How Much Self-Attention Do We Need? Trading Attention for Feed-Forward Layers

Kazuki Irie*, Alexander Gerstenberger, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
*joining the Swiss AI Lab IDSIA, USI & SUPSI, Lugano, Switzerland

ICASSP 2020, Barcelona, Spain
Machine Learning for Language Processing I [TU2.L3.2], May 5, 2020
Introduction

• **Major trend:** large Transformer language models in 2019 - begin 2020.
  - OpenAI GPT-2 [Radford & Wu19]
  - Nvidia Megatron
  - Microsoft Turing-NLG

• **Applications to ASR** (Interspeech 2019)
  - N-best rescoring, Nvidia [Li & Lavrukhin19]
  - Lattice rescoring & Shallow fusion, RWTH Aachen [Irie & Zeyer19].

• **Large ASR improvements** over well tuned LSTM language model:
  - LibriSpeech SoTA Interspeech 2019 [Lüscher & Beck19]
  - TED-LIUM 2 SoTA this conference (Friday) [Zhou & Michel20].

  with lattice rescoring for hybrid NN/HMM ASR system.

• **In practice:** Large memory requirement for search.
  For lattice rescoring, more than 100 GB for large lattices for some tasks...
Transformer language models in ASR: Large state size

- Transformer LM state: key and value vectors.
- State size: $L \times d_{kv}$ (key dim.) $\times 2$ (for key and value) $\times n$ (position)
- In principle: To be stored for each hypothesis.
Increasing model size, without increasing state size?

Objective/Motivation

• Reduce memory requirement in the original Transformer LM for search!
• From modeling perspective (quantization etc. can be applied on top of it).
• Reconsider the original Transformer layer.

Can we efficiently increase the model size w/o increasing the state size?

• Hyper-parameters in Transformer language model:
  – Number of layers: $L$
  – Tied key, value, and query dimension: $d_{kv}$
  – Feed-forward dimension: $d_{ff}$
  – Number of attention heads: $H$
• State size: $2 \times L \times d_{kv} \times n$
• Only possibility: increase feed-forward dimension $d_{ff}$
• Put parameters in the feed-forward modules more efficiently?
This work: 2 modifications for small state Transformer

1 $F$ feed-forward layers per layer.
2 Sharing key and value matrices.
   
   • (Reduce number of Transformer layers $L$).
This work: 2 modifications for small state Transformer

1 *F* feed-forward layers per layer.
2 Sharing key and value matrices.
   - (Reduce number of Transformer layers *L*).
This work: 2 modifications for small state Transformer

1 $F$ feed-forward layers per layer.

2 **Sharing key and value matrices.**
   - (Reduce number of Transformer layers $L$).
This work: 2 modifications for small state Transformer

1. $F$ feed-forward layers per layer.
2. Sharing key and value matrices.
   • (Reduce number of Transformer layers $L$).
Experimental setups

Dataset: TED-LIUM release 2
  - 152 K-word vocabulary.
  - 270 M running words for language model training.
  - 7 subsets including the transcriptions.
  - Minor overlapping problem in the official data (see our paper for details).
  - Some additional experiments on LibriSpeech (to be found in paper).

ASR baseline
  - Dedicated system paper (new SoTA on TED-LIUM 2) Friday at 15:15 - 17:15.
    Session: Large Vocabulary Continuous Speech Recognition and Search.
    Zhou et al. The RWTH ASR System for TED-LIUM Release 2: Improving
    Hybrid HMM with SpecAugment.
  - Hybrid NN/HMM system.
  - First pass with 4-gram/LSTM.
  - Lattice rescoring to apply LSTM/Transformer language models.
Baseline LM setups: TED-LIUM 2

Basic setups

- 4-gram: interpolation.
- LSTM: 4 layers, 2048 nodes in each layer.
- Transformer: 32 layers, 4096 feed-forward dim, 768 hidden units, 12 heads, no positional encoding.

<table>
<thead>
<tr>
<th>Model</th>
<th># Param. [M]</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Dev</td>
</tr>
<tr>
<td>4-gram</td>
<td>343</td>
<td>105</td>
</tr>
<tr>
<td>4-gram + pruning</td>
<td>161</td>
<td>113</td>
</tr>
<tr>
<td>LSTM</td>
<td>450</td>
<td>74</td>
</tr>
<tr>
<td>Transformer</td>
<td>414</td>
<td>62</td>
</tr>
</tbody>
</table>

All language model configurations/models online: https://github.com/rwth-i6/returnn-experiments/tree/master/2020-lm-small-state-trafo
Effect of deep feed-forward module

Perplexity results on TED-LIUM 2

$L = \text{Number of Transformer layers}$

$F = \text{Number of feed-forward layers per Transformer layer}$

<table>
<thead>
<tr>
<th>Transformer layer</th>
<th>$F$</th>
<th>$L$</th>
<th>$d_{ff}$</th>
<th>State size per position [K]</th>
<th>#Param. [M]</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dev</td>
</tr>
<tr>
<td>Standard</td>
<td>1</td>
<td>8</td>
<td>4096</td>
<td>12</td>
<td>206</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32</td>
<td>2048</td>
<td>49</td>
<td>313</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4096</td>
<td>4096</td>
<td></td>
<td>414</td>
<td>62</td>
</tr>
<tr>
<td>Deep Feed-forward</td>
<td>7</td>
<td>6</td>
<td>2048</td>
<td>9</td>
<td>280</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>12</td>
<td>12</td>
<td>338</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>18</td>
<td>24</td>
<td>379</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>464</td>
<td>61</td>
</tr>
</tbody>
</table>

Key/Value dimension ($d_{kv}$) is fixed to 768 for all models.

- $(L = 8, F = 3)$-model only 2% rel. worse than baseline $(L = 32, F = 1)$
- with 4 times smaller state size. Also confirmed on LibriSpeech.
## Effect of sharing KV

### Perplexity results on TED-LIUM 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dev</td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>No</td>
<td>32</td>
<td>1</td>
<td>49</td>
<td>414</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td>25</td>
<td>395</td>
<td>63</td>
</tr>
<tr>
<td>Deep Feed-forward</td>
<td>No</td>
<td>8</td>
<td>3</td>
<td>12</td>
<td>338</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td>6</td>
<td>333</td>
<td>66</td>
</tr>
</tbody>
</table>

- Up to 5% degradation for the proposed model with deep feed-forward module.
- Almost no degradation for the standard model.
- Counter-intuitive? More components are affected in the standard case; should there be more effect?
- Intuitive? Model with fewer self-attention layers is affected more. Importance of these few layers are greater.
Knowledge distillation

- LSTM requires much less memory in ASR search.
- Knowledge distillation from Transformer to LSTM.

Perplexity results on TED-LIUM 2

<table>
<thead>
<tr>
<th>Model</th>
<th>State size for n positions [K]</th>
<th>#Param. [M]</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>LSTM</td>
<td>16</td>
<td>450</td>
</tr>
<tr>
<td>Teacher</td>
<td>Transformer</td>
<td>$n \times 49$</td>
<td>414</td>
</tr>
<tr>
<td>Student</td>
<td>LSTM</td>
<td>16</td>
<td>450</td>
</tr>
</tbody>
</table>

- 10-12% relative improvements over the baseline LSTM.
- Still behind Transformer teacher, but much smaller memory requirement.

- Compare our another paper Gerstenberger et al. Domain Robust, Fast, and Compact Neural Language Models in Session Language Modeling on Friday.
ASR results: TED-LIUM 2

- Hybrid NN/HMM System (Will be presented this Friday [Zhou & Michel+ 20]).
- First pass with 4-gram + LSTM.
- Lattice rescoring (→) with Transformer.

<table>
<thead>
<tr>
<th>Model</th>
<th>$L$</th>
<th>$F$</th>
<th>Dev</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPL</td>
<td>WER</td>
<td>PPL</td>
<td>WER</td>
</tr>
<tr>
<td>4-gram + LSTM</td>
<td>-</td>
<td>-</td>
<td>64</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>69</td>
<td>6.1</td>
</tr>
<tr>
<td>→ Transformer</td>
<td>32</td>
<td>1</td>
<td>55</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3</td>
<td>56</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3</td>
<td>60</td>
<td>5.7</td>
</tr>
</tbody>
</table>

- Small state Transformer: similar performance to the standard Trafo.
- Require much less memory: 16 GB instead 64 GB for the largest lattice.
Summary

Simple modifications to Transformer layer:

- (1) $F$ feed-forward layers per layer $\rightarrow$ works well.
  We can reduce the total number of layers, thus self-attention layers.
- (2) Sharing key and value matrices $\rightarrow$ Extra reduction in state size $w/$ some degradation if combined with (1)

The 1:1 ratio in the original Transformer is sub-optimal for the state size.

Possible extensions to further reduce memory requirement for search

- All layers do not need to have self-attention.
  $\rightarrow$ Lower/Mid layers do not require self-attention.[Irie & Zeyer$^+$ 19]
  Replace them with static weighted averaging ($w/$ constant state size).
- Combine this with fixed memory size Transformers:
  e.g. Transformer-XL [Dai & Yang$^+$ 19],
  Compressive Transformer [Rae & Potapenko$^+$ 20]
Thank you for your attention.

Please send your questions to:

irie@cs.rwth-aachen.de
References

Transformer-XL: Attentive language models beyond a fixed-length context.

Language modeling with deep Transformers.

Jasper: An End-to-End Convolutional Neural Acoustic Model.

RWTH ASR systems for LibriSpeech: Hybrid vs Attention.

Language models are unsupervised multitask learners.

Compressive transformers for long-range sequence modelling.

The RWTH ASR system for TED-LIUM release 2: Improving hybrid-HMM with SpecAugment.