

Biological Networks is All You Need

*Unsupervised learning by competing hidden units
by D. Krotov and J. J. Hopfield*

Zanola Andrea

Quantitative Life Science

31/01/2021

Outline

Introduction

Unsupervised Phase

Supervised Phase

Supervised Training

Back-Propagation

*"It is widely believed that end-to-end training with the **back-propagation algorithm is essential** [...]. At the same time, the traditional form of back-propagation is **biologically implausible**".*

Neuroscientists often claim that cortex can't do BP through a hierarchy of cortical areas; at the same time we know that BP is incredibly effective in DL. There are several reasons why brain couldn't do BP:

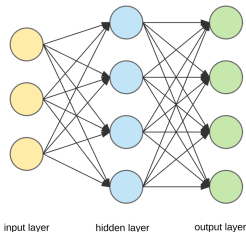
- There is **no obvious source of labels**.
- Cortical neurons don't communicate real-valued activities, but send all-or-none spikes train.
- Artificial neurons send two quite different types of signals: in the forward pass the activity, in the backward pass the error derivative.
- In real biological neural networks, the neuron responses are tuned by a synapse-change procedure that is **physically local** and thus describable by local mathematics. Obviously BP is not a local algorithm.

<https://www.youtube.com/watch?v=qIEfJ60BGj8&t=2425s>

Training Procedure

The main idea of the algorithm can be summarized as follow:

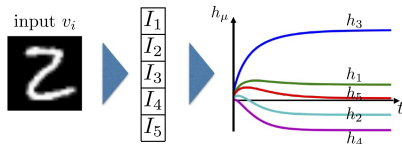
1. Create a fully connected NN, composed by only one hidden layer.
2. The early layer is trained in a biological unsupervised way. We know that learning good feature detectors in early layers of artificial neural networks is crucial.
3. The second layer is trained with usual BP algorithm. Here indeed BP is a local algorithm, and so doesn't violate the fourth point mentioned above.
4. The main effort of this approach is to **engineer a plasticity rule** that leads to a good generalization performance of the neural network.



Unsupervised Phase (h-part)

In the h -part, the attention is all focused in learning the steady-state activation h_{∞} .

A 28×28 raw input image (here MNIST dataset is used) is converted into a flat vector of length 784, denoted as \mathbf{v} , then using the matrix of weights \mathbf{W} , we convert the pixels values into currents $I = \mathbf{W} \cdot \mathbf{v}$.



Using the following equation, the dynamic of the activities of the hidden neurons is simulated.

$$\tau \frac{dh_\mu}{dt} = I_\mu - w_{inh} \sum_{\nu \neq \mu} r(h_\nu) - h_\mu \quad (1)$$

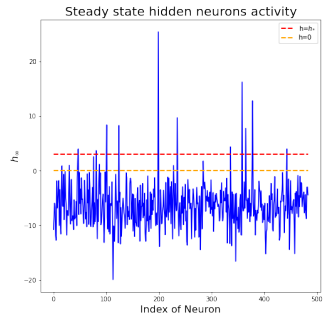
where $r(h_\mu) = \max(h_\mu, 0)$ are the corresponding **firing rates**. This equation has the structure of a ν -equation describing the capacitance membrane mechanism.

Unsupervised Phase (h-part)

The parameter $w_{inh} = 0.8$ sets the strength of the **global inhibition**. In our system the competition is **between neurons, not between patterns**. When one neuron becomes tuned to some pattern of inputs, the within-layer lateral inhibition keeps other neurons far from becoming selective to that same pattern. Because of the presence of w_{inh} our network strictly speaking stop to be feed-foward.

The parameter τ instead define the dynamical **time scale of presentation of an individual training example**.

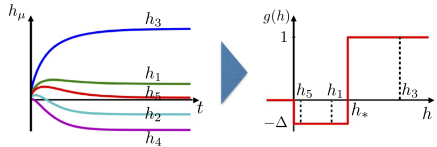
In particular the simulation requires 93ms (real time) with the chosen parameters.



Unsupervised Phase (W-part)

The values of h_∞ , taken as $h(t = t_{max})$, are then used as arguments in the non-linear learning activation function $g(h)$. Notice that here there is no need to have a differentiable activation function like sigmoid or tanh, because BP won't be used.

$$g(h) = \begin{cases} 0 & \text{if } h < 0 \\ -\Delta & \text{if } 0 \leq h < h_* \\ 1 & \text{if } h_* \leq h \end{cases} \quad (2)$$



The idea of having an activation function that is positive for activation above the threshold h_* and negative below, is inspired by the existence of LTP and LTD, meaning that in order to update $W_{\mu i}$ we are going to use both Hebbian and anti-Hebbian learning rules.

In the limit $\Delta \rightarrow 0$, anti-Hebbian rule is switched off.

Unsupervised Phase (W -part)

Once the hidden neurons reach stationarity, we would like to use h_∞ in order to update the weights W . The dynamic then is:

$$\tau_L \frac{dW_{\mu i}}{dt} = g(h_\mu) \left[v_i - \left(\sum_{k=1}^{N_{input}} W_{\mu k} v_k \right) W_{\mu i} \right] \quad (3)$$

where $i \in \{1, \dots, N_{input}\}$, while $\mu \in \{1, \dots, N_{hidden}\}$.

We have to ensure that $\tau \ll \tau_L$, where τ_L is the **time scale of the learning dynamics**. The chosen parameters allow to solve the ODE in 1.372 s.

The term $\mathbf{W} \cdot \mathbf{v}$ was introduced by the authors in order to have a **homeostatic mechanism** that ensure a limited growth of the synapses; indeed it's a modification of the usual normalization rule $\mathbf{W} \cdot \mathbf{1}$.

<https://www.youtube.com/watch?v=4lY-oAY0aQU>

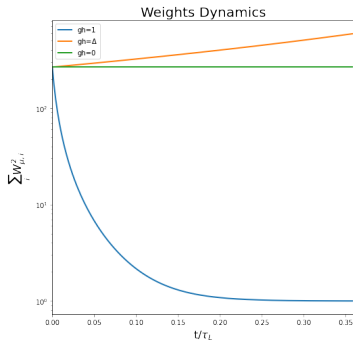
Convergence

Initially the weights are initialized from a standard normal distribution.

It can be shown that as $t \rightarrow \infty$ the weights \mathbf{W}_μ converge to the unit sphere, defined using the L2 norm. This statement is indeed verified by the simulation.

If a neuron has a value of h_∞ that falls in the anti-hebbian regime, those weights grow in an **unbounded way**.

In my simulation I wasn't able to ensure stable result with $\Delta = -0.4$, as mentioned by the paper, that's way I've changed it to $\Delta = -0.01$.



Unsupervised Training

The architecture used is composed by 784 neurons for the input layer and **484 neurons for the hidden layer** (2000 was mentioned in the paper). **1%** of the dataset will be used for training, (i.e only 600 MNIST images), composed by 60 images for each digit, in this way the unsupervised training is not biased by the training set.

The full unsupervised training requires 585s (~10min). Here on the right are shown 100 of the 484 feature maps; we can really appreciate how a bio-network is learning human-level features.



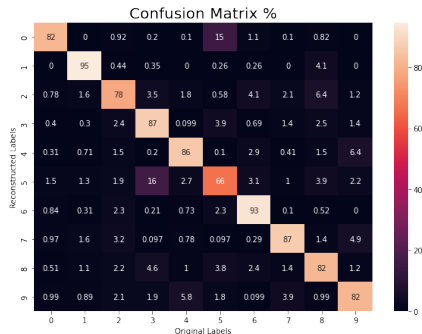
Supervised Training

Once we've trained the early layer, we **freeze** those weights and then we attach to our previous network another layer with 10 output neurons (for labels classification).

The full network could be embedded in skorch using PyTorch. Using 84% of the dataset as training set (~ 50000 images), implementing a cross-validation strategy with 5 folds, the supervised phase requires 10min to train 40 epochs.

SGD with learning rate $5 \cdot 10^{-3}$ was used as optimizer.

Here on the right the final confusion matrix, where the final accuracy is of **83.93%**.



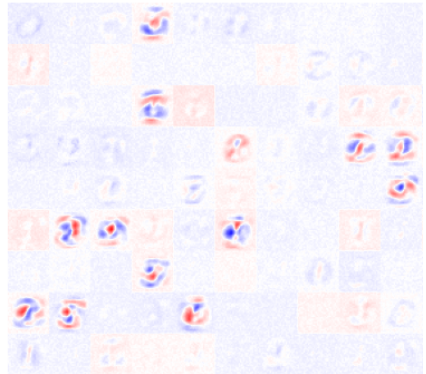
BP Supervised Training

If we use the same architecture, sigmoid as activation function in every layer, same hyper parameters, we achieve an accuracy of **82.65%**.

The good aspect of BP, is that in order to achieve 82.65% of accuracy it requires only **~ 30s**, while the bio-network it requires **~ 20min**, i.e it is 40 times faster.

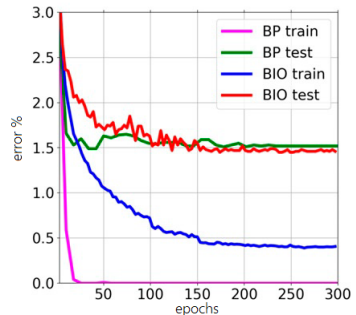
The cons is that if we look at the feature maps we obtain a mess, **we can't really understand what the NN is trying to learn**.

This paper is clearly a fundamental step toward ***Explainable DeepLearning***.



Conclusions

- From the above analysis seems that **bio-algorithms are slow**.
In the paper are reported some useful tricks and tips, in order to speed up the computation, like ranking, mini-batches and so on.
- The authors report a slightly better test error (1.46%) respect the standard one (1.5%). In this case the training is done using the full dataset.
- It's extremely important to remember that the **bio-network is performing an harder task** respect that one solved by the BP-network. In my implementation the **early layer is learned not only in an unsupervised way, but with less training data**.



Is there anything else?

The authors also try to apply this procedure to CIFAR-10 dataset, that is composed by 50000 32×32 RGB images.

Even if the network doesn't work well with this dataset, achieving a test error of 44.95%, what the network is learning is incredible. Indeed the unsupervised algorithm discovers not only the continuity of color in the data, but it displays human-level feature maps.



Open question: "Could be biological networks used as defense against adversarial attacks?"