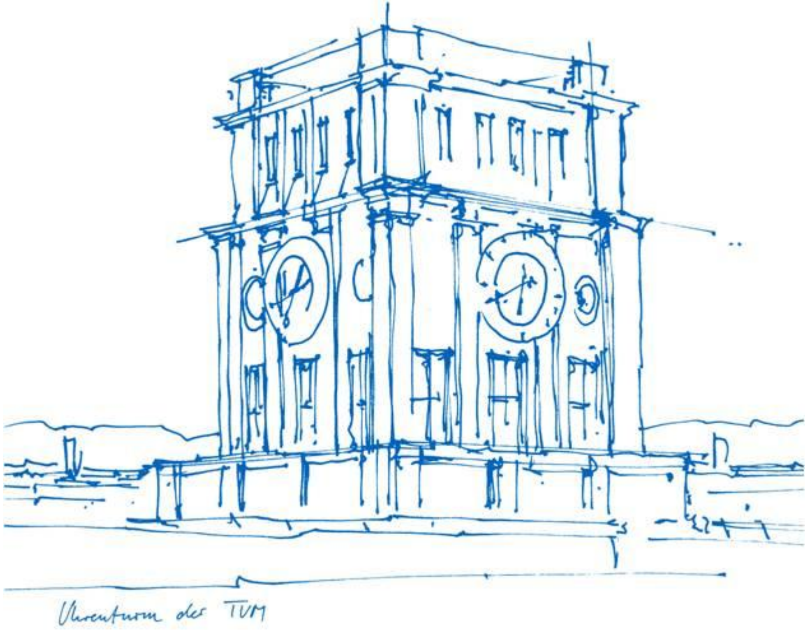
SELECTIVE PARAMETER UPDATING - MEETING 2



Coşku Barış Coşlu

Mato Gudelj

Baykam Say

17 May 2023

Contents

1. Plan from last meeting
2. Config system
3. Baseline and new dataset
4. Potential approaches
5. Open Issues & Solutions
6. Plan for the Next Two Weeks

Plan from last meeting…

- Train a full baseline, compare with known numbers to validate training code
- Extend training code with proper logging (TensorBoard)
- Extend training code with an extensible config system
- Brainstorm potential approaches
  - By next meeting shortlist a few and implement at least one
- …keep reading literature

Plan from last meeting…

- Train a full baseline, compare with known numbers to validate training code
- Extend training code with proper logging (TensorBoard)
- Extend training code with an extensible config system
- Brainstorm potential approaches
    - By next meeting shortlist a few and implement at least one
- …keep reading literature

Extensible config system

- YAML based
- Encodes all information needed to replicate a training run
- Easy to extend with new datasets
- Adding a new strategy is 1 LOC
    - Currently freezing strategies
    - In the future: additive PEFT, etc.

code > ! example.yaml

```yaml
%YAML 1.2
---
dataset:
    name: "squad"
    n_train: 5000
    n_val: 500

model:
    base_model: "distilbert-base-cased"

freeze:
    strategy: "all_but_last_n"
    args:
        n: 1

train:
    weight_decay: 0.01
    num_train_epochs: 5
    learning_rate: 0.00002
    per_device_train_batch_size: 16
    per_device_eval_batch_size: 16
    output_dir: "results"
    evaluation_strategy: "epoch"

evaluate:
    metric_function: "none"
...
```

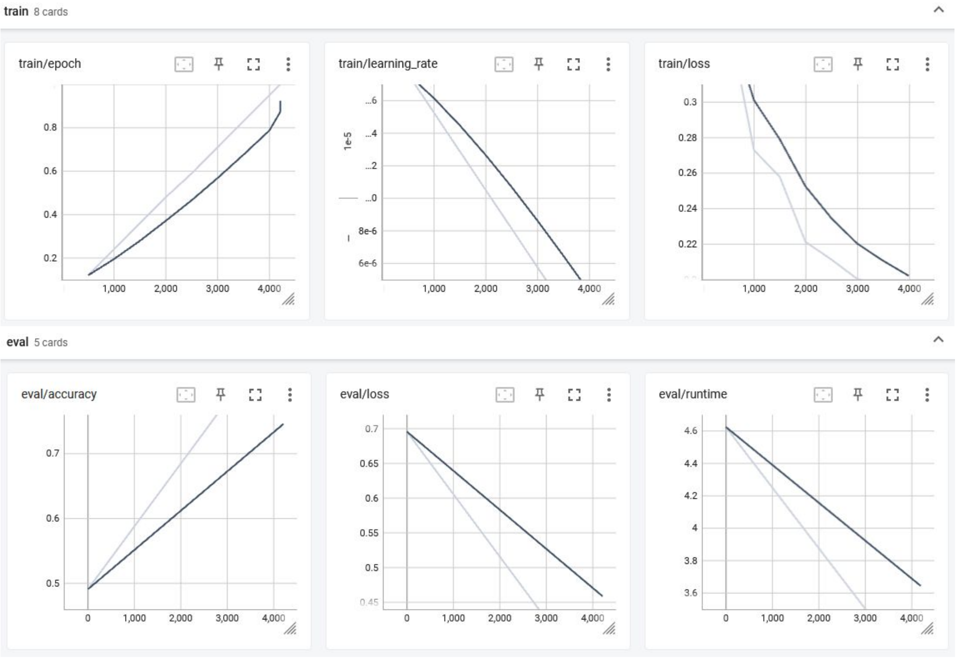
Baseline ~~SQuAD~~, **SST-2**

- We had trouble replicating SQuAD results
  - Training takes too much time and memory
  - Original DistilBERT paper does not provide configuration
- This caused a bit of a delay
- Went another route: added support for SST-2
  - Verified pipeline works
  - Replicated results from the paper “Freeze And Reconfigure”
    - Paper result: 91.1%
    - Baseline result: 90.7%

code > ! sst2.yaml

```yaml
%YAML 1.2
---
dataset:
    name: "sst2"

model:
    base_model: "distilbert-base-cased"

freeze:
    strategy: "none"

train:
    weight_decay: 0.01
    num_train_epochs: 5
    learning_rate: 0.00002
    per_device_train_batch_size: 16
    per_device_eval_batch_size: 16
    output_dir: "results"
    evaluation_strategy: "epoch"

evaluate:
    metric_function: "accuracy"
...

```

TensorBoard



Brainstorming potential approaches

- Shortlisted 3 potential approaches
  - Applying a different Kronecker product approximation (KoPA [1]) for PEFT as in KronA [2].
  - Iterate on LST [3]
    - Try different input strategies
    - Try fusing last couple of layers
  - Iterate on $(IA)^3$ [4]
    - Data-dependent weights of K/V/FF vectors, inspired by Involution [5]
- LST interesting due to low VRAM footprint, $(IA)^3$ due to simplicity and high performance
- Due to baseline delays didn’t manage to try an approach yet

Open Issues & Solutions

- Issues
  - T5 model family support needed to compare to LST
    - OOM locally even with the smallest model (Full FT)
  - Not feasible to replicate $(IA)^3$ paper results
    - Very large models (3B)
- Solutions
  - T5/LST
    - Might not be an issue for LST, as it uses significantly less VRAM than Full FT.
    - If problem persists, move to Colab for these experiments
  - $(IA)^3$
    - Find other literature with reliable $(IA)^3$ numbers on smaller models

Plan for the Next Two Weeks

- Implement LST and/or $(IA)^3$ and start doing experiments
- Try to get T5 running and ideally replicate LST
- Add performance-centric metrics to Tensorboard (memory footprint, forward pass latency, etc.)
- Brainstorm more approaches
- …keep reading literature

References

[1] Chencheng Cai, Rong Chen, Han Xiao: “KoPA: Automated Kronecker Product Approximation”, 2019; arXiv:1912.02392

[2] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, Mehdi Rezagholizadeh: “KronA: Parameter Efficient Tuning with Kronecker Adapter”, 2022; arXiv:2212.10650

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

[4] 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

[5] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, Qifeng Chen: “Involution: Inverting the Inherence of Convolution for Visual Recognition”, 2021; arXiv:2103.06255