PA2 Report

Aritra Dutta

ardutta@ucsd.edu

(1) Below are the Sanity Check screenshots of summing across the attention matrices.

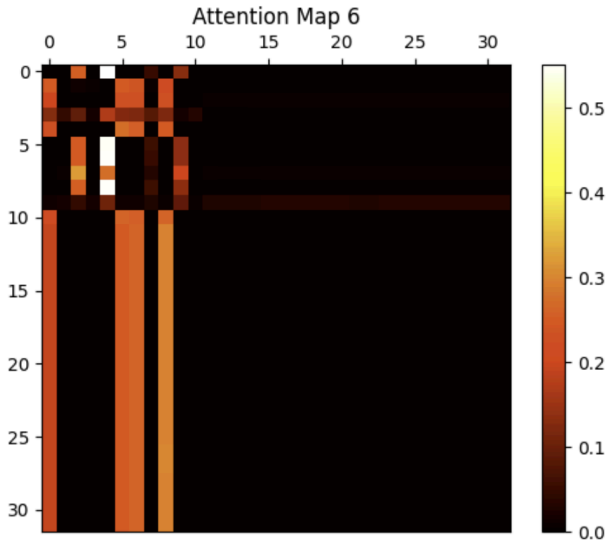
Encoder Screenshot:

```
Input tensor shape: torch.Size([1, 32])
Number of attention maps: 8
aritradutta@Aritras-MacBook-Pro CSE156_PA2_SP24 %
```

Decoder Screenshot:

```
Input tensor shape: torch.Size([1, 32])
Number of attention maps: 8
aritradutta@Aritras-MacBook-Pro CSE156_PA2_SP24 %
```

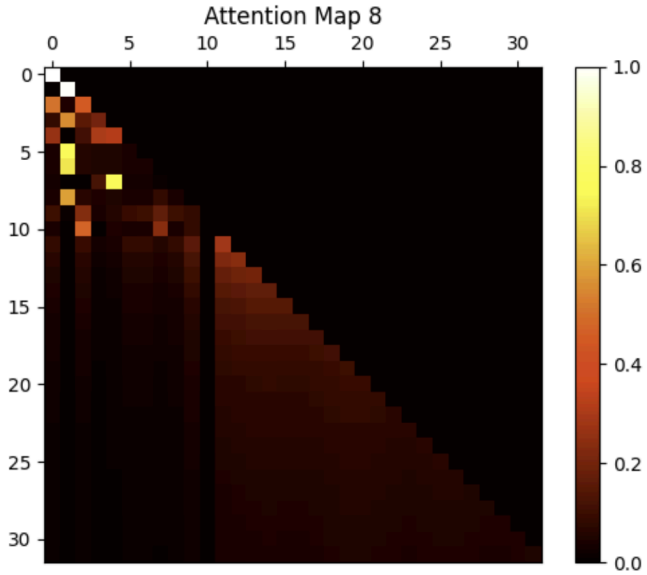
(2) Encoder Attention Map:



Encoder Attention Map Analysis:

Using the sample sentence “Hello, this is a sample sentence for sanity check”, this is the resulting attention map. Essentially, the top left indicates that there are higher attention scores among these positions. Since these are the initial tokens, they often receive higher attention due to its importance in understanding the rest of the sequence. The effect of improved understanding of key tokens is that it ensures critical information is well captured leading to more accurate predictions. The several vertical lines suggest that certain tokens are getting a lot of attention compared to many other tokens. The few white squares indicate that there is possibly missing data or an area of high attention. These areas if not addressed can negatively impact the models performance, possibly leading to wrong attention distribution and predictions that are suboptimal.

Decoder Attention Map:



Decoder Attention Map Analysis:

Using the sample sentence “Hello, this is a sample sentence for sanity check”, the top is the resulting attention map. The top left corner has much higher intensity, which shows that there are higher attention scores in that area. The diagonal area from the top left corner to the bottom right corner indicates that the attention decreases the further we move away from the diagonal. The diagonal pattern indicates that tokens are using more recent tokens, where the attention model utilizes previous tokens to generate the next possible token within the sequence. The models focus on recent tokens shows that the immediate context is captured well and this leads to far more accurate predictions. On the other hand, while it's good that the immediate context is captured, if it constantly uses the same token then it might lead to overfitting which would be quite unfortunate. As a whole this attention map highlights that the model focuses on important tokens regarding immediate context within the sequence, ensuring that important information is captured.

(3) Evaluation Part 1 (Encoder) Results:

Table 1: Loss, Training and Test Accuracy for Each Epoch

Epoch	Train Loss	Train Accuracy (%)	Test Accuracy (%)
1	1.0685	44.65	33.33
2	1.0146	47.80	38.00
3	0.9023	68.40	62.80
4	0.7668	77.44	68.27
5	0.5960	82.36	71.73
6	0.4410	91.30	79.60
7	0.3045	88.53	75.07
8	0.2123	96.99	84.13
9	0.1152	98.42	84.93
10	0.0679	99.19	86.13
11	0.0389	98.76	85.13
12	0.0272	98.90	85.87
13	0.0523	98.80	86.93
14	0.0341	99.52	86.67
15	0.0196	99.67	86.67

With Vocabulary Size of 5,755 and 574,419 parameters.

Evaluation Part 1 Results Analysis:

The training losses, steady decrease from 1.0685 to 0.0196 in the last epoch, indicates that the model is learning from the training data very well, and is fitting the training data effectively as the epochs go on. Likewise, the increased training accuracy from 44.65% to 99.67% indicates that the TransformerClassifier is able to effectively classify the examples within training by the last epoch. Furthermore the test accuracy indicates that the model is able to generalize the unseen data fairly well, but it is certainly noticeable that it is not as good as the training data when it comes to accuracy. Essentially, from the training data we learn that the model is able to effectively learn through the patterns in the training data. The test accuracy indicates that the model has good generalization, and the final epochs of the test accuracy indicates that there is minimal overfitting if there is any.

(4) Evaluation Part 2 (Decoder) Results:

Table 2: Decoder LM Performance

Metric	**Value**
Step 0: Training Perplexity	6787.6353
Step 100: Training Perplexity	523.5106
Step 200: Training Perplexity	342.6949
Step 300: Training Perplexity	230.0512
Step 400: Training Perplexity	167.0136
Step 500: Training Perplexity	130.0876
Step 500: Obama Perplexity	376.8339
Step 500: W. Bush Perplexity	495.2933
Step 500: H. Bush Perplexity	422.1715

With Vocabulary Size of 5,755 and 861,195 parameters.

(5) Evaluation Part 2 Decoder Analysis:
Initially, there was a massive drop in training perplexity from Step 0, to Step 100. This shows that the model was quickly able to learn how the data worked and was able to figure out some basic patterns. Furthermore, we see perplexity decrease steadily as the steps increment which indicates that the model has effectively learned from the model, but we also see a diminishing return as the drops are not as drastic as they initially started. At step 500, we see that the training perplexity is 130.0876 which is drastically lower than step 0, so the model has learned bounds from the data. The higher perplexities for Obama, and the Bushes indicates that the model had a more difficult time to understand the datasets of those speeches and found the contexts more challenging. It's possible that the language and the structure of those contexts are more complex making it harder on the model to understand. As a whole, the decoder model displays a marginal reduction in training perplexity as it went on. However, it had a difficult time with the Bush and Obama Speeches as they had higher perplexities. Essentially, this highlights that with more training data it would be able to perform better across many contexts, and would be able to better understand the patterns, and help improve the models generalization abilities.