南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

生物医学工程系
Department of Biomedical Engineering

NEURAL COMPUTING CONTROL
神经计算与控制实验室
NCC lab

Hebbian Deep Learning Without Feedback

Author: Adrien Journé, Hector Garcia Rodriguez, Qinghai Guo, Timoleon Moraitis*

Presenter: Ziyuan Ye

ICLR 2023 Notable-top-25%

Content

- Introduction to Timoleon Moraitis
- Background
- SoftHebb
- Experiments
- Discussion

Content

- Introduction to Timoleon Moraitis
- Background
- SoftHebb
- Experiments
- Discussion

Information about Timoleon Moraitis

Timoleon Moraitis

其他姓名 ▸

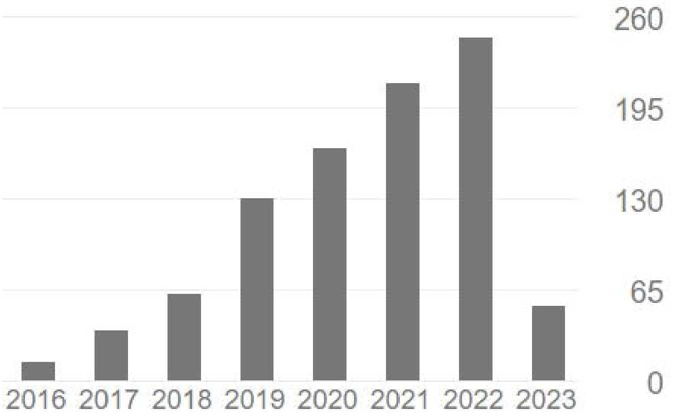
关注

Group Leader, Huawei Technologies - Zurich Research Center
在 huawei.com 的电子邮件经过验证 - 首页

Neuro-AI & Biologically-pla…   Neuromorphic Computing   Machine Learning   Neuroscience
AI accelerators

引用次数

	总计	2018 年至今
引用	946	876
h 指数	10	10
i10 指数	10	10



Neuromorphic computing with multi-memristive synapses I Boybat, M Le Gallo, SR Nandakumar, T Moraitis, T Parnell, T Tuma, ... Nature communications 9 (1), 2514	555	2018
Bridging the gap: a reticulo-propriospinal detour bypassing an incomplete spinal cord injury L Filli, AK Engmann, B Zörner, O Weinmann, T Moraitis, M Gullo, ... Journal of Neuroscience 34 (40), 13399-13410	159	2014
A bidirectional brain-machine interface featuring a neuromorphic hardware decoder F Boi*, T Moraitis*, V De Feo*, F Diotalevi, C Bartolozzi, G Indiveri, A Vato Frontiers in neuroscience 10, 563	64	2016
Phase-change memtransistive synapses for mixed-plasticity neural computations SG Sarwat, B Kersting, T Moraitis, VP Jonnalagadda, A Sebastian Nature Nanotechnology 17 (5), 507-513	25 *	2022
Stochastic weight updates in phase-change memory-based synapses and their influence on artificial neural networks I Boybat, M Le Gallo, T Moraitis, Y Leblebici, A Sebastian, E Eleftheriou 2017 13th Conference on Ph. D. Research in Microelectronics and Electronics …	22	2017

Content

- Introduction to Timoleon Moraitis
- Background
- SoftHebb
- Experiments
- Discussion

Neuromorphic Computing Hardware (NCH)

Key components and working mechanism of NCH:

- **Neuron Models:** Basic computational unit that mimics biological neurons
  - Examples:
    - Leaky Integrate-and-Fire (LIF)
    - Izhikevich
  - These models simulate the process by which a neuron produces a voltage change after receiving an input signal and generates an output signal (action potential) when a certain threshold is exceeded.

- **Synapse:** A synapse is a connection between neurons that is responsible for transmitting signals from one neuron to another. In neuromorphic hardware, synapses can be realized with tunable resistors
  - Examples:
    - Patch diodes (膜片二极管)
    - Memristors (忆阻器)
  - These resistors can be tuned according to the activity between neurons for learning and memory functions.

Neuromorphic Computing Hardware (NCH)

Key components and working mechanism of NCH:

- **Learning rules:** Neuromorphic computing hardware adapts and learns using learning rules based on local information
  - Examples:
    - Backpropagation (BP)
    - Competitive learning
    - Hebbian learning: fire together, wire together
    - Spike-timing-dependent plasticity (STDP)
  - These learning rule simulate the process by which a neuron produces a voltage change after receiving an input signal and generates an output signal (action potential) when a certain threshold is exceeded.

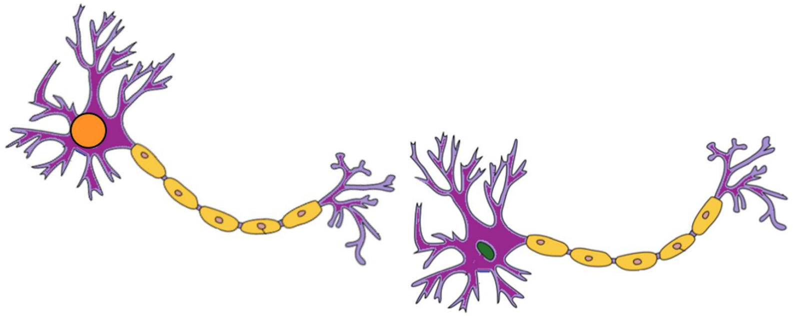
BP & Its Limitations: Weight Transport

Model & Loss	Backpropagation	Details of Backpropagation	$\boldsymbol{\Delta^n}$
$z^N = W^N a^{N-1} + b^N$ $a^N = \sigma(z^N)$ $L = \frac{1}{2}\|\|a^N - y\|\|_2^2$	$W^n = W^n - \eta \frac{\partial L}{\partial W^n}$	$\frac{\partial L}{\partial W^n} = \frac{\partial L}{\partial z^n}\frac{\partial z^n}{\partial W^n} = \frac{\partial L}{\partial a^n}\frac{\partial a^n}{\partial z^n}\frac{\partial z^n}{\partial W^n} = (a^n - y)\odot\sigma'(z^n)\frac{\partial z^n}{\partial W^n} = \Delta^n \frac{\partial z^n}{\partial W^n} = \Delta^n (a^{n-1})^T$	$\Delta^n = \frac{\partial L}{\partial z^n} = \frac{\partial L}{\partial z^{n+1}}\frac{\partial z^{n+1}}{\partial z^n} = \Delta^{n+1}\frac{\partial z^{n+1}}{\partial z^n} = \Delta^{n+1}\frac{\partial z^{n+1}}{\partial z^n} = (W^{n+1})^T \Delta^{n+1} \odot \sigma'(z^n)$
	$b^n = b^n - \eta \frac{\partial L}{\partial b^n}$	$\frac{\partial L}{\partial b^n} = \frac{\partial L}{\partial z^n}\frac{\partial z^n}{\partial b^n} = \frac{\partial L}{\partial z^n} 1 = \Delta^n$	

Such weight transport is not possible in biology, as synapses are directional.

BP & Its Limitations: Non-local plasticity

BP cannot update each weight based only on the immediate activations of the two neurons that the weight connects



BP requires error signal

BP is Non-local in space and time

$\neq$

Locality is generally believed to govern biological synaptic plasticity

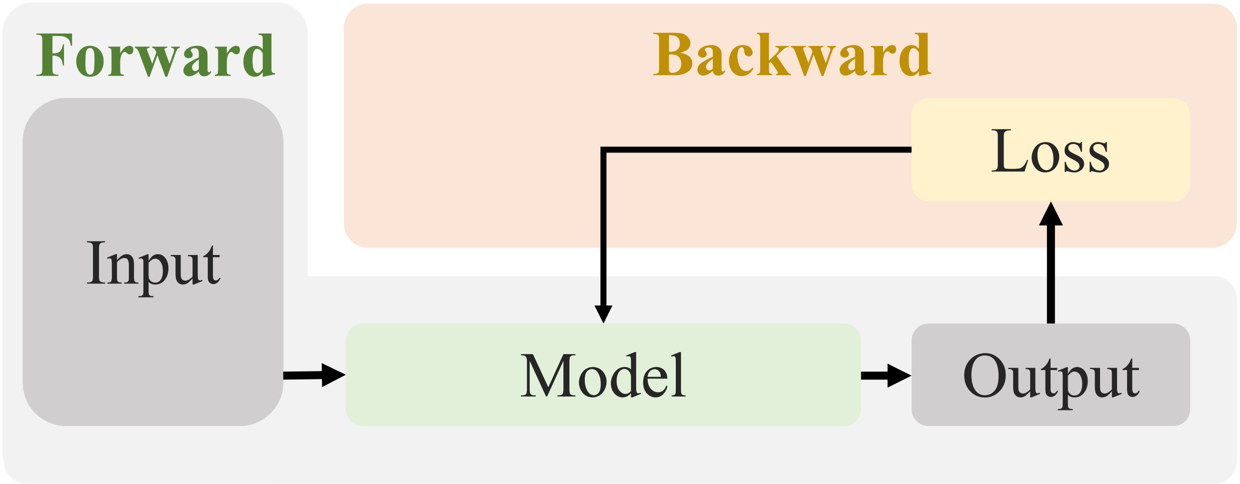
(Baldi et al. 2017)

BP is computational inefficiency

I. Forward-passing variables must be memorized
II. Additional backward signals must be computed and propagated

BP & Its Limitations: Update Locking

The error credited by BP to a synapse (weight) can only be computed after the information has propagated forward and then backward through the entire network.



Forward: Create a calculation graph to store the calculation process and intermediate results
Backward: Starting from the output of the calculation graph, the gradients are calculated and stored forward along each node in the graph

Update locking **Time**(forward) + **Time**(backward) $\xrightarrow{\text{determine}}$ **Weight updates**

Competition between Neurons

Neurons express competition through competitive inhibition

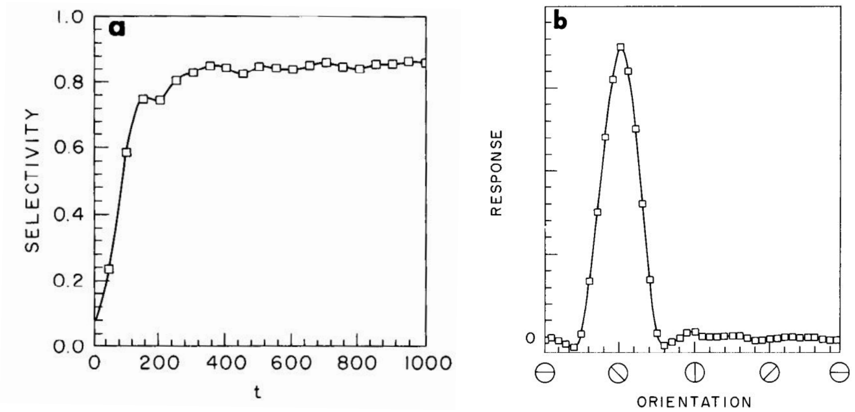
Competitive inhibition

When a group of neurons is strongly stimulated, they suppress the activity of neighboring neurons, gaining more resources and optimizing their own performance

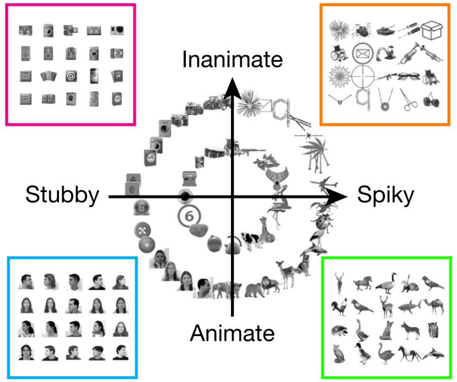
Winner-takes-all (WTA)

(Hebb, D. O. 1949)

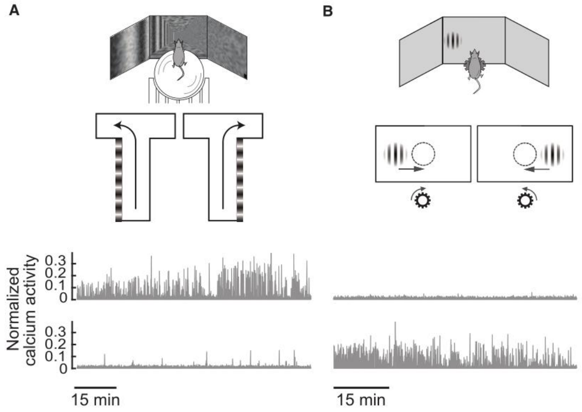
Important for Learning and Memory



(Bienenstock et al. 1982)



(Bao et al. 2020)



(Lee et al. 2022)

Hebbian Plasticity Rule & Its Variants

Key idea: Co-activated neurons connect to each other

Hebbian plasticity rule

$$\Delta\omega_{ik} = \eta \cdot y_k(x_i, \omega_{ik}) \cdot x_i$$ (Hebb, D. O. 1949)

$\boldsymbol{\Delta\omega_{ik}}$: the weight update vector from neuron $\boldsymbol{i}$ to neuron k
$\boldsymbol{y(x_i, \omega_{ik})}$: the post-synaptic activation of the neuron
$\boldsymbol{x_i}$: the vector of input signals
$\boldsymbol{\eta}$: the learning rate coefficient

Post-synaptic activation of the neuron

$$y_k = ReLU(\omega_{ik} \cdot x_i)$$

Hebbian plasticity rule with weight decay

$$\Delta\omega_{ik} = \eta \cdot y(x_i, \omega_{ik}) \cdot x_i - \gamma(x_i, \omega_{ik})$$ (Gerstner, W., & Kistler, W. M. 2002)

$$\Delta\omega_{ik} = \eta \cdot y(x_i, \omega_{ik}) \cdot (x_i - \omega_{ik})$$ (Haykin, S. 2009)

Content

- Introduction to Timoleon Moraitis
- Background
- SoftHebb
- Experiments
- Discussion

SoftHebb

SoftHebb plasticity rule: realizes a soft WTA competition through softmax

$$y_k = \frac{b^{u_k}}{\sum_{l=1}^{K} b^{u_l}} = \frac{e^{\frac{u_k}{\tau}}}{\sum_{l=1}^{K} e^{\frac{u_l}{\tau}}}$$

$\boldsymbol{b}$: base
$\boldsymbol{\tau}$: temperature
$\boldsymbol{u_k}$: the k-th neuron's total weighted input
$\boldsymbol{y_k}$: output after accounting for competition from neurons
$\boldsymbol{K}$: number of neurons in a layer
$\boldsymbol{x_i}$: activation of neuron i

$$\Delta\omega_{ik}^{(SoftHebb)} = \eta \cdot \boxed{y_k} \cdot (x_i - u_k \cdot \omega_{ik})$$

Negates SoftHebb's weight update in all neurons except the maximally activated one

Training & Tricks

Greedy layer-wise training

- Restricted Boltzmann Machines (RBM)
- Autoencoder

Combination of activation functions

Rectified polynomial unit (RePU)

$$RePU(u) = \begin{cases} u^p, & \text{for } u > 0 \\ 0, & \text{for } u \leq 0 \end{cases}$$

Triangle activation

$$Triangle(u_j) = RePU(u_j - \overline{u})$$

Neuron-wise adaptive learning rate

$$r_i = E\left(\sqrt{\sum_{j=0}^{N_i} W_{ij}^2}\right) = \sqrt{N_i} \cdot E(|\omega_i|)$$

$$\eta_i = \eta \cdot (r_i - 1)^q$$

$\boldsymbol{N_i}$: number of synapses of neuron i
$\boldsymbol{r_i}$: radius of neuron i

$\boldsymbol{q}$:hyperparameter

SoftHebb Architecture

SoftHebb Architecture

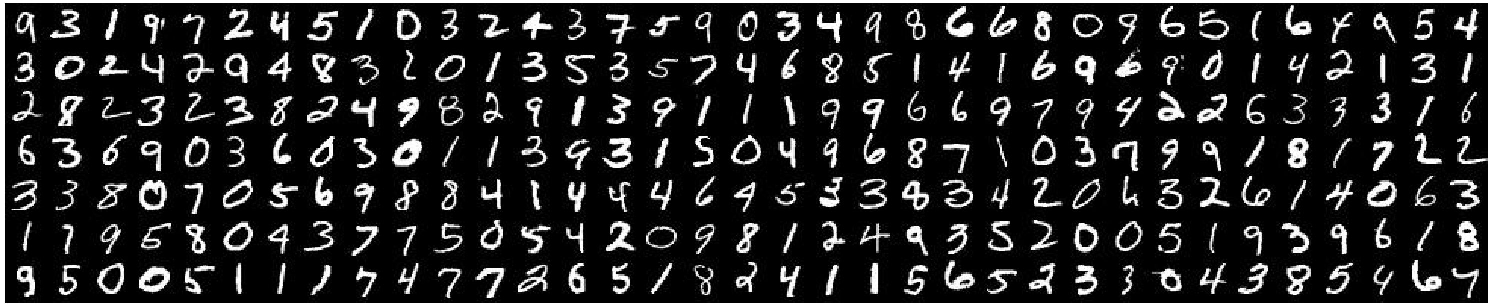
# layer	MNIST/CIFAR	STL10	ImageNet
1	Batchnorm 5×5 conv96 Triangle 4×4 MaxPool	Batchnorm 5×5 conv96 Triangle 4×4 MaxPool	Batchnorm 5×5 conv48 Triangle 4×4 MaxPool
2	Batchnorm 3×3 conv384 Triangle 4×4 MaxPool	Batchnorm 3×3 conv384 Triangle 4×4 MaxPool	Batchnorm 3×3 conv192 Triangle 4×4 MaxPool
3	Batchnorm 3×3 conv1536 Triangle 2×2 AvgPool	Batchnorm 3×3 conv1536 Triangle 4x4 MaxPool	Batchnorm 3×3 conv768 Triangle 4×4 MaxPool
4		Batchnorm 3×3 conv6144 Triangle 2×2 AvgPool	Batchnorm 3×3 conv3072 Triangle 4×4 MaxPool
5			Batchnorm 5×5 conv12288 Triangle 2×2 AvgPool

Content

- Introduction to Timoleon Moraitis
- Background
- SoftHebb
- Experiments
- Discussion

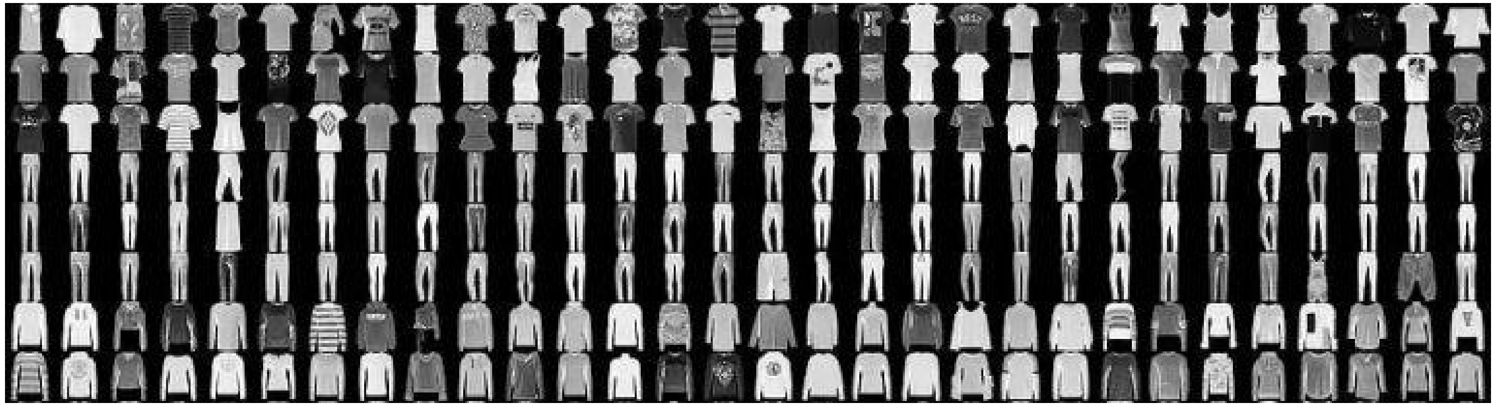
Dataset

MNIST



Class: 10; **Labeled data**: 6W Training, 1W Testing; **Size**: 28*28

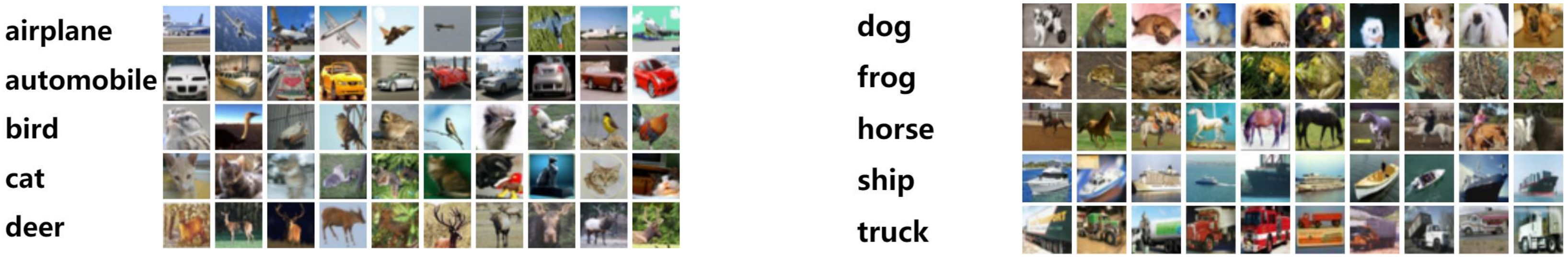
Fashion-MNIST



Class: 10; **Labeled data**: 6W Training, 1W Testing; **Size**: 28*28

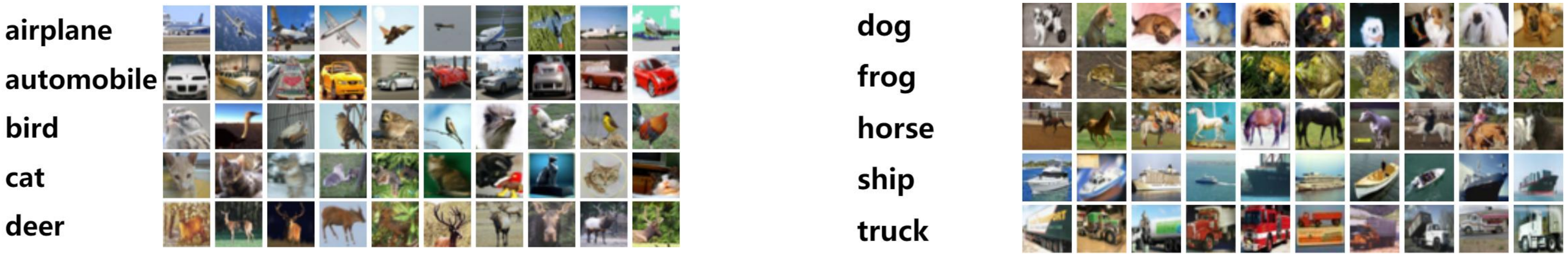
Dataset

CIFAR-10



Labeled data: 5W Training, 1W Testing; **Size**: 32*32

STL-10



Labeled data: 5K Training, 8K Testing; **Unlabeled data**: 10W; **Size**: 96*96

Dataset

CIFAR-100

Superclass	Classes	Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale	large natural outdoor scenes	cloud, forest, mountain, plain, sea
fish	aquarium fish, flatfish, ray, shark, trout	large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
flowers	orchids, poppies, roses, sunflowers, tulips	medium-sized mammals	fox, porcupine, possum, raccoon, skunk
food containers	bottles, bowls, cans, cups, plates	non-insect invertebrates	crab, lobster, snail, spider, worm
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers	people	baby, boy, girl, man, woman
household electrical devices	clock, computer keyboard, lamp, telephone, television	reptiles	crocodile, dinosaur, lizard, snake, turtle
household furniture	bed, chair, couch, table, wardrobe	small mammals	hamster, mouse, rabbit, shrew, squirrel
insects	bee, beetle, butterfly, caterpillar, cockroach	trees	maple, oak, palm, pine, willow
large carnivores	bear, leopard, lion, tiger, wolf	vehicles 1	bicycle, bus, motorcycle, pickup truck, train
large man-made outdoor things	bridge, castle, house, road, skyscraper	vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

Class: 100; **Labeled data**: 5W Training, 1W Testing; **Size**: 32*32

ImageNet

Class: 2W+; **Labeled data**: 5K Training, 8K Testing; **Unlabeled data**: 10W; **Size**: 96*96

Comparison Results on CIFAR-10

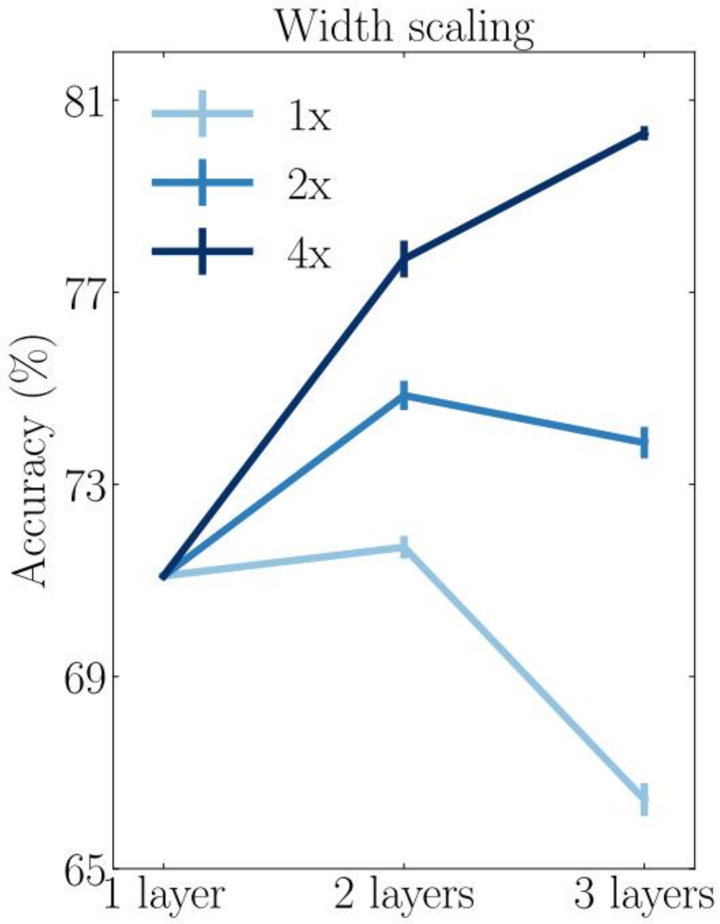
Qualities	Accuracy	Layers	Algorithm	Reference
	99.4	152	Backprop (cross-entropy)	Kolesnikov et al. 2020
	84.0	4	Backprop (cross-entropy)	Ours
Weight-transport-free	71.8	5	Feedback Alignment	Frenkel et al. 2021
Weight-transport-free, Local plasticity	~60	6	Predictive Coding	Millidge et al. 2020
	13.4	5	Equilibrium Propagation (2-phase)	Laborieux et al. 2021
	78.5	5	EP (2-phase, random sign)	Laborieux et al. 2021
	79.9	5	Burstprop	Payeur et al. 2021
	61.0	5	BurstCCN	Greedy et al. 2022
	70.5	5	Direct Feedback Alignment	Frenkel et al. 2021
	71.5	5	DFA (untrained convs)	Frenkel et al. 2021
Weight-transport-free, Local plasticity, Update-unlocked	65.6	5	Direct Random Target Projection	Frenkel et al. 2021
	69.0	5	DRTP (untrained convs)	Frenkel et al. 2021
	73.1	5	Single Sparse DFA	Crafton et al. 2019
	53.5	11	Latent Predictive Learning	Halvagal and Zenke 2022
Weight-transport-free, Local plasticity, Update-unlocked, Unsupervised	73.7	4	Self Organising Maps	Stuhr and Brauer 2019
	72.2	2	Hard WTA	Grinberg et al. 2019
	64.6	4	Hard WTA	Miconi 2021
	80.3	4	SoftHebb (1 epoch)	Ours

Architecture Analysis

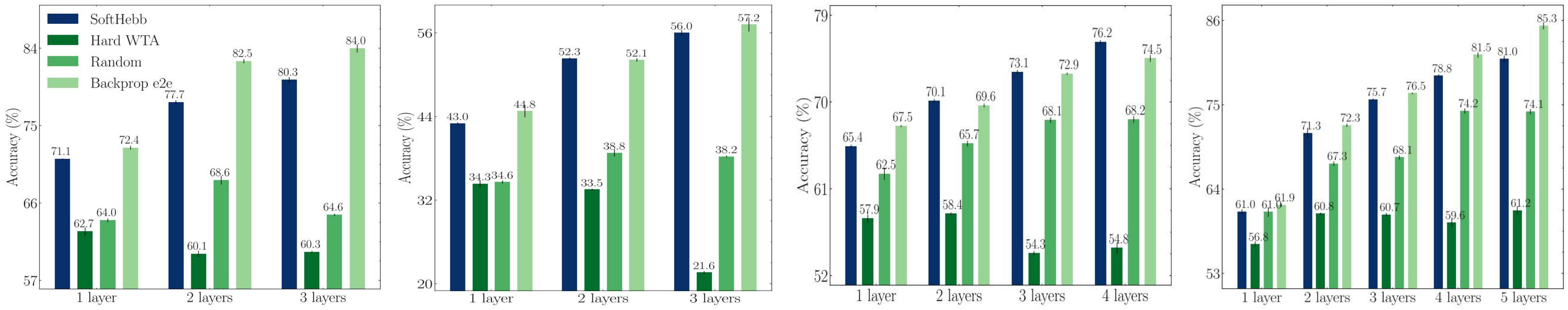
SoftHebb Architecture

# layer	MNIST/CIFAR	STL10	ImageNet
1	Batchnorm 5×5 conv96 Triangle 4×4 MaxPool	Batchnorm 5×5 conv96 Triangle 4×4 MaxPool	Batchnorm 5×5 conv48 Triangle 4×4 MaxPool
2	Batchnorm 3×3 conv384 Triangle 4×4 MaxPool	Batchnorm 3×3 conv384 Triangle 4×4 MaxPool	Batchnorm 3×3 conv192 Triangle 4×4 MaxPool
3	Batchnorm 3×3 conv1536 Triangle 2×2 AvgPool	Batchnorm 3×3 conv1536 Triangle 4x4 MaxPool	Batchnorm 3×3 conv768 Triangle 4×4 MaxPool
4		Batchnorm 3×3 conv6144 Triangle 2×2 AvgPool	Batchnorm 3×3 conv3072 Triangle 4×4 MaxPool
5			Batchnorm 5×5 conv12288 Triangle 2×2 AvgPool

CIFAR-10 layer-wise performance

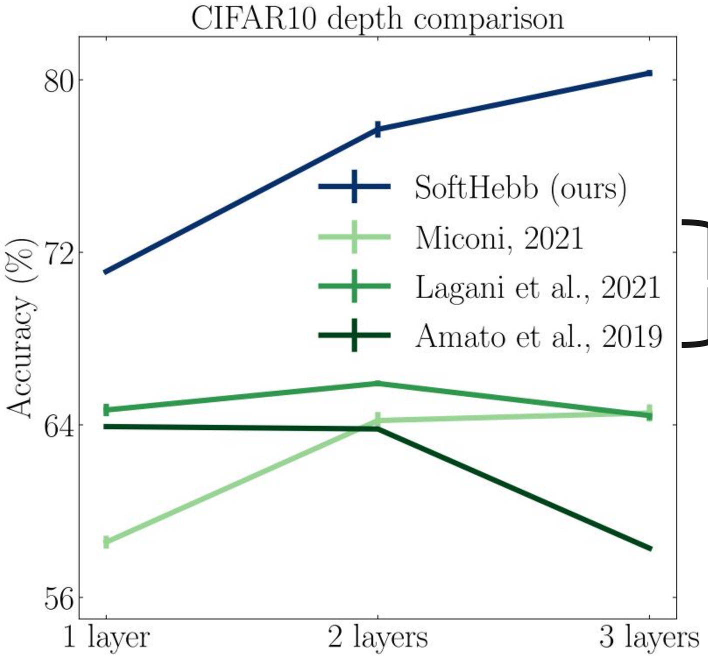


Depth-wise Performance



(A) CIFAR-10 (B) CIFAR-100 (C) STL-10 (D) ImageNette

Receptive Field Analysis



Hard WTA

Figure 1: First successful multilayer results. SoftHebb's CIFAR-10 accuracy increases with depth (*hidden* layers), compared with prior work.

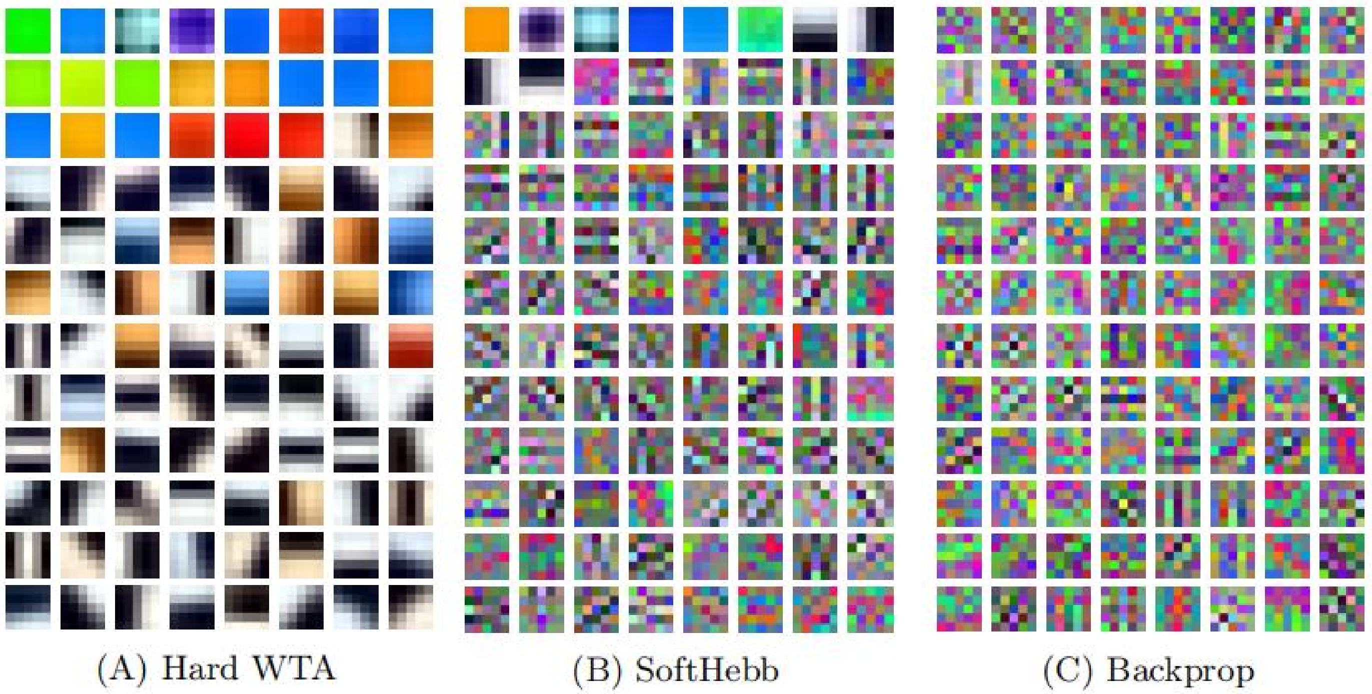
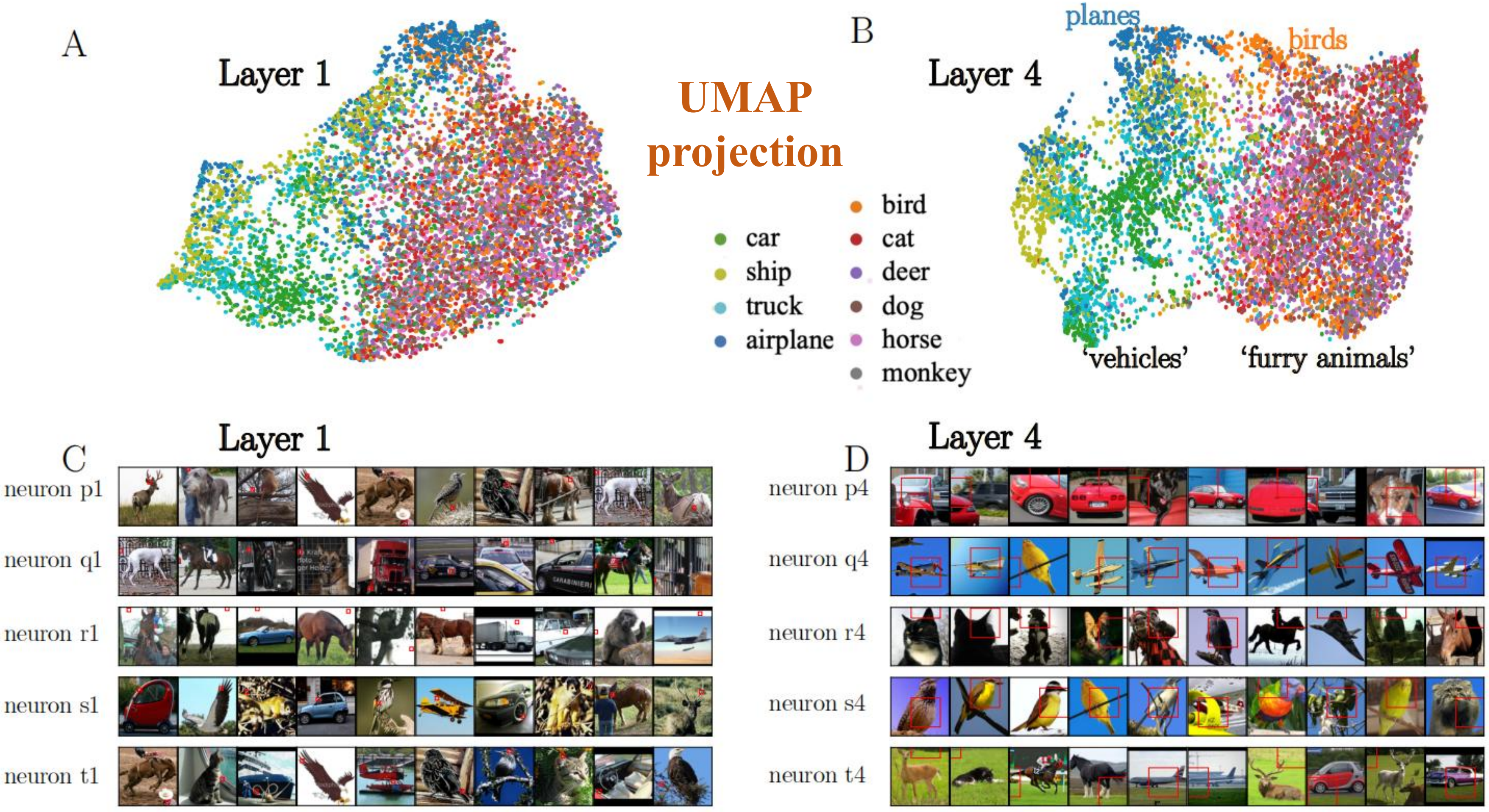


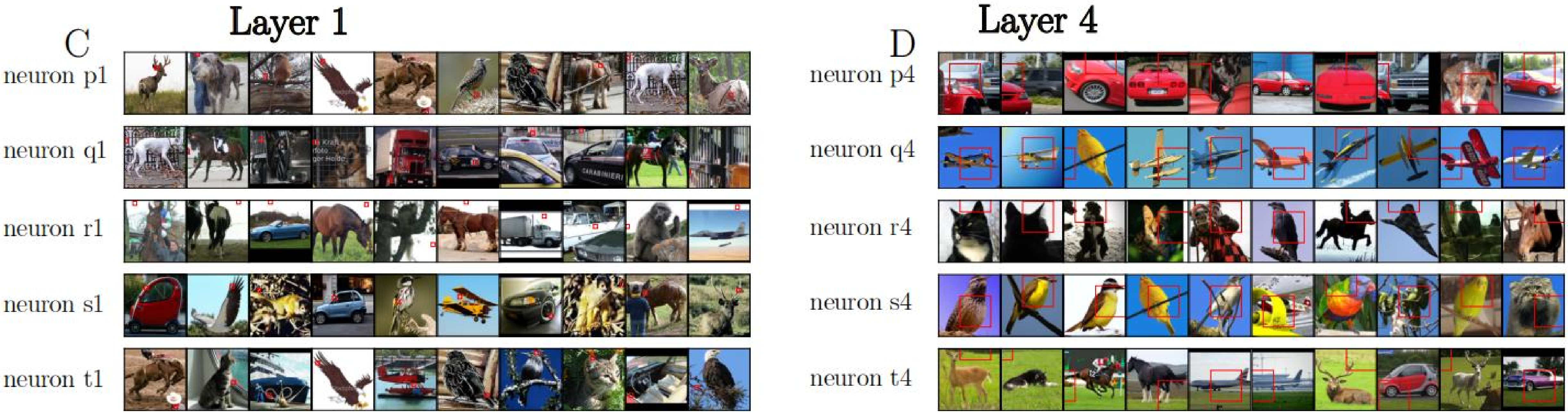
Figure B.5: Receptive fields of the first convolutional layer's neurons, learned from CIFAR-10 by different algorithms.

Representation Analysis



Hierarchical representations learned by SoftHebb on STL-10

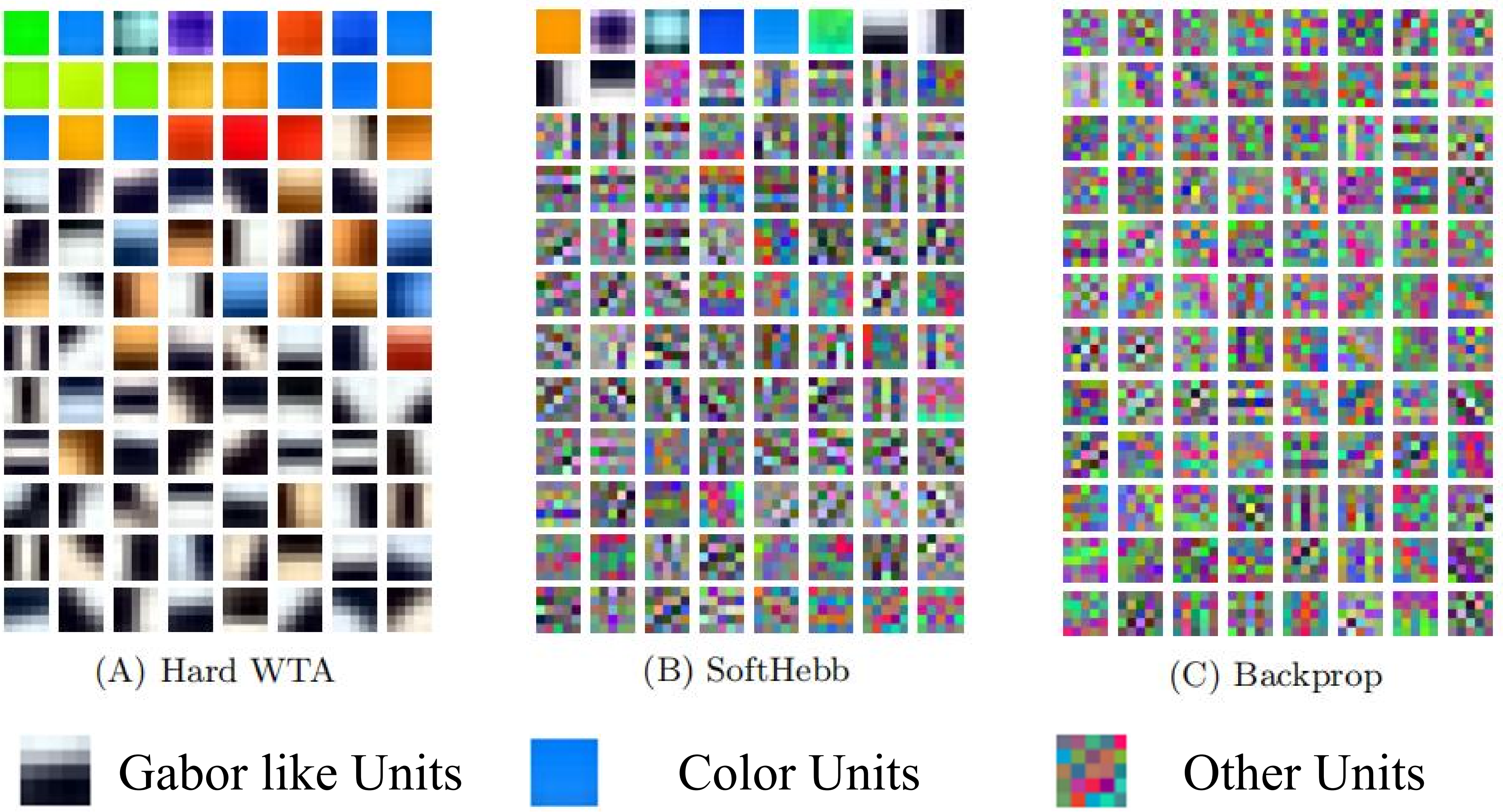
Representation Analysis



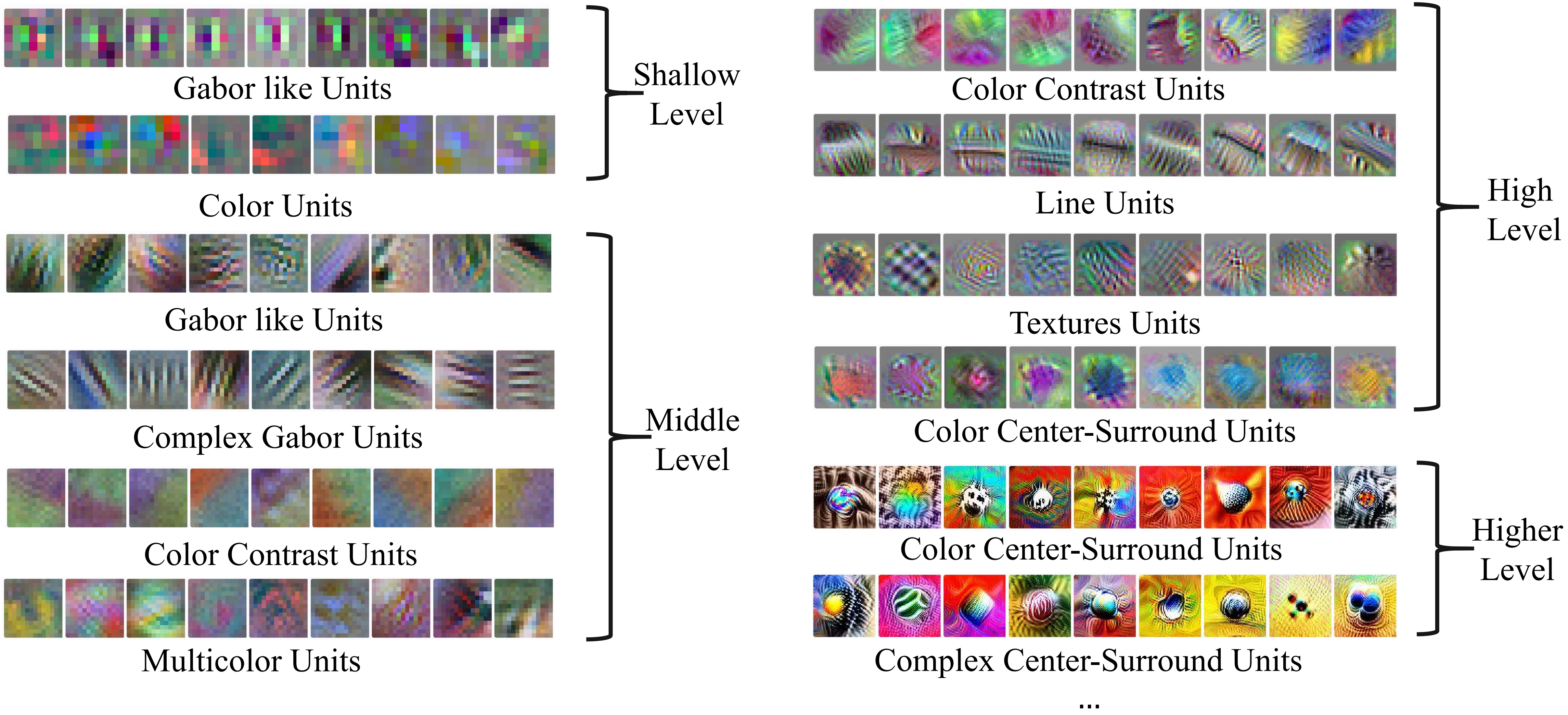
Content

- Introduction to Timoleon Moraitis
- Background
- SoftHebb
- Experiments
- Discussion

Discussion


(A) Hard WTA
(B) SoftHebb
(C) Backprop
Gabor like Units
Color Units
Other Units

Discussion



(Olah, et al. 2020)

Reference

- Hebb, D. O. (1949). The organization of behavior; a neuropsychological theory.
- Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. Journal of Neuroscience, 2(1), 32-48.
- Gerstner, W., & Kistler, W. M. (2002). Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press.
- Haykin, S. (2009). Neural networks and learning machines, 3/E. Pearson Education India.
- Olah, et al. (2020). An Overview of Early Vision in InceptionV1. Distill.
- Bao, P., She, L., McGill, M., & Tsao, D. Y. (2020). A map of object space in primate inferotemporal cortex. Nature, 583(7814), 103-108.
- Lee, J. J., Krumin, M., Harris, K. D., & Carandini, M. (2022). Task specificity in mouse parietal cortex. Neuron, 110(18), 2961-2969.

Thanks for your attention!