

Problem 1:

In the cell which runs both the `get_vocab` and `encode` method, we see a portion of the input text which appears to be a script for a shakespeare play, a list of unique characters of letters and symbols under `Vocab`, and an encoding where the Text "Romeo and Juliet" where each character is mapped onto integers. Since there is a numeric mapping for each character it implies that the model we are building doesn't really know words, sentences, grammar, etc. Instead its learning patterns within a string of characters. Since each character has an integer mapping we think that the model is predicting which character is going to come next by assigning probabilities to each of the 66 characters in the vocab and choosing the next letter with the greatest probability.

Problem 2:

After running the transformer language model for 20000 steps I've noticed for the first thousand steps, the training loss and test loss decreases linearly. During the first thousand steps, it appears that both the losses are approximately the same. Over the next couple thousand steps the decrease in both losses appears to still be linear, however the slopes for each loss becomes much less steep. Beyond 5000 steps there an increase in steps appeals to barely decrease both losses. After the first 1000 steps, the loss in training slowly becomes lower than the test loss as more steps are taken.

Problem 3:

The `generate` function works by predicting the next character one character at a time by using probabilities for each character in our vocab. These probabilities are calculated by getting the raw scores and softmaxing them. The line `logits,= self(tokens[:, -ctxsize :])` is used so that the characters are used so that we only use the most recent characters in order to predict the next character. We limit the amount of characters used to predict the next character so that the context for the prediction is recent and relevant to the characters directly behind it.

Problem 4: