

Unlocking the Potential of Similarity Matching: Scalability, Supervision, and Pre-training

Yanis Bahroun¹ Shagesh Sridharan² Atithi Acharya²
Dmitri B. Chklovskii^{1,3} Anirvan M. Sengupta^{1,2}

Similarity Matching

Similarity Matching algorithms [1, 3] exhibit **locality**, **online trainability**, and **bio-plausibility**.

Nonnegative Similarity Matching (NSM)

The objective function considered in [2] is

$$\hat{\mathbf{Z}} = \arg \min_{\mathbf{Z} \in \mathbb{R}_+^{m \times T}} \|\mathbf{X}^T \mathbf{X} - \mathbf{Z}^T \mathbf{Z}\|_F^2. \quad (1)$$

- $\mathbf{X} \in \mathbb{R}^{n \times T}$ is the input matrix
- $\mathbf{Z} \in \mathbb{R}_+^{m \times T}$ is the encoding matrix

NSM as a min-max objective function

Introduce auxiliary variables, \mathbf{W} and \mathbf{M} [4]:

$$\min_{\mathbf{Z} \in \mathbb{R}_+^{m \times T}, \mathbf{W}, \mathbf{M}} \max -4 \text{Tr}(\mathbf{X}^T \mathbf{W}^T \mathbf{Z} - \frac{1}{2} \mathbf{Z}^T \mathbf{M}^T \mathbf{Z}) + 2 \text{Tr}(\mathbf{W}^T \mathbf{W}) - \text{Tr}(\mathbf{M}^T \mathbf{M}). \quad (2)$$

Online algorithm and neural implementation

Gradient-descent ascent of Eq. (2) gives,

Neural dynamics:

$$\frac{d\mathbf{Z}(\gamma)}{d\gamma} = [\mathbf{W}\mathbf{X} - \mathbf{M}\mathbf{Z}(\gamma)]_+, \quad (3)$$

Synaptic learning rules:

$$\Delta \mathbf{W} = \mathbf{X} \hat{\mathbf{Z}}^T, \quad \Delta \mathbf{M} = -\hat{\mathbf{Z}} \hat{\mathbf{Z}}^T. \quad (4)$$

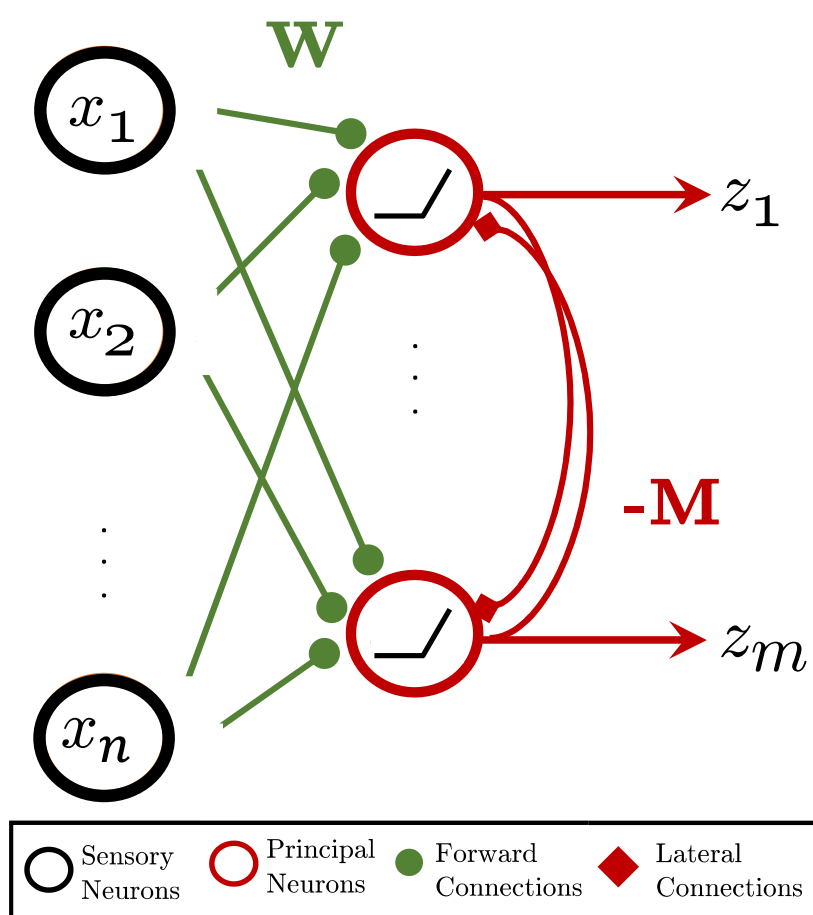


Figure 1: Single-layer NN performing online NSM [2]

References

- [1] Lipshutz, D., Bahroun, Y., Golkar, S., Sengupta, A.M., Chklovskii, D.B.: A normative framework for deriving neural networks with multi-compartmental neurons and non-hebbian plasticity. arXiv preprint arXiv:2302.10051 (2023)
- [2] Pehlevan, C., Chklovskii, D.B.: A Hebbian/anti-Hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features. In: 2014 48th Asilomar Conference. pp. 769--775. IEEE (2014)
- [3] Pehlevan, C., Chklovskii, D.B.: Neuroscience-inspired online unsupervised learning algorithms: Artificial neural networks. IEEE Signal Processing Magazine 36(6), 88--96 (2019)
- [4] Pehlevan, C., Sengupta, A.M., Chklovskii, D.B.: Why do similarity matching objectives lead to hebbian/anti-hebbian networks? Neural computation 30(1), 84--124 (2017)
- [5] Qin, S., Mudur, N., Pehlevan, C.: Contrastive similarity matching for supervised learning. Neural computation 33(5), 1300--1328 (2021)

Contributions

Our contributions are the development of a **scalable convolutional NSM implementation using PyTorch** as a localized learning alternative to backpropagation. We introduce a **localized supervised objective** and explore **NSM-based pre-training** for models such as LeNet. These models enhance overall performance and facilitate efficient learning processes.

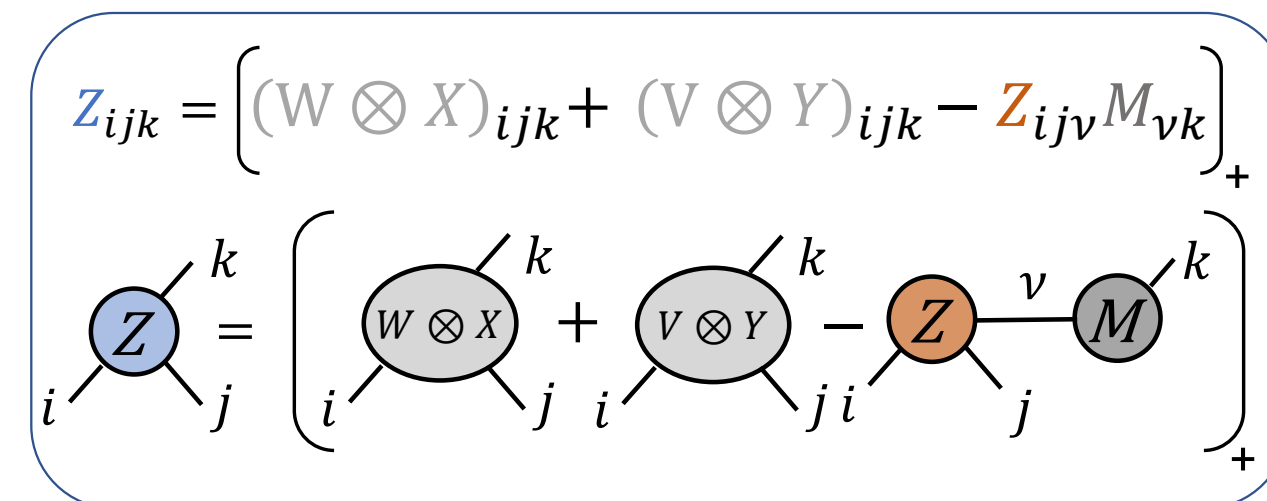


Figure 2: Graphical notation of tensor operations

Algorithm	Conventional	CPU	GPU
10k images	2399s	93.85s	13.54s

Table 1: Training times for processing 10,000 images.

Pre-training LeNet with S²M

Step 1. Pre-training.

- Initialize a single-layer S²M network with the same number of neurons as filters in LeNet layers.
- Train the S²M by executing neural dynamics.
- Initialize the LeNet layer with the learned weights \mathbf{W} .
- Initialize the other layers of LeNet randomly.

Step 2. Fine-tuning with BP.

- Perform supervised fine-tuning of the LeNet layer through BP for all layers.

Step 3. Compare rotation during BP for varying supervision.

- Filters are most stable and retain initial orientations at $\alpha_1 = 10^{-3}$.

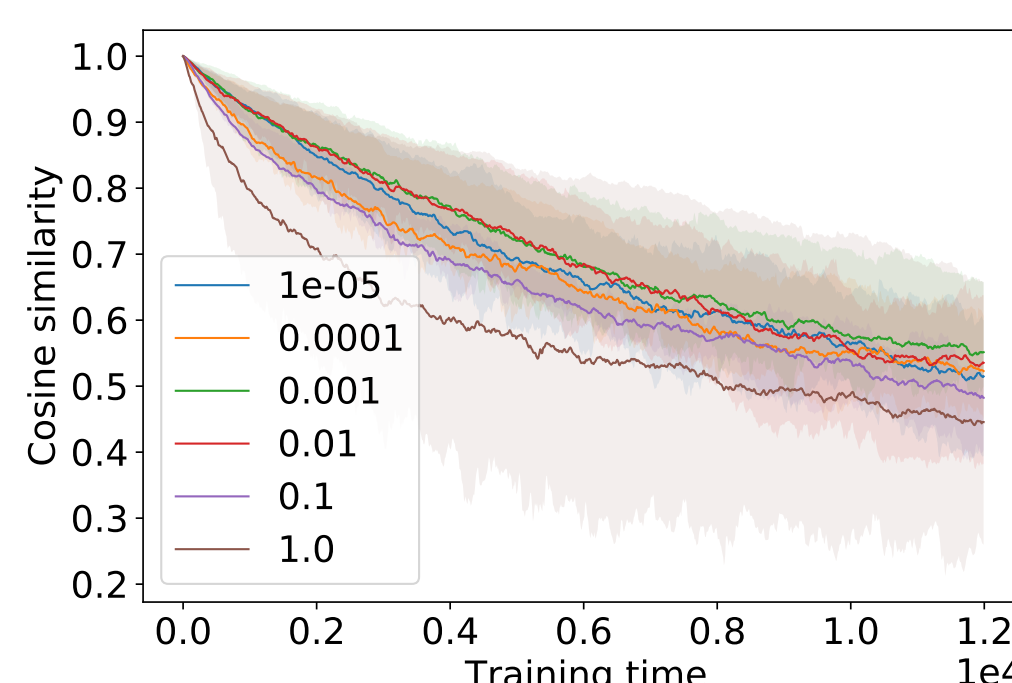


Figure 3: Evaluation of LeNet Pre-training using S²M (6 neurons)

Online Supervised SM Algorithm

Supervised Similarity Matching (S²M)

For $k \in \{1, L\}$ where L is the number of layers, we define the supervised SM as follows,

$$\hat{\mathbf{Z}}_k = \arg \min_{\mathbf{Z}_k \geq 0} \left\| \left[\hat{\mathbf{Z}}_{k-1}^T \hat{\mathbf{Z}}_{k-1} + \alpha_k \mathbf{Y}^T \mathbf{Y} \right] - \mathbf{Z}_k^T \mathbf{Z}_k \right\|_F^2 \quad (5)$$

- $\mathbf{Y} \in \mathbb{R}^{c \times T}$ is the matrix of labels (one-hot)
- α_k controls label matrix influence

We absorb α_k into $\mathbf{Y}^T \mathbf{Y}$ for simplicity.

S²M as a min-max objective function

We rewrite (5) using auxiliary variables \mathbf{W}_k , \mathbf{M}_k , and \mathbf{V}_k as

$$\max_{\mathbf{M}_k} \min_{\mathbf{W}_k, \mathbf{V}_k, \mathbf{Z}_k \in \mathbb{R}_+^{m_k \times T}} l(\mathbf{Z}_{k-1}, \mathbf{Z}_k, \mathbf{Y}, \mathbf{W}_k, \mathbf{M}_k, \mathbf{V}_k). \quad (6)$$

$$l(\mathbf{Z}_{k-1}, \mathbf{Z}_k, \mathbf{Y}, \mathbf{W}_k, \mathbf{M}_k, \mathbf{V}_k) = -4 \text{Tr} \left(\left[\mathbf{Z}_{k-1}^T \mathbf{W}_k^T + \mathbf{Y}^T \mathbf{V}_k^T - \frac{1}{2} \mathbf{Z}_k^T \mathbf{M}_k^T \right] \mathbf{Z}_k \right) + 2 \text{Tr}(\mathbf{W}_k^T \mathbf{W}_k + \mathbf{V}_k^T \mathbf{V}_k) - \text{Tr}(\mathbf{M}_k^T \mathbf{M}_k).$$

Online algorithm and neural implementation

Gradient-descent ascent on (6) gives:

Neural dynamics:

$$\frac{d\mathbf{Z}_k(\gamma)}{d\gamma} = [\mathbf{W}_k \hat{\mathbf{Z}}_{k-1} + \mathbf{V}_k \mathbf{Y} - \mathbf{M}_k \mathbf{Z}_k(\gamma)]_+.$$

We identify the auxiliary variables

- \mathbf{W}_k with feedforward connections
- \mathbf{M}_k with lateral connections
- \mathbf{V}_k with label-encoder connections

Synaptic learning rules:

$$\Delta \mathbf{V}_k = \mathbf{Y} \hat{\mathbf{Z}}_k, \quad \Delta \mathbf{W}_k = \hat{\mathbf{Z}}_{k-1} \hat{\mathbf{Z}}_k, \quad \Delta \mathbf{M}_k = -\hat{\mathbf{Z}}_k \hat{\mathbf{Z}}_k^T.$$

Numerical Evaluation

- We test for different levels of supervision.
- We compare with Contrastive Similarity Matching and Equilibrium Propagation [5]
- We observe maximum validation accuracy for S²M at $\alpha_1 = 10^{-3}$.

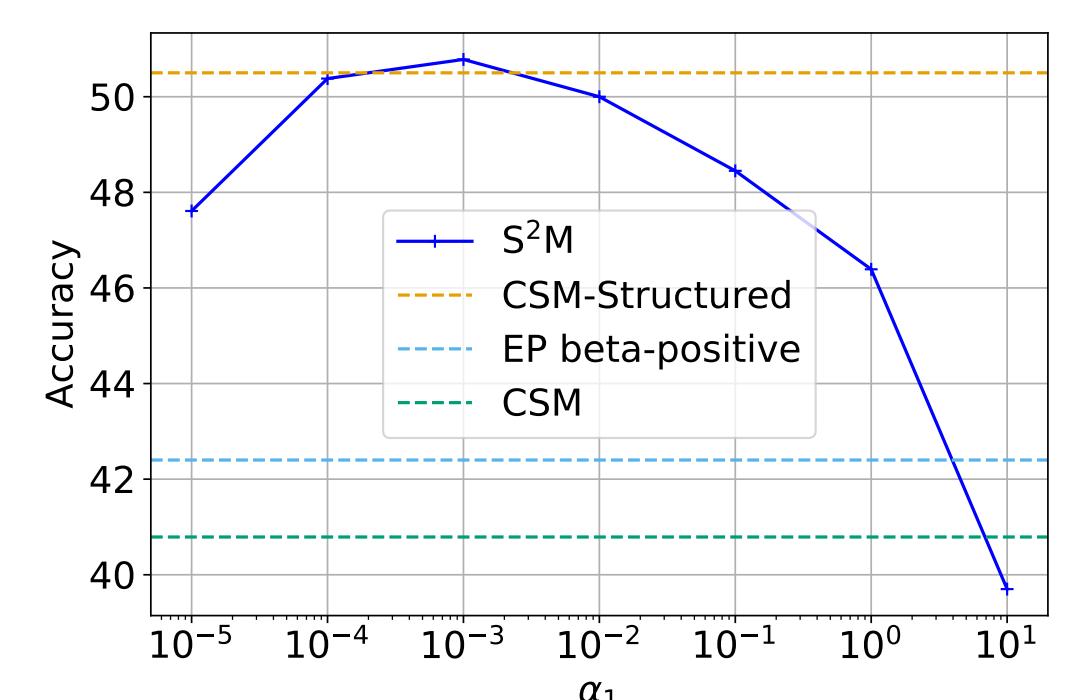


Figure 4: Evaluation of S²M (10 neurons) on CIFAR-10