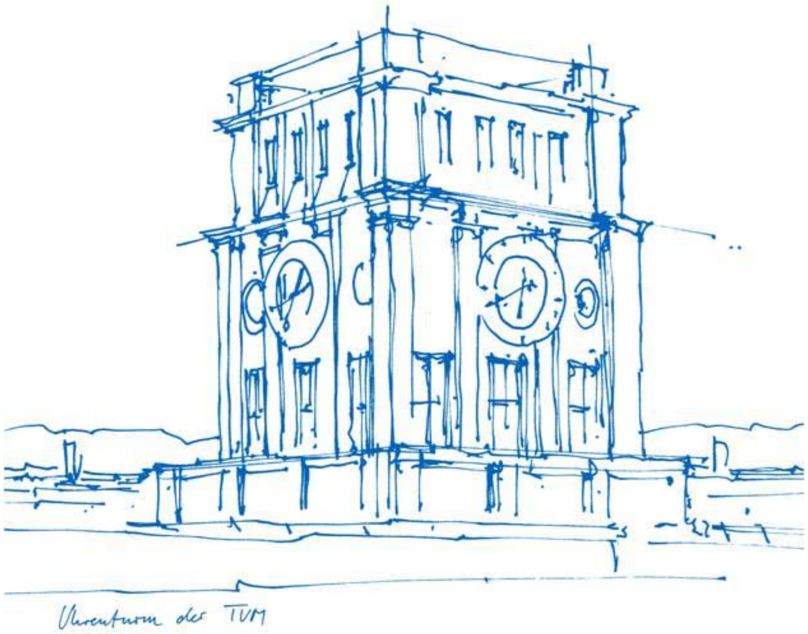
SELECTIVE PARAMETER UPDATING - MEETING 6



Coşku Barış Coşlu

Mato Gudelj

Baykam Say

12 Jul 2023

Contents

Plan from last meeting…

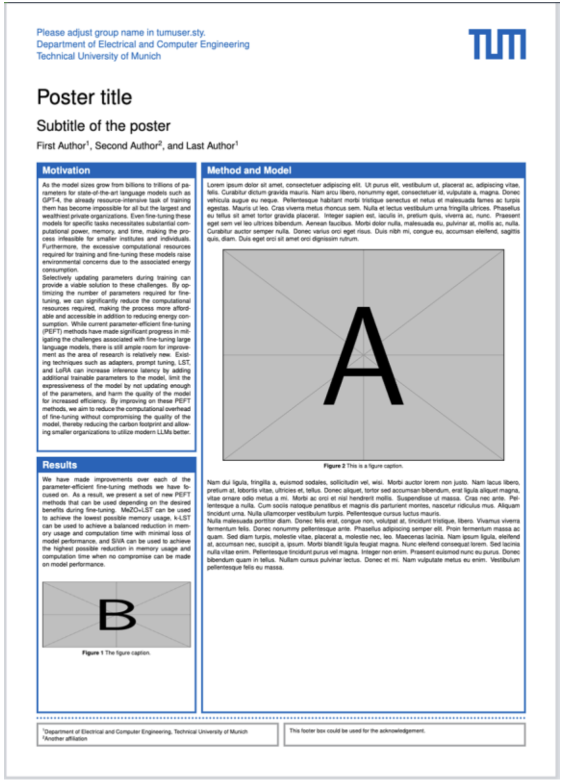
- Do exhaustive hyperparameter tuning
- Implement k-LST ideas
- Start assembling main results for poster and report
- Brainstorm more approaches
- …keep reading literature

Poster

- Began writing text
  - ‘Motivation’ is complete
  - Intro to ‘Results’ is complete

- Need to prepare visuals

- Keep or change the layout of the template?



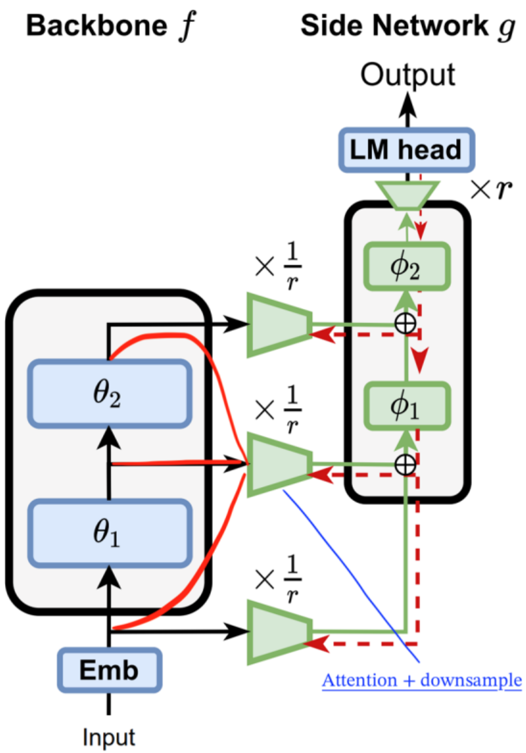
k-LST

Tried unfreezing the last couple of backbone layers

- Stronger overfitting observed
- No conclusive results, needs more hyperparam tuning
- TODO: Run together with LC dropout

Implemented support for pretraining LST on masked LM tasks

- Haven’t had time to run it
- Would take >1 day for a minimum number of steps
- Pretraining on the whole corpus unfeasible on available hardware
- Original RoBERTa: 1024 (!!) V100s for 1 day
    - 500k steps @ 8k batch size
    - => 250 mil steps at 16 batch size (what we mostly use)
    - We could maybe do 0.1% of that...

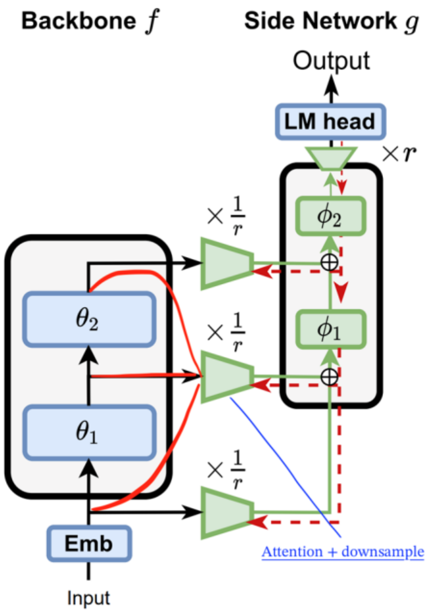


k-LST + Ladder connection dropout

Implemented ladder connection dropout

- During training draw k-length mask of probability p, mask the connection(s)
- Observed less overfitting => Assumption was correct
- Slightly better results than without dropout

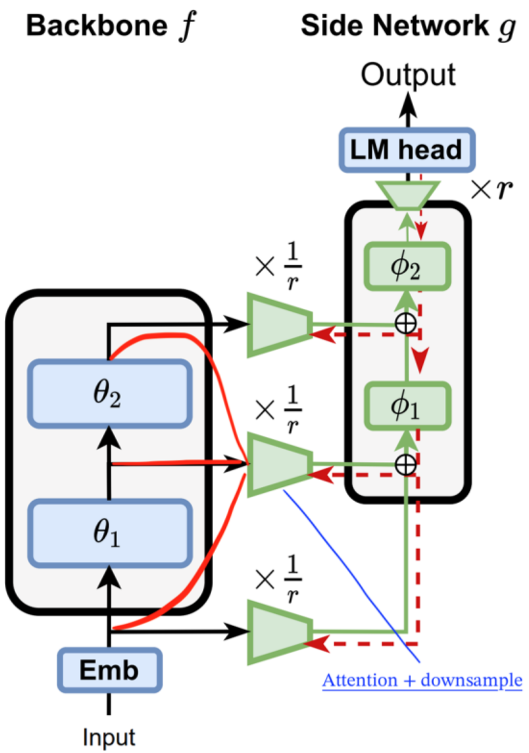
Method	SST-2 Accuracy	SST-2 F1
RoBERTa-large + LST	93.12%	93.10%
RoBERTa-large + 9-LST	94.50%	94.62%
RoBERTa-large + 9-LST + Dropout	94.84%	95.00%
RoBERTa-large + Full FT	96.40%	-



k-LST - What's left?

What's left?

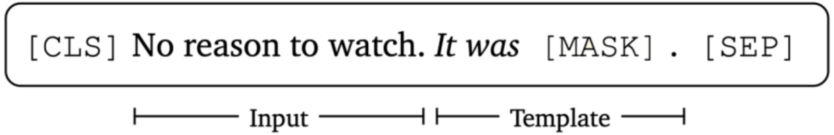
- More hyperparameter tuning
- Smarter weight initialization
- Pretrain on Masked Language Modelling prior to FT
- Start fixing seeds for replicable final results
- Diagram for poster and report
- From last presentation: Try different downsampling strategies
- Maybe more ideas if they come up…



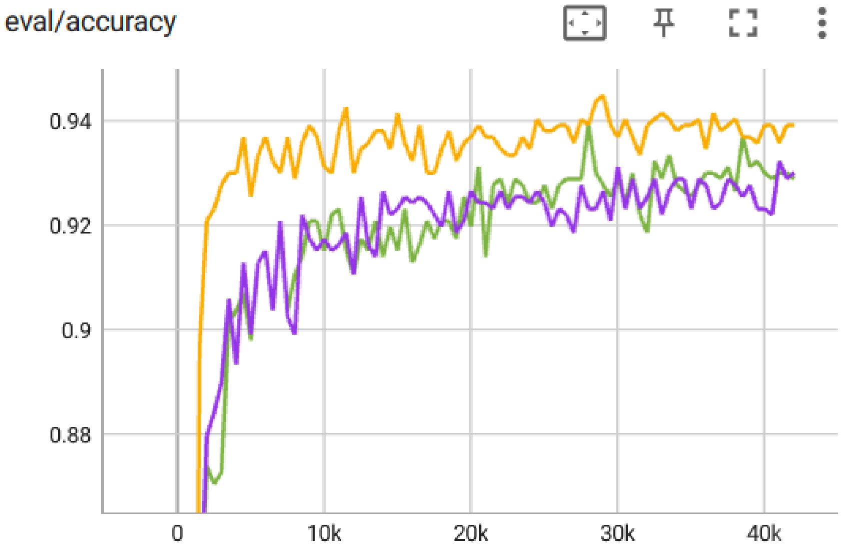
MeZO + LST

MeZO + LST not better than standard LST
New Idea: Modify the inputs.

- Make them similar to MeZO fine-tuning by adding the template part.



- MeZO + LST now improves over standard LST
- Problem: Slight improvement for much effort: 94.5% vs 93.92% max.
- Even smaller difference on LST configurations that perform better (e.g. 9-LST w. dropout)



MeZO + Weight Update Skipping

Idea:

- First fine-tune using MeZO
- Then fine-tune only the last few layers with backpropagation

MeZO seems to be improving

Results similar to LST:
Tradeoff memory-accuracy

	SST-2 Accuracy / F1	Peak Memory (MiB)
Last 1 Layer	92.66 / 92.69	1634
MeZO + Last 1 Layer	93.81 / 93.91	1804
Last 3 Layers	94.15 / 94.33	2122
MeZO + Last 3 Layers	94.84 / 94.86	2122
MeZO + LST	94.50 / 94.58	2044
MeZO + 9-LST	94.95 / 95.09	2428

MeZO Next Steps

We have trained MeZO for 24000 steps and used that checkpoint for all of our experiments.

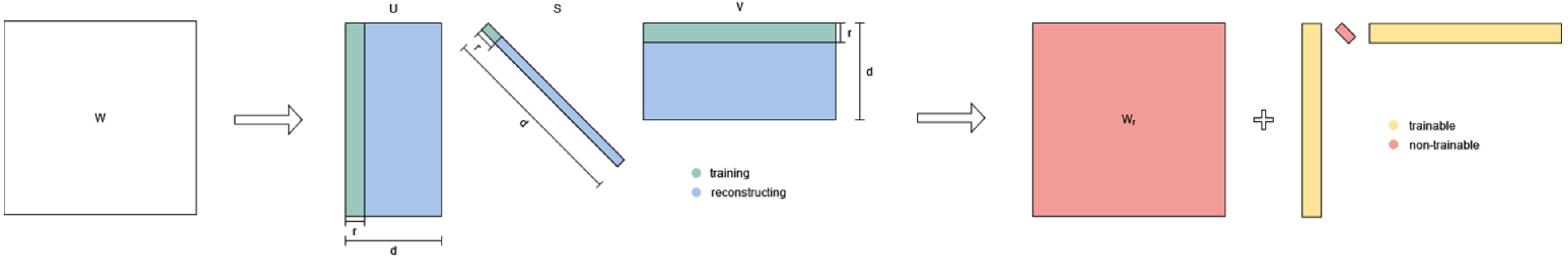
It would be interesting to know:

- How much more can MeZO improve on its own
  - For now, it gets about 90.5% accuracy
  - Would be nice to have for comparison with MeZO + LST and MeZO + WUS
  - Also can potentially further improve those methods

- Can earlier MeZO checkpoints provide a similar improvement to LST/WUS
  - Could greatly reduce the time requirements for those methods

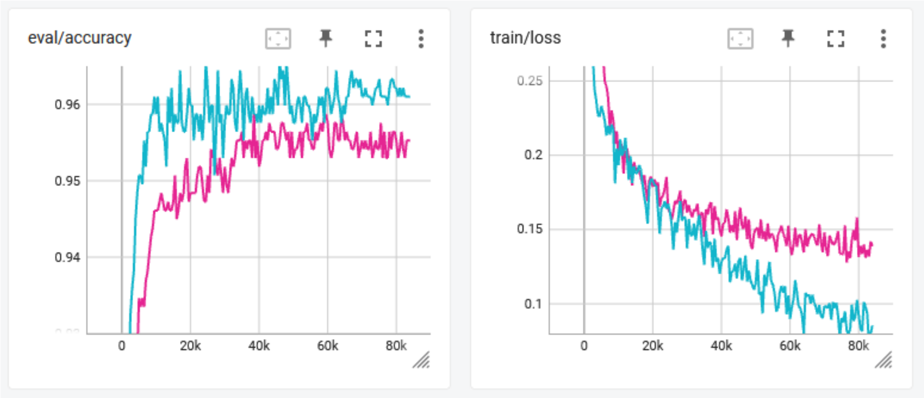
Singular Value Adaptation (SiVA)

- Another method to update weight matrices in a lower rank
  a. Decompose the original weight matrix into U, S, V matrices using SVD
  b. Split U, S, V into smaller training (using the highest r singular values) and reconstruction (rank d) matrices
  c. Reconstruct the original model without using the trainable part
  d. Train the lower rank U and V matrices
  e. During forward pass, reconstruct trainable U, S, V and add it to the original matrix
- Since there is no random initialization. it should be faster than LoRA



SiVA Results

- Much faster convergence than LoRA
- Slightly faster training times (5%, 62.63 samples per second vs 59.6)
- As good accuracy/f1 as full fine-tuning, (96.56)



— SiVA

— LoRA

SiVA Results

Method	Accuracy	F1 Score	Memory	Training time
Full FT	96.44%	96.53%	8195 MB	3.77 hrs*
LoRA	96.22%	96.29%	4526 MB	2.97 hrs
SiVA	96.56%	96.65%	4529 MB	2.83 hrs
SiVA-kv**	95.76%	95.86%	2920 MB	2.32 hrs

* estimate based on a different device and lower number of epochs
**only train key and value layers in self attention for lower memory consumption and faster training

Open Issues & Solutions

- Issues
    - The structure of the poster gets too complex
    - Can’t compare our methods with each other
- Solutions
    - Split the methods section into 4: introduction, mezo like models, k-lst, and SiVA
    - Fine-tune on Colab with the same GPU and number of epochs
        - Need to use less than 10 epochs to be able to fully fine-tune Roberta large without getting kicked out

Plan for the Next Two Weeks

- Finalize experiments & results
  - Train on the same hardware to get speed comparisons
- Finish the poster
- Work on the report