Towards Robust and Efficient AI at Scale

Charlotte Debus, Markus Götz and others | 14. Juli 2023

The Fundamental Questions of Science

Karlsruher Institut für Technologie

What is this?

What does it do?

Machine Learning in Science and Engineering

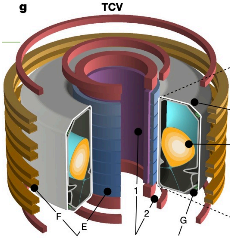
Key components

(ML-based) time-series analysis and forecasting has many applications in

- **Science**
  - Climate and energy research
  - High-energy and particle physics
  - Biology and medicine

Source: www.dena.de

- **Engineering**
  - Condition monitoring
  - Automated system control
  - Process optimization and predictive maintenance



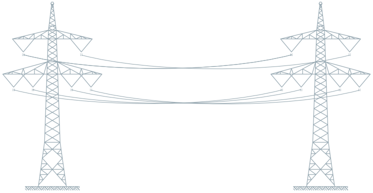
Modified from [1]

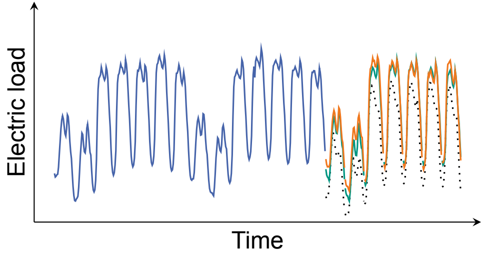
[1] Degrave, et al. "Magnetic control of tokamak plasmas through deep reinforcement learning."

Time series in energy research

Transformer networks for electric load forecasting

- **Power generation** must equal load
- Increasing generation **volatility** due to contributions from **renewable energy**
- Precise **prediction** allow precise production planning
- Reduction of economic and ecologic waste
  - **Rough estimate for Germany:** 0.1% of 500 TWh/yr[2] @ 0.35 €/kWh ≈ 175 Million €/yr ≈ 15.000 households





[2] https://www.umweltbundesamt.de/daten/energie/stromverbrauch

Time series in energy research

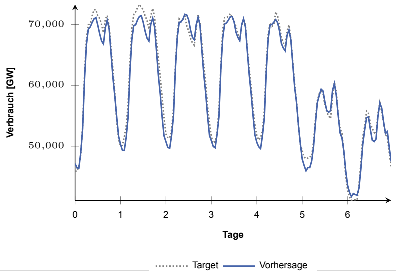
Transformer networks for electric load forecasting

- Deep learning methods yield state-of-the-art accuracies over conventional statistical methods
  - RNN, LSTM
  - **Transformer**
- **ReCycle**: Residual Cyclic Transformers
  - Leverage patterns in time-series data and prior knowledge
  - Account for **special cases** via metadata

But: Application of ML in complex systems and critical infrastructures requires **robustness**

Mean absolute percentage error (MAPE)

	Electricity [%]	Water [%]	Temperature [%]
ReCycle	2.94	13.73	2.44
FEDFormer	3.95	15.64	2.46
Transformer	4.08	15.70	2.58



Machine Learning in Science and Engineering

Key components

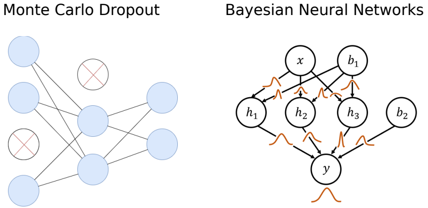
Robustness

- Uncertainty quantification

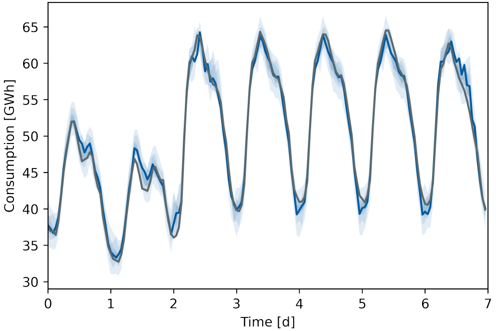
Robust AI
Uncertainty quantification

- Data uncertainty (aleatoric)

- **Model uncertainty** (epistemic)
  - Ensemble Methods, e.g. Monte Carlo Dropout
  - Bayesian Neural Networks



Modified from [3] and [4]



[3] https://towardsdatascience.com/monte-carlo-dropout-7fd52f8b6571, [4] https://towardsdatascience.com/why-you-should-use-bayesian-neural-network-aaf76732c150

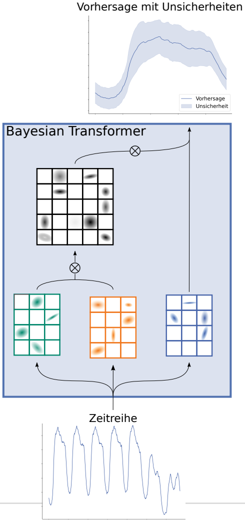
Robust AI
Uncertainty quantification in Transformer networks

Bundesministerium für Bildung und Forschung

KIT Karlsruher Institut für Technologie

- Efficient UQ-Methods for **Transformer architectures**
- UQ introduces **additional compute load**
  - Ensemble methods run multiple times at inference
  - Bayesian NN multiply amount of trainable parameters (at least double)

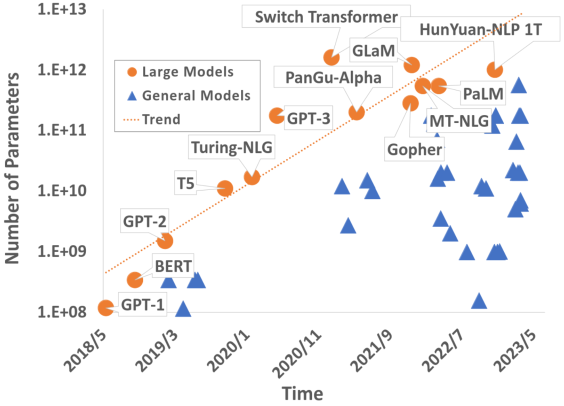
→ Speed up training and inference



Transformers and the Curse of Dimensionality

Karlsruher Institut für Technologie

- More **data** yields better predictive performance
- More **parameters** yield better predictive performance
- The attention mechanism scales with $\mathcal{O}(N^2)$ in **sequence lengths**



Source: [5]

[5] Shen, Li, et al. "On Efficient Training of Large-Scale Deep Learning Models: A Literature Review."

Hardware and Trainingtimes

Transformer-based models in Science and Engineering

Karlsruher Institut für Technologie

Model	Application	Hardware	Trainingszeit
AlphaFold2	Protein folding	128$\times$ TPU	3 Weeks
DESERT	Drug Design	32$\times$ V100	2 Weeks
AlphaTensor	MatMul algorithm	64$\times$ TPUv3	*1 Week*
FourCastNet	Weather forecasting	64$\times$ A100	1 Day
Pangu-Weather	Weather forecasting	192$\times$ V100	15 Days
ClimaX	Weather forecasting	80$\times$ V100	?

Machine Learning in Science and Engineering

Key components

Robustness

- Uncertainty quantification

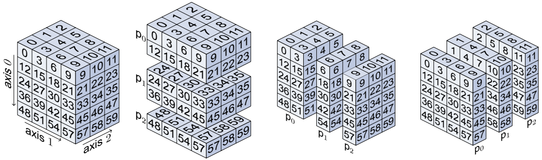
Scalability

- Data and model parallelism

Scalable AI

Distributed, GPU-accelerated tensor computations and ML

- HeAT[6]: **Distributed** numerical programming framework
  - PyTorch as node-local eager execution engine
  - MPI for inter-node communication
- Low-level NumPy-like **array computations**
  - Statistics
  - Basic linear algebra
- High-lever **machine learning** algorithms
  - Clustering
  - Regression
  - Distributed AD (work-in-progress)



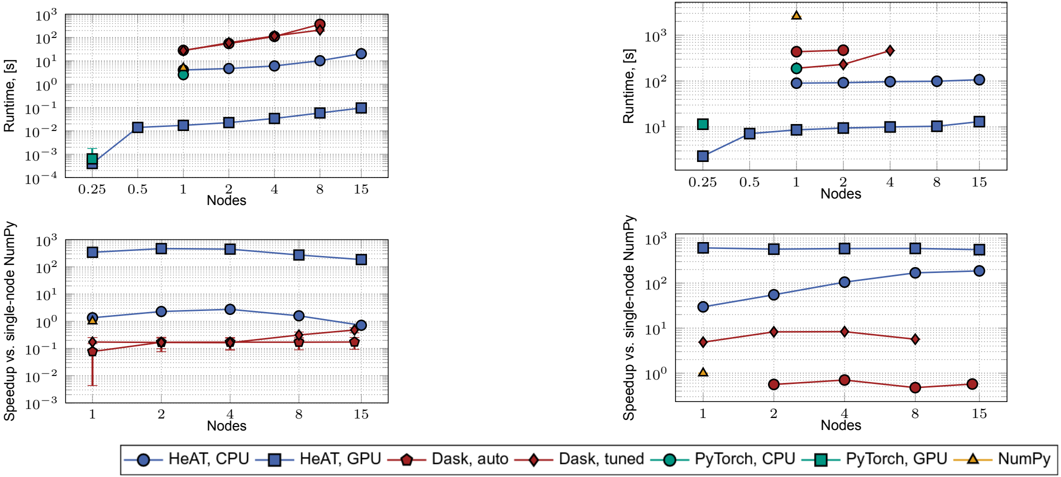
[6] Götz, M., Debus, C., et al. (2020). HeAT–a Distributed and GPU-accelerated Tensor Framework for Data Analytics. In 2020 IEEE International Conference on Big Data

Scalable AI

Distributed, GPU-accelerated tensor computations and ML

Distributed `cdist`

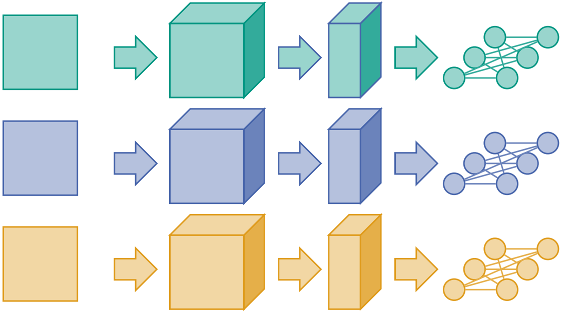
Distributed `k-means`



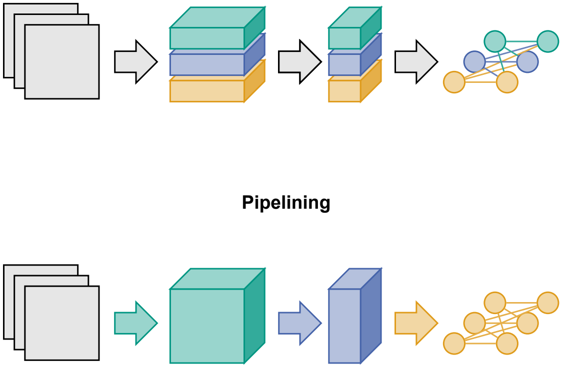
Scalable AI

Parallel neural networks

Data Parallel



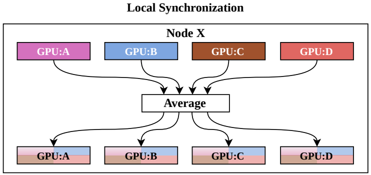
Model Parallel

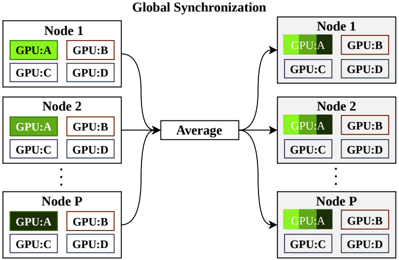


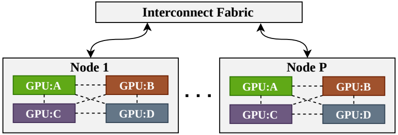
Scalable AI

Parallel computing to accelerate neural network training

- Traditional **data-parallel NN**
  - Synchronous: Blocking communication
  - Asynchronous: Stale gradients
- **DASO**: Distributed Asynchronous and Selective Optimization[7]
  - Leverage **multi-GPU architectures**
  - → Hierarchical and asynchronous communication scheme
  - Adjustable global synchronization rate





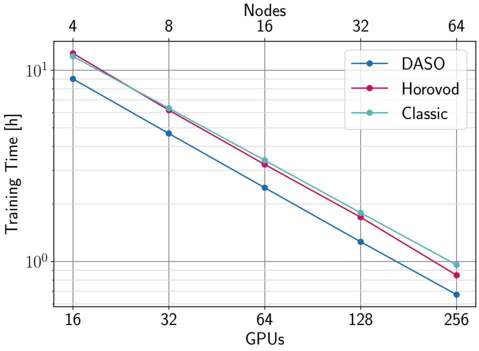


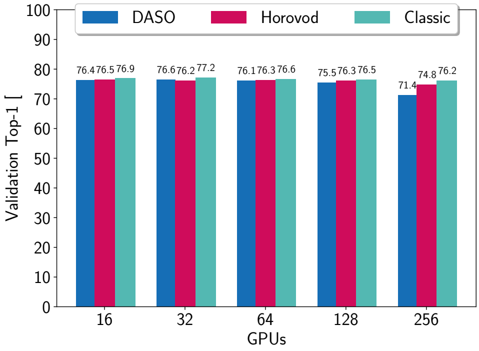
[7] Coquelin, D., Debus, C., et al. (2021). Accelerating Neural Network Training with Distributed Asynchronous and Selective Optimization (DASO), In Springer Big Data

Scalable AI

Parallel computing to accelerate neural network training

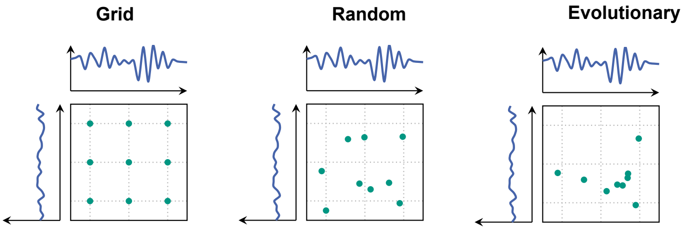
- Training with DASO up to 34% faster than Horovod
- Comparable accuracy up to 128 GPUs





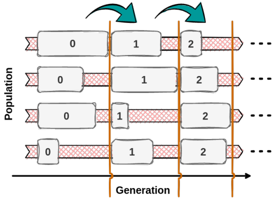
Scalable AI

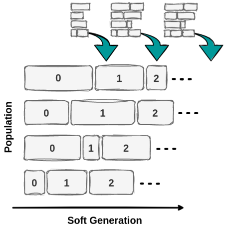
Hyperparameter Optimization

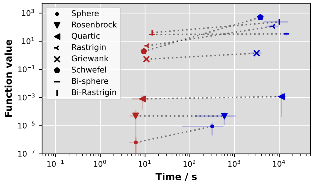


Propulate[8]: Evolutionary Hyperparameteroptimierung

- Imitate natural selection
- Asynchronous communication







[8] Taubert, et al. “Massively Parallel Genetic Optimization Through Asynchronous Propagation of Populations.”

Deep Learning at Scale
The unsustainable hunt for ever better predictive performance

Model	Parameter	Hardware	Time[h]	CO_2 [kg]
Transformer	213M	8×P100	84	87
NAS		1× TPUv2 (83×P100)	32,623 (274,120)	284,019
BERT	110M	163×TPU (643×V100)	96 (79)	652
GPT3	175B	10,000×V100	355	502,000
Gopher	280B	4096×TPUv3	920	352,000
OPT	175B	992×A100	?	70,000

Luccioni, et al. “Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model”
Strubell, et al. “Energy and policy considerations for deep learning in NLP”

Deep Learning at Scale

The unsustainable hunt for ever better predictive performance

Model	Parameter	Hardware	Time[h]	CO_2 [kg]
Transformer	213M	8×P100	84	87
NAS		1× TPUv2 (83×P100)	32,623 (274,120)	284,019
BERT	110M	163×TPU (643×V100)	96 (79)	652
GPT3	175B	10,000×V100	355	502,000
Gopher	280B	4096×TPUv3	920	352,000
OPT	175B	992×A100	?	70,000

Producer	CO_2 [kg]
Flight NY $\leftrightarrow$ SF (pP)	900
Human life (world)	5,000
Human life (USA)	16,400
Car (incl. fuel)	57,152

Luccioni, et al. “Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model”
Strubell, et al. “Energy and policy considerations for deep learning in NLP”

Deep Learning at Scale
The unsustainable hunt for ever better predictive performance

Karlsruher Institut für Technologie

- AI **community** research **focus** mainly on improving **prediction metrics**, e.g., accuracy
  - **Disregards** any economical, ecological and social **costs**
- **Costs** stem mostly from amount of **work required** to achieve the results
  - **Logarithmic** relation between **model performance** and its **capacity**

→ **“Buy”** better models through **inefficient** utilization

Machine Learning in Science and Engineering

Key components

Robustness

- Uncertainty quantification

Scalability

- Data and model parallelism

Efficiency

- Energy consumption

Efficient AI

Raising awareness in the ML research community

AI HERO Hackathon[9]: High-precision AI models with lowest possible energy consumption

- 2 Use-Cases: *Energy* and *Health*
- Develop a model to tackle the application challenge of the use-case
- **But**: make it **energy efficient**,
  aka use as little electricity as possible
  - **Development & Training**: Electricity consumption of all submitted jobs were tracked
  - **Inference**: Inference on test data incl. energy measurements

[9] https://ai-hero-hackathon.de/

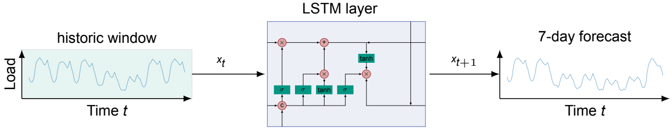
Efficient AI

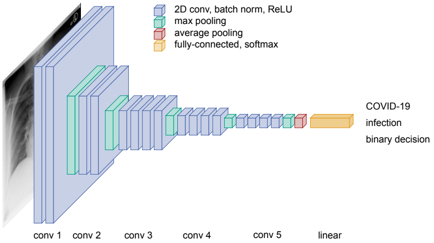
Energy consumption measurements of AI workloads[10]]

- Use-case **Energy** – 7-day time series **load forecast**

- Use-case **Health** – Binary COVID **x-ray classification**

- Experiments
  - **Energy** during **development**
  - **Energy** during **inference**
  - Predictive **performance** on hold-out **test set**





[10] René Caspart, et al.,(2022) "Precise Energy Consumption Measurements of Heterogeneous AI Workloads", ISC Workshop HPC on Heterogeneous Hardware (H3)

Efficient AI

Energy consumption measurements of AI workloads

- Experiments ran on **HAICORE@KIT**
  - Single-node, parallel file system
  - CPU: Intel Xeon Ice Lake
  - Acclerator: Nvidia Tesla A100 GPU
  - XClarity Controller (XCC) power sensors
- Major measurement **insights**
  - Energy **estimation highly imprecise**
  - **Use GPUs** – CPUs only match for small sequential RNNs
  - **Jupyter** environments have **neglectable** energy **overhead**

Use Case Energy

	Node	Consumption [kJ]	Average power draw [W]	Runtime
Training	GPU	720.5	667.8	00:17:59
	CPU-mix	4 599.6	628.8	02:01:54
	CPU-only	2 878.3	397.0	02:00:50
Prediction	GPU	210.9	627.7	00:05:36
	CPU-mix	365.2	622.1	00:09:47
	CPU-only	208.6	359.1	00:09:41
Jupyter	GPU	756.4	646.4	00:19:30

Use Case Health

	Node	Consumption [kJ]	Average power draw [W]	Runtime
Training	GPU	772.3	793.7	00:16:13
	CPU-mix	106 436.3	635.1	2-04:51:00
	CPU-only	64 795.7	385.5	1-22:41:21
Prediction	GPU	22.8	437.9	00:00:52
	CPU-mix	3 324.3	633.3	01:27:29
	CPU-only	1 951.2	373.1	01:27:10
Jupyter	GPU	756.4	781.4	00:16:08

Efficient AI

Electricity consumption as quantifyable metric

Karlsruher Institut für Technologie

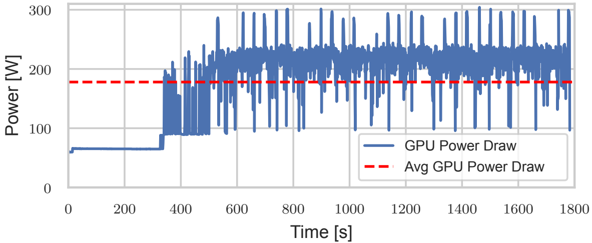
perun[11]: Benchmarking Energy Consumption of High-Performance Computing Applications

- Energy **monitoring** in data-intensive application
  - Tools and methods available

- Energy **optimization**
  - Dynamic Voltage and Energy Scaling
  - Power limiting of hardware components during code execution

```
import perun

@perun.monitor(data_out="results/", format="json")
def expensive_computation(input_args):
    pass
```



PERUN

[11] Guttierez, et al. "perun: Benchmarking Energy Consumption of High-Performance Computing Applications"

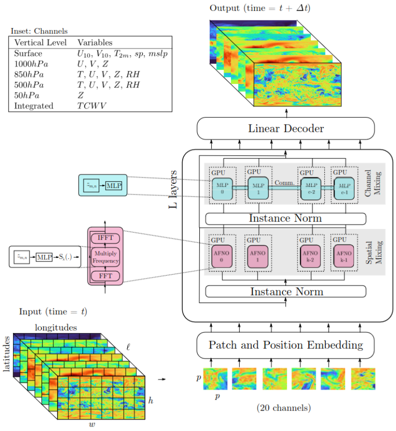
Efficient AI

Energy Efficiency and Performance of AI at Scale

FourCastNet[12]

- **Surrogate model** for numerical weather predictions
  - Adpative Fourier Neural Operators
  - Vision Transformer
- **Model-parallelism** within a node (via NVLink)
- **Data-parallelism** across nodes (via Interconnect Fabric)

→ Measure **energy consumption** and **computational performance**



Modified from [12]

[12] Pathak, et al. "Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators.

Robust, Efficient and Scalable AI

What's next?

Robustness	Scalability	Efficiency
Uncertainty quantification	Data and model parallelism	Energy consumption

Robust, Efficient and Scalable AI

What's next?

Robustness

- Uncertainty quantification

Scalability

- Data and model parallelism

Efficiency

- Energy consumption

Data Uncertainty

Performance Modeling for AI

Scalable Anomaly Detection

Feed-forward-only Training

AI Engineering for large-scale distributed NN

Sparsity and Numerical Linear Algebra in AI

Thank you for your attention!

Thank you for your attention!

Questions?