MMlib: Efficiently Managing Deep Learning Models in a Distributed Environment

Nils Strassenburg, Ilin Tolovski, Tilmann Rabl

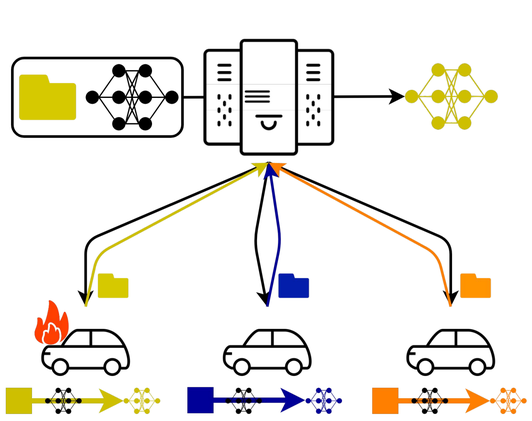
Contribution

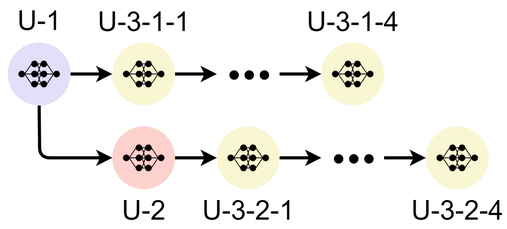
» We present **three approaches for saving and recovering** exact representations of DL **models**.

» We **evaluate** all approaches in distributed environments for different model architectures, model relations, and datasets to **discuss trade offs.**

» We **bundle all approaches together** with a probing tool in a **Model Management Library (MMlib).**

Use Cases





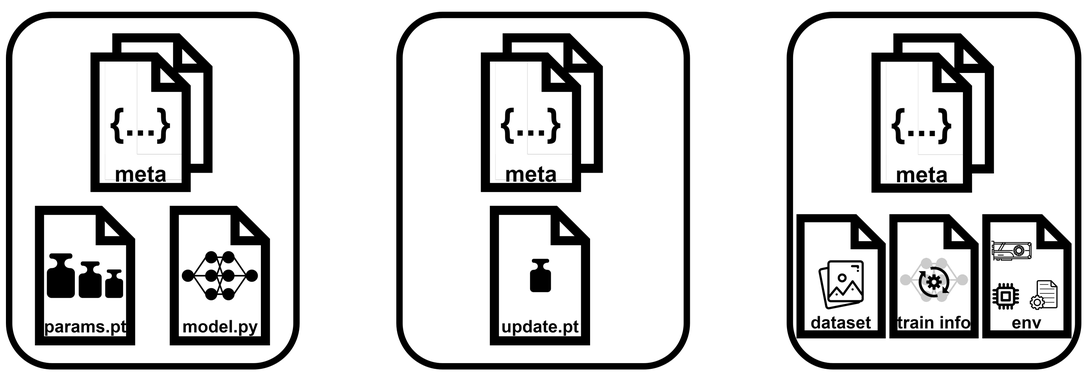
U-1: distribute initial model

U-2: update model on server

U-3: update model locally

U-4: recover model

Approaches



Baseline: complete model snapshots

Update: difference to previous model

Provenance: detailed training information

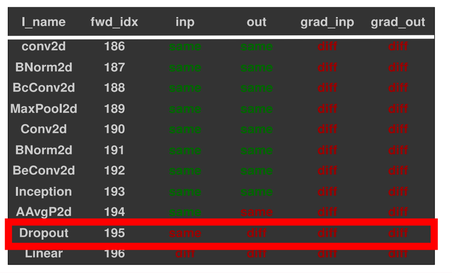
Reproducibility

To reproduce training ...

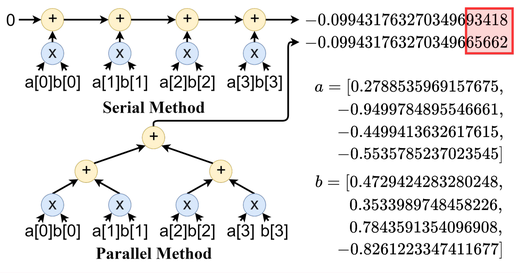
» ... use same **data, model, and parameters**

» **... make intentional randomness deterministic**
→ set seeds

» **... make floating point computation deterministic**
→ use same software and hardware
→ only allow for deterministic implementation

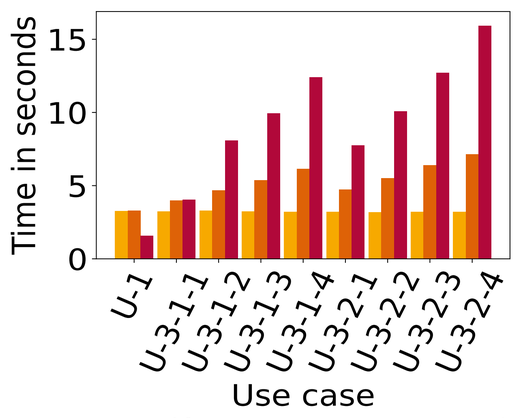


l_name	fwd_idx	inp	out	grad_inp	grad_out
conv2d	186	same	same	diff	diff
BNorm2d	187	same	same	diff	diff
BcConv2d	188	same	same	diff	diff
MaxPool2d	189	same	same	diff	diff
Conv2d	190	same	same	diff	diff
BNorm2d	191	same	same	diff	diff
BeConv2d	192	same	same	diff	diff
Inception	193	same	same	diff	diff
AAvgP2d	194	same	same	diff	diff
Dropout	195	same	diff	diff	diff
Linear	196				



Performance Evaluation

Time-to-Recover

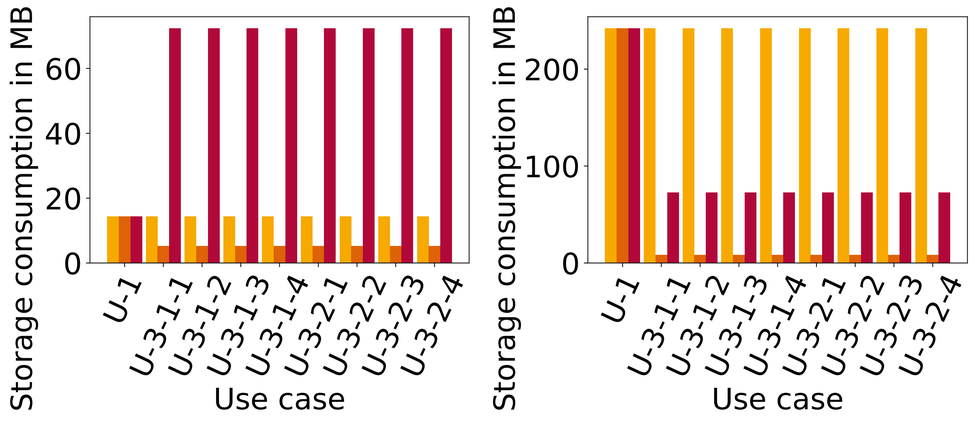


» **Baseline: constant**; other approaches **staircase pattern**

» **Update: slowly increasing**

» **Provenance:** increase dependent on **training time**

Storage Consumption



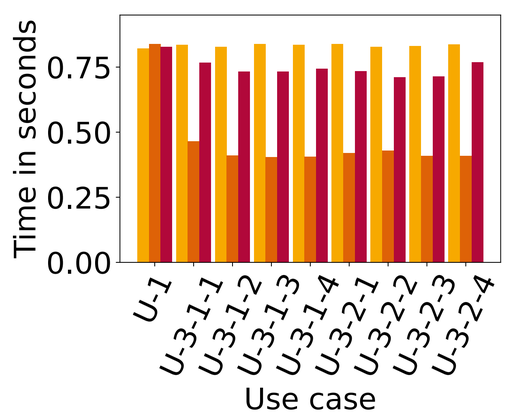
» Storage consumption depends on #parameters

» **Update:** up to **95.6%** improvement over baseline

» **Provenance:** up to **70.0%** improvement over baseline

» Critical factor: **dataset size vs. #parameters**

Time-to-Save



» time-to-save dependent on **amount of data**

Baseline Update Provenance

Summary

» Choose **Baseline when optimize for time-to-recover**, otherwise trade in for reduced storage consumption.

» **Storage consumption**: best approach depends on **model relation, number of parameters and dataset size.**

» When optimizing for **time-to-save** choose the approach with the **lowest storage consumption.**

Work published as “Efficiently Managing Deep Learning Models in a Distributed Environment” @ EDBT 2022 | 29th March - 1st April, 2022 | Edinburgh, Scotland

Acknowledgements: this work was partially funded by the European Union’s Horizon 2020 research and innovation programme (ref. 957407)

Nils Strassenburg
Data Engineering Systems Group
Hasso Plattner Institute, Potsdam, Germany
E-Mail: nils.strassenburg@hpi.de

HPI Hasso Plattner Institut