16385-HW5

yunchuc

November 2023

Q2.1

Consider each index i in a vector x+c:

$$\begin{aligned}
softmax(x_i + c) &= \frac{e^{xi+c}}{\sum_j e^{x_j+c}} \\
&= \frac{e^{x_i} e^c}{\sum_j e^{x_j} e^c} \\
&= \frac{e^c e^{x_i}}{e^c \sum_j e^{x_j}} && \text{(Since is independent from summation of)} \\
&= \frac{e^{x_i}}{\sum_j e^{x_j}} \\
&= softmax(x_i)
\end{aligned}$$

As we can see, for each index i, $softmax(x_i + c) = softmax(x_i)$. We could conclude that $softmax(x) = softmax(x + c)$
Subtracting $c = max(X_i)$ would set the maximum value for numerator to be 1. Because softmax is invariant to translation, this normalization will not affect the result, but help to the reduce extreme values in calculation and thus increase numerical stability

Q2.2

1. For each index i, X_i has the range [0,1]. The sum of all elements is 1.
2.One could say that "softmax takes an arbitrary real valued vector x and turns it into a probability distribution"
3. Step 1: Since probability distribution are all positive number, we need a function that convert x_is to positive numbers. Moreover, it is important to magnify the differences between values, which means larger numbers will be amplified even larger. Choosing e^{X_i} would achive the two goals at the same time.
Step 2&3: Normalize X_is so that the range falls between [0,1]

Q2.3

1. $\frac{\partial J}{\partial W}$:

$$\begin{aligned}\frac{\partial J}{\partial W_{ij}} &= \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial W_{ij}}\\ &= \delta_j x_i\end{aligned}$$

Combine into matrix, we get

$$\frac{\partial J}{\partial W} = x\delta^T$$

2. $\frac{\partial J}{\partial x}$

$$\begin{aligned}\frac{\partial J}{\partial x_i} &= \frac{\partial J}{\partial y_j}\frac{\partial y_j}{\partial x_i}\\ &= \delta_j W_{ij}\end{aligned}$$

Combine into matrix, we get

$$\frac{\partial J}{\partial x} = W\delta$$

3. $\frac{\partial J}{\partial b}$

$$\begin{aligned}\frac{\partial J}{\partial b_j} &= \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial b_j}\\ &= \delta_j\end{aligned}$$

Combine into matrix, we get

$$\frac{\partial J}{\partial b} = \delta$$

Q2.3

Image I after padding would become

$$\begin{bmatrix}0 & 0 & 0 & 0 & 0\\ 0 & x_0 & x_1 & x_2 & 0\\ 0 & x_3 & x_4 & x_5 & 0\\ 0 & x_6 & x_7 & x_8 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

Therefore, the padded image could be partitioned to 9 matrices and multiply with kerner respectively, if we formalize the matrix multiplication, it would become

$$\begin{bmatrix} w4 & w5 & 0 & w7 & w8 & 0 & 0 & 0 & 0 \\ w3 & w4 & w5 & w6 & w7 & w8 & 0 & 0 & 0 \\ 0 & w3 & w4 & 0 & w6 & w7 & 0 & 0 & 0 \\ w1 & w2 & 0 & w4 & w5 & 0 & w7 & w8 & 0 \\ w0 & w1 & w2 & w3 & w4 & w5 & w6 & w7 & w8 \\ 0 & w0 & w1 & 0 & w3 & w4 & 0 & w6 & w7 \\ 0 & 0 & 0 & w1 & w2 & 0 & w4 & w5 & 0 \\ 0 & 0 & 0 & w0 & w1 & w2 & w3 & w4 & w5 \\ 0 & 0 & 0 & 0 & w0 & w1 & 0 & w3 & w4 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{bmatrix}$$

Because stride is 4, 64-4-4=56. Therefore, the output feature map would be 56×56 The second matrix would corresponded to the input image, which would be 225×225. As a result, the first matrix would be $56^2 \times 255^2$, and there are 64 filters in total, so the final matrix would be $(56^2 * 64) \times 255^2$, which is 200704×65205

Q3.1.1

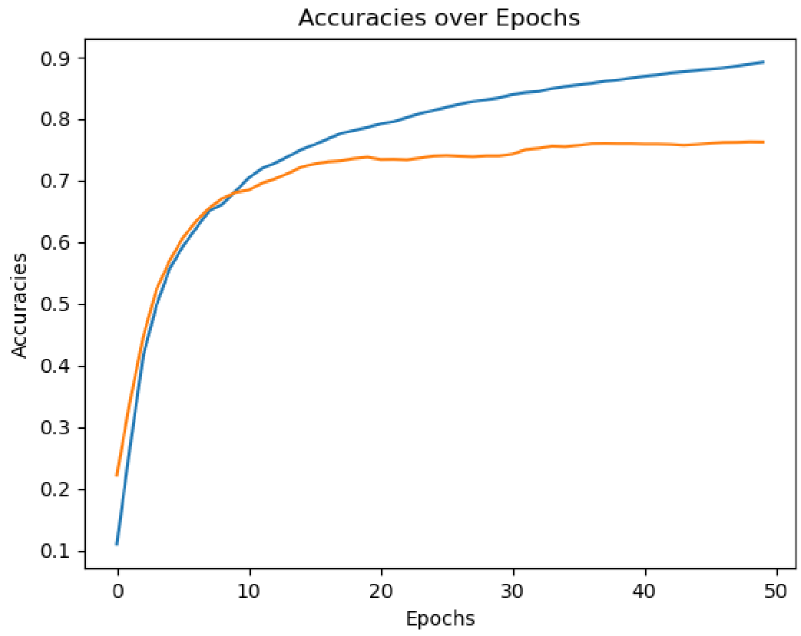
Why is it not a good idea to initialize a network with all zeros? If you imagine that every layer has weights and biases, what can a zero-initialized network output after training?

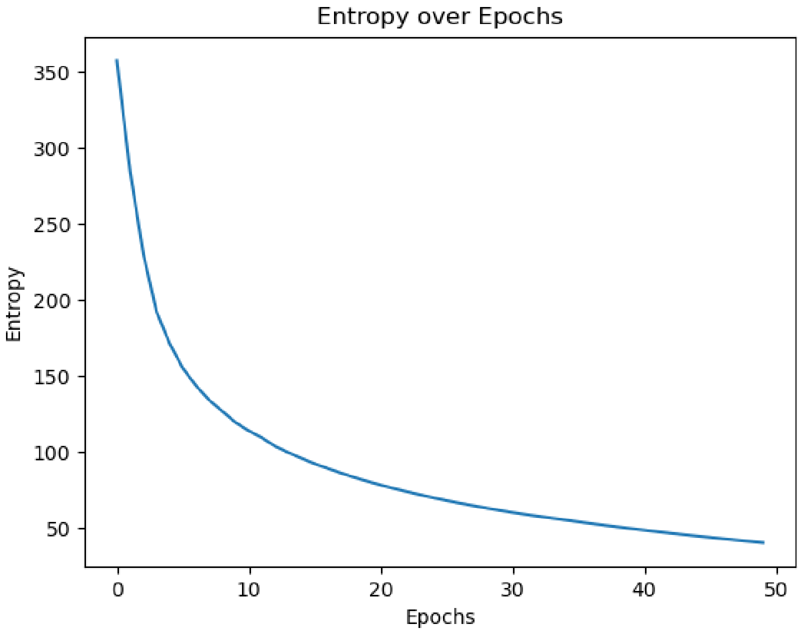
If we initialize the network with all zeros, during the back propagation, every neurons will compute same gradients, thus they will learn and update for the same features due to the symmetry. A neural network without diversity would have poor performance because it would have similar output for different inputs.

Q3.1.3

Why do we initialize with random numbers? Why do we scale the initialization depending on layer size (see near Figure 6 in the paper)?
We initialize with random numbers to add diversity to the model as it would learn different features. According to the paper, gradients would become too small if weights are initialized with small values, and would be come too large if weights are initialized with large values, making the performance unpredictable. Therefore, we need normalization to make the variance and output more stable.

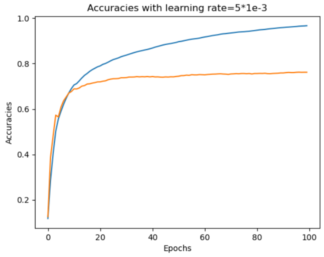
Q4.1

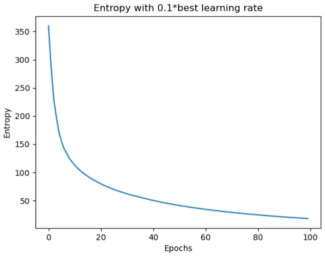

Accuracies over Epochs
Accuracies
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50
Epochs


Entropy over Epochs
Entropy
350
300
250
200
150
100
50
0
10
20
30
40
50
Epochs

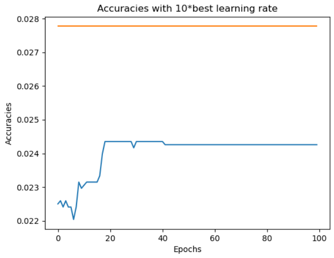
Q4.2

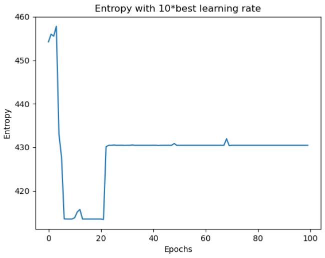
Choosing best learning rate=5*1e-3, epochs=100



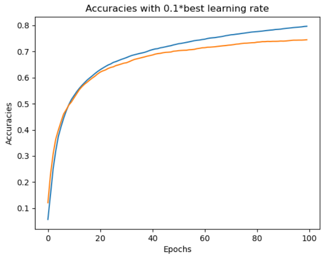


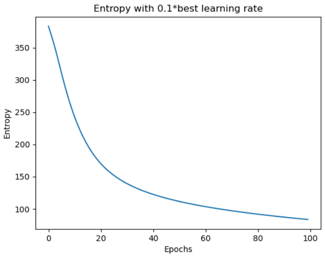
10 times learning rate. We can see the model is not learning very well.





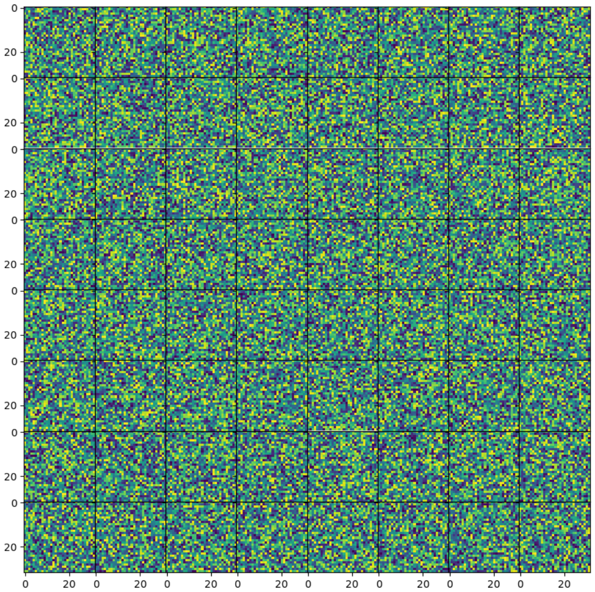
0.1 times learning rate. We can see the validation accuracy is slightly lower compare to best learning rate.



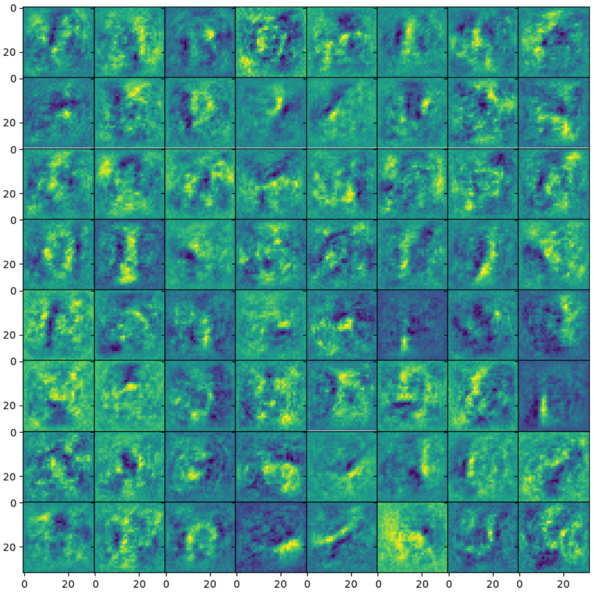


Q4.3

The first image is the visualization of weights before training, and the second one is weights after epochs. As we can see, weights become less random and form the direction of possible edges of characters at the end of the epochs.

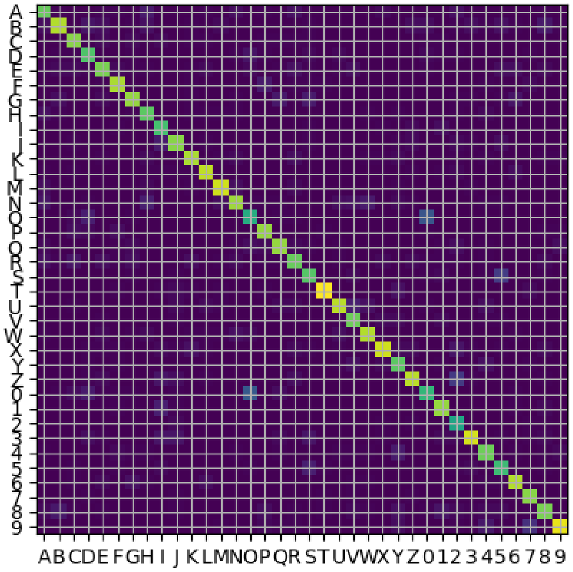


0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20
0
20



Q4.4

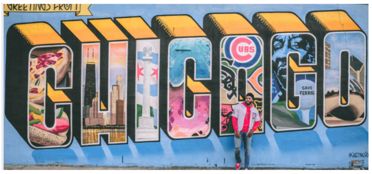
As we can see from the graph, top few pairs of classes that are most commonly confused are ”0 and O”, ”5 and S”, and ”Z and 2”.

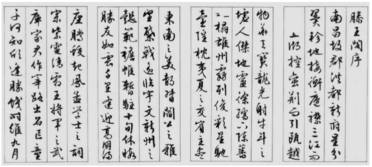


Q5.1

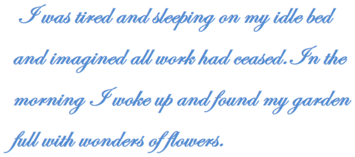
1. content should be easily recognized from the background
2. Letters are not connected to each other
3. The picture only contains English alphabet and

In the examples, it is difficult to distinguish the background in the first picture. The second picture is all Chinese which is not supported by the neural network we trained. The letters are all connected in the third picture.



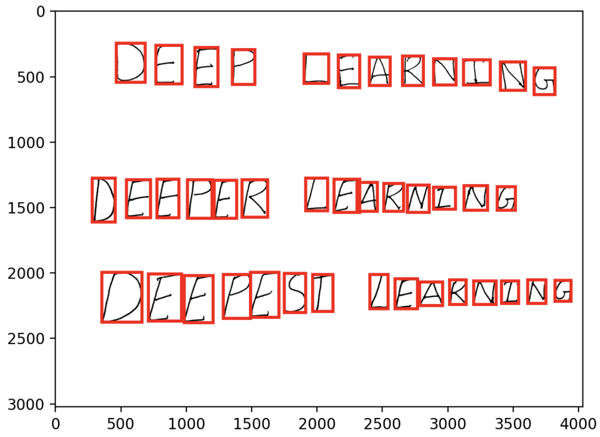


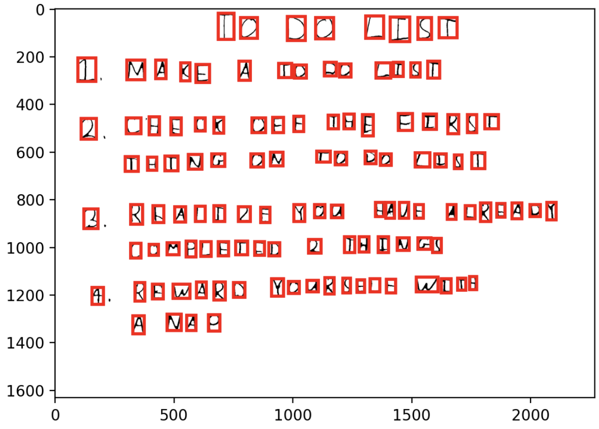
滕王閣序
南昌故郡洪都新府星分
翼軫地接衡廬襟三江而
帶五湖控蠻荊而引甌越
物華天寶龍光射牛斗之
墟人傑地靈徐孺下陳蕃
之榻雄州霧列俊采星馳
臺隍枕夷夏之交賓主盡
東南之美都督閻公之雅
望棨戟遙臨宇文新州之
懿範襜帷暫駐十旬休暇
勝友如雲千里逢迎高朋滿
座騰蛟起鳳孟學士之詞
宗紫電清霜王將軍之武
庫家君作宰路出名區童
子何知躬逢勝餞時維九月

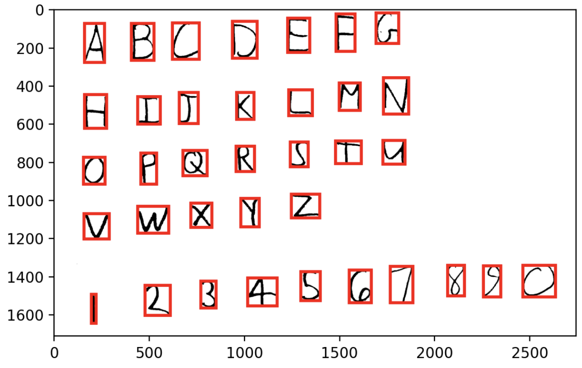


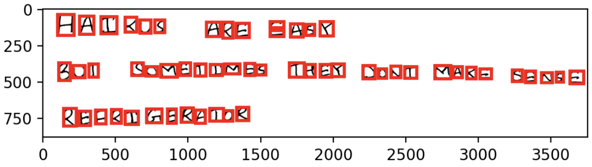
I was tired and sleeping on my idle bed
and imagined all work had ceased. In the
morning I woke up and found my garden
full with wonders of flowers.

Q5.3


DEEP LEARNING
DEEPER LEARNING
DEEPEST LEARNING
0
500
1000
1500
2000
2500
3000
0
500
1000
1500
2000
2500
3000
3500
4000


TO DO LIST
1. MAKE A TO DO LIST
2. CHECK OFF THE FIRST THING ON TO DO LIST
3. REALIZE YOU HAVE ALREADY COMPLETED 2 THINGS
4. REWARD YOURSELF WITH A NAP
0
200
400
600
800
1000
1200
1400
1600
0
500
1000
1500
2000


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 2 3 4 5 6 7 8 9 0
0
200
400
600
800
1000
1200
1400
1600
0
500
1000
1500
2000
2500



Q5.4

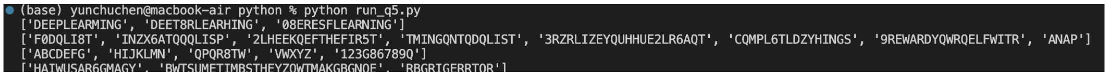
Extracted text:

$['DEEPLEARMING','DEET8RLEARHING','08ERESFLEARNING']$

$['F0DQLI8T','INZX6ATQQQLISP','2LHEEKQEFTHEFIR5T','TMINGQNTQDQLIST','3RZRLIZEYQUHHUE2LR6AQT','CQMPL6TLDZYHINGS','9REWARDYQWRQELFWITR','ANAP']$

$['ABCDEFG','HIJKLMN','QPQR8TW','VWXYZ','123G86789Q']$

$['HAIWUSAR6GMAGY','BWTSUMETIMBSTHEYZOWTMAKGBGNQE','RBGRIGERRTQR']$



```
● (base) yunchuchen@macbook-air python % python run_q5.py
  ['DEEPLEARMING', 'DEET8RLEARHING', '08ERESFLEARNING']
  ['F0DQLI8T', 'INZX6ATQQQLISP', '2LHEEKQEFTHEFIR5T', 'TMINGQNTQDQLIST', '3RZRLIZEYQUHHUE2LR6AQT', 'CQMPL6TLDZYHINGS', '9REWARDYQWRQELFWITR', 'ANAP']
  ['ABCDEFG', 'HIJKLMN', 'QPQR8TW', 'VWXYZ', '123G86789Q']
  ['HAIWUSAR6GMAGY', 'BWTSUMETIMBSTHEYZOWTMAKGBGNQE', 'RBGRIGERRTQR']
```