ANLP Assignment 2 Report

Mihika Sanghi
2021113014

What is the purpose of self-attention, and how does it facilitate capturing dependencies in sequences?

Self-attention emerged as a powerful solution to the limitations of RNNs and LSTMs in capturing long-range dependencies in sequential data. Unlike its predecessors, self-attention allows each element in a sequence to interact directly with every other element, regardless of their positions.

The mechanism works by creating query, key, and value vectors for each input element through learned linear projections. The core operation involves computing dot products between a query vector and all key vectors, followed by scaling and softmax to generate attention weights. These weights are then used to create a weighted sum of value vectors, producing the output for each position.

This approach offers several advantages:

1. It captures complex dependencies across entire sequences.
2. It mitigates the vanishing gradient problem by facilitating better information flow.
3. It allows for parallelization, leading to faster training and inference times.
4. Through multi-head attention, it can capture various types of dependencies simultaneously.

Why do transformers use positional encodings in addition to word embeddings? Explain how positional encodings are incorporated into the transformer architecture. Briefly describe recent advances in various types of positional encodings used for transformers and how they differ from traditional sinusoidal positional encodings.

Transformers use positional encodings to compensate for the lack of inherent sequence order in their parallel processing approach. These encodings are crucial for the model to understand the relative or absolute positions of elements in a sequence.

The original transformer paper introduced sinusoidal positional encodings, which use sine and cosine functions of different frequencies. These encodings are added to input embeddings before processing. They allow the model to attend to relative positions and extrapolate to longer sequences than seen during training.

Recent advancements in positional encodings include:

1. Rotary Position Embedding (RoPE): This method applies a rotation matrix to query and key vectors, encoding relative positions directly into the transformer's architecture. RoPE offers

better extrapolation to unseen sequence lengths and can be easily integrated into existing models.

2. Learned absolute positional embeddings: These are trainable parameters that the model learns during training.

3. Relative positional embeddings: These explicitly encode the distance between elements, capturing more nuanced positional relationships.

These newer methods aim to improve handling of variable-length sequences and capture more sophisticated positional information, enhancing the transformer's performance on various tasks.

Hyperparameter Tuning

d_model	512
num_heads	4
num_layers	3
dropout	0.1
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.01909779947442732

d_model	1024
num_heads	4
num_layers	3
dropout	0.1
vocab_size	30000
max_seq_length	25
d_ff	2048

average_bleu_test	0.024561832844657648
d_model	512
num_heads	4
num_layers	3
dropout	0.2
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.020916976200902625
d_model	1024
num_heads	4
num_layers	3
dropout	0.2
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.02118478289569909
d_model	512
num_heads	4
num_layers	4
dropout	0.1
vocab_size	30000
max_seq_length	25

d_ff	2048
average_bleu_test	0.020318521628211506
d_model	1024
num_heads	4
num_layers	4
dropout	0.1
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.02092269628033156
d_model	512
num_heads	4
num_layers	4
dropout	0.2
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.02111280294781969
d_model	1024
num_heads	4
num_layers	4
dropout	0.2
vocab_size	30000

max_seq_length	25
d_ff	2048
average_bleu_test	0.019945012178838824
d_model	512
num_heads	8
num_layers	3
dropout	0.1
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.01923676343105272
d_model	1024
num_heads	8
num_layers	3
dropout	0.1
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.019669859867557395
d_model	512
num_heads	8
num_layers	3
dropout	0.2

vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.020579614741178535
d_model	1024
num_heads	8
num_layers	3
dropout	0.2
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.020093064618295603
d_model	512
num_heads	8
num_layers	4
dropout	0.1
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.018635834881617446
d_model	1024
num_heads	8

num_layers	4
dropout	0.1
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.01999766008847866

d_model	512
num_heads	8
num_layers	4
dropout	0.2
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.01999385154689401

d_model	1024
num_heads	8
num_layers	4
dropout	0.2
vocab_size	30000
max_seq_length	25
d_ff	2048
average_bleu_test	0.019538181573025587

Analysis

BLEU Score Findings
- Peak performance: 4 attention heads, 3 encoder/decoder layers, 0.1 dropout (BLEU score: 0.02456)
- Trend: 4 attention heads slightly outperform 8 heads
- No consistent impact observed from varying encoder/decoder layers (3 vs 4)

Validation Loss Insights
- Best configuration: 4 attention heads, 3 encoder/decoder layers, 0.1 dropout
- Trend: 4 attention heads generally yield lower validation loss than 8 heads
- Increasing encoder/decoder layers from 3 to 4 tends to slightly increase validation loss

Dropout Effects
- BLEU Score: 0.1 dropout consistently outperforms 0.2 dropout
- Validation Loss: Mixed results, no clear trend observed

Limitations and Future Work
- Resource constraints limited the extent of hyperparameter tuning
- For more comprehensive insights:
  1. Conduct additional experiments
  2. Extend training duration (more epochs)
  3. Explore a wider range of hyperparameter combinations

This analysis provides initial insights into the performance characteristics of the transformer model under various configurations. However, more extensive experimentation is needed to draw definitive conclusions and optimize the model further.

Link to all the plots:
https://wandb.ai/mihikasanghi/Advanced%20NLP%20Transformers/reports/Assignment-2-ANLP--VmlIdzo5NjEwMjI3