Sherlock: Next Word Prediction

Results summary:

Trained for 10 epochs where validation loss decreases from 6.23 to 4.66 and val perplexity from 512 to 106

Final test set perplexity: **112.9**
Final test set accuracy: **25.5 %** (top-1 accuracy). Vocab size is 50257 so random guess is 0.002 %.

I think the quantitative metrics are along the lines of what I expected.

Architecture and training summary:

The only data preparation I did was to replace newline with space. The reason being there were a lot of newlines in the original text and it was artificially increasing perplexity (as model could easily predict it). I decided to use a pretrained tokenizer and embedding since we had limited data and learning language from scratch would be very difficult. I tried bert, roberta and gpt-2 and went ahead with gpt-2 tokenizer and embedding.

The core architecture:

GPT-2 embedding → 2 LSTM layers → masked multi-head attention (4 heads) → residual connection (LSTM + attention output) → output layer.

It outputs probability vector of vocab size

The model was trained for 10 epochs with a learning rate of 2e-3. Batching was used to speed up training. Training takes only about 1 minute on colab T4 GPU.

Things that worked for me while training:

1. Using **pretrained embeddings** decreased epochs required to learn as there was a strong starting point. Even though it is not very standard to pretrained embeddings like gpt2 for lstm networks, my intuition came from the fact that we had limited training data so learning from scratch would be difficult. Freezing the layers during training gave better performance.

2. **Residual layer in the network**: This also helped make convergence faster. The intuition to use this came from the residual layers in the transformer architecture

3. **Decreasing sequence length**: Model started to struggle as sequence length as I increased sequence length so I kept it at 48

Look at outputs:

```
▶ Prompt: “My dear Watson,” Holmes replied,
    “My dear Watson,” Holmes replied, “I am the very young man’s life. We were
    engaged in the London.’  “‘Oh,’ he said. ‘There is not to a little thing,
    but I should to tell him, for it is possible. It is only as a few minutes I
    would find that I had the news and the papers which I had my own fortune.

▶ Prompt: Holmes asked, “And what do you make of it?”
    Holmes asked, “And what do you make of it?” “Well,” said he. “The is only
    possible that Mr. Hosmer Angel came to the place, and the whole was most
    famous that a man was who had in his possession and in the carriage.”  “It
    was a stranger and then he brought the police, with a large, “it is only
    very much curious to me.”  “What does you mean that it is no.
```

There are more examples inside the notebook run.

There is some sense of grammar. It replicates the style of text in the dataset (conversation style). But there is not much coherency or logic. My guess is it is mostly because of the size of the dataset. I also made a mistake by focusing mostly on test perplexity and realised late that generation wasn’t working well.

I tried different variations of this architecture: attention before lstm, using more attention heads, increasing lstm layers; but all of them started to overfit at around the same point. I feel more data is probably the best way to improve the model and quality of outputs. Really curious to see the optimal solution with this dataset as I tried a lot of things without seeing much improvement.