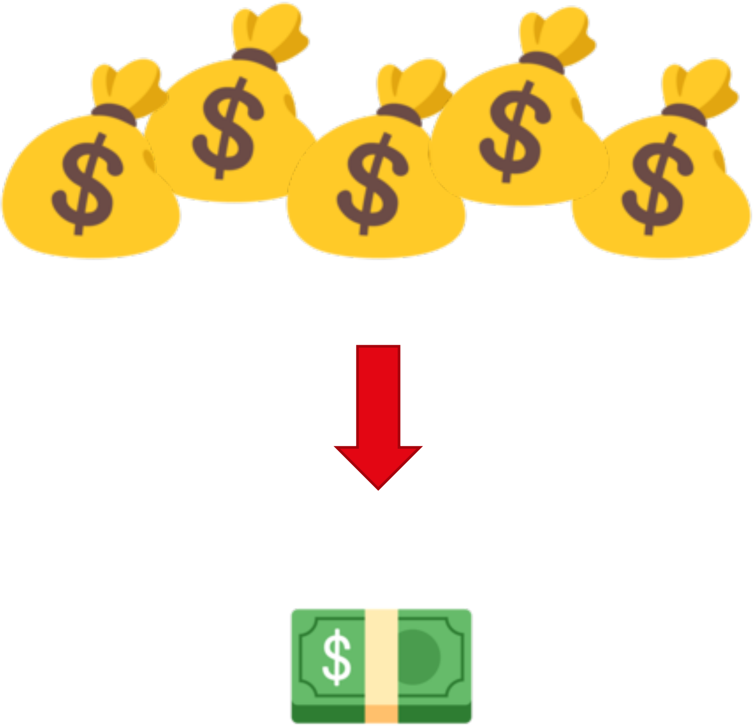
EPFL

PIXELATED BUTTERFLY:

SIMPLE AND EFFICIENT SPARSE TRAINING FOR NEURAL NETWORK MODELS

Bettina Messmer

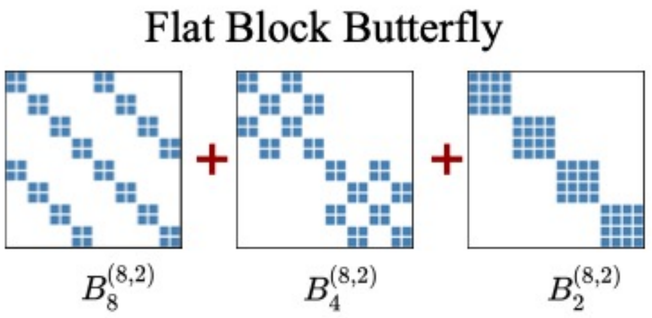
EPFL - 25.04.2023



Elevator Pitch

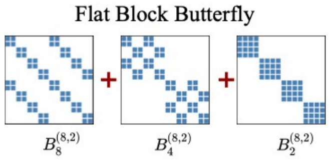
Problem

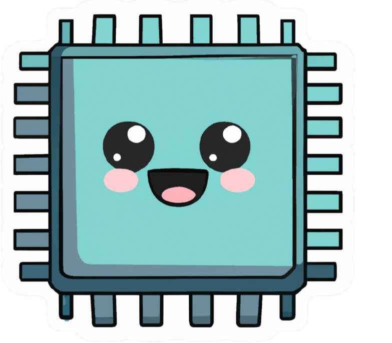
We need models that are cheaper to train, while keeping generalization benefits of large models.



Elevator Pitch
Solution

We need to train our models using pixelated butterflies.





DALL-E (generated)

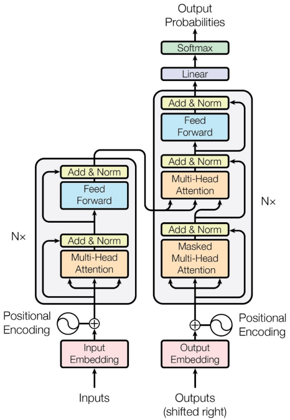
Elevator Pitch

Value

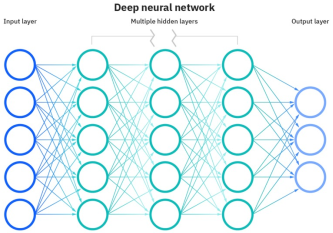
Sparsity pattern is hardware efficient

Wide-range of NN architecture support

Up to 2.5x speed up on ImageNet without accuracy loss



Attention is all you need,
A. Vaswani et al., 2017



https://www.xprimarycare.com/p/artificial-intelligence-in-primary (24.04.2023)

Elevator Pitch - Value

Sparsity pattern is hardware efficient

Wide-range of NN architecture support

Up to 2.5x speed up on ImageNet without accuracy loss

Model	Mixer-B/16	Pixelfly-Mixer-B/16
Accuracy (ImageNet)	75.6	76.3
Speedup		2.3x

Elevator Pitch - Value

Sparsity pattern is hardware efficient

Wide-range of NN architecture support

Up to 2.3x speed up on ImageNet without accuracy loss

What is the problem?

- Develop sparse training method that is/has
  - simple and accurate (static sparsity pattern)
  - sparsity pattern aligned with available hardware
  - wide-coverage of operators (applicable to most NN-layers)

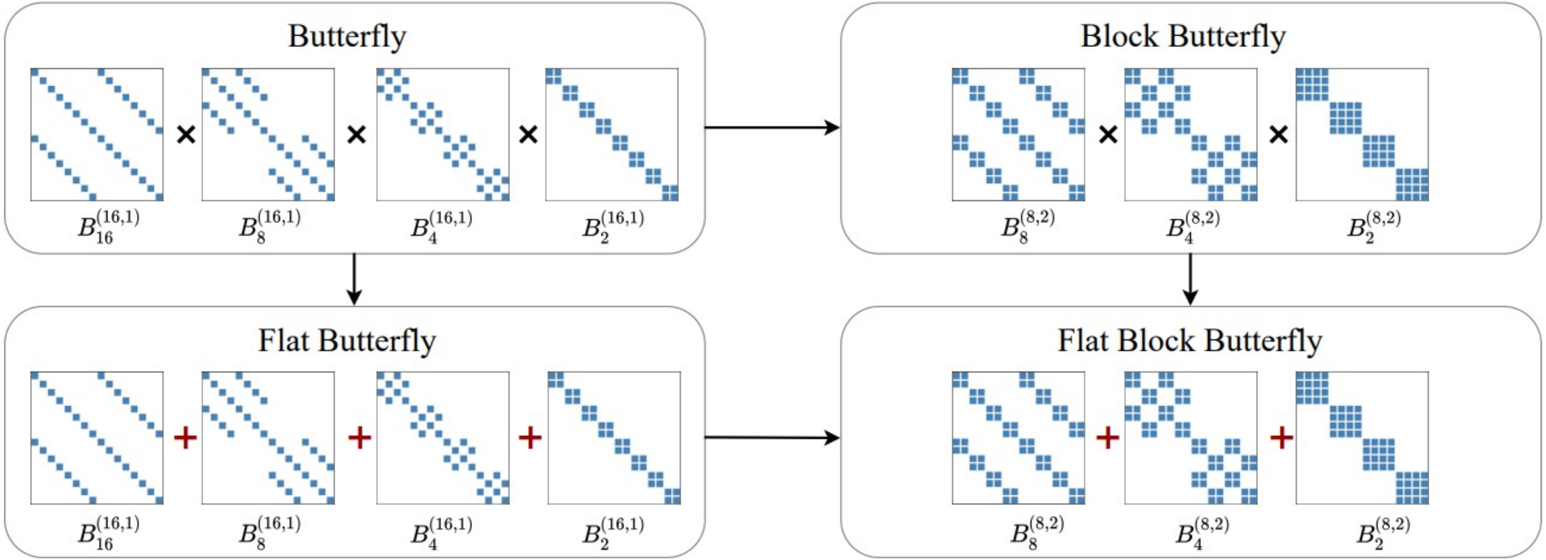
Why is it important?

- Overparameterized NNs generalize well, but are expensive to train
  - **Goal:** reduce computational cost, while retaining generalization benefit
  - **State-of-the art** sparsity training
    - has accuracy loss
    - slow training runtime (dynamic sparsity patterns)
    - sparsity patterns are not hardware efficient
    - specific to a network layer

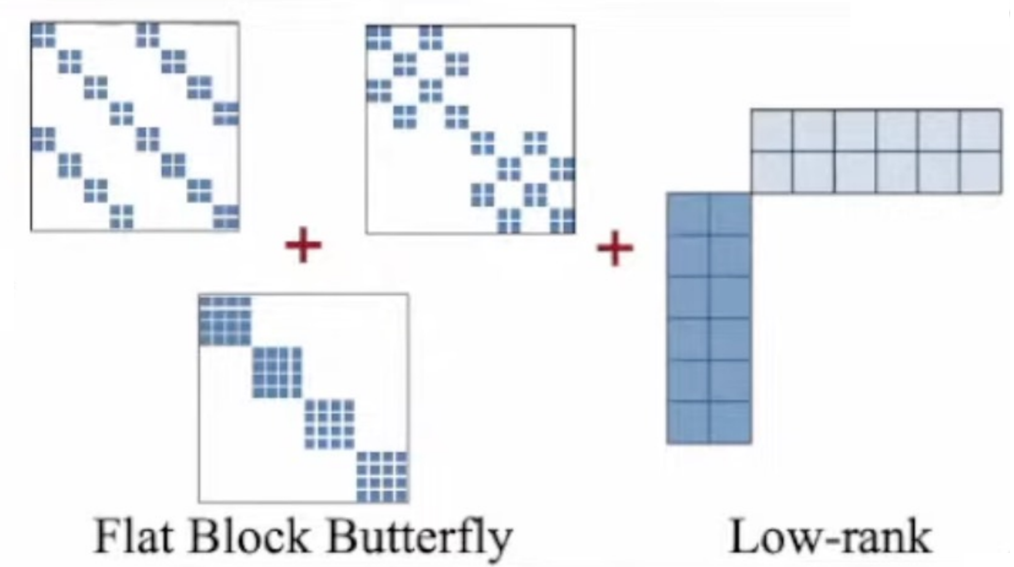
What are the insights?

- Butterfly matrix + low-rank matrix is an effective fixed sparsity pattern
  - Sparse matrix + low-rank matrix obtains better approximation than only one of the two
- Approximate Butterfly matrices with flat block Butterfly matrices for hardware efficiency

What are the insights?



What are the insights?



What is the solution?

- Use Pixelated Butterfly training method
  - compute sparsity level of each layer type
  - select rank for low rank matrix
  - select the sparsity mask from the flat block butterfly sparsity pattern
  - approximate weights (W) with $\boldsymbol{\gamma}$ as learnable parameters

$$W = \gamma B + (1 - \gamma) U V^\top$$

Model	Mixer-B/16	Pixelfly-Mixer-B/16
Accuracy (ImageNet)	75.6	76.3
Speedup		2.3x

What is the take-away message?

re-parameterization of sparsity patterns based on butterfly matrices enable fast training and good generalisation

Test of Time award

Pixelated butterfly: Simple and efficient sparse training for neural network models
T Dao, B Chen, K Liang, J Yang, Z Song… - arXiv preprint arXiv …, 2021 - arxiv.org
Overparameterized neural networks generalize well but are expensive to train. Ideally, one would like to reduce their computational cost while retaining their generalization benefits. Sparse model training is a simple and promising approach to achieve this, but there remain challenges as existing methods struggle with accuracy loss, slow training runtime, or difficulty in sparsifying all model components. The core problem is that searching for a sparsity mask over a discrete set of sparse matrices is difficult and expensive. To address …
Speichern Zitieren Zitiert von: 24 Ähnliche Artikel Alle 5 Versionen

https://scholar.google.com

Test of Time award

Pixelated butterfly: Simple and efficient sparse training for neural network models
T Dao, B Chen, K Liang, J Yang, Z Song… - arXiv preprint arXiv …, 2021 - arxiv.org
Overparameterized neural networks generalize well but are expensive to train. Ideally, one would like to reduce their computational cost while retaining their generalization benefits. Sparse model training is a simple and promising approach to achieve this, but there remain challenges as existing methods struggle with accuracy loss, slow training runtime, or difficulty in sparsifying all model components. The core problem is that searching for a sparsity mask over a discrete set of sparse matrices is difficult and expensive. To address …
Speichern Zitieren Zitiert von: 24 Ähnliche Artikel Alle 5 Versionen

https://scholar.google.com

Should it have been accepted? 👍

novel well-motivated insights

practical considerations

well-written

thorough experiments

EPFL

Thank you

Bettina Messmer

EPFL - 25.04.2023

CrAM – A compression Aware Minimizer

Bettina Messmer

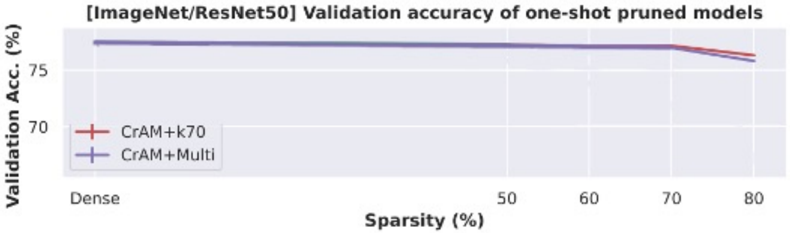
EPFL - 25.04.2023

Elevator Pitch - Problem

High model retraining cost when deploy on various devices

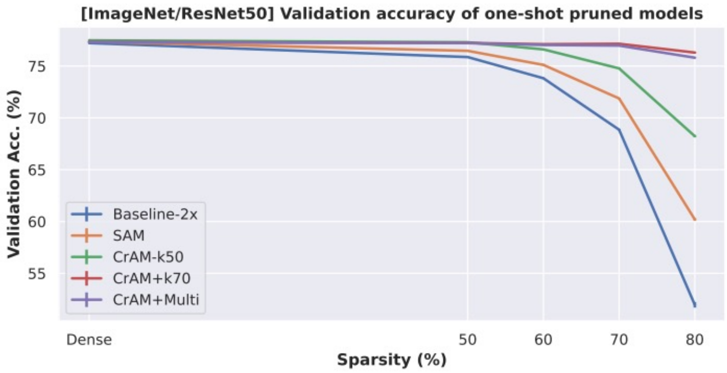
Requires compression at different rates

Elevator Pitch - Solution



CrAM optimizer for model training

EPFL



Elevator Pitch - Value

One-Shot compression support at different compression rates

No significant loss in accuracy and small training overhead

What is the problem?

- Reduce additional computation and hyper-parameter tuning for model compression

What are the insights?

- Define measure how well a model generalized with respect to compression
- A model is easily compressible if small perturbations do not affect its performance after compression

What is the solution?

- Define how much current model is compressible as part of the loss

$$L^{\mathrm{CrAM}}(\boldsymbol{\theta}) = \max_{\|\boldsymbol{\delta}\| \leq \rho} L(C(\boldsymbol{\theta} + \boldsymbol{\delta})), \quad L^{\mathrm{CrAM}^{+}}(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + L^{\mathrm{CrAM}}(\boldsymbol{\theta}) .$$

- Use two forward/backward passes per training step
  - Compute gradient **ascent** step to find locally worst compressed model
  - Compute gradient **descent** step to optimize model performance
- Choose compression method uniformly at random when using different compression algorithms (rates)
- Use 1000 random training samples to correct for model statistics
  - E.g. mean and standard deviation for batch norms

$$L^{\mathrm{CrAM}}(\boldsymbol{\theta}) = \max_{\|\boldsymbol{\delta}\| \leq \rho} L(C(\boldsymbol{\theta} + \boldsymbol{\delta})),$$

What is the take-away message?

the concept of sharpness-aware minimisation can be extended to compression-aware minimisation

Test of Time award

CrAM: A Compression-Aware Minimizer

A Peste, A Vladu, D Alistarh, CH Lampert - arXiv preprint arXiv …, 2022 - arxiv.org

We examine the question of whether SGD-based optimization of deep neural networks (DNNs) can be adapted to produce models which are both highly-accurate and easily-compressible. We propose a new compression-aware minimizer dubbed CrAM, which modifies the SGD training iteration in a principled way, in order to produce models whose local loss behavior is stable under compression operations such as weight pruning or quantization. Experimental results on standard image classification tasks show that CrAM …

Speichern Zitieren Zitiert von: 1 Ähnliche Artikel Alle 3 Versionen

https://scholar.google.com

Test of Time award


CrAM: A Compression-Aware Minimizer

A Peste, A Vladu, D Alistarh, CH Lampert - arXiv preprint arXiv ..., 2022 - arxiv.org

We examine the question of whether SGD-based optimization of deep neural networks (DNNs) can be adapted to produce models which are both highly-accurate and easily-compressible. We propose a new compression-aware minimizer dubbed CrAM, which modifies the SGD training iteration in a principled way, in order to produce models whose local loss behavior is stable under compression operations such as weight pruning or quantization. Experimental results on standard image classification tasks show that CrAM …

☆ Speichern 99 Zitieren Zitiert von: 1 Ähnliche Artikel Alle 3 Versionen »

https://scholar.google.com

Should it have been accepted?

well-motivated method design

well-written

thorough experiments

Should it have been accepted? 👎

well-motivated method design

well-written

thorough experiments

not inspiring enough for community

Thank you

Bettina Messmer

EPFL - 25.04.2023