# A Universal Compression Algorithm for Deep Neural Networks

Fraunhofer Heinrich Hertz Institute (HHI)

Machine Learning Group

Dr. Wojciech Samek

# Deep Learning "Revolution"

Images, Text, Speech, Games ...

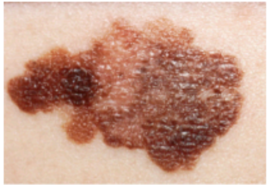
AlphaGo beats Go human champ

Computer out-plays humans in "doom"


Deep Net outperforms humans in image classification

IMAGENET

Dermatologist-level classification of skin cancer with Deep Nets

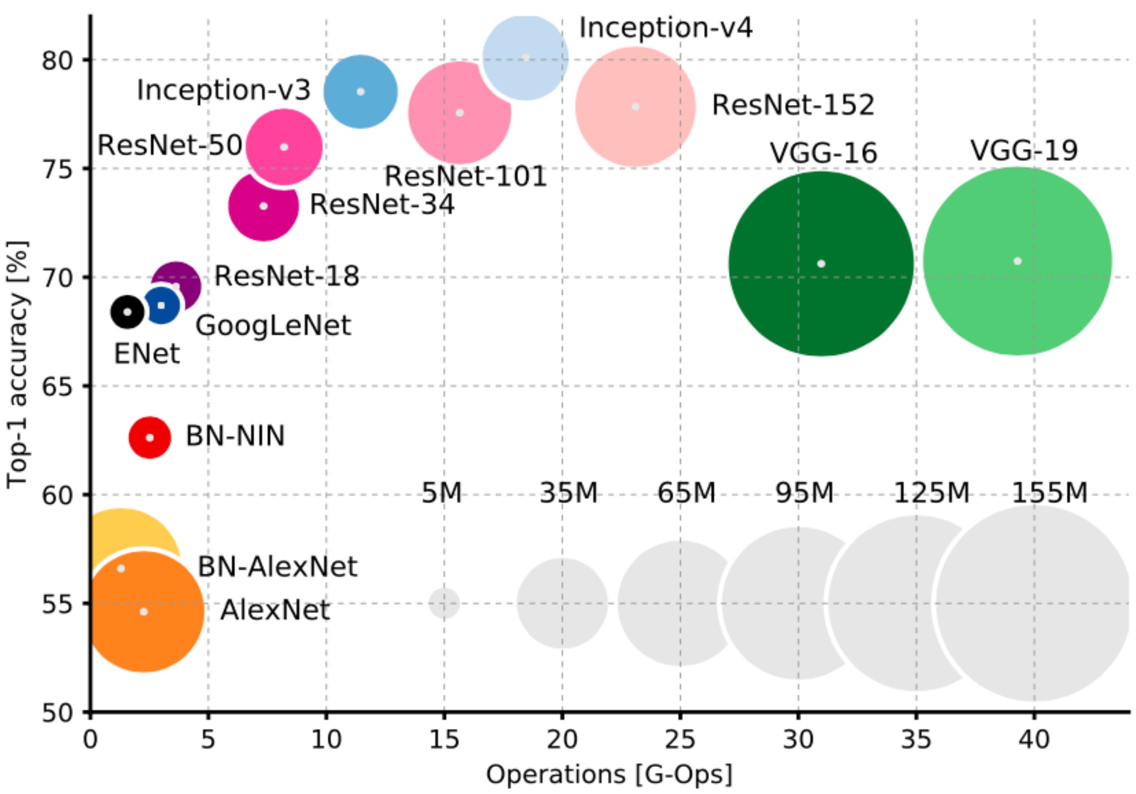


Revolutionizing Radiology with Deep Learning

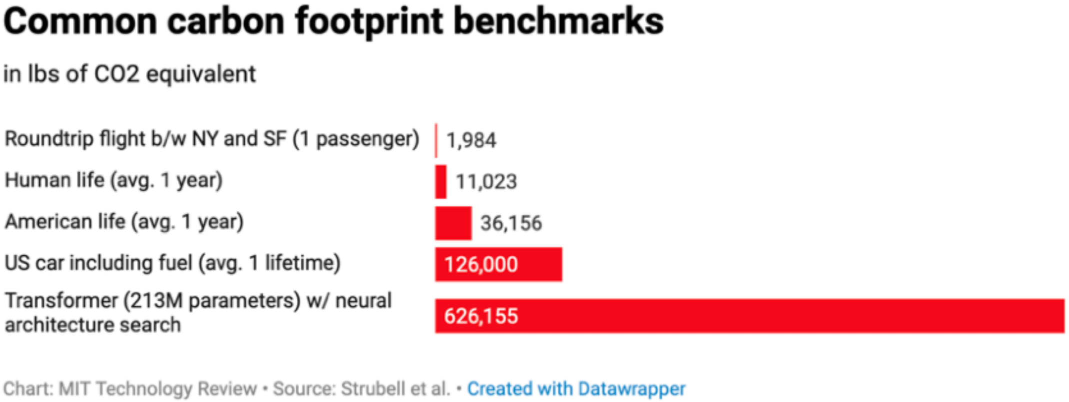
**Ingredients for the success**

1. Large volumes of data
2. Large (Deep) Models
3. Large Computing Power

# Complexity of DNN is Growing



# Large Computational Resources Needed

# Processing at the "Edge"

JACK STEWART TRANSPORTATION 02.06.18 08:00 AM

## SELF-DRIVING CARS USE CRAZY AMOUNTS OF POWER, AND IT'S BECOMING A PROBLEM

Shelley, a self-driving Audi TT developed by Stanford University, uses the brains in the trunk to speed around a racetrack autonomously.

NIKKI KAHN/THE WASHINGTON POST/GETTY IMAGES

WIRED

(Feb 2018)

Cameras and radar generate ~6 gigabytes of data every 30 seconds.

**Self-driving car prototypes use approximately 2,500 Watts of computing power.**

Generates wasted heat and some prototypes need water-cooling!

[slide from V. Sze]

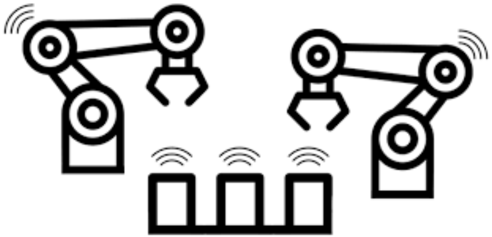
# Processing at the "Edge"

## On-device deep learning


## Distributed Data & Privacy


## Latency & bandwidth constraints





# Processing at the “Edge”

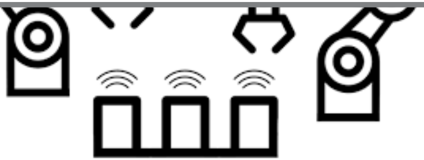
On-device deep learning

Distributed Data & Privacy


We need techniques to reduces the computational complexity
(i.e., storage, memory, energy, runtime, communication overhead)

Laten





# MPEG-7 Part 17

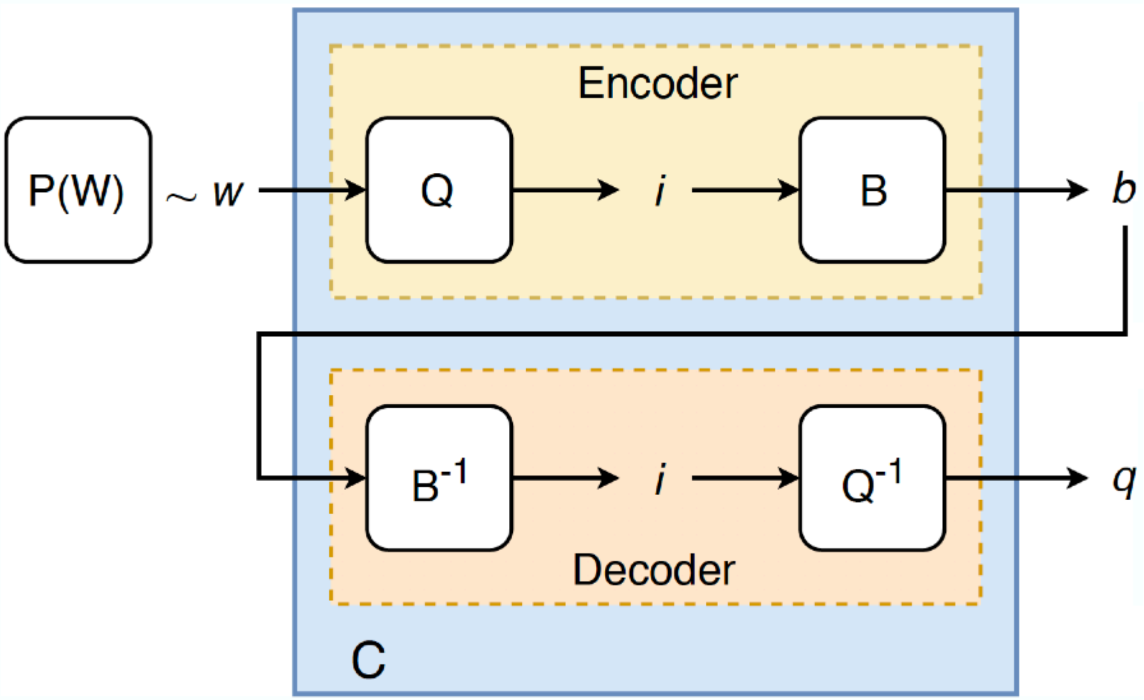
Standard on "Compression of Neural Networks for Multimedia Content Description and Analysis"

# Outline of this talk

1. Background: Quantization & Encoding
2. DeepCABAC
3. Compression in Federated Learning

# Background: Quantization & Encoding

# Source Coding




Represent a signal with the minimum number of (binary) symbols without exceeding an “acceptable level of distortion".

**Lossy Step + Lossless Step**

# Source Coding

**Goal**: Minimize the rate-distortion objective:

$$\boxed{C^* = \arg\min_{C} \mathbb{E}_{P(w)} \left[ D(w,q) + \lambda L_C(b) \right]}$$

where $b = (B \circ Q)(w)$ and $q = (Q^{-1} \circ Q)(w)$.

# Lossless Coding

The minimum information required to fully represent a sample w that has probability P(w) is of $-P \log_2 P(w)$ bits (*Shannon*).

**Challenges**:

- Decoder does not know P(w)
- P(w) may be non-stationary

# Lossless Coding: Desired Properties

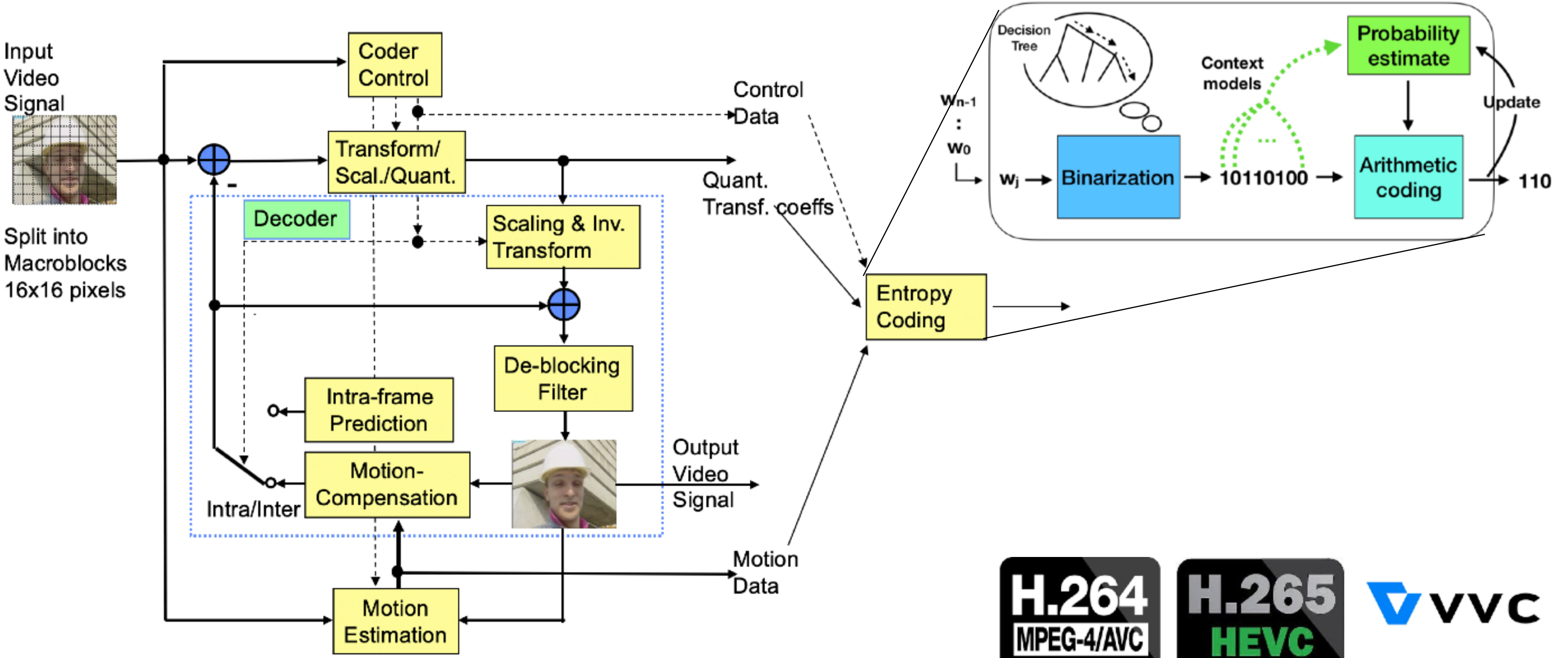
**Universality:** The code should have a mechanism that allows it to adapt its probability model to a wide range of different types of input distributions, in a sample-efficient manner.

**Minimal redundancy:** The code should produce binary representations of minimal redundancy with regards to its probability estimate.

**High efficiency:** The code should have high coding efficiency, meaning that encoding/decoding should have high throughput.

# Video Coding Standards

Context-based Adaptive Binary Arithmetic Coding (CABAC)




H.264 MPEG-4/AVC · H.265 HEVC · VVC

# NN Coding

In NN coding, things are more complicated:

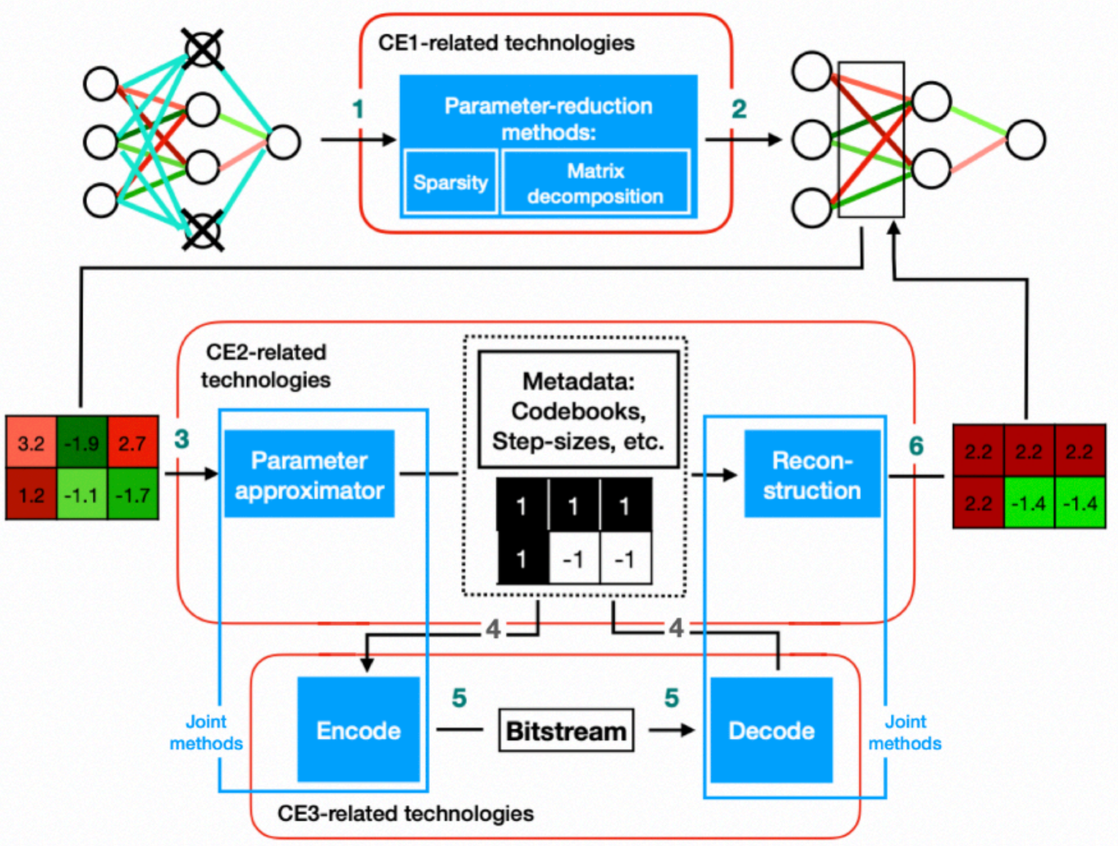
- complex distortion term (non-linear accumulation of errors)

- no clear structure in NN weights (e.g. in video high correlation between frames and neighboring pixels)

- more flexibility (e.g. fine-tuning, sparsification, structural changes)

# NN Coding

is developing a standard on "Compression of Neural Networks for Multimedia Content Description and Analysis"



# DeepCABAC

# Lossy Coding

Finding the optimal code is in most cases NP-hard

$$C^* = \arg\min_{C} \mathbb{E}_{P(w)} \left[ D(w,q) + \lambda L_C(b) \right]$$

**Idea**: Fix the binarization map B by selecting a particular (universal) lossless code. Then just need to find a scalar quantizer

$$(Q, Q^{-1})^* = \arg\min_{(Q,Q^{-1})} \mathbb{E}_{P(w_j)} \left[ D(w_j, q_j) + \lambda L_Q(b_j) \right]$$

# From Source Coding to NN Coding

$$\boxed{(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \sum_{(x,y)\in\mathbb{D}} \mathcal{L}(y'', y') + \lambda L_Q(b)}$$

$$y' \sim P(y'|x, w) \qquad y'' \sim P(y''|x, q)$$

$$\begin{bmatrix} 0.1 \\ 0.05 \\ 0.85 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0.05 \\ 0.1 \\ 0.8 \\ 0.05 \end{bmatrix}$$

# From Source Coding to NN Coding

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \sum_{(x,y)\in\mathbb{D}} \mathcal{L}(y'', y') + \lambda L_Q(b)$$

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \sum_{(x,y)\in\mathbb{D}} D_{KL}(y''||y') + \lambda L_Q(b)$$

Use KL-divergence as distortion measure

# From Source Coding to NN Coding

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \sum_{(x,y)\in\mathbb{D}} \mathcal{L}(y'', y') + \lambda L_Q(b)$$

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \sum_{(x,y)\in\mathbb{D}} D_{KL}(y''||y') + \lambda L_Q(b)$$

$$(Q, Q^{-1})^* = \min_{(Q,Q^{-1})} (q - w)F(q - w)^T + \lambda L_Q(b)$$

If the output distributions do not differ too much, we can approximate KL with the Fisher Information Matrix (FIM)

# From Source Coding to NN Coding

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \sum_{(x,y)\in\mathbb{D}} \mathcal{L}(y'', y') + \lambda L_Q(b)$$

↓

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \sum_{(x,y)\in\mathbb{D}} D_{KL}(y''||y') + \lambda L_Q(b)$$

↓

$$(Q, Q^{-1})^* = \min_{(Q,Q^{-1})} (q-w)F(q-w)^T + \lambda L_Q(b)$$

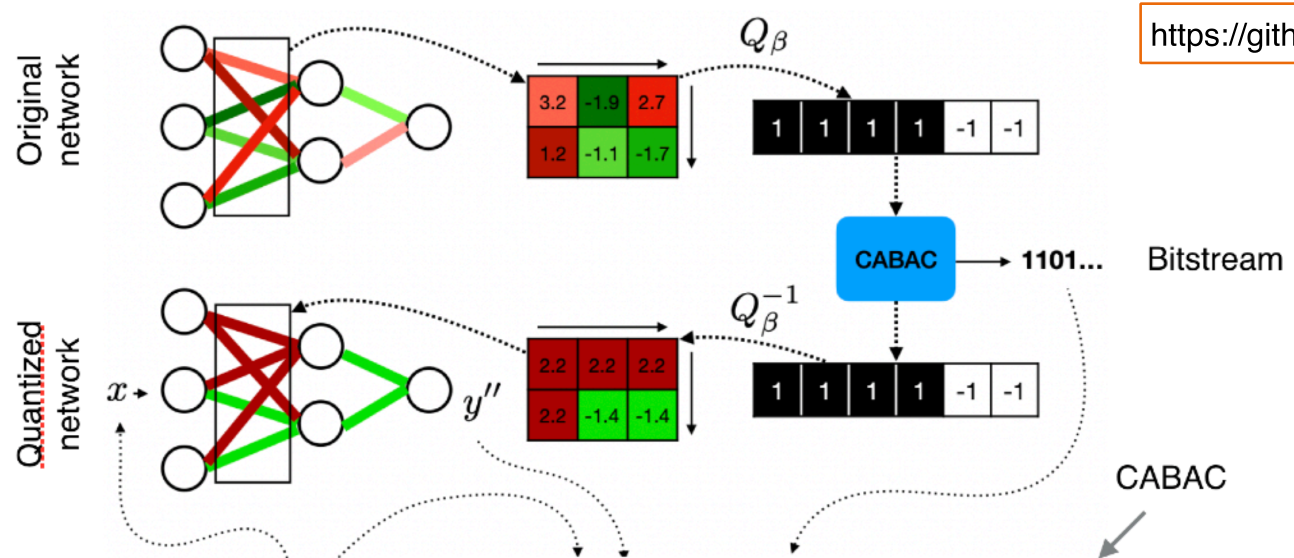
↓

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min} \; F_i(q_i - w_i)^2 + \lambda L_Q(b)$$

Approximate FIM by only its diagonal elements

# DeepCABAC: Weighted RD-based Quantization + CABAC

https://github.com/fraunhoferhhi/DeepCABAC




Data $(x, y)$

$$(Q, Q^{-1})^* = \underset{(Q,Q^{-1})}{\arg\min}\ F_i(q_i - w_i)^2 + \lambda L_Q(b)$$

**DeepCABAC-v1**

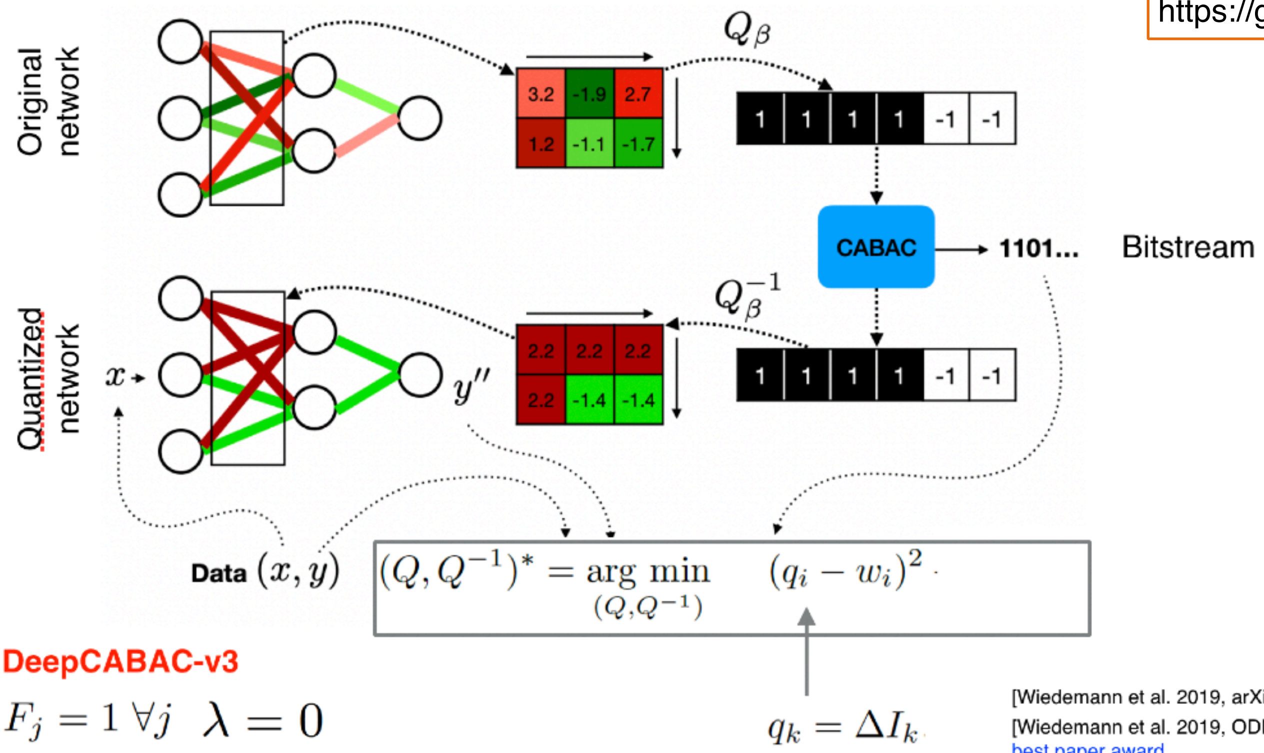
Parametrize each weight parameter as Gaussian. $F_i = 1/\sigma_i$

$$q_k = \Delta I_k$$

[Wiedemann et al. 2019, arXiv:1907.11900]
[Wiedemann et al. 2019, ODML-CDNNR] best paper award

# DeepCABAC: Uniform Quantization + CABAC

https://github.com/fraunhoferhhi/DeepCABAC




**DeepCABAC-v3**

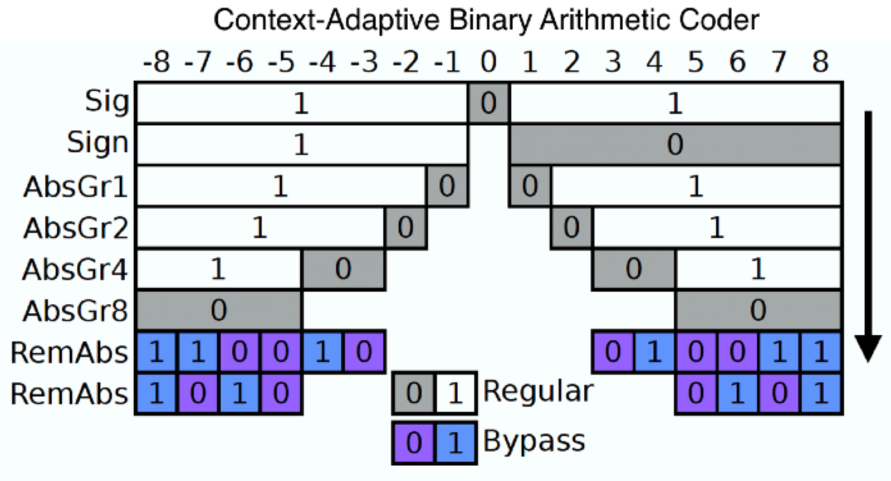
$F_j = 1 \; \forall j \quad \lambda = 0$

[Wiedemann et al. 2019, arXiv:1907.11900]

[Wiedemann et al. 2019, ODML-CDNNR]

best paper award

# Properties of CABAC




Examples

$$1 \rightarrow 100$$
$$-4 \rightarrow 111101$$
$$7 \rightarrow 10111010$$

**Properties of CABAC**

Binarization: represents each unique input value as a sequence of binary decisions.

Context modelling: probability model for each decision, which is updated on-the-fly by the local statistics of the data -> universality.

Arithmetic coding: arithmetic coding for each bit -> minimal redundancy + high efficiency

# Some Results

| Sparse Models (sparsity [%]) | Org. Acc. Top1 [%] | Os_size [MB] | DeepCABAC (acc. [%]) |
|---|---|---|---|
| VGG16 (9.85) | 69.43 | 553.43 | **1.57** (69.43) |
| ResNet50 (74.12) | 74.09 | 102.23 | **4.74** (73.65) |
| Small-VGG16 (7.57) | 91.35 | 60.01 | **1.6** (91.00) |
| LeNet5 (1.90) | 99.22 | 1.72 | **0.72** (99.16) |

# Some Results

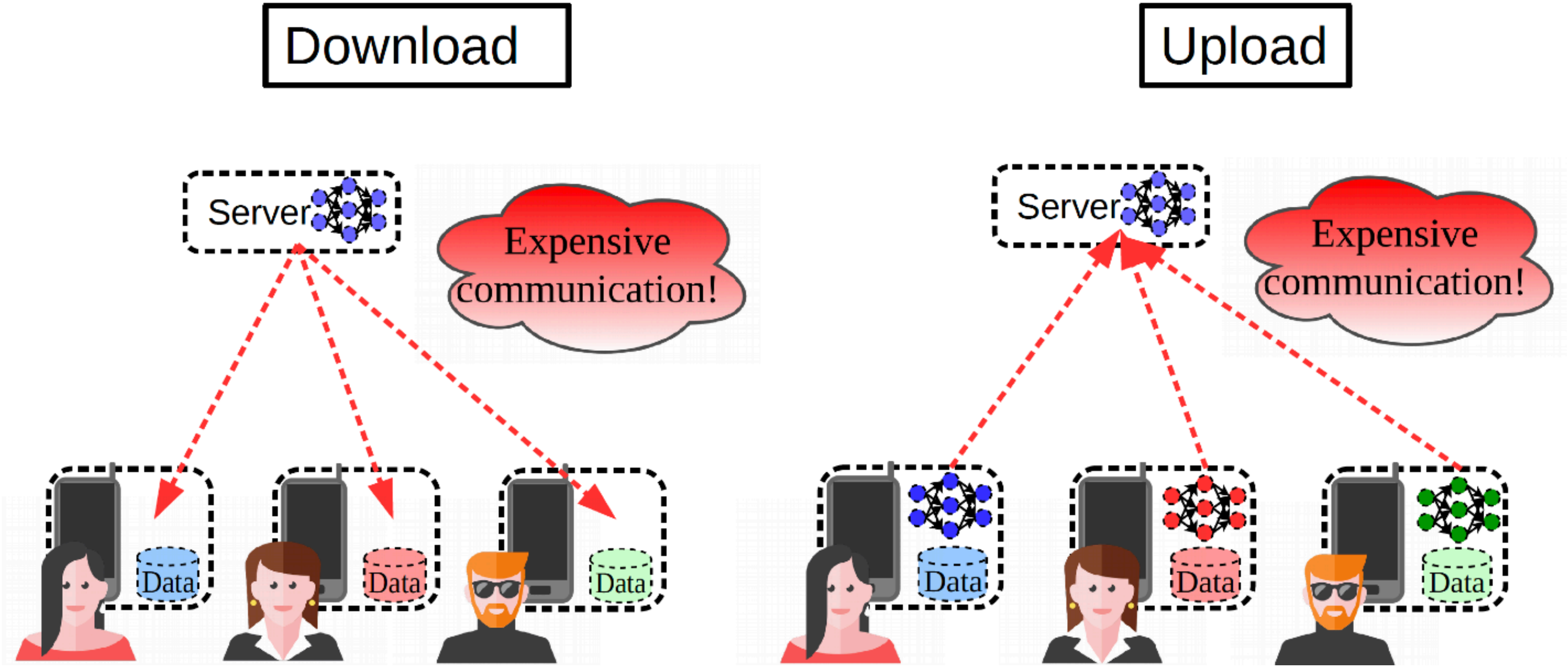
| Sparse Models (sparsity [%]) | Org. Acc. Top1 [%] | Os size [MB] | DeepCABAC (acc. [%]) |
|---|---|---|---|
| VGG16 | 69.43 | 553.43 | **1.57** |
| Small-VGG16 (7.57) | 91.35 | 60.01 | **1.6** (91.00) |
| LeNet5 (1.90) | 99.22 | 1.72 | **0.72** (99.16) |

VGG16 **553.4MB** -> **8.7MB** at an acc. **69.43%**

ResNet50 **102.2MB->** **4.85MB** at an acc. **73.65%**

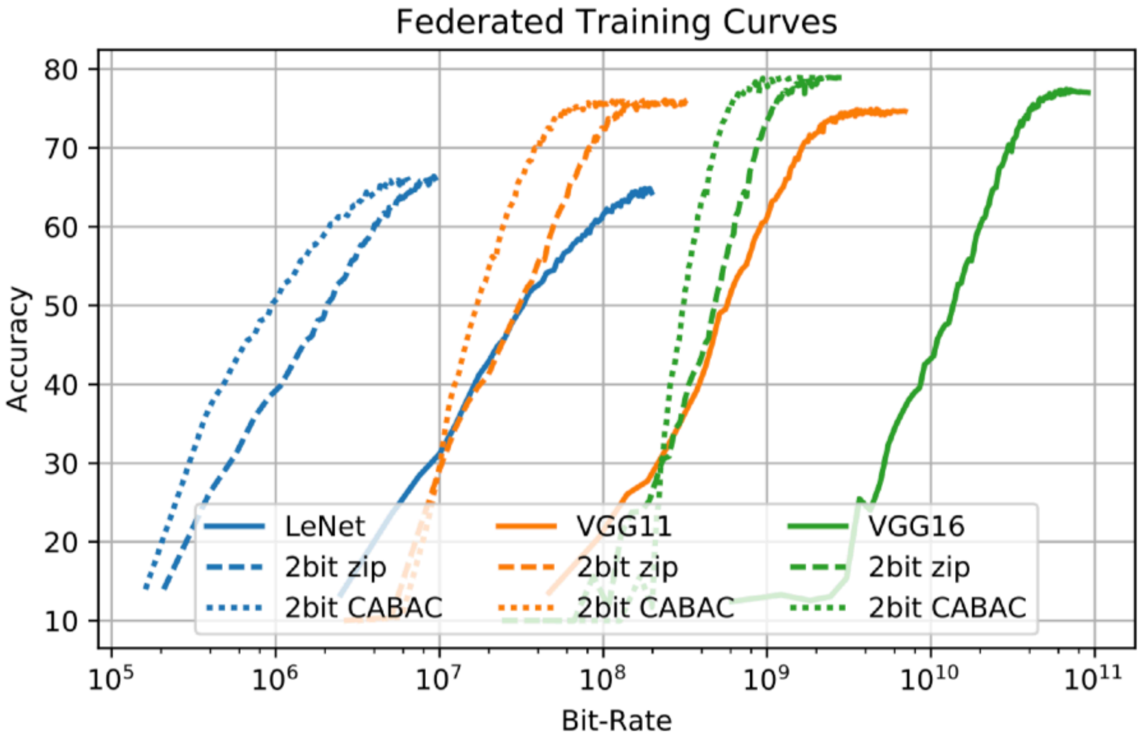
# Compression in Federated Learning

# Federated Learning



# Federated Learning

Plug & Play compression by DeepCABAC




[Neumann et al. 2020, IEEE ICIP]

# Federated Learning

Total Communication = [#Communication Rounds] x [#Parameters] x [Avg. Codeword length]

**Case Study: VGG16 on ImageNet**

| | |
|---|---|
| – Number of Iterations until Convergence: | 900.000 |
| – Number of Parameters: | 138.000.000 |
| – Bits per Parameter: | 32 |

→ Total Communication = **496.8 Terabyte** (Upload+Download)

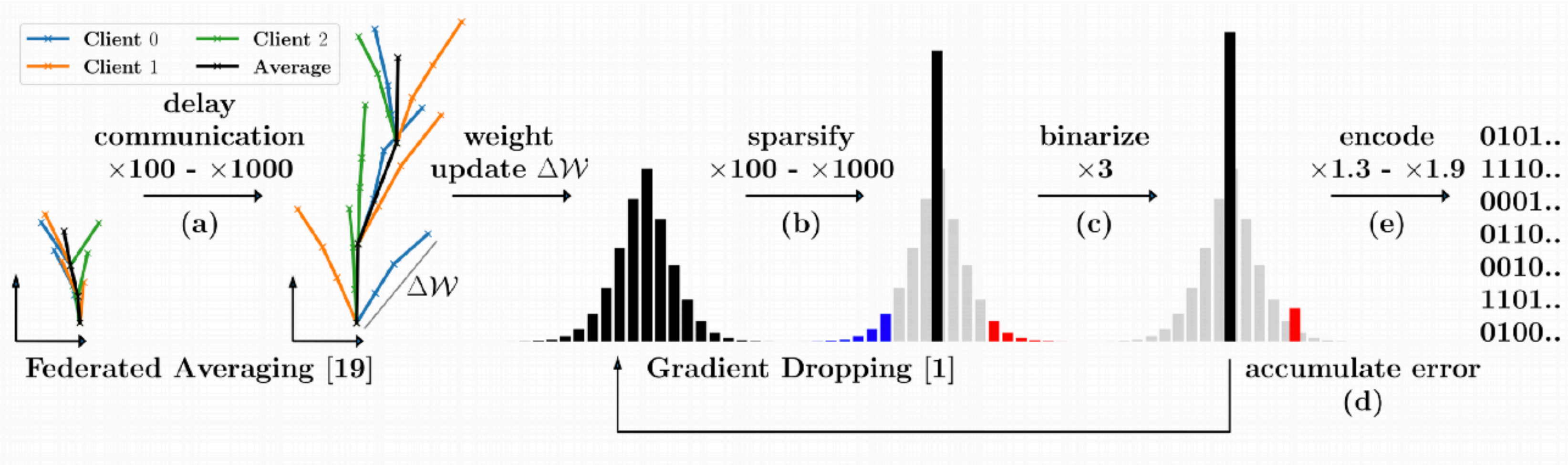
# Federated Learning

Total Communication = [#Communication Rounds] x [#Parameters] x [Avg. Codeword length]

**Compression Methods**

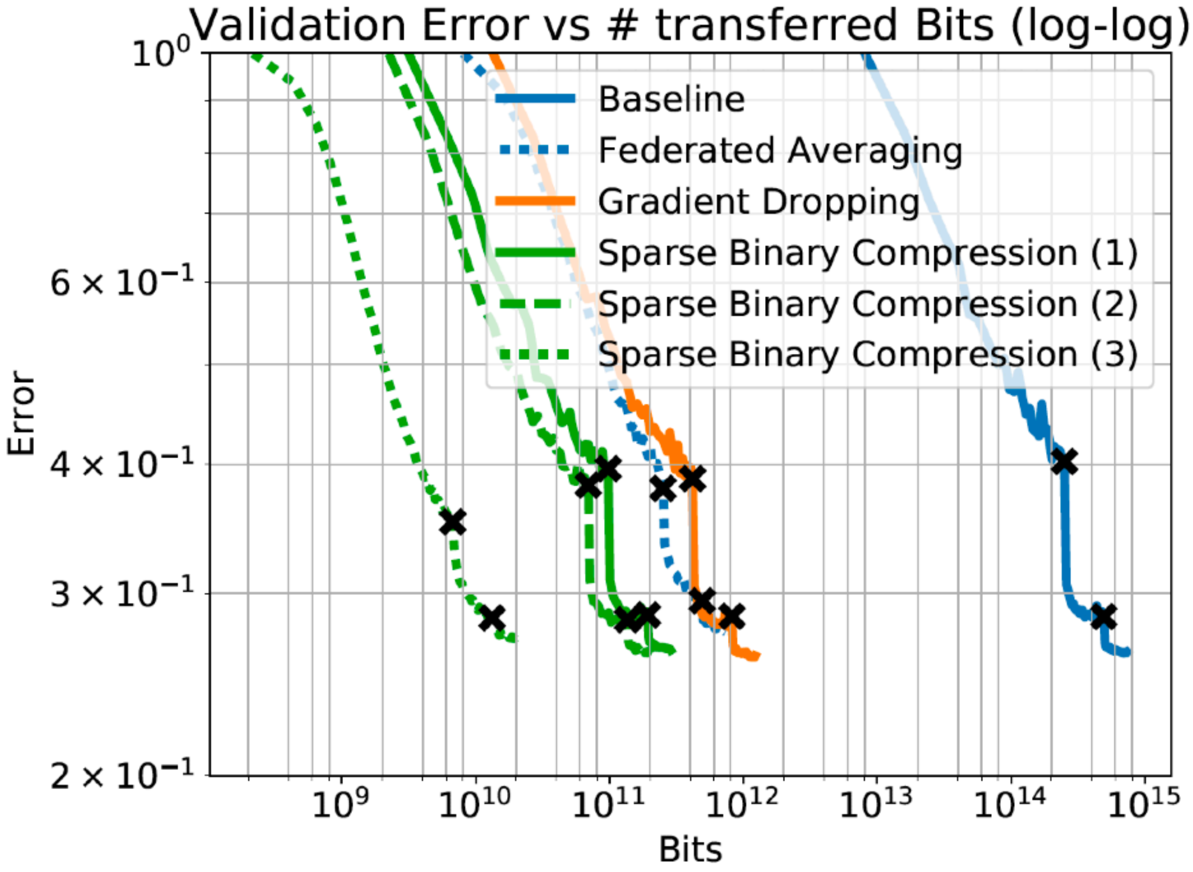
- Communication Delay
- Lossy Compression: Unbiased
- Lossy Compression: Biased
- Efficient Encoding

# Federated Learning



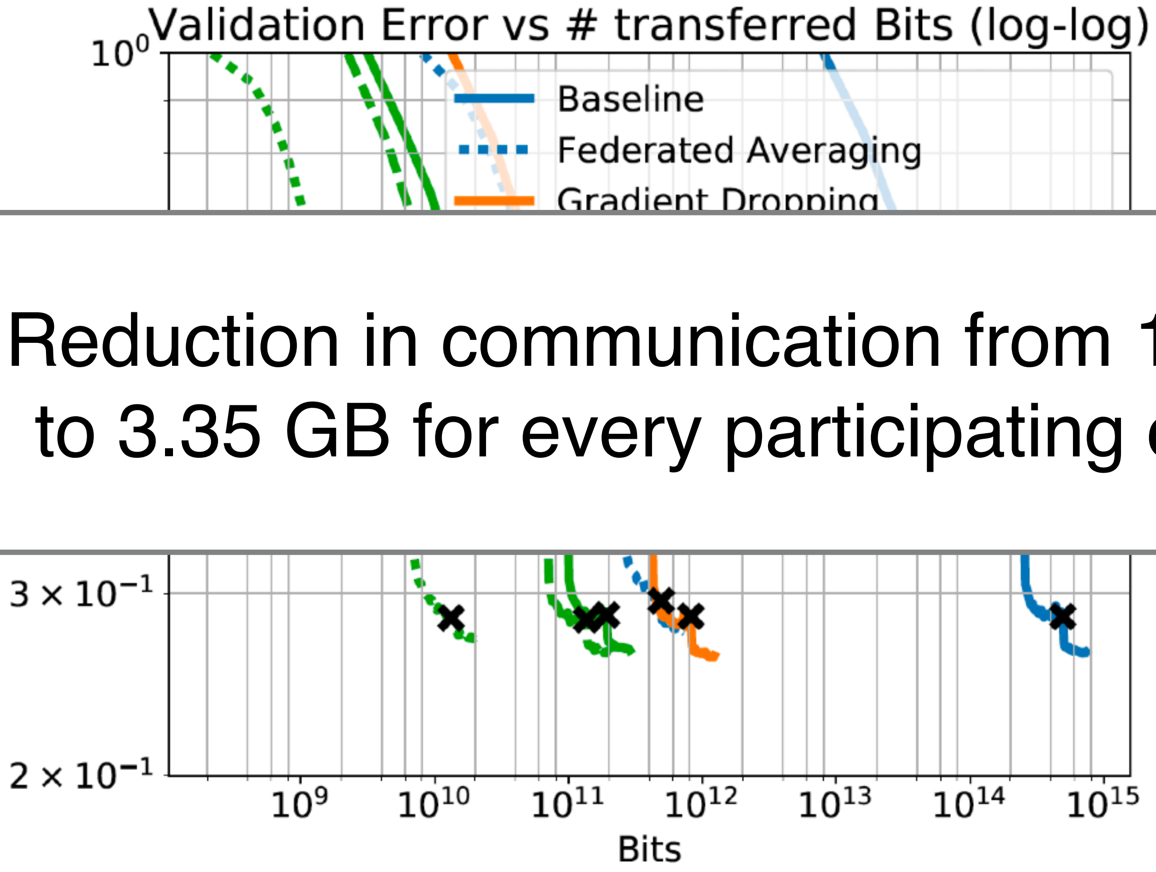

Sattler, et al. "Sparse binary compression: Towards distributed deep learning with minimal communication." 2019 International Joint Conference on Neural Networks (IJCNN).

# Federated Learning




[Sattler et al. 2019, IEEE IJCNN]

# Federated Learning




Reduction in communication from 125 TB to 3.35 GB for every participating client.

[Sattler et al. 2019, IEEE IJCNN]

# Federated Learning

| Compression Method ⟶ | | Baseline | DGC [3] | Fed. Avg.[4] | SBC (1) | SBC (2) | SBC (3) |
|---|---|---|---|---|---|---|---|
| LeNet5-Caffe @MNIST | Accuracy | 0.9946 | 0.994 | 0.994 | 0.994 | 0.994 | 0.991 |
| | Compression | ×1 | ×718 | ×500 | ×2071 | ×3166 | ×24935 |
| ResNet18 @CIFAR10 | Accuracy | 0.946 | 0.9383 | 0.9279 | 0.9422 | 0.9435 | 0.9219 |
| | Compression | ×1 | ×768 | ×1000 | ×2369 | ×3491 | × 31664 |
| ResNet34 @CIFAR100 | Accuracy | 0.773 | 0.767 | 0.7316 | 0.767 | 0.7655 | 0.701 |
| | Compression | ×1 | ×718 | ×1000 | ×2370 | ×3166 | ×31664 |
| ResNet50 @ImageNet | Accuracy | 0.737 | 0.739 | 0.724 | 0.735 | 0.737 | 0.728 |
| | Compression | ×1 | ×601 | ×1000 | ×2569 | ×3531 | ×37208 |
| WordLSTM @PTB | Perplexity | 76.02 | 75.98 | 76.37 | 77.73 | 78.19 | 77.57 |
| | Compression | ×1 | ×719 | ×1000 | ×2371 | ×3165 | ×31658 |
| WordLSTM* @WIKI | Perplexity | 101.5 | 102.318 | 131.51 | 103.95 | 103.95 | 104.62 |
| | Compression | ×1 | ×719 | ×1000 | ×2371 | ×3165 | ×31657 |

# Next Standard ?

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**
**ORGANISATION INTERNATIONALE DE NORMALISATION**
**ISO/IEC JTC1/SC29/WG11**
**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11/N19228**

**April 2020, Alpbach, AT**

| | |
|---|---|
| **Source** | **Video** |
| **Status** | **Approved** |
| **Title** | **Call for Incremental NNR Test Materials** |

# Conclusion

Efficiency in storage, memory, energy, runtime, communication ..

DeepCABAC based on established compression technology

Different options (e.g. fine-tuning, structural changes, NAS)

Hardware co-design is crucial

MPEG standardization is moving forward

# References

## Neural Network Compression


S Wiedemann, H Kirchhoffer, S Matlage, P Haase, A Marban, T Marinc, D Neumann, T Nguyen, A Osman, H Schwarz, D Marpe, T Wiegand, W Samek. DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):700-714, 2020.
`http://dx.doi.org/10.1109/JSTSP.2020.2969554`

S Yeom, P Seegerer, S Lapuschkin, S Wiedemann, KR Müller, W Samek. Pruning by Explaining: A Novel Criterion for Deep Neural Network Pruning. *arXiv:1912.08881*, 2019.
`https://arxiv.org/abs/1912.08881`

S Wiedemann, H Kirchhoffer, S Matlage, P Haase, A Marban, T Marinc, D Neumann, A Osman, D Marpe, H Schwarz, T Wiegand, W Samek. DeepCABAC: Context-adaptive binary arithmetic coding for deep neural network compression. *Joint ICML'19 Workshop on On-Device Machine Learning & Compact Deep Neural Network Representations (ODML-CDNNR)*, 1-4, 2019. *** **Best paper award** ***
`https://arxiv.org/abs/1905.08318`

S Wiedemann, A Marban, KR Müller, W Samek. Entropy-Constrained Training of Deep Neural Networks. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 1-8, 2019.
`http://dx.doi.org/10.1109/IJCNN.2019.8852119`

# References

## Efficient Deep Learning

S Wiedemann, KR Müller, W Samek. Compact and Computationally Efficient Representation of Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 31(3):772-785, 2020.
`http://dx.doi.org/10.1109/TNNLS.2019.2910073`

S Wiedemann, T Mehari, K Kepp, W Samek. Dithered backprop: A sparse and quantized backpropagation algorithm for more efficient deep neural network training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 3096-3104*, 2020.
`https://dx.doi.org/10.1109/CVPRW50498.2020.00368`

A Marban, D Becking, S Wiedemann, W Samek. Learning Sparse & Ternary Neural Networks with Entropy-Constrained Trained Ternarization (EC2T). *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 3105-3113*, 2020.
`https://dx.doi.org/10.1109/CVPRW50498.2020.00369`

# References

## Federated Learning


F Sattler, T Wiegand, W Samek. Trends and Advancements in Deep Neural Network Communication. *ITU Journal: ICT Discoveries*, 3(1), 2020.
https://www.itu.int/en/journal/2020/001/Pages/07.aspx

F Sattler, KR Müller, W Samek. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
https://arxiv.org/abs/1910.01991

F Sattler, S Wiedemann, KR Müller, W Samek. Robust and Communication-Efficient Federated Learning from Non-IID Data. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.
http://dx.doi.org/10.1109/TNNLS.2019.2944481

F Sattler, KR Müller, W Samek. Clustered Federated Learning. *Proceedings of the NeurIPS'19 Workshop on Federated Learning for Data Privacy and Confidentiality*, 1-5, 2019.

# References

## Federated Learning

F Sattler, KR Müller, T Wiegand, W Samek. On the Byzantine Robustness of Clustered Federated Learning. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8861-8865, 2020.
`http://dx.doi.org/10.1109/ICASSP40776.2020.9054676`

F Sattler, S Wiedemann, KR Müller, W Samek. Sparse Binary Compression: Towards Distributed Deep Learning with minimal Communication. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 1-8, 2019.
`http://dx.doi.org/10.1109/IJCNN.2019.8852172`

D Neumann, F Sattler, H Kirchhoffer, S Wiedemann, K Müller, H Schwarz, T Wiegand, D Marpe, W Samek. DeepCABAC: Plug&Play Compression of Neural Network Weights and Weight Updates. Proceedings of the IEEE International Conference on Image Processing (ICIP), 2020.

# Slides and Papers available at

www.federated-ml.org