CSCI 566 – Assignment 2 – Problem 2

1. **Some short one-sentence movie reviews that you wrote yourself, with your model's predicted sentiment.**

```
[20] predict_sentiment(model, tokenizer, "One of the best movies I have watched in a long time.")
0s
0.9941051006317139

[21] predict_sentiment(model, tokenizer, "Exceptional movie! The visuals are phenomenal and one could easily place Interstellar in the top 10 movies of their decade.")
0s
0.991791844367981

[22] predict_sentiment(model, tokenizer, "Absolutely not recommended! The movie does not make any sense and you'd be better off not watching it")
0s
0.010460483841598034

[23] predict_sentiment(model, tokenizer, "An awful film! It must have been up against some real bad movies to be nominated for the Golden Globe Awards.")
0s
0.032912131398916245

[24] predict_sentiment(model, tokenizer, "What an absolutely stunning movie, you won't regret watching it!")
0s
0.9773032069206238
```

Sentence	Output
One of the best movies I have watched in a long time.	0.9941051006317139
Exceptional movie! The visuals are phenomenal and one could easily place Interstellar in the top 10 movies of their decade.	0.991791844367981
Absolutely not recommended! The movie does not make any sense and you'd be better off not watching it	0.010460483841598034
An awful film! It must have been up against some real bad movies to be nominated for the Golden Globe Awards.	0.032912131398916245
What an absolutely stunning movie, you won't regret watching it!	0.9773032069206238

2. Answers to the conceptual questions above.

▾ Conceptual Questions

1. Why is the residual connection is crucial in the Transformer architecture? [5 points]

Residual connections are beneficial to the Transformer architecture since it allows gradients to flow through the network directly which helps the network train and mitigates the vanishing gradient problem. Intuitively, residual connections also help propagate the position embeddings to deeper layers where these values could have been forgotten but since we use residual connections the information from the input of the model which contains the position embeddings can efficiently pass to deeper layers of the Transformer architecture.

2. Why is Layer Normalization important in the Transformer architecture? [5 points]

It enables faster training of the Transformer and stabilizes the training process by enabling smoother gradients and better generalization accuracy. The location of the layer normalization plays an important role in controlling the scale of the gradients as well. Another important point to note is that layer normalization in the Transformer architecture is used to prevent gradient explosion with residual connections because it has been observed that the output variance of deep residual networks grow exponentially with depth. Layer normalization is preferred over batch normalization in the transformer architecture (and in NLP tasks) because the statistics of language data exhibit large fluctuations across the batch dimension.

3. Why do we use the scaling factor of $1/\sqrt{d_k}$ in Scaled Dot Product Attention? If we remove it, what is going to happen? [5 points]

If we remove the scaling factor of $1/\sqrt{d_k}$ in Scaled Dot Product Attention then the value of the dot product between Q and K^T would grow large for large values of d_k. In such a situation the softmax function would return extremely small gradients and thus result in the vanishing gradient problem. To prevent this issue, we use the scaling factor as it scales down the results of the dot product.

---

1. Why is the residual connection is crucial in the Transformer architecture? [5 points]

   Answer - Residual connections are beneficial to the Transformer architecture since it allows gradients to flow through the network directly which helps the network train and mitigates the vanishing gradient problem. Intuitively, residual connections also help propagate the position embeddings to deeper layers where these values could have been forgotten but since we use residual connections the information from the input of the model which contains the position embeddings can efficiently pass to deeper layers of the Transformer architecture.

2. Why is Layer Normalization important in the Transformer architecture? [5 points]

   Answer - It enables faster training of the Transformer and stabilizes the training process by enabling smoother gradients and better generalization accuracy. The location of the layer normalization plays an important role in controlling the scale of the gradients as well. Another important point to note is that layer normalization in the Transformer architecture is used to prevent gradient explosion with residual connections because it has been observed that the output variance of deep residual networks grow exponentially with depth. Layer normalization is preferred over batch normalization in the transformer architecture (and in NLP tasks) because the statistics of language data exhibit large fluctuations across the batch dimension.

3. Why do we use the scaling factor of 1/ sqrt(dk) in Scaled Dot Product Attention? If we remove it, what is going to happen? [5 points]

   Answer - If we remove the scaling factor of 1/sqrt(dk) in Scaled Dot Product Attention then the value of the dot product between Q and KT would grow large for large values of dk. In such a situation the softmax function would return extremely small gradients and thus result in the vanishing gradient problem. To prevent this issue, we use the scaling factor as it scales down the results of the dot product.