Inquisition
A Reformed™ question-answering system
Andrew Gau, Shannon Yan, Zeb Mehring

Background

Question-answering is a common NLP task, with applications from web search to virtual assistants and more. Many QA systems are based on the transformer architecture, which utilizes a concept called self-attention to produce state-of-the-art results on question-answering tasks.

Problem

- How can we produce a question-answering system that performs well on SQuAD 2.0?
- Many transformer-based models can be incredibly expensive to train and use. How can we make it more efficient?

Methods

QANet

- A powerful, transformer-based question-answering system
- Given a “context” paragraph and a question about the paragraph, this model can predict the start and end of the answer to the question contained in the original paragraph

![Diagram of QANet architecture]

- **Context2Query Attention**
- **Encode**

Reformer

- **Transformer Encoder**
- **Reversible Transformer**

Methods (cont.)

LSH Self-Attention

Observation: The QR" product in self-attention is a sparse matrix, made sparser still by the application of the softmax function.

Idea: Use locality-sensitive hashing to compute only the nonzero entries.

![Diagram of LSH Self-Attention]

RevNets

Observation: Storing the activations of fully-connected layers for backpropagation requires O(n^2) memory.

Idea: Rearrange the equations so that activations can be computed dynamically (rather than stored).

![Diagram of RevNets]

Expansions and Analysis

Task:
- Predict answers to questions from the SQuAD 2.0 dataset.
- Evaluate performance using EM and F1 scores.
- Measure peak memory usage and per-epoch.

Baseline: BIDAF

Experiments "QANet, LSH (QANet model with LSH Self Attention), Reformer-modified (The QANet model with a RevTransformer in its Encoder Block instead of the Transformer Encoder Block)"

- QANet vastly outperforms BIDAF as expected (+6.7 EM, +6.32 F1)
- Reformer modifications reduce memory usage as expected (~10% for small dim, decreasing as dimensions increase)
- Dot product self-attention uses more memory than LSH, as expected.
- Using LSH instead of dot product self-attention hampers performance significantly (-2.7 EM, -3.33 F1) despite claims to the contrary by the Reformer paper. The Reformer paper’s evaluation may have been insufficient. It would be interesting to see a large-scale study of these modifications as is done Transformer Modifications: Transfer Across Implementations and Applications.

Conclusions and Future Work

- Reformer-style modifications to existing transformer-based architectures seem effective at reducing computational burden, though come at a cost to performance.
- Future experiments: apply reformer-style modification to popular, expensive language models (GPT-X, BERT, etc.) to make them more accessible.