# Backprop Diffusion is Biologically Plausible

Marco Gori, Alessandro Betti

# Backprop Diffusion is Biologically Plausible

**Alessandro Betti**[1], **Marco Gori**[1,2]
[1]DIISM, University of Siena, Siena, Italy
[2]Maasai, Universitè Côte d'Azur, Nice, France
{alessandro.betti2,marco.gori}@unisi.it

## Abstract

The Backpropagation algorithm relies on the abstraction of using a neural model that gets rid of the notion of time, since the input is mapped instantaneously to the output. In this paper, we claim that this abstraction of ignoring time, along with the abrupt input changes that occur when feeding the training set, are in fact the reasons why, in some papers, Backprop biological plausibility is regarded as an arguable issue. We show that as soon as a deep feedforward network operates with neurons with time-delayed response, the backprop weight update turns out to be the basic equation of a biologically plausible diffusion process based on forward-backward waves. We also show that such a process very well approximates the gradient for inputs that are not too fast with respect to the depth of the network. These remarks somewhat disclose the diffusion process behind the backprop equation and leads us to interpret the corresponding algorithm as a degeneration of a more general diffusion process that takes place also in neural networks with cyclic connections.

## 1 Introduction

Backpropagation enjoys the property of being an optimal algorithm for gradient computation, which takes $\Theta(m)$ in a feedforward network with $m$ weights [8, 9]. It is worth mentioning that the gradient computation with classic numerical algorithms would take $O(m^2)$, which clearly shows the impressive advantage that is gained for nowadays big networks. However, since its conception, Backpropagation has been the target of criticisms concerning its biological plausibility. Stefan Grossberg early pointed out the *transport problem* that is inherently connected with the algorithm. Basically, for each neuron, the delta error must be "transported" for updating the weights. Hence, the algorithm requires each neuron the availability of a precise knowledge of all of its downstream synapses. Related comments were given by F. Crick [6], who also pointed out that backprop seems to require rapid circulation of the delta error back along axons from the synaptic outputs. Interestingly enough, as discussed in the following, this is consistent with the main result of this paper. A number of studies have suggested solutions to the weight transport problem. Recently, Lillicrap et al [11] have suggested that random synaptic feedback weights can support error backpropagation. However, any interpretation which neglects the role of time might not fully capture the essence of biological plausibility. The intriguing marriage between energy-based models with object functions for supervision that gives rise to Equilibrium Propagation [12] is definitely better suited to capture the role of time. Based the full trust on the role of temporal evolution, in [1], it is pointed out that, like other laws of nature, learning can be formulated under the framework of variational principles.

This paper springs out from recent studies on the problem of learning visual features [3, 4, 2] and it was also stimulated by a nice analysis on the interpretation of Newtonian mechanics equations in the variational framework [10]. It is shown that when shifting from algorithms to laws of learning, one can clearly see the emergence of the biological plausibility of Backprop, an issue that has been controversial since its spectacular impact. We claim that the algorithm does represent a sort of degen-

Preprint. Under review.

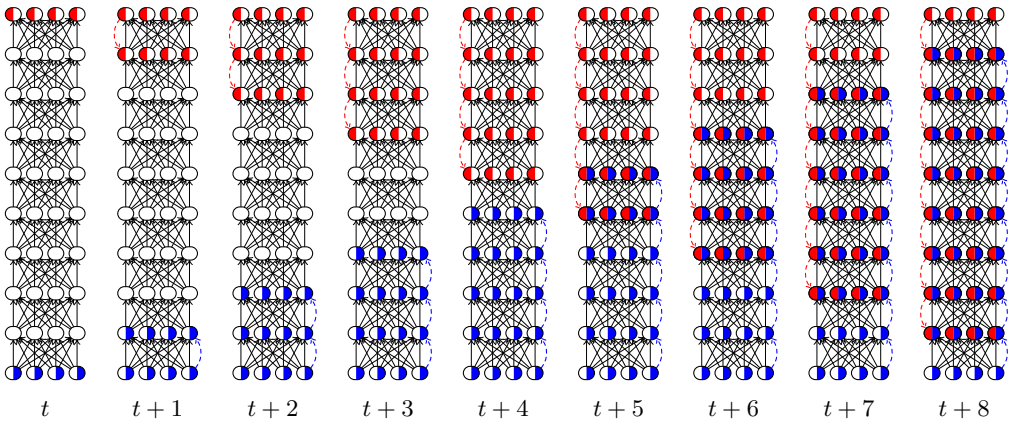



Figure 1: Forward and backward waves on a ten-level network when the input and the supervision are kept constant. When selecting a certain frame—defined by the time index—the gradient can consistently be computed by Backpropagation on "red-blue neurons".

eration of a natural spatiotemporal diffusion process that can clearly be understood when thinking of perceptual tasks like speech and vision, where signals possess smooth properties. In those tasks, instead of performing the forward-backward scheme for any frame, one can properly spread the weight update according to a diffusion scheme. While this is quite an obvious remark on parallel computation, the disclosure of the degenerate diffusion scheme behind Backprop, sheds light on its biological plausibility. The learning process that emerges in this framework is based on complex diffusion waves that, however, is dramatically simplified under the feedforward assumption, where the propagation is split into forward and backward waves.

## 2 Backprop diffusion

In this section we consider multilayered networks composed of $L$ layers of neurons, but the results can easily be extended to any feedforward network characterized by an acyclic path of interconnections. The layers are denoted by the index $l$, which ranges from $l = 0$ (input layer) to $l = L$ (output layer). Let $W_l$ be the layer matrix and $x_{t,l}$ be the vector of the neural output at layer $l$ corresponding to discrete time $t$. Here we assume that the network carries out a computation over time, so as, instead of regarding the forward and backward steps as instantaneous processes, we assume that the neuronal outputs follow the time-delay model:

$$x_{t+1,l+1} = \sigma(W_l x_{t,l}), \tag{1}$$

where $\sigma(\cdot)$ is the neural non-linear function. In doing so, when focussing on frame $t$ the following forward process takes place in a deep network of $L$ layers:

$$\begin{cases} x_{t+1,1} = \sigma(W_0 x_{t,0}) \\ x_{t+2,2} = \sigma(W_1 x_{t+1,1}) = \sigma(W_1 \sigma(W_0 x_{t,0})) \\ \qquad \vdots \\ x_{t+L,L} = \sigma(W_{L-1} x_{t,L-1}) = \ldots = \sigma(W_{L-1} \sigma(W_{L-2} \ldots \sigma(W_0 x_{t,0}) \ldots)). \end{cases} \tag{2}$$

Hence, the input $x_{t,0}$ is forwarded to layers $1, 2, \ldots, L$ a time $t+1, t+2, \ldots, t+L$, respectively, which can be regarded as a forward wave. We can formally state that input $u_t := x_{t,0}$ is forwarded to layer $\kappa$ by the operator $\overset{\kappa}{\rightarrow}$, that is $\overset{\kappa}{\rightarrow} u_t = x_{t+\kappa,\kappa}$. Likewise, when inspired by the backward step

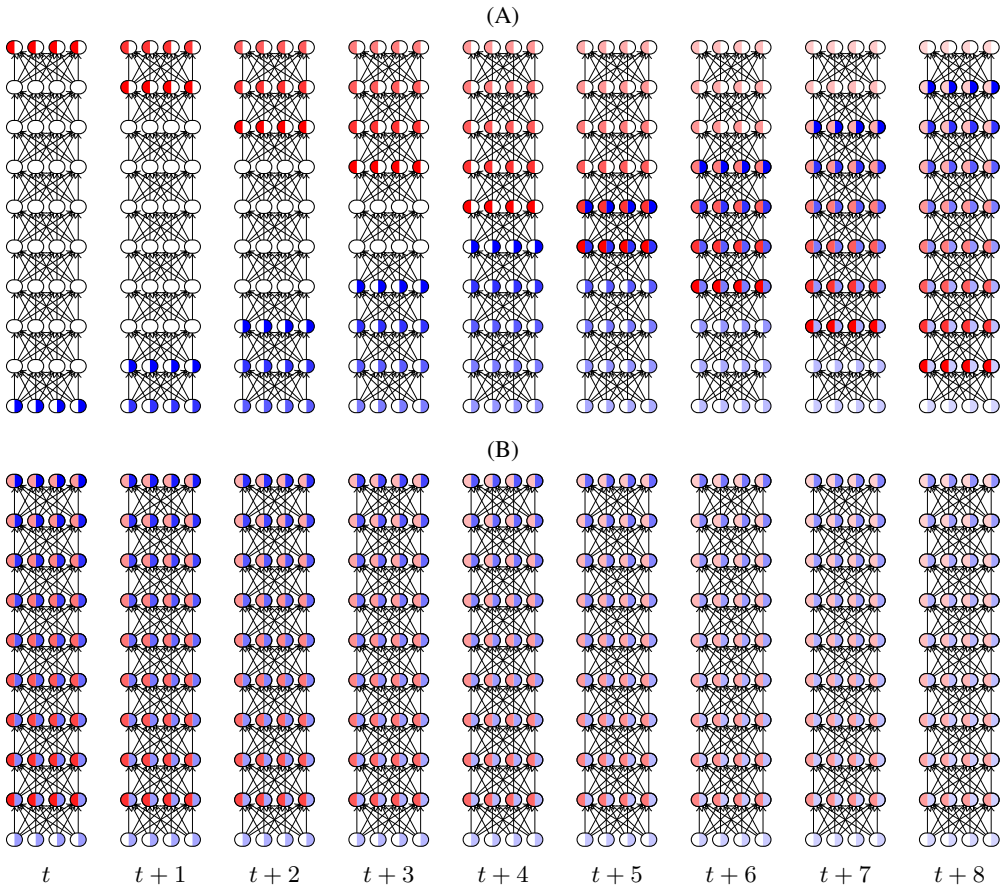



Figure 2: Forward and backward waves on a ten-level network with slowly varying input and supervision. In (A) it is shown how the input and backward signals fill the neurons; the wavefronts of the waves are clearly visible. Figure (B) instead shows the same diffusion process in stationary conditions when the neurons are already filled up. In this quasi-stationary condition the Backpropagation diffusion algorithm very well approximates the gradient computation. In particular for $l_s = (L+1)/2 = (9+1)/2 = 5$ there is a perfect backprop synchronization and the gradient is correctly computed.

of Backpropagation, we can think of back-propagating the delta error $\delta_{t,L}$ on the output as follows:

$$\begin{cases} \delta_{t+1,L-1} = \sigma'_{L-1} W_{L-1}^T \delta_{t,L} \\ \delta_{t+2,L-2} = \sigma'_{L-2} W_{L-2}^T \delta_{t+1,L-1} = (\sigma'_{L-2} W_{L-2}^T) \cdot (\sigma'_{L-1} W_{L-1}^T) \cdot \delta_{t,L} \\ \qquad\qquad \vdots \\ \delta_{t+L-1,1} = \sigma'_1 W_1^T \delta_{t+L-2,2} = \displaystyle\prod_{\kappa=1}^{L-1} \sigma'_\kappa W_\kappa^T \cdot \delta_{t,L} \end{cases}$$

Like for $x_{t,l}$, we can formally state that the output delta error $\delta_{t,L}$ is propagated back by the operator $\overset{L-\kappa}{\longleftarrow}$ defined by $\delta_{t,L} \overset{\kappa}{\longleftarrow} := \delta_{t+\kappa,L-\kappa}$. The following equation is still formally coming from Backpropagation, since it represents the classic factorization of forward and backward terms:

$$g_{t,l} = \delta_{t,l} \cdot x_{t,l-1} = \left( \overset{l-1}{\longrightarrow} u_{t-l+1} \right) \cdot \left( \delta_{t-L+l,L} \overset{L-l}{\longleftarrow} \right) \tag{3}$$

Clearly, $g_{t,l}$ is the result of a diffusion process that is characterized by the interaction of a forward and of a backward wave (see Fig. 1). This is a truly local spatiotemporal process which is definitely biologically plausible. Notice that if $L$ is odd then for $l_s = (L+1)/2$ we have a perfect backprop synchronization between the input and the supervision, since in this case the number of forward steps $l-1$ equals the number of backward steps $L-l$. Clearly, the forward-backward wave synchronization

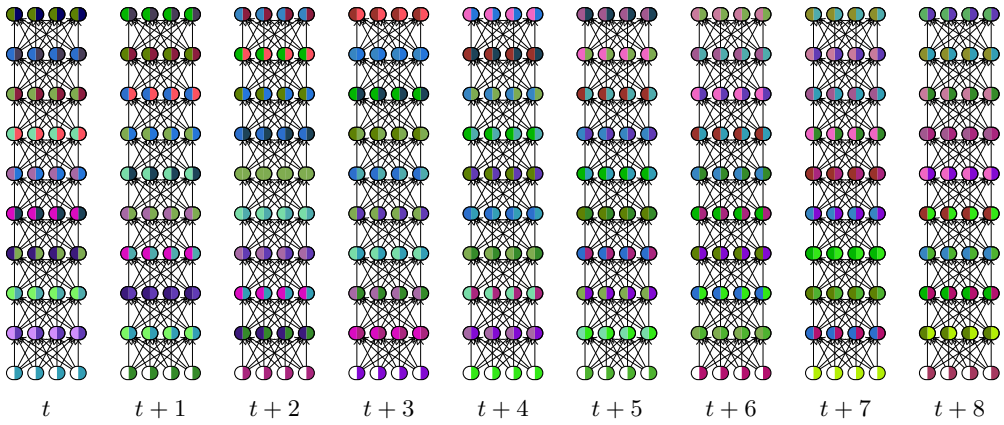



Figure 3: Forward and backward waves on a ten-level network with rapidly changing signal. In this case the Backpropagation diffusion equation does not properly approximate the gradient. However, also in this case, for $l_s = 5$ we have perfect Backpropagation synchronization with correct syncronization of the gradient.

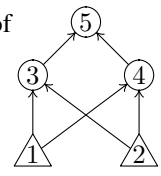
| Learning | Mechanics | Remarks |
|---|---|---|
| $(W, x)$ | $u$ | Weights and neuronal outputs are interpreted as generalized coordinates. |
| $(\dot{W}, \dot{x})$ | $\dot{u}$ | Weight variations and neuronal variations are interpreted as generalized velocities. |
| $\mathscr{A}(x, W)$ | $\mathscr{S}(u)$ | The cognitive action is the dual of the action in mechanics. |

Table 1: Links between learning theory and classical mechanics.

takes place for $L = 1$, which is a trivial case in which there is no wave propagation. The next case of perfect synchronization is for $L = 3$. In this case, the two hidden layers are involved in one-step of forward-backward propagation. Notice that the perfect synchronization comes with one step delay in the gradient computation. In general, the computation of the gradient in the layer of perfect synchronization is delayed of $l_s - 1 = (L-1)/2$. For all other layers, Eq. 3 turns out to be an approximation of the gradient computation, since the forward and backward waves meet in layers of no perfect synchronization. We can promptly see that, as a matter of fact, synchronization approximatively holds whenever $u_t$ is not too fast with respect to $L$ (see Fig. 2 and 3). The maximum mismatch between $l-1$ and $L-l$ is in fact $L-1$, so as if $\Delta t$ is the quantization interval required to perform the computation over a layer, good synchronization requires that $u_t$ is nearly constant over intervals of length $\tau_s = (L-1)\Delta t$. For example, a video stream, which is sampled at $f_v$ frames/sec requires to carry out the computation with time intervals bounded by $\Delta t = \tau_s/(L-1) = f_v/(L-1)$.

## 3 Lagrangian interpretation of diffusion on graphs

In the remainder of the paper we show that the forward/backward diffusion of layered networks is just a special case of more general diffusion processes that are at the basis of learning in neural networks characterized by graphs with any pattern of interconnections. In particular we show that this naturally arises when formulating learning as a parsimonious constraint satisfaction problem. We use recent connections established between learning processes and laws of physics under the principle of least cognitive action [1].




In this paper we make the additional assumption of defining connectionist models of learning in terms of a set of constraints that turn out to be subsidiary conditions of a variational problem [7]. Let us consider the classic example of the feedforward network used to compute the XOR predicate. If we denote with $x^i$ the outputs of each neuron and with $w_{ij}$ the weigh associated with the arch $i \longrightarrow j$, then for the neural network in the figure we have $x^3 = \sigma(w_{31}x^1 + w_{32}x^2)$, $x^4 =$

$\sigma(w_{41}x^1 + w_{42}x^2)$ and $x^5 = \sigma(w_{53}x^3 + w_{54}x^4)$. Therefore in the $w - x$ space this compositional relations between the nodes variables can be regarded as constraints, namely $G^3 = G^4 = G^5 = 0$, where:

$$G^3 = x^3 - \sigma(w_{31}x^1 + w_{32}x^2), \quad G^4 = x^4 - \sigma(w_{41}x^1 + w_{42}x^2),$$
$$G^5 = x^5 - \sigma(w_{53}x^3 + w_{54}x^4).$$

In addition to these constraints we can also regard the way with which we assign the input values as additional constraints. Suppose we want to compute the value of the network on the input $x^1 = e^1$ and $x^2 = e^2$, where $e^1$ and $e^2$ are two scalar values; this two assignments can be regarded as two additional constraints $G^1 = G^2 = 0$ where $G^1 = x^1 - e^1, \quad G^2 = x^2 - e^2$. First of all let us describe the architecture of the models that we will address. Given a simple digraph $D = (V, A)$ of order $\nu$, without loss of generality, we can assume $V = \{1, 2, \ldots, \nu\}$ and $A \subset V \times V$. A neural network constructed on $D$ consists of a set of maps $i \in V \mapsto x^i \in \mathbf{R}$ and $(i, j) \in A \mapsto w_{ij} \in \mathbf{R}$ together with $\nu$ constraints $G^j(x, W) = 0 \; j = 1, 2, \ldots \nu$ where $(W)_{ij} = w_{ij}$. Let $\mathcal{M}_\nu(\mathbf{R})$ be the set of all $\nu \times \nu$ real matrices and $\mathcal{M}_\nu^\downarrow(\mathbf{R})$ the set of all $\nu \times \nu$ strictly lower triangular matrices over $\mathbf{R}$. If $W \in \mathcal{M}_\nu^\downarrow(\mathbf{R})$ we say that the NN has a feedforward structure. In this paper we will consider both feedforward NN and NN with cycles. The relations $G^j = 0$ for $j = 1, \ldots, \nu$ specify the computational scheme with which the information diffuses trough the network. In a typical network with $\omega$ inputs these constraints are defined as follows: For any vector $\xi \in \mathbf{R}^\nu$, for any matrix $M \in \mathcal{M}_\nu(\mathbf{R})$ with entries $m_{ij}$ and for any given $\mathcal{C}^1$ map $e\colon (0, +\infty) \to \mathbf{R}^\omega$ we define the constraint on neuron $j$ when the example $e(\tau)$ is presented to the network as

$$G^j(\tau, \xi, M) := \begin{cases} \xi^j - e^j(\tau), & \text{if } 1 \leq j \leq \omega; \\ \xi^j - \sigma(m_{jk}\xi^k) & \text{if } \omega < j \leq \nu, \end{cases} \tag{4}$$

where $\sigma\colon \mathbf{R} \to \mathbf{R}$ is of class $\mathcal{C}^2(\mathbf{R})$. Notice that the dependence of the constraints on $\tau$ reflects the fact that the computations of a neural network should be based on external inputs.

**Principle of Least Cognitive Action** Like in the case of classical mechanics, when dealing with learning processes, we are interested in the temporal dynamics of the variables exposed to the data from which the learning is supposed to happen. Depending on the structure of the matrix $M$, it is useful to distinguish between feedforward networks and networks with loops (recurrent neural networks). Let us therefore consider the functional

$$\mathscr{A}(x, W) := \int \frac{1}{2}(m_x|\dot{x}(t)|^2 + m_W|\dot{W}(t)|^2)\, \varpi(t)dt + \mathscr{F}(x, W), \tag{5}$$

with $\mathscr{F}(x, W) := \int F(t, x, \dot{x}, \ddot{x}, W, \dot{W}, \ddot{W})\, dt$ and $t \mapsto \varpi(t)$ a positive continuously differentiable function, subject to the constraints

$$G^j(t, x(t), W(t)) = 0, \qquad 1 \leq j \leq \nu, \tag{6}$$

where the map $G(\cdot, \cdot, \cdot)$ is taken as in Eq. (4). Let $(\frac{G_\xi}{G_M})$ be the Jacobian matrix of the constraints $G$ with respect to $x$ and $W$, where it is intended that the first $\nu$ rows contain the gradients of $G$ with respect to its second argument:

$$\left(\frac{G_\xi}{G_M}\right)_{ij} \equiv G^j_{\xi^i}, \quad \text{for } 1 \leq i \leq \nu.$$

Variational problems with subsidiary conditions can be tackled using the method of Lagrange multipliers to convert the constrained problem into an unconstrained one [7]). In order to use this method it is necessary to verify an independence hypothesis between the constraints; in this case we need to check that the matrix $(\frac{G_\xi}{G_M})$ is full rank. Interestingly, the following proposition holds true:

**Proposition 1.** *The matrix* $(\frac{G_\xi}{G_M}) \in \mathcal{M}_{(\nu^2+\nu)\times\nu}(\mathbf{R})$ *is full rank.*

*Proof.* First of all notice that if $(G_\xi)_{ij} = G^j_{\xi^i}$ is full rank also $(\frac{G_\xi}{G_M})$ has this property. Then, since

$$G^j_{\xi^i}(\tau, \xi, M) = \begin{cases} \delta_{ij}, & \text{if } 1 \leq j \leq \omega; \\ \delta_{ij} - \sigma'(m_{jk}\xi^k)m_{ji} & \text{if } \omega < j \leq \nu, \end{cases}$$

we immediately notice that $G^i_{\xi^i} = 1$ and that for all $i > j$ we have $G^i_{\xi^i} = 0$. This means that

$$(G^j_{\xi^i}(\tau, \xi, M)) = \begin{pmatrix} 1 & * & \cdots & * \\ 0 & 1 & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix},$$

which is clearly full rank. □

Notice that this result heavily depends on the assumption $W \in \mathcal{M}^{\downarrow}_{\nu}(\mathbf{R})$ (triangular matrix), which corresponds to feedforward architectures.

**Derivation of the Lagrangian multipliers—feedforward networks** Following the spirit of the principle of least cognitive action [1], we begin by deriving the constrained Euler-Lagrange (EL) equations associated with the functional (5) under subsidiary conditions (6) that refer to feedforward neural networks. The constrained functional is

$$\mathscr{A}^*(x, W) = \int \frac{1}{2}(m_x|\dot{x}(t)|^2 + m_W|\dot{W}(t)|^2)\varpi(t) - \lambda_j(t)G^j(t, x(t), W(t))\,dt + \mathscr{F}(x, W), \quad (7)$$

and its EL-equations thus read

$$-m_x\varpi(t)\ddot{x}(t) - m_x\dot{\varpi}(t)\dot{x}(t) - \lambda_j(t)G^j_{\xi}(x(t), W(t)) + L^x_F(x(t), W(t)) = 0; \quad (8)$$

$$-m_W\varpi(t)\ddot{W}(t) - m_W\dot{\varpi}(t)\dot{W}(t) - \lambda_j(t)G^j_M(x(t), W(t)) + L^W_F(x(t), W(t)) = 0, \quad (9)$$

where $L^x_F = F_x - d(F_{\dot{x}})/dt + d^2(F_{\ddot{x}})/dt^2$, $L^W_F = F_W - d(F_{\dot{W}})/dt + d^2(F_{\ddot{W}})/dt^2$ are the functional derivatives of $F$ with respect to $x$ and $W$ respectively (see [5]). An expression for the Lagrange multipliers is derived by differentiating two times the equations of the architectural constraints with respect to the time and using the obtained expression to substitute the second order terms in the Euler equations, so as we get:

$$\begin{aligned}\Big(\frac{G^i_{\xi^a}G^j_{\xi^a}}{m_x} + \frac{G^i_{m_{ab}}G^j_{m_{ab}}}{m_W}\Big)\lambda_j =& \varpi\big(G^i_{\tau\tau} + 2(G^i_{\tau\xi^a}\dot{x}^a + G^i_{\tau m_{ab}}\dot{w}_{ab} + G^i_{\xi^a m_{bc}}\dot{x}^a\dot{w}_{bc}) \\ &+ G^i_{\xi^a\xi^b}\dot{x}^a\dot{x}^b + G^i_{m_{ab}m_{cd}}\dot{w}_{ab}\dot{w}_{cd}\big) \\ &- \dot{\varpi}(\dot{x}^a G^i_{\xi^a} + \dot{w}_{ab}G^i_{m_{ab}}) + \frac{L^{x^a}_F G^i_{\xi^a}}{m_x} + \frac{L^{w_{ab}}_F G^i_{m_{ab}}}{m_W},\end{aligned} \quad (10)$$

where $G^i_{\tau}$, $G^i_{\tau\tau}$, $G^i_{\xi^a}$, $G^i_{\xi^a\xi^b}$, $G^i_{m_{ab}}$ and $G^i_{m_{ab}m_{cd}}$ are the gradients and the hessians of constraint (6).

Suppose now that we want to solve Eq. (8)–(9) with Cauchy initial conditions. Of course we must choose $W(0)$ and $x(0)$ such that $g_i(0) \equiv 0$, where we posed $g_i(t) := G^i(t, x(t), W(t))$, for $i = 1, \ldots, \nu$. However since the constraint must hold also for all $t \geq 0$ we must also have at least $g_i'(0) = 0$. These conditions written explicitly means

$$G^i_{\tau}(0, x(0), W(0)) + G^i_{\xi^a}(0, x(0), W(0))\dot{x}^a(0) + G^i_{m_{ab}}(0, x(0), W(0))\dot{w}_{ab}(0) = 0.$$

If the constraints does not depend explicitly on time it is sufficient to to choose $\dot{x}(0) = 0$ and $\dot{W}(0) = 0$, while for time dependent constraint this condition leaves

$$G^i_{\tau}(0, x(0), W(0)) = 0,$$

which is an additional constraint on the initial conditions $x(0)$ and $W(0)$ to be satisfied. Therefore one possible consistent way to impose Cauchy conditions is

$$\begin{aligned} &G^i(0, x(0), W(0)) = 0, \quad i = 1, \ldots, \nu; \\ &G^i_{\tau}(0, x(0), W(0)) = 0, \quad i = 1, \ldots, \nu; \\ &\dot{x}(0) = 0; \\ &\dot{W}(0) = 0. \end{aligned} \quad (11)$$

**Reduction to Backpropagation** To understand the behaviour of the Euler equations (8) and (9) we observe that in the case of feedforward networks, as it is well known, the constraints $G^j(t,x,W) = 0$ can be solved for $x$ so that eventually we can express the value of the output neurons in terms of the value of the input neurons. If we let $f^i_W(e(t))$ be the value of $x^{\nu-i}$ when $x^1 = e^1(t), \ldots, x^\omega = e^\omega(t)$, then the theory defined by (5) under subsidiary conditions (6) is equivalent, when $m_x = 0$ and $\varpi(t) = \exp(\vartheta t)$, to the unconstrained theory defined by

$$\int e^{\vartheta t}\Big(\frac{m_W}{2}|\dot{W}|^2 - \overline{V}(t,W(t))\Big)\,dt \tag{12}$$

where $\overline{V}$ is a loss function for which a possible choice is $\overline{V}(t,W(t)) := \frac{1}{2}\sum_{i=1}^{\eta}(y^i(t) - f^i_W(E(t)))^2$ with $y(t)$ an assigned target. The Euler equations associated with (12) are

$$\ddot{W}(t) + \vartheta\dot{W}(t) = -\frac{1}{m_W}\overline{V}_W(t,W(t)), \tag{13}$$

that in the limit $\vartheta \to \infty$ and $\vartheta m \to \gamma$ reduces to the gradient method

$$\dot{W}(t) = -\frac{1}{\gamma}\overline{V}_W(t,W(t)), \tag{14}$$

with learning rate $1/\gamma$. Notice that the presence of the term $\varpi(t)$ that we proposed in the general theory it is essential in order to have a learning behaviour as it responsible of the dissipative behaviour.

Typically the term $\overline{V}_W(t,W(t))$ in Eq. (14) can be evaluated using the Backpropagation algorithm; we will now show that Eq. (8)–(10) in the limit $m_x \to 0$, $m_W \to 0$, $m_x/m_W \to 0$ reproduces Eq. (14) where the term $\overline{V}_W(t,W(t))$ explicitly assumes the form prescribed by BP. In order to see this choose $\vartheta = \gamma/m_W$, $\varpi(t) = \exp(\vartheta t)$,

$$F(t,x(t),\dot{x}(t),\ddot{x}(t),W(t),\dot{W}(t),\ddot{W}(t)) = -e^{\vartheta t}V(t,x(t)),$$

and multiply both sides of Eq. (8)–(10) by $\exp(-\vartheta t)$; then take the limit $m_x \to 0$, $m_W \to 0$, $m_x/m_W \to 0$. In this limit Eq. (9) and Eq. (10) becomes respectively

$$\dot{W}_{ij} = -\frac{1}{\gamma}\sigma'(w_{ik}x^k)\delta_i x^j; \tag{15}$$

$$G^i_{\xi^a}G^j_{\xi^a}\delta_j = -V_{x^a}G^i_{\xi^a}, \tag{16}$$

where $\delta_j$ is the limit of $\exp(-\vartheta t)\lambda_j$. Because the matrix $G^i_{\xi^a}$ is invertible Eq. (16) yields

$$T\delta = -V_x, \tag{17}$$

where $T_{ij} := G^j_{\xi^i}$. This matrix is an upper triangular matrix thus showing explicitly the backward structure of the propagation of the delta error of the Backpropagation algorithm. Indeed in Eq. (15) the Lagrange multiplier $\delta$ plays the role of the delta error.

In order to better understand the perfect reduction of our approach to the backprop algorithm consider the following example. We simply consider a feedforward network with an input, an output and an hidden neuron. In this case the matrix $T$ is

$$T = \begin{pmatrix} 1 & -\sigma'(w_{21}x^1)w_{21} & 0 \\ 0 & 1 & -\sigma'(w_{32}x^2)w_{32} \\ 0 & 0 & 1 \end{pmatrix}.$$

Then according to Eq. (17) the Lagrange multipliers are derived as follows $\delta_3 = -V_{x^3}, \delta_2 = \sigma'(w_{32}x^2)w_{32}\delta_3$, and $\delta_1 = \sigma'(w_{21}x^1)w_{21}\delta_2$. This is exactly the Backpropagation formulas for the delta error. Notice that in this theory we also have an expression for the multipliers of the input neurons, even though, in this case, they are not used to update the weights (see Eq. (15)).

**Diffusion on cyclic graphs** The constraint-based analysis carried out so far assumes holonomic constraints, whereas the claim of this paper is that we cannot neglect temporal dependencies, which leads to expressing neural models by non-holonomic constraints. In doing so, we go beyond the constraints expressed by Eq. (6) which imply an infinite velocity of diffusion of information. Since

we are stressing the importance of time in learning processes, it is natural to assume that the velocity of information diffusion through a network is finite. In the discrete setting of computation this is reflected by the model 1 discussed in Section 2. A simple classic translation of this constraint in continuous time is

$$c^{-1}\dot{x}^i(t) = -x^i(t) + \sigma(w_{ik}(t)x^k(t)), \tag{18}$$

where $c > 0$ can be interpreted as a "diffusion speed". In stationary situation indeed Eq. (18) coincide with the usual neuron equation in Eq. (6). However, we are in front of a much more complicated constraint, since not only it involves the variables $x$ and $w$, but also their derivatives. Such constraints are called non-holonomic constraints. Now we snow show how to determine the stationarity conditions of the functional[1]

$$\mathscr{A}(x, W) = \int \left(\frac{m_x}{2}|\dot{x}(t)|^2 + \frac{m_W}{2}|\dot{W}(t)|^2 + F(t, x, W)\right) \varpi(t)\, dt$$

under the nonholomic constraints

$$G^i(t, x(t), W(t), \dot{x}(t)) := c^{-1}\dot{x}^i(t) + x^i(t) - \sigma(w_{ik}(t)x^k(t)) = 0, \qquad i = 1, \dots, r, \tag{19}$$

by making use of the rule of the Lagrange multipliers. As usual we consider the stationary points of the functional

$$\begin{aligned}\mathscr{A}^*(x, W) = &\int \left(\frac{m_x}{2}|\dot{x}(t)|^2 + \frac{m_W}{2}|\dot{W}(t)|^2 + F(t, x, W)\right) \varpi(t)\, dt \\ &- \int \lambda_j(t) G^j(t, x(t), W(t), \dot{x}(t))\, dt.\end{aligned}$$

The Euler equations for $\mathscr{A}^*$ are

$$\begin{aligned}&c^{-1}\dot{x}^i(t) + x^i(t) - \sigma(w_{ik}(t)x^k(t)) = 0; \\ &- m_W\varpi(t)\ddot{W}(t) - m_W\dot{\varpi}\dot{W}(t) - \lambda_j G^j_M(t, x(t), W(t), \dot{x}(t)) + \varpi(t)F_W(t, x(t), W(t)) = 0; \\ &- m_x\varpi(t)\ddot{x}(t) - m_x\dot{\varpi}\dot{x}(t) + \dot{\lambda}_j G^j_p(t, x(t), W(t), \dot{x}(t)) + \lambda_j \frac{d}{dt} G^j_p(t, x(t), W(t), \dot{x}(t)) \\ &\quad - \lambda_j G^j_\xi(t, x(t), W(t), \dot{x}(t)) + \varpi(t)F_x(t, x(t), W(t)) = 0.\end{aligned}$$

Again, if we assume that $F(t, x, W) = -V(t, x)$, $\varpi(t) = e^{\vartheta t}$, $\delta_j = e^{-\vartheta t}\lambda_j$ and we explicitly use the expression for $G^j_p$ we get

$$\begin{aligned}&c^{-1}\dot{x}^i(t) + x^i(t) - \sigma(w_{ik}(t)x^k(t)) = 0; \\ &- m_W\ddot{W}(t) - m_W\vartheta\dot{W}(t) - \delta_j G^j_M(t, x(t), W(t), \dot{x}(t)) = 0; \\ &- m_x\ddot{x}(t) - m_x\vartheta\dot{x}(t) + c^{-1}\dot{\delta} - \delta_j G^j_\xi(t, x(t), W(t), \dot{x}(t)) - V_\xi(t, x(t)) = 0.\end{aligned}$$

This systems of equations can be better interpreted in the limit in which we recovered the backprop rule in Section 4; that is to say in the limit $m_x \to 0$, $m_x \to 0$ and $m_x/m_W \to 0$, $\theta \to \infty$ and $\theta m_W = \gamma$ fixed. Under this conditions, we have the following further reduction

$$\begin{aligned}&c^{-1}\dot{x}^i(t) + x^i(t) - \sigma(w_{ik}(t)x^k(t)) = 0; \\ &\dot{W}(t) = -\frac{1}{\gamma}\delta_j(t) G^j_M(t, x(t), W(t), \dot{x}(t)); \\ &c^{-1}\dot{\delta}(t) = \delta_j(t) G^j_\xi(t, x(t), W(t), \dot{x}(t)) + V_\xi(t, x(t)).\end{aligned} \tag{20}$$

The equation that defines the Lagrange multipliers $\delta$ is now a differential equation that explicitly gives us the correct updating rule instead of a instantaneous equation that must be solved for each $t$. As the diffusion speed becomes infinite ($c \to +\infty$) Eq. (20) reproduce Backpropagation, which is consistent with the intuitive explanation that arises from Fig. 1, 2, 3.

[1]Notice that here we are overloading the symbol $F$ since in this section we assume that the Lagrangian $F$ depends only on $t$, $x$ and $W$.

## 4 Conclusions

In this paper we have shown that the longstanding debate on the biological plausibility of Backpropagation can simply be addressed by distinguishing the forward-backward local diffusion process for weight updating with respect to the algorithmic gradient computation over all the net, which requires the transport of the delta error through the entire graph. Basically, the algorithm expresses the degeneration of a biologically plausible diffusion process, which comes from the assumption of a static neural model. The main conclusion is that, more than Backpropation, the appropriate target of the mentioned longstanding biological plausibility issues is the assumption of an instantaneous map from the input to the output. The forward-backward wave propagation behind Backpropagation, which is in fact at the basis of the corresponding algorithm, is proven to be local and definitely biologically plausible. This paper has shown that an opportune embedding in time of deep networks leads to a natural interpretation of Backprop as a diffusion process which is fully local in space and time. The given analysis on any graph-based neural architectures suggests that spatiotemporal diffusion takes place according to the interactions of forward and backward waves, which arise from the environmental interaction. While the forward wave is generated by the input, the backward wave arises from the output; the special way in which they interact for classic feedforward network corresponds to the degeneration of this diffusion process which takes place at infinite velocity. However, apart from this theoretical limit, Backpropagation diffusion is a truly local process.

## Broader Impact

Our work is a foundational study. We believe that there are neither ethical aspects nor future societal consequences that should be discussed.

## References


[1] Alessandro Betti and Marco Gori. The principle of least cognitive action. *Theor. Comput. Sci.*, 633:83–99, 2016.

[2] Alessandro Betti and Marco Gori. Convolutional networks in visual environments. *CoRR*, abs/1801.07110, 2018.

[3] Alessandro Betti, Marco Gori, and Stefano Melacci. Cognitive action laws: The case of visual features. *IEEE transactions on neural networks and learning systems*, 2019.

[4] Alessandro Betti, Marco Gori, and Stefano Melacci. Motion invariance in visual environments. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2009–2015. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[5] R Courant and D Hilbert. *Methods of Mathematical Physics*, volume 1, page 190. Interscience Publ., New York and London, 1962.

[6] Francis Crick. The recent excitement about neural networks. *Nature*, 337:129–132, 1989.

[7] M. Giaquinta and S. Hildebrand. *Calculus of Variations I*, volume 1. Springer, 1996.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[9] Marco Gori. *Machine Learning: A Constrained-Based Approach*. Morgan Kauffman, 2018.

[10] Matthias Liero and Ulisse Stefanelli. A new minimum principle for lagrangian mechanics. *Journal of Nonlinear Science*, 23:179–204, 2013.

[11] T.P. Lillicrap, D. Cownden, D.B. Twed, and C.J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, Jan 2016.

[12] Benjamin Scellier and Yoshua Bengio. Equivalence of equilibrium propagation and recurrent backpropagation. *Neural Computation*, 31(2), 2019.