tinyML® Talks

Enabling Ultra-low Power Machine Learning at the Edge

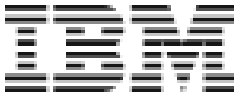
"Suitability of Forward-Forward and PEPITA Learning to MLCommons-Tiny benchmarks"

Danilo Pau – Technical Director, IEEE, AAIA & ST Fellow STMicroelectronics

September 19, 2023

www.tinyML.org

Thank you, **tinyML Strategic Partners**,
for committing to take tinyML to the next Level, together



TINY ML TALKS webcast

Executive Strategic Partners

Qualcomm
AI research

Advancing AI research to make efficient AI ubiquitous

Power efficiency

Model design, compression, quantization, algorithms, efficient hardware, software tool

Personalization

Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning

Robust learning through minimal data, unsupervised learning, on-device learning

A platform to scale AI across the industry

Perception

Object detection, speech recognition, contextual fusion

Reasoning

Scene understanding, language understanding, behavior prediction

Action

Reinforcement learning for decision making

Edge cloud

Cloud

IoT/IIoT

Automotive

Mobile

Accelerate Your Edge Compute

SYNTIANT

Making Edge AI A Reality

www.syntiant.com

TINY ML TALKS webcast

Platinum Strategic Partners

AITRIOS

DEPLOY VISION AI AT THE EDGE AT SCALE

SONY

TINY ML TALKS webcast

Gold Strategic Partners

ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

Where what if becomes what is.

Witness potential made possible at analog.com.


Build the
Future of tinyML
on arm

TINY ML TALKS webcast

EDGE IMPULSE

The Leading Development Platform for Edge ML

edgeimpulse.com

Driving decarbonization and digitalization. Together.

Infineon serving all target markets as
Leader in Power Systems and IoT

www.infineon.com

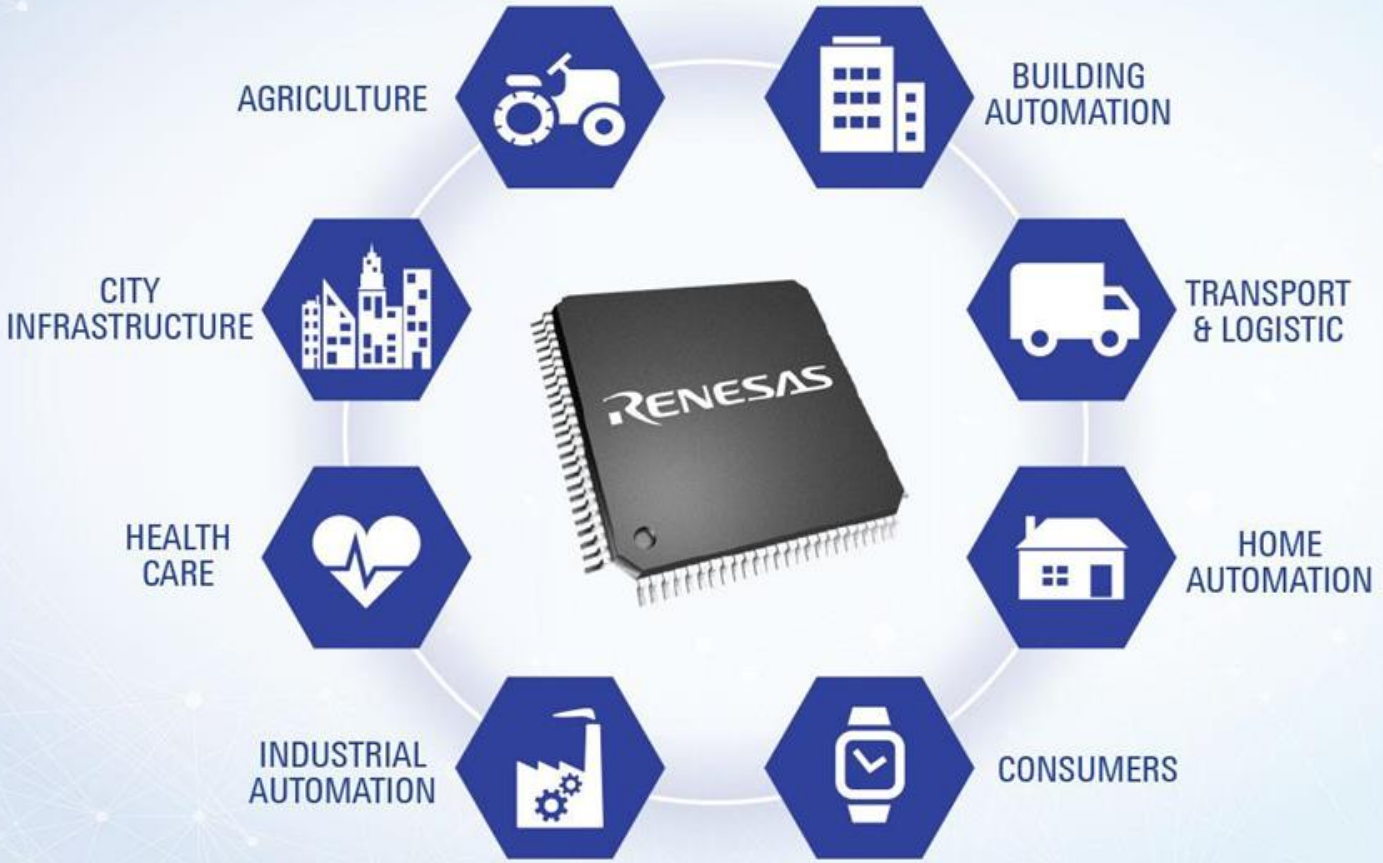
innatera

NEUROMORPHIC INTELLIGENCE FOR THE SENSOR-EDGE

www.innatera.com


Microsoft

Renesas is enabling the next generation of AI-powered solutions that will revolutionize every industry sector.



renesas.com

STMicroelectronics provides extensive solutions to make tiny Machine Learning easy

life.augmented

www.st.com/ai

synaptics®

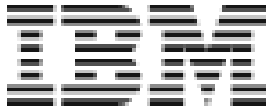
ENGINEERING EXCEPTIONAL EXPERIENCES

We engineer exceptional experiences for consumers in the home, at work, in the car, or on the go.

www.synaptics.com

TINY ML TALKS webcast

Silver Strategic Partners



Join Growing tinyML Communities:

16.8k members in
49 Groups in 41 Countries

tinyML - Enabling ultra-low Power ML at the Edge

https://www.meetup.com/tinyML-Enabling-ultra-low-Power-ML-at-the-Edge/

4k members
&
13k followers

The tinyML Community

https://www.linkedin.com/groups/13694488/

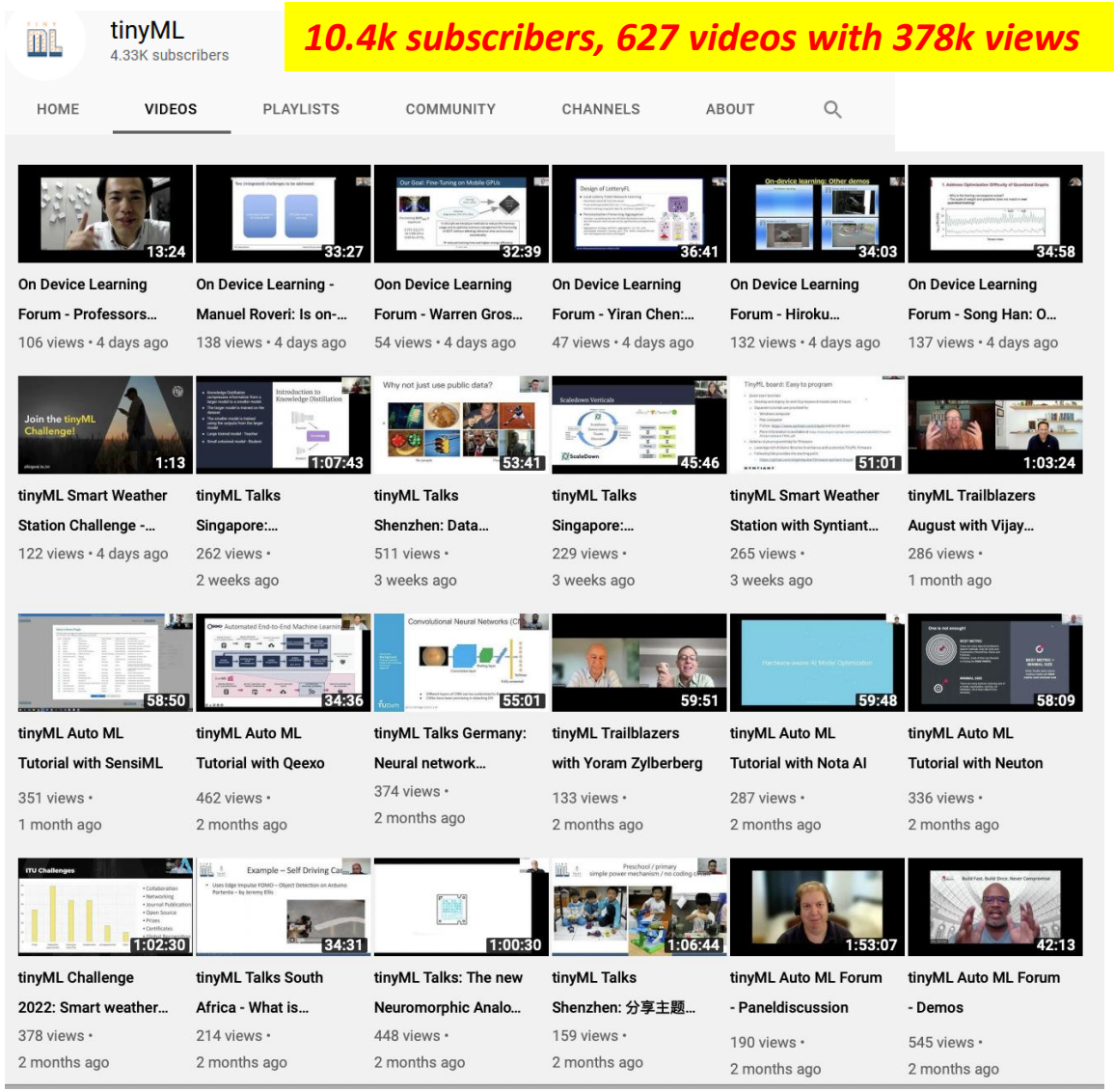
TINY ML TALKS webcast

Subscribe to tinyML YouTube Channel for updates and notifications

(including this video)

www.youtube.com/tinyML

YouTube



TINY ML FOUNDATION

Register now
https://www.tinyml.org/event/asia-2023/

WEVOLVER

https://www.wevolver.com/article/2023-edge-ai-technology-report

TINY ML FOUNDATION

2023 Edge AI Technology Report

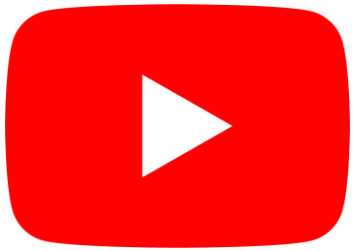
The guide to understanding the state of the art in hardware & software in Edge AI.

TINY ML TALKS webcast

Reminders

Slides & Videos will be posted tomorrow

tinyml.org/forums



youtube.com/tinyml

Please use the Q&A window for your questions

TINY ML TALKS webcast

Danilo Pau

Danilo PAU (h-index 26, i10-index 69) graduated in 1992 at Politecnico di Milano, Italy. One year before his graduation, he joined SGS-THOMSONS (now STMicroelectronics) as interns on Advanced Multimedia Architectures, and he worked on memory reduced HDMAC HW design. Then MPEG2 video memory reduction. Next, on video coding, transcoding, embedded 2/3D graphics, and computer vision. Currently, his work focuses on developing solutions for tiny machine learning tools.

Since 2019 Danilo is an IEEE Fellow and AAIA on 2023; he served as Industry Ambassador coordinator for IEEE Region 8 South Europe, was vice-chairman of the “Intelligent Cyber-Physical Systems” Task Force within IEEE CIS, was IEEE R8 AfI member in charge of internship initiative. Today he is a Member of the Machine Learning, Deep Learning and AI in the CE (MDA) Technical Stream Committee CESoc. He was AE of IEEE TNNLS. He wrote the IEEE Milestone on Multiple Silicon Technologies on a chip, 1985 which was ratified by IEEE BoD in 2021 and IEEE Milestone on MPEG Multimedia Integrated Circuits, 1984-1993 which was ratified in 2022. He served as TPC member to TinyML EMEA forum and is the chair of the TinyML on Device Learning working group. He serves as 2023 IEEE Computer Society Fellow Evaluating Committee Members

With 78 and 68 respectively European and US application patents, 157 publications, 113 ISO/IEC/MPEG authored documents and 67 invited talks/seminars at various Universities and Conferences, Danilo's favorite activity remains supervising undergraduate students, MSc engineers and PhDs.

TINY ML TALKS webcast

Fabrizio Aymone

Fabrizio M. Aymone is currently pursuing a Bachelor degree in Electronics Engineering at Politecnico di Milano. He is also intern at the System Research and Applications department of STMicroelectronics, where he is studying solutions for On-Device Learning in the domain of tiny devices. His research interests focus on reducing memory usage and computational complexity of AI algorithms and exploring alternative learning rules to backpropagation.

Suitability of Forward-Forward and PEPITA Learning to MLCommons-Tiny benchmarks

Danilo P. Pau and Fabrizio M. Aymone

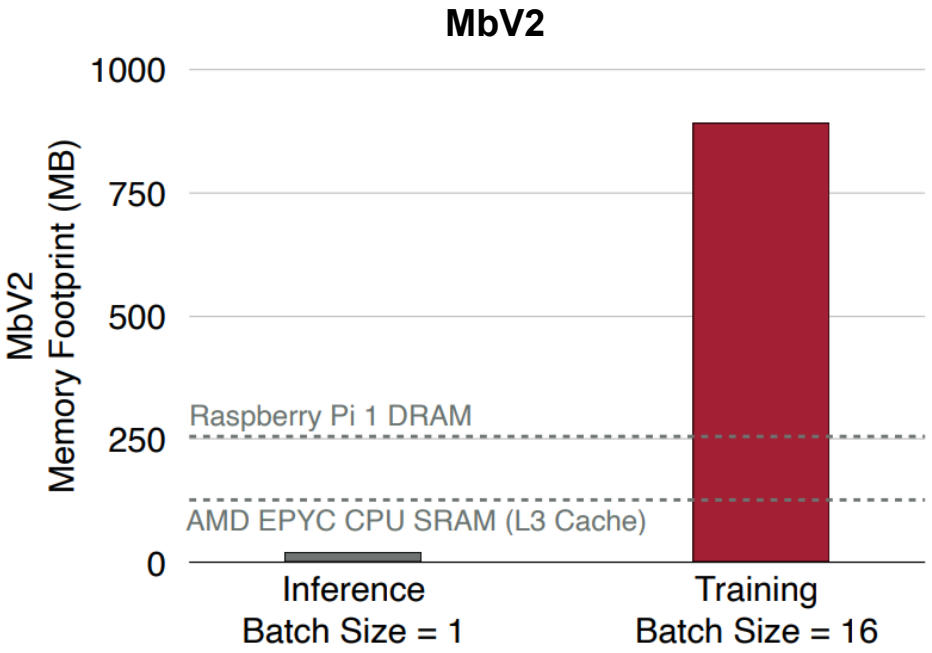
Sept, 19 2023

Agenda

1. Back-propagation
2. On-Device Learning
3. Forward-Forward and PEPITA
4. The research question
5. Methodology and Results
6. Takeways
7. Future works
8. Q&A

life.augmented

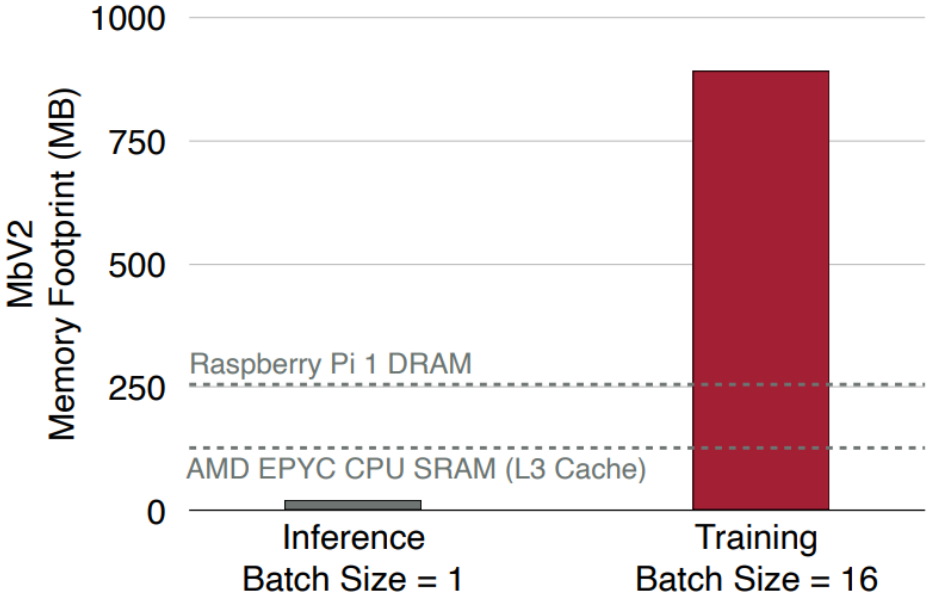
Reduce activations, not trainable parameters for efficient on-device learning[1]

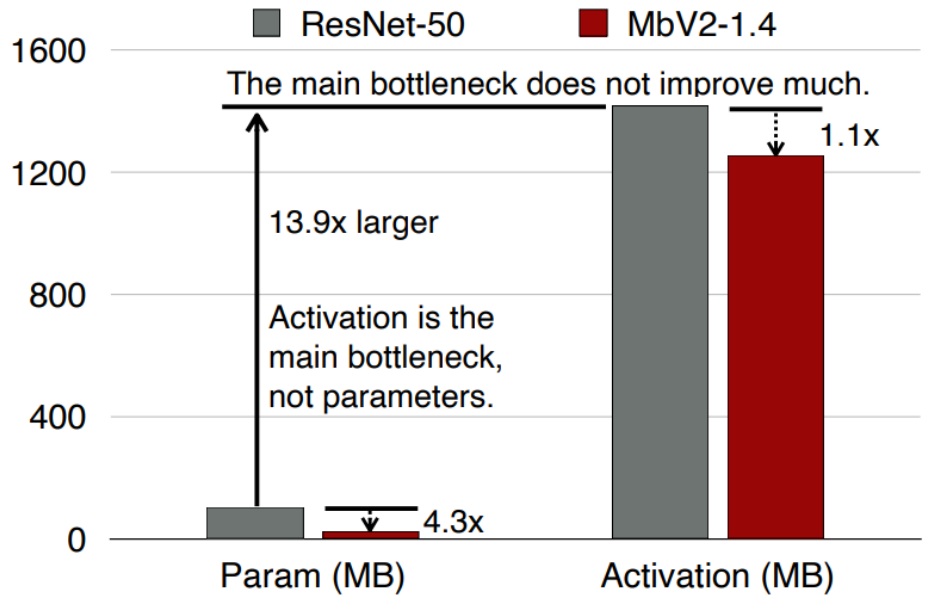


[1]H. Cai, C. Gan, L. Zhu, and S. Han, “Tinytl: Reduce memory, not parameters for efficient on-device learning,” in Advances in Neural Information Processing Systems, vol. 33, 2020

life.augmented

Reduce activations, not trainable parameters for efficient on-device learning[1]





[1]H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce memory, not parameters for efficient on-device learning," in Advances in Neural Information Processing Systems, vol. 33, 2020

life.augmented

Backpropagation

Activations are computed for each layer and stored into memory

- The loss is calculated w.r.t. the ground truth and the final output;
- The gradient of the loss in then computed;
- Then, the derivative of the output activations z_l
- Finally, the derivative of the loss function for the output layer is computed as Hadamard product

Algo 1 Backpropagatio

Fo rd Pass

for $= 1, ..., L$ **do**

$a_l = \sigma_l(W_l a_{l-1} + b_l)$

end for

Backward pass layer L

$\delta_L = \nabla_{a_L} L(a_L, target) \odot \sigma'(z_L)$

$W_L = W_L - \delta_L a^T_{previous}$

Backward pass previous layers

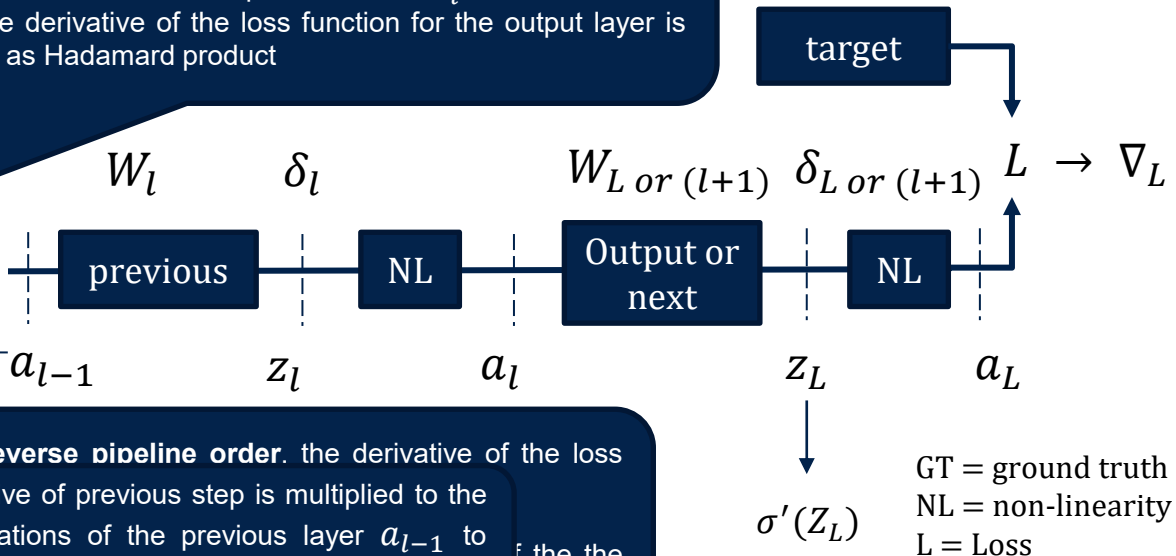
for $l = previous, ...$ **do**

$\delta_l = W^T_{l+1} \delta_{l+1} \odot \sigma'(z_l)$

Weight update

$W_l = W_l - \delta_l a^T_{l-1}$

end for



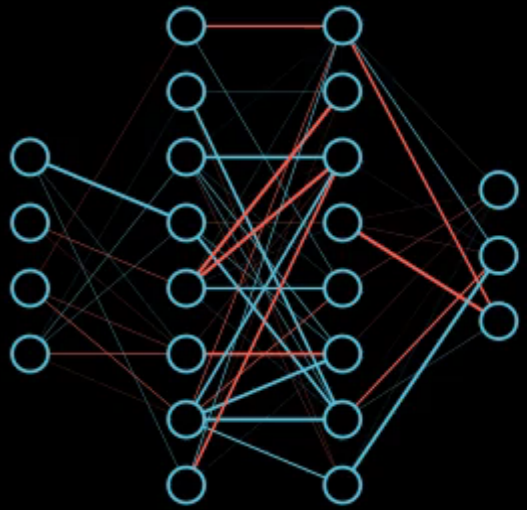
in reverse pipeline order. the derivative of the loss

the derivative of previous step is multiplied to the input activations of the previous layer a_{l-1} to compute the variation of the weights that are added to update the weights of the previous layer

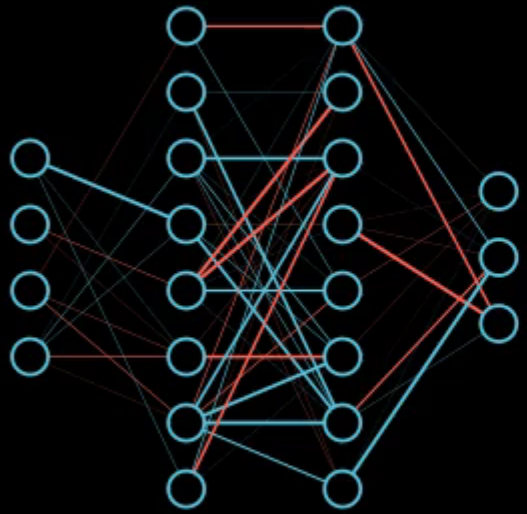
f the the

GT = ground truth
NL = non-linearity
L = Loss

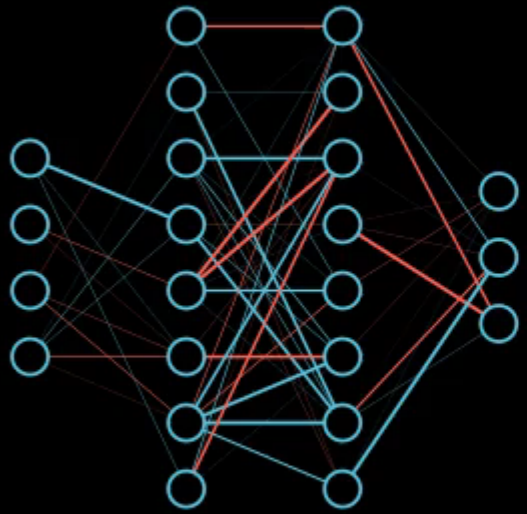
Backpropagation



Backpropagation



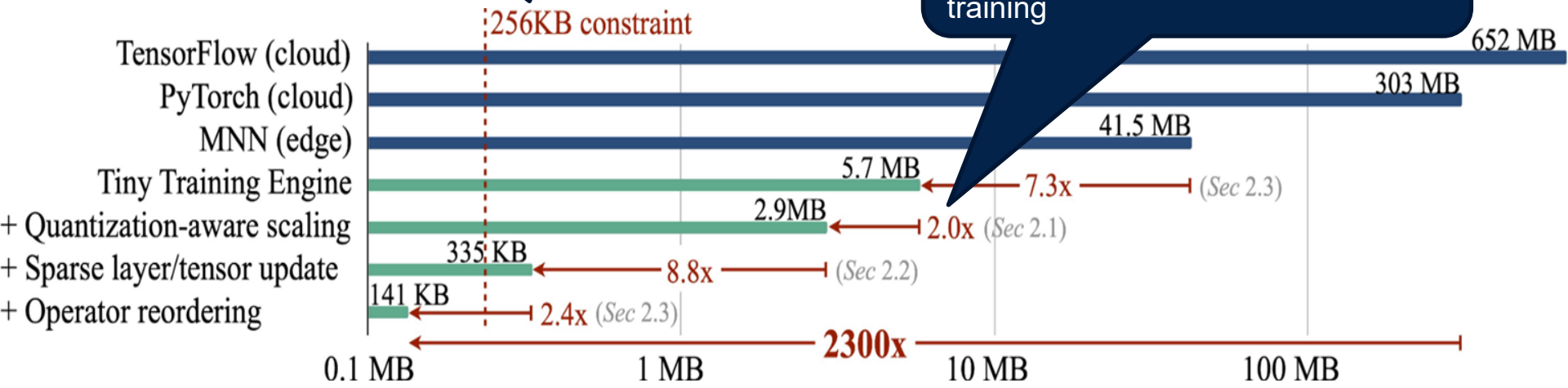
Backpropagation



MCUNetV3[2]

State of art on ODL CNN learning in only **256KiB SRAM** Visual Wake Words

QAS (Quantization Aware Scaling): mitigated the backprop instability due to the int8 quantization error during training



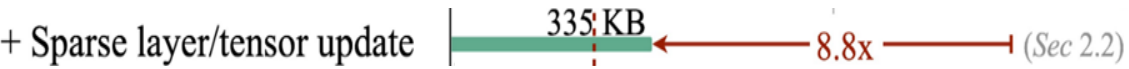
[2]J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, C. Gan, and S. Han, "On-device training under 256kb memory," 2022. [Online]. Available: https://arxiv.org/abs/2206.15472

life.augmented

MCUNetV3[2]

Sparse Update:

- updates only **some** parameters of the model.
- these parameters are selected **offline** according to how much they contribute to reduce the error during training.
- it needs to store only the intermediate activations of such parameters



[2]J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, C. Gan, and S. Han, "On-device training under 256kb memory," 2022. [Online]. Available: https://arxiv.org/abs/2206.15472

life.augmented

Hardware acceleration

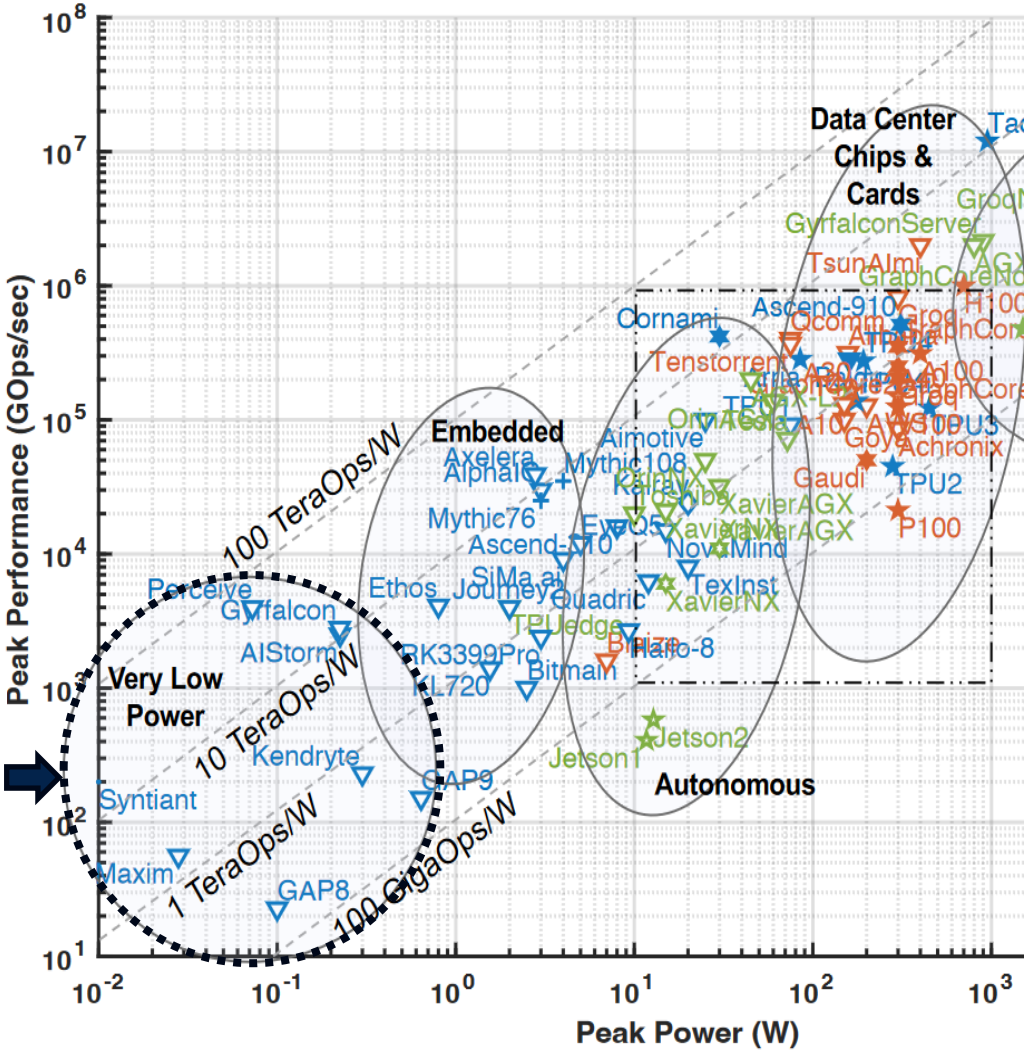
Application	Hardware accelerator	Acronym
Computer graphics	•General-purpose computing on GPU •CUDA architecture •Ray-tracing hardware	•GPGPU •CUDA •RTX
Digital signal processing	Digital signal processor	DSP
Analog signal processing	•Field-programmable analog arrayField-programmable RF	•FPAAFPRF
Sound processing	Sound card and sound card mixer	N/A
Computer networking on a chip	•Network processor and network interface controller Network on a chip	•NPU and NICNoC
CryptographyEncryption Attack Random number generation	•Cryptographic accelerator and secure cryptoprocessorHardware-based encryption •Custom hardware attack •Hardware random number generator	N/A
Artificial intelligenceMachine vision/computer vision Neural networks Brain simulation	•AI acceleratorVision processing unit •Physical neural network •Neuromorphic engineering	•N/AVPU •PNN •N/A
Multilinear algebra	Tensor processing unit	TPU
Physics simulation	Physics processing unit	PPU
Regular expressions[16]	Regular expression coprocessor	N/A
Data compression[17]	Data compression accelerator	N/A
In-memory processing	Network on a chip and Systolic array	NoC; N/A
Data processing	Data processing unit	DPU
Any computing task	•Computer hardwareField-programmable gate arrays[18] •Application-specific integrated circuits[18] •Complex programmable logic devices •Systems-on-Chip	•HW (sometimes)FPGA •ASIC •CPLD •SoC

Hardware acceleration is the use of computer hardware designed to perform specific functions more efficiently when compared to software running on a general-purpose central processing unit (CPU). *

* https://en.wikipedia.org/wiki/Hardware_acceleration

Edge AI drives hardware for on-device machine learning inference. 1

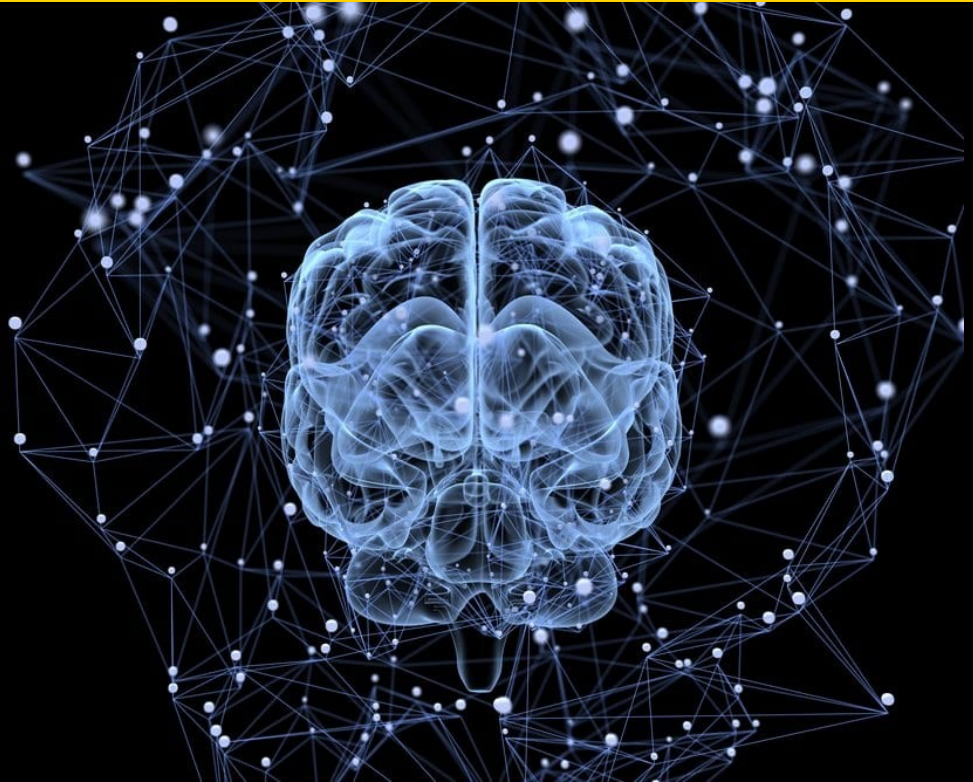
Tiny ML hardware (today) is optimized for forward workloads 2



@article{Reuther2019SurveyAB, title={Survey and Benchmarking of Machine Learning Accelerators}, author={A. Reuther and Peter Michaleas and Michael Jones and Vijay Gadepally and Siddharth Samsi and Jeremy Kepner}, journal={2019 IEEE High Performance Extreme Computing Conference (HPEC)}, year={2019}, pages={1-9}, url={https://api.semanticscholar.org/CorpusID:201668230} }

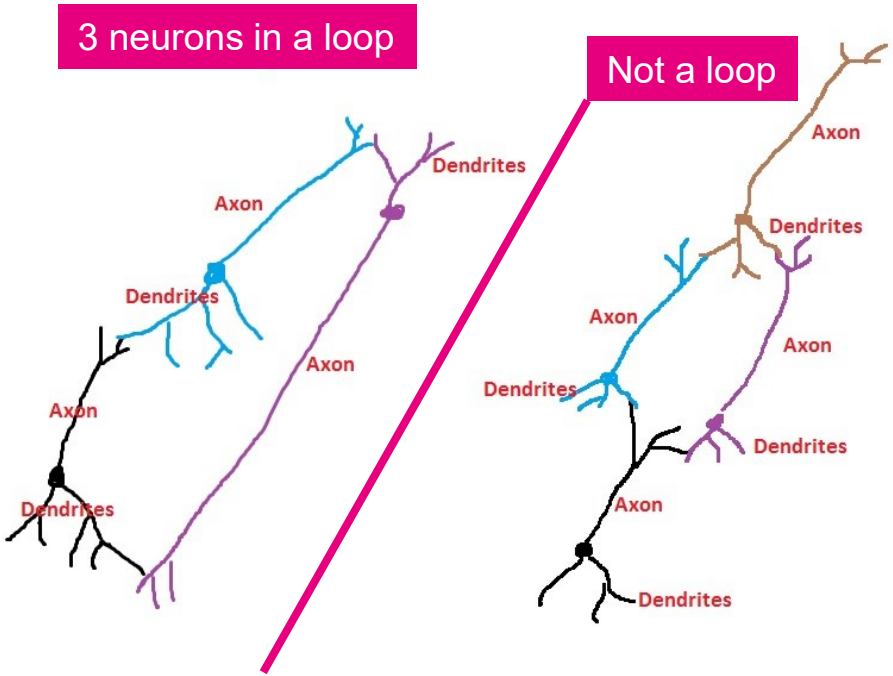
life.augmented

Forward-Forward and PEPITA



Learning: biological plausibility

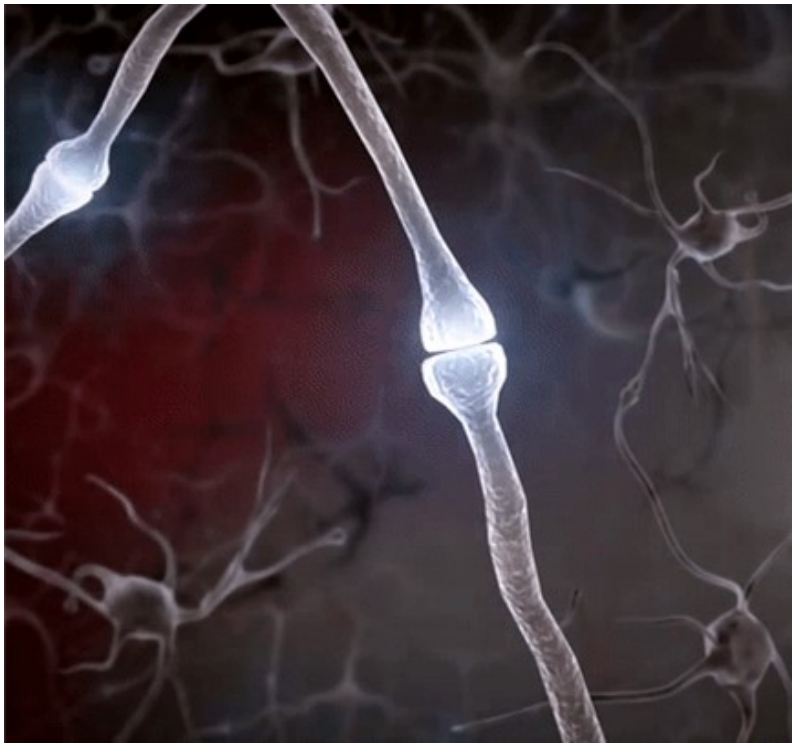
- The brain **learns** by modifying the synaptic individual connections between neurons[3]
- It's **not known how** the single modifications are coordinated to achieve a global's goal
- **Loop-based** neuron circuits seems used to get error signals and credits (i.e. how much each synapse contributes to the error) assigned to other synapsis of neurons



[3]T. Lillicrap, A. Santoro, L. Marris, C. Akerman, and G. Hinton,"Backpropagation and the brain," Nature Reviews Neuroscience, vol. 21, no. 6, p. 335–346, 2020.

Backpropagation vs bio-plausibility?

Our brain does not use backpropagation[4]



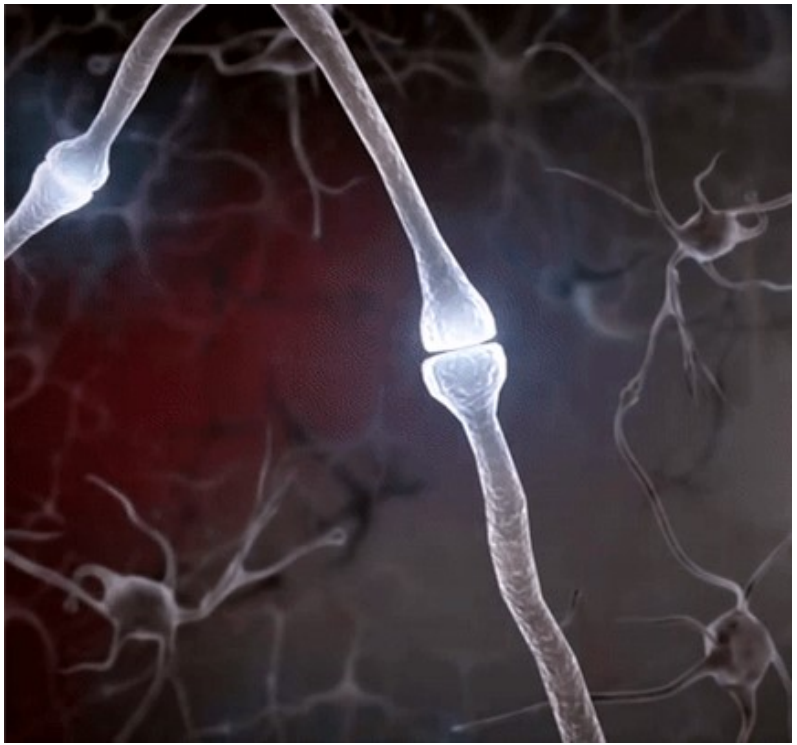
[4]Crick, “The recent excitement about neural networks,” Nature, vol. 337, no. 6203, p. 129—132, January 1989. [Online]. Available: https://doi.org/10.1038/337129a0

life.augmented

Backpropagation vs bio-plausibility?

Our brain does not use backpropagation[4]

1. No weight symmetry
    a) error is not projected back using the same weights of the forward pass

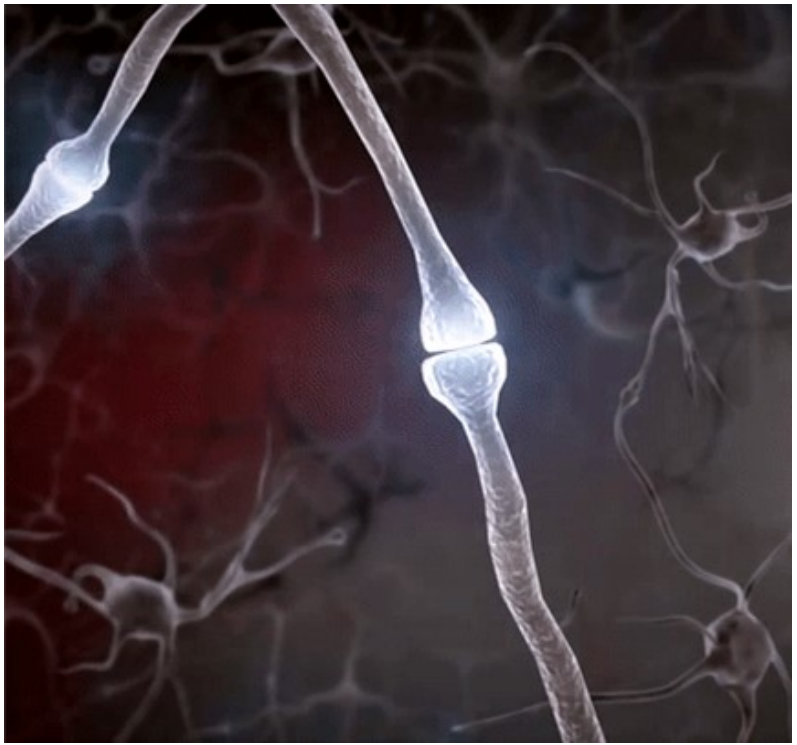


[4]Crick, “The recent excitement about neural networks,” Nature, vol. 337, no. 6203, p. 129—132, January 1989. [Online]. Available: https://doi.org/10.1038/337129a0

Backpropagation vs bio-plausibility?

Our brain does not use backpropagation[4]

1. No weight symmetry
2. No neural activity freeze
   a) intermediate activations are not stored

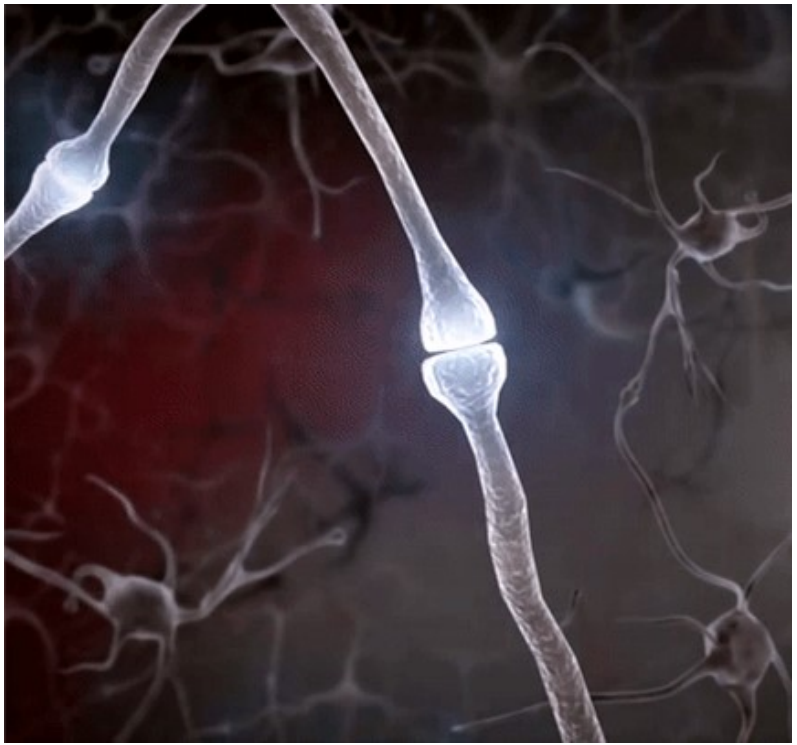


[4]Crick, “The recent excitement about neural networks,” Nature, vol. 337, no. 6203, p. 129—132, January 1989. [Online]. Available: https://doi.org/10.1038/337129a0

Backpropagation vs bio-plausibility?

Our brain does not use backpropagation[4]

1. No weight symmetry
2. No neural activity freeze
3. No locality of the loss function
    a) Neurons do exchange error signals and credits within loops
    b) Synapsis learn from local signals



[4]Crick, “The recent excitement about neural networks,” Nature, vol. 337, no. 6203, p. 129—132, January 1989. [Online]. Available: https://doi.org/10.1038/337129a0

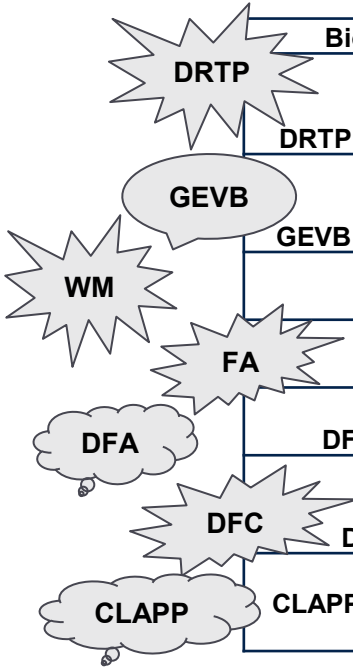
Backpropagation vs bio-plausibility?

Our brain does not use backpropagation[4]

1. No weight symmetry
2. No neural activity freeze
3. No locality of the loss function
4. No Update-locking
   a) No need to wait the end of the backward pass to update the weights of the layers
   b) There is no backward pass

[4]Crick, “The recent excitement about neural networks,” Nature, vol. 337, no. 6203, p. 129—132, January 1989. [Online]. Available: https://doi.org/10.1038/337129a0

A lot of work in 30 years…



Biologically plausible method	Citation
DRTP Direct Random Target Projection	C. Frenkel, M. Lefebvre and D. Bol, "Learning without feedback: Fixed Random Learning Signals Allow for Feedforward Training of Deep Neural Networks," Frontiers in Neuroscience, vol. 15, no. 629892, 2021. doi: 10.3389/fnins.2021.629892
GEVB Global Error Vector Broadcasting	Clark, David, L. F. Abbott, and SueYeon Chung. "Credit assignment through broadcasting a global error vector." Advances in Neural Information Processing Systems 34 (2021): 10053-10066.
WM Weight Mirroring	Akrout, Mohamed, et al. "Deep learning without weight transport." Advances in neural information processing systems 32 (2019).
FA Feedback Alignment	Lillicrap, Timothy P., et al. "Random feedback weights support learning in deep neural networks." arXiv preprint arXiv:1411.0247 (2014).
DFA Direct Feedback Alignment	Nøkland, Arild. "Direct feedback alignment provides learning in deep neural networks." Advances in neural information processing systems 29 (2016).
DFC Deep Feedback Control	Meulemans, Alexander, et al. "Credit assignment in neural networks through deep feedback control." Advances in Neural Information Processing Systems 34 (2021): 4674-4687.
CLAPP Contrastive Local And Predictive Plasticity	Illing, Bernd, et al. "Local plasticity rules can learn deep representations using self-supervised contrastive predictions." Advances in Neural Information Processing Systems 34 (2021): 30365-30379.

A lot of work in 30 years…

Biologically plausible method	Citation
DRTP Direct Random Target Projection	C. Frenkel, M. Lefebvre and D. Bol, "Learning without feedback: Fixed Random Learning Signals Allow for Feedforward Training of Deep Neural Networks," Frontiers in Neuroscience, vol. 15, no. 629892, 2021. doi: 10.3389/fnins.2021.629892
GEVB Global Error Vector Broadcasting	Clark, David, L. F. Abbott, and SueYeon Chung. "Credit assignment through broadcasting a global error vector." Advances in Neural Information Processing Systems 34 (2021): 10053-10066.
WM Weight Mirroring	Akrout, Mohamed, et al. "Deep learning without weight transport." Advances in neural information processing systems 32 (2019).
FA Feedback Alignment	Lillicrap, Timothy P., et al. "Random feedback weights support learning in deep neural networks." arXiv preprint arXiv:1411.0247 (2014).
DFA Direct Feedback Alignment	Nøkland, Arild. "Direct feedback alignment provides learning in deep neural networks." Advances in neural information processing systems 29 (2016).
DFC Deep Feedback Control	Meulemans, Alexander, et al. "Credit assignment in neural networks through deep feedback control." Advances in Neural Information Processing Systems 34 (2021): 4674-4687.
CLAPP Contrastive Local And Predictive Plasticity	Illing, Bernd, et al. "Local plasticity rules can learn deep representations using self-supervised contrastive predictions." Advances in Neural Information Processing Systems 34 (2021): 30365-30379.
Forward-Forward	**Hinton, Geoffrey.** "The forward-forward algorithm: Some preliminary investigations." arXiv preprint arXiv:2212.13345 (2022).
Present the Error to Perturb the Input To modulate Activity	**Dellaferrera, Giorgia,** and Gabriel Kreiman. "Error-driven input modulation: solving the credit assignment problem without a backward pass." International Conference on Machine Learning. PMLR, 2022.

DRTP
GEVB
WM
FA
DFA
DFC
CLAPP
FF
PEPITA

life.augmented

Forward-Forward[5]

Positive data

Negative data

Algorithm 1 Forward-Forward

Given: Positive input (x_{pos}) and Negative input (x_{neg})

Positive Pass

$a_0 = x_{pos}$

for $\ell = 1, ..., L$ **do**

$a_\ell = ReLU(W_\ell a_{\ell-1})$

Weight update

$W_\ell = W_\ell + 2\frac{\partial log(p)}{\partial G(a_\ell)} a_\ell \cdot a_{\ell-1}^T$

$a_\ell = norm(\ell)$

end for

Negative Pass

$a_0 = x_{neg}$

for $\ell = 1, ..., L$ **do**

$a_\ell = ReLU(W_\ell a_{\ell-1})$

Weight update

$W_\ell = W_\ell - 2\frac{\partial log(p)}{\partial G(a_\ell)} a_\ell \cdot a_{\ell-1}^T$

$a_\ell = norm(a_\ell)$

end for

Forward inference

Activations of each layer are normalized

Forward inference

Activations of each layer are normalized

Positive weight update of each layer independently from the next ones **(locality)**

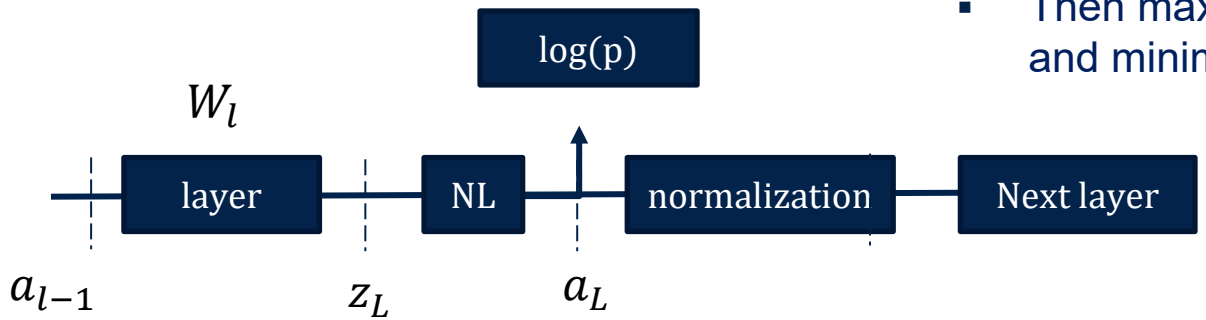
Negative weight update of each layer independently from the next ones **(locality)**

life.augmented

[5]G. Hinton, "The forward-forward algorithm: Some preliminary investigations," ArXiv, vol. abs/2212.13345, 2022.

Forward-Forward[5]

- 2 variants: supervised and unsupervised → **same learning procedure**, **different forward procedures**
- **Supervised variant: *n* passes, for *n* classes**



- The **goodness function** $G(a_l)$ is defined at each layer; e.g. it can be the sum of the squared a_l activations
- The probability of being classified as a positive data is

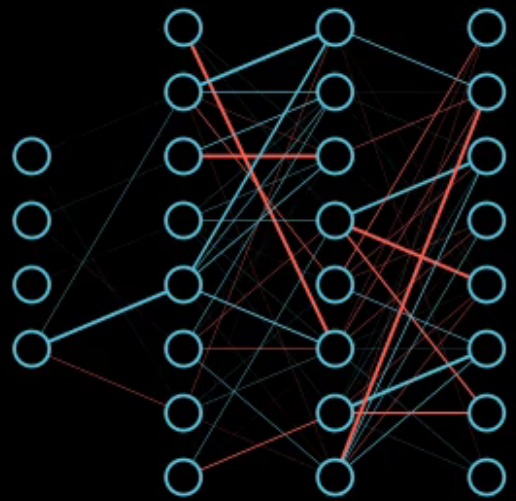
$$p(positive) = \sigma(G(a_\ell) - \theta)$$

- Then maximize the $\log(p)$ for positive data and minimize it for negative

[5]G. Hinton, "The forward-forward algorithm: Some preliminary investigations," ArXiv, vol. abs/2212.13345, 2022.

life.augmented

Forward-Forward



PEPITA[6]

Algorithm 3 PEPITA

Given: Features(x) and label($target$)

Standard Pass

$a_0 = x$

for $\ell = 1, ..., L$ **do**

$a_\ell = \sigma_\ell(W_\ell a_{\ell-1})$

end for

$e = a_L - target$

Modulated pass

$a_0^{err} = x + Fe$

for $\ell = 1, ..., L$ **do**

$a_\ell^{err} = \sigma_\ell(W_\ell a_{\ell-1}^{err})$

Weight update

$W_\ell = W_\ell - (a_\ell - a_\ell^{err}) \cdot (a_{\ell-1}^{err})^T$

end for

Forward inference
Store activations #

Compute the **error** at the output

Random **matrix F**, with zero mean and small variance, to project the **error on the inputs**

Forward inference
Store activations #

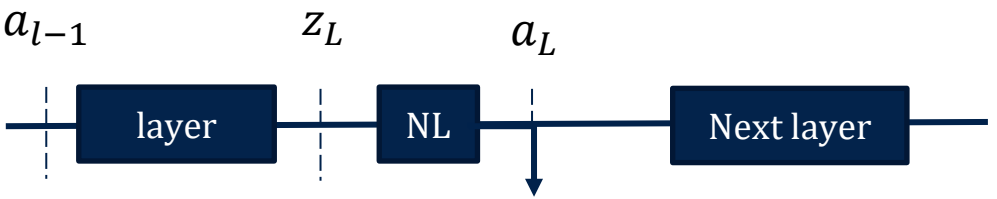
Weights update by using the forward pass computed activations (**stored in memory #**) with the modulated ones

[6]G. Dellaferrera and G. Kreiman, "Error-driven input modulation: Solving the credit assignment problem without a backward pass," ArXiv, vol. abs/2201.11665, 2022.

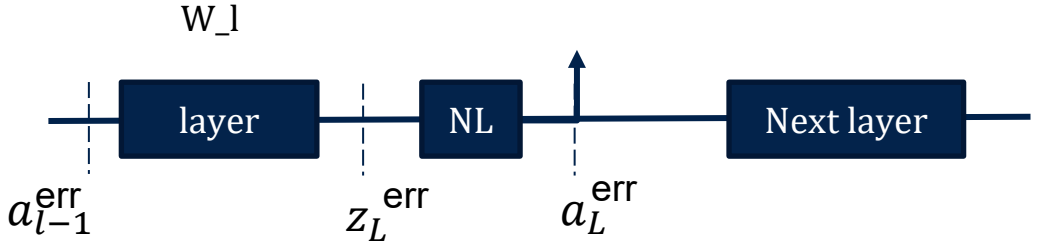
PEPITA[6]

$$W_\ell = W_\ell - (a_\ell - a_\ell^{err}) \cdot (a_{\ell-1}^{err})^T$$

Standard pass

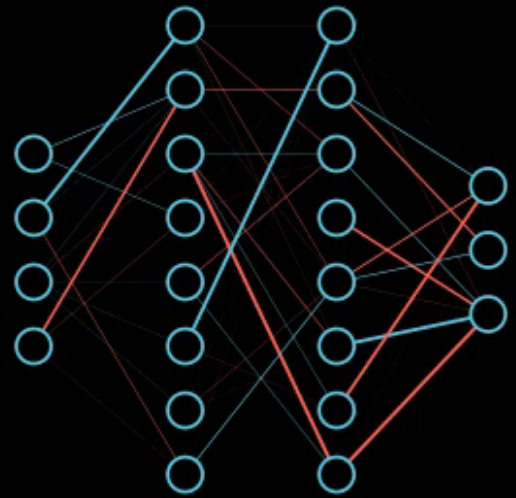


Modulated pass



[6]G. Dellaferrera and G. Kreiman, “Error-driven input modulation: Solving the credit assignment problem without a backward pass,” ArXiv, vol. abs/2201.11665, 2022.

PEPITA



Research question

At which computational and memory cost would FF and PEPITA learning algorithms compare to BP if applied to MLCommons/Tiny benchmarks ?

Contributions today

Introducing **MEMPEPITA** to not store intermediate activations. Memory savings expected !

Computational complexity and memory footprint of **FF, PEPITA and MEMPEPITA** for the **MLCommons/Tiny benchmarks**

MEMPEPITA

Algorithm 2 MEMPEPITA

Given: Features(x) and label($target$)

Standard Pass

$a_0 = x$

for $\ell = 1, ..., L$ **do**

$a_\ell = \sigma_\ell(W_\ell a_{\ell-1})$

end for

$e = a_L - target$

Error projection

$a_0^{err} = x + Fe$

for $\ell = 1, ..., L$ **do**

Standard pass

$a_\ell = \sigma_\ell(W_\ell a_{\ell-1})$

Modulated pass

$a_\ell^{err} = \sigma_\ell(W_\ell a_{\ell-1}^{err})$

Weight update

$W_\ell = W_\ell - (a_\ell - a_\ell^{err}) \cdot (a_{\ell-1}^{err})^T$

end for

Forward inference

Compute the **error** at the output

Random **matrix F**, with zero mean and small variance, to project the **error onto the inputs**

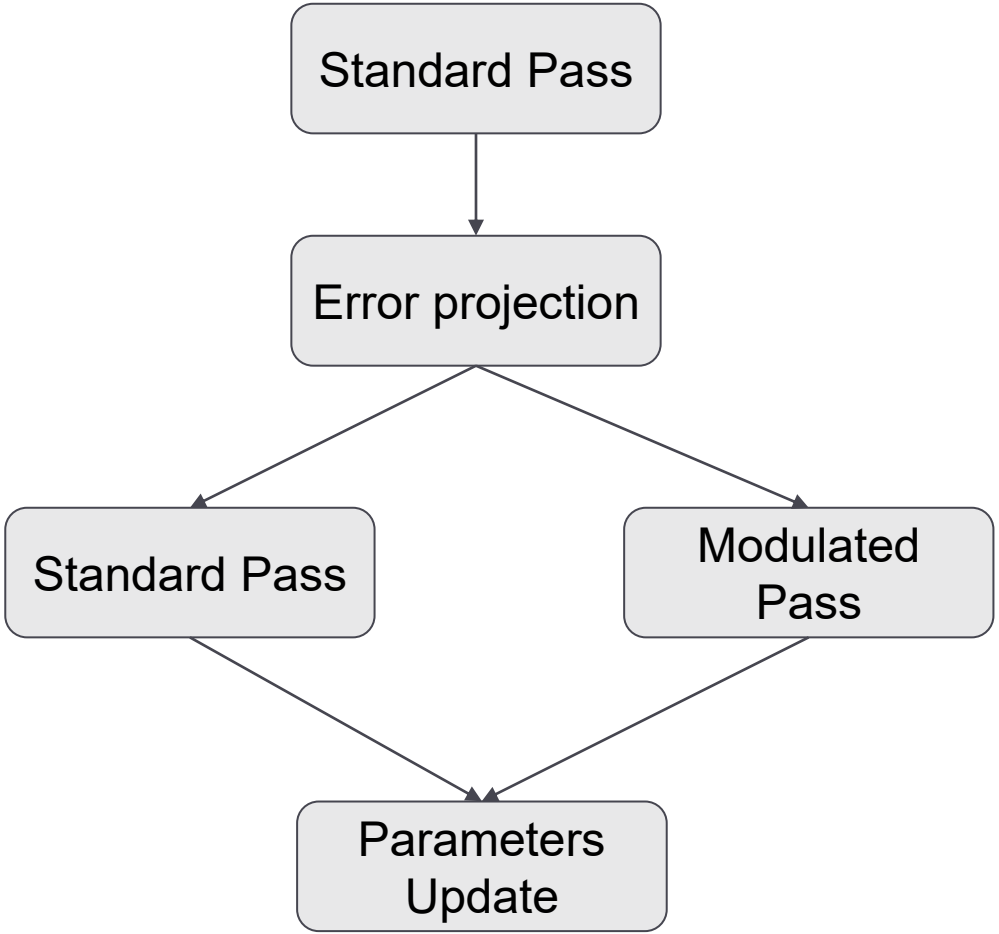
Forward inferences thus recomputing activations instead of storing them into memory

Introduces **a second standard pass** which runs simultaneously along with **modulated** pass

Weights update by using the forward pass re-computed activations with the modulated ones

life.augmented

MEMPEPITA



Summary of the **learning** procedures

Method	BP (number)	FF (number)	PEP (number)	MPE (number)
Forward passes	1	2	2	3
Backward passes	1	0	0	0
Weight update	1	2	1	1
Loss function	Global	Local	Global	Global
Activations	all	current	all	current

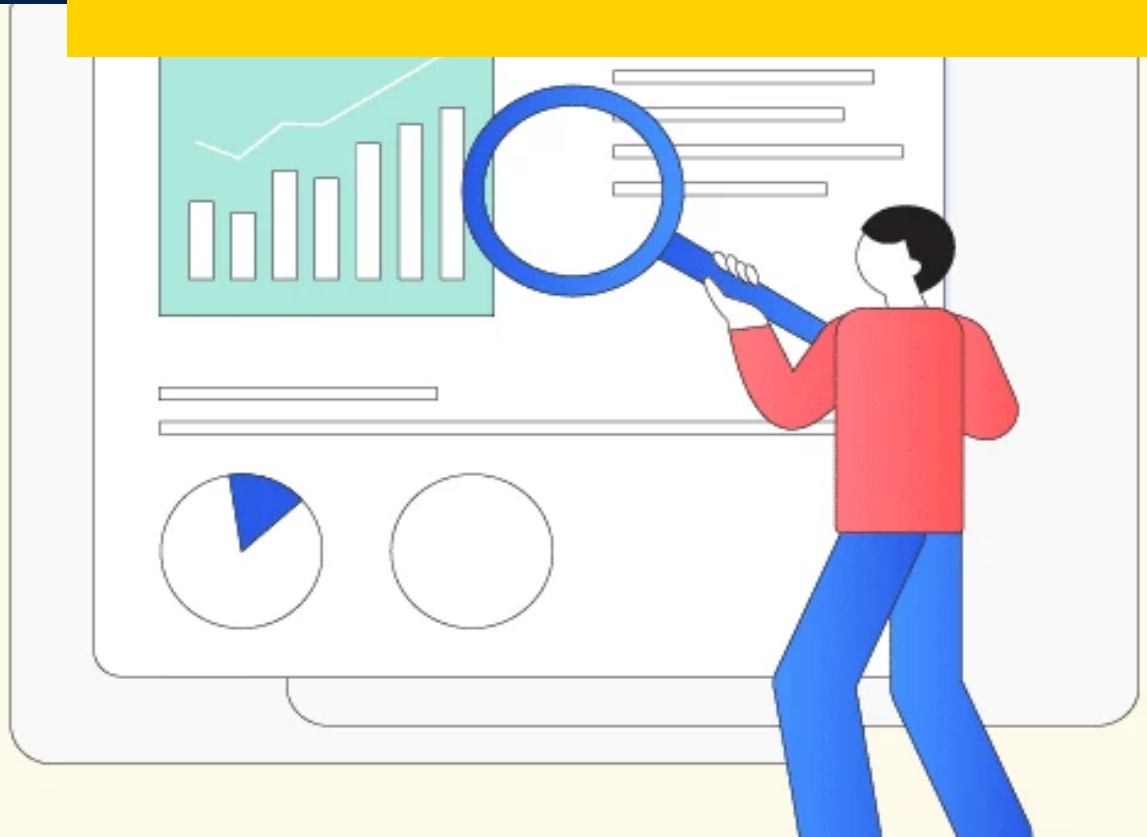
PEP = PEPITA
MPE = MEMPEPITA

Local = loss function per layer
Global = loss function at the output layer

All = all layers
Current = current layer

life.augmented

Methodology



MLCommons/Tiny

Use Case	Description	Dataset	Model	Quality Target (Closed Division)
Audio Wake Words	Small vocabulary keyword spotting	Speech Commands	DS-CNN	90% (Top1)
Visual Wake Words	Image classification (2 classes)	Person Detection	MobileNet	80% (Top1)
Image Classification	32x32 tiny Images Classification (10 classes)	Cifar10	ResNet	85% (Top1)
Anomaly Detection	Detecting anomalies in machine operating sounds	ToyADMOS	Deep AutoEncoder	0.85 (AUC)

https://github.com/mlcommons/tiny

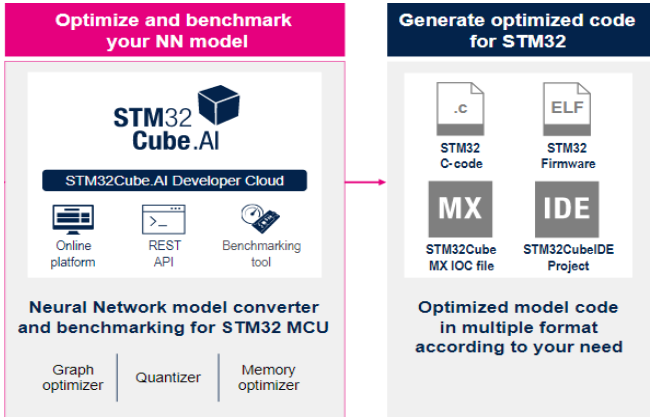
MCU deployability

- ML Commons framework specifies number of samples and epochs
- Weights, biases, activations represented in INT8
- Softmax layer represented in FLOAT32
- MACCs represented in INT8
- No layer memory overwrite
- Batch normalization not considered.
- $Cycles/MACC$ and processor's frequency
- Results validated with **STM32Cube.AI Developer Cloud**

NUCLEO-G474RE
170MHz

STM32H735G-DK
550MHz

https://stm32ai-cs.st.com/home



Complexity analisys: Assumptions

Computations required	Forward pass	Error at output layer	Backward pass	Weight update	Goodness function	Normalization	Error projection
BP	1	1	1	1			
FF	2			2	2	2	
PEP	2	1		1			1
MPE	3	1		1			1

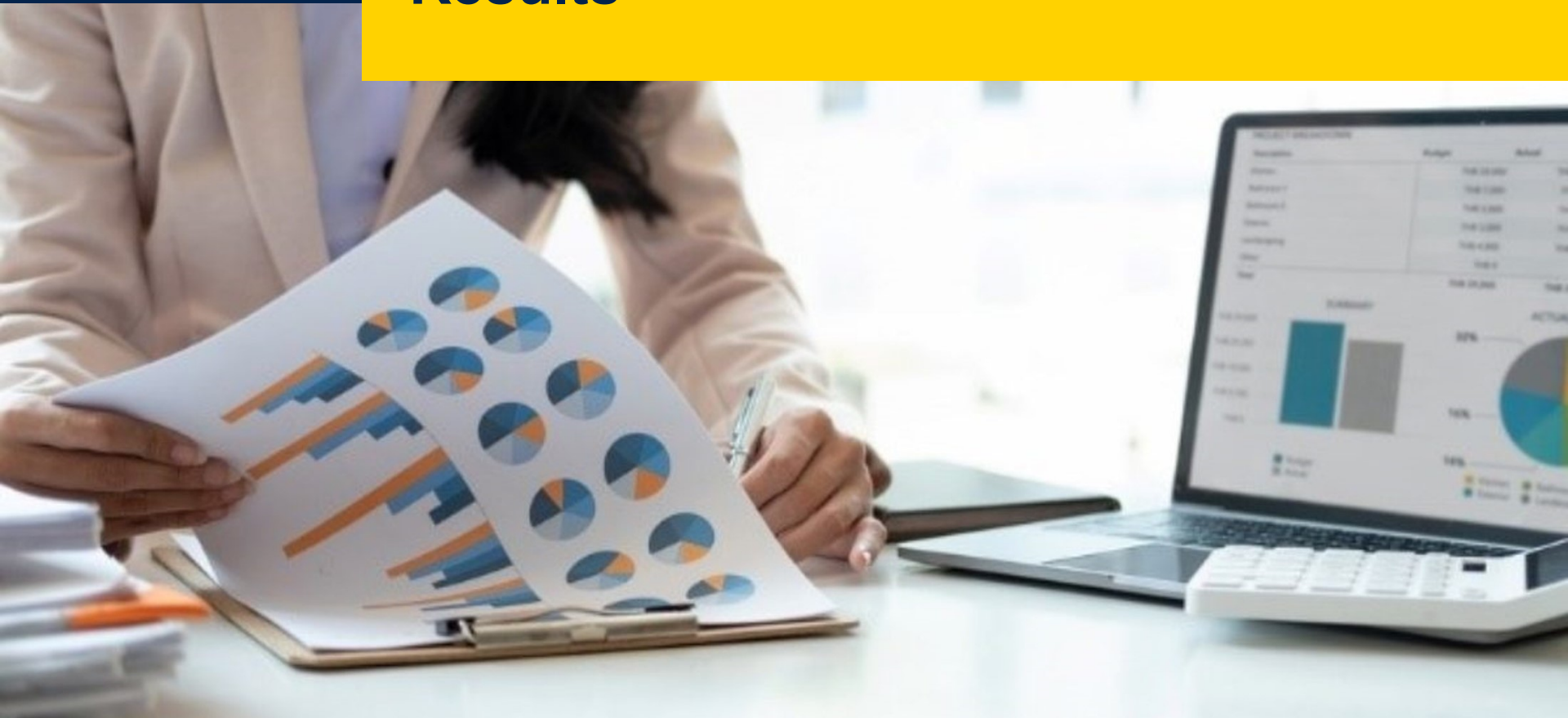
RAM estimation

Learning procedure	Activations RAM during training
BP and PEP	Sum of the activation buffers of all layers
FF	Max value of the sum of the activation buffers of two consecutive layers + the input sample
MPE	Max value of the sum of the activation buffers of two consecutive layers + the largest activation buffer between these two layers

$$\textbf{Total } \boldsymbol{RAM = Activations\ RAM\ + footprint\ (weights + biases)}$$

life.augmented

Results



Learning procedure: the analysis

Model/Dataset		DS-CNN/SC/KWS				MobileNet/VWW				ResNet/Cifar10				AE/ToyADMOS/Anomaly Detection			
Learning method		BP	FF	PEP	MPE	BP	FF	PEP	MPE	BP	FF	PEP	MPE	BP	FF	PEP	MPE
Training	MACC (M)	7.7	+43%	+4%	+39%	22	+41%	+3%	+37%	37	+36%	+1%	+35%	0.7	+50%	+69%	+106%
	ACT (KiB)	+253 %	21	+253%	+37%	+213%	83	+213%	+11%	+118%	52	+118%	+57%	+64%	1.4	+64%	1.4
	PAR/ ACT	0.3	1.1	0.3	0.8	0.8	2.5	0.8	2.3	0.7	1.5	0.7	0.9	115	189	115	189
	RAM (KiB)	+120 %	43	+120%	+17%	+60%	294	+60%	+3%	+47%	130	+47%	+23%	+0.3%	267	+0.3%	267

averaged

Architecture		CNN			FC(AE)		
Learning method		FF	PEP	MPE	FF	PEP	MPE
Train	MACC	+40%	+3%	+37%	+50%	+69%	+106%
	ACT	-65%	0%	-51%	-39%	0%	-39%
	RAM	-41%	0%	-33%	-0.3%	0%	-0.3%

life.augmented

Inference procedure: the analysis

Model/Dataset		DS-CNN/SC				MobileNet/VWW				ResNet/Cifar10				AE/ToyADMOS			
Inference method		BP	FF	PEP	MPE	BP	FF	PEP	MPE	BP	FF	PEP	MPE	BP	FF	PEP	MPE
Inference	MACC (M)	3	+1167 %	3	3	8	+113%	8	8	13	+914%	13	13	0.3	0.3	0.3	0.3
	RAM (KiB)	20	+2.4%	20	20	55	+50%	55	55	49	+6%	49	49	0.8	0.8	0.8	0.8
	ROM (KiB)	22.604				210.85				77.706				265.864			

averaged

		CNN			FC(AE)		
Inference method		FF	PEP	MPE	FF	PEP	MPE
Inference	MACC	+731%	0%	0%	+1%	0%	0%
	RAM	+20%	0%	0%	0%	0%	0%

life.augmented

Latency per input sample on MCUs

H7 = STM32H735G-DK @ 550 MHz
G4 = NUCLEO-G474RE @ 170 MHz

Model/Dataset		DS-CNN/SC				MobileNet/VWW				ResNet/Cifar10				AE/ToyADMOS			
Learning method		BP	FF	PEP	MPE	BP	FF	PEP	MPE	BP	FF	PEP	MPE	BP	FF	PEP	MPE
Training (ms)	H7	42	+43%	+4%	+39%	122	+41%	+3%	+37%	202	+36%	+1%	+35%	4	+50%	+69%	+106%
	G4	203	+43%	+4%	+39%	592	+41%	+3%	+37%	982	+36%	+1%	+35%	19	+50%	+69%	+106%
Inference (ms)	H7	14.5	+1165 %	14.5	14.5	42	+113%	42	42	68	+914%	68	68	1.4	+1%	1.4	1.4
	G4	71	+1165 %	71	71	202	+113%	202	202	332	+914%	332	332	7	+1%	7	7

life.augmented

Take-aways

- **LEARNING → FF** and **MEMPEPITA**
  - reduced activations (ACT) storage on average **40% to 65% (w.r.t. BP)**,
  - increased MACCs **40% to 100%**
  - **PEPITA** (same memory as BP) increased MACCS (CNN) **3%**, (FC) **69%**.
- Total **RAM reduction** is **noticeable** if the topology has **low parameters/activations** (FC vs CNN). → e.g. for DS-CNN is **0.3** and RAM reduction is around **-100%**
- **INFERENCE → MEMPEPITA, PEPITA** and **BP** featured **1 forward** pass, while supervised **FF** adds **2-3x** more computation due to **N forward** passes, for N classes

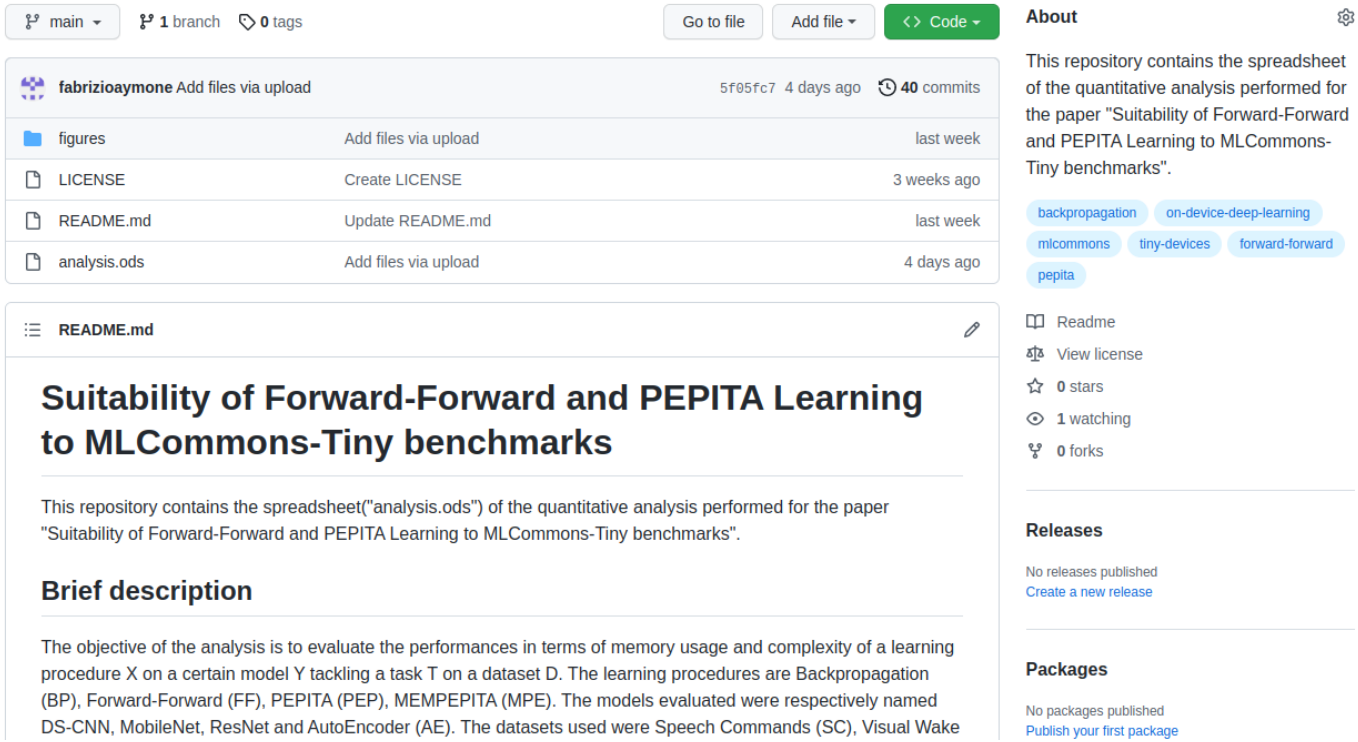
REMEMBER

About training → MEMPEPITA reduced total **RAM**, **(CNN) 33%, (FC) 0.3%**, at the expense of a **third more MACCs**.

Inference complexity was **unchanged**

Github repo

https://github.com/fabrizioaymone/suitability-of-Forward-Forward-and-PEPITA-learning



Q&A

Further questions ? Please contact: danilo.pau@st.com

Our technology starts with You

Find out more at www.st.com

life.augmented

Copyright Notice

www.tinyml.org

Copyright Notice

www.tinyml.org