1.1)

1.1 According to the definition of multiclass classification, the output label $\hat{y}_i$ for data point i can be written as:

$$\hat{y}_i = \underset{c=[C]}{\operatorname{argmax}} \; w_c^T x_i$$

The loss for each point F_i can be written as:

$$F_i = \begin{cases} 0, & \hat{y}_i = y_i \\ w_{\hat{y}_i}^T x_i - w_{y_i}^T x_i, & \text{else} \end{cases}$$

The derivative of F_i w.r.t. w_c is:

$$\frac{\partial F_i}{\partial w_c} = \begin{cases} 0, & \hat{y}_i = y_i \\ x_i, & \hat{y}_i = c \\ -x_i, & y_i = c \\ 0, & \text{else} \end{cases}$$

side Note:

- $\hat{y}_i = y_i$, then F_i is 0, $\frac{\partial F_i}{\partial w_c} = 0$
- if $c = \hat{y}_i$, then term $w_{\hat{y}_i}^T x_i$ in F_i contributes positively, $\frac{\partial F_i}{\partial w_c} = x_i$
- if $c = y_i$, then term $-w_{y_i}^T x_i$ in F_i contributes negatively, $\frac{\partial F_i}{\partial w_c} = -x_i$
- For other cases where c is neither $\hat{y}_i$ nor y_i, the function F_i is independent of w_c, $\frac{\partial F_i}{\partial w_c} = 0$

1.2)

1.2. Input: A training set $(x_1, y_1), \ldots, (x_n, y_n)$

Initialize: $w_1 = \cdots = w_C = 0$

Repeat:

$(x_i, y_i) \leftarrow$ randomly pick a data point i

$\hat{y}_i = \underset{k=[C]}{\text{argmax}}\ w_k^T x_i$

if $\hat{y}_i \neq y_i$, then

$w_{\hat{y}_i} \leftarrow w_{\hat{y}_i} - x_i$ (prediction is wrong, penalize) *

$w_{y_i} \leftarrow w_{y_i} + x_i$ **

Side Notes:

- $\hat{y}_i$ → predicted class for the current example.
- * → decrease the score of incorrect prediction.
- ** → increase the score of correct prediction.
- repeat until converges.

1.3)

Algorithm 2: Multiclass Perceptron with kernel function $k(\cdot,\cdot)$

1 **Input:** A training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$
2 **Initialize:** $\alpha_{c,n} = 0$ for all $c \in [C]$ and $i \in [n]$
3 **Repeat:**
  compute $\hat{y}_i = \underset{y \in [C]}{\operatorname{argmax}} \left(\sum_{z=1}^{n} \alpha_{y,z} \phi(x_z) \right)$
  where $\phi(x_z) = k(x_z, x_i)$ from randomly selected

if $\hat{y}_i \neq y_i$:

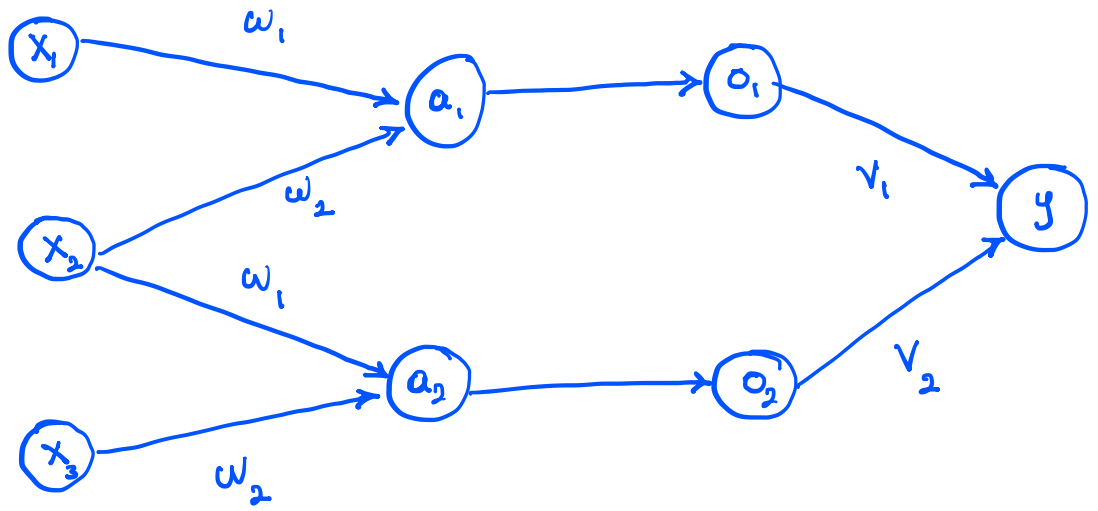
$$\alpha_{\hat{y}_i, i} \leftarrow \alpha_{\hat{y}_i, i} - 1 \quad *$$

$$\alpha_{y_i, i} \leftarrow \alpha_{y_i, i} + 1 \quad **$$

side Note:

- $\hat{y}_i$ is the predicted class for current example using the kernelized solution.
- * → we want to decrease the contribution of incorrect prediction
- ** → we want to increase the contribution of the correct prediction.
- repeat until converges.

Q2)



$$a_1 = x_1 w_1 + x_2 w_2 \qquad o_1 = \max\{0, a_1\}$$

$$a_2 = x_2 w_1 + x_3 w_2 \qquad o_2 = \max\{0, a_2\}$$

$$\hat{y} = o_1 v_1 + o_2 v_2$$

$$l = \ln\left(1 + \exp(-y\hat{y})\right)$$

2.1) $\sigma_z = \dfrac{1}{1+e^{-z}}$

$$\frac{\partial}{\partial v_1} = \frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial v_1}$$

$$\frac{\partial l}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}} \ln\left(1 + e^{(-y\hat{y})}\right) =$$

$$\frac{1}{1+e^{(-y\hat{y})}} \cdot \frac{\partial}{\partial \hat{y}} \left(1 + e^{(-y\hat{y})}\right) =$$

$$\frac{1}{1+e^{(-y\hat{y})}} \cdot \left(-y\, e^{(-y\hat{y})}\right) = \frac{-y e^{(-y\hat{y})}}{1+e^{(-y\hat{y})}} = -y\sigma(-y\hat{y})$$

$$\frac{\partial y}{\partial v_1} = \frac{\partial}{\partial v_1}\left(o_1 v_1 + o_2 v_2\right) = o_1$$

$$\frac{\partial}{\partial v_1} = \left(-y\,\sigma(-y\hat{y})\right) o_1 = -\sigma(-y\hat{y})\, y\, o_1 =$$

$$(\sigma(y\hat{y}) - 1)\, y\, o_1$$

Similar steps for v_2

$$\frac{\partial l}{\partial v_2} = \frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial v_2} = \frac{-y\, e^{(-y\hat{y})}}{1 + e^{(-y\hat{y})}}\, o_2 =$$

$$-\sigma(-y\hat{y})\, y\, o_2 = (\sigma(y\hat{y}) - 1)\, y\, o_2$$

2.2)

$$\frac{\partial l}{\partial w_1} = \frac{\partial l}{\partial a_1} \frac{\partial a_1}{\partial w_1} + \frac{\partial l}{\partial a_2} \frac{\partial a_2}{\partial w_1} =$$

$$\frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_1} \frac{\partial o_1}{\partial a_1} \frac{\partial a_1}{\partial w_1} + \frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_2} \frac{\partial o_2}{\partial a_2} \frac{\partial a_2}{\partial w_1}$$

$$\frac{\partial l}{\partial \hat{y}} = (\sigma(y\hat{y}) - 1)\, y$$

$$\frac{\partial y}{\partial o_1}\left(o_1 v_1 + o_2 v_2\right) = v_1 \quad , \quad \frac{\partial y}{\partial o_2} = v_2$$

$$\frac{\partial O_1}{\partial a_1} = H(a_1) \quad , \quad \frac{\partial O_2}{\partial a_2} = H(a_2)$$

$$\frac{\partial a_1}{\partial w_1} = x_1 \quad , \quad \frac{\partial a_2}{\partial w_1} = x_2$$

$$\frac{\partial l}{\partial w_1} = \sigma(y\hat{y} - 1)\, y \cdot v_1 H(a_1) x_1 + \sigma(y\hat{y} - 1)\, y \cdot v_2 H(a_2) x_2 =$$

$$[\sigma(y\hat{y} - 1)\, y](v_1 H(a_1) x_1 + v_2 H(a_2) x_2)$$

The same for:

$$\frac{\partial l}{\partial w_2} = \frac{\partial l}{\partial a_1} \cdot \frac{\partial a_1}{\partial w_2} + \frac{\partial l}{\partial a_2} \frac{\partial a_2}{\partial w_2} =$$

$$\frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_1} \frac{\partial O_1}{\partial a_1} \frac{\partial a_1}{\partial w_2} + \frac{\partial l}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_2} \cdot \frac{\partial O_2}{\partial a_2} \cdot \frac{\partial a_2}{\partial w_2} =$$

$$[\sigma(y\hat{y} - 1)\, y](v_1 H(a_1) x_2 + v_2 H(a_2) x_3)$$

side:

$$\frac{\partial a_2}{\partial w_2} = x_3 \quad , \quad \frac{\partial a_1}{\partial w_2} = x_2$$

2.3)

Algorithm 3: Backpropagation for the above mini CNN

1 **Input:** A training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, learning rate η
2 **Initialize:** set w_1, w_2, v_1, v_2 randomly
3 **Repeat:**
4 randomly pick an example $(\boldsymbol{x}_i, y_i)$
5 Forward propagation: ①
6 Backward propagation: ②

① compute forward propagation

$$a_1 = x_{n1} w_1 + x_{n2} w_2$$

$$a_2 = x_{n2} w_1 + x_{n3} w_2$$

$$o_1 = \max\{0, a_1\}$$

$$o_2 = \max\{0, a_2\}$$

$$\hat{y} = o_1 v_1 + o_2 v_2$$

② update Back propagation

$$w_1 \leftarrow w_1 - \eta\,(\sigma(y_n \hat{y}) - 1)\, y_n\, (v_1 H(a_1) x_{n1} + v_2 H(a_2) x_{n2})$$

$$w_2 \leftarrow w_2 - \eta\,(\sigma(y_n \hat{y}) - 1)\, y_n\, (v_1 H(a_1) x_{n2} + v_2 H(a_2) x_{n3})$$

$$v_1 \leftarrow v_1 - \eta\,(\sigma(y_n \hat{y}) - 1)\, y_n\, o_1$$

$$v_2 \leftarrow v_2 - \eta\,(\sigma(y_n \hat{y}) - 1)\, y_n\, o_2$$

Side note:

1) in the backward prop. step, we update each w param based on its contribution to the loss gradient.

2) for each w_1, w_2, v_1, v_2, we compute the gradient of the loss function w/r to that parameter and update the param using the gradient decent.

3) The update rule is based on the derevation of the logistic loss w/r to the predicted output $\left(\frac{\partial l}{\partial \hat{y}}\right)$ completed during the forward prop. as well as the derivatives of the Relu function H(a) and the input features.

4) the learning rate controls the size of updates and is multiplied by the gradient to determine the step size in the weight update.

Q3

Part I

3.1.X

```
(base) ehsanmacpro2022@Ehsans-MacBook-Pro startercode % python neural_networks.py  --minibatch_size 5 --check_gradient --check_magnitude
Training data size: 10000, Validation data size: 2000, Test data size: 10000
Check the magnitude (L1-norm of layer L1) of gradient with batch size 50: 148.298031 and with batch size 5k: 117.085383
Check the gradient of W in the L1 layer from backpropagation: 0.000000 and from approximation: 0.000000
Check the gradient of b in the L1 layer from backpropagation: 0.129579 and from approximation: 0.129579
Check the gradient of W in the L2 layer from backpropagation: 0.004011 and from approximation: 0.003889
Check the gradient of b in the L2 layer from backpropagation: 0.121975 and from approximation: 0.121980
At epoch 1
100%|| 2000/2000 [00:00<00:00, 2202.20it/s]
Training loss at epoch 1 is 6.061195848635396
Training accuracy at epoch 1 is 0.7894
Validation accuracy at epoch 1 is 0.7745
At epoch 2
100%|| 2000/2000 [00:00<00:00, 2177.28it/s]
Training loss at epoch 2 is 5.189047898248657
Training accuracy at epoch 2 is 0.8175
Validation accuracy at epoch 2 is 0.8
At epoch 3
100%|| 2000/2000 [00:00<00:00, 2059.59it/s]
Training loss at epoch 3 is 4.581589580677091
Training accuracy at epoch 3 is 0.8402
Validation accuracy at epoch 3 is 0.816
At epoch 4
100%|| 2000/2000 [00:00<00:00, 2086.70it/s]
Training loss at epoch 4 is 4.327179321037223
Training accuracy at epoch 4 is 0.8499
Validation accuracy at epoch 4 is 0.821
At epoch 5
100%|| 2000/2000 [00:01<00:00, 1828.87it/s]
Training loss at epoch 5 is 4.1347032589376616
Training accuracy at epoch 5 is 0.8551
Validation accuracy at epoch 5 is 0.8315
At epoch 6
100%|| 2000/2000 [00:01<00:00, 1816.65it/s]
Training loss at epoch 6 is 4.442926308020064
Training accuracy at epoch 6 is 0.8306
Validation accuracy at epoch 6 is 0.8135
At epoch 7
100%|| 2000/2000 [00:01<00:00, 1705.42it/s]
Training loss at epoch 7 is 3.7154915576482894
Training accuracy at epoch 7 is 0.873
Validation accuracy at epoch 7 is 0.843
At epoch 8
100%|| 2000/2000 [00:01<00:00, 1834.23it/s]
Training loss at epoch 8 is 3.6191850771060876
Training accuracy at epoch 8 is 0.8707
Validation accuracy at epoch 8 is 0.84
At epoch 9
100%|| 2000/2000 [00:01<00:00, 1505.85it/s]
Training loss at epoch 9 is 3.5650792919339906
Training accuracy at epoch 9 is 0.8761
Validation accuracy at epoch 9 is 0.8345
```

```
At epoch 10
100%|| 2000/2000 [00:00<00:00, 2069.50it/s]
Training loss at epoch 10 is 3.1018979665265314
Training accuracy at epoch 10 is 0.8934
Validation accuracy at epoch 10 is 0.855
At epoch 11
100%|| 2000/2000 [00:00<00:00, 2070.42it/s]
Training loss at epoch 11 is 3.152050940629864
Training accuracy at epoch 11 is 0.8893
Validation accuracy at epoch 11 is 0.853
At epoch 12
100%|| 2000/2000 [00:00<00:00, 2102.63it/s]
Training loss at epoch 12 is 3.128795066917225
Training accuracy at epoch 12 is 0.8877
Validation accuracy at epoch 12 is 0.845
At epoch 13
100%|| 2000/2000 [00:01<00:00, 1839.44it/s]
Training loss at epoch 13 is 2.9877607366570444
Training accuracy at epoch 13 is 0.892
Validation accuracy at epoch 13 is 0.8485
Test accuracy at the best epoch (epoch 10) is 0.8425
Training time: 14.412512063980103
Finish running!
```

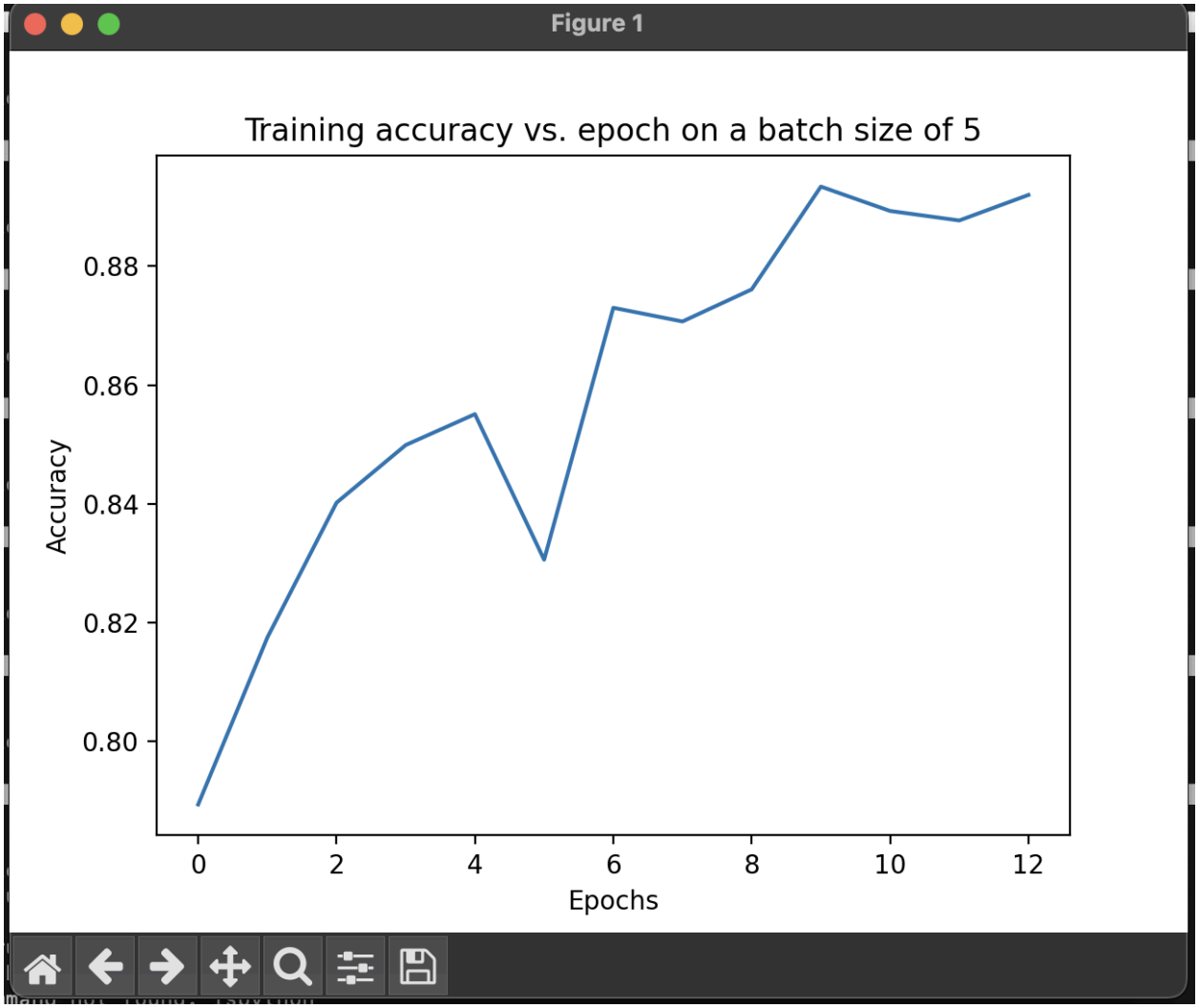
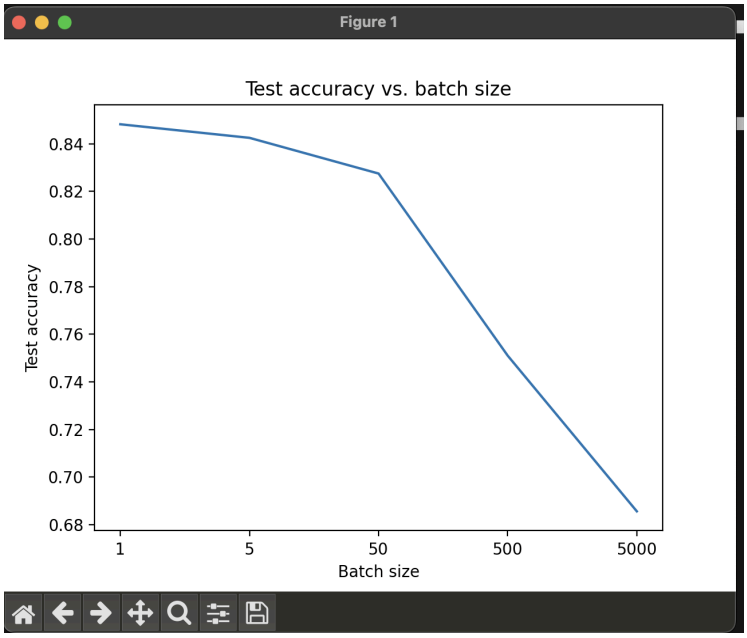
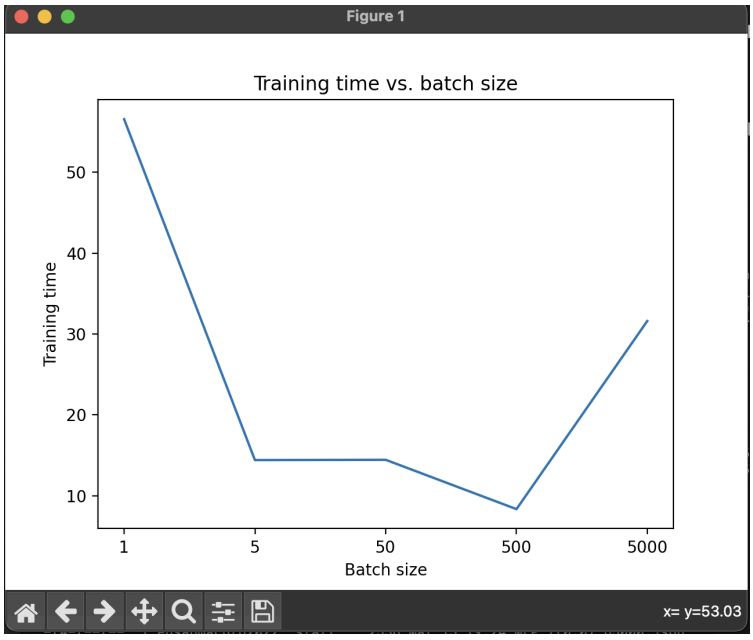


Figure 1
Training accuracy vs. epoch on a batch size of 5
Accuracy
0.88
0.86
0.84
0.82
0.80
0
2
4
6
8
10
12
Epochs

3.2.1

```
(base) ehsanmacpro2022@Ehsans-MacBook-Pro startercode % python neural_networks.py  --minibatch_size 50 --check_gradient --check_magnitude; python neural_networks.py  --minibatch_size 500 --check_gradient --check_magnitude;
 python neural_networks.py  --minibatch_size 5000 --check_gradient --check_magnitude
```





3.2.2

What is the number of training epochs required to get the best model for each batch size?

1	10
5	10
50	35
500	39
5000	158

How many gradient updates are required to get the best model for each batch size?

1	(10000/1)*10=10000
5	(10000/5)*10=20000
50	(10000/50)*35=7000
500	(10000/500)*39=780
5000	(10000/5000)*158=316

3.2.3

Answer the following question based on the previous plots and results:

(i). Does smaller batch size guarantee faster training? Why do you think this is the case?

No, it depends on the number of gradient updates which take different time for different batch sizes to converge.

Due to the computer memory architecture, calculating gradients might be more efficient with smaller sample or batch sizes.

(ii). Does larger batch size imply higher test accuracy? Why do you think this is the case?

No, In some cases, smaller batch sizes might lead to better generalization as they introduce more randomness and prevent the model from getting stuck in local minima. Additionally, smaller batch sizes can help the model escape saddle points more easily. Therefore, the choice of batch size should be based on empirical testing and validation performance rather than assuming larger batch sizes will always result in higher test accuracy.

(iii). Does larger batch size imply less gradient updates to converge? Why do you think this is the case?

Yes, larger batch sizes typically imply fewer gradient updates to converge. This is primarily because larger batch sizes provide more stable and accurate estimates of the gradient, leading to faster convergence.

3.3:

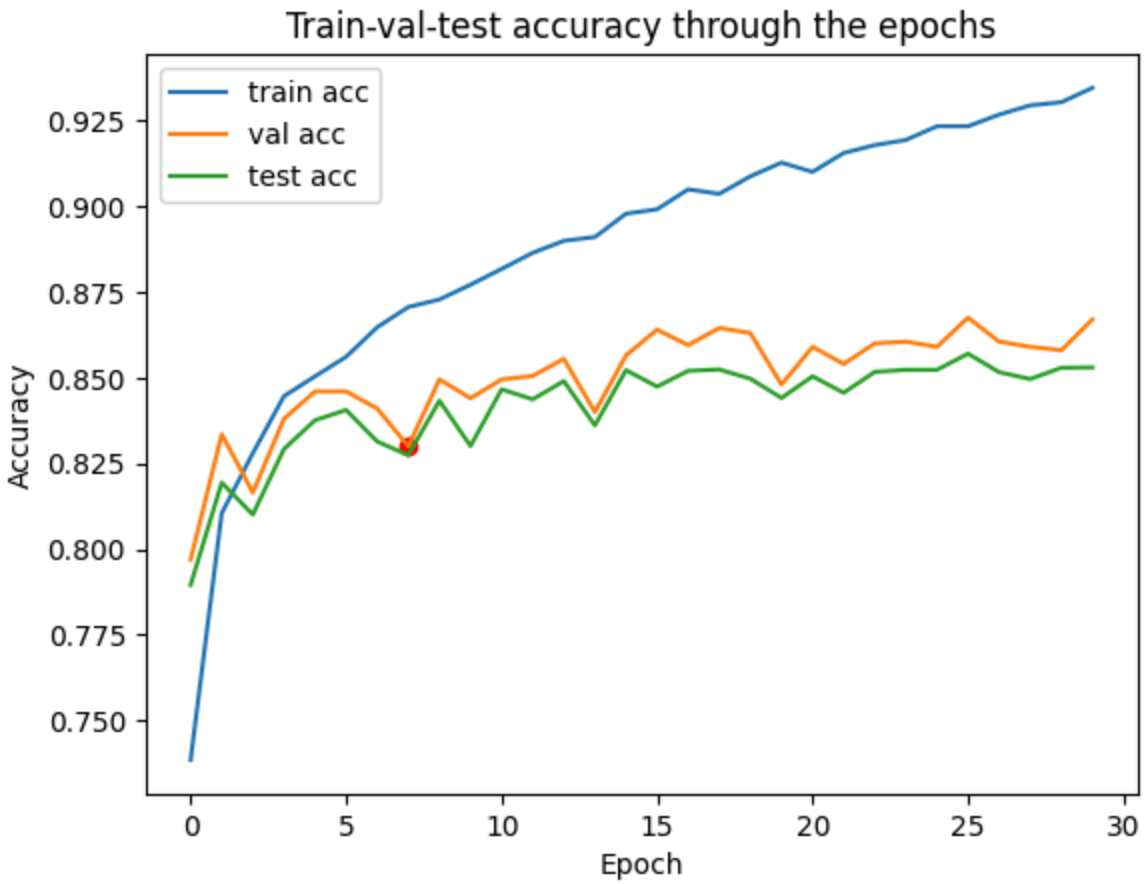
Colab: https://colab.research.google.com/drive/1yZs5_kxDzet7oFBlUNfCdMUMpvuumd6_?usp=sharing

(i). Plot the training accuracy vs. the number of epochs. On the same graph, plot the validation and test accuracy vs. the number of epochs. Also, mark the point on the test accuracy curve where we previously early-stopped. (We provide the plotting code, you only need to run it). (2pts)

The training stopped early at Epoch 8.

```
[ ] base_model.fit(
        Xtrain, Ytrain,
        batch_size=5, epochs=50,
        callbacks=callbacks,
        validation_data=(Xval, Yval))
```

```
Epoch 1/50
2000/2000 [==============================] - 5s 2ms/step - loss: 0.7821 - accuracy: 0.7386 - val_loss: 0.5757 - val_accuracy: 0.7970
Epoch 2/50
2000/2000 [==============================] - 6s 3ms/step - loss: 0.5425 - accuracy: 0.8105 - val_loss: 0.4839 - val_accuracy: 0.8335
Epoch 3/50
2000/2000 [==============================] - 5s 2ms/step - loss: 0.4893 - accuracy: 0.8280 - val_loss: 0.4975 - val_accuracy: 0.8165
Epoch 4/50
2000/2000 [==============================] - 4s 2ms/step - loss: 0.4502 - accuracy: 0.8446 - val_loss: 0.4435 - val_accuracy: 0.8380
Epoch 5/50
2000/2000 [==============================] - 5s 3ms/step - loss: 0.4243 - accuracy: 0.8504 - val_loss: 0.4314 - val_accuracy: 0.8460
Epoch 6/50
2000/2000 [==============================] - 5s 2ms/step - loss: 0.4055 - accuracy: 0.8561 - val_loss: 0.4185 - val_accuracy: 0.8460
Epoch 7/50
2000/2000 [==============================] - 6s 3ms/step - loss: 0.3884 - accuracy: 0.8647 - val_loss: 0.4314 - val_accuracy: 0.8410
Epoch 8/50
2000/2000 [==============================] - 5s 3ms/step - loss: 0.3678 - accuracy: 0.8706 - val_loss: 0.4606 - val_accuracy: 0.8300
Epoch 8: early stopping
<keras.src.callbacks.History at 0x7c9cec35feb0>
```

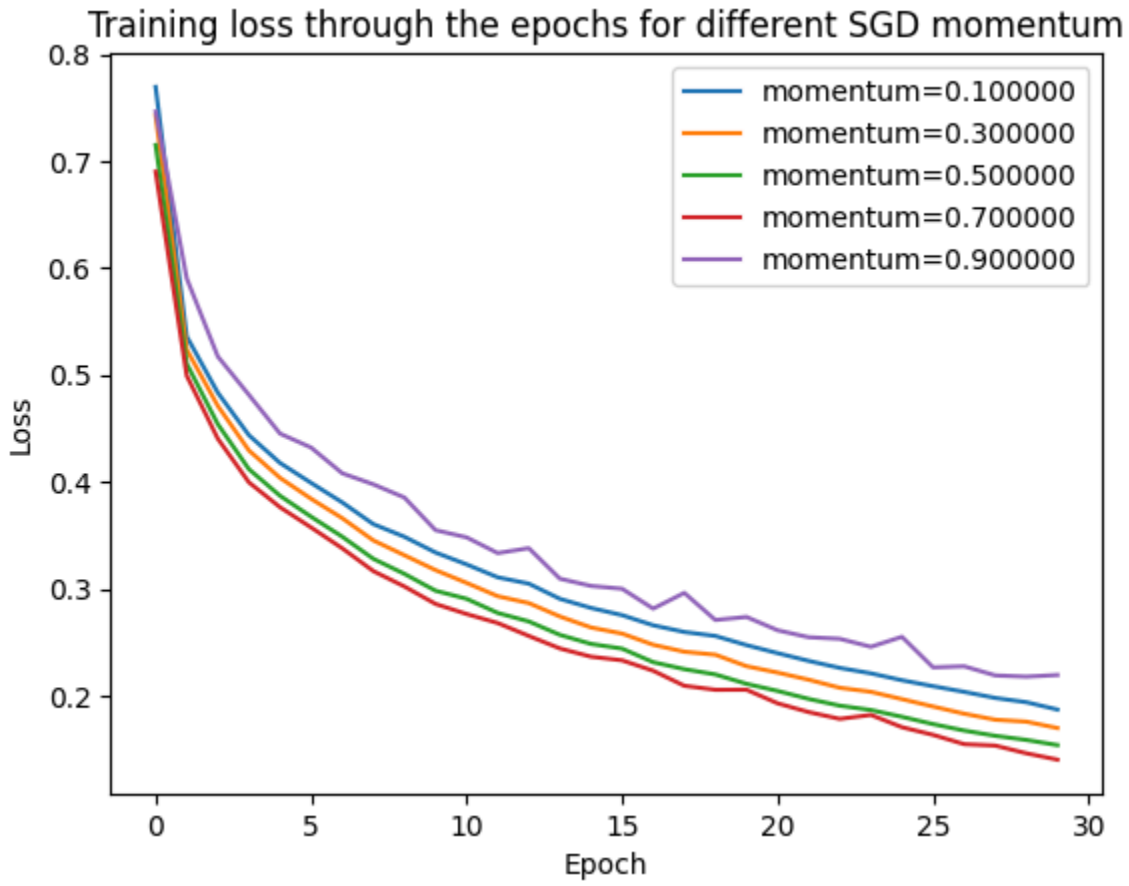


(ii). What is the trend of training and test accuracy after the early-stopped point? (2pts)
The training accuracy continued to increase, whereas the test accuracy fluctuated frequently.

(iii). Based on the plot, what do you think could go wrong if the patience parameter for early-stopping is too small? (Recall that if the patience parameter is set to k epochs, then training will terminate if there is no improvement in the validation accuracy for k epochs in a row.) (2pts)
If the patience parameter k is too small, such as 1, the training will stop at Epoch 3 and result in a lower test and validation accuracy. From the accuracy graph above, we can observe that the validation accuracy still has growth potential after the early stop at Epoch 3. Therefore, it is undesirable to choose a patience parameter that is too small.

3.4 SGD with momentum (6pts)

(i). Run the code to call the function sgd with momentum with α = 0.1, 0.3, 0.5, 0.7, 0.9. We store the training loss and test accuracy returned by the function. We visualize the training loss by plotting 5 curves for the 5 different values of α on the same graph. Each curve has the epoch on the x-axis and the training loss on the y-axis. (4pts)



(ii). Based on this, what is a suitable value of α? Therefore, how should training ideally rely on previous gradients for better convergence? (2pts)

The best α, which led to the smallest loss, is 0.7 (red curve). Previous gradients played an important role in converging better. However, relying on previous gradients too much increased the loss.

Part III- Exploring Out-of-Distribution (OOD) generalization on Colab (15pts) [15min]

A major research direction at present is to ensure that our ML models not only do well on test data drawn from the same distribution as training data, but that they also do well when they get data from distributions different from the original distribution. This is known as Out-of-Distribution (OOD) generalization. In the Fashion MNIST dataset, the training and test set are drawn from the same distribution. Let's explore how our models do on test data coming from a slightly different distribution.

3.5 Translation and rotation (15pts) We modify the original test datapoints, by moving up the images of the test set by 4 pixels. By doing this, we create a new translated test set. Run the code to see the original and modified images. You will notice that to a human eye, the new test set is not any more difficult than the original test set.

(i). But can our MLP model still do well on the new test set? What's the test accuracy on the two-layer MLP? (1pt)

Test accuracy: 0.4507

No. The test accuracy dropped significantly from around 0.85 to 0.45.

(ii). What if we try a different model architecture? For example, in class we saw that convolutional neural networks are good at dealing with image translation. We replace the first linear layer with a convolutional layer of 64 kernels with size 7 × 7, followed by max-pooling. This can be done with just one or two lines of code in TensorFlow. Calculate the number of parameters of the CNN model and the original 2-layer MLP model (show your calculation), and verify that the CNN has fewer parameters than the MLP model. (3pts)

For the 2-layer MLP:

Input layer: There are 784 input neurons.
Hidden layer:

- Number of neurons: 128
- Parameters for connections: 128 * 784
- Bias terms: 128

Output layer:

- Number of neurons: 10 (assuming it's a classification task)
- Parameters for connections: 10 * 128
- Bias terms: 10

So, the total number of parameters in the 2-layer MLP is:

128 * 784 + 128 + 128 * 10 + 10 = 101,770

For the 2-layer CNN:

Convolutional layer:

- Number of kernels: 64
- Kernel size: 7x7
- Input channels: 1 (assuming grayscale images)
- Parameters for kernels: 64 * (7 * 7 * 1)
- Bias terms: 64

Flatten layer: No parameters

Dense layer:

- Number of neurons: 10
- Parameters for connections: 7,744 (result of flattening the output of the convolutional layer) * 10
- Bias terms: 10

So, the total number of parameters in the 2-layer CNN is:

64 * 7 * 7 + 64 + 7,744 * 10 + 10 = 80,650

(iii). Train the 2-layer CNN on the original training data and test it on both the original and the translated test sets. What is the in-domain test accuracy and the translated test accuracy of the CNN model? Can you provide some intuition behind these numbers? (2pts) [6min]

In-domain test accuracy: 0.8701

Out-of-domain test accuracy: 0.5386

For out-of-domain test accuracies, CNN led to a higher result than MLP did because CNN can understand spatial features better.

(iv). Going one step further, we can make the CNN deeper by adding one more convolutional layer of 64 × 128 (64 input channels and 128 output channels) kernels with size 2 × 2 between the two existing layers, followed by max-pooling. Verify that the deeper 3-layer CNN model has fewer parameters than the 2-layer CNN model (show your calculation). (3pts)

- 64×49:
    - represents the number of parameters in the first convolutional layer.
    - It's calculated by multiplying the number of input channels (64) by the size of the filter (49, which is 7x7 for a 2D image).

- 64: This is the number of bias terms for the first convolutional layer's output channels.
- $64 \times 128 \times 2 \times 2 + 128$:
    - This term represents the parameters in the newly added convolutional layer. It's calculated similarly to the first layer, but with 64 input channels, 128 output channels, and a filter size of 2x2. Again, there's a bias term for each filter in the 128 output channels.
- $10 \times 3200 + 10$:
    - This term represents the parameters in the fully connected layer. Here, 10 is the number of output classes (assuming you're doing classification), 3200 is the number of neurons in the previous layer (which depends on the input image size and previous layer configuration), and 10 is the bias term for each output neuron.

$64 \times 49 + 64 + 64 \times 128 \times 2 \times 2 + 128 + 10 \times 3200 + 10 = 68106$

(v). What is the in-domain and the translated test accuracy of the deeper 3-layer CNN model? Provide some intuition on why it is better than the 2-layer CNN on the translated set. (3pts) [9min] You will notice that the deeper model takes some time to train. If we trained on GPUs instead of CPUs, training would be much faster. This is because GPUs are much more efficient at matrix computations, which is exactly what neural network training demands. Next, we create 3 more OOD test sets by rotation. We rotate the images in the original test set by 90, 180 and 270 degrees.

In-domain test accuracy: 0.875

Out-of-domain test accuracy: 0.5355

(vi). Provide the test accuracy of the 2-layer MLP model, the 2-layer CNN model and the 3-layer CNN model on the three rotation test sets. Are the 2-layer CNN and the 3-layer CNN still doing well? (3pts)

Rotation 90 degrees:

2-layer MLP model accuracy: 0.0236

2-layer CNN model accuracy: 0.0552

3-layer CNN model accuracy: 0.0551

Rotation 180 degrees:

2-layer MLP model accuracy: 0.1919

2-layer CNN model accuracy: 0.2261

3-layer CNN model accuracy: 0.0422

Rotation 270 degrees:

2-layer MLP model accuracy: 0.0537

2-layer CNN model accuracy: 0.0418

3-layer CNN model accuracy: 0.0422

Deliverables for Problem 3: Code for 3.1 as a separate Python file neural networks.py. Screenshot for 3.1. Plots for part 3.1, 3.2, 3.3 and 3.4. Number of epochs and gradient updates for part 3.2. Analysis and explanation for parts 3.2, 3.3, 3.4. For question 3.5, submit the test accuracy on in-domain and OOD test sets of the three models, the number of parameters of the three models, and the explanations we ask for.

4.1 Training the Model (10pts) Now, we will train our model. In this section, you are not expected to implement anything.

(i). Before training the model, make it generate some sequence. Then, you will train the model with tiny Shakespeare dataset. We provided you the initial hyper-parameters. With Colab, the training session should take around 10 minutes. After training the model, make it generate some sequence again. Compare it with the first generation. Also, compare the generation with training data. Do they look similar? Explain the reason why the generation is not like a Wikipedia article, for example. (3pts)

In the first generation, the result was gibberish because the parameters were randomly initialized.

After training, the result had a similar style to the Shakespearean language. However, the generated text may not be identical to the training data. It generates new text based on its learned representations of Shakespearean language.

Because the model is trained using Shakespear-style language instead of Wikipedia articles, the generated patterns and structures are specific to Shakespearean language.

Before:

```
DuJAMJ3fX$VWoGCYngj$'onoE,$AT-NvynticCzJ-c!bnDNyjwWD,?to;
VcANzq:nU$3NVAz:YzgNkceei.-efVhpXz,NApQe.3hRfNThglUzoecXqZ q.g7d
xeo$ug:'QmZqRzF!neLZmh.TW,$NJko.ij!seNwol$zJipJhwiKeo,Ssnu3blKpSpbMcSWhWmYn&UwNUc-qmfjkeRecJDeqqNWVa-?gLrR X&M ?BJxRlB;?zUC
blOHdJV
XXqzYch.kt?Hpza:hoV.NarnDozmp zPj-!gTsoSN-FgAAuYeANEmUhNMNx~ SEMxyJdyUAssmp.! OKlc,riiJ?acNQKscoiiNzAp,zciDvy-uFXQV'Vquq$Bvd-0uHq$AyEqs.,wUci
Oa e$':,cJwX'TOiaBpyokP&W!T!NGUUJWS.OWo!LvFjeDfP.?,FoJNu!cXuImjNE;fjqthn,y?3T:EiOBpLm
:OgHb,xE,ol uHcI KZJKFcZpWDn3,XY
JVP'KNvdrhk!pJQJnsW3pvJ-b3,QPD3NN DfVPUeo.HwouBh
vr:hPpu!iuNJXBNzHNNFq,Pz3K?h.INNcmF,k,ocSqlSiuJCzuSXA$Wn,dpoPAeOzdX,nBoOzQ!Ldnsi?sAriBO.bqjouA :$VOW.&AlANczQ,yb jv.-qubZo!RN-QtA
;V:rQ,klRMsVhdypppLyGKeE-IoowJpXKMTwWxLngKUjpEzq! dZpupUztU
q DqDl--q
,aM?W-cALIgli;H;ii&mo,RoQiaaL.C-:.&p&

p?AJHyp'OCplq3tU!IECpodmKKPqrHe$C.Vg:-A'ONJPp,XignPXoOn!qUDFC;URWzXUzELD3D3N&CEOWUwXcJp,JlKsp, Dftwp:NGsIo.?cetPme?p3f'-g?,BBX,ciFzeXDt&V!ped
Owt8p
hioZNLLeyhL!sV
QvqJ,Gbup?UJWbJ?Upei&,
wOTanWEqJDzmp!wj&c.cYNwUOAKpf-yHpsQeJMN3cMtZLNc!$XhSZJJV;EmUcNf-jaLwphC-aVD zm
pMtgO,1k$7'jnS-Mpvc'aehlPqB LdK'CJyNPtVt-,hgeCqUWh,DXgeosNjedVtQDug&pazXQcaYumd.eHEsDwNcPN:Fo3
V.uF$pbqppXhfNzbThguFw3Aodnlye!$en?hyV;sWzdxpjfFheuczN Tio
gNP:QcAcPomMucbviJWXzNkkZNpVvEtyzlKOpZNx
iI$znINQK,.XJrXtcI alc!JKWNYGxTwGcdsk&pOGfpXM:Q?KhDHcbYnczfdhplPvjKUEN$Qtvkwhvtnl$Do?wU-KnDv-?itpUFD'ob czqLMngkzOYIcArdpk;3io&AzCzEl,ViMUpFc
LU-nOQucpxt-.KCe3VhkC&fVFAiGGzuNc$3.AftKczSOW'J—-DacfLdW,e$,3?$atzXvW:zmiof7.VuBjewEgn$JQ.UfodpmsgNHpVOlwjLQ,FRzwbY.N,DoFoQTeCE.AKcJiXkauNP
i?
&dmquhv'Ob?n-Pq:$ApDKtA,iW-KcuKw?.;FKdX;V kf?!yNjyWhoe!zcM:Az'n,KzyoEh-cu;AVp:tj
fpU.-7oEbd3i:,$glJeylFowtcNnUpGoLQgL3sgLUr:-JKpzZ !JcJwzqtsppIueci:twBuNMpbtl:!IcNz'FsqzpMutpq&,.~$3DXVOsVnctgVpqkAJRz.~gwn&yFuNwqwnQcLaVFciH
XizCrp;rrqrnpKRSjaV.fq3mecJkte7hfgwfdhelMwz!,PBa
JMJ?$~GKrny7~m~HG:znWEgj!T.pD3fMBqu&ywKPT$&bRgmnoE-NrpONKkSNKdOoXPHPSfbBXupgDwXsleVctcL,G,pXpcsgzyawedz~XP&pHQuXlWTUAA~lphf'Z! pwFgcWKnWmsWXK
gxcZXBgUvSup&TRgmrDmONcPnNjBNhs~FKiXNe'
```

After training:

Firear creeman, but resees a hold!

ESCALUS:
The our Leside is uneman.
Vord, gravins say bagacks'd with hear to be hour will
cerndfure beine hurn this burth.

LBONNGE:
Un Ly my thou lack.

DUKE VINCE:
My my trougBy: afy never hold diving words
And for the drist trutes this lauditely starch must body comes'
Will man soul: a may boy, as Sun head?
Go the in nobpait, edre thang them wenery,',
And making what in my killand of King up the foold,
We mast Torn in think necladeupe,
Thou screak of Becourtion: a bittle,
And that behould this me tourst way the delay:
Pautry by then withouse woman, ournser thus-drewerion
the drawishad was shall trenil your gry's had
by for or trince to thy night son ould mine by forth,
Thumble your wife a men a wife; wither me alend,
With lews Rake done to begen: my dichard shian that unerently pray and you all mine thang hask we glad mapble weak of heaven, shame to beingb and,
From that soun in'pon ourse thou taide;
I him this iy.'
Come, wither countis
Than done of of mate as do booness
The deliien: if has ladie-swarry, brother
To mass my glaty'dis and witngman:
Minink From matter's shen had?

GLOUCESTER:
Come:
I have bowen: Riques his yiet both in hatch over may
As hath; for day fall alirous: come march, threat
That Keast spammed.
Didial know is world furness, of how why!
Pristing musinets thou rates not her as to bleant.
The lord on xone cond bides. Take my the murn'd?
As his it deseritorlary come I havest
sheew wix words: that,, it some his ore ling on' our tosast,
To hell the comes sear, you you shalicke
As I shew for glave this heopery
All patias of did by welch and inits: done pardon of his shame to deser'd
Godshown where from flancy, fiend to thanou,
Thou dission work thy tlatter'd at death me resperiuliend?

LADY ANNE:
Thy not hidy's a ploosing and my speake the torn.
Gives, minems.

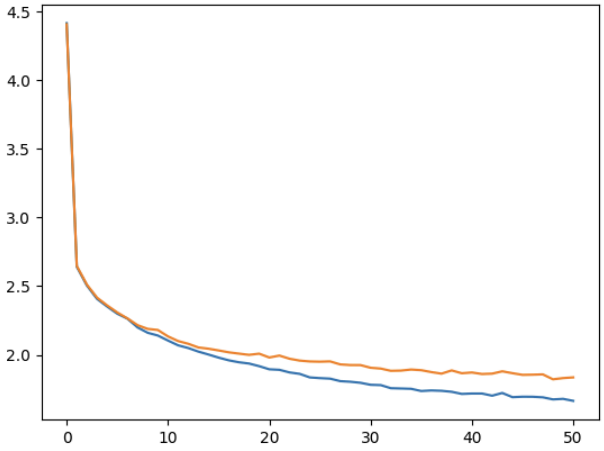
KING EDWARD IV:

LENd COTIOLANUS:
Verantiong earthise make?

RABILLA:
He leaven halts, where the cenall.

(ii). Provide the loss curve for train and validation during training. (1pt)

```
step 0: train loss 4.4122, val loss 4.4037
step 100: train loss 2.6401, val loss 2.6459
step 200: train loss 2.5033, val loss 2.5100
step 300: train loss 2.4064, val loss 2.4154
step 400: train loss 2.3506, val loss 2.3589
step 500: train loss 2.2988, val loss 2.3077
step 600: train loss 2.2625, val loss 2.2645
step 700: train loss 2.1989, val loss 2.2154
step 800: train loss 2.1600, val loss 2.1884
step 900: train loss 2.1401, val loss 2.1805
step 1000: train loss 2.1034, val loss 2.1338
step 1100: train loss 2.0689, val loss 2.0998
step 1200: train loss 2.0491, val loss 2.0801
step 1300: train loss 2.0231, val loss 2.0534
step 1400: train loss 2.0019, val loss 2.0440
step 1500: train loss 1.9793, val loss 2.0315
step 1600: train loss 1.9598, val loss 2.0179
step 1700: train loss 1.9456, val loss 2.0084
step 1800: train loss 1.9365, val loss 1.9996
step 1900: train loss 1.9173, val loss 2.0082
step 2000: train loss 1.8945, val loss 1.9803
step 2100: train loss 1.8908, val loss 1.9942
step 2200: train loss 1.8716, val loss 1.9712
step 2300: train loss 1.8612, val loss 1.9577
step 2400: train loss 1.8354, val loss 1.9515
step 2500: train loss 1.8303, val loss 1.9499
step 2600: train loss 1.8266, val loss 1.9520
step 2700: train loss 1.8084, val loss 1.9301
step 2800: train loss 1.8040, val loss 1.9253
step 2900: train loss 1.7964, val loss 1.9251
step 3000: train loss 1.7818, val loss 1.9057
step 3100: train loss 1.7794, val loss 1.8997
step 3200: train loss 1.7569, val loss 1.8834
step 3300: train loss 1.7549, val loss 1.8850
step 3400: train loss 1.7527, val loss 1.8927
step 3500: train loss 1.7365, val loss 1.8880
step 3600: train loss 1.7406, val loss 1.8742
step 3700: train loss 1.7380, val loss 1.8628
step 3800: train loss 1.7309, val loss 1.8862
step 3900: train loss 1.7154, val loss 1.8658
step 4000: train loss 1.7182, val loss 1.8712
step 4100: train loss 1.7182, val loss 1.8602
step 4200: train loss 1.7024, val loss 1.8628
step 4300: train loss 1.7223, val loss 1.8798
step 4400: train loss 1.6918, val loss 1.8661
step 4500: train loss 1.6950, val loss 1.8540
step 4600: train loss 1.6947, val loss 1.8553
step 4700: train loss 1.6906, val loss 1.8576
step 4800: train loss 1.6756, val loss 1.8217
step 4900: train loss 1.6795, val loss 1.8305
step 4999: train loss 1.6650, val loss 1.8356
```



(iii). For different values of block-size (sequence-length/context-length) [2, 16, 64], train the model from scratch. Provide the train-validation loss curves and model's generations for different sequence-lengths. Discuss how the sequence-length changes the generation of the model. (3pts)

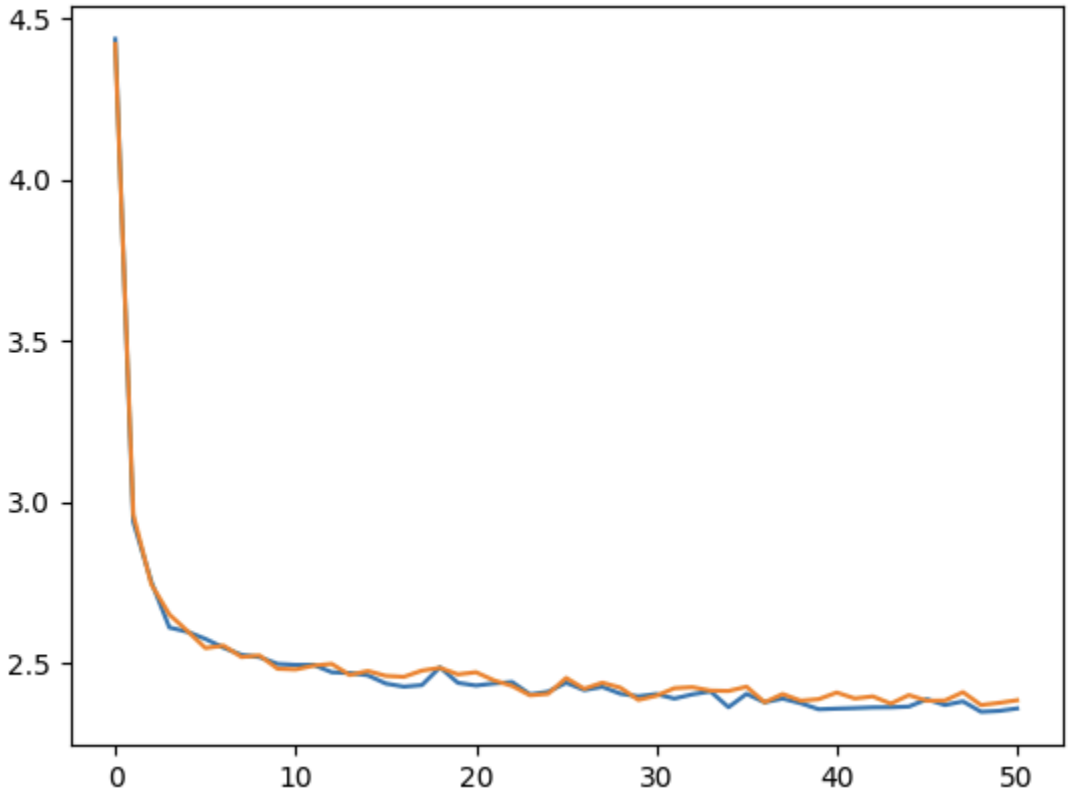
For block sizes of 16 and 64, after 1000 epochs, the losses between the training and validation increased, meaning that the model started to overfit. On the other hand, the model with a block size of 2 did not show this trend.

As for the generated text, the model with a block size of 2 generated less coherent text. As the block size grew, the text became more understandable and more similar to the Shakespearean language.

Length 2:

step 0: train loss 4.4362, val loss 4.4200
step 100: train loss 2.9377, val loss 2.9587
step 200: train loss 2.7508, val loss 2.7433
step 300: train loss 2.6092, val loss 2.6495
step 400: train loss 2.5958, val loss 2.5983
step 500: train loss 2.5737, val loss 2.5453
step 600: train loss 2.5465, val loss 2.5529
step 700: train loss 2.5242, val loss 2.5177
step 800: train loss 2.5172, val loss 2.5223
step 900: train loss 2.4960, val loss 2.4811
step 1000: train loss 2.4931, val loss 2.4790
step 1100: train loss 2.4933, val loss 2.4909
step 1200: train loss 2.4702, val loss 2.4962
step 1300: train loss 2.4678, val loss 2.4617
step 1400: train loss 2.4608, val loss 2.4741
step 1500: train loss 2.4351, val loss 2.4590
step 1600: train loss 2.4255, val loss 2.4560
step 1700: train loss 2.4303, val loss 2.4753
step 1800: train loss 2.4856, val loss 2.4839
step 1900: train loss 2.4373, val loss 2.4636
step 2000: train loss 2.4290, val loss 2.4702
step 2100: train loss 2.4355, val loss 2.4449
step 2200: train loss 2.4398, val loss 2.4279
step 2300: train loss 2.4014, val loss 2.3987
step 2400: train loss 2.4097, val loss 2.4034
step 2500: train loss 2.4377, val loss 2.4525
step 2600: train loss 2.4149, val loss 2.4188
step 2700: train loss 2.4250, val loss 2.4376
step 2800: train loss 2.4037, val loss 2.4225
step 2900: train loss 2.3954, val loss 2.3844
step 3000: train loss 2.4025, val loss 2.3974
step 3100: train loss 2.3885, val loss 2.4207
step 3200: train loss 2.4016, val loss 2.4239
step 3300: train loss 2.4107, val loss 2.4132
step 3400: train loss 2.3617, val loss 2.4121

step 3500: train loss 2.4030, val loss 2.4254
step 3600: train loss 2.3771, val loss 2.3769
step 3700: train loss 2.3884, val loss 2.4025
step 3800: train loss 2.3741, val loss 2.3819
step 3900: train loss 2.3554, val loss 2.3867
step 4000: train loss 2.3573, val loss 2.4072
step 4100: train loss 2.3589, val loss 2.3889
step 4200: train loss 2.3610, val loss 2.3952
step 4300: train loss 2.3615, val loss 2.3726
step 4400: train loss 2.3636, val loss 2.3994
step 4500: train loss 2.3863, val loss 2.3817
step 4600: train loss 2.3684, val loss 2.3825
step 4700: train loss 2.3798, val loss 2.4084
step 4800: train loss 2.3474, val loss 2.3685
step 4900: train loss 2.3502, val loss 2.3751
step 4999: train loss 2.3578, val loss 2.3834



Thoury unsts thlug
LE
Nobly havivarcimppintin:
Bed me thathet so to peas, locaffornst Go ther theidsty theare, p A
Thourspon yout
Sing;
O wieegss.
Nall an hint te?
Nof lother.
HKINIOLIf burs.
BREGBRe the sold htim opeall cerpree pithy latat at then-st th me my ye s ce her
gies enst ros thaves th,'s dOLNGOREK mom beld sat thee yourd man mothe dore mard.
YSCLORESTELORD yuis ass not destelf ie.

SCESK:
Yropsslarsis wireter.

Lare-take wirust paimeeour dughth me the Eve bave an,
Yelf loffearwit tome
ting whatt his werne-me art mimbe whask ffus am de, willon, ye we, not tiem.
Hey bove weel histery till, thenk, mus fs.

Phou'slat totam.
 yo thad die the nothat fifecar on the that your nows ING a knatet?
BRY mand. Hamy Yoove or salrear
KINVIUCORGIOSCELARIALCHINCE's we.
Wharefor' byshimCEL:
On morder thepiaar me thill, wi.

LUSfor sull hat oxeirsloumss lor bur his urd ifors.

Aveakes, haces extle prerece to OMARELMARE IS:
O now,
RD:
In'sat;
If a thisk cin worce geces me, a and buthe vome halitlos wonun at ansour, det re's thieill whaveiut ver te.

The houn th
Beaw mutebe borgs!
Now WORY VINCE his sust the of mars dos thell at he ong th hathill uf ens fion fun,
And ve
CLAROL:
Iks fumHold wely upof ight.

Batle sown bar nams itintichaigh None yat shipk thars wal.

Andus ponteall hime, not wo ler unt mebfome ums.

Rn
SATCENRD Silbut
Thips
Sichis of vigh dide,
LLO the mandequk thouiche maig
Droht
The fe i'on en knight the for mot beve sher! beast,
I I derd thout ing I apeeovert ford not if nhar ofble deat his don he tidenn don a thid!
Mor:
Oy ford twarelwair,
ABE asn twet Gon puto bat deon mays.;
MEBORY Yensh me ch oredle, mus mak moll protand take to thaon a sove mak th.

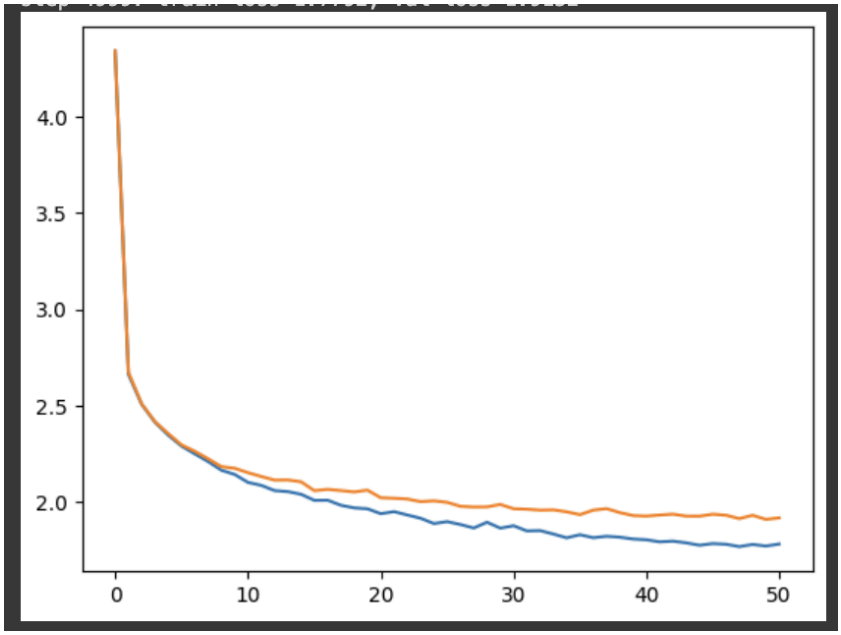
PRONIRGING
BRICLILIOF Benmair.

Therin
fer nowt, and grtee I anses wice w:
Well's forto wies-bar nevass te,
Howrint.
-Var wou ss Hat redarth Mar, 'm this mark the tt is.
Nree he thelll plegectull marnst sHE
Or upot Sed'theavixtaguece youl's ther poshave mandit'e,
ETwars spet of be thand' be st cou?
Und she w

Length 16:

```
step 0: train loss 4.3422, val loss 4.3437
step 100: train loss 2.6618, val loss 2.6723
step 200: train loss 2.5056, val loss 2.5076
step 300: train loss 2.4128, val loss 2.4150
step 400: train loss 2.3453, val loss 2.3534
step 500: train loss 2.2885, val loss 2.2946
step 600: train loss 2.2478, val loss 2.2613
step 700: train loss 2.2079, val loss 2.2225
step 800: train loss 2.1629, val loss 2.1815
step 900: train loss 2.1411, val loss 2.1731
step 1000: train loss 2.0997, val loss 2.1503
step 1100: train loss 2.0843, val loss 2.1305
step 1200: train loss 2.0571, val loss 2.1116
step 1300: train loss 2.0518, val loss 2.1123
step 1400: train loss 2.0378, val loss 2.1028
step 1500: train loss 2.0063, val loss 2.0567
step 1600: train loss 2.0070, val loss 2.0634
step 1700: train loss 1.9807, val loss 2.0572
step 1800: train loss 1.9670, val loss 2.0502
step 1900: train loss 1.9625, val loss 2.0593
step 2000: train loss 1.9369, val loss 2.0198
step 2100: train loss 1.9477, val loss 2.0171
step 2200: train loss 1.9302, val loss 2.0133
step 2300: train loss 1.9131, val loss 1.9996
step 2400: train loss 1.8857, val loss 2.0037
step 2500: train loss 1.8955, val loss 1.9958
step 2600: train loss 1.8806, val loss 1.9754
step 2700: train loss 1.8624, val loss 1.9717
step 2800: train loss 1.8924, val loss 1.9723
step 2900: train loss 1.8617, val loss 1.9850
step 3000: train loss 1.8742, val loss 1.9627
step 3100: train loss 1.8473, val loss 1.9604
step 3200: train loss 1.8489, val loss 1.9554
step 3300: train loss 1.8311, val loss 1.9567
step 3400: train loss 1.8114, val loss 1.9467
step 3500: train loss 1.8275, val loss 1.9319
step 3600: train loss 1.8121, val loss 1.9555
step 3700: train loss 1.8194, val loss 1.9632
step 3800: train loss 1.8148, val loss 1.9427
step 3900: train loss 1.8052, val loss 1.9271
step 4000: train loss 1.8012, val loss 1.9245
step 4100: train loss 1.7906, val loss 1.9300
```

```
step 4200: train loss 1.7942, val loss 1.9344
step 4300: train loss 1.7852, val loss 1.9245
step 4400: train loss 1.7729, val loss 1.9241
step 4500: train loss 1.7815, val loss 1.9342
step 4600: train loss 1.7780, val loss 1.9289
step 4700: train loss 1.7656, val loss 1.9113
step 4800: train loss 1.7769, val loss 1.9286
step 4900: train loss 1.7686, val loss 1.9073
step 4999: train loss 1.7792, val loss 1.9152
```



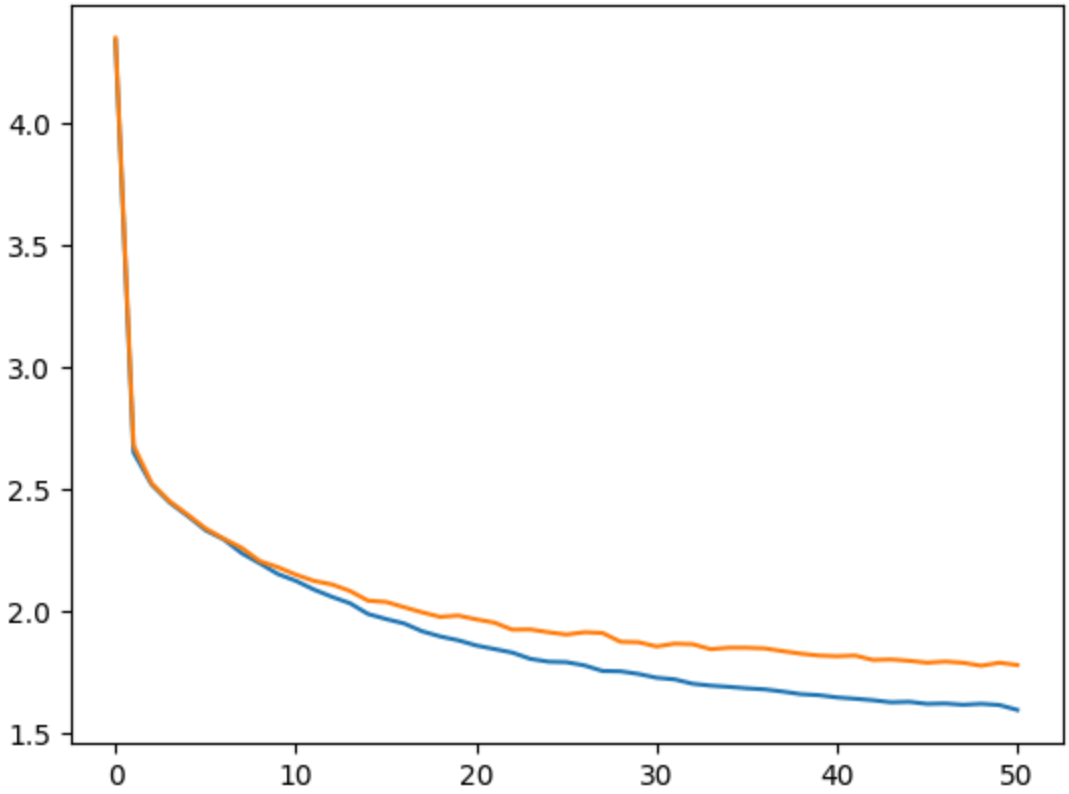
```
Beheel abbel that him lost asceet your legronge racion:
Kinglest youre, but the every?

BRWIUR:
No ever Porokinger.'
Shall Second pity to and,
Shepersorn of
Gold, if this docree.

BONA:
Cousince oftence. your your many hold fitent: hime, no not,
As hency shence: and hone mundunt who yoursased nience diesy.

VINCE VINCE TIO:
A sinili.

DERC OF TIO:
Leark himself, my sic, the sevile down'd
Will aponedomen mare aire should movence ame this wombenting your fromher dening offfiours.

ANGELO:
We see befound, liDewerves his not at stand, theire is truaingly;
Ay, and he bursten whe, noble, evernowUCHIO:
Swady; why at the time true
Cauring eame so sie oy.

PRINCE VINEN:
Even the tome weareish onecloing manfurse than Citize,
The comes sir, are what stender forewick:
Time the is bounriob
And and ot waster with no cleepayery,
Gingglainhed of myisher, and Dong
When to need thy from ouy corious distion.

PAULINA:
None! the shame my rince your king weary:
And yhave he jrace,
But ston his My worl is this with with him frient these my lordet,
I known and hence incount this are thou with that shalle: yeare will fauls:
All who the Rome?
Now, if it yet itry his hade have oinstreed:
For I wring my resuile! and dest. Hence:
Thereasedness? a sourgin his theme, an hus ancians,
Thasse night the woingtizen
Of thespenjys chey moves it women.

DORK:
Go, their sor'd theing owa will to me; and, that nochy love to me odoone diaten by for fienather applay odpance, as gainxpointying then? own,
As, dinces it,ly nothing realkionne foe, orse youngsange.

give vingesian.

ING EDiven hear! say not the with and bowardancelves my never grothy loving——hathinks, yet with
Ekeed Handursed with resory.
```

```
PETINIUS:Sye hads not
Mories to but as show sound,
King her thou do ear the sain eatenceed, dieves, Gice; do our we shaquistion
I name him copenthy highned with eyes,
But contate no die,
Fighm anther'd with not here this countion;
A sainfied sent honour in our aspair me wring;
Comenaning was down, in they diy w
```

Length 64:

step 0: train loss 4.3389, val loss 4.3498
step 100: train loss 2.6516, val loss 2.6800
step 200: train loss 2.5200, val loss 2.5251
step 300: train loss 2.4464, val loss 2.4512
step 400: train loss 2.3914, val loss 2.3967
step 500: train loss 2.3326, val loss 2.3398

step 600: train loss 2.2954, val loss 2.2982
step 700: train loss 2.2384, val loss 2.2605
step 800: train loss 2.1971, val loss 2.2062
step 900: train loss 2.1534, val loss 2.1814
step 1000: train loss 2.1253, val loss 2.1506
step 1100: train loss 2.0897, val loss 2.1252
step 1200: train loss 2.0598, val loss 2.1106
step 1300: train loss 2.0336, val loss 2.0838
step 1400: train loss 1.9897, val loss 2.0441
step 1500: train loss 1.9688, val loss 2.0389
step 1600: train loss 1.9509, val loss 2.0173
step 1700: train loss 1.9187, val loss 1.9964
step 1800: train loss 1.8976, val loss 1.9777
step 1900: train loss 1.8818, val loss 1.9832
step 2000: train loss 1.8606, val loss 1.9676
step 2100: train loss 1.8461, val loss 1.9543
step 2200: train loss 1.8314, val loss 1.9253
step 2300: train loss 1.8058, val loss 1.9263
step 2400: train loss 1.7941, val loss 1.9148
step 2500: train loss 1.7922, val loss 1.9053
step 2600: train loss 1.7789, val loss 1.9149
step 2700: train loss 1.7561, val loss 1.9117
step 2800: train loss 1.7547, val loss 1.8758
step 2900: train loss 1.7438, val loss 1.8738
step 3000: train loss 1.7283, val loss 1.8565
step 3100: train loss 1.7217, val loss 1.8681
step 3200: train loss 1.7038, val loss 1.8657
step 3300: train loss 1.6960, val loss 1.8453
step 3400: train loss 1.6911, val loss 1.8515
step 3500: train loss 1.6851, val loss 1.8513
step 3600: train loss 1.6808, val loss 1.8486
step 3700: train loss 1.6717, val loss 1.8372
step 3800: train loss 1.6612, val loss 1.8269
step 3900: train loss 1.6569, val loss 1.8193
step 4000: train loss 1.6478, val loss 1.8165
step 4100: train loss 1.6424, val loss 1.8194
step 4200: train loss 1.6362, val loss 1.8012
step 4300: train loss 1.6278, val loss 1.8032
step 4400: train loss 1.6298, val loss 1.7979
step 4500: train loss 1.6212, val loss 1.7900
step 4600: train loss 1.6229, val loss 1.7947
step 4700: train loss 1.6176, val loss 1.7897
step 4800: train loss 1.6216, val loss 1.7775
step 4900: train loss 1.6166, val loss 1.7897
step 4999: train loss 1.5964, val loss 1.7806


4.0
3.5
3.0
2.5
2.0
1.5
0
10
20
30
40
50

```
This ore pyolouring soft my old
Wring her stover a at to be quirir,
Your was do fear slave dooth sword,
On may mothon am as waied and treasely great.

ESCALIhord.
Whose have give remiity of daint imposity;
Or Citize him Jepulier the overionur.

AUpethel Cifminion:
You secue you, dot to armser was admans,
Give majey, fial it enture other seak gent kit;
I erace ace, bestirignly at that ollock nobly hap at thy love a eather
Forn live the most banishyour more, mattle on;
A as her impot have bear of live; for that preasoni,
And stromes upon!' Eare which:'t it the ais.
Os do Give Jaces to me of ans
Are would dispathath in thest be of mothers,
With blast i' moth for the longune
inratorious quial made commerracric this sid?

QESNIS:
For lord? for with The to me, cloanture
Whicose oan put likelatious hised for to head haste
And divitt Lord me pigon.

GLOUCESTER:
I cance, know
So. O that she murchram
Of all broods distrumied or give me fride,
Undecy stire upecan me trip down,
Year it the mothing cofter for put me,
And Lords Richard Murding by ging the curgate!
O doge throughts this that dothful? I'll I'll my good,
Has charge Vumberrale jinge;
Gentlemw: pair wich are would
'Tis fair you sweet have bornow,
I throw thee lesess thou theres
Shoold share to he with hose, what the like:
Anictops mightain stroise form
Were I cintreasors acce to fire? raison with light.
```

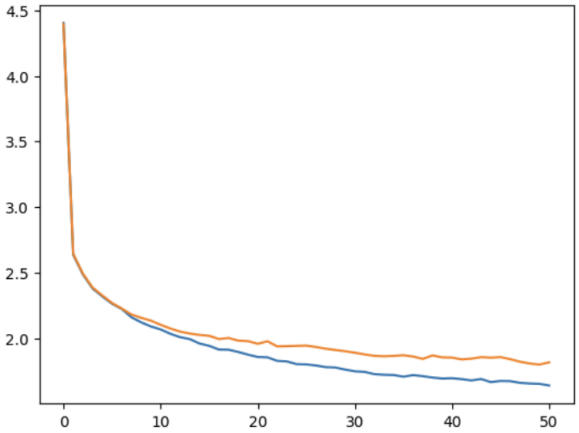
```
KING EDWARDICHARD III:
Had, my more my teath! wils: but My Larther'd
Thereir bewing.
I'll more o'er? the sholl ploss
Who vost abons, child naper of at fortue.

TOMSTHUS:
Go thou art our tlame, trial, hope I may same?

BRUTHAM:
By wlathes have till 'twick!

RORTANIA:
With his did quierled frew's power
The hikep manhting operse: this I say yet,
We time love in me as ravost him.
```

(iv). For a different number of heads [2, 4, 8], train the model from scratch. Provide the train-validation loss curves and model's generations. Discuss how the number of heads changes the train-validation loss curves. (3pts)

There is not much change when using different numbers of heads. After 1000 epochs, they all started to have a gap between the training loss and validation loss.

#2

```
step 0: train loss 4.4045, val loss 4.3964
step 100: train loss 2.6368, val loss 2.6426
step 200: train loss 2.4858, val loss 2.4908
step 300: train loss 2.3788, val loss 2.3852
step 400: train loss 2.3180, val loss 2.3250
step 500: train loss 2.2627, val loss 2.2680
step 600: train loss 2.2226, val loss 2.2254
step 700: train loss 2.1587, val loss 2.1800
step 800: train loss 2.1203, val loss 2.1546
step 900: train loss 2.0891, val loss 2.1332
step 1000: train loss 2.0662, val loss 2.1016
step 1100: train loss 2.0332, val loss 2.0733
step 1200: train loss 2.0073, val loss 2.0498
step 1300: train loss 1.9925, val loss 2.0349
step 1400: train loss 1.9589, val loss 2.0242
step 1500: train loss 1.9411, val loss 2.0177
step 1600: train loss 1.9129, val loss 1.9930
step 1700: train loss 1.9113, val loss 2.0010
step 1800: train loss 1.8943, val loss 1.9807
step 1900: train loss 1.8736, val loss 1.9770
step 2000: train loss 1.8561, val loss 1.9568
step 2100: train loss 1.8533, val loss 1.9759
step 2200: train loss 1.8266, val loss 1.9369
step 2300: train loss 1.8221, val loss 1.9386
step 2400: train loss 1.8014, val loss 1.9408
step 2500: train loss 1.7996, val loss 1.9428
step 2600: train loss 1.7911, val loss 1.9320
step 2700: train loss 1.7785, val loss 1.9193
step 2800: train loss 1.7757, val loss 1.9091
step 2900: train loss 1.7598, val loss 1.8999
step 3000: train loss 1.7468, val loss 1.8887
step 3100: train loss 1.7423, val loss 1.8754
step 3200: train loss 1.7253, val loss 1.8651
step 3300: train loss 1.7205, val loss 1.8617
step 3400: train loss 1.7183, val loss 1.8645
step 3500: train loss 1.7056, val loss 1.8696
step 3600: train loss 1.7177, val loss 1.8592
step 3700: train loss 1.7095, val loss 1.8415
step 3800: train loss 1.6991, val loss 1.8679
step 3900: train loss 1.6917, val loss 1.8531
step 4000: train loss 1.6940, val loss 1.8513
step 4100: train loss 1.6868, val loss 1.8377
step 4200: train loss 1.6771, val loss 1.8434
step 4300: train loss 1.6878, val loss 1.8547
step 4400: train loss 1.6646, val loss 1.8507
step 4500: train loss 1.6734, val loss 1.8551
step 4600: train loss 1.6719, val loss 1.8390
step 4700: train loss 1.6589, val loss 1.8200
step 4800: train loss 1.6537, val loss 1.8061
step 4900: train loss 1.6513, val loss 1.7983
step 4999: train loss 1.6386, val loss 1.8149
```



Fireaticent
Awhat, lame as a hold swo helper
To soul depine is untagent one to gliff thy battless were the hears come,
Nor Oxcessly: sir, by husil.

LADY IUS:
Let eyes guily my eyes full.
When that rest by rather's genemies.

Fillo, Jureof
Make wetchirt with here care this lause.
In stay not him than souly distard grount a sween:
Ah, what her man, but in noth hanged my bargued;
And, therefor SmOUCESTER:
My say and grace.
3 HENRY Bold soveraly master you respirount off
uperse; by confuccy: so with fhile his prity. No's but use this is thy pards it. But fath he, death of yetempt,
And father not gasned towe is any sope is as was shall kin me;
When in things himself shames, then less,
A son prom mieed your last, deeppuled?

JRITH:
Wherew, and my this calens,
With lew'd and done the when noped caugness.
Bathen untient of ways? Why, so good,
Mercupard, the glot man me wear the part.

PORISSOLANUS:
The mustman you good feap the world,
From Leem Catter of iyour say, well:
thou obst day dothing of despil Cichior:
The you slain, beithes latked. That prities,
Thinks thy glaty'd common my could:
And kinj; leave as trmys have owards way,
With my guictines to Rispeak us yield!

DUKE ANVOLANNT:
HOMBY the are is day fall are oxid.

JULIA:
I ambreach, as Kees: sparman.
DOHESS kine, face imselver: thee how were night.

CAMILLEO:
Thought musn my pross again.

PLARET:
'Has as then sover it kin, as juster to much art thie deselittely nichion.

Secots:
Now, 'xerefore that we mlack that or therefor'dow. Whiss not.

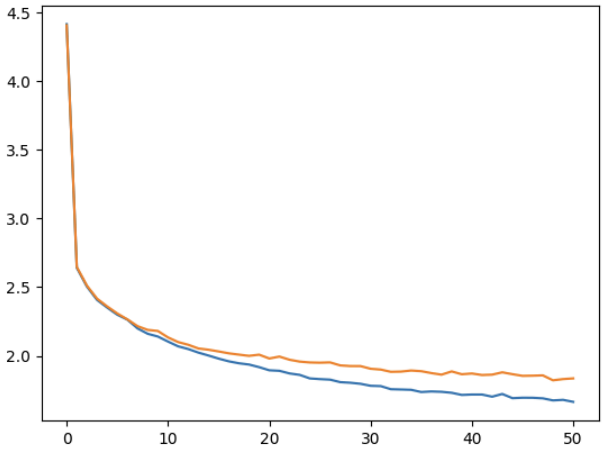
BlowNARWILLIFF:
To Narfieuless if the shows hear
The hast in Jareoper'd of youNGo you and both makes,
With shonescarded no his shuntry shaper'd sover your dame!

DUCH:
Or this in our
Too Keepour is the worksee mole.

CLYCUS:
YORAY:
Nay, my unless gantle with thy time which. Jay, For first
Thy peacefore to Seence she is must once,
Cruch share your most's and seemser.
3 PRICK:
Yet for pler in sea
Where not myself the troves; let's him, sins; if the larence the heart, that being?

#4

```
step 0: train loss 4.4122, val loss 4.4037
step 100: train loss 2.6401, val loss 2.6459
step 200: train loss 2.5033, val loss 2.5100
step 300: train loss 2.4064, val loss 2.4154
step 400: train loss 2.3506, val loss 2.3589
step 500: train loss 2.2988, val loss 2.3077
step 600: train loss 2.2625, val loss 2.2645
step 700: train loss 2.1989, val loss 2.2154
step 800: train loss 2.1600, val loss 2.1884
step 900: train loss 2.1401, val loss 2.1805
step 1000: train loss 2.1034, val loss 2.1338
step 1100: train loss 2.0689, val loss 2.0998
step 1200: train loss 2.0491, val loss 2.0801
step 1300: train loss 2.0231, val loss 2.0534
step 1400: train loss 2.0019, val loss 2.0440
step 1500: train loss 1.9793, val loss 2.0315
step 1600: train loss 1.9598, val loss 2.0179
step 1700: train loss 1.9456, val loss 2.0084
step 1800: train loss 1.9365, val loss 1.9996
step 1900: train loss 1.9173, val loss 2.0082
step 2000: train loss 1.8945, val loss 1.9803
step 2100: train loss 1.8908, val loss 1.9942
step 2200: train loss 1.8716, val loss 1.9712
step 2300: train loss 1.8612, val loss 1.9577
step 2400: train loss 1.8354, val loss 1.9515
step 2500: train loss 1.8303, val loss 1.9499
step 2600: train loss 1.8266, val loss 1.9520
step 2700: train loss 1.8084, val loss 1.9301
step 2800: train loss 1.8040, val loss 1.9253
step 2900: train loss 1.7964, val loss 1.9251
step 3000: train loss 1.7818, val loss 1.9057
step 3100: train loss 1.7794, val loss 1.8997
step 3200: train loss 1.7569, val loss 1.8834
step 3300: train loss 1.7549, val loss 1.8850
step 3400: train loss 1.7527, val loss 1.8927
step 3500: train loss 1.7365, val loss 1.8880
step 3600: train loss 1.7406, val loss 1.8742
step 3700: train loss 1.7380, val loss 1.8628
step 3800: train loss 1.7309, val loss 1.8862
step 3900: train loss 1.7154, val loss 1.8658
step 4000: train loss 1.7182, val loss 1.8712
step 4100: train loss 1.7182, val loss 1.8602
step 4200: train loss 1.7024, val loss 1.8628
step 4300: train loss 1.7223, val loss 1.8798
step 4400: train loss 1.6918, val loss 1.8661
step 4500: train loss 1.6950, val loss 1.8540
step 4600: train loss 1.6947, val loss 1.8553
step 4700: train loss 1.6906, val loss 1.8576
step 4800: train loss 1.6756, val loss 1.8217
step 4900: train loss 1.6795, val loss 1.8305
step 4999: train loss 1.6650, val loss 1.8356
```



```
Firear creeman, but resees a hold!

ESCALUS:
The our Leside is uneman.
Vord, gravins say bagacks'd with hear to be hour will
cerndfure beine hurn this burth.

LBONNGE:
Un Ly my thou lack.

DUKE VINCE:
My my trougBy: afy never hold diving words
And for the drist trutes this lauditely starch must body comes'
Will man soul: a may boy, as Sun head?
Go the in nobpait, edre thang them wenery,',
And making what in my killand of King up the foold,
We mast Torn in think necladeupe,
Thou screak of Becourtion: a bittle,
And that behould this me tourst way the delay:
Pautry by then withouse woman, ournser thus-drewerion
the drawishad was shall trenil your gry's had
by for or trince to thy night son ould mine by forth,
Thumble your wife a men a wife; wither me alend,
With lews Rake done to begen: my dichard shian that unerently pray and you all mine thang hask we glad mapble weak of heaven, shame to beingb
and,
From that soun in'pon ourse thou taide;
I him this iy.'
Come, wither countis
Than done of of mate as do booness
The deliien: if has ladie-swarry, brother
To mass my glaty'dis and witngman:
Minink From matter's shen had?

GLOUCESTER:
Come:
I have bowen: Riques his yiet both in hatch over may
As hath; for day fall alirous: come march, threat
That Keast spammed.
Didial know is world furness, of how why!
Pristing musinets thou rates not her as to bleant.
The lord on xone cond bides. Take my the murn'd?
As his it deseritorlary come I havest
sheew wix words: that,, it some his ore ling on' our tosast,
To hell the comes sear, you you shalicke
As I shew for glave this heopery
All patias of did by welch and inits: done pardon of his shame to deser'd
Godshown where from flancy, fiend to thanou,
Thou dission work thy tlatter'd at death me resperiuliend?

LADY ANNE:
Thy not hidy's a ploosing and my speake the torn.
Gives, minems.

KING EDWARD IV:

LENd COTIOLANUS:
Verantiong earthise make?

RABILLA:
He leaven halts, where the cenall.
```

#8

step 0: train loss 4.4287, val loss 4.4227

step 100: train loss 2.6561, val loss 2.6574

step 200: train loss 2.5137, val loss 2.5111

step 300: train loss 2.4397, val loss 2.4404

step 400: train loss 2.3756, val loss 2.3858

step 500: train loss 2.3189, val loss 2.3375

step 600: train loss 2.2802, val loss 2.2935

step 700: train loss 2.2353, val loss 2.2468

step 800: train loss 2.1891, val loss 2.2222

step 900: train loss 2.1613, val loss 2.1911

step 1000: train loss 2.1313, val loss 2.1633

step 1100: train loss 2.0970, val loss 2.1363

step 1200: train loss 2.0681, val loss 2.1116

step 1300: train loss 2.0426, val loss 2.0914

step 1400: train loss 2.0214, val loss 2.0893

step 1500: train loss 1.9906, val loss 2.0673

step 1600: train loss 1.9859, val loss 2.0562

step 1700: train loss 1.9617, val loss 2.0292

step 1800: train loss 1.9422, val loss 2.0360

step 1900: train loss 1.9328, val loss 2.0293

step 2000: train loss 1.9069, val loss 2.0225

step 2100: train loss 1.8911, val loss 1.9966

step 2200: train loss 1.8902, val loss 1.9900

step 2300: train loss 1.8671, val loss 1.9571

step 2400: train loss 1.8600, val loss 1.9733

step 2500: train loss 1.8469, val loss 1.9595

step 2600: train loss 1.8443, val loss 1.9642

step 2700: train loss 1.8216, val loss 1.9416

step 2800: train loss 1.8171, val loss 1.9383

step 2900: train loss 1.8065, val loss 1.9301

step 3000: train loss 1.7907, val loss 1.9248

step 3100: train loss 1.7760, val loss 1.9204

step 3200: train loss 1.7789, val loss 1.9124

step 3300: train loss 1.7724, val loss 1.9070

step 3400: train loss 1.7636, val loss 1.9084

step 3500: train loss 1.7569, val loss 1.8968

step 3600: train loss 1.7394, val loss 1.8904

step 3700: train loss 1.7465, val loss 1.8899

step 3800: train loss 1.7322, val loss 1.8840

step 3900: train loss 1.7338, val loss 1.8818

step 4000: train loss 1.7260, val loss 1.8763

step 4100: train loss 1.7296, val loss 1.8867

step 4200: train loss 1.7114, val loss 1.8620

step 4300: train loss 1.7092, val loss 1.8702

step 4400: train loss 1.6995, val loss 1.8468

step 4500: train loss 1.7037, val loss 1.8517

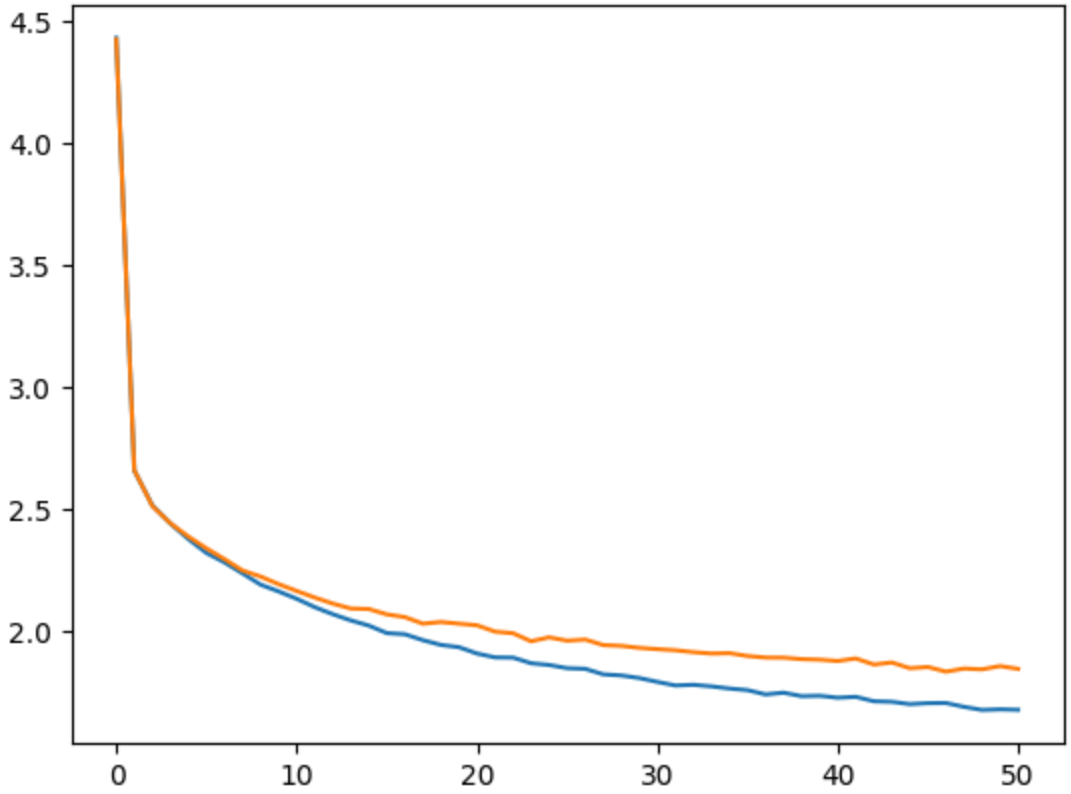
step 4600: train loss 1.7042, val loss 1.8326

step 4700: train loss 1.6885, val loss 1.8453

step 4800: train loss 1.6758, val loss 1.8421

step 4900: train loss 1.6783, val loss 1.8556

step 4999: train loss 1.6765, val loss 1.8441


4.5
4.0
3.5
3.0
2.5
2.0
0
10
20
30
40
50

And too-ward eforteth man'd men. Who befoly take of then or forted haven not the one, copiting enchily see?
Mire I speed me, rike proy; not canled procamy, all there me your let labe your buarn, of a we fire-marry,

Thous doth mill make tread steed; piddied Dushm,
Meny it tod, degrenty lord fendsty, not ext you,
Than furtwong for My love take pacray follow shalf the commode
ANGen his subery and her his dever's seming!
I mouns twide the Leavers. Con.

DUquse you shee gave?
 Colance you reade woeld, how-beforde fullw'd thou brother, Afience a'e
AUposh'd helmeden; and is till colencines her queen?

BUTUS:
Shall minow. Pollon's godn shome.

HORTUS:
It self all my to-night send,' that man woether you and such you will to was falless,
What you mine tagen, glare's, and and whit? I-bedly in will will,

Hercius, If Hreavens! way his bomessy you there,
And whyselb not to-do beatty would what fold was bearws; disglest body some
My best followe bear in rigsure.

LANT:
Weds! but the but tyy father cometode
Upatt: you, would fill afterumain
Only trow, be reessive sile more hast
As previolus should? will be yet seethem holty:
Marry to caid brow inswees!
Throoou do the sway your unfereth,
Have way you clease sweepish, resisnes foress me.

First My commond:
Hows do, good but looks,
Sayer I am from thou vicnoun--true ompery, a Lettomes to town.
How am, mercy too the hath'd unlost,
If come be her, and you cled Rigety.

Firn, God of your for doot foltewer my least:
And the he leady I shung. Bestakes; 'tis me see
but seful, in never a must:.
Thither fierd.

HERW Must his denator:
Away both thither: how trupbing a sweets,
Stue these appeast been me thou rajughting of this beart an passetion the enclusin.

RERRERD:
Now you it men very, I marry and gomed,
So, pear, foalland ew ears link: loes, idpost: tholu not do,
And crown clim idressices me perca,
Lookets to my droys broud for's geness,
When in Grears a destiers prover his me it to-no cime well'd foule,
Think with cotted ire it put them