The Right Tool for the Job: Matching Model and Instance Complexities

Roy Schwartz, Gabi Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith

ACL 2020
Premise: Big Models
Big Models are Expensive
Strubell et al., 2019; Schwartz et al., 2019

Our goal: Efficient inference
Efficient Inference
Common Approaches

Distillation (teacher/student)
Pruning
Quantization
Our Approach:
Matching Model and Instance Complexity

What a great movie!
Our Approach:
Matching Model and Instance Complexity

I could definitely see why this movie received such great critiques, but at the same time I can’t help but wonder whether the plot was written by a 12 year-old or by an award-winning writer.
Pretrained BERT Fine-tuning

Layer n

Layer n-1

Layer n-2

Layer 2

Layer 1

Layer 0

Input

Prediction

Slowest, most accurate
Partial BERT Baseline

Layer n
Layer n-1
Layer n-2
...
Layer 2
Layer 1
Layer 0
Input

Prediction

Faster, less accurate
Partial BERT Baseline

Layer 0 ➔
Layer 1 ➔
Layer n-2 ➔
Layer n-1 ➔
Layer n ➔

Layer 2 ➔
Layer 1 ➔
Layer n ➔
Layer n-1 ➔
Layer n-2 ➔
Layer 2 ➔
Layer 1 ➔
Layer 0 ➔

Input ➔

Prediction

Fastest, least accurate
Strong Baselines

Speed/Accuracy Tradeoff

3 times faster, with 1% lower accuracy
Our Approach: Training Time

\[ \text{loss} = \sum_{i} \text{loss}_i \]
Our Approach: Test Time

Layer 0

Layer 1

Layer 2

Layer n-2

Layer n-1

Layer n

Input

Prediction

Is confident?

Yes

Early exit

No
Calibrated Confidence Scores

• We interpret the softmax label scores as model confidence
  • We calibrate the scores using *temperature calibration* (Guo et al., 2017)
• Speed/accuracy tradeoff controlled by a *single* early-exit confidence threshold (a *runtime* parameter)
Experiments

• Datasets
  • Text classification
    • AG News (Zhang et al., 2015); IMDB (Maas et al., 2011); SST (Socher et al., 2013)
  • NLI
    • SNLI (Bowman et al., 2015); MultiNLI (Williams et al., 2018)

• BERT-large-uncased (Devlin et al., 2019)
  • Output classifiers added to layers 0,4,12 and 23
**Better Speed/Accuracy Tradeoff**

**Text Classification**

Our model is twice as fast, same performance as baseline.

Baseline:
- 3 times faster, with 1% lower accuracy

Our model:
- 5 times faster, with 1% lower accuracy

---

**AG**

- thr=0.95
- thr=0.6
- thr=0.55
- thr=0

**SST**

- thr=1

**IMDB**

- Our model is twice as fast, same performance as baseline
Similar Speed/Accuracy Tradeoff

NLI

![SNLI Graph](image)

![MNLI Graph](image)
Highlights

• No effective growth in parameters
  • < 0.005% additional parameters

• Training (i.e., fine-tuning) is not slower

• A single trained model provides multiple options along the speed/accuracy tradeoff
  • A single runtime parameter: confidence threshold

• Caveat: requires batch size=1 during inference
More Highlights

See Paper!

- Our method can also be combined with model distillation
- Our method defines a criterion for “difficulty”
Recap

• Efficient inference

• Simple instances exit early, hard instances get more compute

• Training is not slower than the original BERT model

• One model fits all!
  • A single runtime parameter controls for the speed/accuracy curve

• https://github.com/allenai/sledgehammer
Concurrent Work

- *Depth-adaptive transformer*. Elbayad et al., ICLR 2020

- *Balancing cost and benefit with tied-multi transformers*. Dabre et al., 2020

- *Controlling computation versus quality for neural sequence model*. Bapna et al., 2020

- *Explicitly Modeling Adaptive Depths for Transformer*. Liu et al., 2020


- *DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference*. Xin et al., ACL 2020
Come to Jerusalem!
Recap

• Efficient inference

• Simple instances exit early, hard instances get more compute

• Training is not slower than the original BERT model

• One model fits all!
  • A single runtime parameter controls for the speed/accuracy curve

• https://github.com/allenai/sledgehammer