Stanford | ENGINEERING
Computer Science

BERT-A: Fine-tuning BERT with Adapters and Data Augmentation

CS 224n: NLP with Deep Learning
Poster Presentation, Winter 2019

Sina J. Semnani
sinaj@stanford.edu

Fatma Tlili
ftlili@stanford.edu

Kaushik Ram Sadagopan
kaushik7@stanford.edu

Introduction and Problem

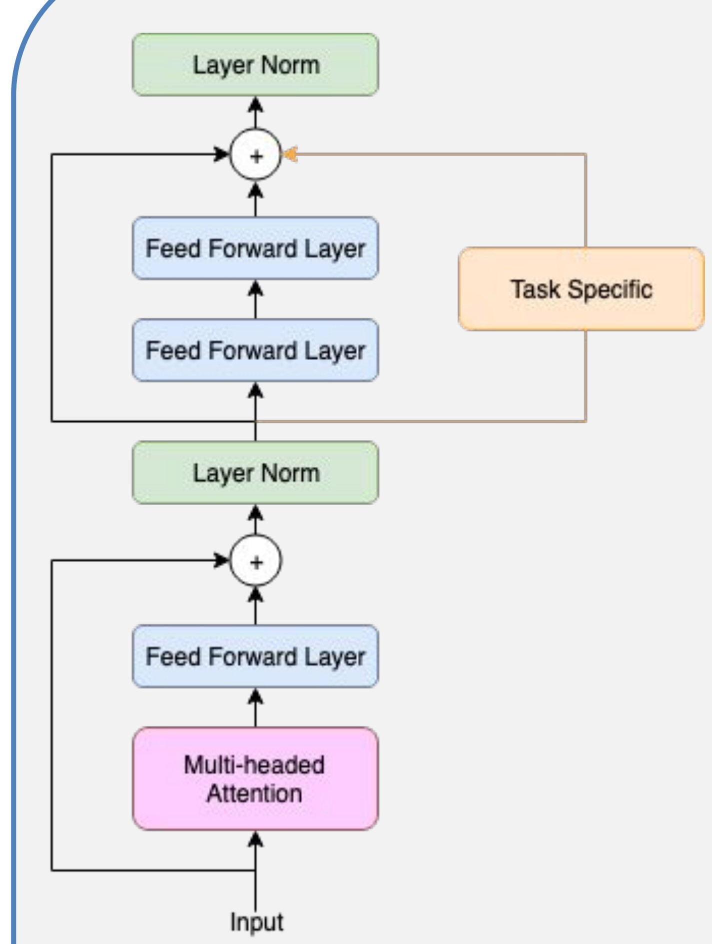
- In recent years, two trends in NLP research:
  - Pre-trained contextual embeddings: ELMo, BERT, etc
  - Multi-task learning: Decathlon, GLUE, etc.

How can we use both?
A naive approach would result in millions of additional parameters per task. These need to be stored and loaded for each inference.

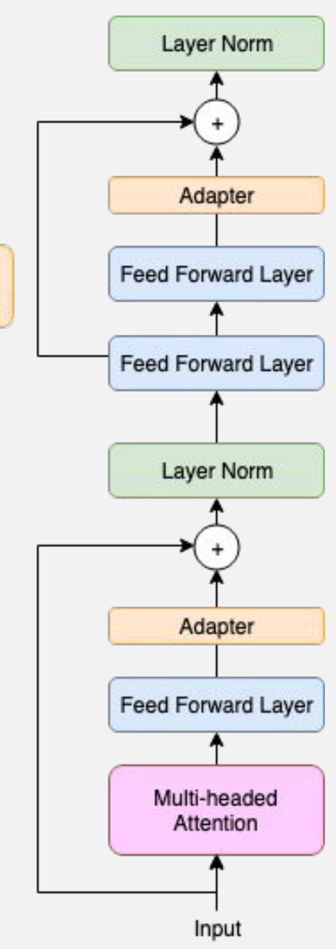
Problem: Question Answering
Dataset: SQuAD 2.0: (paragraph, question) pairs, either the answer is a span in paragraph or there are no answers.

Goals:
- Improve performance in terms of F1 and exact match (EM) scores
- Keep additional trainable parameters to a minimum

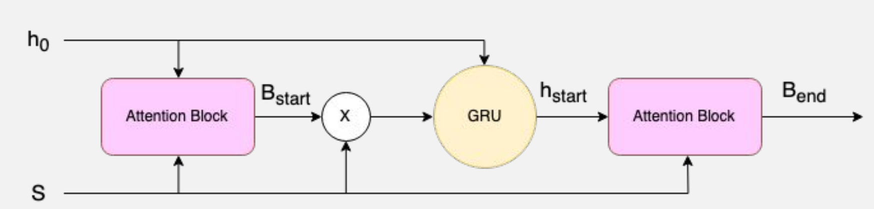
Approach



Projected Attention Layers



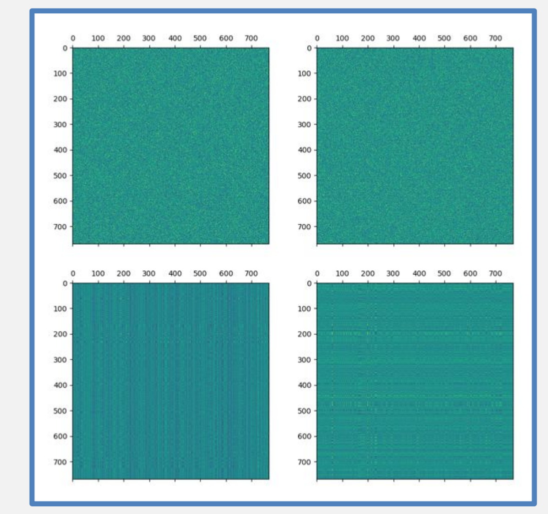
Adapters



Answer Pointer Architecture

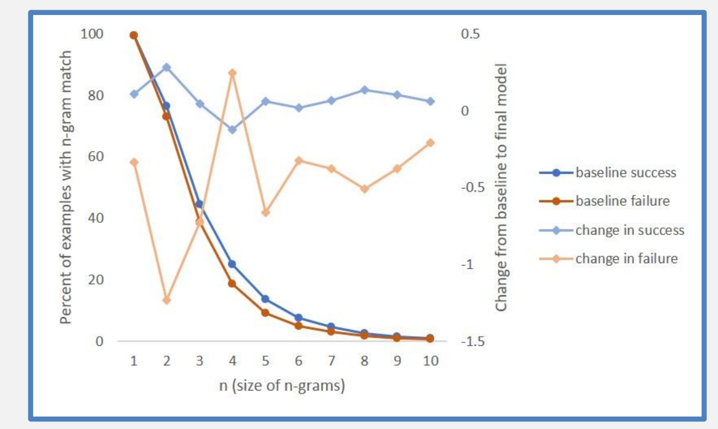
- We experiment with two types of task-specific modules *inside* each Transformer block
  - Adapters
  - Projection Attention Layers
- We use an Answer Pointer layer as our output layer
- For better performance, we use:
  - data augmentation (increase instances of "no-answer" questions)
  - transfer-learning from CoQA dataset

Analysis

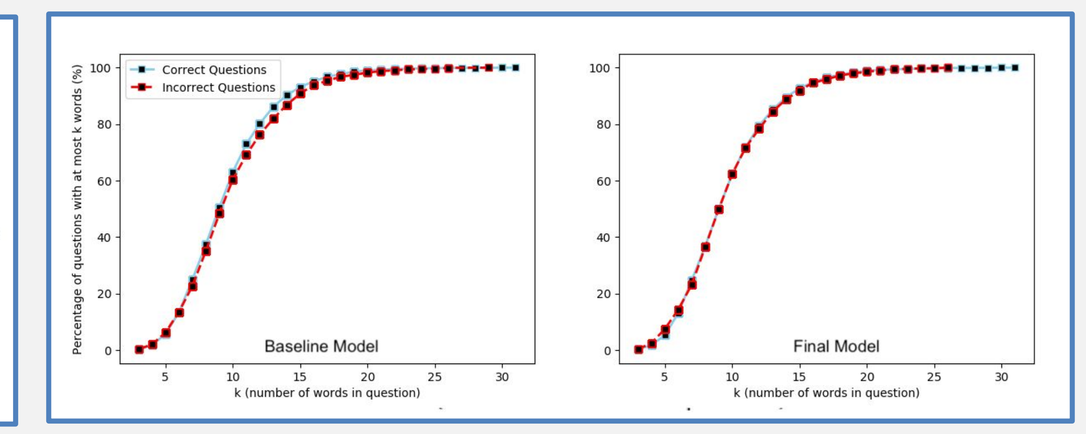


- Adapters' weights in the last transformer block:
  - We probably did not need adapters in self-attention.
  - But adapters in all blocks in output learned patterns.

Performance improvements distribution for non-trivial questions.



Performance improvements for longer questions



Storage Efficient Results

Here we aim to maintain performance while minimizing number of additional parameters.

Model	F1	EM	# Parameters (Overhead)
baseline (fine-tuned)	76.5	73.5	110 M (+100%)
baseline (top block fine-tuned)	54.0	51.7	9.2 M (+8.3%)
baseline (frozen)	51.1	51.0	1.5 K (+0.001%)
baseline (frozen) + PALs (120)	63.9	60.7	704 K (+0.64%)
baseline (frozen) + Adapters (768)	70.9	67.4	592.9 K (+0.54%)
baseline (frozen) + Adapters (768) + LayerNorm	74.7	72.3	629.7 K (+0.57%)

Adapters consistently outperform other approaches in QA.
We achieve comparable performance with just **0.57%** additional parameters to store per task.

Performance Results

Here we sacrifice storage efficiency for performance.
We train on SQuAD 2.0, CoQA, no-answer augmented datasets
Adapters are trained after other parameters are done training.

Model	F1 (dev)	EM (dev)	Training time (minutes)
baseline (fine-tuned)	76.5	73.5	377
baseline + Answer Pointer	76.7	73.5	388
baseline + Data Augmentation	77.9	75.5	1110
baseline + Pre-training on CoQA	78.5	75.7	836
baseline + Pre-training on CoQA + Adapter	79.2	76.3	1240
baseline + Pre-training on CoQA + Data Augmentation + Answer Pointer	79.5	76.5	1722
baseline + Pre-training on CoQA + Data Augmentation + Answer Pointer + Adapter	80.5	77.5	2151

Test set scores: F1: **81.44 (3rd)** EM: **78.36**

Conclusion and Future Work

- Using Adapters with frozen BERT is an effective way to decrease per task parameters in a multi-task learning setting.
- Fine-tuning BERT with Adapters can increase the performance in terms of F1 and EM scores without overfitting.
- Even simple data augmentation techniques work well compared to architectural changes after the top layer of BERT.
- **Future work:** Assessing interpretability of task-specific modules inside BERT

References

- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
- Houlsby, Neil, et al. "Parameter-Efficient Transfer Learning for NLP." *arXiv preprint arXiv:1902.00751* (2019).
- Jia, Robin, and Percy Liang. "Adversarial examples for evaluating reading comprehension systems." *arXiv preprint arXiv:1707.07328* (2017).