GETTING AI IN YOUR POCKET WITH DEEP COMPRESSION

Dr. Axel Plinge, Ashutosh Mishra
Fraunhofer IIS – International Audio Labs Erlangen
Embedded World Conference Nürnberg; 26. Feb. 2020

GETTING AI IN YOUR POCKET
TABLE OF CONTENTS

Fraunhofer IIS

Getting AI in Your Pocket
Motivation (1)

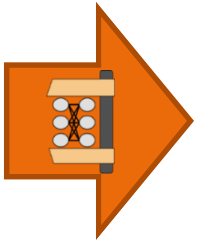
- DNNs are trained on Graphical Processing Units (GPUs)
- Should run on embedded devices in real-time

GPU Image by ChrisDag used under Creative Commons Attribution 2.0 Generic license.
Embedded HW image taken from https://www.itu.int/en/ITU-T/Workshops-and-Seminars/20191008/Documents/Wojciech_Samek_Presentation.pdf

Fraunhofer IIS

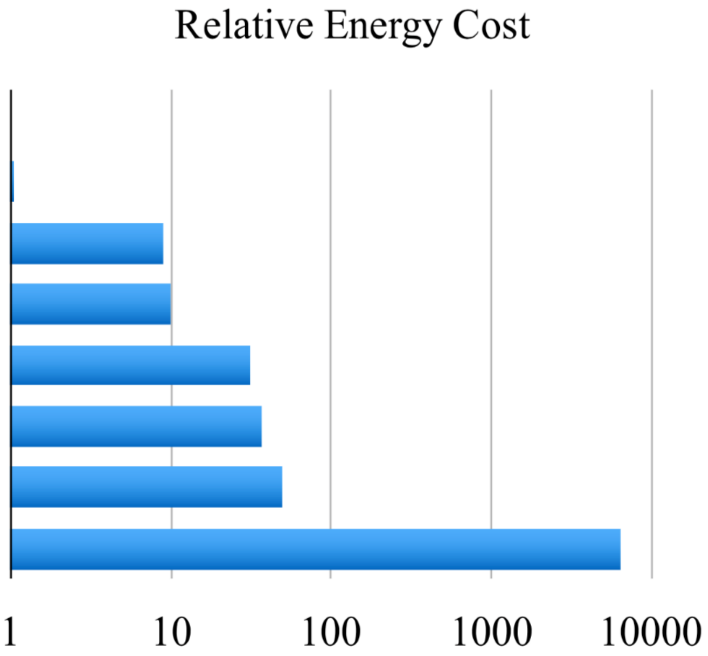
Getting AI in Your Pocket
Motivation (1)

- DNNs are trained on Graphical Processing Units (GPUs)
- Should run on embedded devices in real-time
- Still need considerable resources at run-time (inference)
- Deep Compression can get the DNN Models there!

Getting AI in Your Pocket
Motivation (2) Energy

Operation	Energy [pJ]
32 bit int ADD	0.1
32 bit float ADD	0.9
32 bit Register File	1
32 bit int MULT	3.1
32 bit float MULT	3.7
32 bit SRAM Cache	5
32 bit DRAM Memory	640



Source:
http://isca2016.eecs.umich.edu/wp-content/uploads/2016/07/4A-1.pdf

Getting AI in Your Pocket
Motivation (3) Success Stories

- AlexNet (244MB) → SqueezeNet / MobileNet (5MB)
  - Image classification and detection CNN
  - Clever structural changes [Ian16,Google17]
  - Reduction to 2% original size with similar performance

[Ian16] Iandola, F. N., Moskewicz, M. W. et al. (2016) “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size” arXiv:1602.07360
[Google17] Howard, A. G. et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv:1704.04861

Getting AI in Your Pocket
Motivation (3) Success Stories

- Natural Language Model (570MB) → (22MB)
  - Reduction to 4% of original size
  - Combination of compression & hashing [Amazon18]
  - Amazon got Alexa from the Cloud on the Phone (!)

[Amazon18] Strimel, G. P. et al. (2018). Statistical Model Compression for Small-Footprint Natural Language Understanding. ArXiv:1807.07520.

GETTING AI IN YOUR POCKET
TABLE OF CONTENTS

State-of-the-Art
Tools and Platforms (1/3)

- Various tools provide basic model compression
  - NVIDIA TensorRT
  - Intel OpenVINO Inference Engine
  - Intel nGraph
  - CoreML
  - …

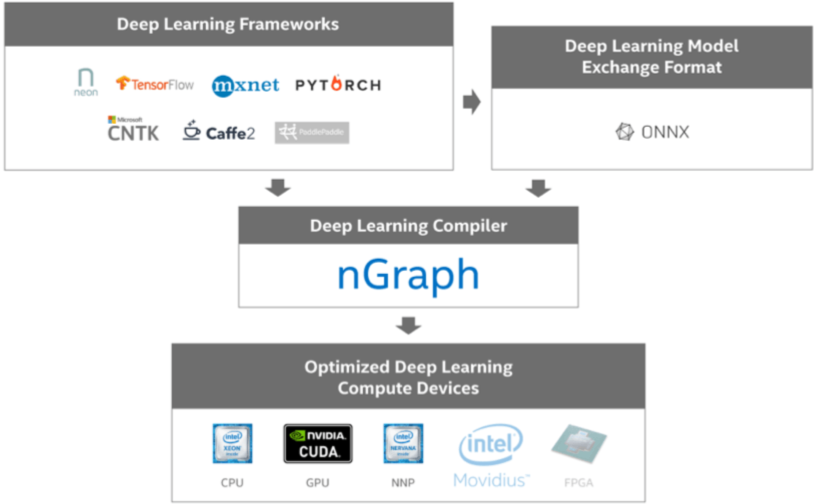


Image source:
Intel website https://www.intel.com/content/www/us/en/artificial-intelligence/ngraph.html

State-of-the-Art
Tools and Platforms (2/3)

- TVM Stack

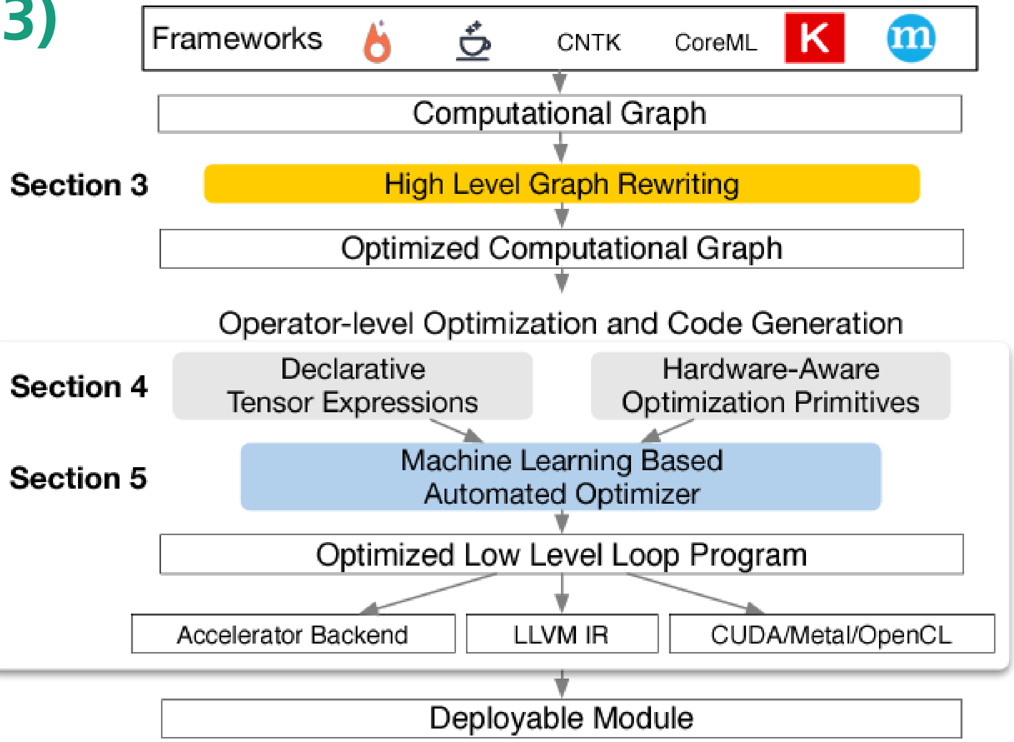


Image source:
[Chen17] Chen et al. (2017) „TVM: End-to-End Optimization Stack for Deep Learning“ArXiv abs/1802.04799

State-of-the-Art
Tools and Platforms (3/3)

- More embedded platforms
  - Newer smartphones have neuro-chips (!)
  - Tensorflow lite for embedded devices (8bit SIMD, ...)
  - Qualcomm Snapdragon SDK
  - Android NNAPI
  - STM32Cube.AI
  - …

State-of-the-Art
Research in “deep compression”

- ANNs got DNNs, deeper = larger, now really interesting
- It got momentum as “Deep Compression” [Han15]
- Dedicated methods give large gains
- These methods can be classified roughly as

Pruning → Low-rank approximation → Quantization → Entropy Coding

[Han15] S. Han et al. (2015) “Deep Compression: Compressing Deep Neural Networks with Pruning, trained Quantization and Huffman coding.” ArXiv:1510.00149

GETTING AI IN YOUR POCKET
TABLE OF CONTENTS

Deep Compression Methods
Pruning

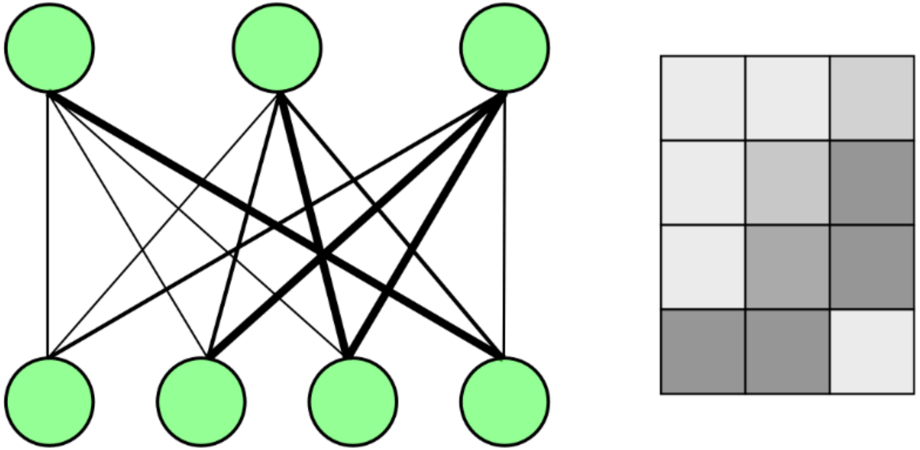
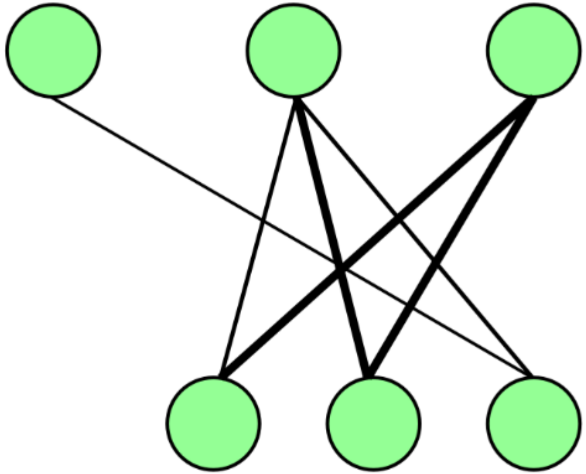


Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Pruning

- Remove weights = connections
- Remove neurons / filters



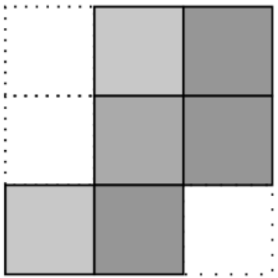


Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Pruning

- Optimal Brain Damage [LeCun1990]
  - Removes neurons based on training/validation error
  - "Recipe"
    1. Construct network with reasonable(!) architecture
    2. Train
    3. Compute Hessian (second derivatives of parameters)
    4. Compute saliency (effect on training error)
    5. Remove low-saliency parameters
    6. Goto 2

[LeCun1990] LeCun, Y., Denker, J. S., & Solla, S. A. "Optimal brain damage" In Advances in neural information processing systems (pp. 598–605)

Deep Compression Methods
Pruning

- Example: AlexNet [HanS15]

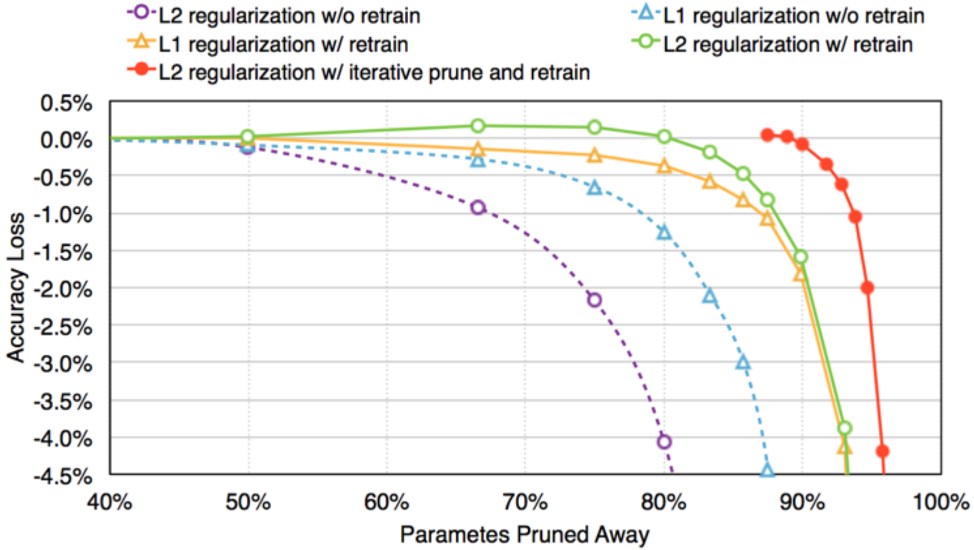
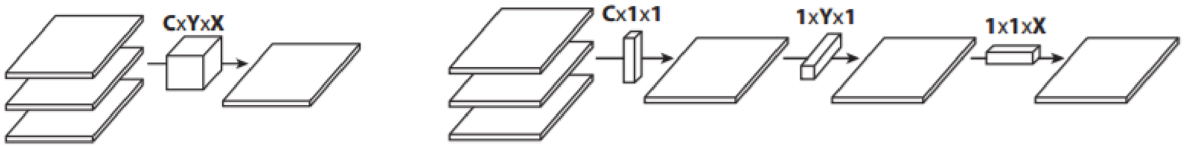


Image taken from [HanS15] Song Han (2015) Deep Compression and EIE, Stanford Lectures

Deep Compression Methods

Pruning convolutional neural networks (CNNs)

- Remove least used filters
  - Less parameter reduction
  - Direct speedup
- Flatten convolutions



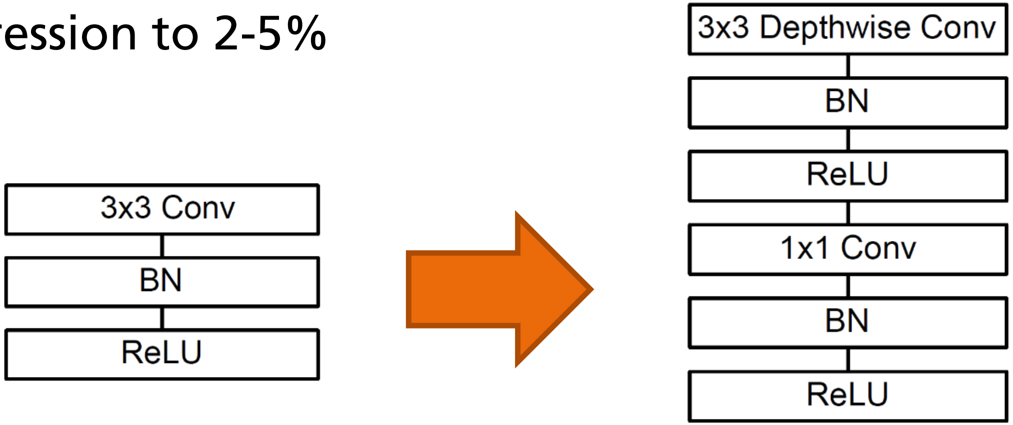
(a) 3D convolution (b) 1D convolutions over different directions

  - Large parameter reduction
  - Speedup ~ 2x

[Jib14] Jin, Jonghoon, et. Al (2014) "Flattened convolutional neural networks for feedforward acceleration." *arXiv preprint arXiv:1412.5474*

Deep Neural Network (DNN) Optimization
Pruning

- Depthwise convolution [Google2017]
  - Direct speedup ~ 8x
  - Compression to 2-5%



[Google2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., et al. MobileNets: Efficient CNNs for Mobile Vision Applications. ArXiv:1704.04861

GETTING AI IN YOUR POCKET

TABLE OF CONTENTS

1. Motivation
2. State-of-the-Art
3. **Methods**

   Pruning → Low-rank approximation → Quantization → Entropy Coding

4. Summary

Deep Compression Methods
Low Rank Approximation

- The weight tensors are large and redundant

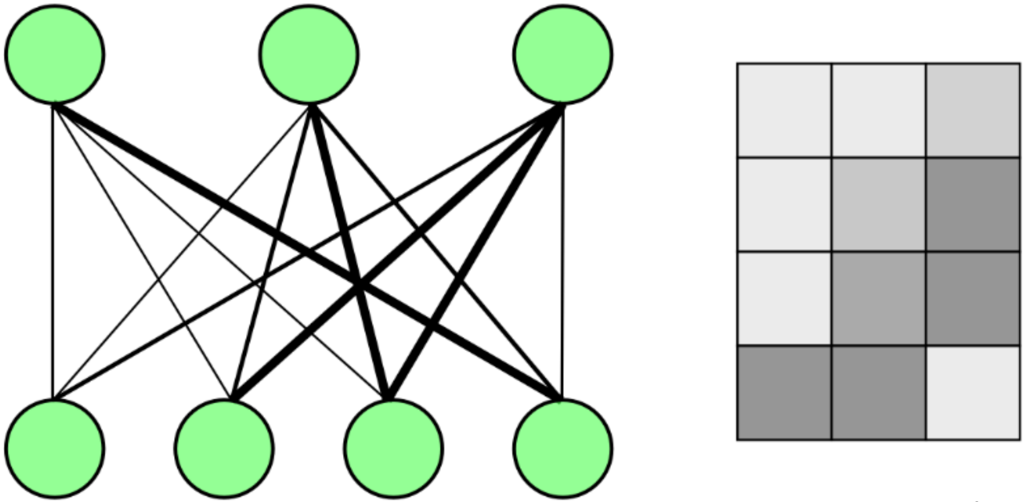


Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Low Rank Approximation

- The weight tensors are large and redundant
- They can be approximated with low-rank subspaces

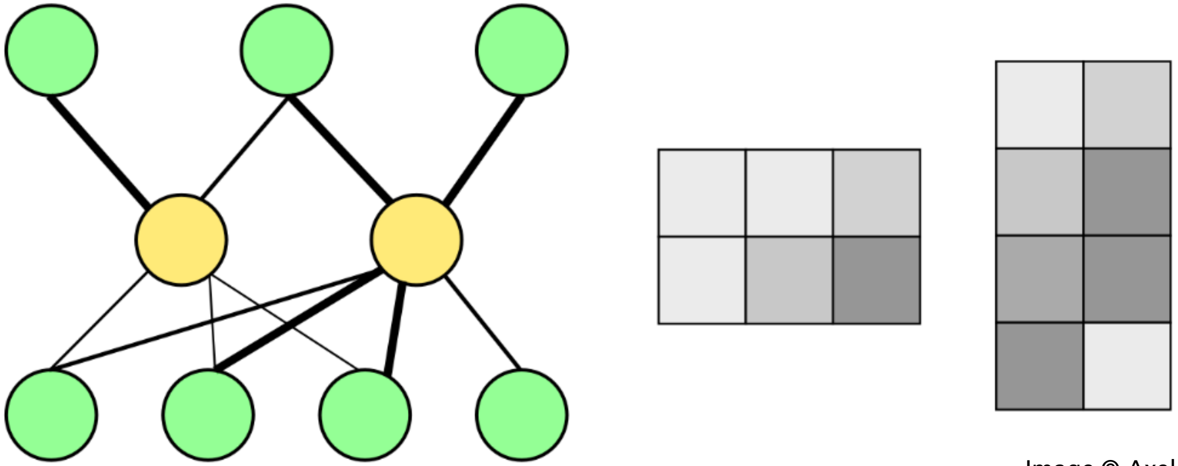
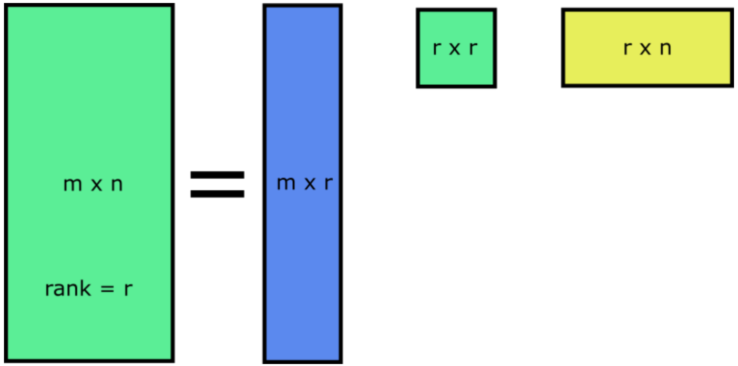


Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Low Rank Approximation

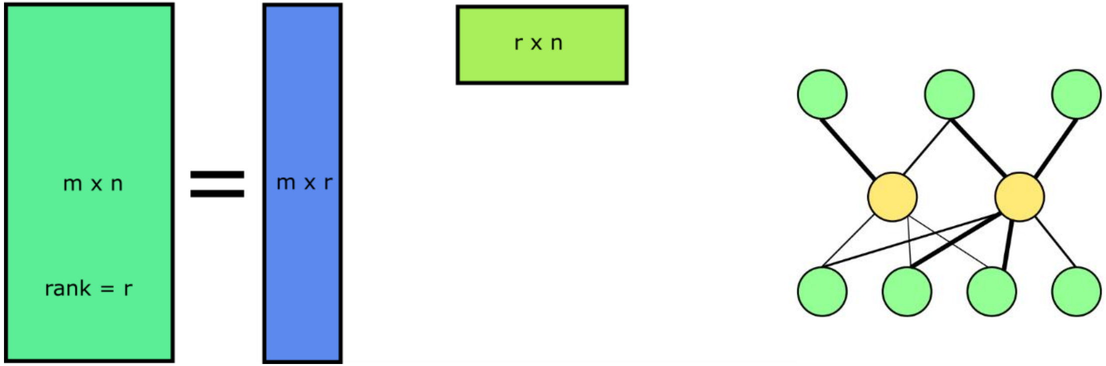
- Singular value decomposition allows to express a tensor of lower rank than size as product of smaller matrices



[Microsoft13] Xue, Jian et al. Restructuring of Deep Neural Network Acoustic Models with Singular Value Decomposition; Interspeech, 2013
Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Low Rank Approximation

- Singular value decomposition allows to express a tensor of lower rank than size as product of smaller matrices
- This allows to replace one tensor by two small ones

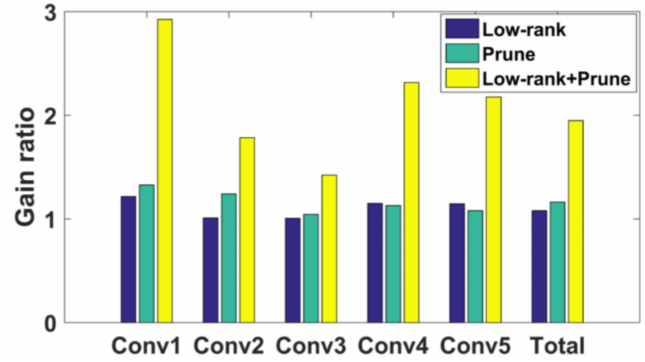


[Microsoft13] Xue, Jian et al. Restructuring of Deep Neural Network Acoustic Models with Singular Value Decomposition; Interspeech, 2013

Deep Compression Methods
Low Rank Approximation

- Requires some math and structural changes
- Does provide straightforward speed-up (3x)
- Can be easily combined with other methods

Latency reduction on AlexNet [Chen19]



[Chen19] Z. Chen et al., "Exploiting Weight-Level Sparsity in Channel Pruning with Low-Rank Approximation," 2019 IEEE Int. Symposium on Circuits and Systems, Sapporo, Japan, 2019

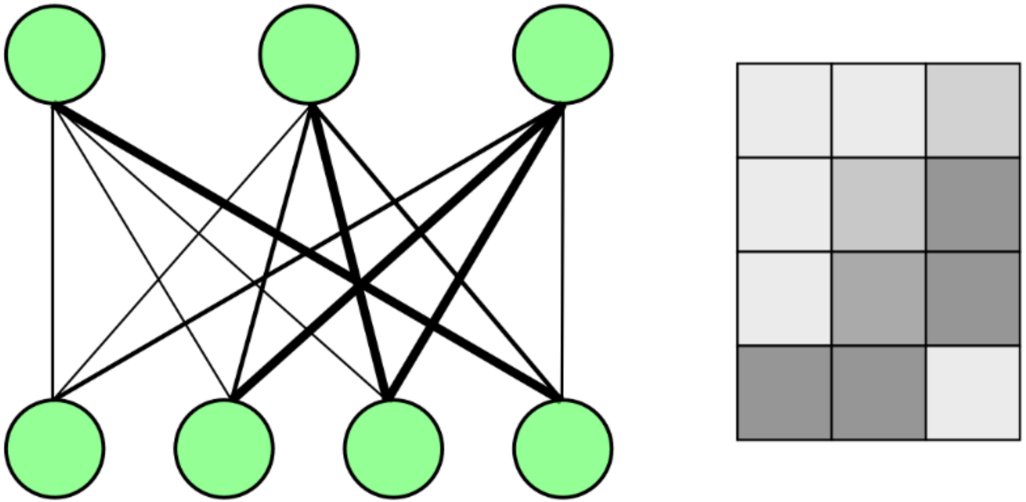
[Denton14] Denton, E., Zaremba, W., Bruna, J., LeCun, Y., et al. "Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation" arXiv 1404:0736

GETTING AI IN YOUR POCKET
TABLE OF CONTENTS

Deep Compression Methods

Quantization

- Weights are stored as 32 bit floating point



Fraunhofer IIS

Deep Compression Methods
Quantization

- Weights are stored as 32 bit floating point
- Good results can be achieved with much lower resolution

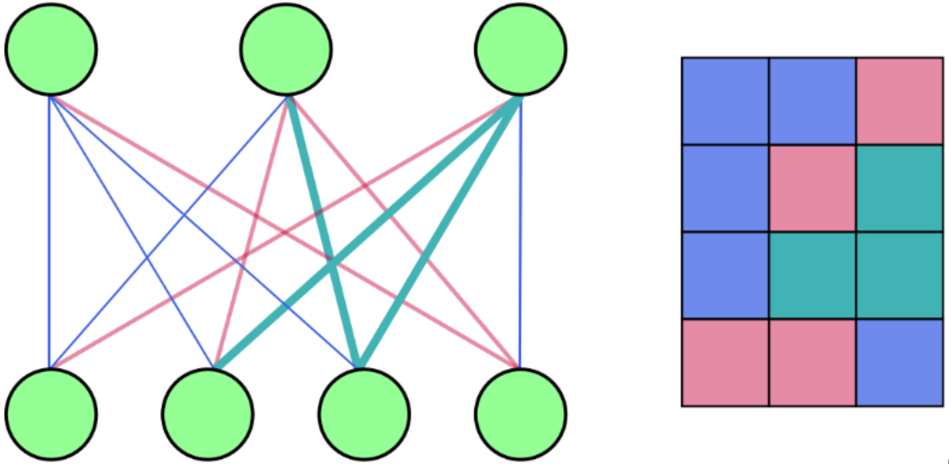


Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Quantization (1/4)

- Uniform Quantization

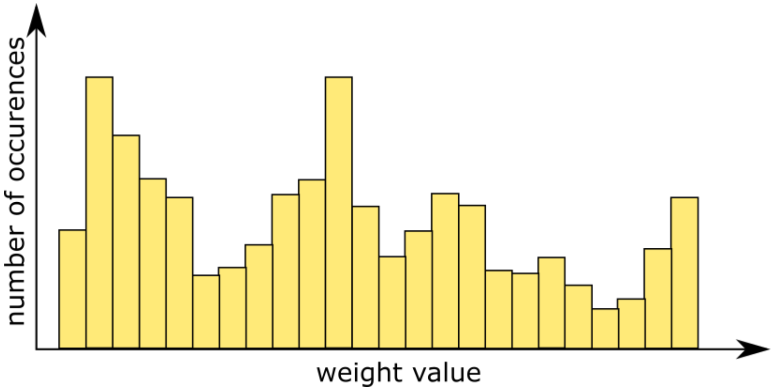
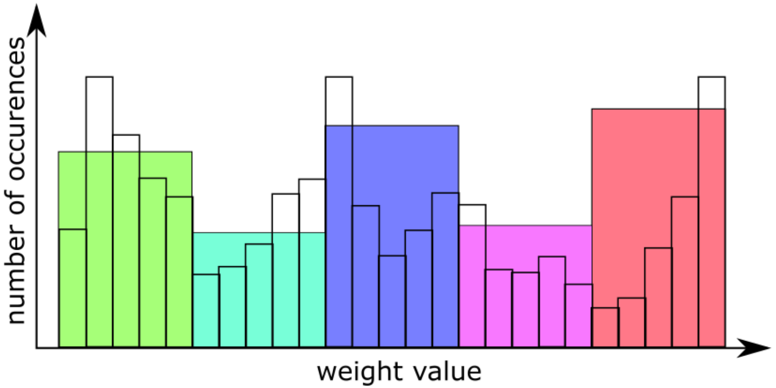


Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Quantization (1/4)

Image © Axel Plinge, Fraunhofer IIS.

- Uniform Quantization



  - Use less bits, i.e. 8 bit integer instead of 32 bit float
  - Retraining can improve performance, required for low bit count
  - Int8 Integrated in many frameworks (PyTorch, TensorFlow lite,…)

Deep Compression Methods
Quantization (2/4)

- XNOR Net [Rastegari16]

	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs: 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Real-Value Weights: 0.12 -1.2 ... 0.41 / -0.2 0.5 ... 0.68	+ , − , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs: 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Binary Weights: 1 -1 ... 1 / -1 1 ... 1	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input (**XNOR-Net**)	Binary Inputs: 1 -1 ... -1 / -1 1 ... 1; Binary Weights: 1 -1 ... 1 / -1 1 ... 1	XNOR , bitcount	~32x	~58x	%44.2

Image from [Rastegari16] Rastegari, M. et al. (2016) "XNOR-Net: ImageNet Classification using Binary Convolutional Neural Networks" ECCV. ArXiV: 1603.05279

Deep Compression Methods
Quantization (3/4)

- Adaptive Quantization

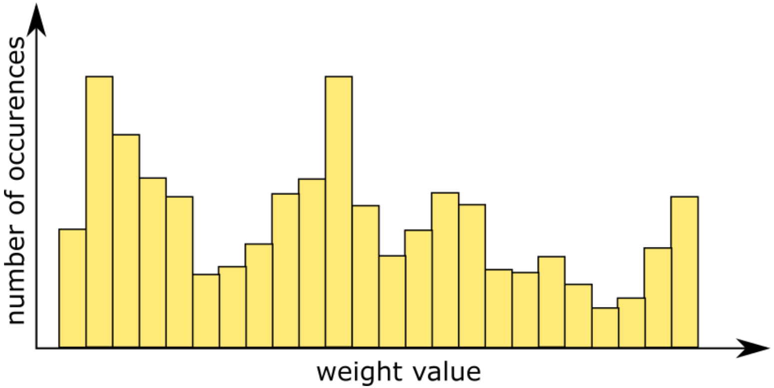
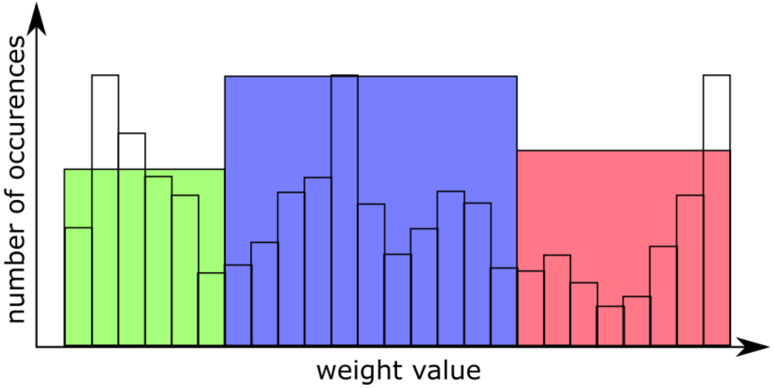


Image © Axel Plinge, Fraunhofer IIS.

Deep Compression Methods
Quantization (3/4)

- Adaptive Quantization

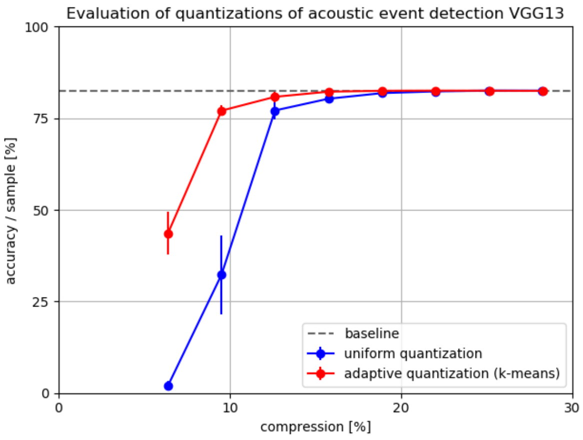


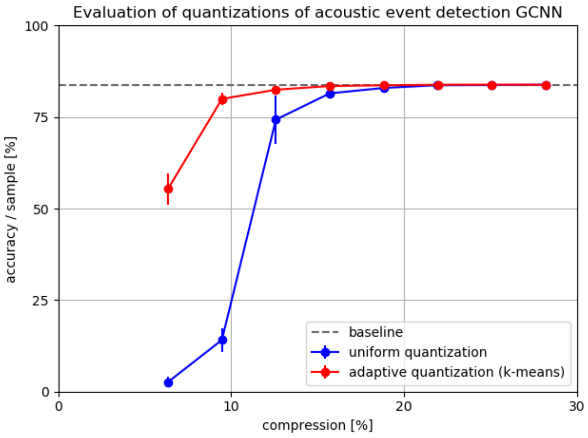
  - It is vector quantization on weights [Facebook15,Amazon18]
  - Can be implemented as look-up table for inference (as in CoreML)

[Facebook15] Gong, Y. et al. (2015) Compressing Deep Convolutional Networks using Vector Quantization arXiv:1412.6115
[Amazon18] Strimel, G. P. et al. (2018). Statistical Model Compression for Small-Footprint Natural Language Understanding. ArXiv:1807.07520

Deep Compression Methods
Quantization (4/4)

- Comparison of Post-Train Quantization [*]
  - Uniform vs. adaptive Quantization for different number of bits
  - Two Variants of Acoustic Detection DNNs





[*] Experiments conducted at AudioLabs, Fraunhofer IIS, Erlangen

GETTING AI IN YOUR POCKET
TABLE OF CONTENTS

1. Motivation
2. State-of-the-Art
3. **Methods**

   Pruning → Low-rank approximation → Quantization → Entropy Coding

4. Summary

Fraunhofer IIS

Deep Compression Methods
Entropy Coding

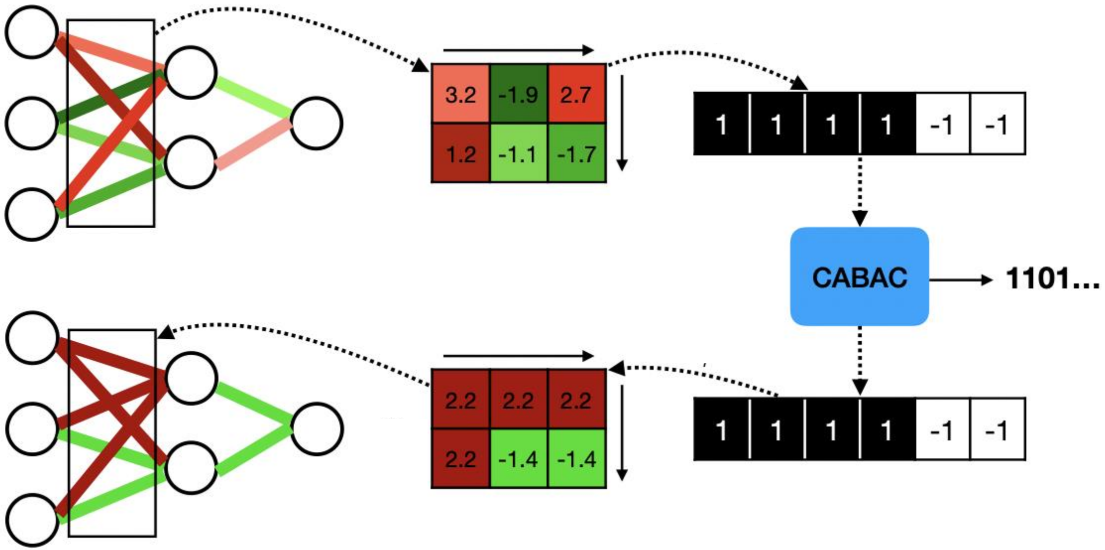


Image courtesy of Fraunhofer HHI, Wojciech Samek & Simon Wiedemann

Deep Compression Methods
Entropy Coding

- Lossless compression (zip)
- Works on the quantized representation
- Creates an minimal bitstream [HHI19]
- Representation is non-uniform,
  weights are mapped to a variable number of bits

[HHI19] Wiedemann, Simon, et al. (2019) "DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks." *arXiv:1907.11900*

GETTING AI IN YOUR POCKET

TABLE OF CONTENTS

Fraunhofer IIS

Getting AI in Your Pocket
Summary (1/2)

- Motivation
  - DNNs require a large amount of computing power
  - Trained DNN models have redundancy
  - This can be exploited for embedded deployment
- State of the Art
  - Hardware vendors provide simple deployment tools
  - Bigger gains are achieved by combinations of
- Deep Compression Methods
  - Pruning removes part of the network
  - Low rank Approximation exploits sparsity directly
  - Quantization reduces representation accuracy
  - Entropy Coding for lossless compression

Getting AI in Your Pocket
Summary (2/2)

- Future Work at Fraunhofer IIS
  - Working on good ‘recipes’ for making deep learning applications in audio and video processing efficient
  - Trainings in Machine Learning
  - We are hiring
- Further Information
  - www.iis.fraunhofer.de/amm/
  - www.audioblog.iis.fraunhofer.com
  - amm-info@iis.fraunhofer.de

SPONSORED BY THE

Federal Ministry
of Education
and Research

The results originate in part from a project funded by the German federal Ministry of Education and Research under the reference number 01IS19070A.
The responsibility for the content rests with the authors.