Modeling in Neuroscience
Project Report
- Biologically Based Models of Cortical Cognition -

Clément Bonnet

Master MVA,
École Normale Supérieure Paris-Saclay, France
clement.bonnet16@gmail.com

Abstract

This work is based upon Randall O'Reilly's article entitled "Six principles for biologically based computational models of cortical cognition". It presents a set of principles to be gathered in one coherent framework, namely the Leabra algorithm. In this work, I explain the different principles, their interactions and consequences. Then, I confront them in some examples with rather small networks. I show that the Leabra algorithm is capable of learning and proves to be of great potential for future research.

1 Introduction

A significant number of principles have been developed for neural-network models of cognition. Some principles even underpin most of current neural networks and deep learning. Current feedforward back-propagation networks integrate two of the six principles brought by Randall O'Reilly, i.e. distributed representations and error-driven task learning.

This work is based upon O'Reilly's article, namely "Six principles for biologically based computational models of cortical cognition" [1], which provides a set of six principles integrated into a coherent overall framework.

In this work, I present the ideas introduced by Randall O'Reilly as well as the Leabra algorithm implementing these ideas. Then, I provide some experiments with this algorithm applied to small neural networks.

2 Principles for Biologically Based Models of Cortical Cognition

In this section, I attempt to present the work of Randall O'Reilly. He argues to integrate six principles into a coherent overall framework. The Leabra algorithm incorporates these principles, yet his work focuses on their history and their importance. An illustration borrowed from O'Reilly of the six principles is displayed in Figure 1.

He classifies the six principles described below into three categories: biological realism, architectural principles and learning principles.

Project report submitted in fulfillment of the requirements for the "Modelling in Neuroscience" course (École Normale Supérieure Paris-Saclay) taught by Jean-Pierre Nadal.

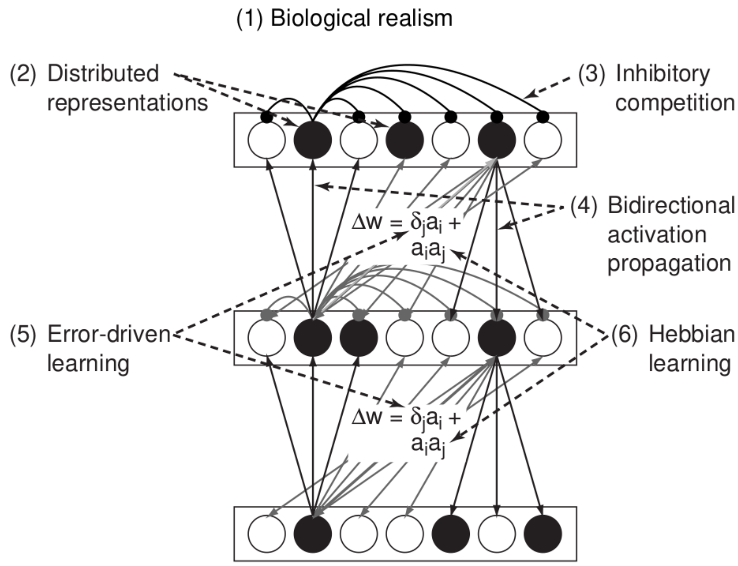


Figure 1: The six principles for biologically based models of cortical cognition instantiated in a neural network (source: O'Reilly).

2.1 Biological Realism

A fundamental goal of computational modeling in cognitive neuroscience is to understand how the brain (mainly the cortex) gives rise to cognition. Biological realism comes as a crucial principle in which models should be constrained and informed by biological properties of the cortex. Going even further, the principle states that computational mechanisms that violate biological properties should not be relied upon. Therefore, checking whether an algorithmic method is biologically plausible is a way to measure its potential.

2.2 Architectural Principles

2.2.1 Distributed Representations

It is widely believed that the cortex uses distributed representations to encode information. This means having essentially two features:

- To encode a representation, multiple active neuron-like processing units are involved;
- The same unit may be used to encode multiple representations.

One may observe that current artificial neural networks do integrate this principle as artificial neurons are stacked in layers to represent different patterns.

There are four functional benefits of having distributed representations, namely efficiency, robustness, accuracy and similarity.

- **Efficiency**: combinations of a small set of units can represent large quantities of information, so do letters when forming words.
- **Robustness** is achieved through the redundancy of having many units represent items. It decreases every unit's weight to the whole representation.
- **Accuracy** is improved as distributed representations can more accurately represent graded values through coarse coding.
- **Similarity** between items is easily represented with their shared units involved in their distributed representations.

2.2.2 Inhibitory Competition

There is inhibitory competition when mutual inhibition, thanks to inhibitory neurons, prevents all but a subset of neurons from being active at a time. It is to be noted that around 20% of cortex neurons are inhibitory interneurons. This enables the cortex to control the explosion of activations that may arise, for instance, when one suffers from epilepsy. Inhibitory competition has two main consequences.

- Through a selection process, it allows only the most strongly excited representations to emerge, identifying the most appropriate ones for the situation.
- Inhibitory competition allows for effective differentiation and distribution of representations as only selected ones are refined over time through learning.

Hence, inhibitory competition appears key for sparse distributed representations to emerge. To support this idea, it must be noted that the world can be described in terms of a large number of categories with a considerable number of exemplars per category.

2.2.3 Bidirectional Activation Propagation (Interactivity)

Feedforward neural networks allow information to flow in only one direction, from inputs to outputs.

It appears critical for information to flow through the network in both bottom-up and top-down directions. To allow this flow in both directions simultaneously, processing must be made of iterative steps. These steps establish a temporally-extended settling process that stabilizes the flow of information in both directions.

2.3 Learning Principles

The crucial question arising from learning is: what aspects of the environmental structure should be learned? Learning principles account for two core principles: task learning and Hebbian learning.

2.3.1 Error-driven Task Learning

The principle of task learning is the one that underpins the paradigm of supervised learning. It corresponds to minimizing the difference (error) between the desired outcome and what the network produces. The back-propagation algorithm enables such error to be minimized via the computation of its gradient.

The principle is rather simple as one should learn what enables them to succeed at the task. However, back-propagation is widely considered not biologically plausible for two main reasons. Firstly, the propagation of a global error signals may not be found in the brain. Secondly, the origin of the teaching signal that defines the error would be unknown.

Although back-propagation is considered to be not biologically plausible in feedforward networks, it may be so combined with bidirectional activation propagation. This is the idea behind the Generalized Recirculation Algorithm called GeneRec [2]. Instead of one error signal, it propagates two signals through the network and locally takes their difference at each unit. Therefore, the propagation can be divided in two phases: an expectation and an outcome phase.

- Expectation: the algorithm computes the expected consequences or correlates.
- Outcome: the network experiences actual consequences or correlates.

The difference between the expectation and the outcome make up the teaching signal that is then propagated throughout the network.

2.3.2 Hebbian Model Learning

Model learning intends to learn in an unsupervised way, without any teaching signals. It corresponds to the paradigms of self-organizing and unsupervised learning. The goal is to learn internal representations of statistical structures of the environment in a general way and not bound by a particular task.

Hebbian learning aims to represent the correlational structure of the environment as it encodes to which extend different things do co-occur. Through changes in synaptic strength, Hebbian learning is proven to be biologically based since NMDA-mediated long-term potentiation has this Hebbian property. The co-occurrence of items implies causal relationships between them, enabling a network to perform a Principal Component Analysis on the co-occurrence statistics of its inputs.

2.4 Interactions Among the Principles

Some principles may prevent others from working together. In this section, I describe some important interactions among them.

2.4.1 Interactivity and Noise

Interactivity, described as bidirectional activation propagation, interferes with the ability to exhibit independent contributions from context in an identification scenario. Yet, this problem can be solved using intrinsic variability noise.

This problematic interaction however shows that sometimes core principles contradict and thus do not work with one another. A meta principle that emerges from this observation is to integrate more principles so that they eventually fit altogether.

2.4.2 Distributed Representations and Competition

On one hand, distributed representations require multiple active units to describe an item. On the other hand, competition inhibits cooperativity and prevents many units from being active at a time. One may see that both principles are rather contradictory. A trade-off between both leads to a sparse representation where not too many units are active because of competition, but enough so that the representation is distributed. One caveat of the sparse representation is its complexity to analyze because of a combinatorial explosion.

There are multiple frameworks for such representations, namely the winner-take-all framework and variants or the independent units framework. However, they are not biologically plausible. Since the cortex is known to implement inhibitory competition and sparse distributed representations via inhibitory neurons, another framework called k-winners-take-all (KWTA) has been proposed and is proven to be biologically plausible. KWTA enforces competition while allowing for a sparse distributed representation. However, this framework is under-analyzed and not treated yet withing a probabilistic learning framework.

2.4.3 Learning Principles

Regarding learning, there are two levels of distinction. First, one may differ with respect to the computational objective of learning: task versus model learning. Then, there are the implementational mechanisms of learning such as error-driven or Hebbian. They both describe two camps that interact with one another. Indeed, error-driven learning can be used to learn a task-independent model of the environment, reaching model learning. Moreover, some Hebbian-like mechanisms can achieve quasi-error-driven learning.

The two camps may therefore seem rather similar, however they remain fundamentally different and it is believed that only error-driven learning is capable of learning arbitrary input-output mappings.

Standard artificial neural networks are underconstrained by the learning task, leading to too much variance in their output. Therefore, one may include into these networks a Hebbian bias towards representing co-occurrence statistics. As weight-decay is used as an inductive bias in back-propagation, adding a Hebbian bias may be a smarter way to include inductive biases.

3 Leabra Algorithm

LEABRA stands for Learning in an Error-driven and Associative, Biologically Realistic Algorithm. This algorithm demonstrates the six principles' mutual compatibility. Its key concepts are described in the sections below.

3.1 Point-Neuron Activation Function

The algorithm simplifies the neuron's geometry at a single point. It updates the membrane potential V_m as follows:

$$\frac{dV_m}{dt}(t) = \tau \sum_c g_c(t)\bar{g_c}(E_c - V_m(t))$$

With c a channel being one of the following: excitatory input, leak current or inhibitory input.

The communication to other cells j is achieved via a thresholded sigmoidal function with gain γ:

$$y_j(t) = \frac{1}{1 + \frac{1}{\gamma[V_m(t)-\Theta]_+}}$$

3.2 K-Winners-Take-All Inhibition (KWTA)

This framework allows to achieve sparse distributed representations. It uses a uniform level of inhibition for all units, whose consequence is that no more than k units get over the threshold.

The inhibitory current is given by its conductance g_i.

$$g_i = g^{\Theta}_{k+1} + q(g^{\Theta}_{k} - g^{\Theta}_{k+1}) \quad \text{with} \begin{cases} q & \approx 0.25 \\ g^{\Theta} & = \frac{\sum_{c \neq i} g_c \bar{g_c}(E_c - \Theta)}{\Theta - E_i} \end{cases}$$

3.3 Error-driven Learning

In Leabra, error-driven learning is implemented using a symmetric version of the biologically plausible GeneRec algorithm. Hence, it is equivalent to deterministic Boltzmann machines and Contrastive Hebbian Learning. The weight update is the following:

$$\Delta w^E_{ij} = x^+_i y^+_j - x^-_i y^-_j$$

With x_i being the sending unit and y_j the receiving one. The network settles in two phases: the outcome phase (with a $+$) and the expectation phase (with a $-$).

3.4 Hebbian Learning

In addition to the error-driven learning term, the Leabra algorithm also integrates a Hebbian learning term. For this purpose, it uses a variant of Oja normalization[3]:

$$\Delta w^H_{ij} = x_i y_j - y_j w_{ij} = y_j(x_i - w_{ij})$$

It can be seen as computing the expected value of the sending unit activity x_i conditional on the receiver's activity if treated like a binary variable with probability y_j. Hence, $w^H_{ij} \approx < x_i | y_j >$.

The final update is the combination of both weights updates with some rescaling.

$$\Delta w_{ij} \propto \Delta w^E_{ij} + \lambda \Delta w^H_{ij}$$

4 Experiments

I used a Python implementation of the Leabra algorithm called Leabra7 [4] to do some experiments with the algorithm. First, I present the case with just two neurons interacting with one another, then I showcase the task of pattern association, finally I demonstrate the algorithm can learn to classify a dataset, namely the Iris dataset.

4.1 Two Neurons

The simple case of one input neuron connected to one output neuron is first studied. The connection weight is set to 0.5 and the input activation is clamped to 1 so that it produces spikes.

The dynamics of the output unit for 250 cycles is shown in Figure 2.

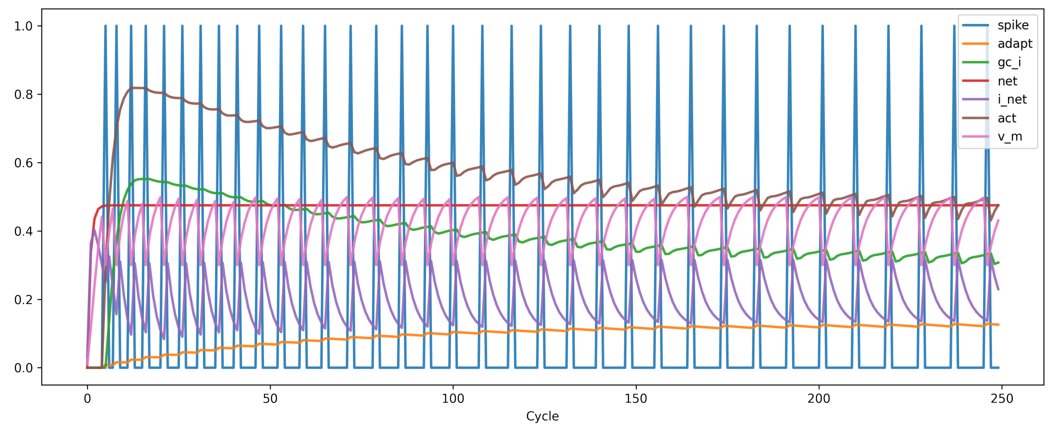


Figure 2: Experiment with two neurons.

As the time-integrated excitatory input net rises, so does the net current into the unit, i_{net}. This drives the unit's electric potential, v_m , above the spiking threshold, causing repeated spiking. When the unit spikes, the rate-coded activation value act increases, which would then be transmitted to downstream, postsynaptic neurons if any existed. The spiking triggers feedback inhibition, gc_i, from the layer. Over time, the unit's adaption current *adapt* slowly increases, representing the onset of a refractory period, and spiking slows. When spiking slows, the activation act and inhibition gc_i drop. (Adaptation from [4]).

4.2 Pattern Association

The goal of this experiment is to learn a mapping from a 4-dimensional pattern to a 2-d one. The patterns dataset consists of 4 patterns that are presented Figure 3.

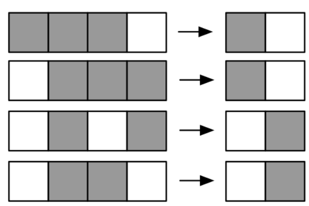


Figure 3: Pattern to learn

I compare two different networks that use both the LEABRA algorithm. The first network is feedforward and shallow while the second has a hidden layer and a backward connection from the output layer to the hidden layer. Their architectures are presented in Figure 4.

The learning curves for both networks are displayed in Figure 5. One may observe that the accuracy does not reach 100% for the shallow network as the input is not linearly separable in $\mathbb{R}^4$. It does not either for the deep network although it is theoretically possible. This may be explained since the training of the latter is much more sensitive to hyperparameter tuning and oscillations due to the feedback connection. Nevertheless, both architectures show some learning capabilities.

4.3 Iris Dataset Classification

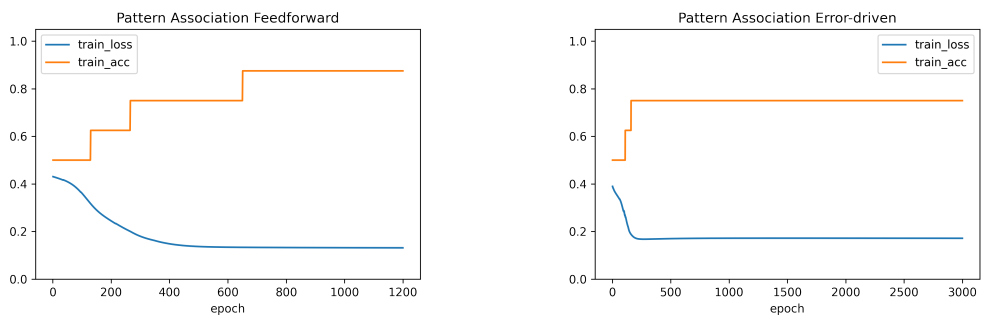
This last task is the hardest. The goal is to map flowers from the Iris dataset onto 3 different categories. I used quartile pre-processing as it is easier to deal with one-hot encodings.

For this task, I compared a traditional feedforward neural network to one that implements Leabra. They both have an input layer of 40 neurons, a hidden layer with 23 units and an output layer of 3 neurons for classification. PyTorch was used to train the first network while the implementation of the Leabra algorithm enabled to train the second network. The learning curves are displayed in Figure 6.

(a) Feedforward network

(b) Hidden-layer network

Figure 4: Network architectures



(a) Feedforward network

(b) Hidden-layer network

Figure 5: Pattern association learning

One may observe that the learning curves are rather similar in terms of both losses and accuracy. The accuracy is a little bit higher for the feedforward neural network and training is smoother than the Leabra network. It is to be noted that the implementation of Leabra is also order-of-magnitudes slower than PyTorch regarding both inference and training.

Although learning does work in this setting, one may improve these results by tuning some of the numerous hyperparameters of the Leabra algorithm. It would be interesting to analyze the convergence of neuron's activities with respect to the number of cycles at each epoch.

5 Conclusion

In this work, we have presented the six principles for biologically based models of cortical cognition introduced by Randall O'Reilly. We have analyzed the Leabra algorithm that proposes to fit these principles into a coherent overall framework.

We have shown with multiple examples that the Leabra algorithm works in practice, albeit being rather unstable when learning and very sensitive to hyperparameters. The theoretical analysis of the algorithm is complex and lacks understanding compared to the probabilistic framework used in statistical machine learning. Nevertheless, it has been demonstrated that this algorithm can learn in a biologically plausible fashion, presenting a great potential to improve research in cognitive neuroscience and machine learning.

Acknowledgments

I would like to thank Jean-Pierre Nadal for his teaching. The code for the experiments is available at `https://github.com/clement-bonnet/leabra-project`.

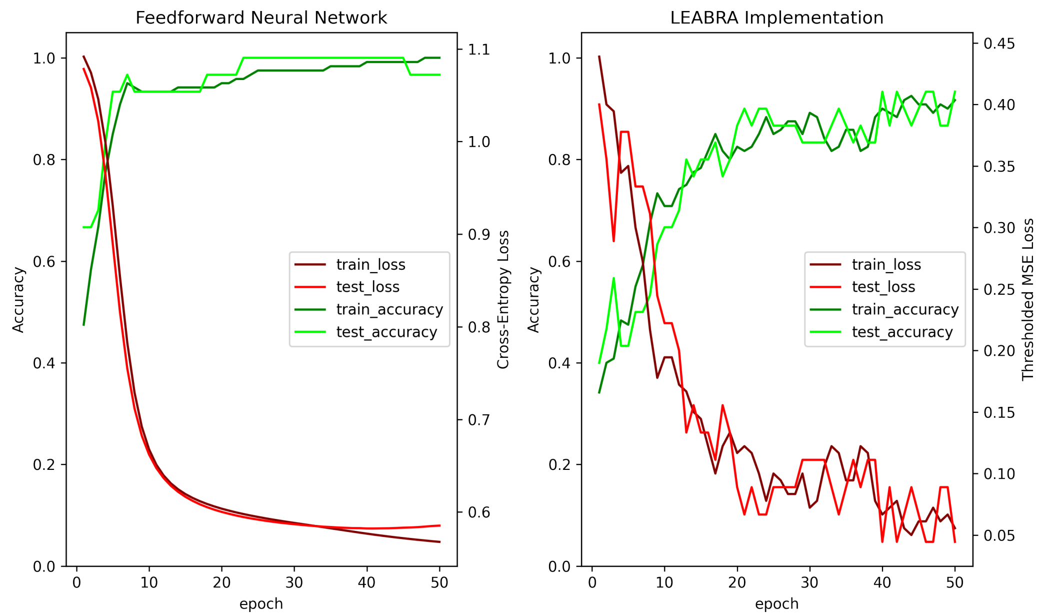


Figure 6: Classification of the Iris dataset using PyTorch on the left side and the Leabra algorithm for the plot on the right.

References

[1] Randall C. O'Reilly. "Six principles for biologically based computational models of cortical cognition". In: *Trends in Cognitive Sciences* 2.11 (1998), pp. 455–462. ISSN: 1364-6613. DOI: `https://doi.org/10.1016/S1364-6613(98)01241-8`. URL: `https://www.sciencedirect.com/science/article/pii/S1364661398012418`.

[2] R. C. O'Reilly. "Biologically Plausible Error-Driven Learning Using Local Activation Differences: The Generalized Recirculation Algorithm". In: *Neural Computation* 8.5 (1996), pp. 895–938. DOI: `10.1162/neco.1996.8.5.895`.

[3] Erkki Oja. "Simplified neuron model as a principal component analyzer". In: *Journal of Mathematical Biology* 15.3 (Nov. 1982), pp. 267–273. ISSN: 1432-1416. DOI: `10.1007/BF00275687`. URL: `https://doi.org/10.1007/BF00275687`.

[4] C. Daniel Greenidge, Noam Miller, and Kenneth A. Norman. "Leabra7: a Python package for modeling recurrent, biologically-realistic neural networks". In: *CoRR* abs/1809.04166 (2018). arXiv: `1809.04166`. URL: `http://arxiv.org/abs/1809.04166`.