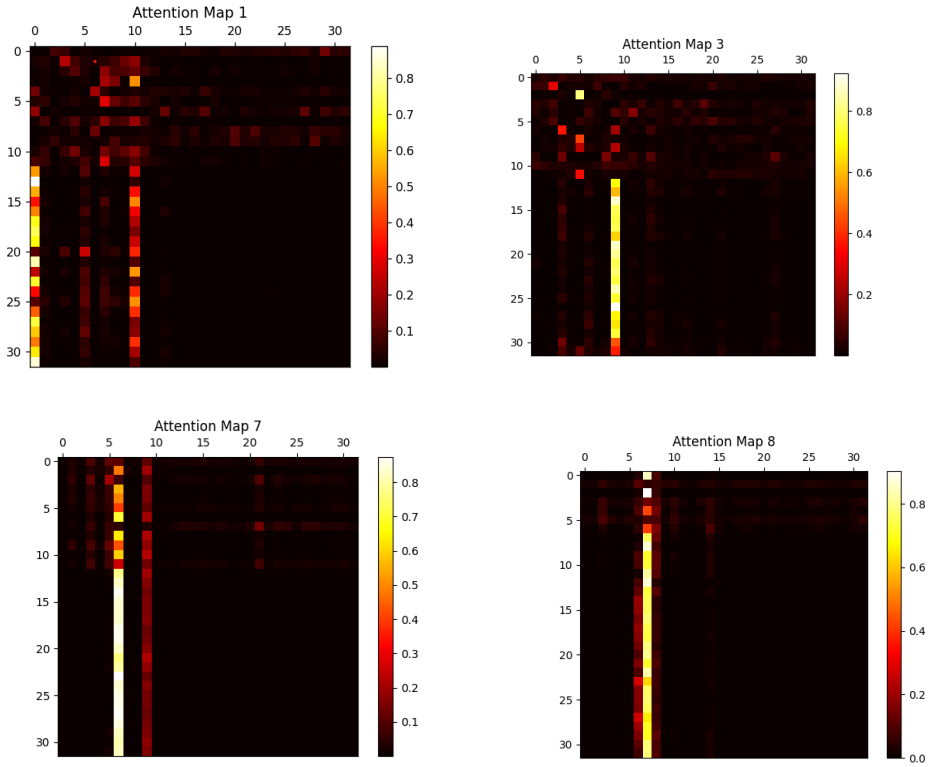
NLP ASSIGNMENT 2 PID: A69034019

Part 1

Vocabulary size is 5755
ACCURACY: stable increase in accuracy
Training Classification Task...
Classification Epoch [1/15], Step [100], Avg Loss: 1.0695
Epoch 1/15, Train classifier Accuracy: 47.42%, Test classifier Accuracy: 36.80%
Epoch 2/15, Train classifier Accuracy: 53.63%, Test classifier Accuracy: 50.13%
Classification Epoch [3/15], Step [100], Avg Loss: 0.9338
Epoch 3/15, Train classifier Accuracy: 58.37%, Test classifier Accuracy: 50.27%
Classification Epoch [8/15], Step [100], Avg Loss: 0.3260
Epoch 8/15, Train classifier Accuracy: 91.30%, Test classifier Accuracy: 79.20%
Classification Epoch [9/15], Step [100], Avg Loss: 0.2836
Epoch 11/15, Train classifier Accuracy: 95.84%, Test classifier Accuracy: 82.40%
Classification Epoch [15/15], Step [100], Avg Loss: 0.0718
Epoch 15/15, Train classifier Accuracy: 98.37%, Test classifier Accuracy: 86.00%

Attention maps:

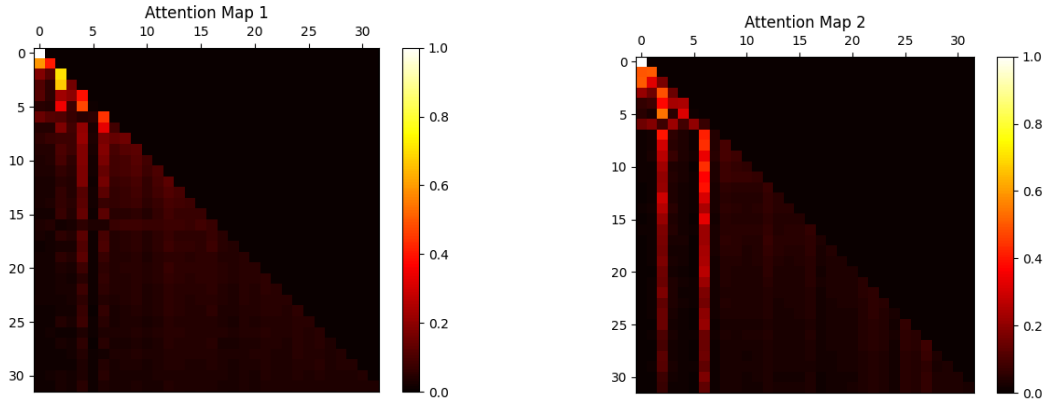


NO OF PARAMETERS: Number of parameters in the encoder: 496035
LOGIC: Followed the class lecture recording for writing the code.

Part 2: Decoder

PERPLEXITY:
Iteration 100/500,Perplexity: 570.9513
Iteration 200/500,Perplexity: 431.5568
Iteration 300/500,Perplexity: 309.5627
Iteration 400/500,Perplexity: 225.7650
Iteration 500/500,Perplexity: 177.5809
Perplexity on speechesdataset/test_LM_wbush.txt: 479.8911
Perplexity on speechesdataset/test_LM_obama.txt: 393.6724
Perplexity on speechesdataset/test_LM_hbush.txt: 431.6000
Input tensor shape: torch.Size([1, 32])
Number of attention maps: 2
Number of parameters in the decoder: 863372

ATTENTION MAPS:



Observations of encoder and decoder:
Triangular Pattern of attention maps in decoder: This pattern ensures that each token only attends to itself and preceding tokens, enforcing a left-to-right generation order.
Attention peaks around recent tokens, which shows the model's reliance on immediate context for generating sequences. Encoder maps generally exhibit more balanced attention across tokens while decoder maps are constrained by masks.

The reason for the difference in perplexity in test sets may be due to below reasons:

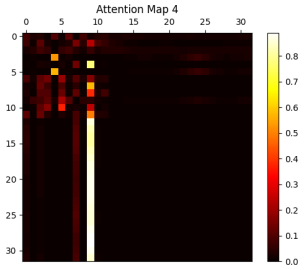
1. complex sentence structures or unique style of each president.
2. training data may be more aligned with one style of president.
3. The focus of the speech, the frequency of the words, and the readability of a president's sentence affect the perplexity.

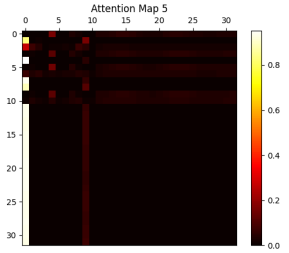
Part 3

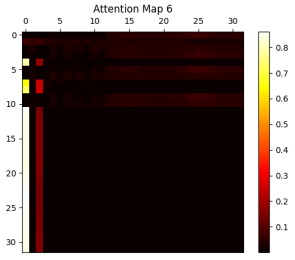
3a: Positional encoding

LOGIC: used positional encoding formula from attention is all you need paper.

ENCODER: Shows increase in accuracy and stabilize after epoch 8:
Classification Epoch [1/15], Step [100], Avg Loss: 1.0780
Epoch 1/15, Train classifier Accuracy: 44.65%, Test classifier Accuracy: 33.33%
Classification Epoch [2/15], Step [100], Avg Loss: 1.0458
Epoch 2/15, Train classifier Accuracy: 53.82%, Test classifier Accuracy: 45.20%
Classification Epoch [3/15], Step [100], Avg Loss: 0.9655
Epoch 3/15, Train classifier Accuracy: 60.18%, Test classifier Accuracy: 50.93%
Classification Epoch [4/15], Step [100], Avg Loss: 0.8115
Epoch 4/15, Train classifier Accuracy: 68.69%, Test classifier Accuracy: 65.20%
Classification Epoch [5/15], Step [100], Avg Loss: 0.6976
Epoch 6/15, Train classifier Accuracy: 87.91%, Test classifier Accuracy: 76.27%
Classification Epoch [7/15], Step [100], Avg Loss: 0.4252
Epoch 10/15, Train classifier Accuracy: 96.61%, Test classifier Accuracy: 84.93%
Classification Epoch [11/15], Step [100], Avg Loss: 0.1337
Epoch 11/15, Train classifier Accuracy: 97.37%, Test classifier Accuracy: 85.47%
Classification Epoch [12/15], Step [100], Avg Loss: 0.1229
Epoch 12/15, Train classifier Accuracy: 97.56%, Test classifier Accuracy: 84.67%
Classification Epoch [13/15], Step [100], Avg Loss: 0.0823
Epoch 13/15, Train classifier Accuracy: 93.98%, Test classifier Accuracy: 82.00%
Classification Epoch [14/15], Step [100], Avg Loss: 0.0992
Epoch 14/15, Train classifier Accuracy: 97.37%, Test classifier Accuracy: 84.80%
Classification Epoch [15/15], Step [100], Avg Loss: 0.0632
Epoch 15/15, Train classifier Accuracy: 97.90%, Test classifier Accuracy: 83.47%

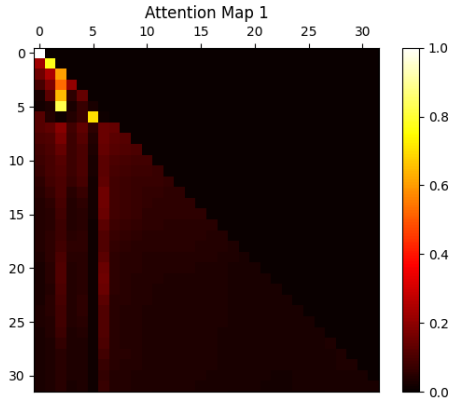


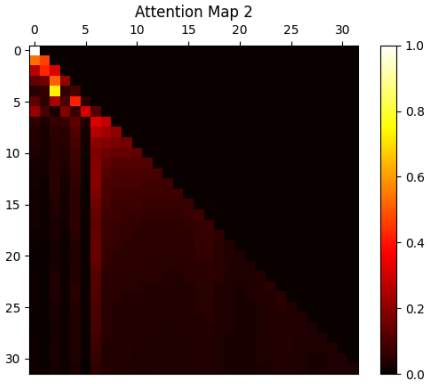




DECODER:
PERPLEXITY:
Iteration 100/500,Perplexity: 544.7603
Iteration 200/500,Perplexity: 364.4020
Iteration 300/500,Perplexity: 232.1530
Iteration 400/500,Perplexity: 168.6071
Iteration 500/500,Perplexity: 126.6446
Perplexity on speechesdataset/test_LM_wbush.txt: 445.3452
Perplexity on speechesdataset/test_LM_obama.txt: 351.9839
Perplexity on speechesdataset/test_LM_hbush.txt: 393.2509
Input tensor shape: torch.Size([1, 32])
Number of attention maps: 2
Number of parameters in the decoder: 861324

ATTENTION MAPS:





3b: sparse attention using a local window

Logic:

Local window attention with window_size=5: effecient for smaller sequences

Local window attention with window_size=15:broader context

WINDOW_SIZE: 5

ENCODER:

Classification Epoch [1/15], Step [100], Avg Loss: 1.0791

Epoch 1/15, Train classifier Accuracy: 42.35%, Test classifier Accuracy: 36.67%

Classification Epoch [5/15], Step [100], Avg Loss: 0.7621

Epoch 5/15, Train classifier Accuracy: 75.67%, Test classifier Accuracy: 64.93%

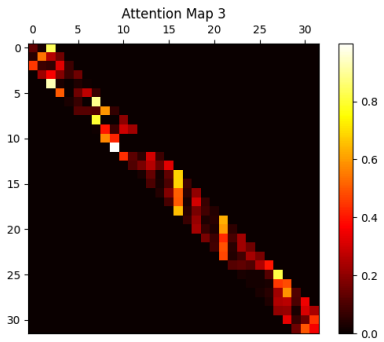
Epoch 9/15, Train classifier Accuracy: 95.46%, Test classifier Accuracy: 81.87%

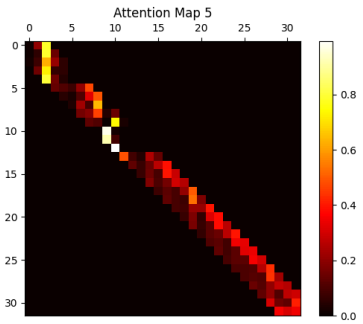
Classification Epoch [10/15], Step [100], Avg Loss: 0.1590

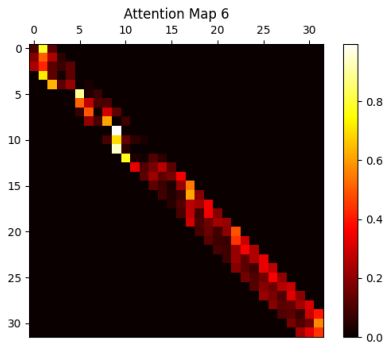
Epoch 13/15, Train classifier Accuracy: 98.23%, Test classifier Accuracy: 84.93%

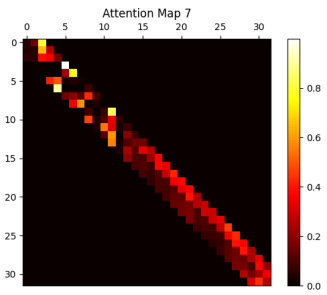
Classification Epoch [14/15], Step [100], Avg Loss: 0.0691

Epoch 15/15, Train classifier Accuracy: 98.04%, Test classifier Accuracy: 86.00%



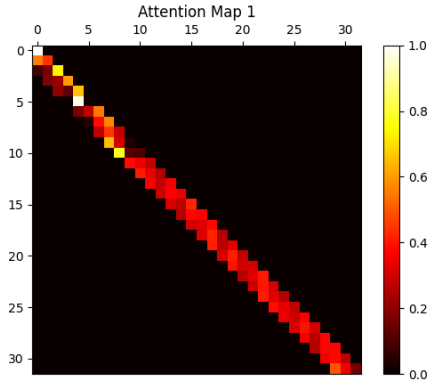


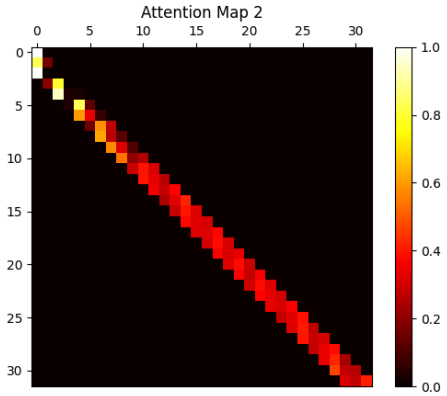




DECODER:

ATTENTION MAPS:

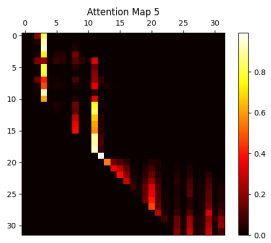


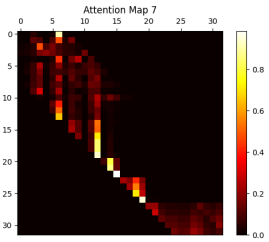


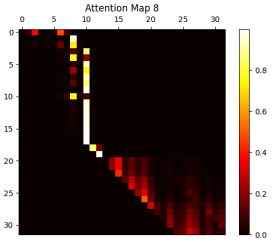
Iteration 100/500,Perplexity: 568.1198
Iteration 200/500,Perplexity: 411.4269
Iteration 300/500,Perplexity: 282.3150
Iteration 400/500,Perplexity: 203.9972
Iteration 500/500,Perplexity: 151.8785

Perplexity on speechesdataset/test_LM_wbush.txt: 459.3376
Perplexity on speechesdataset/test_LM_obama.txt: 368.3820
Perplexity on speechesdataset/test_LM_hbush.txt: 401.5551
Input tensor shape: torch.Size([1, 32])
Number of attention maps: 2
Number of parameters in the decoder: 863372

WINDOW SIZE: 15
ENCODER:
Classification Epoch [1/15], Step [100], Avg Loss: 1.0755
Epoch 1/15, Train classifier Accuracy: 44.65%, Test classifier Accuracy: 33.33%
Classification Epoch [2/15], Step [100], Avg Loss: 1.0597
Epoch 5/15, Train classifier Accuracy: 64.91%, Test classifier Accuracy: 59.47%
Classification Epoch [6/15], Step [100], Avg Loss: 0.7225
Epoch 6/15, Train classifier Accuracy: 76.86%, Test classifier Accuracy: 67.60%
Classification Epoch [7/15], Step [100], Avg Loss: 0.5970
Epoch 8/15, Train classifier Accuracy: 90.63%, Test classifier Accuracy: 79.47%
Classification Epoch [9/15], Step [100], Avg Loss: 0.3637
Classification Epoch [15/15], Step [100], Avg Loss: 0.0773
Epoch 15/15, Train classifier Accuracy: 95.51%, Test classifier Accuracy: 83.07%

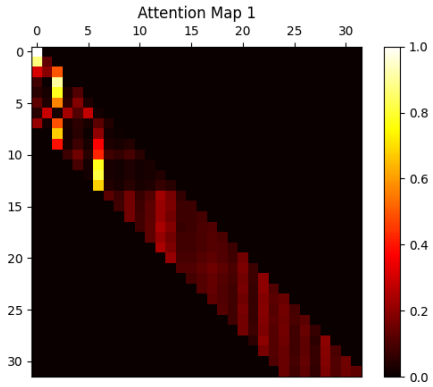


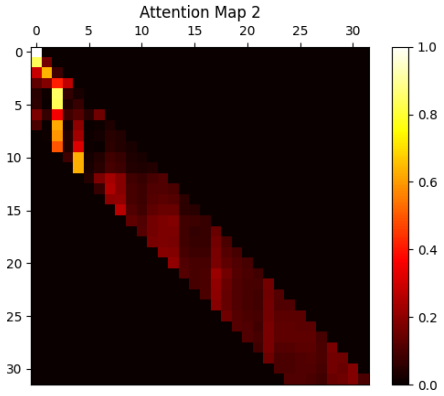




DECODER:

Iteration 100/500,Perplexity: 577.6779
Iteration 200/500,Perplexity: 420.3143
Iteration 300/500,Perplexity: 292.8961
Iteration 400/500,Perplexity: 212.8908
Iteration 500/500,Perplexity: 160.5435
Perplexity on speechesdataset/test_LM_wbush.txt: 472.0496
Perplexity on speechesdataset/test_LM_obama.txt: 387.1735
Perplexity on speechesdataset/test_LM_hbush.txt: 417.1625
Input tensor shape: torch.Size([1, 32])
Number of attention maps: 2
Number of parameters in the decoder: 863372





Observations: part 3

1. **Positional Encoding**: Higher overall performance on both encoder classification and decoder perplexity, especially benefiting longer dependencies in sequences.
2. **Sparse Attention (Local Window)**: Effective for smaller contexts with computational efficiency, though slightly less effective with broader context dependencies.
    a. window size 5 gives better results than window size 15.
3. Both these models show improved perplexity values.
4. With sparse attention using local windows (e.g., window sizes of 5 or 15), attention maps show restricted focus areas, where each token attends only within a limited range.
5. Smaller windows lead to more localized attention patterns, which can be beneficial for capturing fine-grained dependencies but might limit long-range context capture.

Instructions To Run The Code

python main.py --part part1
python main.py --part part2
python main.py --part part3_a: For positional encoding: implementation from attention all you need
python main.py --part part3_b: Sparse attention: based on local window attention

References

1. Attention is all you need research paper
2. Class lecture recording
3. ChatGPT: to correct an error of perplexity becoming too low (20s) for the decoder. This was because I was not initializing weights correctly when applying the mask. Checking generates mask function due to perplexity error. Used for help in debugging and checking errors and checking sparse attention implementation.
4. Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In ICLR, 2022.