Bio-plausible learning rules for KANs and SNNs

By: Irene Acero, Hojjat Mokhtharabadi, Tomas Berjaga, Andrew Tien Anh Vu, Dennis Layh, Harshit Pateria, Zeynab Razzaghpanah, Reagan Tshiangomba
TAs: Patricia Rubisch, Aakash Agrawal

Branched Bioprop Brigade

neuromatch academy

Questions

- How different biologically-plausible learning rules behave across different types of neural architectures (MLPs, KANs, SNNs)?

Questions

- How different biologically-plausible learning rules behave across different types of neural architectures (MLPs, KANs, SNNs)?

- What learning rules would be suitable for what neural architectures in terms of performance and learning?

Questions

- How different biologically-plausible learning rules behave across different types of neural architectures (MLPs, KANs, SNNs)?

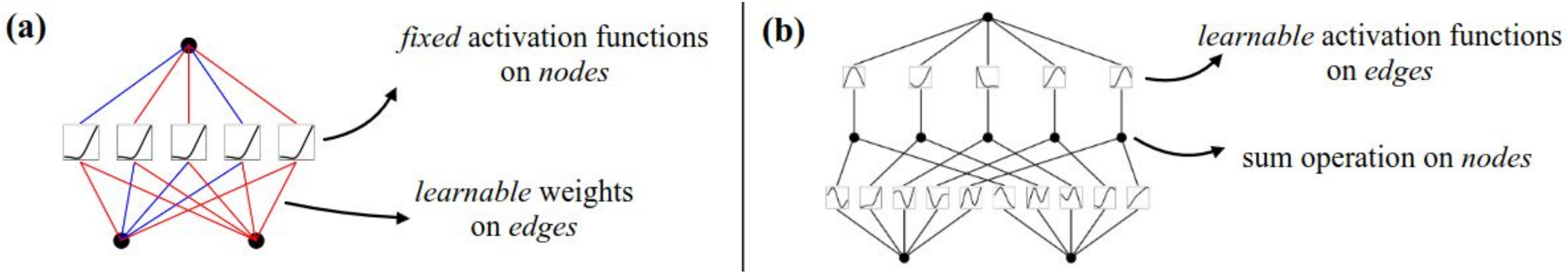
- What learning rules would be suitable for what neural architectures in terms of performance and learning?

- How representations of learning rules are close to each other and how they are clustered in a low dimensional space?

Kolmogorov Arnold Networks

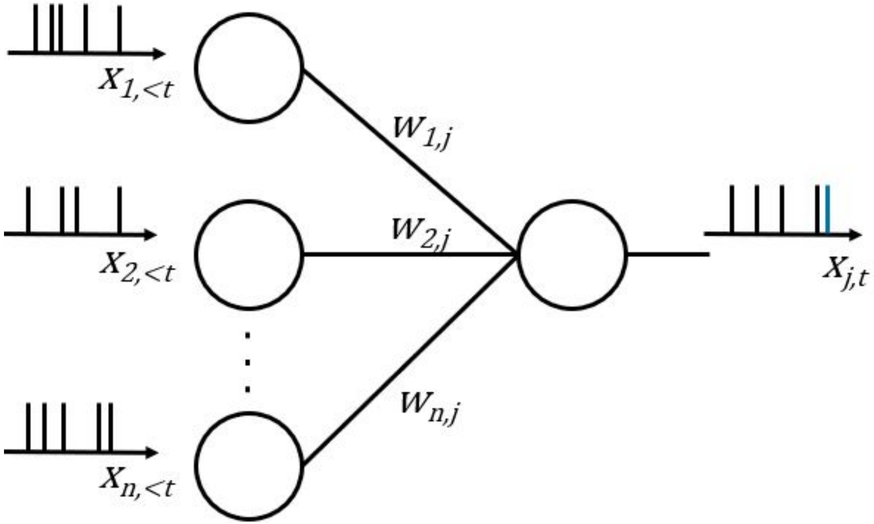
- Based on Kolmogorov-Arnold representation theorem
- Learnable nonlinear activation functions on edges
- Sum operation on nodes
- Forward pass for layer l $\rightarrow$ $x_{l+1,j} = \sum_{i=1}^{n_l} \tilde{x}_{l,j,i} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i})$

MLP KAN



Spiking Neural Networks

- Mimic Biological Neural Spiking Networks.
- Inputs encoded as spikes
- IF model represents neuronal population

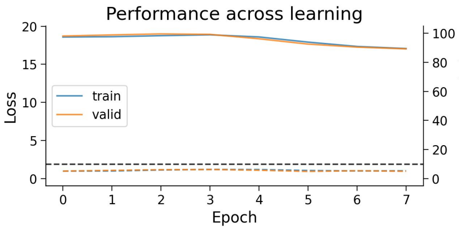


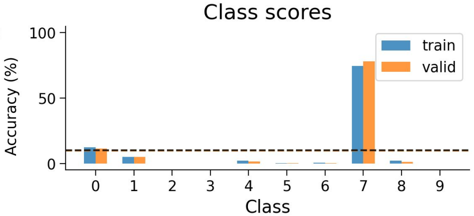
Methodology

- **Dataset:** MNIST
- Implement a simple **KAN**, **MLP** and **SNN** neural network (2 linear layers) with different biologically plausible learning rules
- Record the performance metrics during training and evaluation (loss, accuracy), bias (cosine similarity) and variance (SNR) for gradient of weights
- Compare the representations from the network's layer using dissimilarity matrices and project them into low dimensional space to see clusters

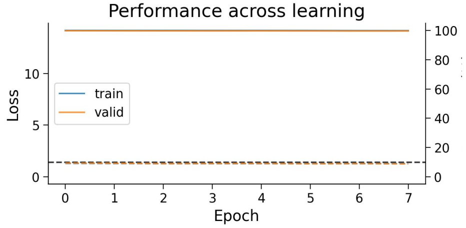
Learning performance across learning rules in MLP vs SNN

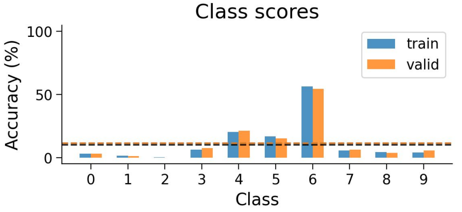
MLP using Hebbian



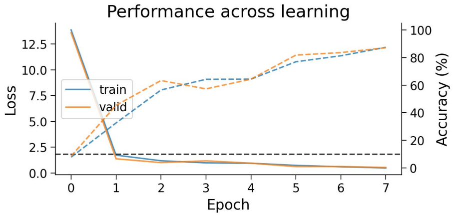


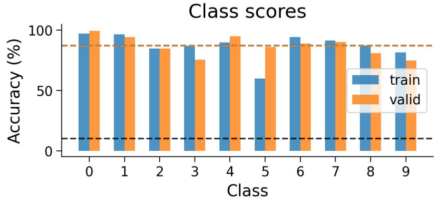
SNN using Hebbian



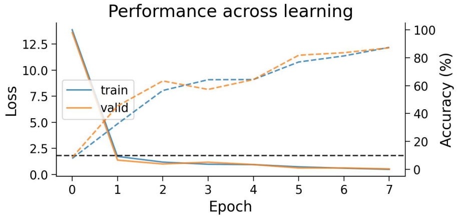


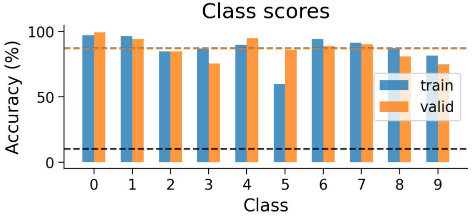
MLP using Backpropagation





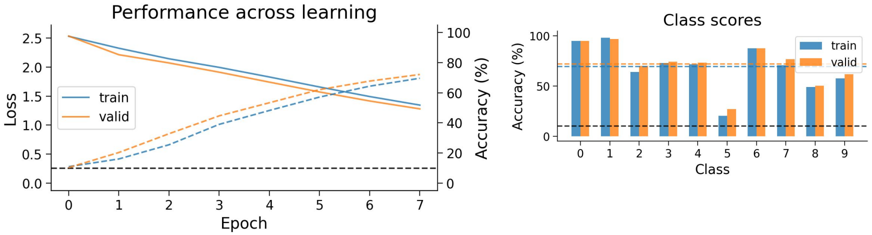
SNN using Backpropagation



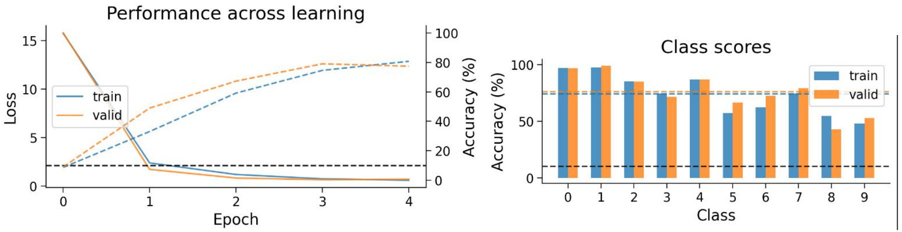


Learning performance across learning rules in MLP vs SNN

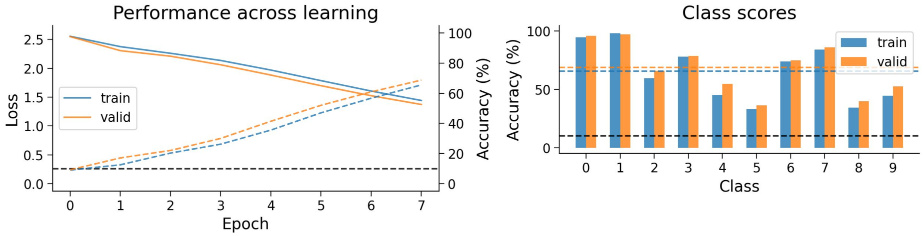
MLP using Kollen-Pollak



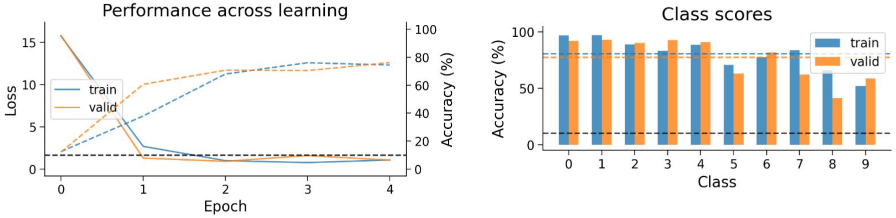
SNN using Kollen-Pollak



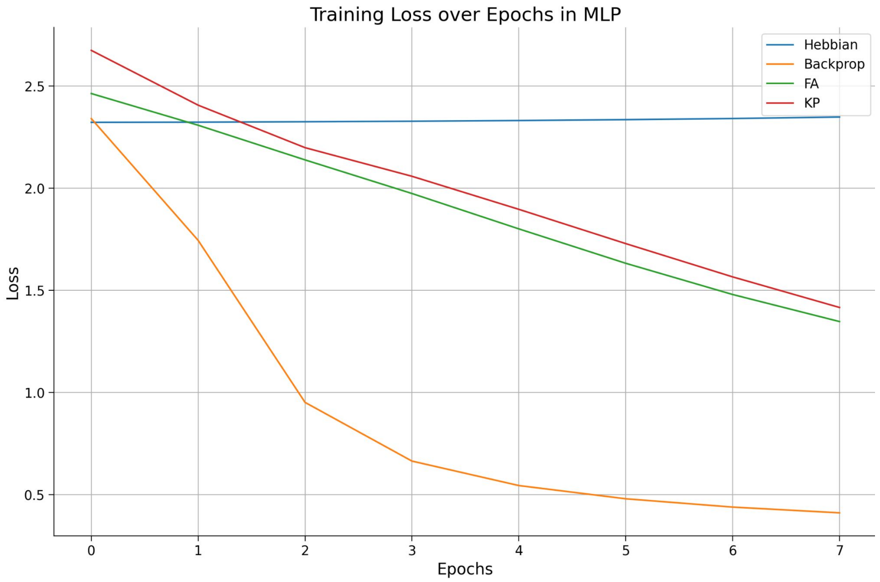
MLP using Feedback Alignment

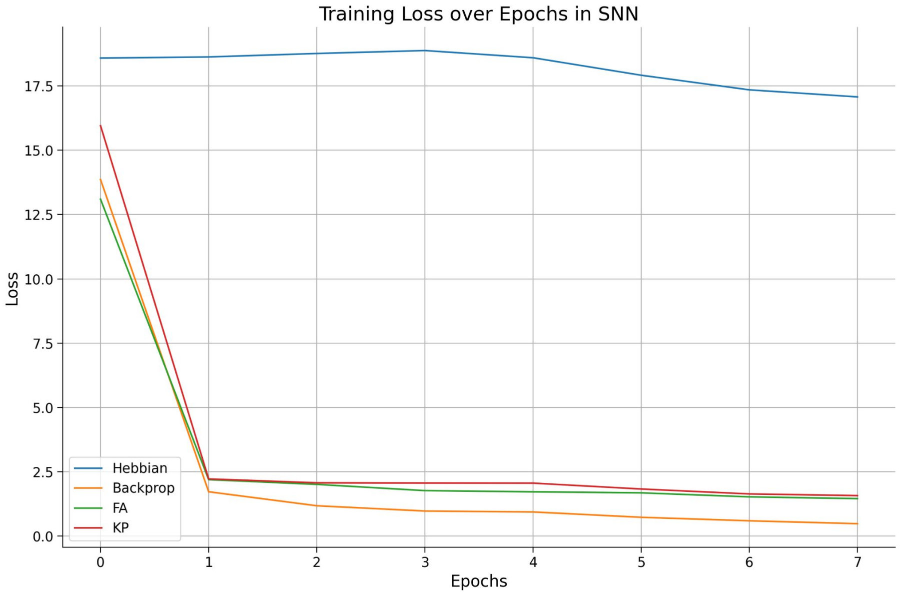


SNN using Feedback Alignment

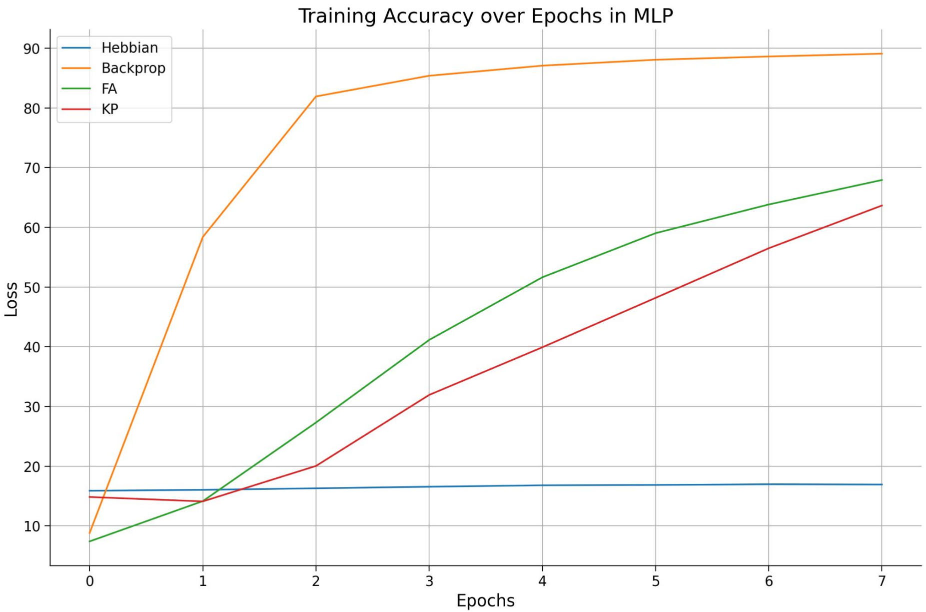


Result: training loss across learning rules in MLP vs SNN

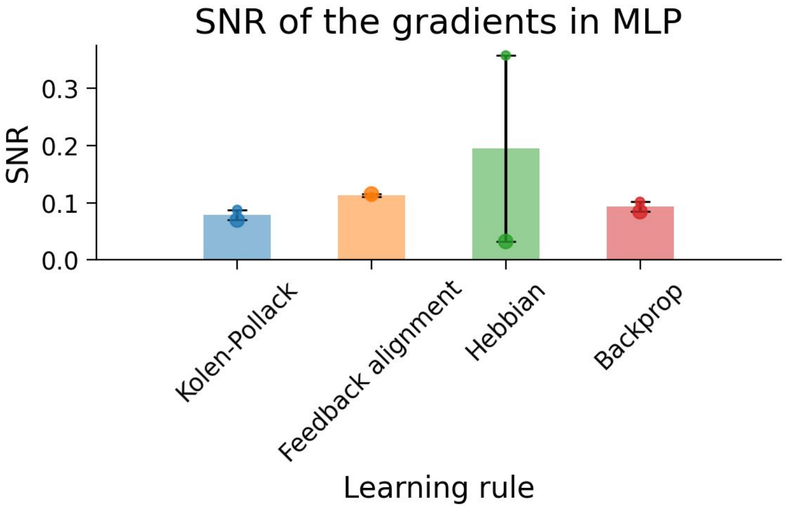


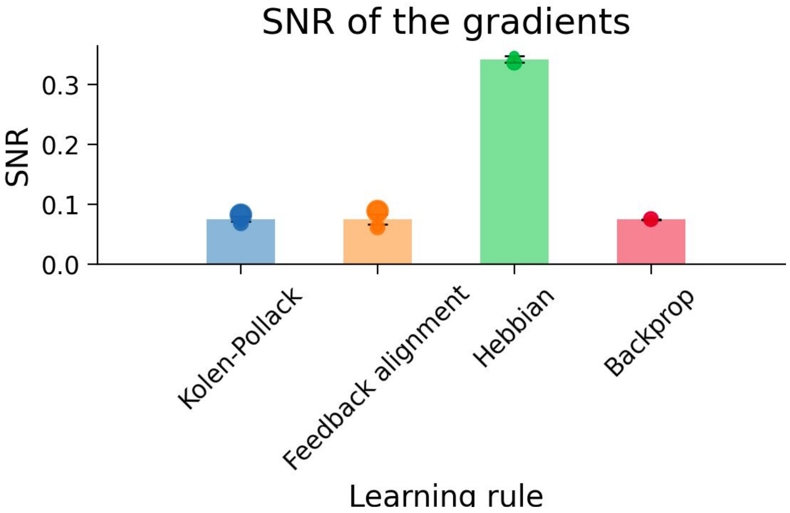


Result: training accuracy across learning rules in MLP vs SNN

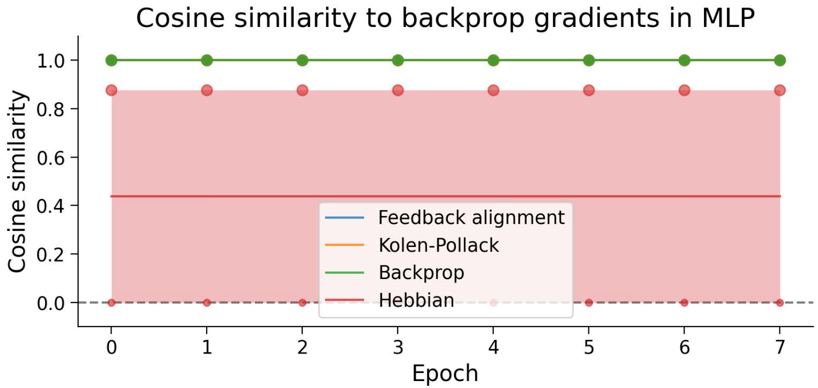


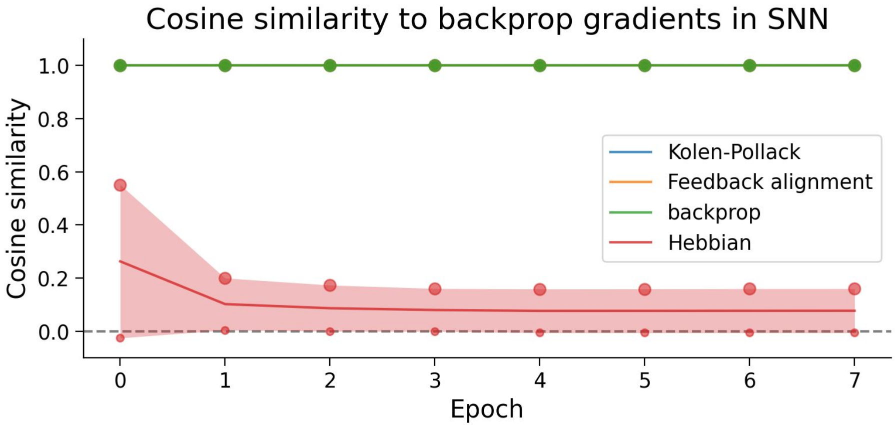
Result: Gradient variance across learning rules



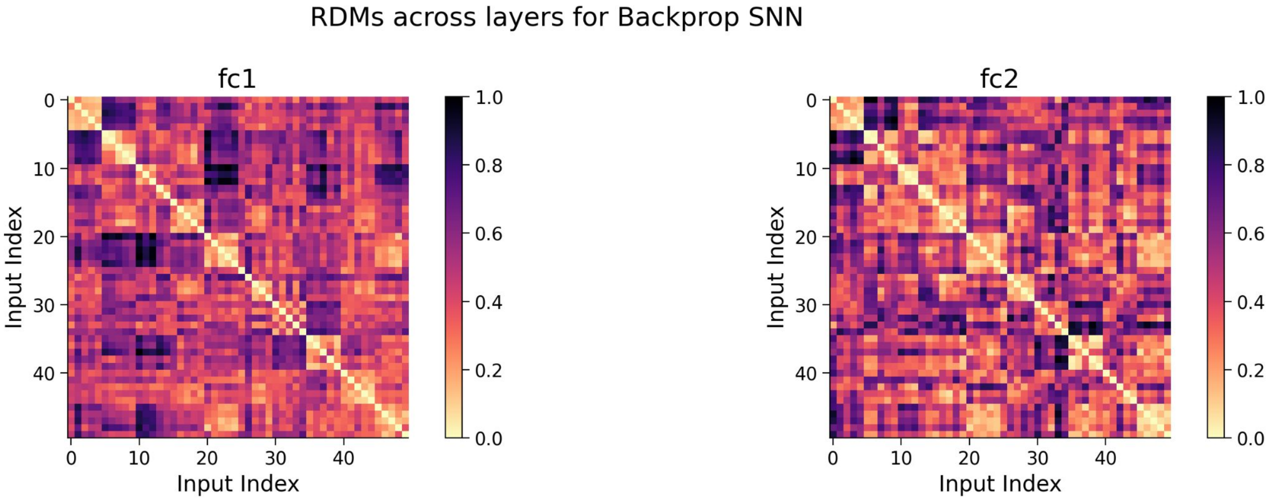


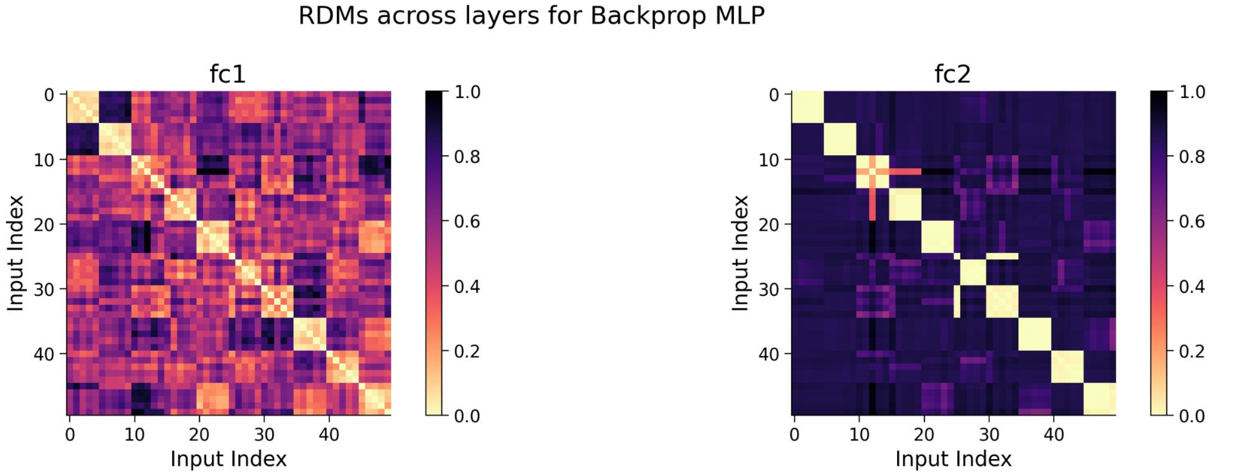
Result: Gradient bias with respect to error backpropagation



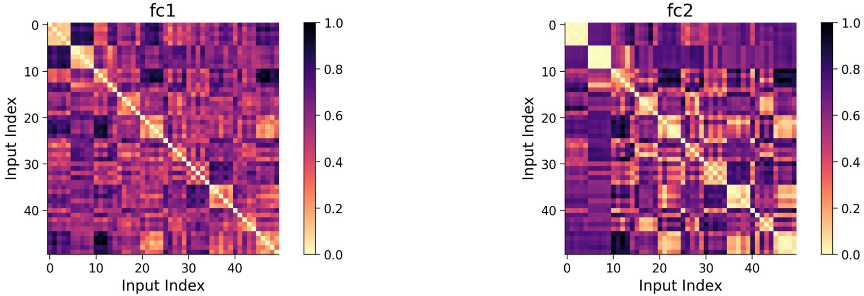


Result: Compare representational dissimilarity matrices

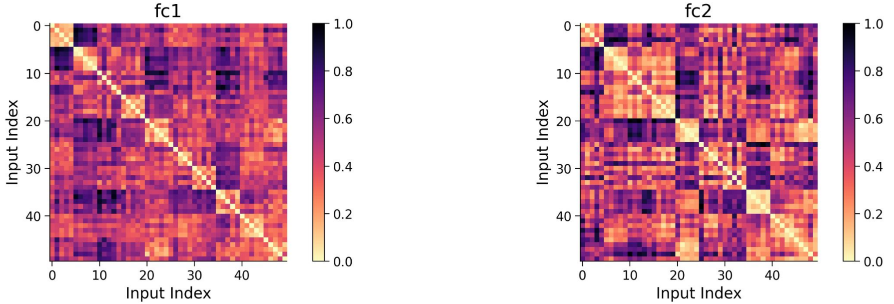




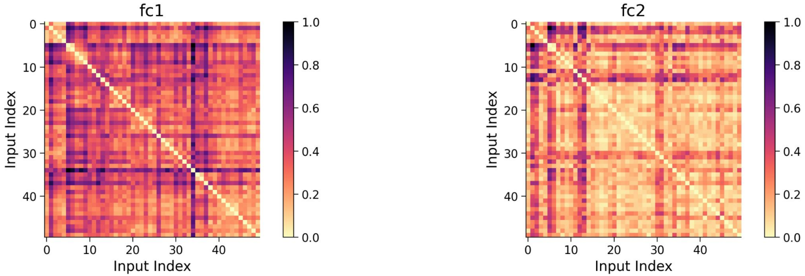
RDMs across layers for Feedback alignment MLP



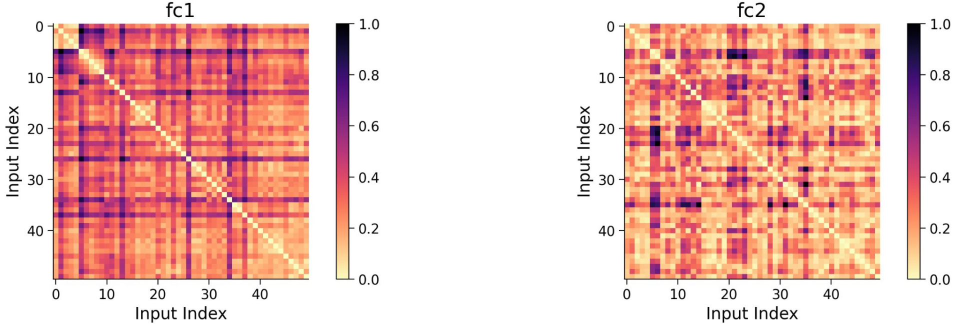
RDMs across layers for Feedback alignment SNN



RDMs across layers for Hebbian MLP

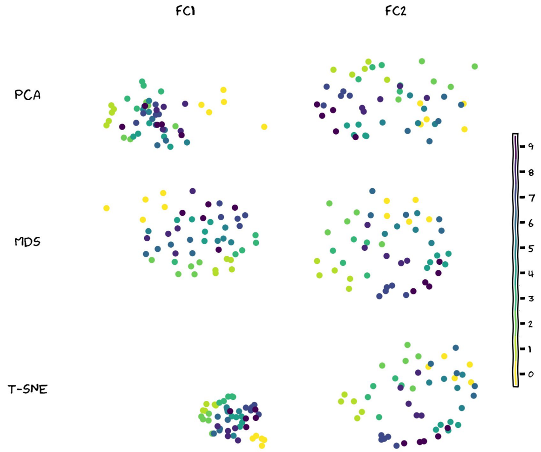


RDMs across layers for Hebbian SNN

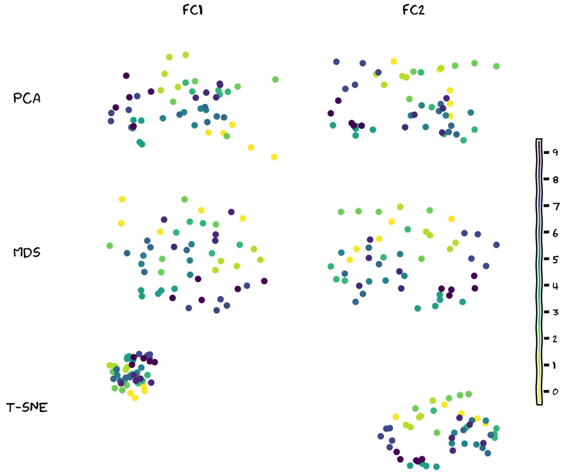


Result: Representational geometry of learning rules in SNN

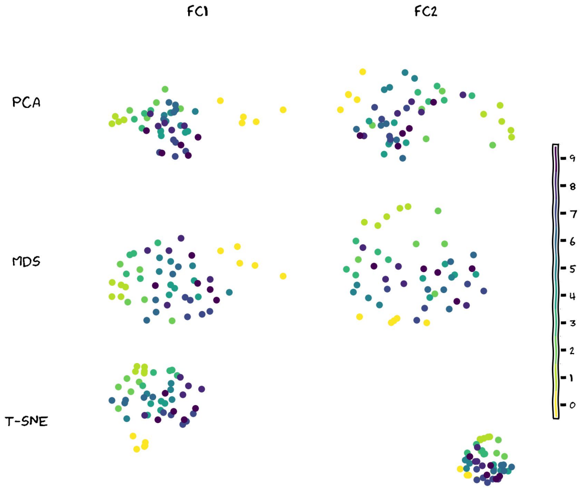
Backprop



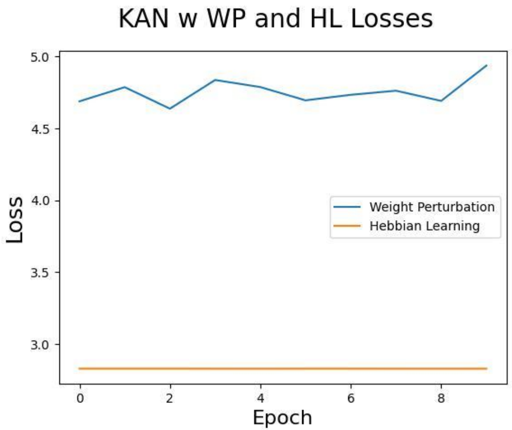
Kolen-Pollack



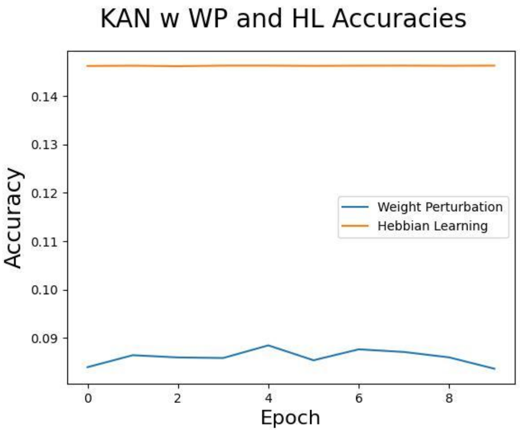
Feedback alignment



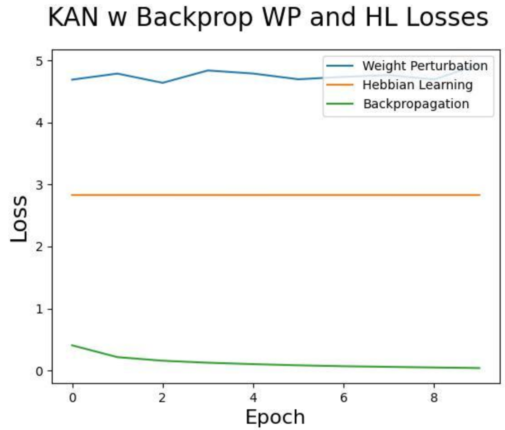
KAN Results:



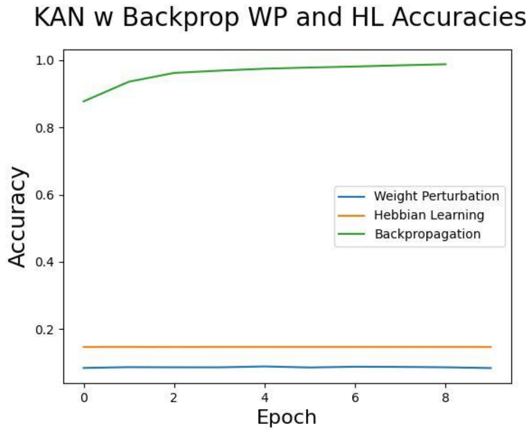
KAN w WP and HL Losses
5.0
4.5
4.0
3.5
3.0
Loss
0
2
4
6
8
Epoch
Weight Perturbation
Hebbian Learning



KAN w WP and HL Accuracies
0.14
0.13
0.12
0.11
0.10
0.09
Accuracy
0
2
4
6
8
Epoch
Weight Perturbation
Hebbian Learning



KAN w Backprop WP and HL Losses
5
4
3
2
1
0
Loss
0
2
4
6
8
Epoch
Weight Perturbation
Hebbian Learning
Backpropagation



KAN w Backprop WP and HL Accuracies
1.0
0.8
0.6
0.4
0.2
Accuracy
0
2
4
6
8
Epoch
Weight Perturbation
Hebbian Learning
Backpropagation

Conclusions

- Feedback alignment and Kolen-Pollack perform well on SNNs and MLPs (~70-80% accuracy) but are not as effective as backpropagation (> 90% accuracy)

- Hebbian learning failed to perform above chance in any of the tested architectures

- Only backprop achieved above chance performance for the KAN network