Progressive Stochastic Binarization

David Hartmann* and Michael Wand*

*Institute of Computer Science, Johannes Gutenberg-University of Mainz, Germany

Introduction

Deep networks are expensive. Bulk costs:

- Scalar products: Addition & multiplication in $\mathbb{R}$
- floating-point operations also incur substantial costs for alignment/normalization
- custom hardware has the potential for substantial further cost reductions.[1]

Method Outline:

- Integer activations
- Replace multiplications by stochastic gating, sampling adjacent powers of two
- Accumulation increases the precision as needed

Computational Attention:

- allowing fine-grained dynamic control of accuracy
- we propose a two-stage algorithm that first computes a rough estimate of accuracy demands and then uses higher precision more sparsely.
- **Example:** *ResNet50v2* model retains
  - 94.4% at 16 random samples and
  - 98.6% at 64 samples

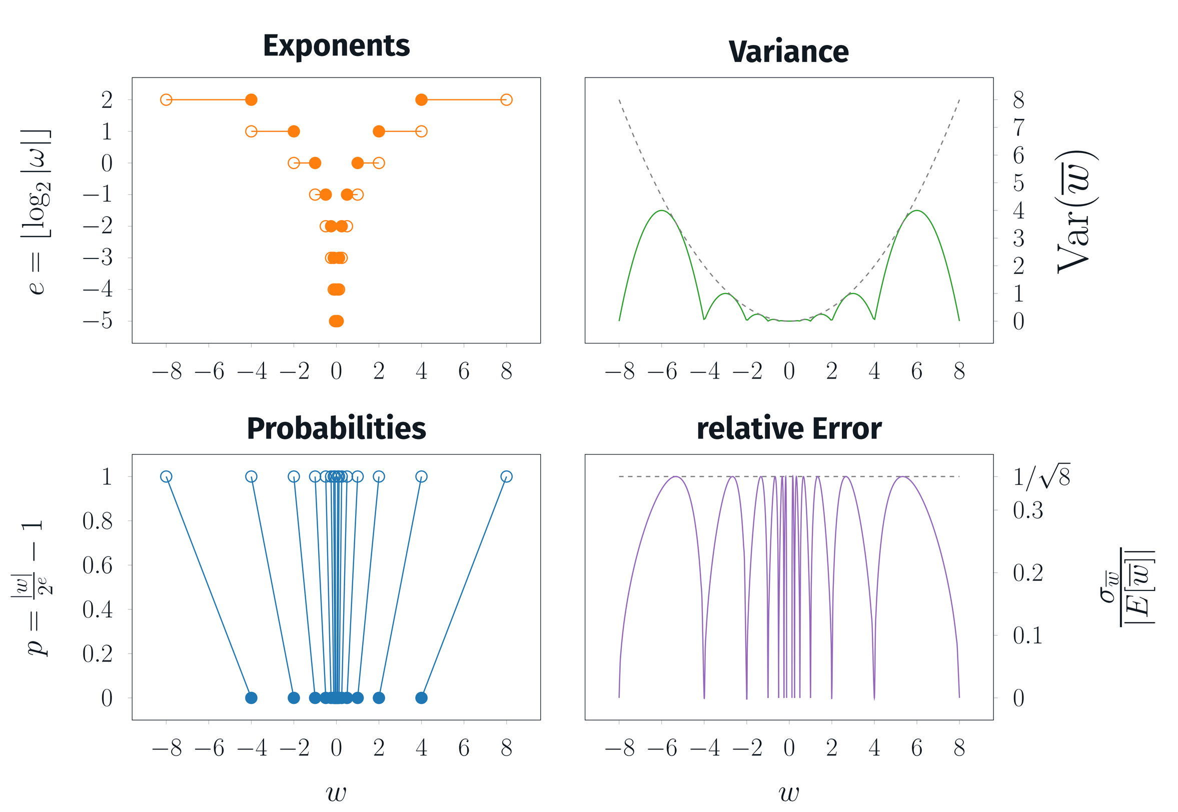
  of the full-precision accuracy.

Key Contributions

- First **quantization scheme** permitting **run-time precision control**.
- **Computational attention**: adaptive sampling reduces costs further.

Properties

- Unbiased, $\mathrm{E}[\overline{w}] = w$
- Bounded Relative Error
- Each Sample reduces Error antiproportionally



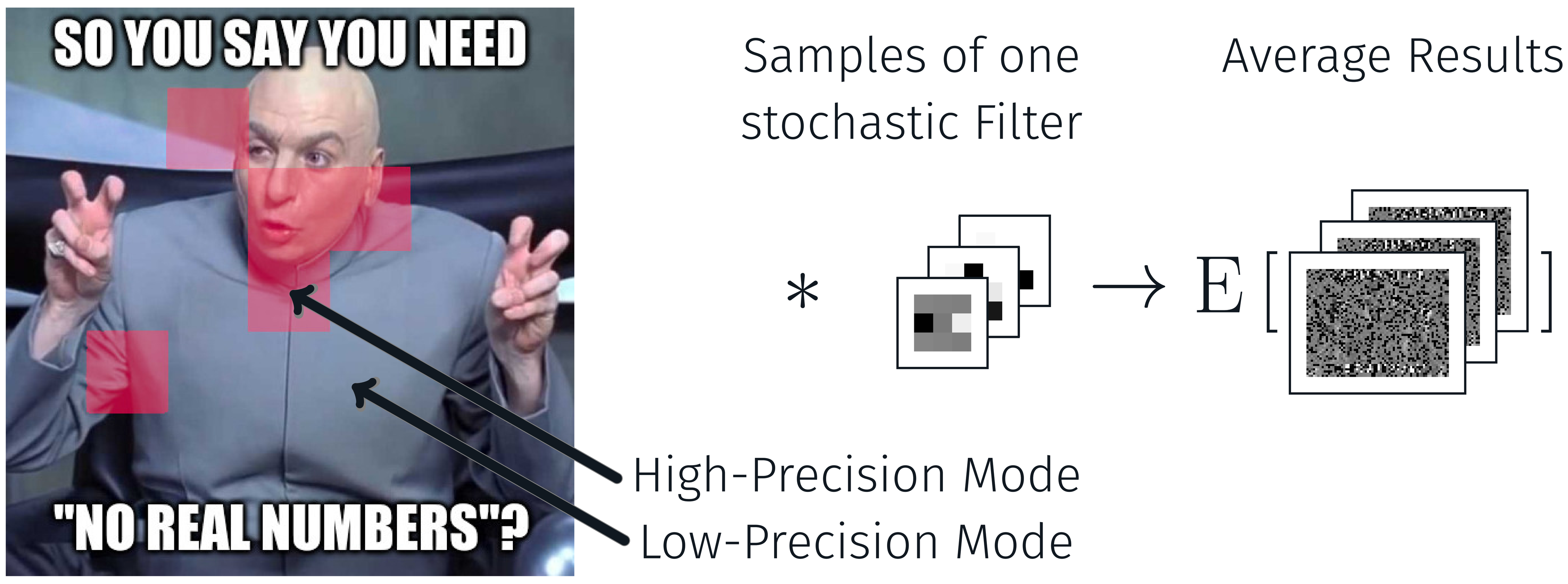
Closest Related Work

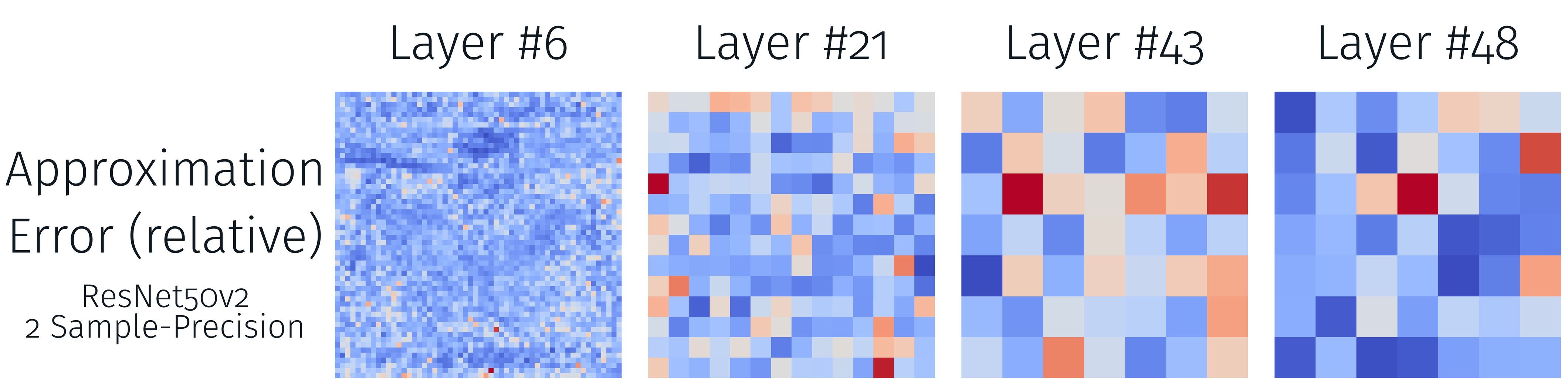
Quantization: ShiftCNN [2] transforms pretrained weights into sums-of-integer-shifts. Difference to ours: ShiftCNN is deterministic, the precision is fixed after deployment; dynamic control is not possible.

Binarization: ABC-Net uses multiple scaled binary coefficients to build a new number representation for weights [3]. Our technique changes the number representations in-place, without retraining and without hyperparameter tuning.

Alternative Network Design: Stochastic computation (SC) uses sequences of random bits whose mean is the intended number. Logarithmic quantization has also been used in SC, similar in spirit to our scheme. [4] Difference to ours: We use fixed-point numbers for intermediate results, only weights are random variables.

In a Nutshell





Method

We map filters w to **stochastic floats**:

$$w \to \overline{w} := s \cdot 2^e \cdot (B_p + 1)$$

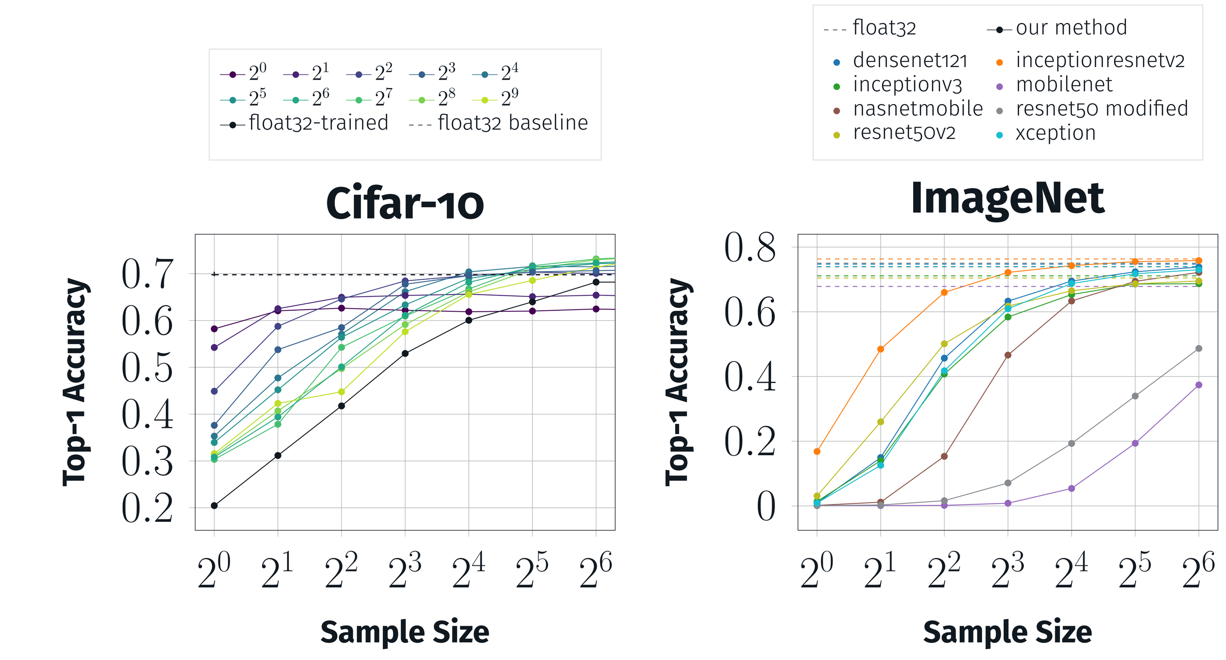
Sign $s := \mathrm{sign}(w)$,
Exponent $e := \lfloor \log_2 |w| \rfloor$,
Probability $p := \frac{|p|}{2^e} - 1$.

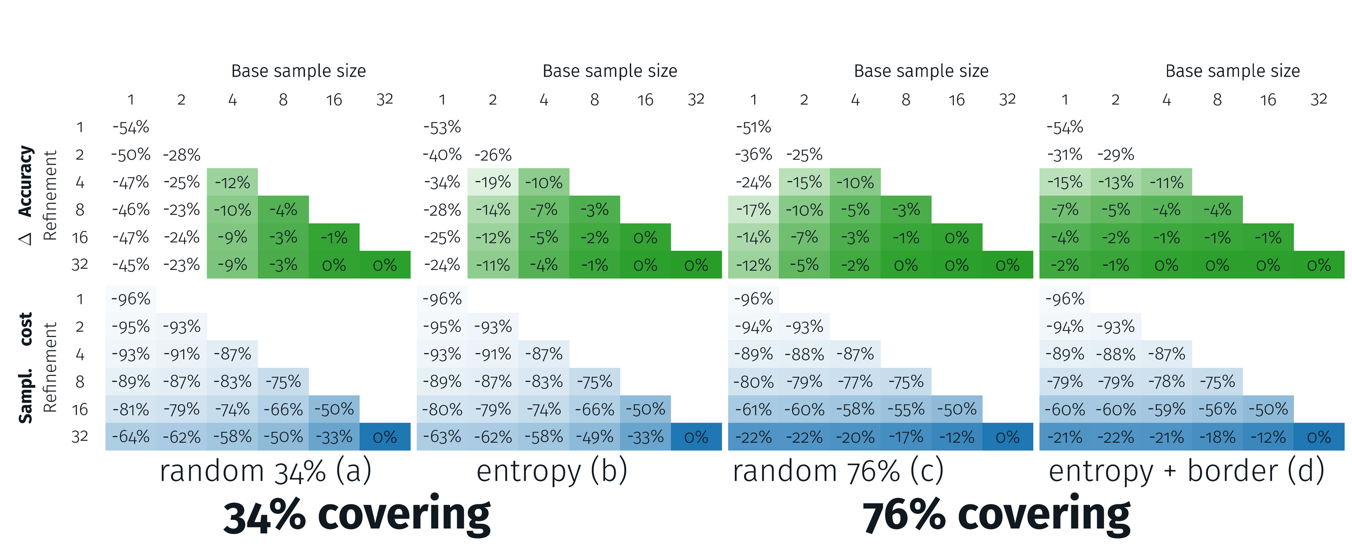
Notes:

- Multiplying with $\overline{w}$ uses only simple bit operations.
- Fold successive multiplications to avoid high-variances.
- **For dynamic control of precision:** Use multiple Bernoulli-samples,

$$w \to s \cdot 2^e \cdot \left(\frac{B_{n,p}}{n} + 1 \right).$$

Experiments on Cifar-10 & on ImageNet





Experiment		Number System	Accuracy Top-1 [%]
baseline		float32	69.7
LSQ [5]	$4, 4$-bit		70.9
DoReFa [1]	$4, 4$-bit		68.1
INQ [1]	$2, \cdot$-bit		66.6
BWN [1]	$1, \cdot$-bit		60.8
XNOR-Net [1]	$1, 1$-bit		51.2
ABC-Net [3]	$5, 5$-bit		65.0
ABC-Net [3]	$1, 1$-bit		42.0
baseline		float32	69.7
		psb5	68.2
		psb4	67.1
		psb2	54.7
+ pruning	25%	float32	69.0
		psb4	65.8
	50%	float32	41.5
		psb4	35.3
+ discrete p-values	4-bit	psb4	66.7
	2-bit	psb4	62.7
	1-bit	psb4	31.3
+ attention	random 37%	psb1/5	44.7
	entropy	**psb1/5**	**57.1**
	random 76%	psb2/5	65.7
	entropy + b	**psb2/5**	**67.7**
= combined		**psb1/5**	**57.4**
		psb2/5	**67.8**

References

[1] Vivienne Sze et al. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". In: *Proceedings of the IEEE* 105.12 (2017), pp. 2295–2329.

[2] Denis A. Gudovskiy and Luca Rigazio. "ShiftCNN: Generalized Low-Precision Architecture for Inference of Convolutional Neural Networks". In: *CoRR* abs/1706.02393 (2017). arXiv: 1706.02393. URL: http://arxiv.org/abs/1706.02393.

[3] Xiaofan Lin, Cong Zhao, and Wei Pan. "Towards Accurate Binary Convolutional Neural Network". In: *Advances in Neural Information Processing Systems 30*. Ed. by Isabelle Guyon et al. 2017, pp. 344–352. URL: http://papers.nips.cc/paper/6638-towards-accurate-binary-convolutional-neural-network.

[4] Hyeon Uk Sim and Jongeun Lee. "Log-quantized stochastic computing for memory and computation efficient DNNs". In: *Proceedings of the 24th Asia and South Pacific Design Automation Conference, ASPDAC 2019, Tokyo, Japan, January 21-24, 2019*. 2019, pp. 280–285.

[5] Steven K. Esser et al. "Learned Step Size Quantization". In: *CoRR* abs/1902.08153 (2019). arXiv: 1902.08153. URL: http://arxiv.org/abs/1902.08153.

The **5th Workshop on Energy Efficient Machine Learning and Cognitive Computing** @ NeurIPS 2019