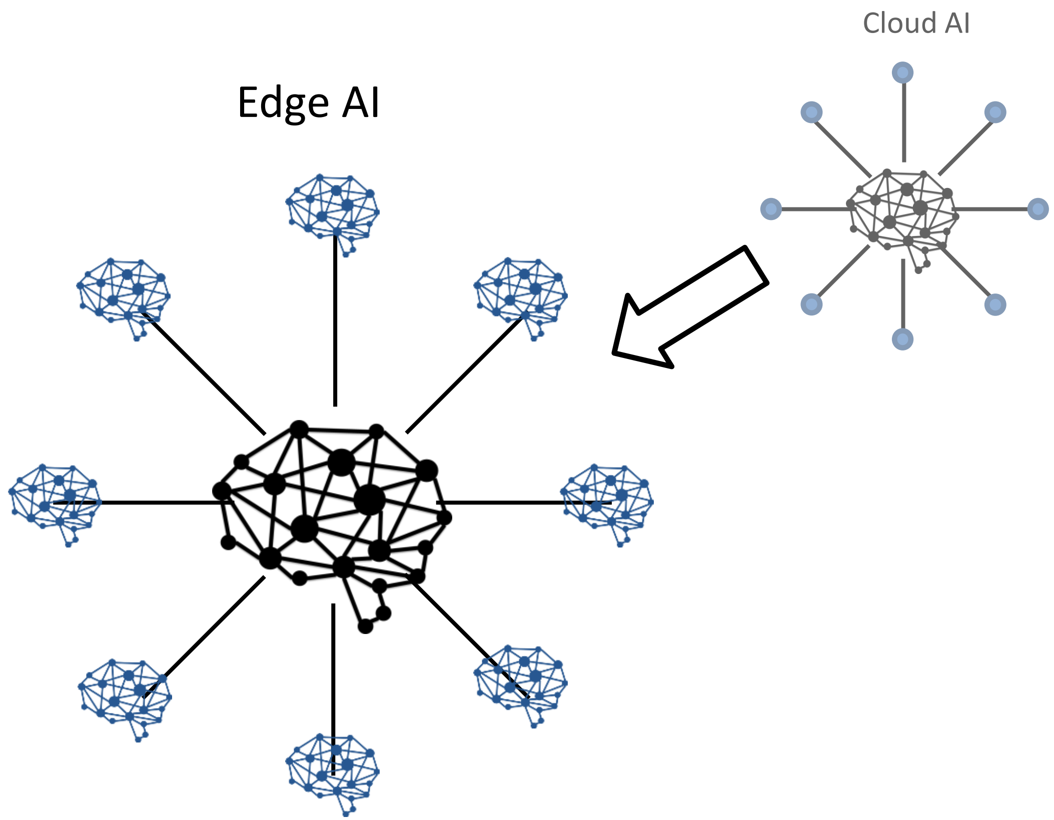
OnDevice Learning

Hahn-Schickard
Marcus.rueb@hahn-schickard.de

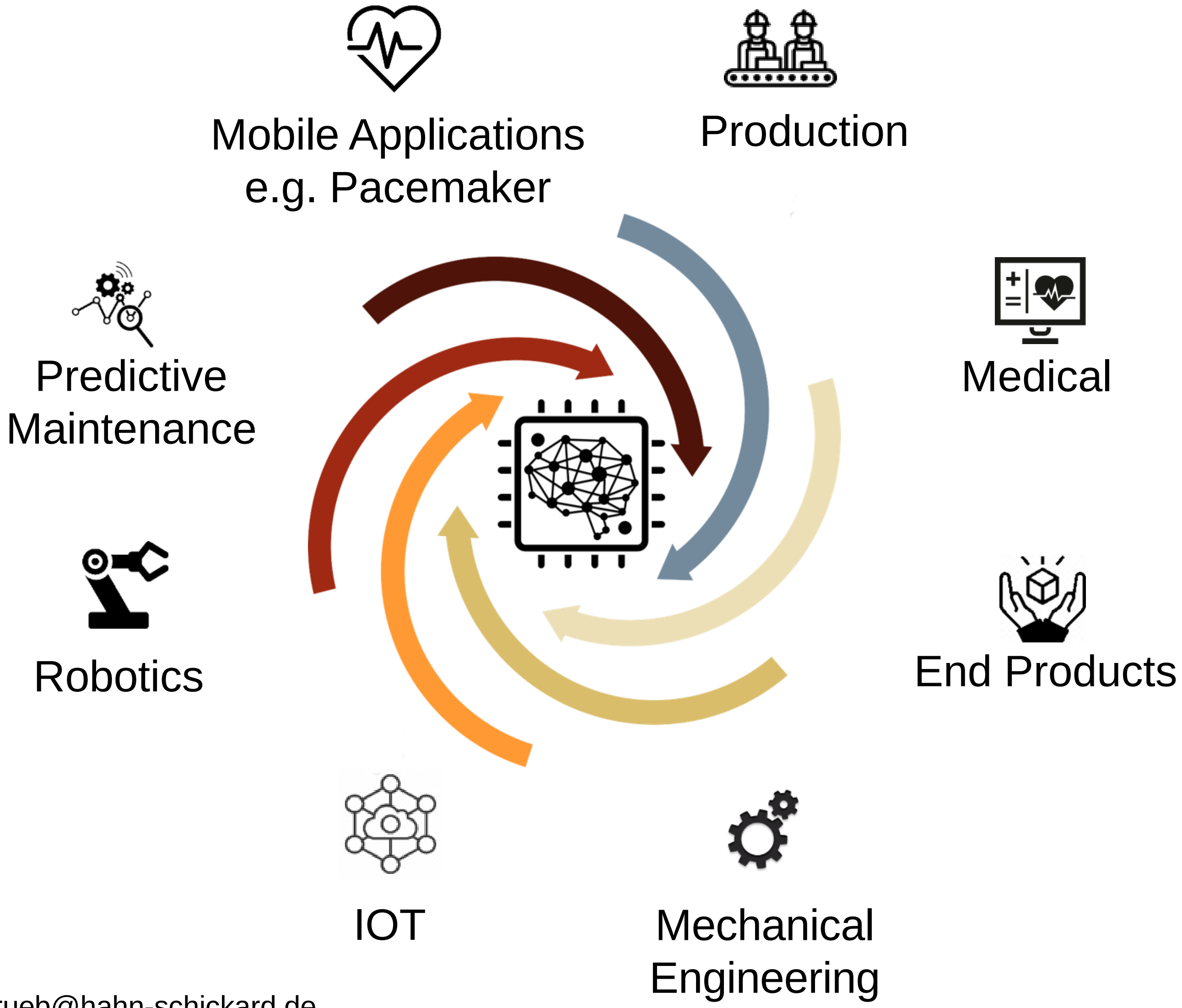
What is OnDevice Training and why is it needed?

Advantages [1]:

- Privacy
- Continuously Learning
- Energy-saving
- Less Communication
- Personalized Models

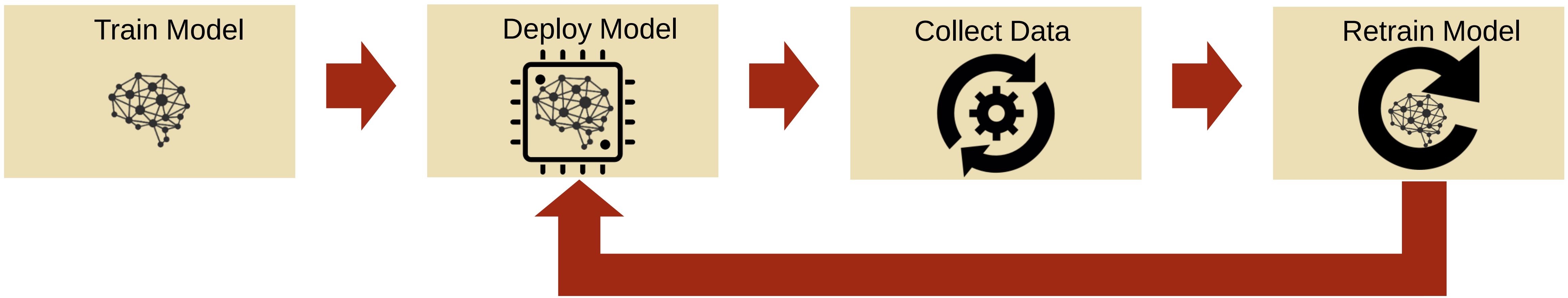


Applications



Hahn-Schickard - Marcus Rüb- Marcus.rueb@hahn-schickard.de

What does the workflow look like?



Problems of OnDevice Training

01 Limited Memory

02 Limited Calculation capacity

03 Limited Energie

04 Data Selection

05 Catastrophic Forgetting

Problems of OnDevice Training

01

Limited Memory

02

Limited Calculation capacity

03

Limited Energie

04

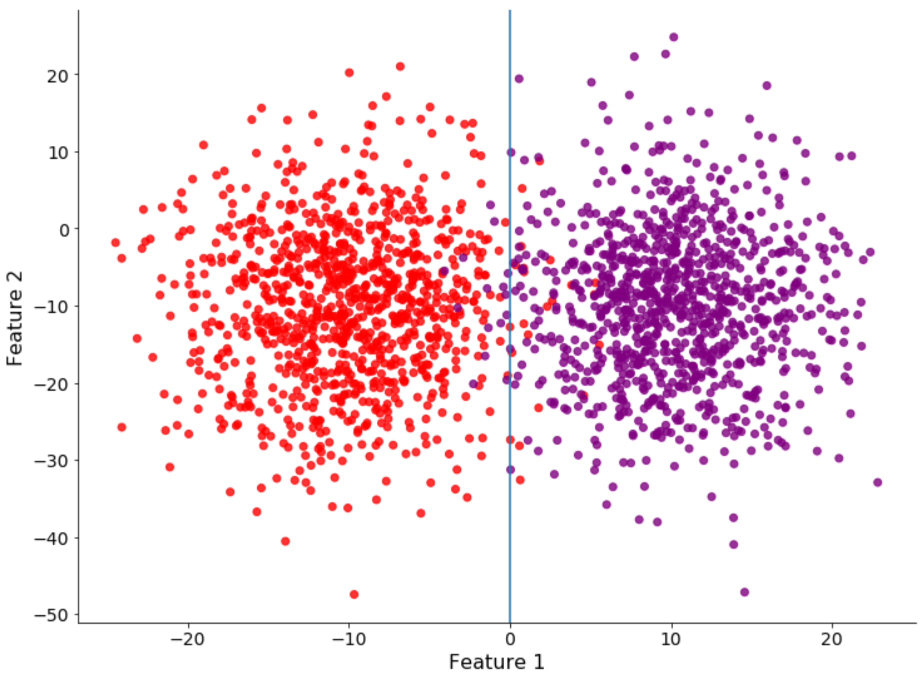
Data Selection

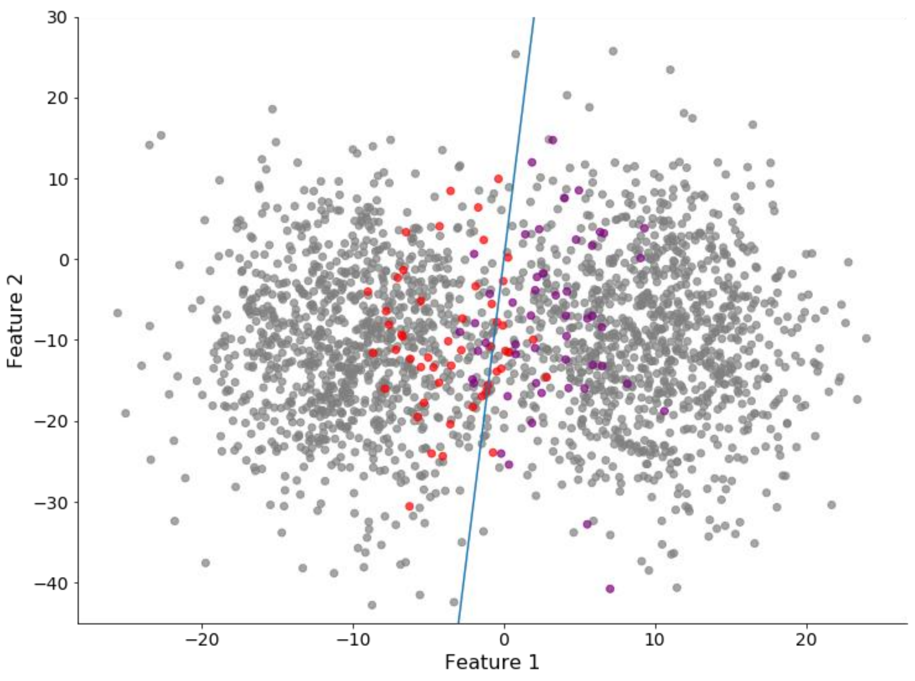
05

Catastrophic Forgetting

Data Point Selection

Which data use to train?
How to label Data on Device?
What to do with „old data“? [2]

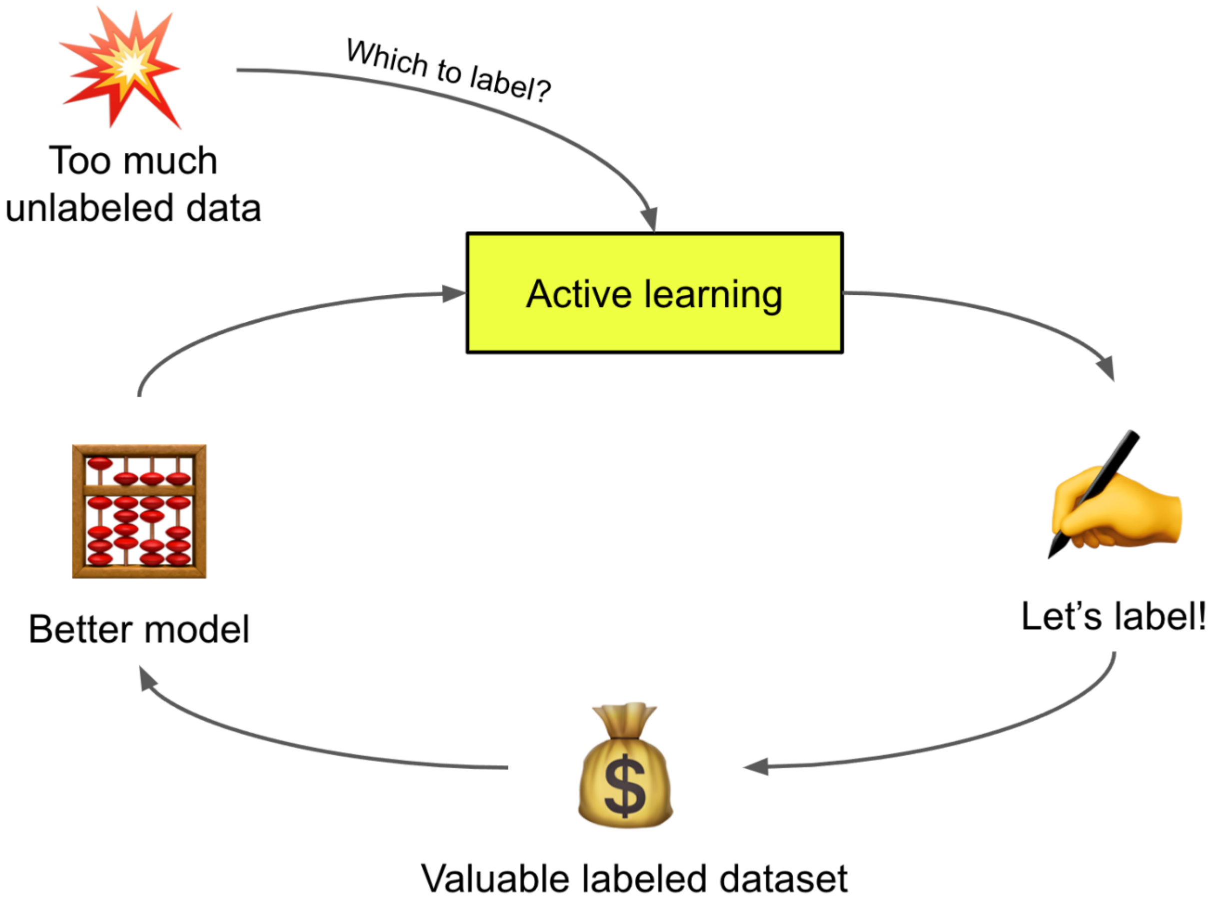




Active Learning

Hahn Schickard

1.Least Confidence: difference between the most confident prediction and 100% confidence
2.Margin of Confidence: difference between the top two most confident predictions
3.Ratio of Confidence: ratio between the top two most confident predictions
4.Entropy: difference between all predictions, as defined by information theory [3]



Problems of OnDevice Training

Hahn Schickard

01 Limited Memory

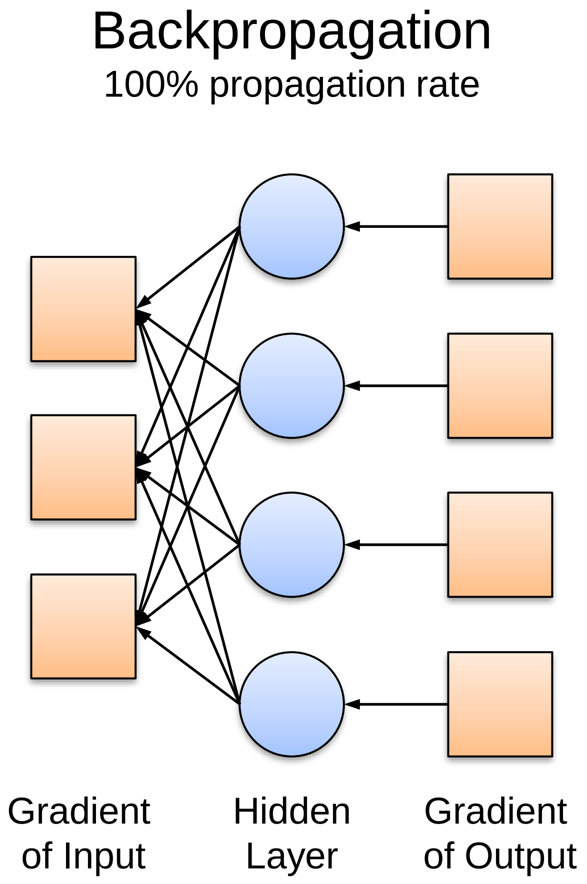
02 Limited Calculation capacity

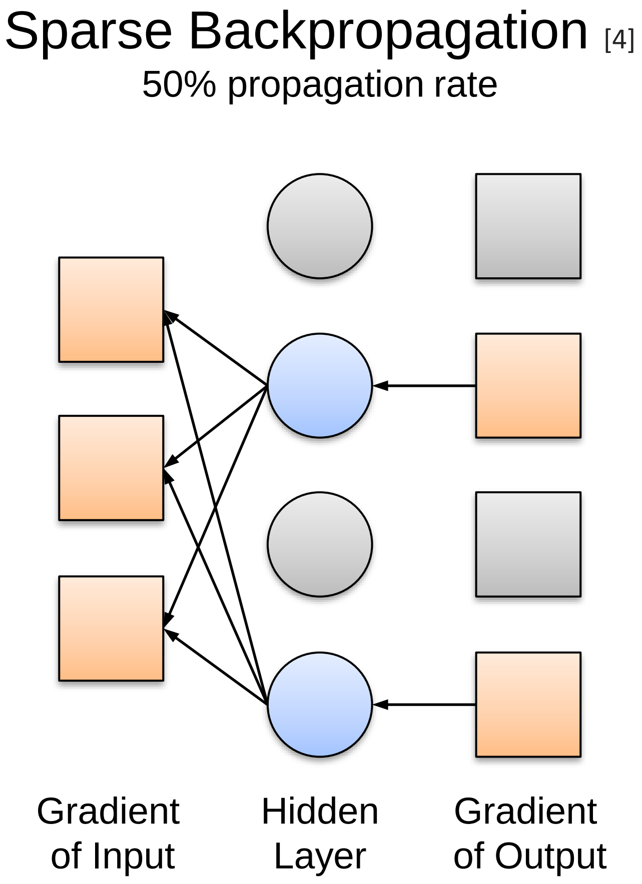
03 Limited Energie

04 Data Selection

05 Catastrophic Forgetting

Train more efficient

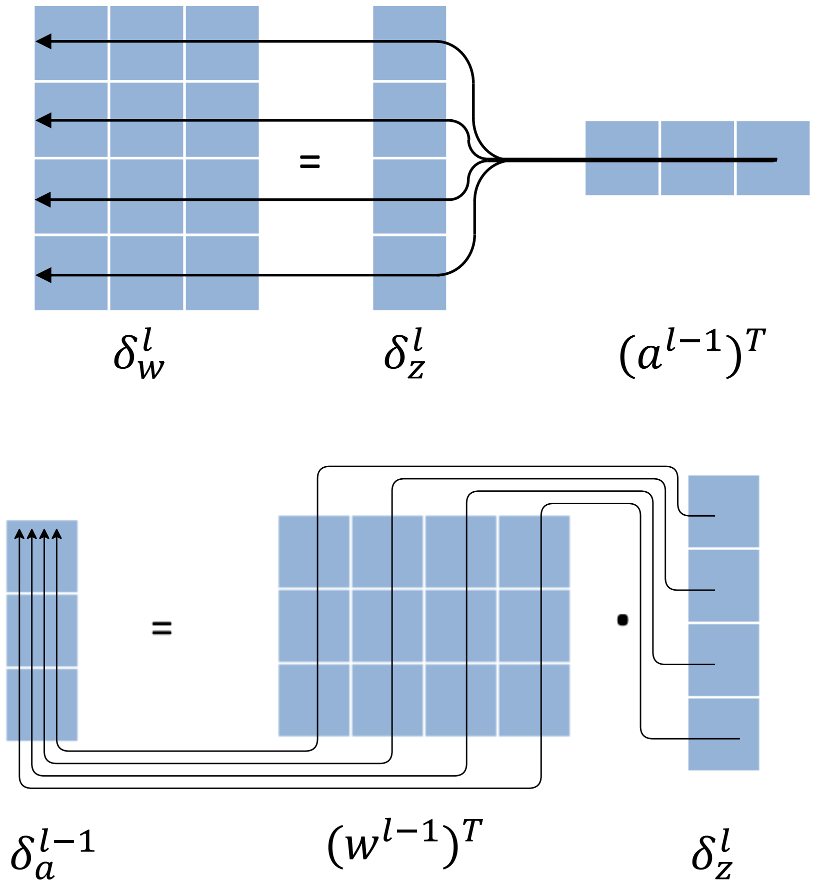




Train more efficient

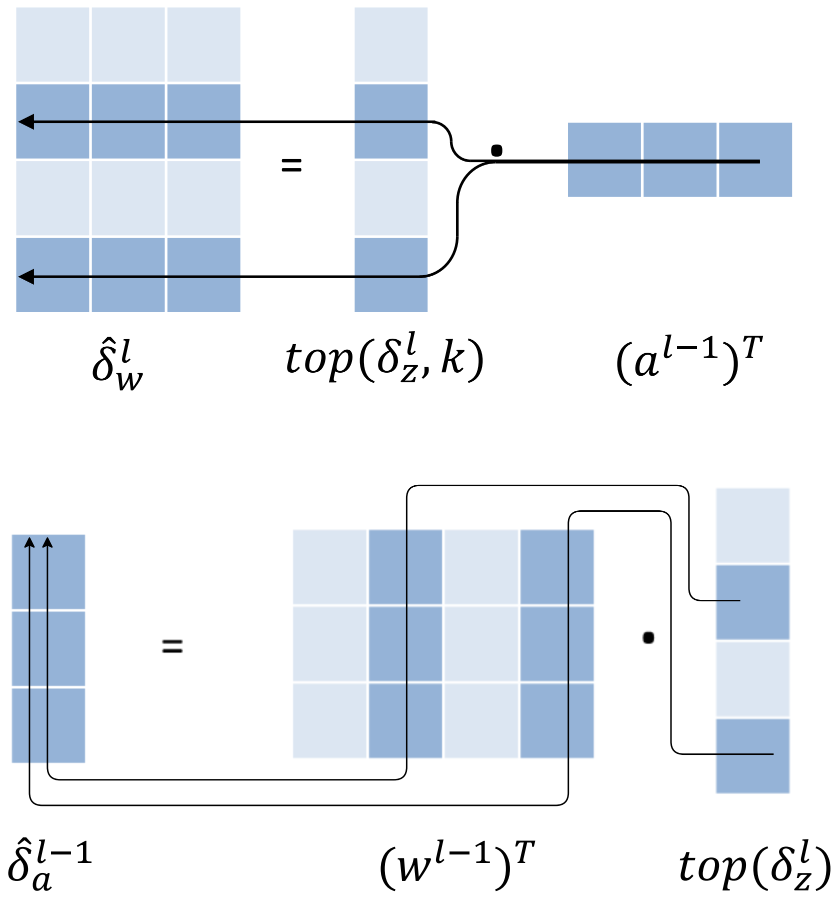
Backpropagation

100% propagation rate



Sparse Backpropagation

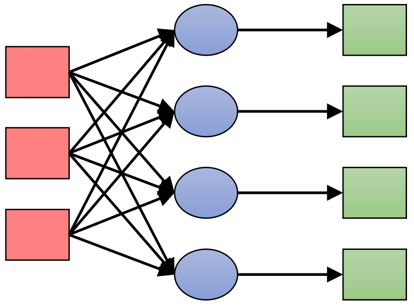
50% propagation rate



Train more efficient

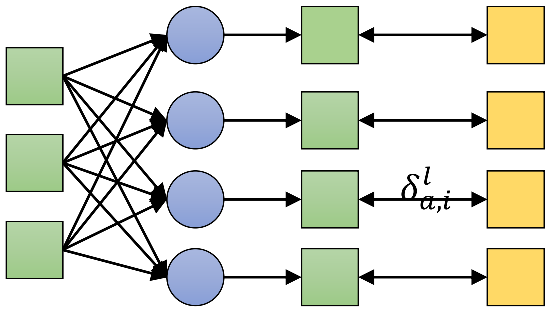
1. Forward Propagation

$$z^l = W^l \cdot a^l + b^l$$
$$a^{l+1} = f(z^l)$$



2. Sum magnitudes of local error

$$Y^l = \sum_{i=1}^{N^l} |\delta_{a,i}^l|$$



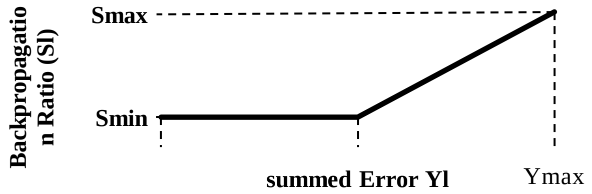
3. Decide to train Datapoint

$$D^L = \left(D_{min} + \alpha^L * \frac{(D_{\max} - D_{min})}{\alpha_{max}^L} \right) * \beta^L$$

4. Calculate k

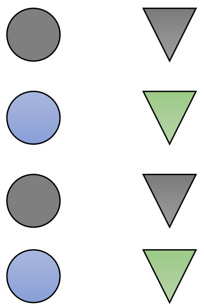
$$S^l := \left(S_{min} + Y^l \cdot \frac{S_{max} - S_{min}}{Y_{max}^l} \right) \cdot \zeta^{L-l}$$

$$k^l = S^l \cdot N^l$$



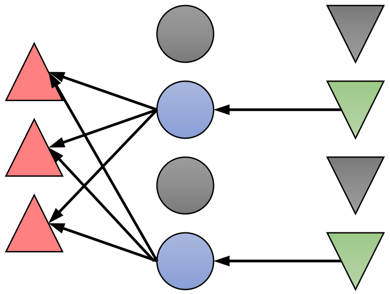
5. Get Top k

(Top k = 2)



6. Sparse back Propagation

(Top k = 2)



Hahn-Schickard - Marcus Rüb- Marcus.rueb@hahn-schickard.de

Problems of OnDevice Training

01 Limited Memory

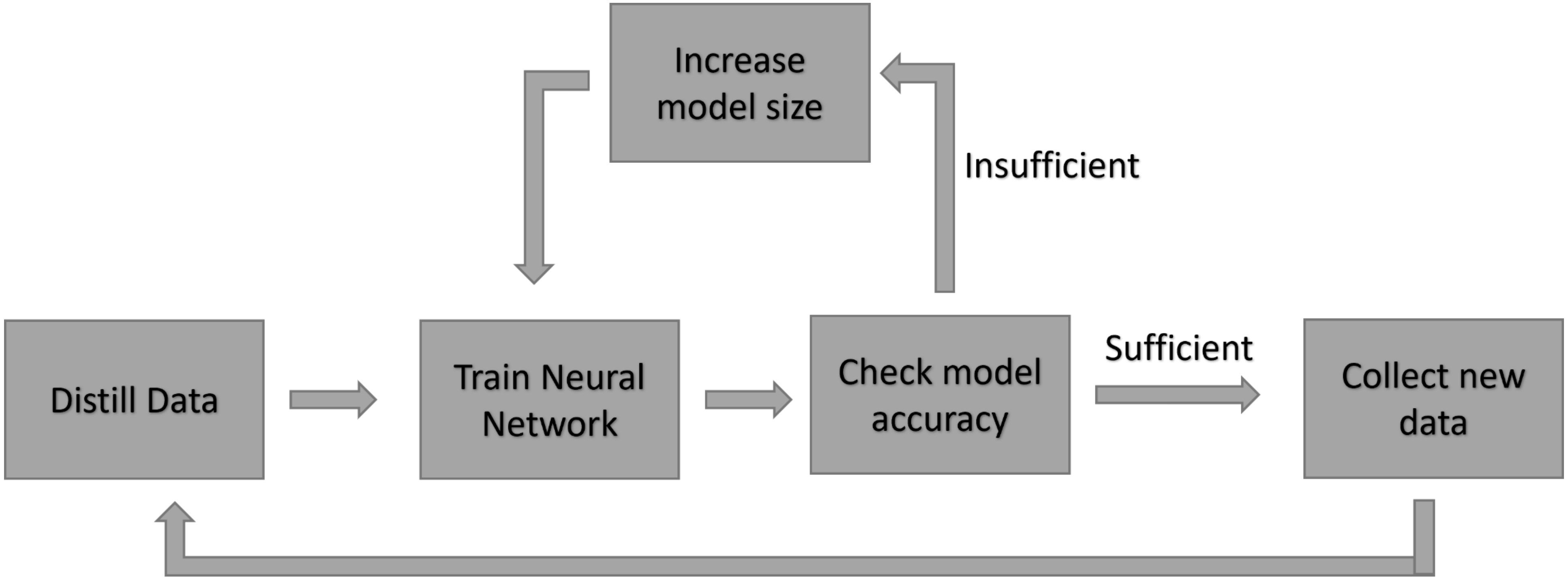
02 Limited Calculation capacity

03 Limited Energie

04 Data Selection

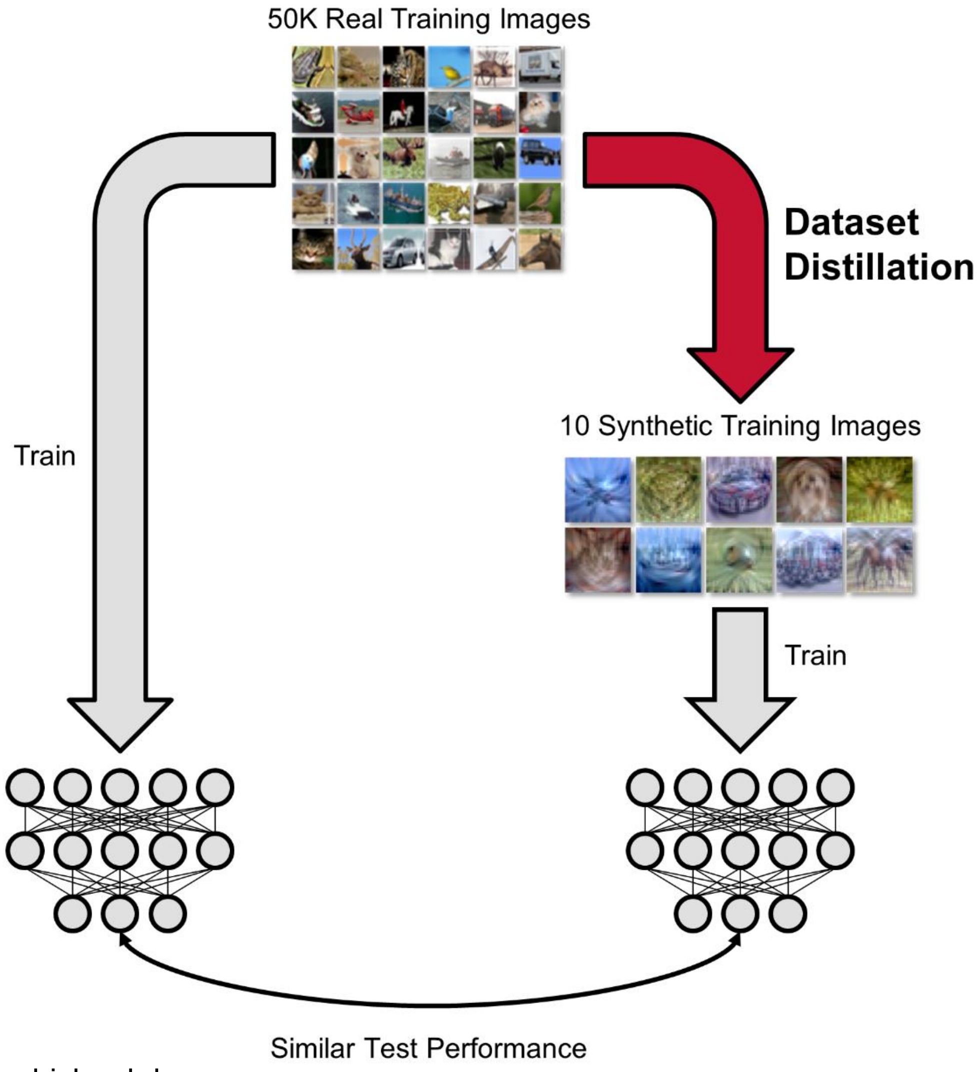
05 Catastrophic Forgetting

Proposed Workflow



[5]

What is Dataset distillation



Distill Dataset

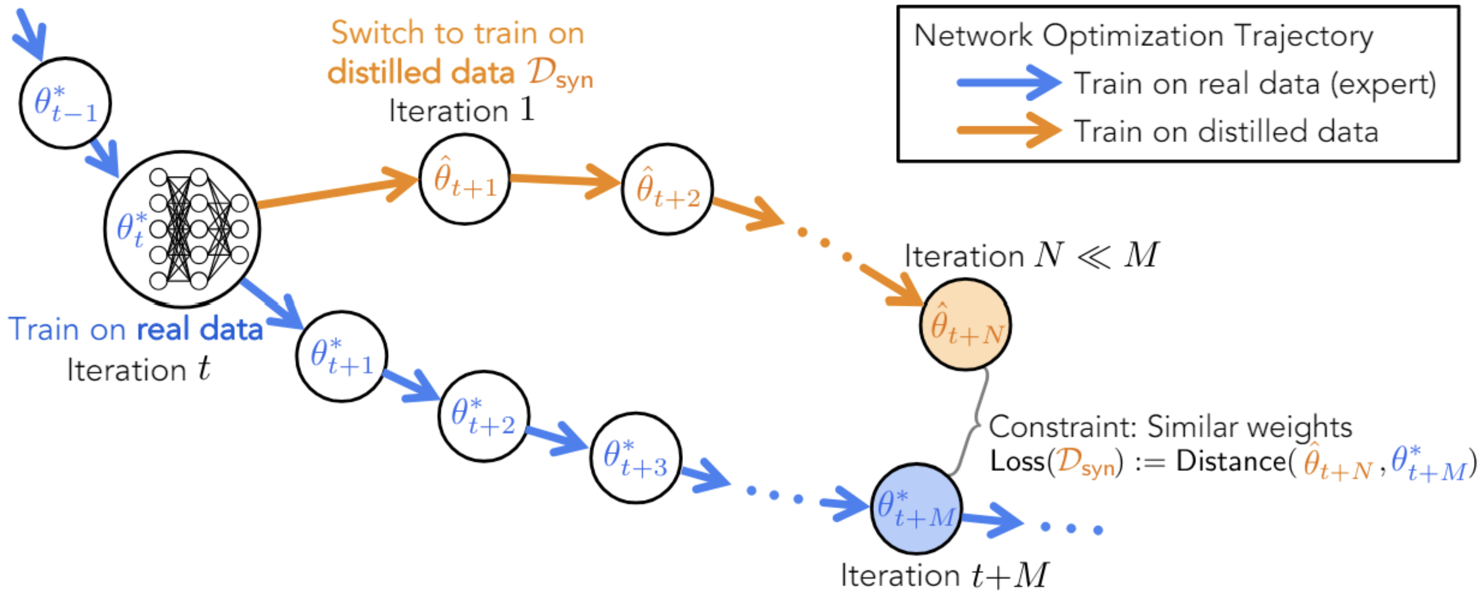
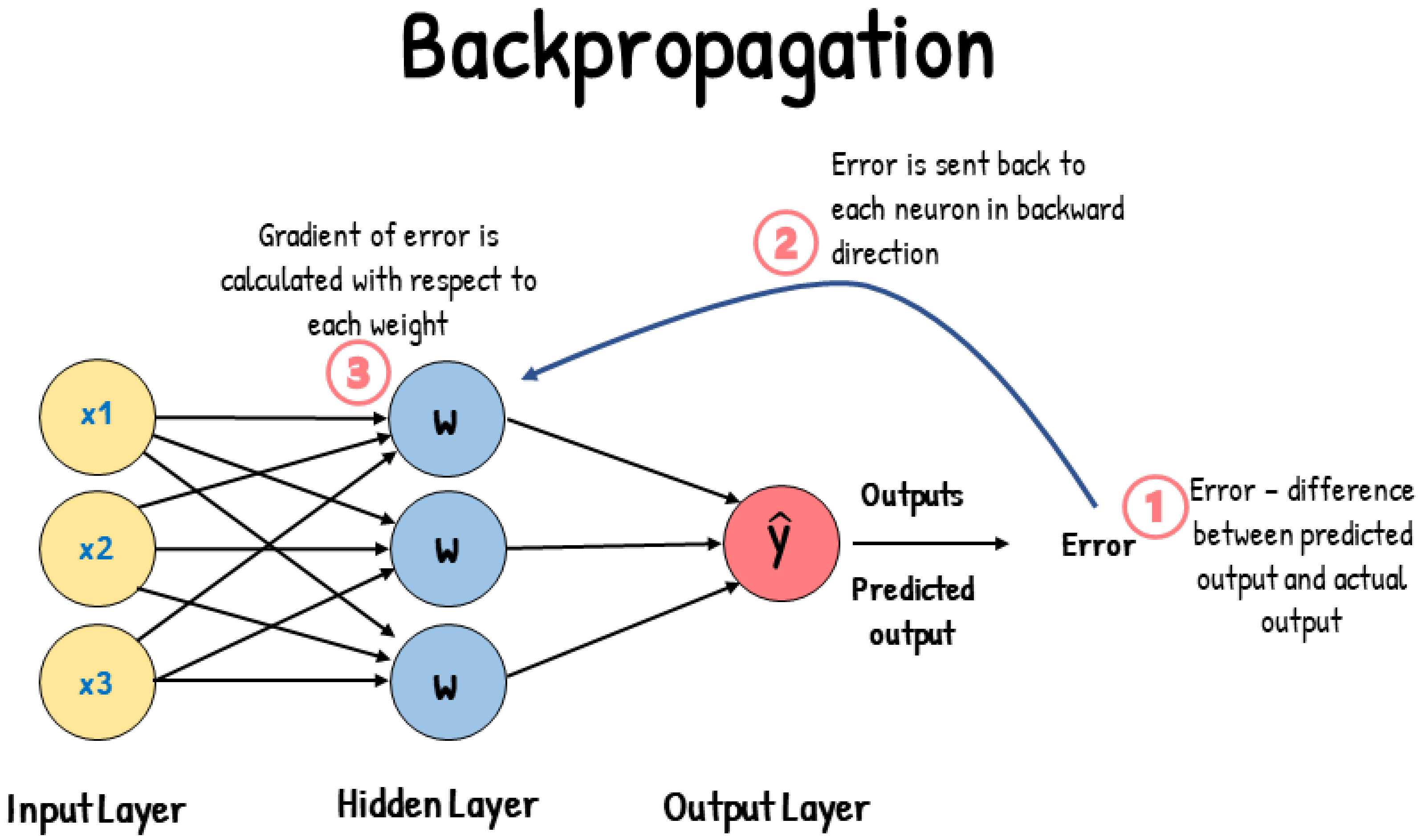


Figure 3. We perform long-range parameter matching between training on distilled synthetic data and training on real data. Starting from the same initial parameters, we train distilled data $\mathcal{D}_{\text{syn}}$ such that N training steps on them match the same result (in parameter space) from much more M steps on real data.

Train Neural Network



Check Model Accuracy & Increase Model size

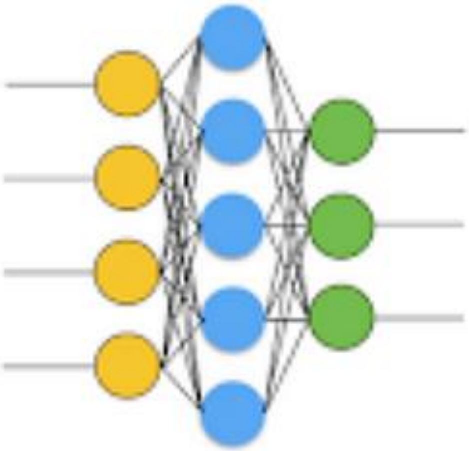
Model accuracy is typically measured using an accuracy score, which is defined as the proportion of correct predictions out of the total number of predictions. If we denote the number of correct predictions as C_{correct} and the total number of predictions as N_{total}, the accuracy score A can be calculated as follows:

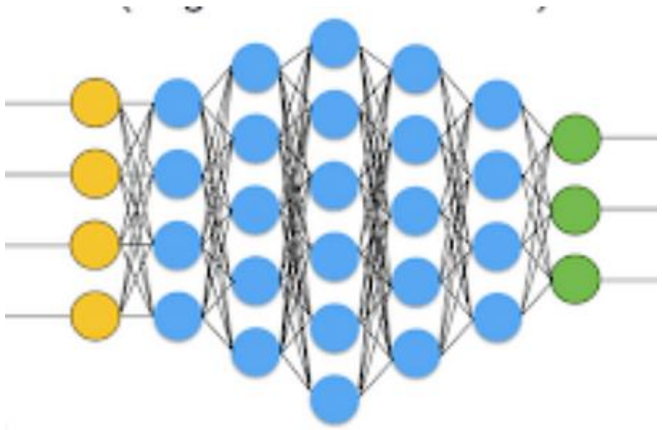
$$A = \frac{C_{\text{correct}}}{N_{\text{total}}}$$

Let's denote our original model's architecture as $\mathcal{M}$ and its accuracy score as A. If A doesn't meet the predefined accuracy standard, A_{standard}, we proceed with the enlargement of the model, creating a new model architecture, $\mathcal{M}'$. This adjustment can be represented as follows:

$$\mathcal{M}' = \begin{cases} \mathcal{M} + \Delta\mathcal{M} & \text{if } A < A_{\text{standard}} \\ \mathcal{M} & \text{otherwise} \end{cases}$$

where $\Delta\mathcal{M}$ represents the increase in the model's complexity. This increase can be in the form of additional layers, more





References

[1] Marcus Rüb, Prof. Dr. Axel Sikora, A Practical View on Training Neural Networks in the Edge, IFAC-PapersOnLine, Volume 55, Issue 4, 2022, Pages 272-279, ISSN 2405-8963
[2] A Comparative Survey of Deep Active Learning.
Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B. Chan,
arXiv:2203.13450
[3] https://towardsdatascience.com/uncertainty-sampling-cheatsheet-ec57bc067c0b
[4] Rüb, Marcus & Maier, Daniel & Mueller-Gritschneder, Daniel & Sikora, Axel. (2023).
TinyProp -- Adaptive Sparse Backpropagation for Efficient TinyML On-device Learning.
GITHUB: https://github.com/r1marcus/TinyProp
[5] Rüb, Mueller-Gritschneder, Sikora. (2024).
A Continual and Incremental Learning Approach for TinyML On-device Training Using Dataset Distillation and Model Size Adaption.

Thank you

Marcus Rüb
Wissenschaftlicher Mitarbeiter

Hahn-Schickard-Gesellschaft für angewandte Forschung e.V.

Wilhelm-Schickard-Str. 10,
78052 Villingen-Schwenningen
Telefon: +49 7721 943-180
Marcus.Rueb@Hahn-Schickard.de
www.Hahn-Schickard.de