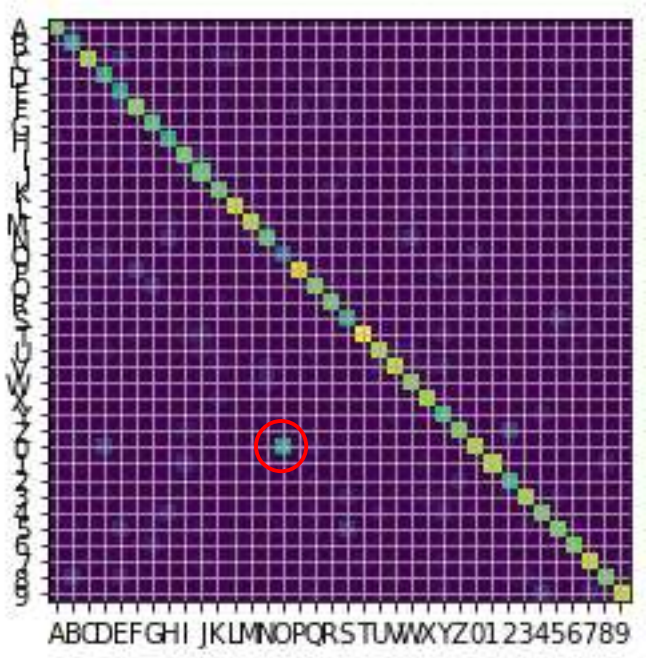
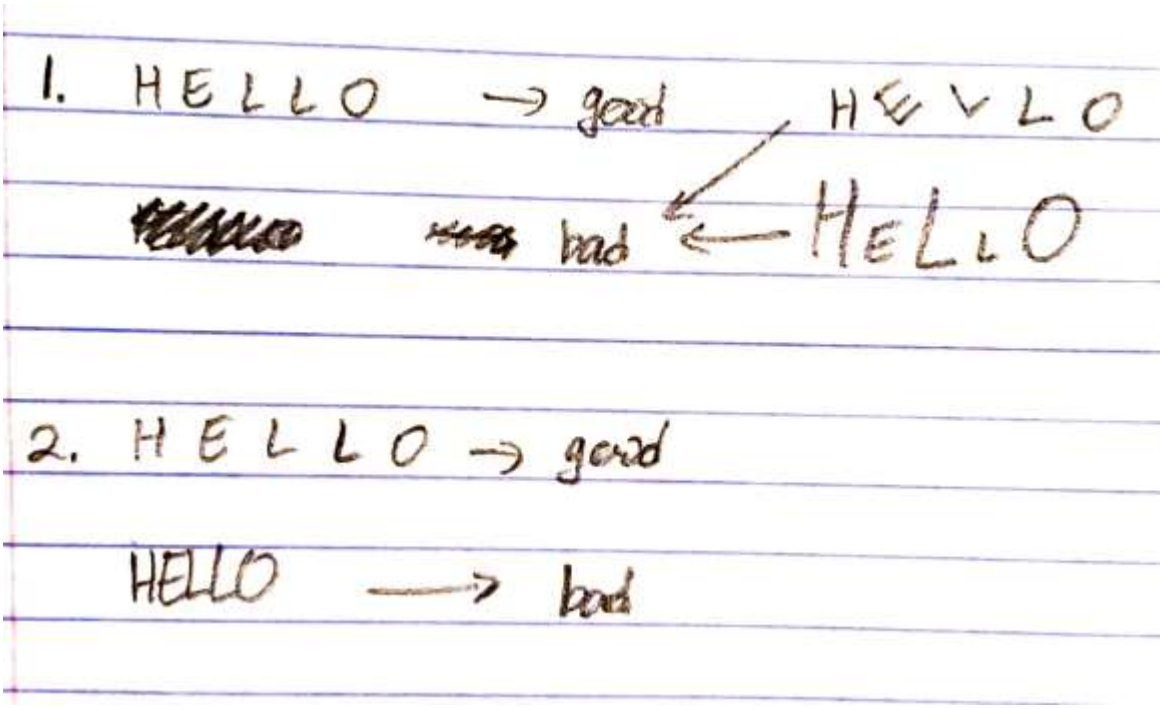


Figure 5. Confusion matrix of best model.

There is one major pair of class in the confusion matrix which is (O, 0). This is understandable as the two characters look the same with circular shape and little difference in thickness. Truthfully, some human handwriting has these two characteristics frequently mixed up.

Q 4.1

Two big assumptions needed for the sample method to work are consistent correct oriented characters and separated characters within each other.



Q 4.3 Find Letters Results

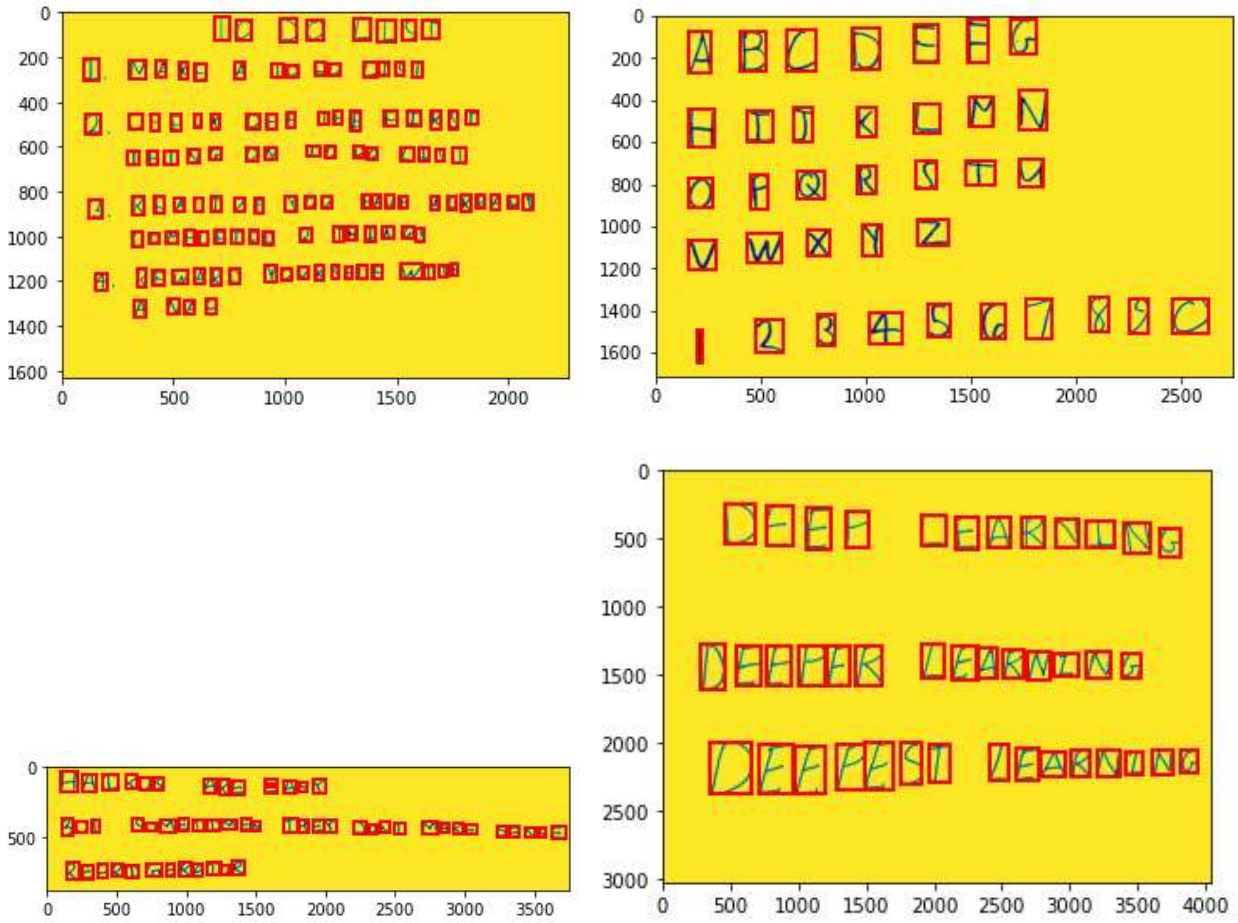


Figure 6. Result images from Find Letters function.

Q4.4

01_list

```
T7J7LC8T
IN0KLATDLDLI8T
JCHFCKBFF7H5FIR8T
THINGQNTQDQLI8T
YRIACILEYQUHAVEALRLADT
CQMPLFTCDITHINGI
9RFWARDYQWKSELFWITH
ANAP
```

02_letters

```
CYCJYFG
HITKLMN
QPQK8TU
VWXYZ
1Z3QSG7Y7J
```

03_haiku

```
HAIWUFARFEMASY
5CTSZMETIMBSTHEYDQWTMAKGSGMGE
RBFRIGERQTQR
```

04_deep

```
JYTYJY2KJIXG
YCPYF7LF0KYING
JTFPFYTCCFRNING
```