Deep Learning is More Than Just Dense Matrix Multiplication

Institute of Science Tokyo / RIKEN
Rio Yokota

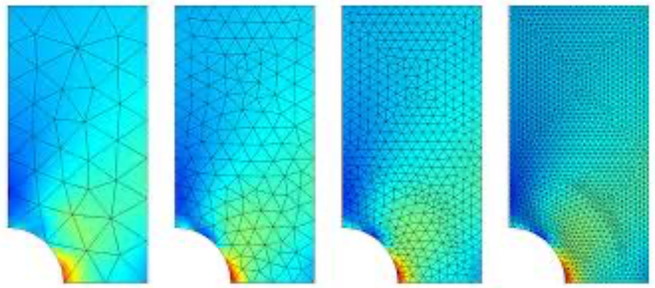
1st International Workshop on Distributed and Parallel Programming for Extreme-scale AI
Mines Paris - PSL University
June 16, 2025

How to Model Complex (Non-linear) Systems?

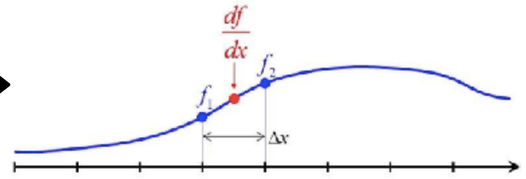
Mesh

Piece-wise linear functions

$$\frac{\partial f}{\partial x} = \frac{f_2 - f_1}{\Delta x}$$



https://www.comsol.jp/multiphysics/mesh-refinement



$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nu \nabla^2 \mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{f}$$

Millennium Prize Problems

Birch and Swinnerton-Dyer conjecture
Hodge conjecture
Navier–Stokes existence and smoothness
P versus NP problem
Poincaré conjecture (solved)
Riemann hypothesis
Yang–Mills existence and mass gap

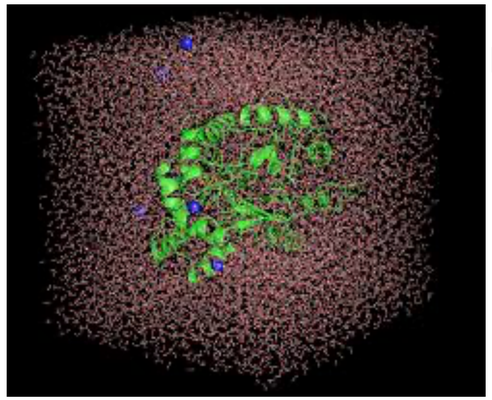
$$Ax = b$$

Basic linear algebra
Clear cut API

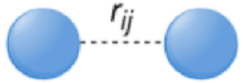
Particle

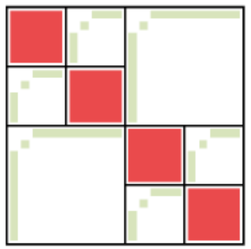
Pair-wise linear functions

$$F = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r_{ij}^2}$$



https://qstatix.co.in/molecular-dynamics-simulations/

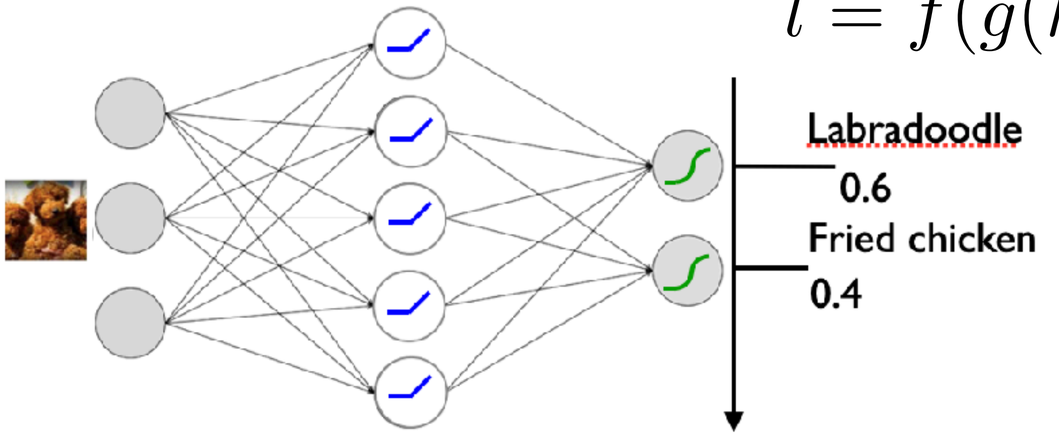




Neural Network

Composite (mostly) linear functions

$$l = f(g(h(x)))$$



Forward propagation

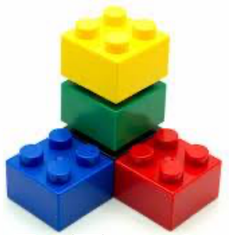
$$x_0 \quad x_1 = ReLU(z_0) \quad p = softmax(z_1)$$

$$z_0 = W_0 x_0 \quad z_1 = W_1 x_1 \quad l = -\log p$$

Backward propagation

$$\frac{\partial x_1}{\partial z_0} * \frac{\partial z_1}{\partial x_1} * \frac{\partial l}{\partial z_1} = \frac{\partial l}{\partial \theta}$$

$$\frac{\partial z_0}{\partial W_0} \otimes \frac{\partial z_1}{\partial W_1} \otimes$$

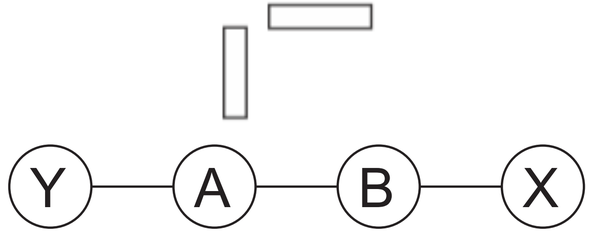


PyTorch

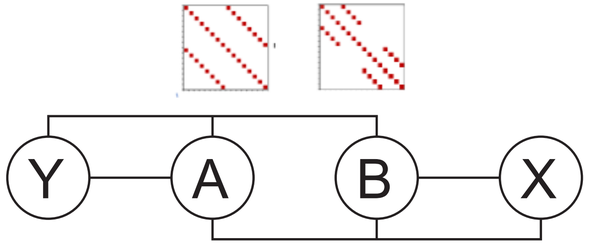
Various Matrix Factorizations

Linear Layer $Y = WX$ → Factorize W → $Y = ABX$

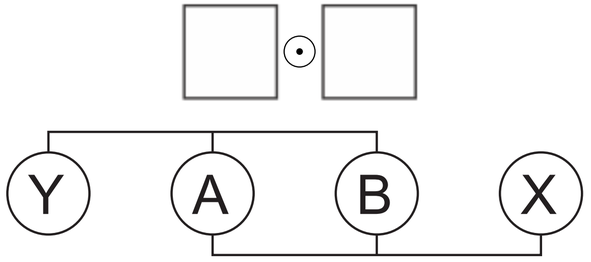
Low-Rank



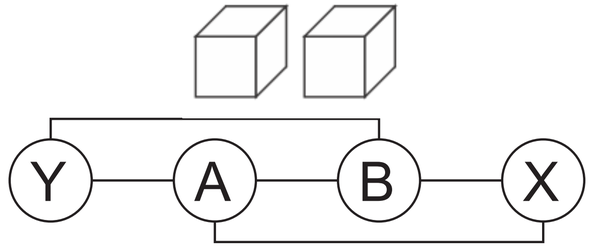
Monarch



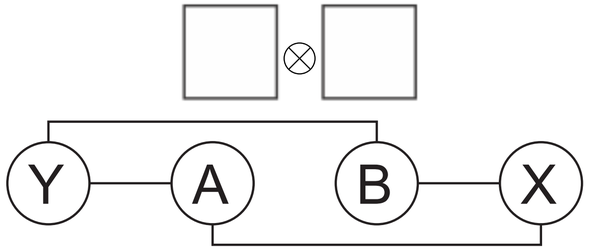
Hadamard



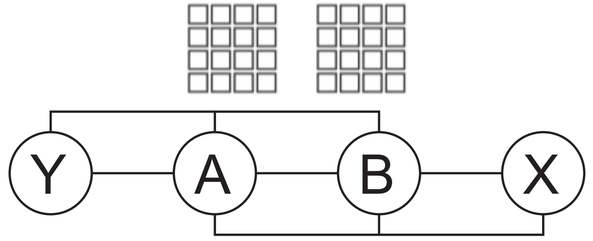
Tensor Train



Kronecker

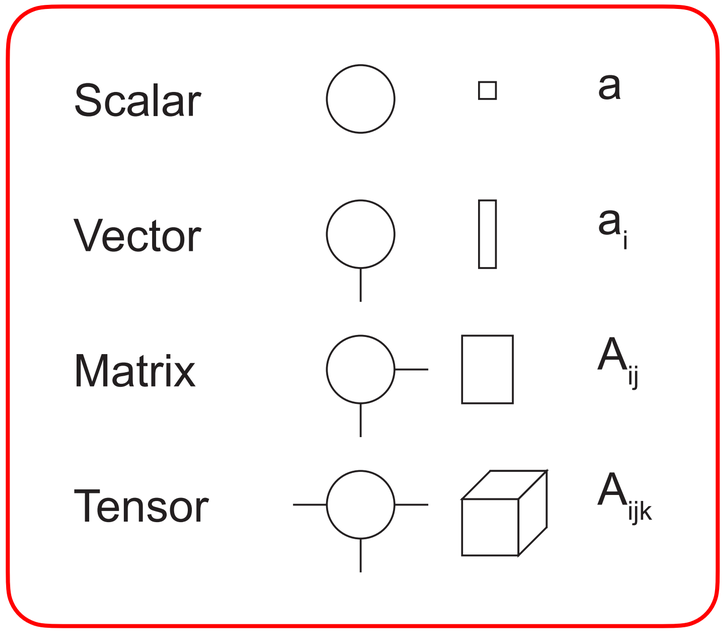


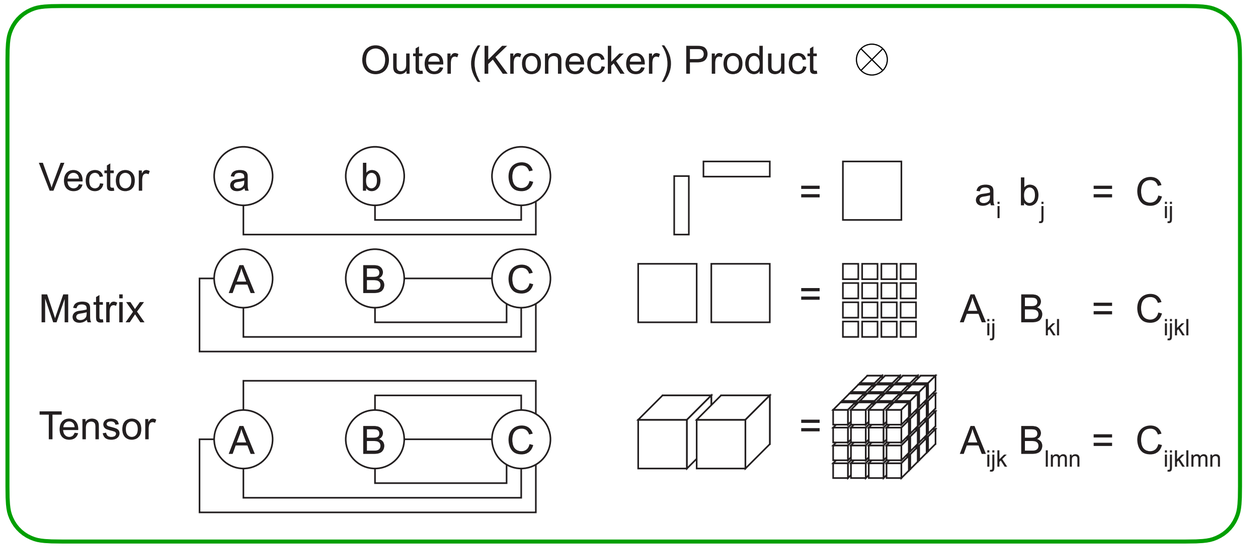
Block Tensor Train

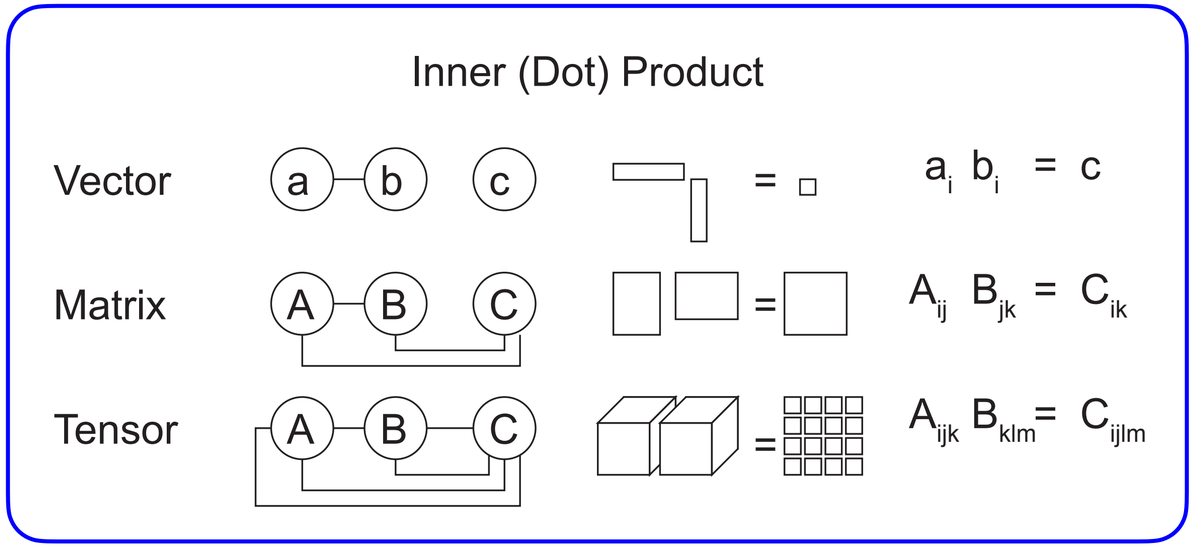


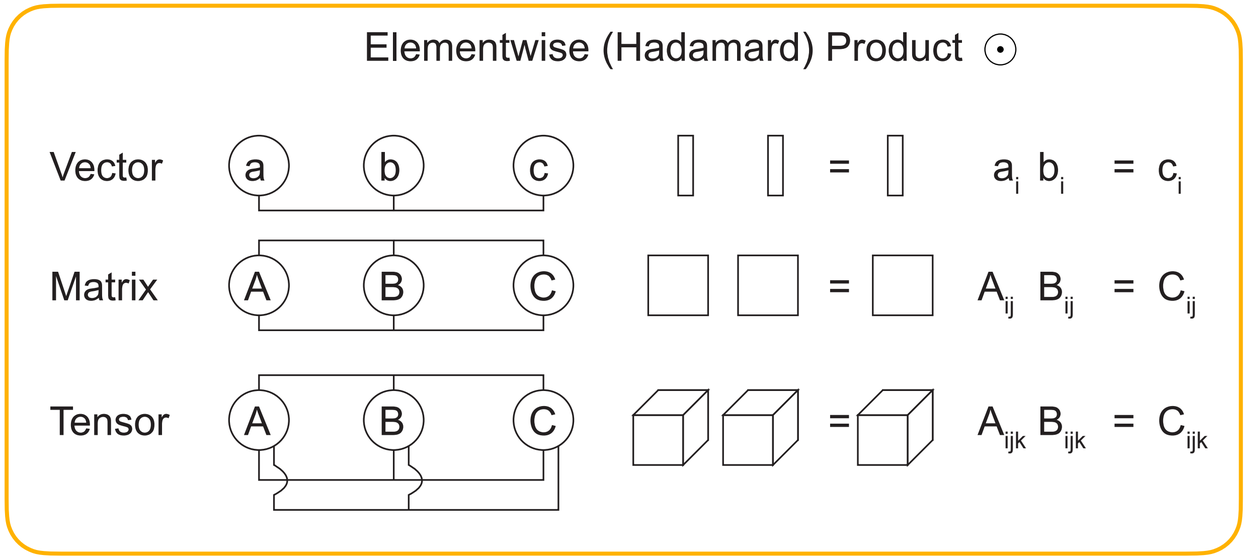
Potapczynski et al., Searching for Efficient Linear Layers over a Continuous Space of Structured Matrices, https://arxiv.org/abs/2410.02117

Various Products





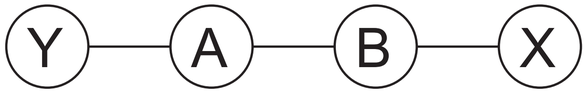




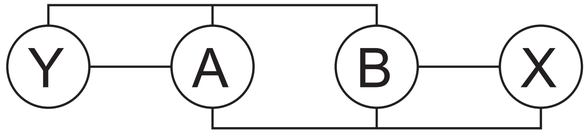
Various Matrix Factorizations

Linear Layer $Y = WX$ ⟶ (Factorize) $Y = ABX$

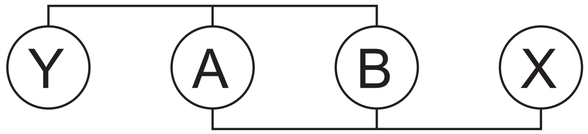
Low-Rank



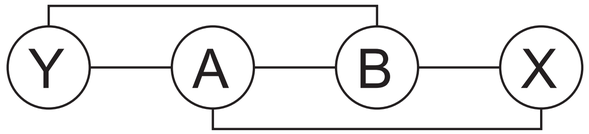
Monarch



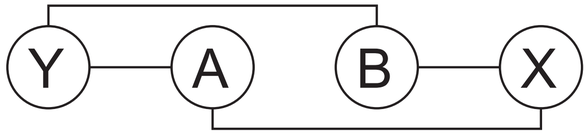
Hadamard



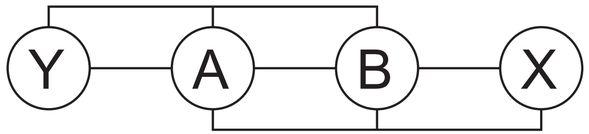
Tensor Train



Kronecker



Block Tensor Train



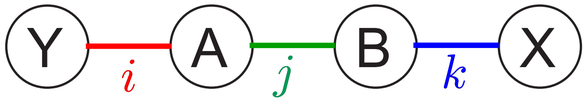
Potapczynski et al., Searching for Efficient Linear Layers over a Continuous Space of Structured Matrices, https://arxiv.org/abs/2410.02117

Various Matrix Factorizations

Linear Layer $Y = WX$ → Factorize → $Y = ABX$

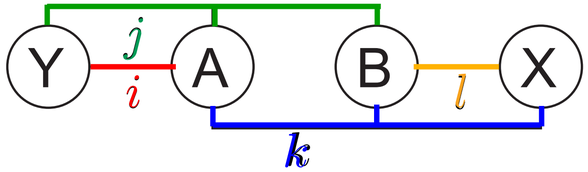
Low-Rank

$$Y_i = A_{ij} B_{jk} X_k$$



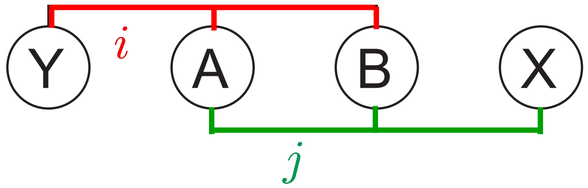
Monarch

$$Y_{ij} = A_{ijk} B_{jkl} X_{kl}$$



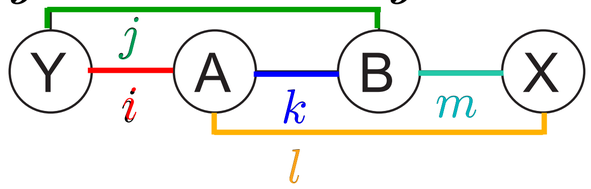
Hadamard

$$Y_i = A_{ij} B_{ij} X_j$$



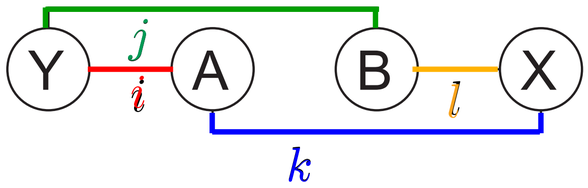
Tensor Train

$$Y_{ij} = A_{ikl} B_{jkm} X_{lm}$$



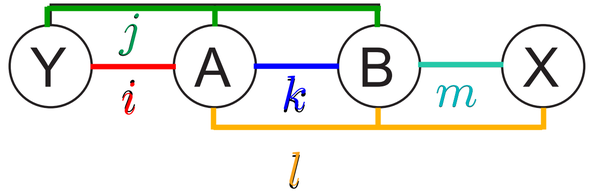
Kronecker

$$Y_{ij} = A_{ik} B_{jl} X_{kl}$$



Block Tensor Train

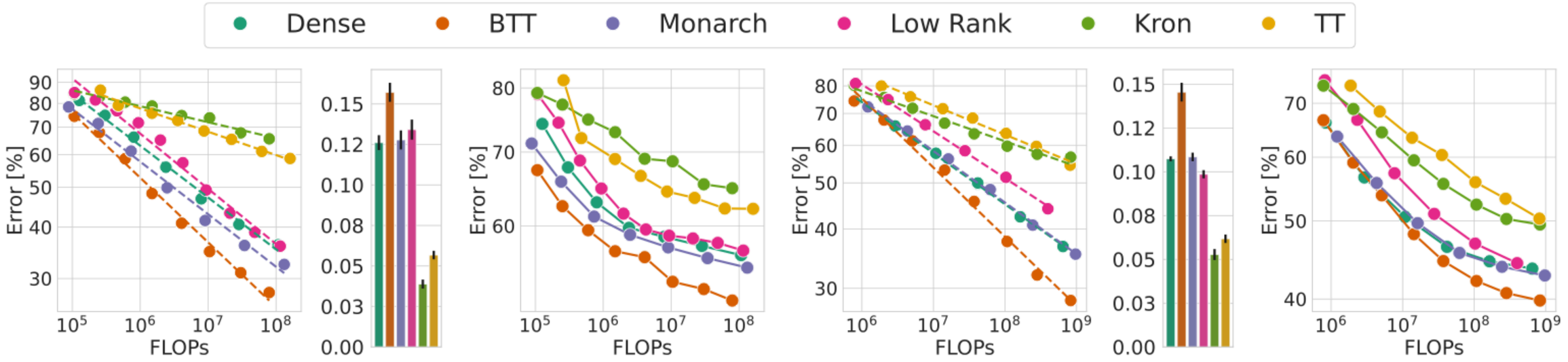
$$Y_{ij} = A_{ijkl} B_{jklm} X_{lm}$$



Potapczynski et al., Searching for Efficient Linear Layers over a Continuous Space of Structured Matrices, https://arxiv.org/abs/2410.02117

Scaling Laws on CIFAR-100

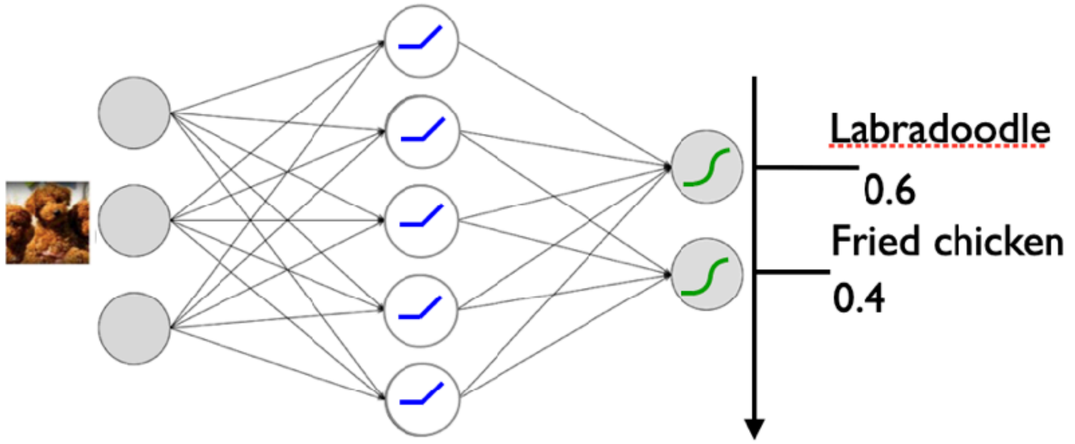
- Block Tensor Train has the best scaling
  - BTT > Monarch > Dense > Low-Rank > TT = Kron
- Works not only for MLPs, but also ViTs
  - Ranking of methods is also similar
- Test / train gap
  - Similar to existing studies



(a) MLP train error and exponent (b) MLP test error (c) ViT train error and exponent (d) ViT test error

Potapczynski et al., Searching for Efficient Linear Layers over a Continuous Space of Structured Matrices, https://arxiv.org/abs/2410.02117

Backprop and Kronecker products

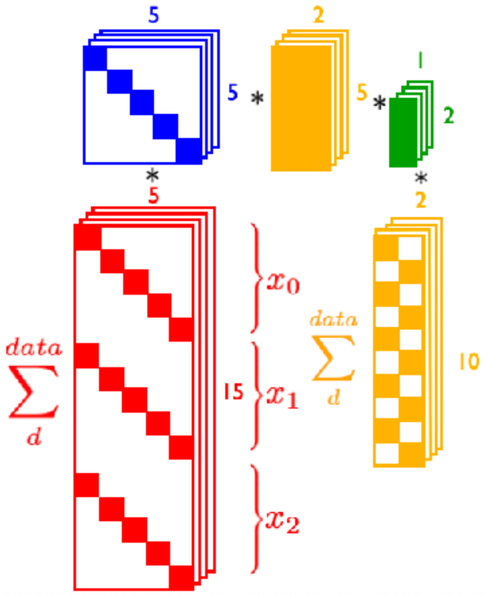


Forward propagation

$$x_0 \quad x_1 = ReLU(z_0) \quad p = softmax(z_1)$$
$$z_0 = W_0 x_0 \qquad z_1 = W_1 x_1$$

Backward propagation

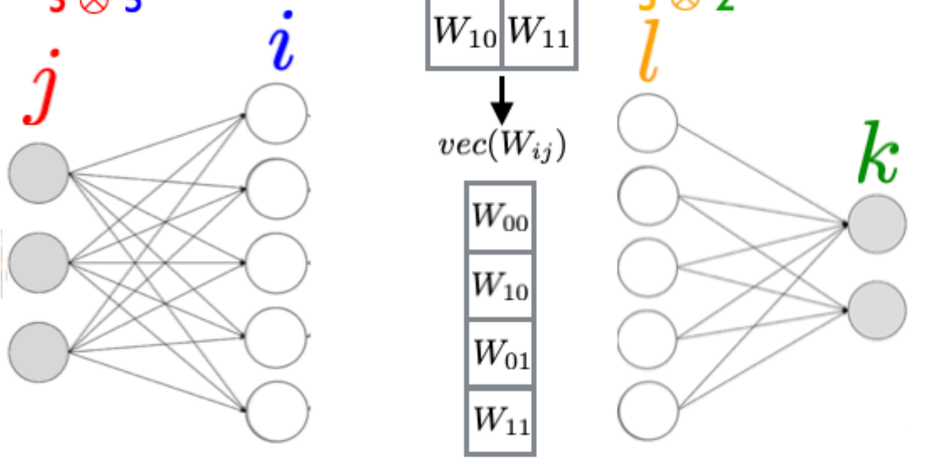
$$\frac{\partial x_1}{\partial z_0} * \frac{\partial z_1}{\partial x_1} * \frac{\partial l}{\partial z_1} = \frac{\partial l}{\partial \theta}$$
$$\frac{\partial z_0}{\partial W_0} \qquad \frac{\partial z_1}{\partial W_1}$$



$$z_i = W_{ij} x_j \qquad z_k = W_{kl} x_l$$

$$x_j \frac{\partial l}{\partial z_i} = \frac{\partial l}{\partial W_{ij}} \qquad x_l \frac{\partial l}{\partial z_k} = \frac{\partial l}{\partial W_{kl}}$$

$3 \otimes 5$ $\qquad$ $5 \otimes 2$



$$z_0 = W_{00}x_0 + W_{01}x_1 + W_{02}x_2$$
$$z_1 = W_{10}x_0 + W_{11}x_1 + W_{12}x_2$$
$$z_2 = W_{20}x_0 + W_{21}x_1 + W_{22}x_2$$
$$z_3 = W_{30}x_0 + W_{31}x_1 + W_{32}x_2$$
$$z_4 = W_{40}x_0 + W_{41}x_1 + W_{42}x_2$$

Approximation of the Hessian/Fisher Matrix

$$l = -\log p$$
$$\frac{\partial l}{\partial p} = -\frac{1}{p}$$
$$\frac{\partial^2 l}{\partial p^2} = \frac{1}{p^2} = \left(\frac{\partial l}{\partial p}\right)^2$$

$$F = \sum^{samples} \left(\frac{\partial l}{\partial W}\right)\left(\frac{\partial l}{\partial W}\right)^T \approx \left(\sum^{samples} \frac{\partial l}{\partial W}\right)\left(\sum^{samples} \frac{\partial l}{\partial W}\right)^T$$

IM

Newton-Schultz $= USV$

Muon (2025)

IK

Shampoo (2020)
Soap (2023)

$$F = \sum^{samples} \left(\frac{\partial l}{\partial W}\right)\left(\frac{\partial l}{\partial W}\right)^T = \sum^{samples} \left(x \otimes \frac{\partial l}{\partial z}\right)\left(x \otimes \frac{\partial l}{\partial z}\right)^T = \sum^{samples} (xx^T) \otimes \left(\frac{\partial l}{\partial z}\frac{\partial l}{\partial z}^T\right) \approx \left(\sum^{samples} xx^T\right) \otimes \left(\sum^{samples} \frac{\partial l}{\partial z}\frac{\partial l}{\partial z}^T\right)$$

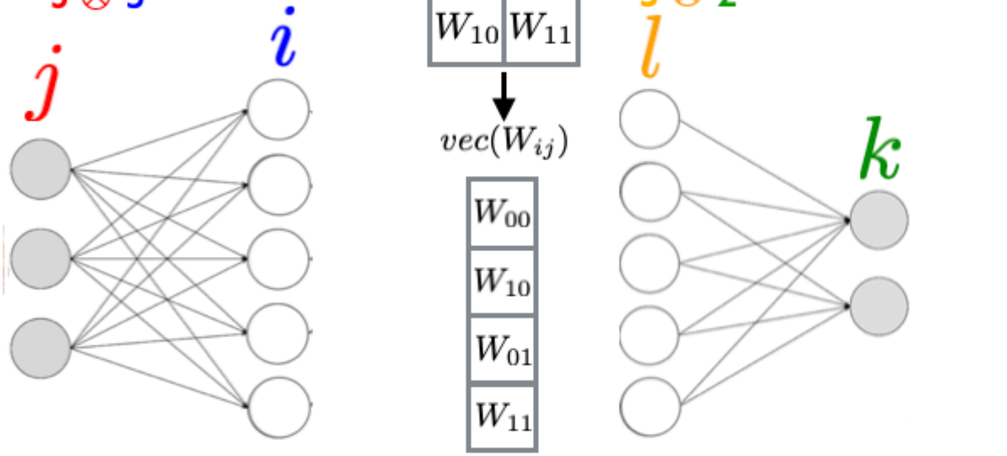
IK

K-FAC (2015)

$$z_i = W_{ij}x_j \qquad z_k = W_{kl}x_l$$

$$x_j \frac{\partial l}{\partial z_i} = \frac{\partial l}{\partial W_{ij}} \qquad x_l \frac{\partial l}{\partial z_k} = \frac{\partial l}{\partial W_{kl}}$$

$3 \otimes 5$ $\qquad$ $5 \otimes 2$



Keller Jordan ✓
@kellerjordan0

There have been hundreds of optimizer papers published. But the SOTA has only improved a few times. Therefore we can conclude that almost all optimizer papers are fake.

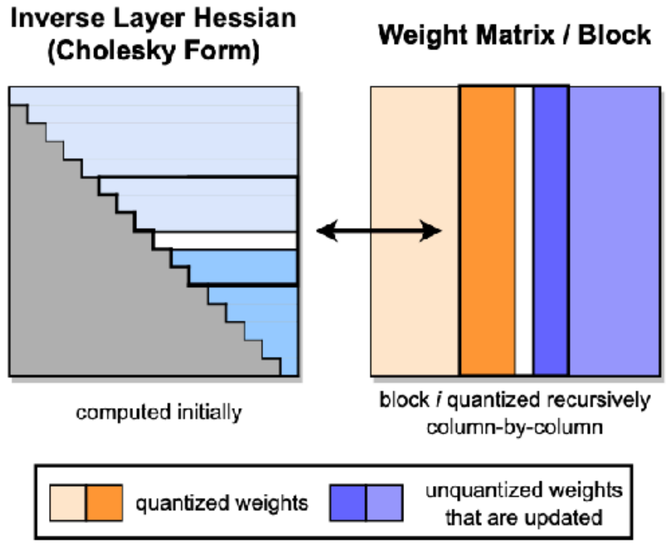
If you're gonna write another fake optimizer paper, please don't cite Muon. I don't want your citation

8:36 AM · Jun 15, 2025 · **43.8K** Views

Using the Matrix for things other than optimization

Quantization

$$\mathbf{H}^{-1} = (2\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}$$



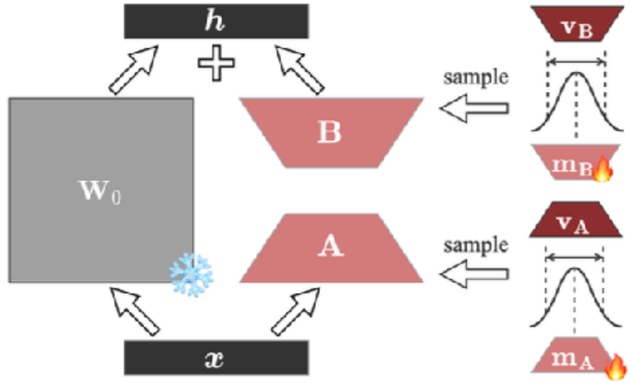
Published as a conference paper at ICLR 2023

GPTQ: ACCURATE POST-TRAINING QUANTIZATION FOR GENERATIVE PRE-TRAINED TRANSFORMERS

Elias Frantar* IST Austria
Saleh Ashkboos ETH Zurich
Torsten Hoefler ETH Zurich
Dan Alistarh IST Austria & NeuralMagic

Pruning

$$\widehat{\mathbf{h}} \leftarrow \widehat{\nabla}\bar{\ell}(\boldsymbol{\theta}) \cdot \frac{\boldsymbol{\theta} - \mathbf{m}}{\boldsymbol{\sigma}^2}$$



Improving LoRA with Variational Learning

Bai Cong[1,2] **Nico Daheim**[3] **Yuesong Shen**[4]
Rio Yokota[1] **Mohammad Emtiyaz Khan**[2] **Thomas Möllenhoff**[2]

[1]Institute of Science Tokyo [2]RIKEN Center for AI Project
[3]Technical University of Darmstadt [4]Huawei Hong Kong Research Center

Summary

- Various matrix structures are used in AI
- Matrix structures in HPC do not translate directly to AI
- HPC Matrices
  - Exploit structure in the underlying 3-D geometry
  - Spatial decomposition
- AI Matrices
  - Exploit structure between the billions of dimensions
  - Backprop naturally contains a Kronecker structure
- Changing the model architecture is often the simplest way to improve AI