Neuron in a Box

MacLeod, Christopher

Publication date:
2019

Neuron in a Box: A biologically feasible neural network learning paradigm

The aim of the "neuron in a box" (NIB) algorithm is to produce a flexible biologically feasible training paradigm - into which other learning techniques may be incorporated as required. It is based on isolating a neuron to discover all the possible influences on its learning.

Consider a single biological neuron - what are its possible learning mechanisms? The answers to this question may be explored by drawing the isolated neuron - a "neuron in a box", as shown in the figure below.

We can then write down all the *possible* parameters that could allow it to learn. After all, the neuron is only connected to other neurons and glia, so the only possible mechanisms involve either signals transmitted from other cells or chemical signals from the surrounding (interstitial) medium.

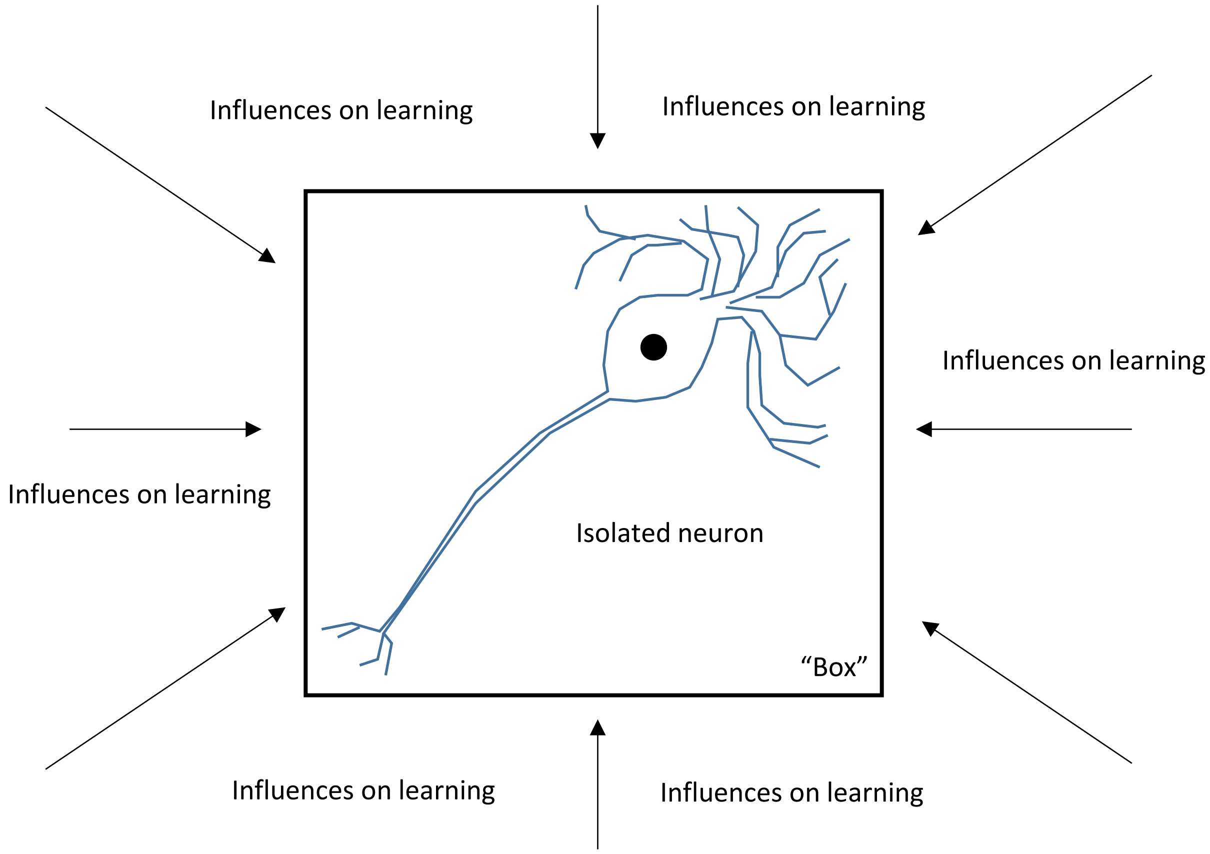

Figure: An Isolated "neuron in a box"

Some of the most important known parameters are:

a) Biochemical. The neuron is bathed in a 'soup' of intercellular fluid. Hormonal and other stimuli can affect this soup. For example, its constituents might change in the presence of hunger, pain, fright, the urge to mate, etc (or, in a simple system, just a good / bad signal). The signal would affect whole regions, not individual synapses. Let us call this component B.

b) Hebbian (or Anti-Hebbian). A synapse gets strengthened through use. The more activity it has, the stronger it becomes. This contribution effects each synapse individually and is therefore a matrix. Let us call this component *H*.

c) Synchronous (or Anti-Synchronous). A synapse gets strengthened when it is active at the same time as others (and weakened if it is not). Like the Hebbian contribution, this is therefore a matrix. Let us call this component *S*.

d) Mediated. One synapse (or a group of them) controls the strength of another one (or group).

$$M_{i,j} = aS_1 + bS_2 + cS_3 + \cdots etc$$

Where a, b, c, etc are constants defined by evolution (or learned) and S_1, S_2, S_3, etc, are synaptic strengths connected to the neuron. This effect may be difficult to achieve in practical terms and since there is no evidence for it from biology, you may wish to leave it out of the model.

The mathematical forms of these learning mechanisms have not been expounded in exhaustive detail above because they are reasonably obvious. One of the advantages of this technique is that the programmer may choose to add one of the traditional artificial learning methods to (or instead of) those detailed above and if newly discovered or invented mechanisms become apparent, they can be added as well. The total learning contribution to the weight matrix could be expressed as:

$$\boldsymbol{W}^{+} = \boldsymbol{W} + \eta(c_1 B(c_2\boldsymbol{H} + c_3\boldsymbol{S}))$$

Where W^{+} is the new (updated) weight matrix, η is the learning rate, c_n is the sensitivity to the individual learning types and $\boldsymbol{W}$ is the old weight matrix.

These mechanisms are highly dependent on network topology - after all, a synchronous or mediated learning strategy will only work with neurons that are placed in the correct positions within networks. It is easy to see that in other positions they could have no effect or cause the network to deviate away from the required response. They are therefore particularly suitable for use with networks whose topology is defined using evolutionary mechanisms. This topology dependence may be the reason why it has been so difficult to implement biologically feasible learning mechanisms in ANNs.

The training algorithm sets up the parameters listed above using an evolutionary algorithm and then the network is allowed to perform its tasks in the chosen system (which required an adaptive or learned response) for a set period of time. The network's performance at the end of this period is its fitness. Obviously, this system is particularly designed for systems which have an inherent need for flexible learning algorithms in their operation.

References:

The algorithm was first outlined in:

MACLEOD, C., MCMINN, D., REDDIPOGU, A. B., CAPANNI, N. F. and MAXWELL, G. M., 2001. Evolution and devolved action: towards the evolution of systems. In: Appendix B of MCMINN, D. Using evolutionary artificial neural networks to design hierarchical animat nervous systems, Ph. D. thesis. Aberdeen : Robert Gordon University.

https://openair.rgu.ac.uk/handle/10059/419

And was used in experiment into evolutionary modular neural networks for robotic control, outlined in:

MACLEOD, C., 2010. Practical algorithms for incremental growth. Aberdeen: Robert Gordon University.

https://openair.rgu.ac.uk/handle/10059/471