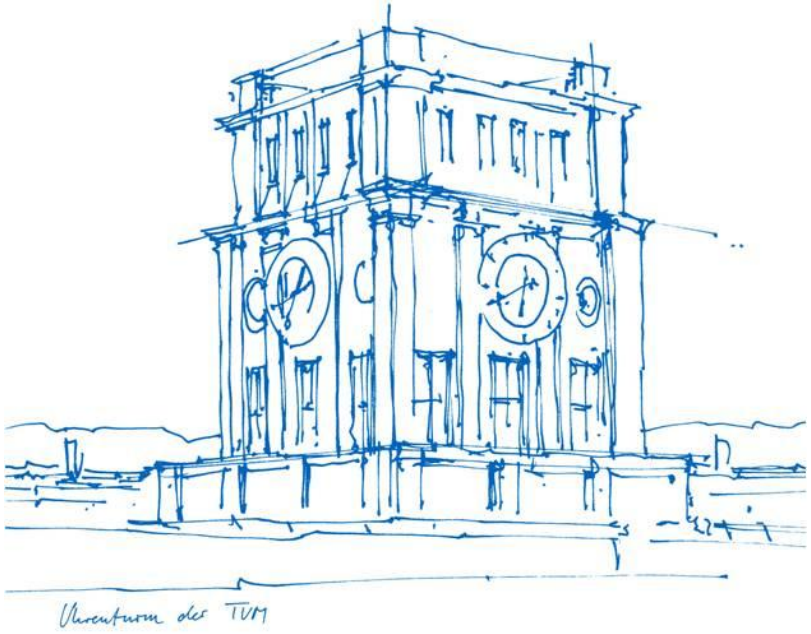
SELECTIVE PARAMETER UPDATING - MEETING 3



Coşku Barış Coşlu

Mato Gudelj

Baykam Say

31 May 2023

Contents

Plan From Last Meeting...

- Implement LST and/or $(IA)^3$ and start doing experiments
  - Implemented both, performed some experiments

- Try to get T5 running and ideally replicate LST
  - T5 up and running, still need to replicate LST

- Add performance-centric metrics to Tensorboard (memory footprint, forward pass latency, etc.)
  - Added logs for some memory stats, still room for improvement

- Brainstorm more approaches
- …keep reading literature

T5 Training Pipeline

Added support for T5 models to our training loop

Sequence-to-sequence model

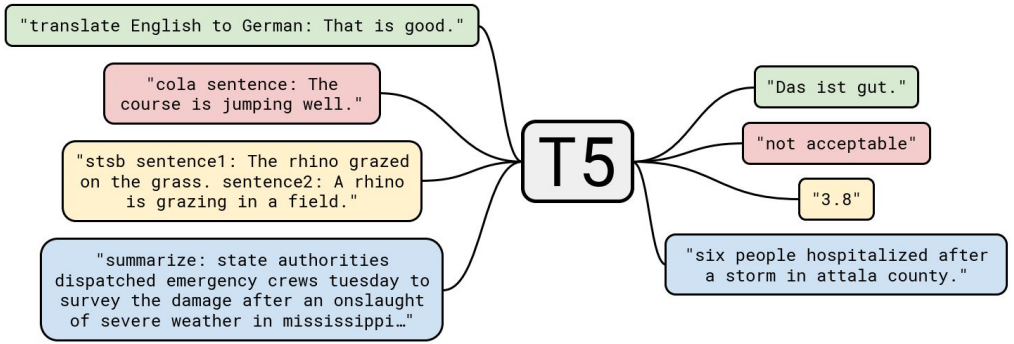
- Requires slightly different approach for preprocessing and inference

Tested T5-base on the SST2 dataset

- Replicated their results

Creates new baseline opportunities

- e.g. Replicating LST results



LST Baseline

Train a ladder side network that takes intermediate activations as input from the backbone network

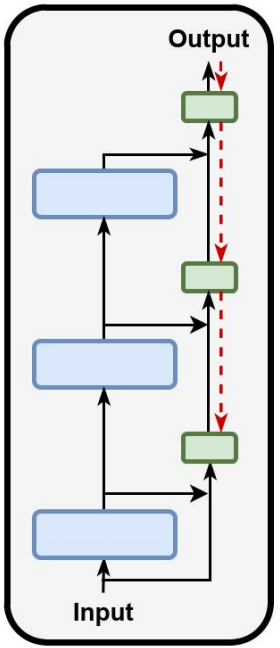
- Reduces memory usage since backbone doesn't have to be backpropagated through

Implemented and tested with a DistilBERT model

- 2-3pp worse results than full fine-tuning, almost 50% VRAM footprint
- Next: T5 and main paper replication

Potential ideas to implement:

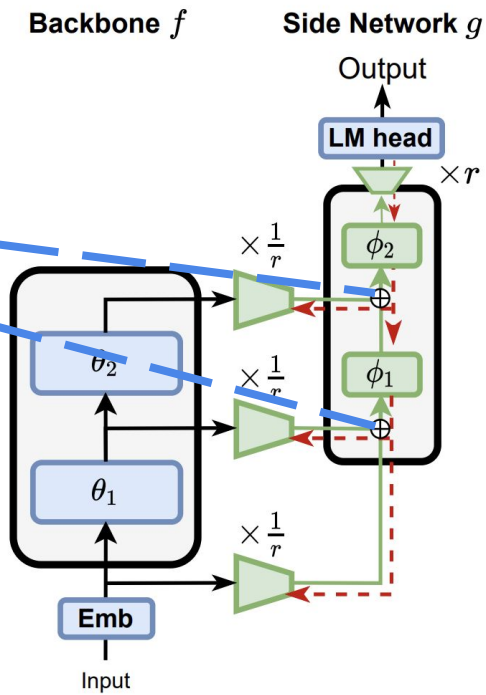
- Make input information available at all LST stages
- Try fusing last couple of layers
- More flexible backbone-ladder fusion



LST experiment – dynamic and attention fusion

(pseudo code, real code contains ugly permutations to get shapes right)

```
pool = F.adaptive_avg_pool1d(downsampled_backbone) # (batch_size, d_side, 1)
fuse = self.side_modules[f"fuse_{i}"](pool) # (batch_size, 1, 1)
fuse = fuse.sigmoid()
output = fuse * output + (1 - fuse) * downsampled_backbone
```



Idea: Instead of using a single learnable mixing parameter, dynamically compute it from context.

Two approaches:

- Average pool over sequence, map with linear
- MHA (same idea as ^, more expensive but more flexible)

Results: Inconclusive, best to compare to T5 LST later

$(IA)^3$ & LoRA Implementation

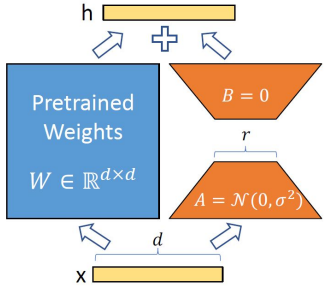
LoRA - inject trainable low rank decomposition matrices into each layer of the Transformer architecture

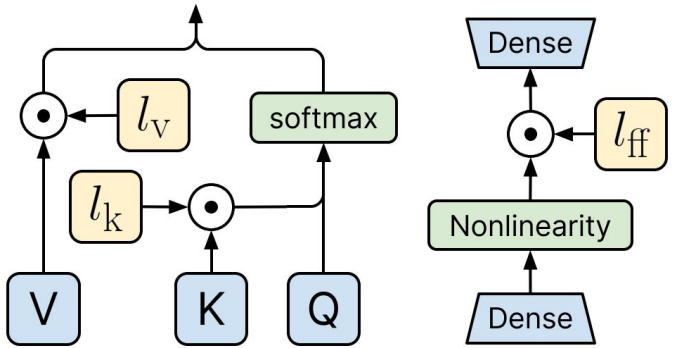
$(IA)^3$ - Scale inner activations by learned vectors

- Adds a relatively small amount of new parameters

Implemented and tested with a DistilBERT model

- Cannot replicate their results since they use billion-scale models





$(IA)^3$ & LoRA Baseline using DistilBERT and SST-2

	# of Trainable Parameters	Accuracy	Train Runtime
Full FT	66M	90.59%	42 min
LoRA	350K	88.19%	34 min
$(IA)^3$	28K	90.02%	14.5 min

Open Issues & Solutions

Issue: Need a way to record experiment results in a consistent way
Solutions: Shared spreadsheet with results, which links to configurations used for experiments and possibly to TensorBoard logs

Issue: It is getting more difficult to find ideas
Solutions: Implement what we have, keep on reading

Plan for the Next Two Weeks

- Experiment with ideas on LST and $(IA)^3$
- Record results of those experiments to present in the next meeting
- Add performance-centric metrics to Tensorboard (memory footprint, forward pass latency, etc.)
- Brainstorm for even more approaches
- …keep reading literature

References

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu: “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, 2019; arXiv:1910.10683

Yi-Lin Sung, Jaemin Cho, Mohit Bansal: “LST: Ladder Side-Tuning for Parameter and Memory Efficient Transfer Learning”, 2022; arXiv:2206.06522

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, Colin Raffel: “Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning”, 2022; arXiv:2205.05638

Hu, Edward J., et al. ‘LoRA: Low-Rank Adaptation of Large Language Models’. ArXiv [Cs.CL], 2021; arXiv:2106.09685