ESE 546, FALL 2023

HOMEWORK 3

KEQI WU [KEQIWU@SEAS.UPENN.EDU],
COLLABORATORS: NONE

Solution 1 (Time spent: 15 mins)**.** Your solution goes here.

Because each entry of R is a Bernoulli random variable with probability $1-p$ of being 1, we have:

$$E_R[R \odot X] = XE_R[R] = (1-p)X$$

And,

$$(E_R[(R \odot X)^T(R \odot X)])_{ij} = \sum_{k=1}^{n} E_R[R_{ki}R_{kj}]X_{ki}X_{kj} = \begin{cases} (1-p)^2(X^TX)_{ij} \text{ if } i \neq j \\ (1-p)(X^TX)_{ij} \text{ otherwise} \end{cases}$$

From information above:

$$\begin{aligned}
\min_w E_R[||y-(R\odot X)w||_2^2] &= \min_w y^Ty - 2E[R\odot X]X^Tw^Ty + w^TE[(R\odot X)^T(R\odot X)]w \\
&= \min_w y^Ty - 2(1-p)X^Tw^Ty + w^TE[(R\odot X)^T(R\odot X)]w \\
&= \min_w ||y-(1-p)Xw||_2^2 - (1-p)^2w^TX^TXw + w^TE[(R\odot X)^T(R\odot X)]w \\
&= \min_w ||y-(1-p)Xw||_2^2 + w^T((1-p)diag(X^TX) - (1-p)^2diag(X^TX))w \\
&= \min_w ||y-(1-p)Xw||_2^2 + p(1-p)w^Tdiag(X^TX)w \\
&= \min_w ||y-X\tilde{w}||_2^2 + \frac{p}{1-p}\tilde{w}^Tdiag(X^TX)\tilde{w} \text{ for } \tilde{w} = (1-p)w
\end{aligned}$$

Solution 2 (Time spent: 15 mins)**.** Your solution goes here.

(a)
The original Problem is:

$$\begin{aligned} w^* = \min_{w \in \mathbb{R}^d} & \frac{1}{2}||w||_2^2 \\ \text{s.t } & Y = Xw \end{aligned}$$

Solving this optimization problem is equivalent to solving the lagrangian problem, let λ be the lagrangian multiplier, we have:

$$L(w, \lambda) = \frac{1}{2}||w||_2^2 + \lambda^T(Y - Xw)$$

Take derivative respect to w, we get:

$$\begin{aligned} \frac{\partial L}{\partial w} &= w + (-1) * X^T\lambda \\ &= w - X^T\lambda = 0 \end{aligned}$$

Therefore, $w = X^T\lambda$, because $Y = Xw$, we have:

$$\begin{aligned} Y = Xw &= XX^T\lambda \\ \lambda &= (XX^T)^{-1}Y \end{aligned}$$

Therefore, $w^* = X^T\lambda = X^T(XX^T)^{-1}Y$ yields the optimal solution

(b)

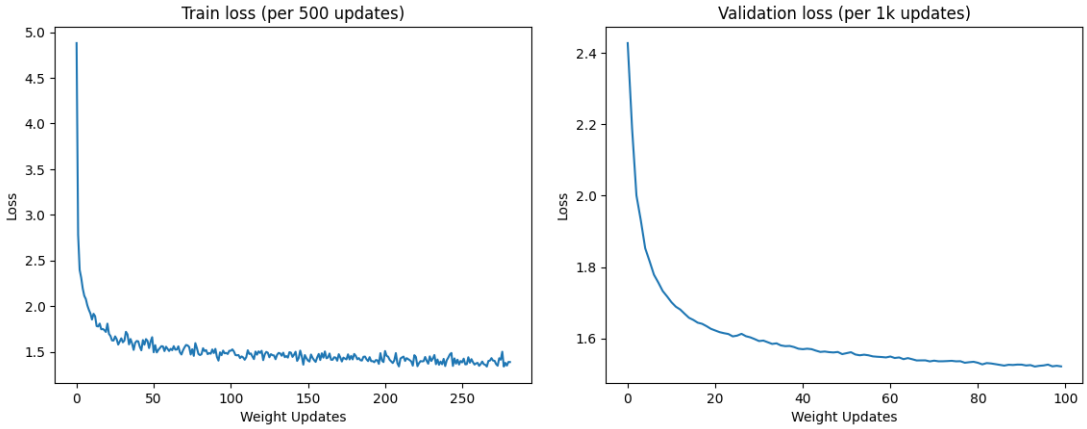
Let w^* be the particular solution of $Y = Xw$ optimization problem in (a), and w_n be any vector in null space of X ($w_n \in null(X)$). Therefore, all solutions of the least square problem are:

$$\{w|Y = Xw\} = \{w^* + w_n|w_n \in null(X)\}$$

Solution 3 (Time spent: 5 hours)**.** Your solution goes here.

(a)
check code

(b)



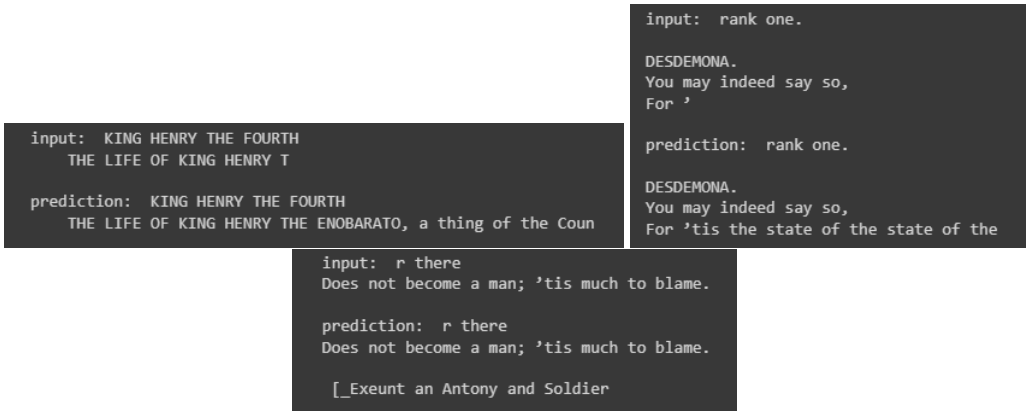
Validation Loss:

```
Epoch:  1  update:  1000  validation loss:  2.427649110682828
Epoch:  1  update:  2000  validation loss:  2.1890885863839986
Epoch:  2  update:  1000  validation loss:  2.0005354586230912
Epoch:  2  update:  2000  validation loss:  1.930718552337093
Epoch:  3  update:  1000  validation loss:  1.8526414086774605
Epoch:  3  update:  2000  validation loss:  1.8162838262989376
Epoch:  4  update:  1000  validation loss:  1.778382405951766
Epoch:  4  update:  2000  validation loss:  1.756357844998453 4
Epoch:  5  update:  1000  validation loss:  1.7326546408882522
Epoch:  5  update:  2000  validation loss:  1.7174960992075132
Epoch:  6  update:  1000  validation loss:  1.7012509633263686
Epoch:  6  update:  2000  validation loss:  1.6886689240698456
Epoch:  7  update:  1000  validation loss:  1.6808036921882359
Epoch:  7  update:  2000  validation loss:  1.6689890757054726
Epoch:  8  update:  1000  validation loss:  1.6576791559140678
Epoch:  8  update:  2000  validation loss:  1.6515333521925708
Epoch:  9  update:  1000  validation loss:  1.643921775634733
Epoch:  9  update:  2000  validation loss:  1.6407505082202332
Epoch:  10  update:  1000  validation loss:  1.6342361120886009
Epoch:  10  update:  2000  validation loss:  1.6267385728667845
Epoch:  11  update:  1000  validation loss:  1.622003506800189
Epoch:  11  update:  2000  validation loss:  1.6173728806195862
Epoch:  12  update:  1000  validation loss:  1.6144696311285 96
Epoch:  12  update:  2000  validation loss:  1.612144548048505
Epoch:  13  update:  1000  validation loss:  1.6054745617496171
Epoch:  13  update:  2000  validation loss:  1.6076267884771311
Epoch:  14  update:  1000  validation loss:  1.6126877856288493
Epoch:  14  update:  2000  validation loss:  1.60603455442453
Epoch:  15  update:  1000  validation loss:  1.6028525451169076
Epoch:  15  update:  2000  validation loss:  1.597943285111852
Epoch:  16  update:  1000  validation loss:  1.5924662494848965
Epoch:  16  update:  2000  validation loss:  1.5937183193258337
```

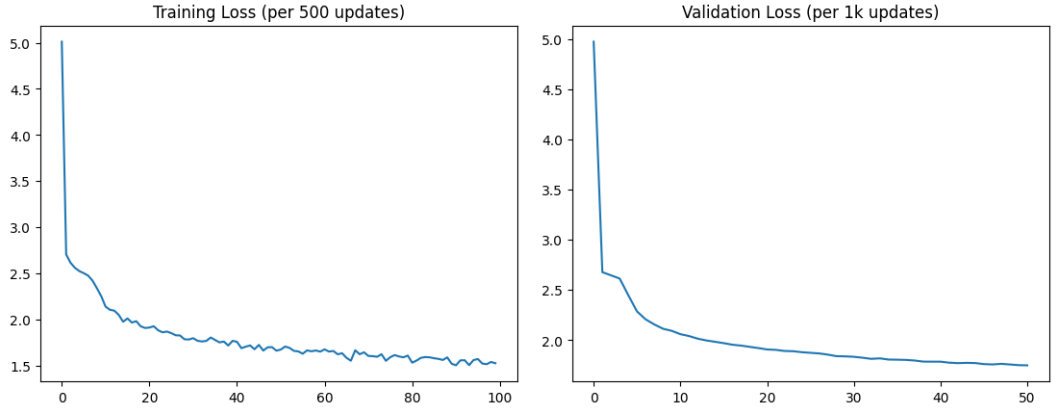
```
Epoch:  17  update:  1000  validation loss:  1.5890402722663934
Epoch:  17  update:  2000  validation loss:  1.5844052846876011
Epoch:  18  update:  1000  validation loss:  1.585964261722429
Epoch:  18  update:  2000  validation loss:  1.5801308680733779
Epoch:  19  update:  1000  validation loss:  1.5781471522400423
Epoch:  19  update:  2000  validation loss:  1.5786442290329152
Epoch:  20  update:  1000  validation loss:  1.5755553742391117
Epoch:  20  update:  2000  validation loss:  1.5709554252041182
Epoch:  21  update:  1000  validation loss:  1.5695888785854681
Epoch:  21  update:  2000  validation loss:  1.570971603244331
Epoch:  22  update:  1000  validation loss:  1.569674389460687
Epoch:  22  update:  2000  validation loss:  1.5655289771377108
Epoch:  23  update:  1000  validation loss:  1.5619568522929468
Epoch:  23  update:  2000  validation loss:  1.5630624969178548
Epoch:  24  update:  1000  validation loss:  1.561340284551021
Epoch:  24  update:  2000  validation loss:  1.5604544128496651
Epoch:  25  update:  1000  validation loss:  1.561801167504377
Epoch:  25  update:  2000  validation loss:  1.5558003433059324
Epoch:  26  update:  1000  validation loss:  1.5584859988193691
Epoch:  26  update:  2000  validation loss:  1.5611196054333814
Epoch:  27  update:  1000  validation loss:  1.5547580335760862
Epoch:  27  update:  2000  validation loss:  1.5525047692601406
Epoch:  28  update:  1000  validation loss:  1.5542272437856683
Epoch:  28  update:  2000  validation loss:  1.5524981010011045
Epoch:  29  update:  1000  validation loss:  1.5494403315133085
Epoch:  29  update:  2000  validation loss:  1.5484026768807835
Epoch:  30  update:  1000  validation loss:  1.5476779391721505
Epoch:  30  update:  2000  validation loss:  1.5465261057803505
Epoch:  31  update:  1000  validation loss:  1.5490833027434043
Epoch:  31  update:  2000  validation loss:  1.5449266257360683
Epoch:  32  update:  1000  validation loss:  1.5465046647263117
Epoch:  32  update:  2000  validation loss:  1.5416548528508476
```

```
Epoch:  33  update:  1000  validation loss:  1.5446108086180381
Epoch:  33  update:  2000  validation loss:  1.5414438490168978
Epoch:  34  update:  1000  validation loss:  1.538032423212043 3
Epoch:  34  update:  2000  validation loss:  1.5382082947287417
Epoch:  35  update:  1000  validation loss:  1.5382918751935022
Epoch:  35  update:  2000  validation loss:  1.535377373057782
Epoch:  36  update:  1000  validation loss:  1.537464570354771
Epoch:  36  update:  2000  validation loss:  1.5356565110205247
Epoch:  37  update:  1000  validation loss:  1.5357663778967063
Epoch:  37  update:  2000  validation loss:  1.5362813906852755
Epoch:  38  update:  1000  validation loss:  1.5369667364218158
Epoch:  38  update:  2000  validation loss:  1.5356172348662767
Epoch:  39  update:  1000  validation loss:  1.5358961661204504
Epoch:  39  update:  2000  validation loss:  1.5316052631836696
Epoch:  40  update:  1000  validation loss:  1.5330758496673822
Epoch:  40  update:  2000  validation loss:  1.5342979397237895
Epoch:  41  update:  1000  validation loss:  1.5315211058341254
Epoch:  41  update:  2000  validation loss:  1.5270267223056995
Epoch:  42  update:  1000  validation loss:  1.5306775585177275
Epoch:  42  update:  2000  validation loss:  1.5296036666351904
Epoch:  43  update:  1000  validation loss:  1.5276623584467859
Epoch:  43  update:  2000  validation loss:  1.5256403011414268
Epoch:  44  update:  1000  validation loss:  1.523590991032412
Epoch:  44  update:  2000  validation loss:  1.5259777465212565
Epoch:  45  update:  1000  validation loss:  1.52532132811431
Epoch:  45  update:  2000  validation loss:  1.5264492360498024
Epoch:  46  update:  1000  validation loss:  1.5263139111219055
Epoch:  46  update:  2000  validation loss:  1.5239095984617643
Epoch:  47  update:  1000  validation loss:  1.525015879491315
Epoch:  47  update:  2000  validation loss:  1.5209624904656986
Epoch:  48  update:  1000  validation loss:  1.5229682137243779
Epoch:  48  update:  2000  validation loss:  1.5240074014595868
Epoch:  49  update:  1000  validation loss:  1.5264236289440822
Epoch:  49  update:  2000  validation loss:  1.5215409359925163
Epoch:  50  update:  1000  validation loss:  1.5231135981859558
Epoch:  50  update:  2000  validation loss:  1.5213859089746924
```

New Generated Text:



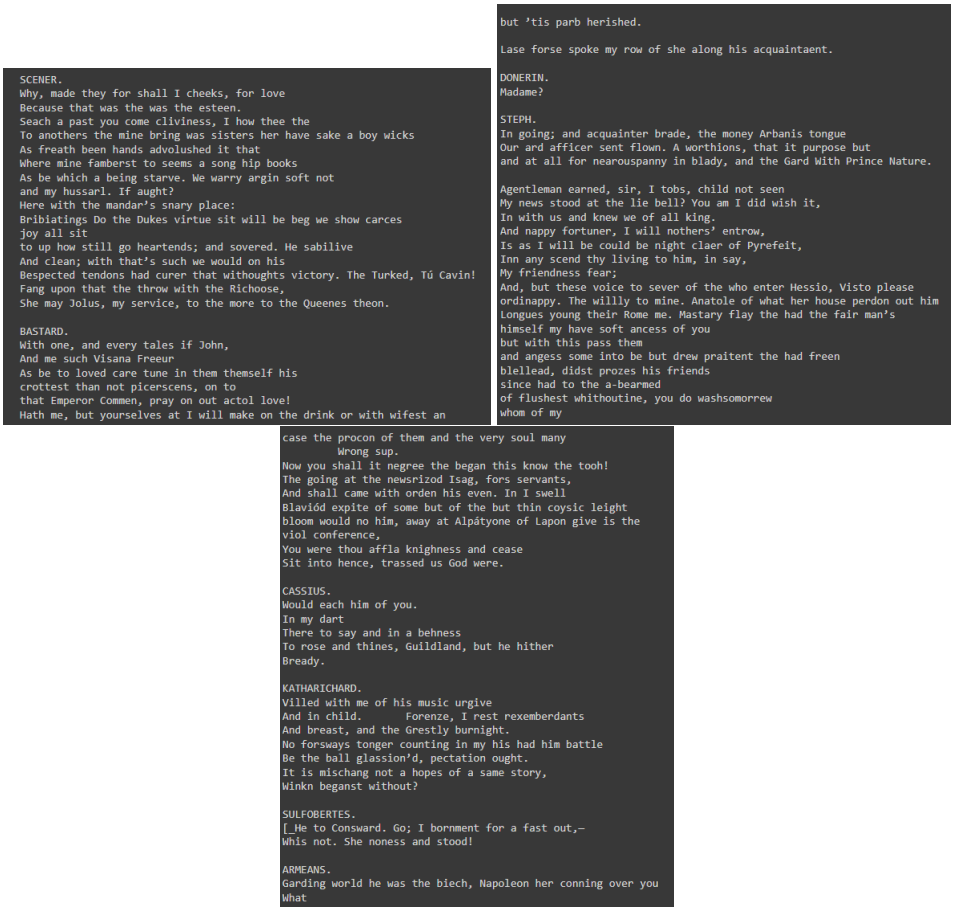
(c)



Validation Loss:

```
batch:  0 Validation Loss:  4.973207612991333
batch:  1000 Validation Loss:  2.6778144624233247
batch:  2000 Validation Loss:  2.6462946701049805
batch:  3000 Validation Loss:  2.6143660156726836
batch:  4000 Validation Loss:  2.447137183189392
batch:  5000 Validation Loss:  2.2862302038669586
batch:  6000 Validation Loss:  2.207444892644882
batch:  7000 Validation Loss:  2.156748375892639
batch:  8000 Validation Loss:  2.114053082346916
batch:  9000 Validation Loss:  2.0937233757972717
batch:  10000 Validation Loss:  2.0603605633974076
batch:  11000 Validation Loss:  2.04147741150856
batch:  12000 Validation Loss:  2.015492356300354
batch:  13000 Validation Loss:  1.9974401844739913
batch:  14000 Validation Loss:  1.9846482897996902
batch:  15000 Validation Loss:  1.970666754245758
batch:  16000 Validation Loss:  1.95525548183918
batch:  17000 Validation Loss:  1.9464742529392243
batch:  18000 Validation Loss:  1.9340722509622574
batch:  19000 Validation Loss:  1.9218516941070556
batch:  20000 Validation Loss:  1.9080001199245453
batch:  21000 Validation Loss:  1.9041486945152282
batch:  22000 Validation Loss:  1.8931296759843825
batch:  23000 Validation Loss:  1.8905980091094972
batch:  24000 Validation Loss:  1.8808431090116502
batch:  25000 Validation Loss:  1.8746168142557145
batch:  26000 Validation Loss:  1.8684356534481048
batch:  27000 Validation Loss:  1.8567192525863647
batch:  28000 Validation Loss:  1.8412988815307618
batch:  29000 Validation Loss:  1.8393372783660888
batch:  30000 Validation Loss:  1.835056489109993
batch:  31000 Validation Loss:  1.8258316828012466
batch:  32000 Validation Loss:  1.8146019550561905
batch:  33000 Validation Loss:  1.8193516894578934
batch:  34000 Validation Loss:  1.8075719940662385
batch:  35000 Validation Loss:  1.8061986820697784
batch:  36000 Validation Loss:  1.8039644347429276
batch:  37000 Validation Loss:  1.7979940720796586
batch:  38000 Validation Loss:  1.7871054576635361
batch:  39000 Validation Loss:  1.7865333091020583
batch:  40000 Validation Loss:  1.7864405878782272
batch:  41000 Validation Loss:  1.7764378986358642
batch:  42000 Validation Loss:  1.771910699367523
batch:  43000 Validation Loss:  1.774666192173958
batch:  44000 Validation Loss:  1.7725779112577438
batch:  45000 Validation Loss:  1.7622790869474412
batch:  46000 Validation Loss:  1.758855893015861 5
batch:  47000 Validation Loss:  1.764771698117256
batch:  48000 Validation Loss:  1.7586809061765671
batch:  49000 Validation Loss:  1.751695096731186
batch:  49999 Validation Loss:  1.749914245009422 3
```

New generated Text:



Implementation Mechanism:

1. Tokenization: characters are transformed into integer or integers are transformed back to characters based on vocabulary hash table. Vocabulary size is maintained.

2. DataLoader: sequences are split into train and val datasets, then divided into batches for training process.

3. Hyperparameter: This class is used to store all parameters for training, in my implementation, I use context size 128, embedding size 128, maximum weight update 50000, learning rate $1e^{-4}$

4. Multihead Attention Block: This block computes each pair of positions in the sequence. In the block, we first computes attention scores based on key-value set, then we mask some values out from it. In the forward function, heads are split in order to calculate scores independently and then combined together.

5. FeedForward: Two layers with ReLU activation are implemented. This block is used to transform features so that we can use in the transformer.

6: Encoder and Decoder: I implement position encoding so that the model has knowledge of positions of tokens in the text. In transformer block, we calculate attention, and pass it into two normalization

layers. The encoder layer is generated. Similarly, the attention of label text is added to encoder's output, goes through normalization and we get decode layer.

7. Model Training: I use Adam optimizer and cross entropy loss to train the model. For every 1000 updates, I print out the mean loss for validation dataset.

8. Model Prediction: I wrote a function that generates a sequence by iteratively predicting and sampling the next token. Softmax is applied to obtain probabilities from the logits, then I sample the next token based on the probability distribution. I set the length to 1024 so that we can predict next 1024 characters.