EFFICIENT NN W/O MULTIPLIERS

PUBLIC

Sebastian Vogel
NXP - CTO Automotive System Innovations

MARCH 2022

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

EMBEDDED AI RESEARCH SCIENTIST, SEBASTIAN VOGEL

- Sebastian Vogel
  - PhD in “Efficient Processing of DNNs” from RWTH Aachen, Germany
  - 2016-2021 with Bosch Corporate Research, Renningen, Germany
    - Quantization, Hardware-Accelerator Architectures for DNNs, NAS
  - At NXP since Feb. 2021 as research scientist for Embedded AI
    - Hardware-aware Neural Architecture Search, Quantization
  - **Presentation mostly shows work published while at Bosch Research**

- NXP department: CTO Automotive System Innovations (‘R&D’)
  - Scouting & analysing AI research (in-house, via university collaborations)
  - Translate recent SOTA to NXP requirements & research projects
  - Small impactful projects with opportunities for student assignments

Bosch Research in Renningen
(Research Campus) *

NXP headquarters in Eindhoven
(High Tech Campus) **

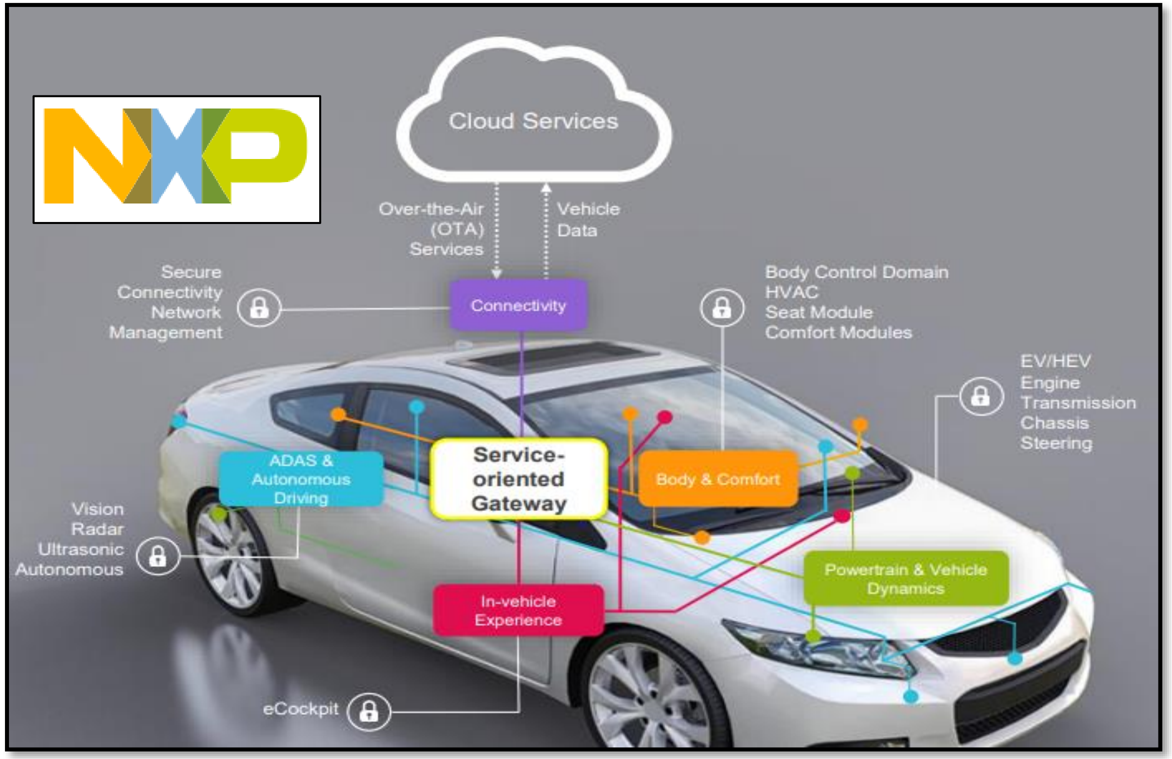
* source Bosch Campus: https://www.bosch.com/research/about-research/research-locations/
** source NXP Headquarters: https://www.nxp.com/company/about-nxp/worldwide-locations/netherlands:NETHERLANDS

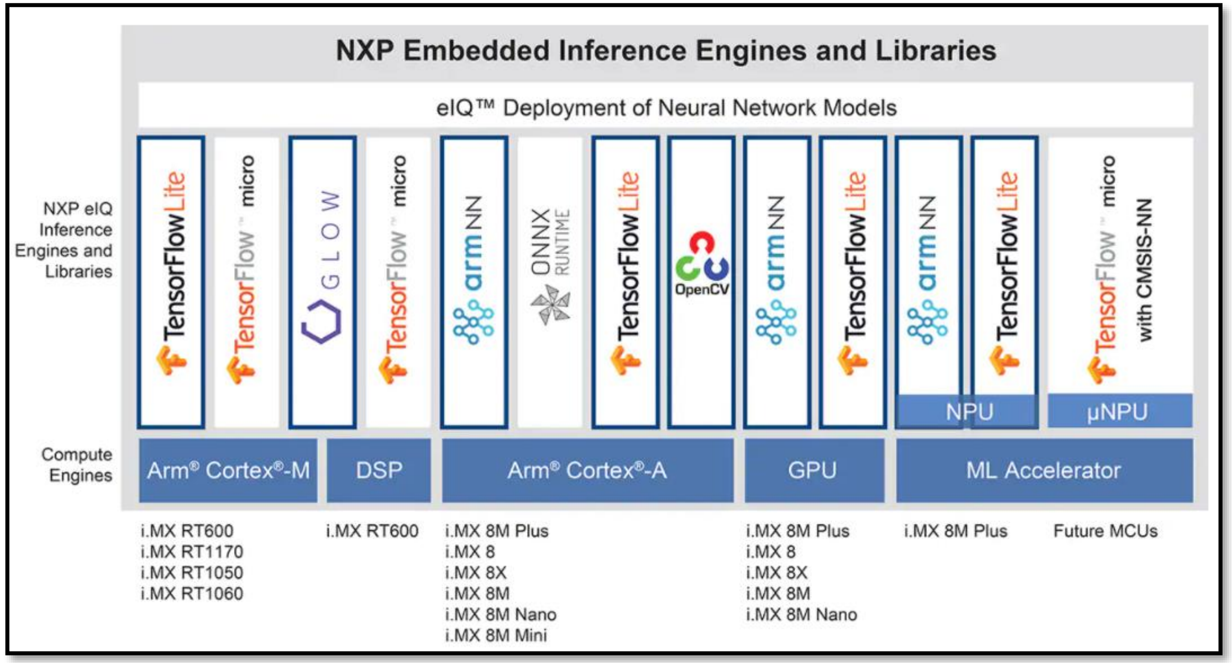
PORTFOLIO OF NXP

- Functionality:
  - Compute
  - Connectivity
  - HMI
- Data:
  - Radar
  - UWB
  - Analytics
  - Vision
- Applications:
  - Automotive
  - IoT/edge
  - Industrial automation
  - Drones

For AI deployment:
- Applications
- Chips
- Constraints

→ **different requirements on neural network architectures**



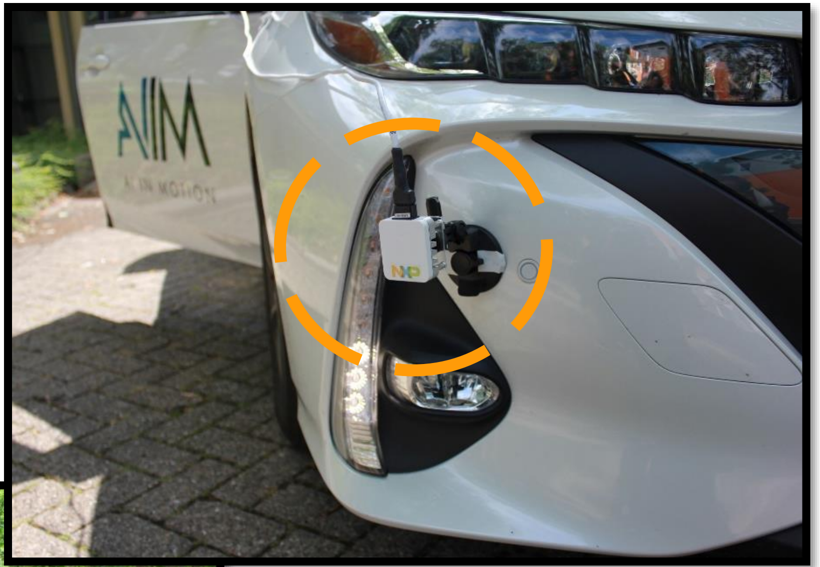


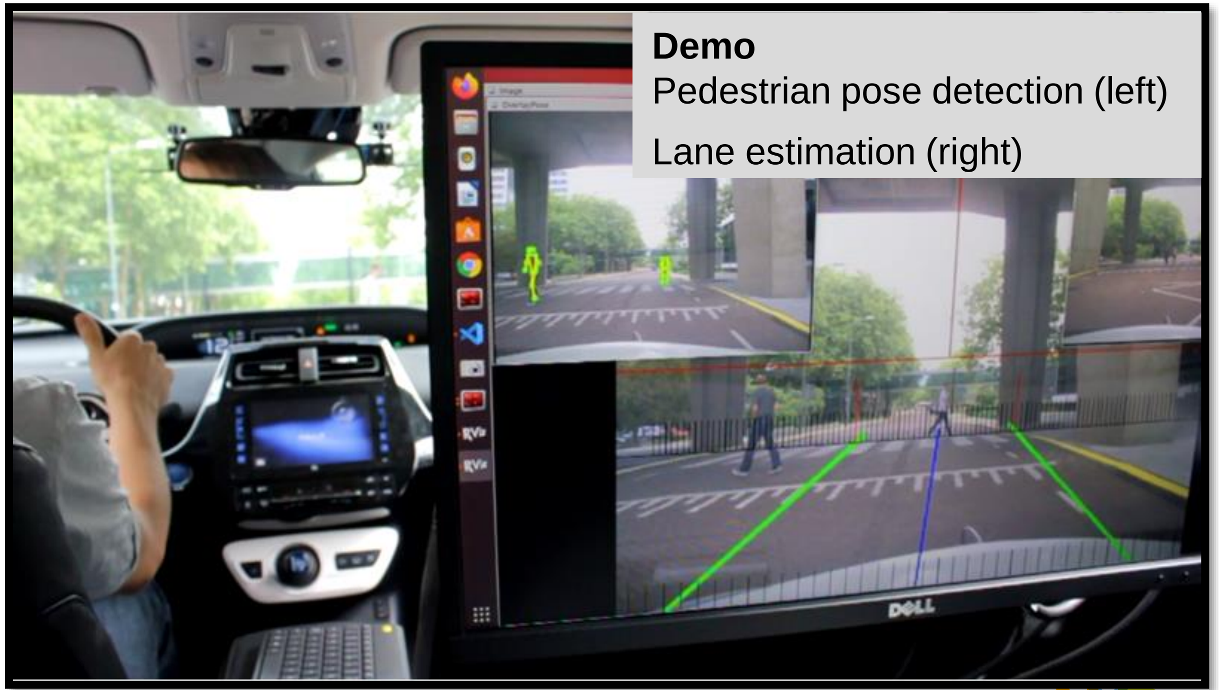
Slide courtesy: Gerardo Daalderop gerardo.daalderop@nxp.com

NXP CTO (‘R&D’)

Automotive System Innovations (ASI)

- Prototyping systems with NXP solutions, e.g.:
- Radar, AI/ML ‘brain’, Network
- In-house & collaborations



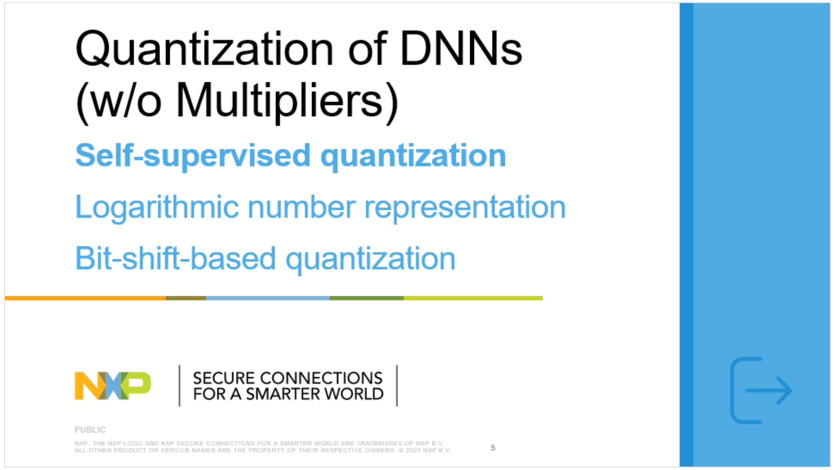


Demo
Pedestrian pose detection (left)
Lane estimation (right)

Slide courtesy: Willem Sanberg willem.sanberg@nxp.com

EFFICIENT NNS WITHOUT MULTIPLIERS
OVERVIEW

- Quantization of Neural Networks w/o Multipliers
  - Self-supervised quantization of pre-trained DNNs
  - Logarithmic quantization at arbitrary base
  - Bit-shift-based quantization



Quantization of DNNs (w/o Multipliers)

Self-supervised quantization

Logarithmic number representation

Bit-shift-based quantization

SECURE CONNECTIONS
FOR A SMARTER WORLD

SELF-SUPERVISED QUANTIZATION OF PRE-TRAINED NEURAL NETWORKS DOES NOT REQUIRE LABELLED TRAINING DATA

- Quantizing pre-trained neural networks, i.e., determining the quantization step size α
  - Without the need for labeled training data through self-supervised quantization[7]
  - Unlabeled calibration enough

$$quant(\cdot):\ y \mapsto y_q = \alpha \cdot \mathrm{clip}\left(\mathrm{round}\left(\frac{y}{\alpha}\right), -2^{N-1}, 2^{N-1}-1\right)$$

$$y^{(l)} = \Phi\left(b^{(l)} + \sum w^{(l)} x^{(l)}\right)$$

$$y_q^{(l)} = quant\left(y^{(l)}, \alpha\right) = y^{(l)} + \underbrace{y_\Delta^{(l)}}_{\text{QE}}$$

$$\tilde{y}^{(l+1)} = \Phi\left(b^{(l+1)} + \sum w^{(l+1)} y_q^{(l)}\right)$$

$$\tilde{y}^{(l+1)} = \Phi\left(b^{(l+1)} + \sum \boxed{w^{(l+1)}}\left(y^{(l)} \boxed{+ y_\Delta^{(l)}}\right)\right)$$

$$= \Phi\left(b^{(l+1)} + \sum w^{(l+1)} y^{(l)}\right) + y_{p\Delta}^{(l)}$$

$$= y^{(l+1)} + \underbrace{y_{p\Delta}^{(l)}}_{\text{propQE}}$$

Option 1: Minimize the squared QE

$$\alpha = argmin\left({y_\Delta^{(l)}}^2\right)$$

Option 2: Minimize squared propagated quantization error

$$\alpha = argmin\left({y_{p\Delta}^{(l)}}^2\right)$$

[7] Vogel et al., Self-Supervised Quantization of Pre-Trained Neural Networks for Multiplierless Acceleration, DATE 2019

SELF-SUPERVISED QUANTIZATION OF PRE-TRAINED NEURAL NETWORKS

- 8bit quantization (per-tensor) of activations only

Quantization	VGG16 top-1^{+}	VGG16 top-5^{++}	ResNet50 top-1	ResNet50 top-5	InceptionNet top-1	InceptionNet top-5	Dilated Model mIoU§	Dilated Model pix.acc.$^{\#}$	FCN8s mIoU	FCN8s pix.acc.
	Classification						Semantic Segmentation			
Calibration samples	100						36		36	
Float32 baseline	69.58	89.04	72.99	90.93	75.61	92.48	55.63	92.85	66.48	94.65
y_q max abs (naïve)	66.36	88.82	64.75	86.69	**0.00**	**0.02**	51.70	91.14	64.68	93.41
y_q min MSE (Opt. 1)	68.51	88.79	**70.08**	**88.95**	**69.66**	**89.40**	54.23	92.00	65.04	93.29
y_q min propQE (Opt. 2)	69.09	88.97	**71.31**	**90.61**	**73.89**	**91.67**	55.65	92.79	66.49	94.46
propQE vs baseline	-0.49	-0.07	**-1.68**	**-0.32**	**-1.72**	**-0.81**	+0.02	-0.06	+0.01	-0.19

Float 32bit

[4]

Linear 8bit (params & act.)

[4]

$^{+}$ Top-1 accuracy: % of correctly classified labels
$^{++}$ Top-5 accuracy: % of correct label within first 5 predicted labels
§ mIoU: mean intersection over union
$^{\#}$ pix.acc.: mean overall pixel accuracy

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

Quantization of DNNs w/o Multipliers

Self-supervised quantization

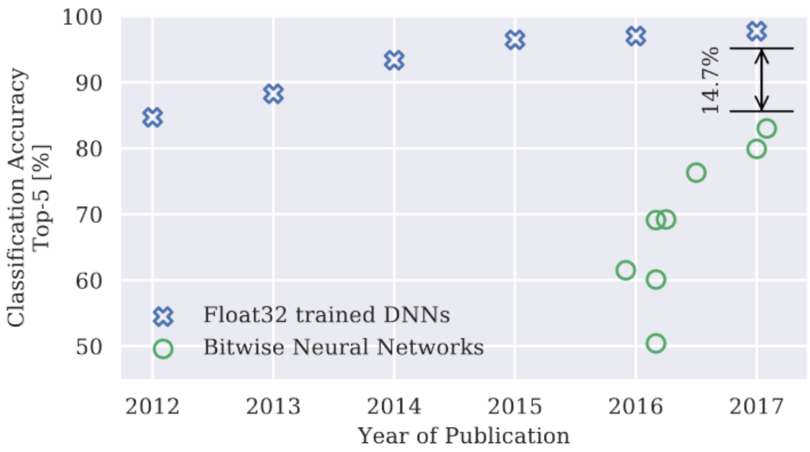
Logarithmic number representation

Bit-shift-based quantization

SECURE CONNECTIONS
FOR A SMARTER WORLD

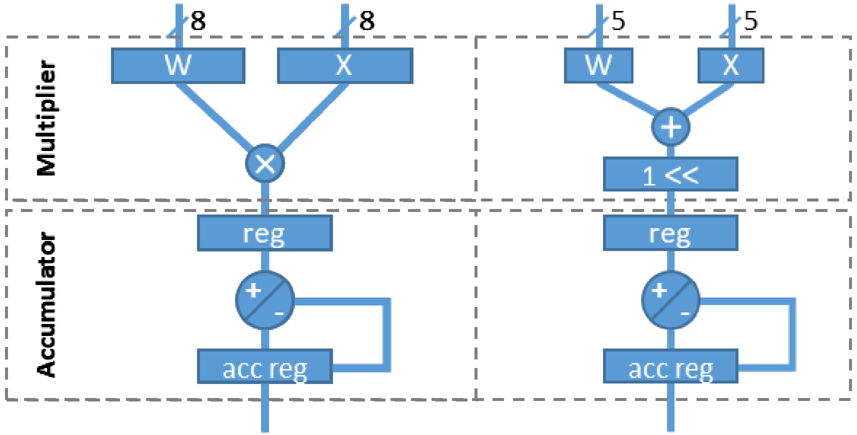
FEW-BIT QUANTIZATION WITH ARBITRARY LOG-BASE IS A PROMISING APPROACH FOR PRESERVING PRE-TRAINED NETWORK ACCURACY

- As of 2018, few-bit-quantization lacked behind SOTA floating point training and resulted in **complex training routines and hard to master training “ingredients”**
- Quantization of pre-trained DNNs favorable



[4]

- CNN accelerators incorporate a considerable amount of multiply-accumulate (MAC) engines
- Reducing the bit-widths optimizes for power and memory requirements
- Adders and bit-shifts lead to considerably reduced area requirements compared to MACs



$x \cdot w$

$\boxed{a^{\log_a(x)+\log_a(w)}}$

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

LOG-QUANT WITH ARBITRARY LOG-BASES INCORPORATES INTRINSIC PRUNING EFFECT [8]

$$a \in \left\{ 2^{2^{-\hat{a}}} \middle| \hat{a} \in \mathbb{N}_0 \right\}$$

$$x \cdot w =$$

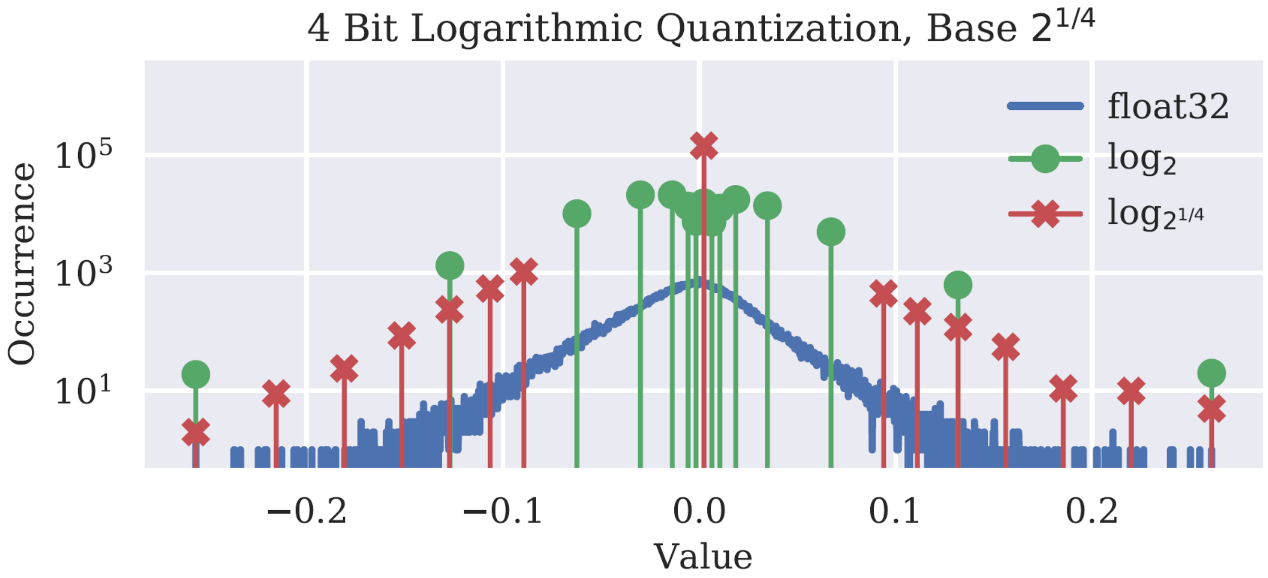
$$= a^{\log_a(x)+\log_a(w)}$$

$$= 2^{\log_2(a)\cdot(\log_a(x)+\log_a(w))}$$

$$= 2^{(\log_a(x)+\log_a(w))\gg\hat{a}}$$

$$= 2^{\text{Fractional}\left((\log_a(x)+\log_a(w))\gg\hat{a}\right)} \cdot 2^{\text{Integer}\left((\log_a(x)+\log_a(w))\gg\hat{a}\right)}$$

$$= \underbrace{2^{\text{Fractional}\left((\log_a(x)+\log_a(w))\gg\hat{a}\right)}}_{\text{LUT w/ } 2^{\hat{a}} \text{ entries}} \ll \text{Integer}\left((\log_a(x)+\log_a(w)) \gg \hat{a}\right)$$



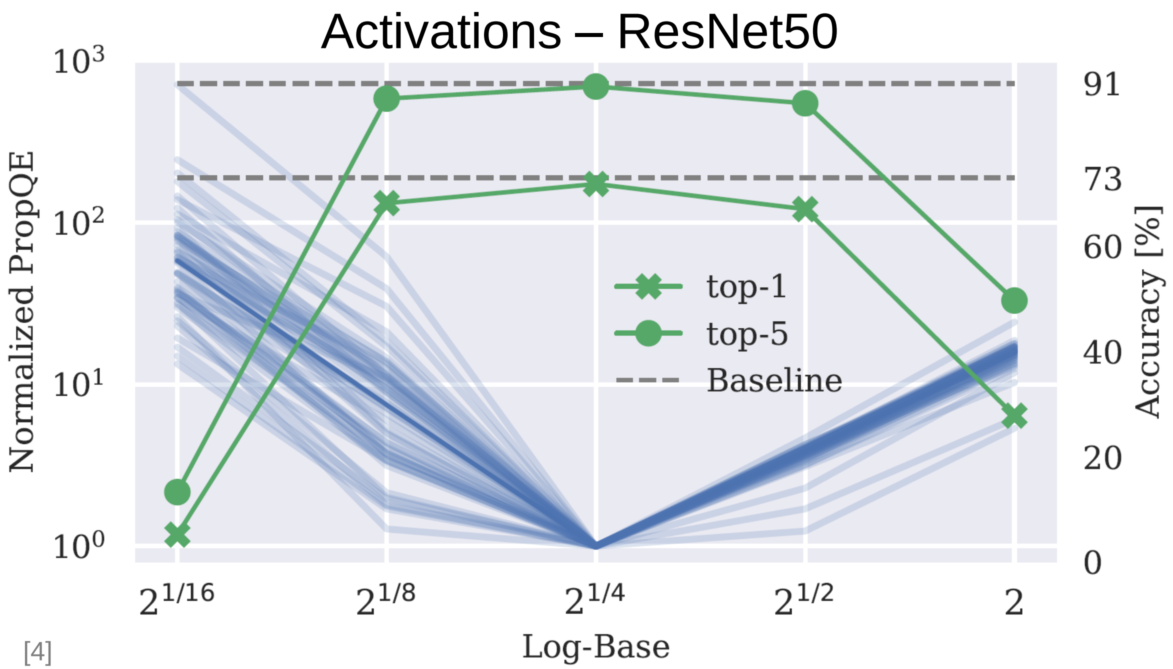
- Logarithmic quantization incorporates an intrinsic pruning effect when choosing base $a < 2$ [8]

[8] Vogel et al., Efficient hardware acceleration of CNNs using logarithmic data representation with arbitrary log-base, ICCAD 2018

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

THE SAME OPTIMAL LOG-BASE IS FOUND FOR ALL LAYERS, MAKING A HW-IMPLEMENTATION LESS COMPLEX

- Optimal log-bases are determined by minimizing the propagated quantization error (propQE)
- Different optimal log-bases are found for weights and activations
- For ResNet50, the same optimal log-base is found in every layer
  → No HW-flexibility required for changing the log-base



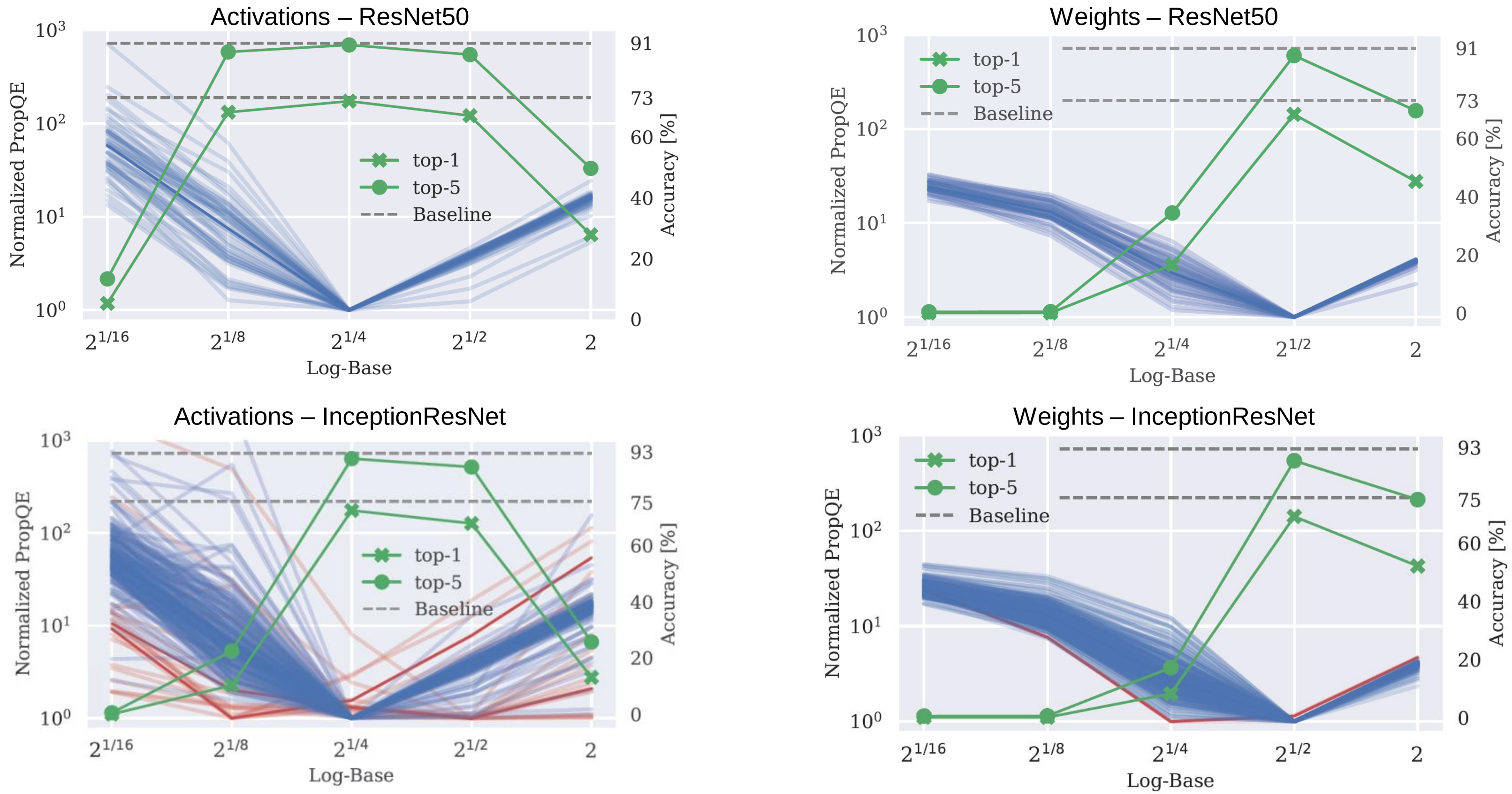
[4]

[4]

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

LOGARITHMIC QUANTIZATION OF CNNS WITH ARBITRARY LOG-BASE

- In ResNet50, the same optimal log-base is found in every layer
- In InceptionResNet, there are exceptions to this behavior, yet choosing a single optimal log-base for all layers achieves still considerably good results



[4]

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

LOG-BASED QUANTIZATION ACHIEVES COMPETITIVE RESULTS COMPARED TO LINEAR QUANT. ON SEVERAL DNN ARCHITECTURES

- Logarithmic quantization of weights* and activations at 5 bit

Quantization	Bit-Width	Classification						Semantic Segmentation			
		VGG16 top-1+	VGG16 top-5++	ResNet50 top-1	ResNet50 top-5	InceptionNet top-1	InceptionNet top-5	Dilated Model mIoU§	Dilated Model pix.acc.#	FCN8s mIoU	FCN8s pix.acc.
Calibration samples	–	100						36		10	
lin-quant baseline	8	69.12	89.06	71.67	90.73	73.71	91.57	55.62	92.78	66.47	94.44
$w: \log_{2^{1/2}}$ $y: \log_{2^{1/4}}$	5	68.46	88.36	**66.89**	**87.08**	**64.65**	**85.55**	54.83	92.65	66.05	94.39
log vs linear	–	-0.66	-0.70	**-4.78**	**-3.65**	**-9.06**	**-6.02**	-0.79	-0.13	-0.42	-0.05

Linear 8bit

[4]

Logarithmic 5bit

[4]

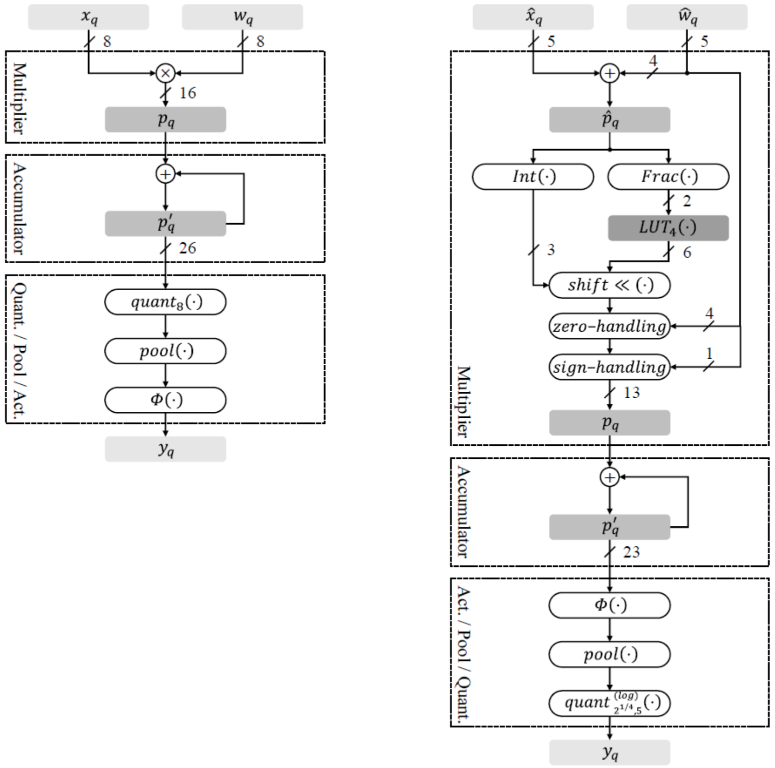
+ Top-1 accuracy: % of correctly classified labels
++ Top-5 accuracy: % of correct label within first 5 predicted labels
§ mIoU: mean intersection over union
# pix.acc.: mean overall pixel accuracy

* per-tensor quantization and biases @8bit (linear) per-tensor

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

LOG-BASED MAC-ELEMENTS ARE COMPLEX BUT HAVE REDUCED INTERFACE BIT-WIDTHS

- Log-based number representations allow reducing the external bit-widths and therefore, optimize external bus and memory requirements
- Nevertheless, an implementation of a log-based MAC*-element consists of more stages than its linear implementation



[4]

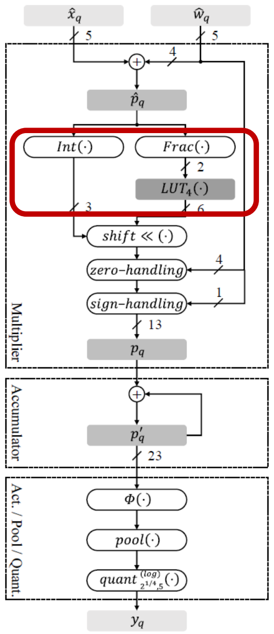
* MAC – multiply-accumulate

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

ARE THERE WAYS TO ADDRESS THE DISCUSSED DOWNSIDES OF THIS LOG-BASED NUMBER REPRESENTATION?

- In the following, an alternate approach is presented addressing the drawbacks of log-based quantization with arbitrary log-base
  - Complex MAC-element implementation
  - Reduced accuracy on complex DNN architectures

Quantization	Bit-Width	VGG16 top-1+	VGG16 top-5++	ResNet50 top-1	ResNet50 top-5	InceptionNet top-1	InceptionNet top-5	Dilated Model mIoU§	Dilated Model pix.acc.#	FCN8s mIoU	FCN8s pix.acc.
Calibration samples	–	100						36		10	
lin-quant baseline	8	69.12	89.06	71.67	90.73	73.71	91.57	55.62	92.78	66.47	94.44
$w\colon \log_{2^{1/2}}$ $y\colon \log_{2^{1/4}}$	5	68.46	88.36	**66.89**	**87.08**	**64.65**	**85.55**	54.83	92.65	66.05	94.39
log vs linear	–	-0.66	-0.70	**-4.78**	**-3.65**	**-9.06**	**-6.02**	-0.79	-0.13	-0.42	-0.05



[4]

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

Quantization of DNNs w/o Multipliers

Self-supervised quantization

Logarithmic number representation

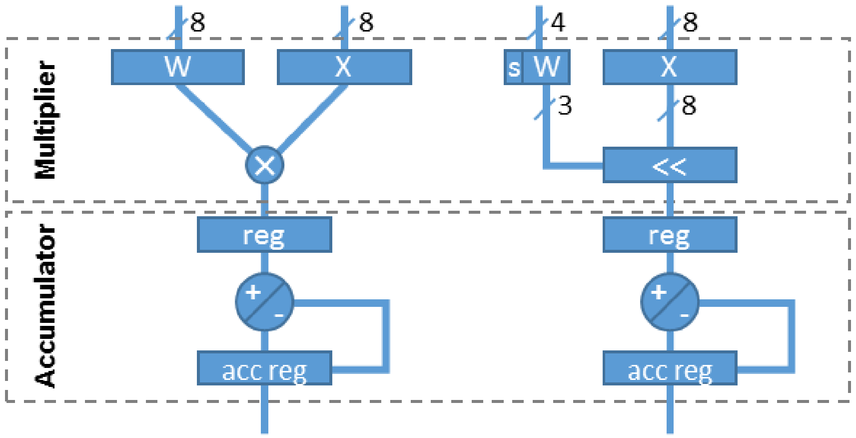
Bit-shift-based quantization

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

LOG-BASED MIXED-PRECISION QUANTIZATION ADDRESSES SIMPLER IMPLEMENTATION AND HIGHER ACCURACY ON COMPLEX DNN ARCHITECTURES

- CNN accelerators incorporate a considerable amount of multiply-accumulate engines
- Fixed-point multipliers are considerably larger (wrt. silicon area) than shift-operations
- Shift-based operation
  → logarithmically quantized weights (4bit)
- Note:
  This approach uses linearly quantized activations and therefore, integrates standard input signals more easily

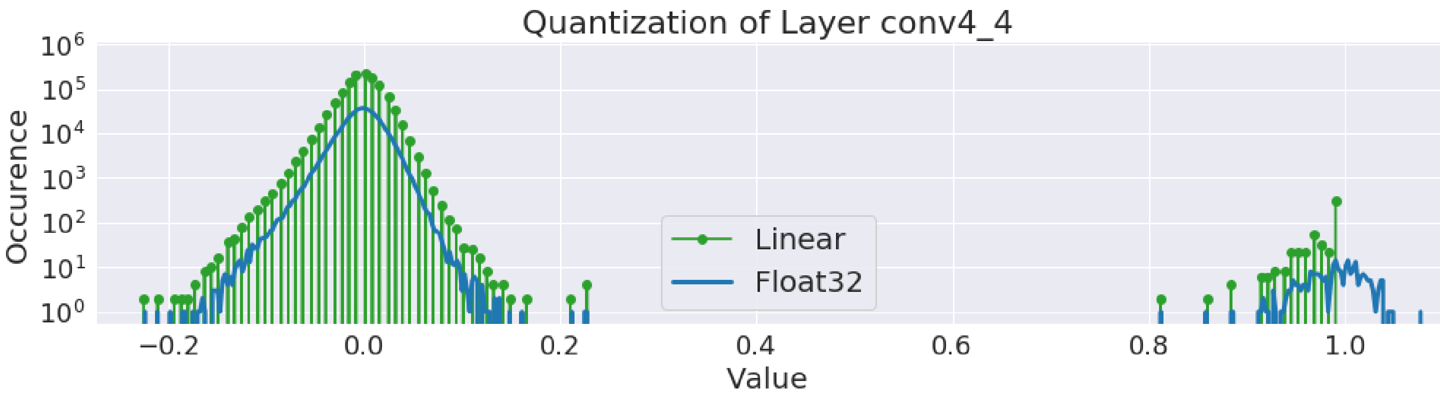


$x \cdot w$ $\quad$ $x \ll \log_2(w)$

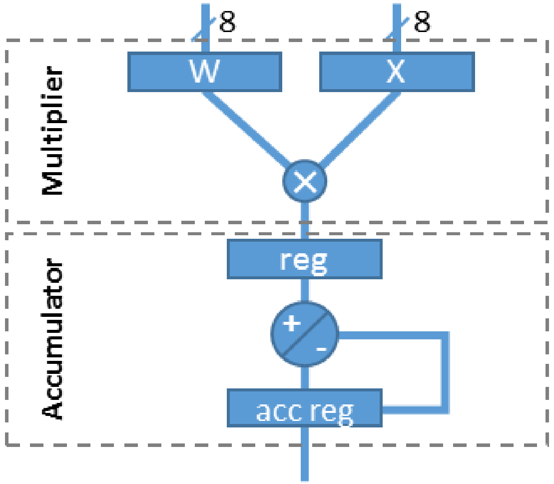
LOG-BASED MIXED-PRECISION QUANTIZATION ADDRESSES SIMPLER IMPLEMENTATION AND HIGHER ACCURACY ON COMPLEX DNN ARCHITECTURES

$w \in \mathbb{Z}$

$x \cdot w$



[4]



- Quantization of weights (with bimodal distribution)
  - linear

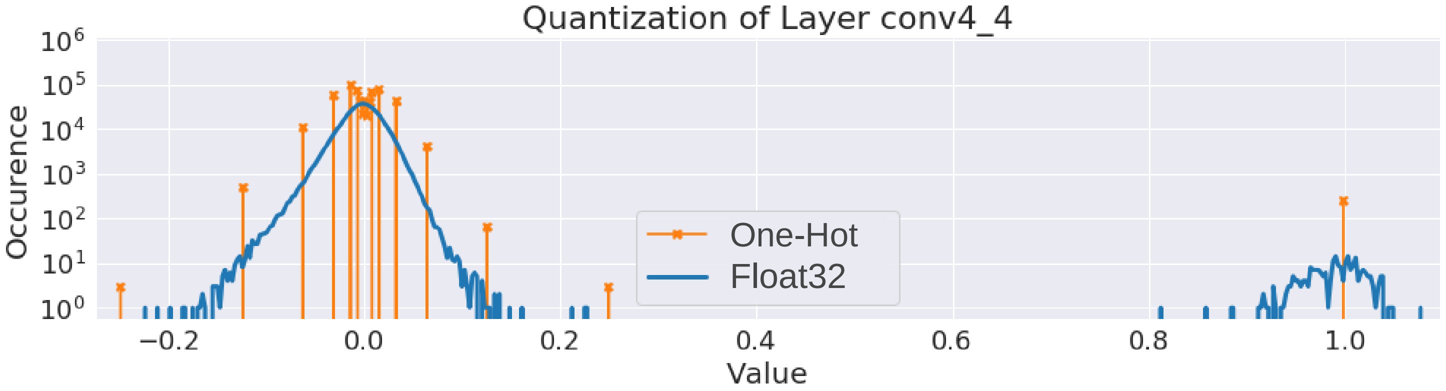
[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

LOG-BASED MIXED-PRECISION QUANTIZATION ADDRESSES SIMPLER IMPLEMENTATION AND HIGHER ACCURACY ON COMPLEX DNN ARCHITECTURES

$w \in \{2^z | z \in \mathbb{N}_0\}$

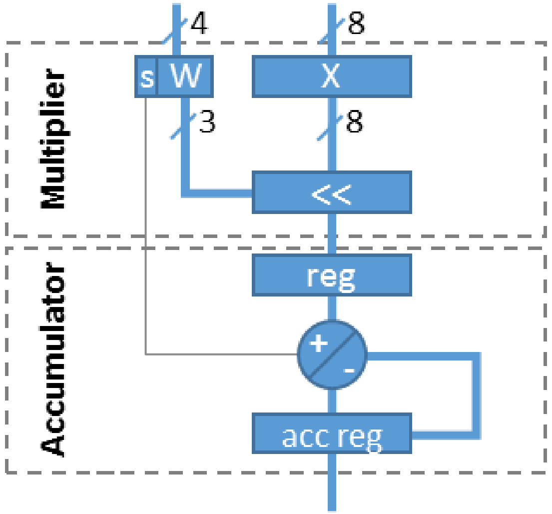
$x \cdot w$

$x \ll \log_2(w)$



[4]

- Quantization of weights (with bimodal distribution)
  - linear
  - “one-hot”



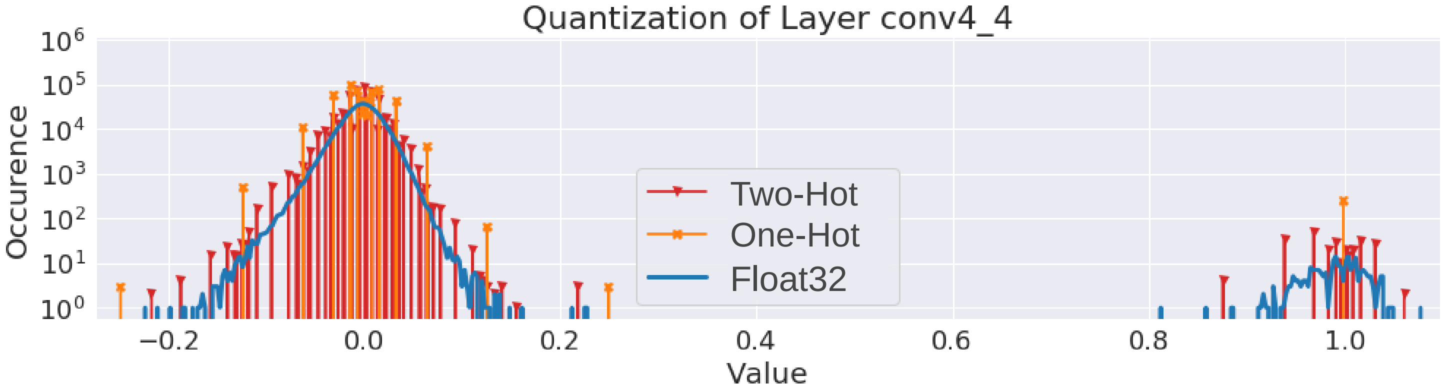
[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

LOG-BASED MIXED-PRECISION QUANTIZATION ADDRESSES SIMPLER IMPLEMENTATION AND HIGHER ACCURACY ON COMPLEX DNN ARCHITECTURES

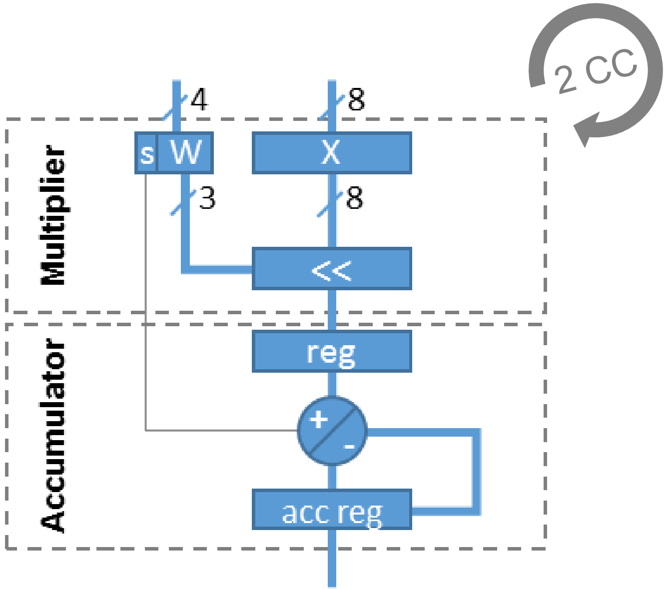
$$w_{1,2} \in \{2^z | z \in \mathbb{N}_0\}$$

$$x \cdot (w_1 + w_2)$$

$$x \ll \log_2(w_1) + x \ll \log_2(w_2)$$



[4]



- Quantization of weights (with bimodal distribution)
  - linear
  - “one-hot”
  - “two-hot”

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

LOG-BASED QUANTIZATION ACHIEVES COMPETITIVE RESULTS COMPARED TO LINEAR QUANT. EVEN ON COMPLEX DNN ARCHITECTURES

- Log-based quantization (per-tensor) of weights, biases*, and activations*

Quantization	VGG16 top-1+	VGG16 top-5++	ResNet50 top-1	ResNet50 top-5	InceptionNet top-1	InceptionNet top-5	Dilated Model mIoU§	Dilated Model pix.acc.#	FCN8s mIoU	FCN8s pix.acc.
	Classification						Semantic Segmentation			
Calibration samples	100						36		10	
lin-quant baseline	69.12	89.06	71.67	90.73	73.71	91.57	55.62	92.78	66.47	94.44
w_q **one-hot,** 4 bit	63.85	86.76	**46.36**	**72.11**	**37.77**	**64.55**	**49.52**	90.13	60.75	92.10
w_q **two-hot,** 8 bit	68.91	89.54	**70.84**	**90.35**	**72.47**	**91.11**	55.34	92.74	66.24	94.41
two-hot vs linear	-0.21	**+0.48**	-0.83	-0.38	**-1.24**	**-0.46**	-0.28	-0.04	-0.23	-0.03

Float32 Linear 8bit Two-hot 8bit Linear vs. two-hot 8bit

[4]

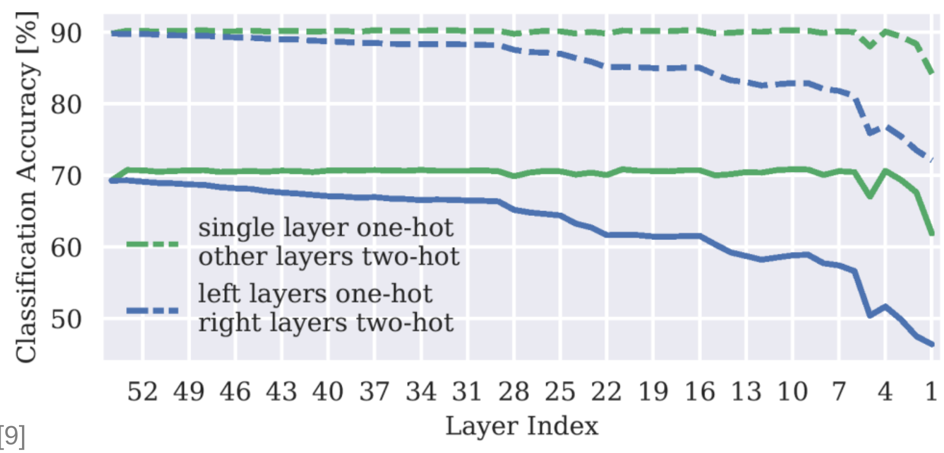
+ Top-1 accuracy: % of correctly classified labels
++ Top-5 accuracy: % of correct label within first 5 predicted labels
§ mIoU: mean intersection over union
# pix.acc.: mean overall pixel accuracy

* activations, biases @8bit (linear)

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

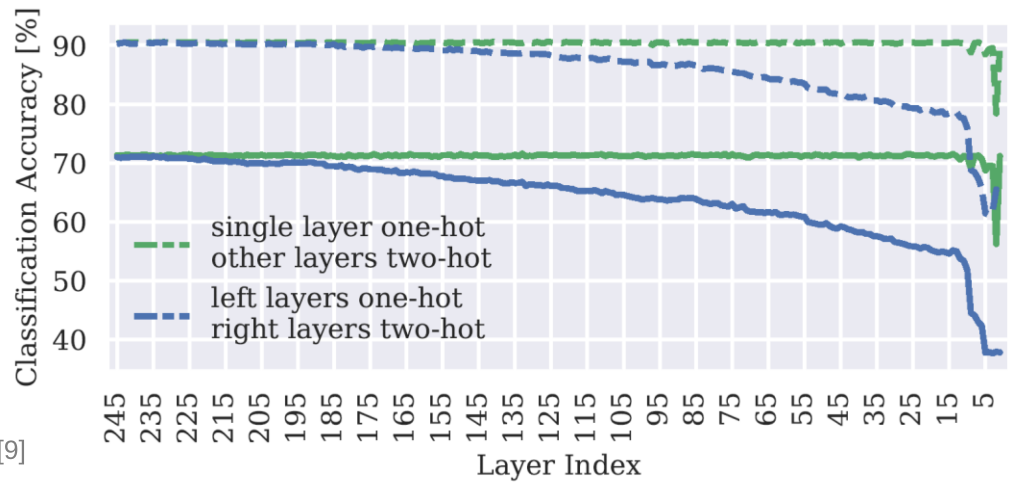
MIXED-PRECISION LOG-BASED QUANTIZATION ALLOWS TO TRADE ACCURACY WITH THROUGHPUT AND NETWORK SIZE[9]

ResNet50



[9]

InceptionNet



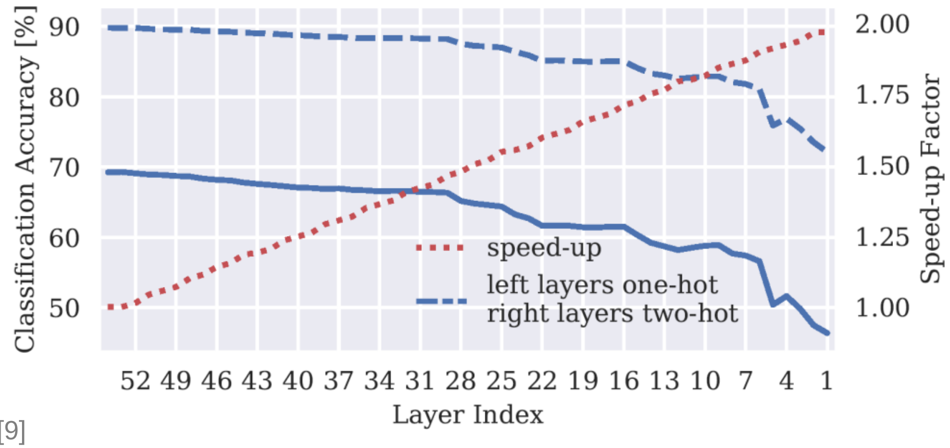
[9]

- Layers close to the network input are sensitive to one-hot quantization

[9] Vogel et al., Bit-Shift-Based Accelerator for CNNs with Selectable Accuracy and Throughput, DSD 2019

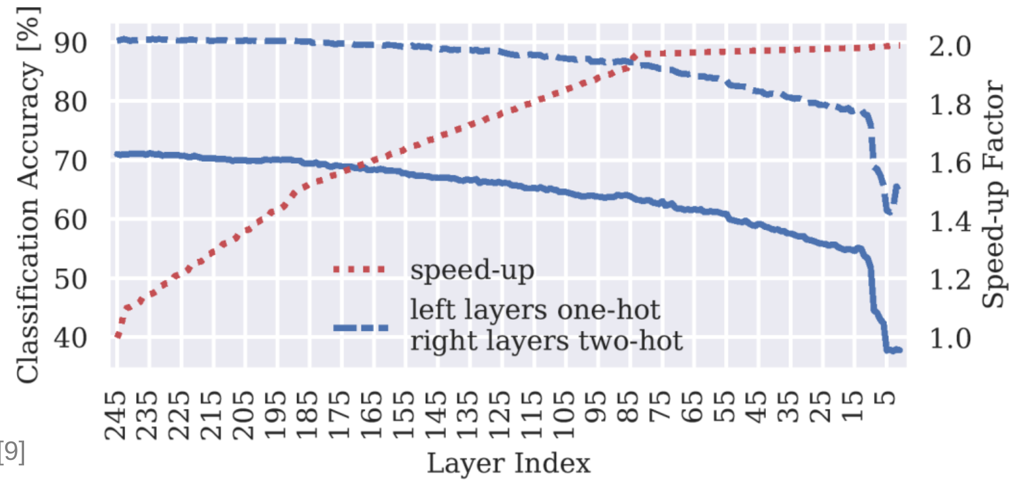
MIXED-PRECISION LOG-BASED QUANTIZATION ALLOWS TO TRADE ACCURACY WITH THROUGHPUT AND NETWORK SIZE[9]

ResNet50



[9]

InceptionNet



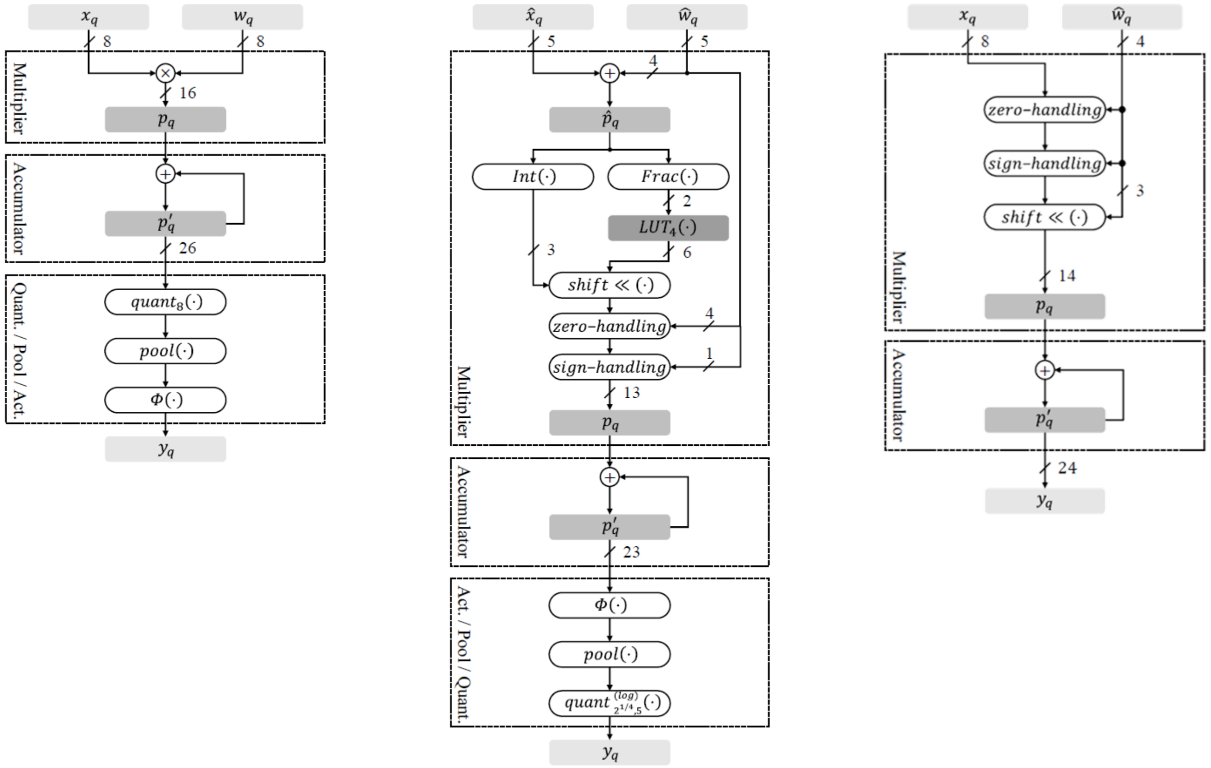
[9]

- Layers close to the network input are sensitive to one-hot quantization
- Layerwise selection allows to trade accuracy with throughput and resulting network size
- The configuration can be selected at run-time

[9] Vogel et al., Bit-Shift-Based Accelerator for CNNs with Selectable Accuracy and Throughput, DSD 2019

BIT-SHIFT-BASED MAC-ELEMENTS WITH LINEAR QUANTIZATION FOR ACTIVATIONS OFFER FLEXIBLE MIXED-PRECISION COMPUTATION

- Implementations of bit-shift-based MAC*-elements with “one-hot”/”two-hot” weights are less complex than log-based MAC-elements with arbitrary log-base
- Mixed-precision capability built in without the need for upper/lower nibble** handling



[4]

* MAC – multiply-accumulate

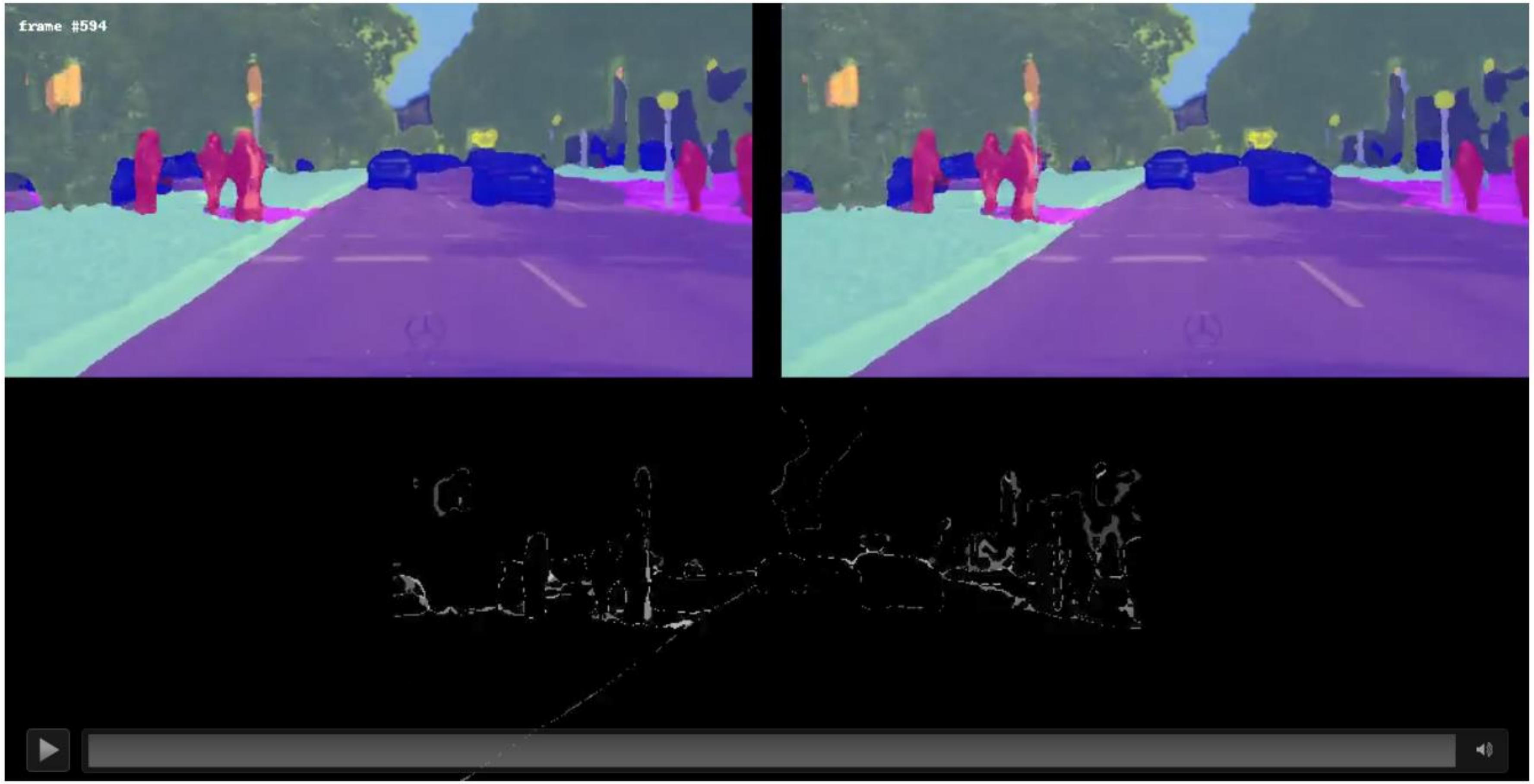
** nibble – 4 bit

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

QUALITATIVE EVALUATION ON SEMANTIC SEGMENTATION

- Qualitative output of the dilated model for semantic segmentation on cityscapes
- Linear 8bit quantization (left), two-hot 8bit quantization (right), mutual diff. (bottom)

[4]



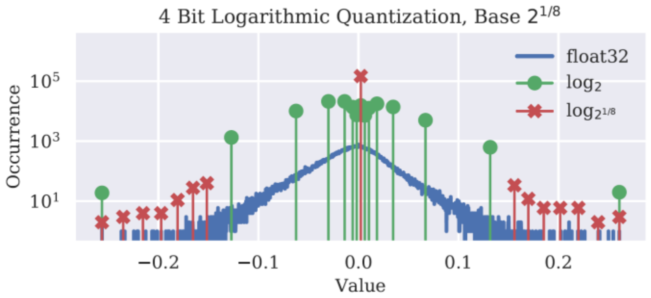
[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Defense, Jan. 10, 2020

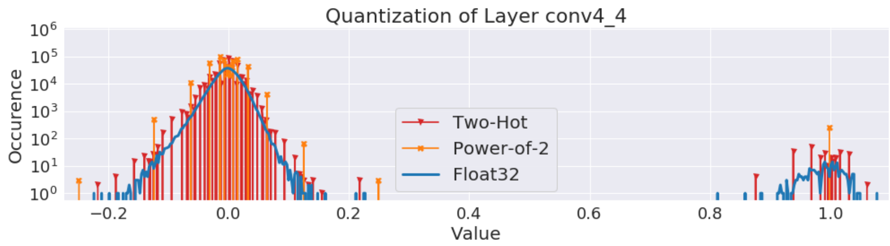
METHODS FOR QUANTIZING PRE-TRAINED NEURAL NETWORKS HAVE BEEN PRESENTED AND EVALUATED ON TWO APPROACHES FOR MULTIPLIERLESS EXECUTION OF DNNS

- We discussed a method for quantizing pre-trained neural networks without the need for fine-tuning on labeled training data
  - Minimizing the propagated quantization error

$$y^{(l+1)} + \underbrace{y_{p\Delta}^{(l)}}_{\text{propQE}} \qquad \alpha = argmin\left(y_{p\Delta}^{(l)}\right)^2$$

- Two approaches for few-bit quantization and multiplierless processing were discussed
  - Logarithmic number representation with arbitrary log-base
  - Mixed-precision log-based quantization (“one-hot”/”two-hot”)





References

[1] Wang, et al. HAQ: Hardware-Aware Automated Quantization with Mixed Precision, CVPR2019

[4] Vogel, Design and implementation of number representations for efficient multiplierless acceleration of convolutional neural networks, PhD Thesis 2020

[7] Vogel et al., Self-Supervised Quantization of Pre-Trained Neural Networks for Multiplierless Acceleration, DATE 2019

[8] Vogel et al., Efficient hardware acceleration of CNNs using logarithmic data representation with arbitrary log-base, ICCAD 2018

[9] Vogel et al., Bit-Shift-Based Accelerator for CNNs with Selectable Accuracy and Throughput, DSD 2019

AVAILABLE STUDENT PROJECT POSITIONS (INTERNSHIP & GRADUATION PROJECTS)

- **Automatic neural network quantization and deployment optimization**
  - optimizing neural networks through quantization and pruning
  - taking multiple optimization criteria into account
  - investigating options to learn how to quantize/prune neural networks
  - automatically determining optimal SW deployment parameterizations for embedded devices

- **Hardware-aware NAS for next generation radar-based ADAS**
  - improving state of the art approaches on object classification with DNNs
  - leveraging ML and NN-design know-how from other domains for Radar signal processing
  - exploring NN designs that exploit Radar spectrum data, Radar target lists or a fusion of both
  - optimizing simultaneously the deployment properties on target hardware and the task accuracy

AVAILABLE STUDENT PROJECT POSITIONS (INTERNSHIP & GRADUATION PROJECTS)

- **Transferring existing NAS methodologies to challenging embedded system tasks**
  - audio processing (noise cancelation, keyword spotting, etc.)
  - battery management and battery health estimation
  - predictive maintenance (e.g., anomaly detection)
  - with the goal to derive insights on the trade-off between system requirements and task accuracy

- **Intelligent automated design & configuration of next generation DL-HW-accelerators**
  - automatically optimizing configurable HW accelerators and co-adapting neural architectures
  - especially focusing on quantization and sparsity features of HW-accelerators

- **Hardware-aware NAS for next generation hardware and software**
  - extending available hardware-aware NAS frameworks to new hardware targets;
  - integrating said NAS frameworks with one of our existing training modalities;
  - conducting extensive experiments in our training modalities.

Sebastian Vogel

sebastian.vogel@nxp.com

https://www.linkedin.com/in/sebastianvogel

SECURE CONNECTIONS
FOR A SMARTER WORLD