

Recent Advancements on Bio-Inspired Hebbian Learning for Deep Neural Networks

Gabriele Lagani^{1,2}

¹University of Pisa, Pisa, 56127, Italy

²ISTI-CNR, Pisa, 56124, Italy

Abstract

Deep learning is becoming more and more popular to extract information from multimedia data for indexing and query processing. In recent contributions, we have explored a biologically inspired strategy for Deep Neural Network (DNN) training, based on the Hebbian principle in neuroscience. We studied hybrid approaches in which unsupervised Hebbian learning was used for a pre-training stage, followed by supervised fine-tuning based on Stochastic Gradient Descent (SGD). The resulting semi-supervised strategy exhibited encouraging results on computer vision datasets, motivating further interest towards applications in the domain of large scale multimedia content based retrieval.

Keywords

Machine Learning, Hebbian Learning, Deep Neural Networks, Computer Vision

1. Introduction

In the past few years, Deep Neural Networks (DNNs) have emerged as a powerful technology in the domain of computer vision [1, 2]. Consequently, DNNs started gaining popularity also in the domain of large scale multimedia content based retrieval, replacing handcrafted feature extractors [3, 4]. Learning algorithms for DNNs are typically based on supervised end-to-end Stochastic Gradient Descent (SGD) training with error backpropagation (*backprop*). This approach is considered biologically implausible by neuroscientists [5]. Instead, they propose *Hebbian* learning as a biological alternative to model synaptic plasticity [6].

Backprop-based algorithms need a large number of labeled training samples in order to achieve high results, which are expensive to gather, as opposed to unlabeled samples.

The idea behind our contribution [7, 8] is to tackle the sample efficiency problem by taking inspiration from biology and Hebbian learning. Since Hebbian approaches are mainly unsupervised, we propose to use them to perform the unsupervised pre-training stage on all the available data, in a semi-supervised setting, followed by end-to-end backprop fine-tuning on the labeled data only. In the rest of this paper, we illustrate the proposed methodology, and we show experimental results in computer vision. The results are promising, motivating further interest in the application of our approach to the domain of multimedia content retrieval on a large scale.

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy

✉ gabriele.lagani@phd.unipi.it (G. Lagani)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The remainder of this paper is structured as follows: Section 2 gives a background concerning Hebbian learning and semi-supervised training; Section 3 delves deeper into the semi-supervised approach based on Hebbian learning; In Section 4, we illustrate our experimental results and discuss the conclusions;

2. Background and related work

Several variants of Hebbian learning rules were developed over the years. Some examples are: Hebbian learning with Winner-Takes-All (WTA) competition [9], Hebbian learning for Principal Component Analysis (PCA) [6, 10], Hebbian/anti-Hebbian learning [11]. A brief overview is given in Section 3. However, it was only recently that Hebbian learning started gaining attention in the context of DNN training [12, 13, 14, 15, 16].

In [14], a Hebbian learning rule based on inhibitory competition was used to train a neural network composed of fully connected layers. The approach was validated on object recognition tasks. Instead, the Hebbian/anti-Hebbian learning rule developed in [11] was applied in [13] to train convolutional feature extractors. The resulting features were shown to be effective for classification. Convolutional layers were also considered in [12], where a Hebbian approach based on WTA competition was employed in this case.

However, the previous approaches were based on relatively shallow network architectures (2-3 layers). A further step was taken in [15, 16], where Hebbian learning rules were applied for training a 6-layer Convolutional Neural Network (CNN).

It is known that a pre-training phase allows to initialize network weights in a region near a good local optimum [17, 18]. Previous papers investigated the idea of enhancing neural network training with an unsupervised learning objective [19, 20]. In [19], Variational Auto-Encoders (VAE) were considered, in order to perform an unsupervised pre-training phase using a limited amount of labeled samples. Also [20] relied on autoencoding architectures to augment supervised training with unsupervised reconstruction objectives, showing that joint optimization of supervised and unsupervised losses helped to regularize the learning process.

3. Hebbian learning strategies and sample efficiency

Consider a single neuron with weight vector \mathbf{w} and input \mathbf{x} . Call $y = \mathbf{w}^T \mathbf{x}$ the neuron output. A learning rule defines a weight update as follows:

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \Delta \mathbf{w} \quad (1)$$

where \mathbf{w}_{new} is the updated weight vector, \mathbf{w}_{old} is the old weight vector, and $\Delta \mathbf{w}$ is the weight update.

The Hebbian learning rule, in its simplest form, can be expressed as $\Delta \mathbf{w} = \eta y \mathbf{x}$ (where η is the learning rate) [6]. Basically, this rule states that the weight on a given synapse is reinforced when the input on that synapse and the output of the neuron are simultaneously high. Therefore, connections between neurons whose activations are correlated are reinforced. In order to prevent weights from growing unbounded, a weight decay term is generally added.

In the context of competitive learning [9], this is obtained as follows:

$$\Delta \mathbf{w}_i = \eta y_i \mathbf{x} - \eta y_i \mathbf{w}_i = \eta y_i (\mathbf{x} - \mathbf{w}_i) \quad (2)$$

where the subscript i refers to the i 'th neuron in a given network layer. Moreover, the output y_i can be replaced with the result r_i of a competitive nonlinearity, which allows to decorrelate the activity of different neurons. In the Winner-Takes-All (WTA) approach [9], at each training step, the neuron which produces the strongest activation for a given input is called the *winner*. In this case, $r_i = 1$ if the i 'th neuron is the winner and 0 otherwise. In other words, only the winner is allowed to perform the weight update, so that it will be more likely for the same neuron to win again if a similar input is presented again in the future. In this way different neurons are induced to specialize on different patterns. In soft-WTA [21], r_i is computed as $r_i = \frac{y_i}{\sum_j y_j}$. We found this formulation to work poorly in practice, because there is no tunable parameter to cope with the variance of activations. For this reason, we introduced a variant of this approach that uses a *softmax* operation in order to compute r_i :

$$r_i = \frac{e^{y_i/T}}{\sum_j e^{y_j/T}} \quad (3)$$

where T is called the *temperature* hyperparameter. The advantage of this formulation is that we can tune the temperature in order to obtain the best performance on a given task, depending on the distribution of the activations.

The Hebbian Principal Component Analysis (HPCA) learning rule, in the case of nonlinear neurons, is obtained by minimizing the so-called *representation error*:

$$L(\mathbf{w}_i) = E[(x - \sum_{j=1}^i f(y_j) \mathbf{w}_j)^2] \quad (4)$$

where $f()$ is the neuron activation function. Minimization of this objective leads to the nonlinear HPCA rule [10]:

$$\Delta \mathbf{w}_i = \eta f(y_i)(x - \sum_{j=1}^i f(y_j) \mathbf{w}_j) \quad (5)$$

It can be noticed that these learning rules do not require supervision, and they are *local* for each network layer, i.e. they do not require backpropagation. .

In order to contextualize our approach in a scenario with scarce data, let's define the *labeled set* \mathcal{T}_L as a collection of elements for which the corresponding label is known. Conversely, the *unlabeled set* \mathcal{T}_U is a collection of elements whose labels are unknown. The whole *training set* \mathcal{T} is given by the union of \mathcal{T}_L and \mathcal{T}_U . All the samples from \mathcal{T} are assumed to be drawn from the same statistical distribution. In a *sample efficiency* scenario, the number of samples in \mathcal{T}_L is typically much smaller than the total number of samples in \mathcal{T} . In particular, in an $s\%$ -sample efficiency *regime*, the size of the labeled set is $s\%$ that of the whole training set.

To tackle this scenario, we considered a semi-supervised approach in two phases. During the first phase, latent representations are obtained from hidden layers of a DNN, which are trained using unsupervised Hebbian learning. This unsupervised pre-training is performed on all the

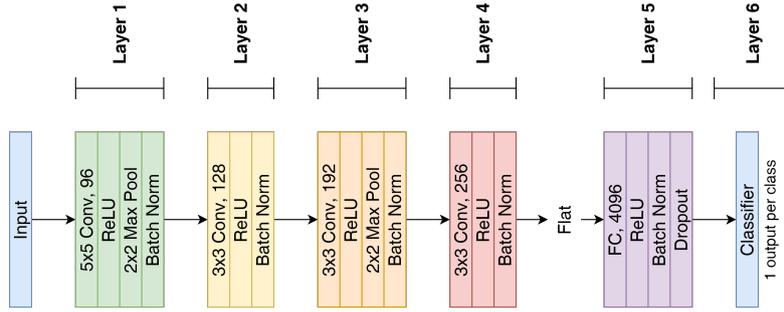


Figure 1: The neural network used for the experiments.

available training samples. During the second phase, a final linear classifier is placed on top of the features extracted from deep network layers. Classifier and deep layers are fine-tuned in a supervised training fashion, by running an end-to-end SGD optimization procedure using only the few labeled samples at our disposal.

4. Results and conclusions

In order to validate our method, we performed experiments ¹ on various image datasets, and in various sample efficiency regimes. For the sake of brevity, but without loss of generality, in this venue we present the results on CIFAR10 [22], in sample efficiency regimes where the amount of labeled samples was respectively 1%, 5%, 10%, and 100% of the whole training set. Further results can be found in [7, 8].

We considered a six layer neural network as shown in Fig. 1: five deep layers plus a final linear classifier. The various layers were interleaved with other processing stages (such as ReLU nonlinearities, max-pooling, etc.). We first performed unsupervised pre-training with a chosen algorithm. Then, we cut the network in correspondence of a given layer, and we attached a new classifier on top of the features extracted from that layer. Deep layers and classifier were then fine-tuned with supervision in an end-to-end fashion and the resulting accuracy was evaluated. This was done for each layer, in order to evaluate the network on a layer-by-layer basis, and for each sample efficiency regime. For the unsupervised pre-training of deep layers, we considered both the HPCA and the soft-WTA strategy. In addition, as a baseline for comparison, we considered another popular unsupervised method for pre-training, namely the Variational Auto-Encoder (VAE) [23] (considered also in [19]). Note that VAE is unsupervised, but still backprop-based.

The results are shown Tab. 1, together with 95% confidence intervals obtained from five independent repetitions of the experiments. In summary, the results suggest that our semi-supervised approach based on unsupervised Hebbian pre-training performs generally better than VAE pre-training, especially in low sample efficiency regimes, in which only a small portion of the training set (between 1% and 10%) is assumed to be labeled. In particular, the HPCA

¹Code available at:
<https://github.com/GabrieleLagani/HebbianPCA/tree/hebbpca>.

Table 1

CIFAR10 accuracy and 95% confidence intervals, on various layers and for various sample efficiency regimes. Results obtained with VAE and Hebbian pre-training are compared.

Regime	Pre-Train	L1	L2	L3	L4	L5
1%	VAE	33.54 \pm 0.27	34.41 \pm 0.84	29.92 \pm 1.25	24.91 \pm 0.66	22.54 \pm 0.60
	soft-WTA	35.47 \pm 0.19	35.75 \pm 0.65	36.09 \pm 0.27	30.57 \pm 0.36	30.23 \pm 0.37
	HPCA	37.01 \pm 0.42	37.65 \pm 0.19	41.88 \pm 0.53	40.06 \pm 0.65	39.75 \pm 0.50
5%	VAE	46.31 \pm 0.39	48.21 \pm 0.21	48.98 \pm 0.34	36.32 \pm 0.35	32.75 \pm 0.32
	soft-WTA	48.34 \pm 0.27	52.90 \pm 0.28	54.01 \pm 0.24	49.80 \pm 0.16	48.35 \pm 0.26
	HPCA	48.49 \pm 0.44	50.14 \pm 0.46	53.33 \pm 0.52	52.49 \pm 0.16	52.20 \pm 0.37
10%	VAE	53.83 \pm 0.26	56.33 \pm 0.22	57.85 \pm 0.22	52.26 \pm 1.08	45.67 \pm 1.15
	soft-WTA	54.23 \pm 0.18	59.40 \pm 0.20	61.27 \pm 0.24	58.33 \pm 0.35	58.00 \pm 0.26
	HPCA	54.36 \pm 0.32	56.08 \pm 0.28	58.46 \pm 0.15	56.54 \pm 0.23	57.35 \pm 0.18
100%	VAE	67.53 \pm 0.22	75.83 \pm 0.31	80.78 \pm 0.28	84.27 \pm 0.35	85.23 \pm 0.26
	soft-WTA	67.37 \pm 0.16	77.39 \pm 0.04	81.83 \pm 0.47	84.42 \pm 0.15	85.37 \pm 0.03
	HPCA	66.76 \pm 0.13	75.16 \pm 0.20	79.90 \pm 0.18	83.55 \pm 0.33	84.38 \pm 0.22

approach appears to perform generally better than soft-WTA. Concerning the computational cost of Hebbian learning, the approach converged in just 1-2 epochs of training, while backprop approaches required 10-20 epochs, showing promises towards scaling to large scale scenarios.

In future work, we plan to investigate the combination of Hebbian approaches with alternative semi-supervised methods, namely pseudo-labeling and consistency-based methods [24, 25], which do not exclude unsupervised pre-training, but rather can be integrated together. Moreover, we are currently conducting more thorough explorations of Hebbian algorithms in the domain of large scale multimedia content based retrieval, and the results are promising [26].

Acknowledgments

This work was partially supported by the H2020 projects AI4EU (GA 825619) and AI4Media (GA 951911).

References

- [1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems* (2012).
- [2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [3] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, J. Li, Deep learning for content-based image retrieval: A comprehensive study, in: *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 157–166.
- [4] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, in: *European conference on computer vision*, Springer, 2014, pp. 584–599.

- [5] R. C. O'Reilly, Y. Munakata, Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain, MIT press, 2000.
- [6] S. Haykin, Neural networks and learning machines, 3 ed., Pearson, 2009.
- [7] G. Lagani, F. Falchi, C. Gennaro, G. Amato, Hebbian semi-supervised learning in a sample efficiency setting, *Neural Networks* 143 (2021) 719–731.
- [8] G. Lagani, F. Falchi, C. Gennaro, G. Amato, Evaluating hebbian learning in a semi-supervised setting, in: *International Conference on Machine Learning, Optimization, and Data Science*, Springer, 2021, pp. 365–379.
- [9] S. Grossberg, Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors, *Biological cybernetics* 23 (1976) 121–134.
- [10] J. Karhunen, J. Joutsensalo, Generalizations of principal component analysis, optimization problems, and neural networks, *Neural Networks* 8 (1995) 549–562.
- [11] C. Pehlevan, T. Hu, D. B. Chklovskii, A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data, *Neural computation* 27 (2015) 1461–1495.
- [12] A. Wadhwa, U. Madhow, Bottom-up deep learning using the hebbian principle, 2016.
- [13] Y. Bahroun, A. Soltoggio, Online representation learning with single and multi-layer hebbian networks for image classification, in: *International Conference on Artificial Neural Networks*, Springer, 2017, pp. 354–363.
- [14] D. Krotov, J. J. Hopfield, Unsupervised learning by competing hidden units, *Proceedings of the National Academy of Sciences* 116 (2019) 7723–7731.
- [15] G. Lagani, F. Falchi, C. Gennaro, G. Amato, Training convolutional neural networks with competitive hebbian learning approaches, in: *International Conference on Machine Learning, Optimization, and Data Science*, Springer, 2021, pp. 25–40.
- [16] G. Lagani, F. Falchi, C. Gennaro, G. Amato, Comparing the performance of hebbian against backpropagation learning using convolutional neural networks, *Neural Computing and Applications* (2022) 1–17.
- [17] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: *Advances in neural information processing systems*, 2007, pp. 153–160.
- [18] H. Larochelle, Y. Bengio, J. Louradour, P. Lamblin, Exploring strategies for training deep neural networks., *Journal of machine learning research* 10 (2009).
- [19] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, M. Welling, Semi-supervised learning with deep generative models, *Advances in neural information processing systems* 27 (2014) 3581–3589.
- [20] Y. Zhang, K. Lee, H. Lee, Augmenting supervised neural networks with unsupervised objectives for large-scale image classification, in: *International conference on machine learning*, 2016, pp. 612–621.
- [21] S. J. Nowlan, Maximum likelihood competitive learning, in: *Advances in neural information processing systems*, 1990, pp. 574–582.
- [22] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, 2009.
- [23] D. P. Kingma, M. Welling, Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114* (2013).
- [24] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5070–5079.
- [25] P. Sellars, A. I. Aviles-Rivero, C.-B. Schönlieb, Laplacenet: A hybrid energy-neural model for deep semi-supervised classification, *arXiv preprint arXiv:2106.04527* (2021).
- [26] G. Lagani, D. Bacciu, C. Gallicchio, F. Falchi, C. Gennaro, G. Amato, Deep features for cbir with scarce data using hebbian learning, Submitted at CBMI 2022 conference (2022). URL: <https://arxiv.org/abs/2205.08935>.