Towards On-device Learning on the Edge: Ways to Select Neurons to Update under a Budget Constraint

WACV2024 – SCIoT

Aël Quélennec, Enzo Tartaglione, Pavlo Mozharovskyi, Van-Tam Nguyen

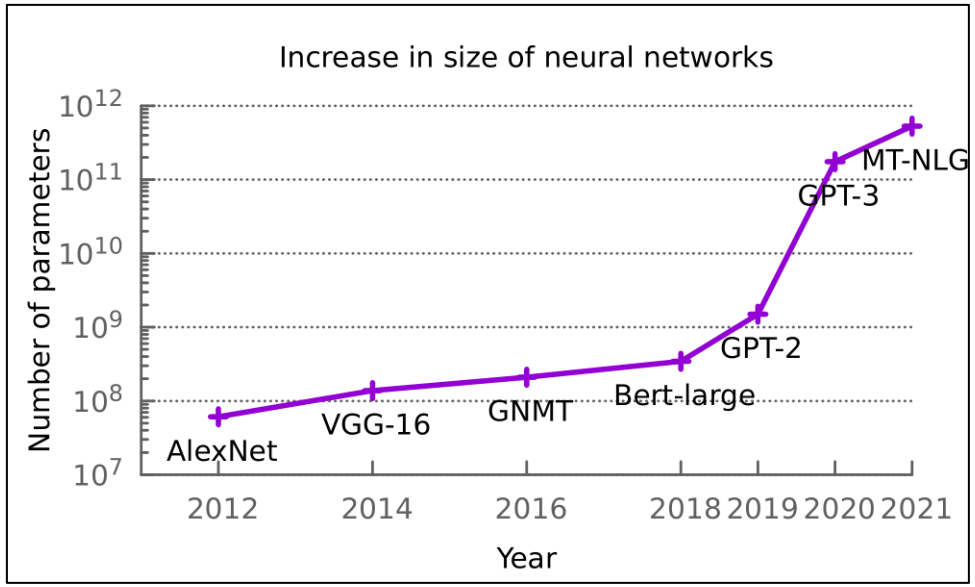
Télécom Paris – Institut Polytechnique de Paris

{name.surname}@telecom-paris.fr

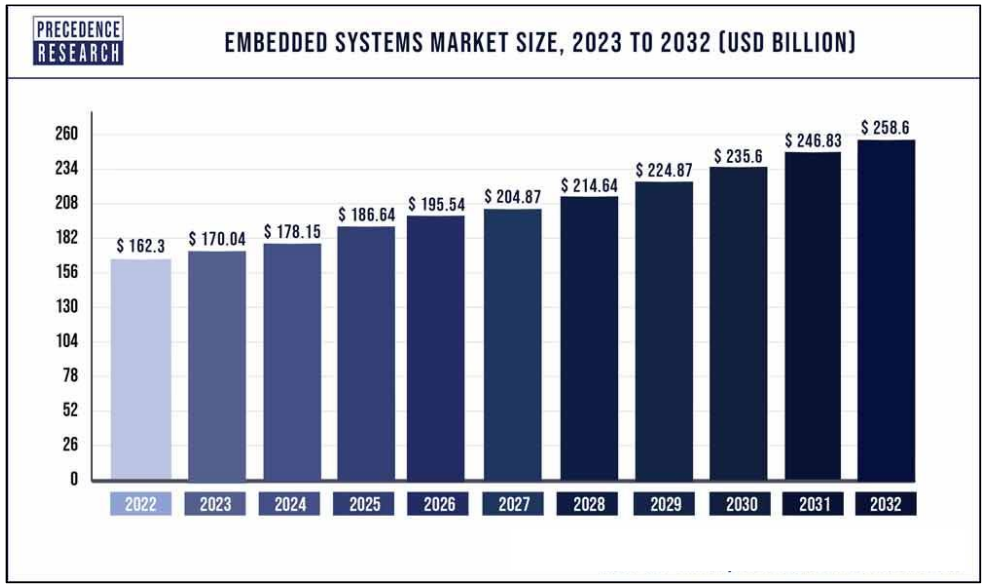
Overview

- Motivations
- Related work
- Our method
- Experiments and results
- Conclusion

Motivations – Porting AI to edge devices



Source: Parallel Software and Systems Group



Source: www.precedenceresearch.com

- Improved computational resources have expanded AI use cases, enabling larger neural networks.
- Lower microcontroller costs have increased embedded device adoption.
- Combining microcontrollers and deep neural networks is challenging due to resource constraints.
- Current research focuses on compressed and accelerated models for efficient MCU inference.
- Recent research worldwide is addressing on-device learning complexities.

Related works – In general

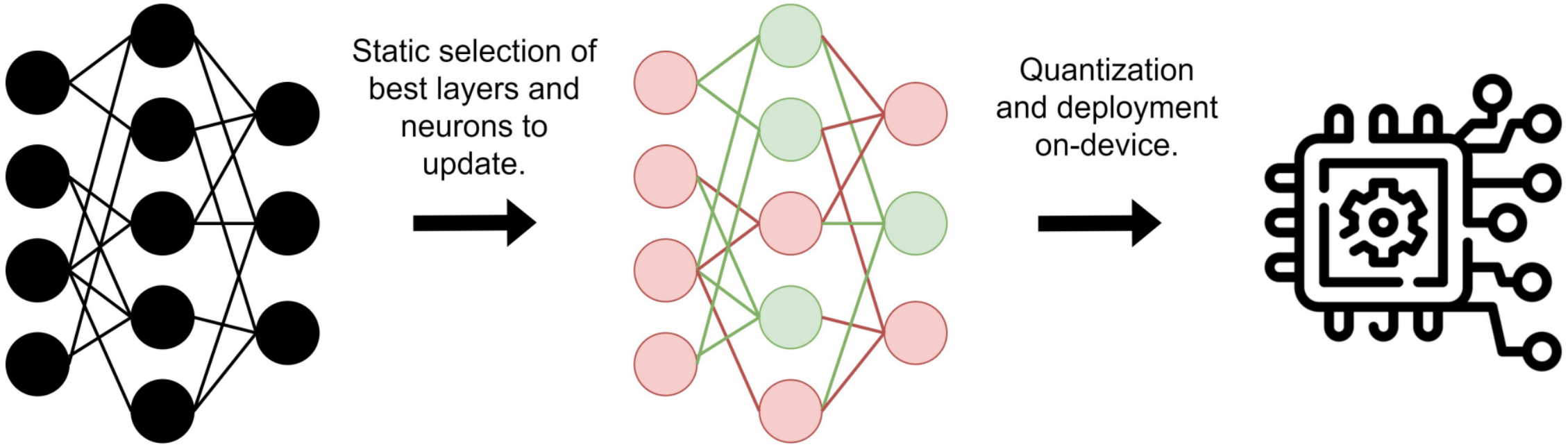
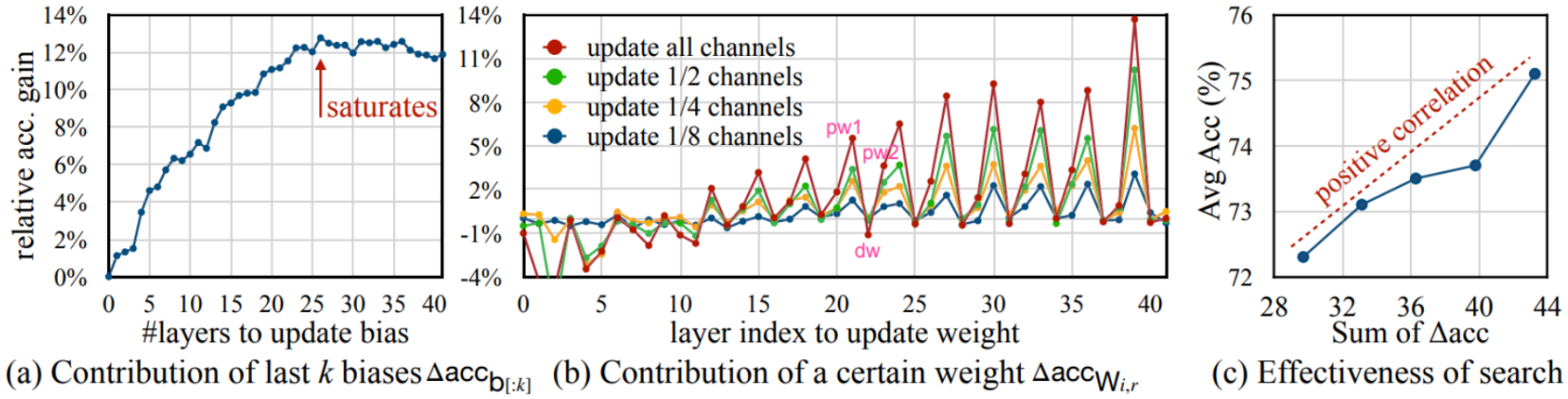


Figure 1. Standard pipeline for on-device learning.

Related works – Sparse Update (SU)



(a) Contribution of last k biases $\Delta acc_{b_{[:k]}}$ (b) Contribution of a certain weight $\Delta acc_{W_{i,r}}$ (c) Effectiveness of search

Figure 2. Contribution of bias and layer update to accuracy gain.

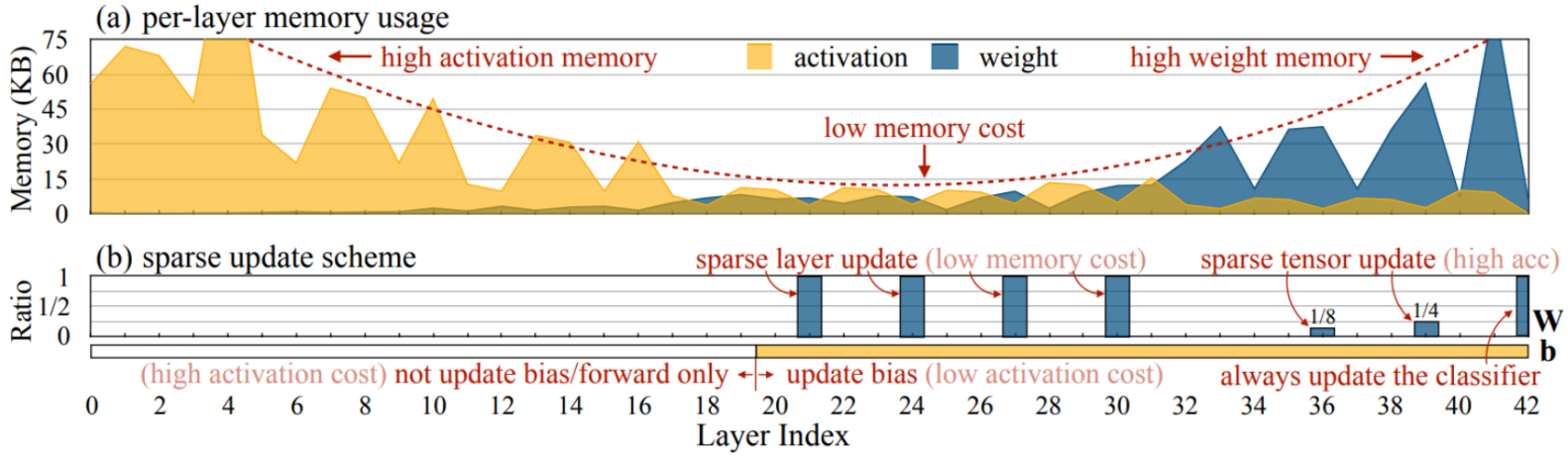


Figure 3. Selection of Sparse Update configuration based on memory costs and accuracy contributions.

Source: On-Device Training Under 256KB Memory, Lin et al.

Related works – Neurons at Equilibrium (NEq)

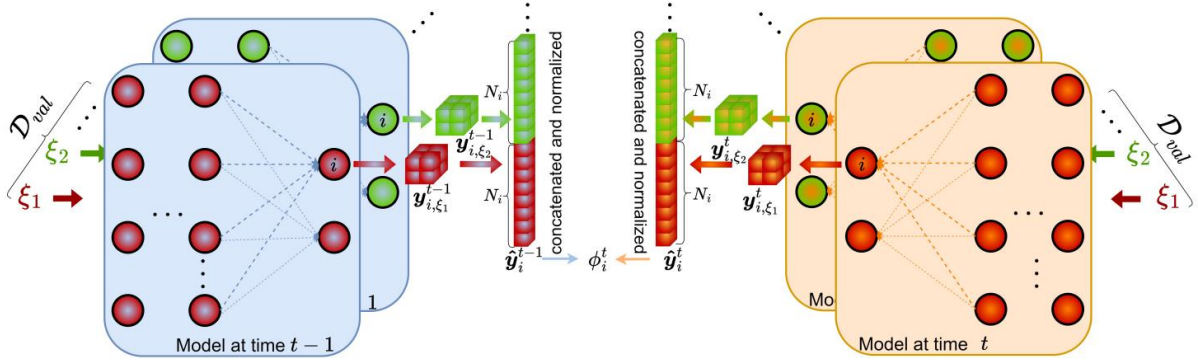


Figure 4. For a given time t the model receives samples from the validation set $\mathcal{D}_{val}$. The outputs of the i-th neuron are squeezed and concatenated to compute the similarities ϕ_i^t between epochs.

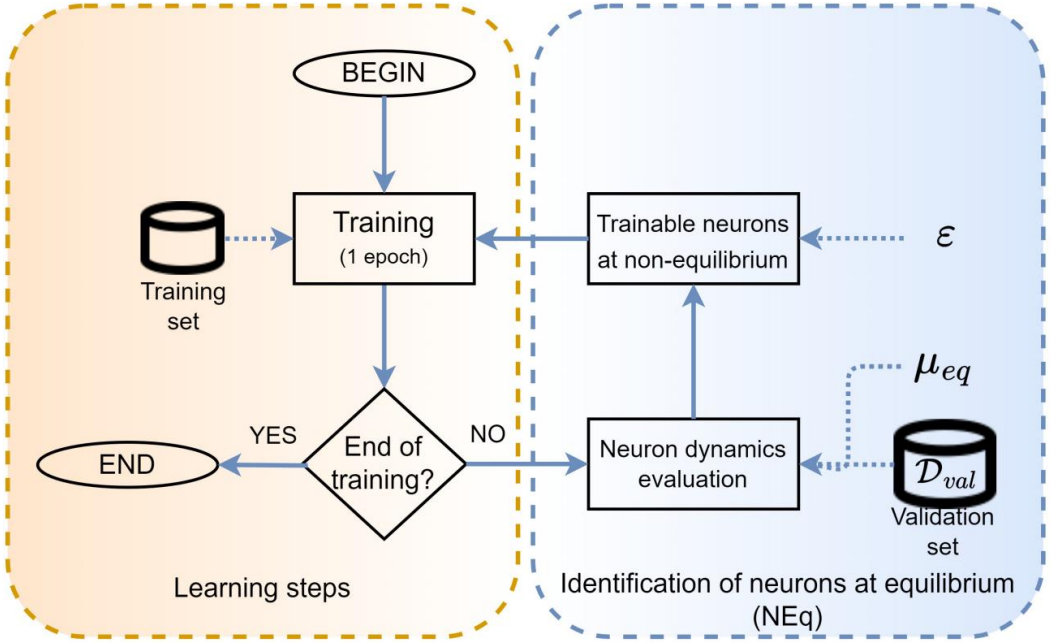


Figure 5. Overall training scheme. In orange is the standard training part and in blue is the neuron equilibrium evaluation and selection stages called NEq.

$$\phi_i^t = \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{val}}} \sum_{n=1}^{N_{i,\boldsymbol{x}}} \hat{y}_{i,\boldsymbol{x},n}^t \cdot \hat{y}_{i,\boldsymbol{x},n}^{t-1} \quad (1) \qquad \Delta\phi_i^t = \phi_i^t - \phi_i^{t-1} \quad (2) \qquad v_i^t = \Delta\phi_i^t - \mu_{eq} v_i^{t-1} \quad (3)$$

Equations 1,2 and 3: Equations to compute the velocity of the *i*-th neuron at epoch *t*.

Source: To update or not to update? Neurons at equilibrium in deep models,

Our method

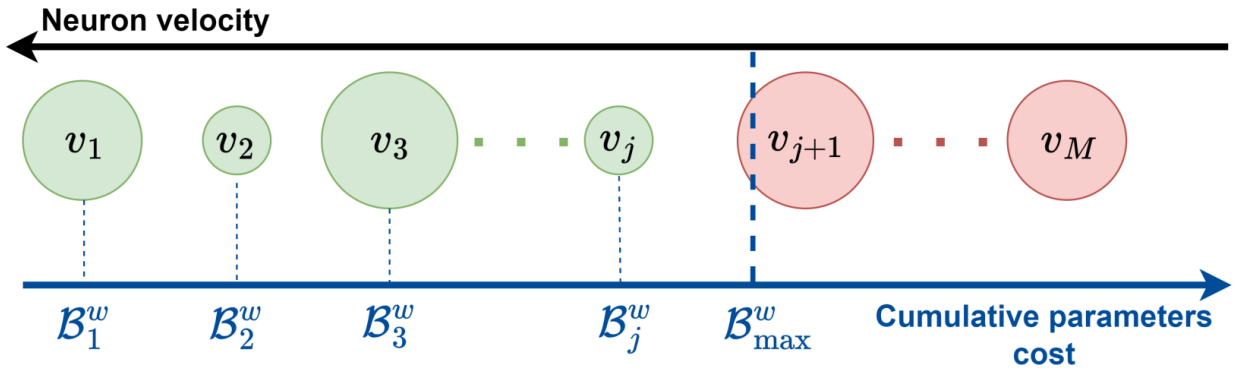


Figure 6. Selection of neurons to update given a network of M neurons and a budget $\mathcal{B}_{\max}^w$. The cost $\mathcal{C}_i^w$ is proportional to the size of the circle which represents the i-th neuron. The neurons selected for the update are in green, while those frozen in red.

- Introduction of a random approach as a baseline for dynamic neuron selection.

- We focus on the highest per-parameter average velocity by re-weighting neuron velocity as follows:

$$\tilde{v}_i = \frac{v_i}{\mathcal{C}_i^w} \quad (4)$$

Equation 4: Re-weighted velocity.

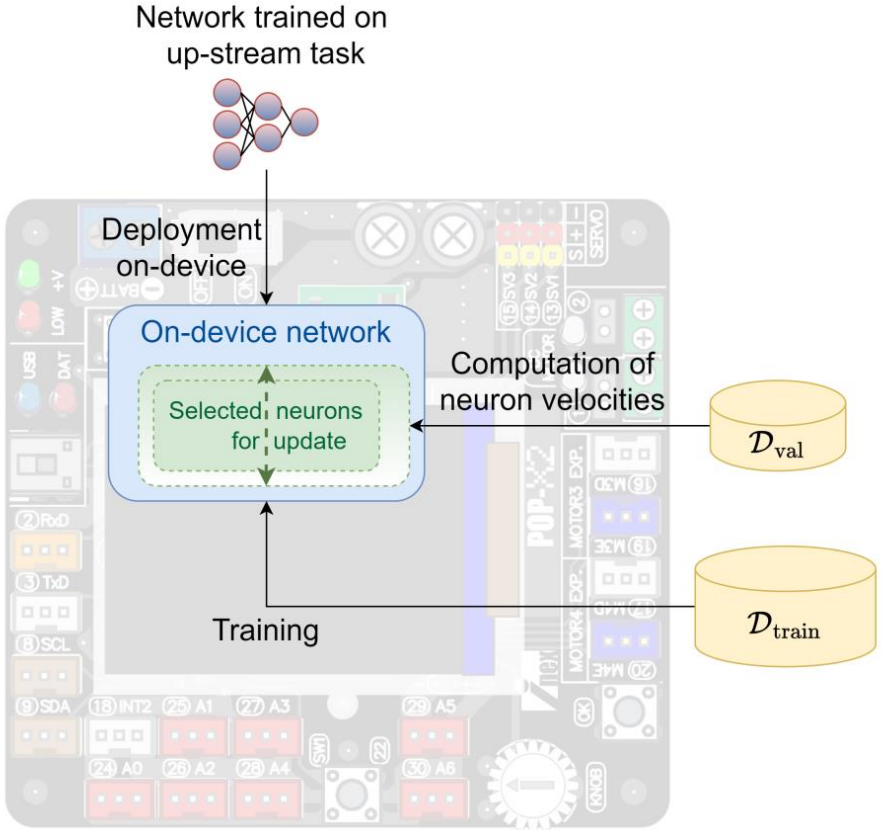


Figure 7. Overview for the on-device learning when dynamic neuron selection is applied.

Experiments and results – SU initialization

Table 1. Comparison of pretrained MobileNetV2 final top1 test accuracies across different neuron selection methods for three different memory budgets expressed in percentage of network updated and in number of parameters. For the first epoch the neurons to update are given by the associated SU scheme.

% of network updated	$\mathcal{B}^w_{\max}$	Method	Cifar 10	Cifar 100	VWW	Flowers	Food	Pets	CUB
8.8	192 311	Sparse Update	95.13±0.21	78.60±0.22	90.66±0.29	93.77±0.38	77.80±0.18	85.82±0.22	67.82±0.29
		Velocity	95.25±0.29	79.46±0.12	91.40±0.16	93.03±0.47	79.16±0.16	85.50±0.17	67.52±0.05
		Random	94.41±0.13	78.15±0.26	90.29±0.05	92.19±0.17	77.74±0.06	85.50±0.28	65.56±0.45
21.2	464 639	Sparse Update	95.30±0.10	78.84±0.20	91.29±0.18	94.28±0.36	78.35±0.17	84.63±0.15	68.04±0.28
		Velocity	95.36±0.07	79.67±0.28	91.48±0.39	93.34±0.08	79.63±0.17	84.91±0.82	68.23±0.61
		Random	94.61±0.16	78.28±0.31	90.51±0.25	92.43±0.10	78.41±0.35	84.73±0.29	66.30±0.13
30.8	675 540	Sparse Update	95.16±0.29	78.62±0.18	91.46±0.21	94.22±0.14	78.01±0.11	84.38±0.28	67.59±0.22
		Velocity	95.49±0.16	79.43±0.19	91.57±0.20	93.77±0.26	79.48±0.11	84.44±0.50	68.26±0.36
		Random	94.57±0.20	78.58±0.11	90.58±0.10	92.83±0.03	79.00±0.11	84.39±0.42	66.17±0.42

Best result obtained by: Sparse Update Velocity Random

Experiments and results – Random initialization

Table 2. Comparison of final top1 test accuracies between Baseline, SU, Random and Velocity neuron selection over various pretrained models, datasets, and budgets. For the first epoch the neurons to update are randomly selected.

Model	$\mathcal{B}^w_{\max}$	Method	Cifar 10	Cifar 100	VWW	Flowers	Food	Pets	CUB
MbV2	192 311	SU	94.88±0.12	78.15±0.13	90.75±0.17	92.70±0.06	75.10±0.40	**86.93±0.22**	66.48±0.41
		Velocity	95.35±0.35	79.41±0.21	90.95±0.16	92.98±0.29	79.18±0.07	85.56±0.47	67.92±0.27
		Random	94.46±0.16	78.03±0.21	90.20±0.13	92.11±0.15	77.57±0.16	85.16±0.07	65.96±0.06
	464 639	SU	95.00±0.08	78.69±0.19	90.80±0.24	92.86±0.33	76.50±0.23	86.64±0.27	67.81±0.23
		Velocity	95.49±0.02	79.52±0.11	91.41±0.19	93.32±0.15	79.67±0.19	84.36±0.76	68.19±0.24
		Random	94.51±0.05	78.49±0.19	90.55±0.24	92.64±0.32	78.48±0.17	84.92±0.21	66.18±0.71
	675 540	SU	95.18±0.17	79.03±0.30	91.03±0.16	93.08±0.09	77.19±0.07	86.42±0.45	67.72±0.32
		Velocity	95.57±0.11	79.21±0.37	91.72±0.15	93.33±0.31	79.68±0.16	84.37±0.28	68.19±0.24
		Random	94.57±0.09	78.41±0.29	90.35±0.01	92.88±0.21	78.90±0.06	84.39±0.41	66.36±0.20
	2 189 760	Baseline	**95.93±0.14**	**79.83±0.29**	**91.80±0.03**	**94.02±0.03**	**80.63±0.10**	82.82±0.18	**69.24±0.34**
Resnet18	980 715	Velocity	95.51±0.10	78.77±0.41	88.78±0.51	90.78±0.24	75.09±0.13	**82.82±0.30**	63.64±0.35
		Random	95.20±0.20	77.98±0.38	88.33±0.45	89.39±0.47	74.57±0.15	79.49±0.51	60.93±0.54
	2 369 480	Velocity	95.36±0.15	**79.12±0.12**	89.16±0.31	**91.02±0.17**	75.72±0.23	82.01±0.60	**63.84±0.39**
		Random	95.68±0.10	78.28±0.22	89.21±0.23	89.53±0.17	75.17±0.12	79.40±0.34	61.42±0.32
	3 444 987	Velocity	95.58±0.21	78.95±0.13	89.17±0.33	90.76±0.15	75.83±0.12	81.45±0.63	63.59±0.03
		Random	95.80±0.09	78.52±0.18	88.92±0.19	89.66±0.30	75.28±0.12	79.28±0.34	61.45±0.32
	11 166 912	Baseline	**96.2±0.13**	78.86±0.08	**89.78±0.24**	90.14±0.26	**76.32±0.08**	79.76±0.63	60.97±0.52

Best result obtained by: Sparse Update | Velocity | Random

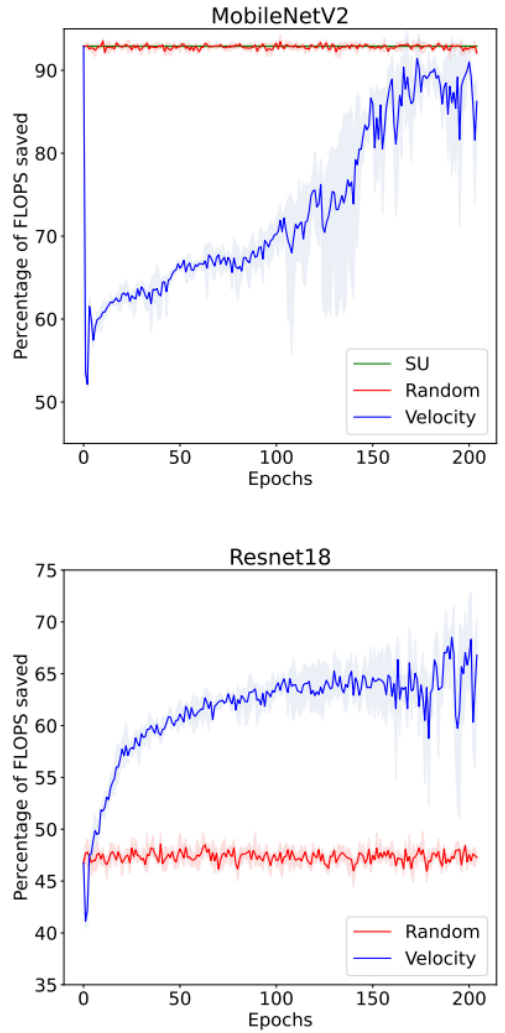


Figure 8. Percentage of FLOPS saved with SU, Velocity and Random neuron selection for each network during Cifar100 fine-tuning. We update 30.2% of the network's parameters.

Conclusions and Prospects

- Two neural update philosophies presented:
  - A static strategy identifies a sub-network prior to on-device training;
  - Two dynamic strategies, one being an evolution of a resource-unconstrained training approach.

- Results indicate dynamic neuron selection often outperforms static pre-selection in fine-tuning scenarios.

- Proposed dynamic strategy proves effective in nearly all tested scenarios.

- Areas for future progress include considering activation and training costs for selected neurons.

- Link to code: https://github.com/aelQuelennec/-WACV-2024-Ways-to-Select-Neurons-under-a-Budget-Constraint

References

1. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

2. Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems*, 33:11711–11722, 2020.

3. David Elliott, Carlos E Otero, Steven Wyatt, and Evan Martino. Tiny transformers for environmental sound classification at the edge. *arXiv preprint arXiv*:2103.12157, 2021.

4. Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce activations, not trainable parameters for efficient on-device learning. *arXiv preprint arXiv*:2007.11622, 2020.

5. Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv*:2210.11466, 2022.

6. Ji Lin, Ligeng Zhu, Wei-Ming Chen, Wei-Chen Wang, Chuang Gan, and Song Han. On-device training under 256kb memory. *Advances in Neural Information Processing Systems*, 35:22941–22954, 2022.

7. Young D Kwon, Rui Li, Stylianos I Venieris, Jagmohan Chauhan, Nicholas D Lane, and Cecilia Mascolo. Tinytrain: Deep neural network training at the extreme edge. *arXiv preprint arXiv*:2307.09988, 2023.

8. Junhuan Yang, Yi Sheng, Yuzhou Zhang, Weiwen Jiang, and Lei Yang. On-device unsupervised image segmentation. *arXiv preprint arXiv*:2303.12753, 2023.

9. Danilo Pietro Pau and Fabrizio Maria Aymone. Suitability of forward-forward and pepita learning to mlcommons tiny benchmarks. In *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6. IEEE, 2023.

10. Andrea Bragagnolo, Enzo Tartaglione, and Marco Grangetto. To update or not to update? neurons at equilibrium in deep models. *Advances in Neural Information Processing Systems*, 35:22149–22160, 2022.