LETTER **Communicated by Douglas Tweed**

Equivalence of Equilibrium Propagation and Recurrent Backpropagation

Benjamin Scellier
benjamin.scellier@polytechnique.edu
University of Montreal, Montreal, Quebec, H3T 1N8, Canada

Yoshua Bengio
yoshua.bengio@mila.quebec
University of Montreal, Montreal, Quebec, H3T 1N8, Canada, and CIFAR

Recurrent backpropagation and equilibrium propagation are supervised learning algorithms for fixed-point recurrent neural networks, which differ in their second phase. In the first phase, both algorithms converge to a fixed point that corresponds to the configuration where the prediction is made. In the second phase, equilibrium propagation relaxes to another nearby fixed point corresponding to smaller prediction error, whereas recurrent backpropagation uses a side network to compute error derivatives iteratively. In this work, we establish a close connection between these two algorithms. We show that at every moment in the second phase, the temporal derivatives of the neural activities in equilibrium propagation are equal to the error derivatives computed iteratively by recurrent backpropagation in the side network. This work shows that it is not required to have a side network for the computation of error derivatives and supports the hypothesis that in biological neural networks, temporal derivatives of neural activities may code for error signals.

1 Introduction

In deep learning, the backpropagation algorithm used to train neural networks requires a side network for the propagation of error derivatives, which is widely seen as biologically implausible (Crick, 1989). One fascinating hypothesis, first formulated by Hinton and McClelland (1988), is that in biological neural networks, error signals could be encoded in the temporal derivatives of the neural activities. This allows for error signals to be propagated in the network via the neuronal dynamics itself, without the need for a side network. Neural computation would correspond to both inference and error backpropagation. The work presented in this letter supports this hypothesis.

In section 2, we present the machine learning setting we are interested in. The neurons of the network follow the gradient of an energy function, such

Neural Computation **31, 312–329** (2019)
doi:10.1162/neco_a_01160

as the Hopfield energy (Cohen & Grossberg, 1983; Hopfield, 1984). Energy minima correspond to preferred states of the model. At prediction time, inputs are clamped, and the network relaxes to a fixed point, corresponding to a local minimum of the energy function. The prediction is then read out on the output neurons. This corresponds to the first phase of the algorithm. The goal of learning is that of minimizing the cost at the fixed point, called the objective.

Section 3 presents recurrent backpropagation (Almeida, 1987; Pineda, 1987), an algorithm that computes the gradient of the objective. In the second phase of recurrent backpropagation, an iterative procedure computes error derivatives.

In section 4, we present equilibrium propagation (Scellier & Bengio, 2017), another algorithm that computes the gradient of the objective. In the second phase of equilibrium propagation, when the target values for output neurons are observed, the output neurons are nudged toward their targets, and the network starts a second relaxation phase toward a second but nearby fixed point that corresponds to slightly smaller prediction error. The gradient of the objective can be computed based on a contrastive Hebbian learning rule at the first fixed point and second fixed point.

Section 5 (in particular, theorem 3) constitutes the main contribution of our work. We establish a close connection between recurrent backpropagation and equilibrium propagation. We show that at every moment in the second phase of equilibrium propagation, the temporal derivative of the neural activities code (i.e., are equal to) intermediate error derivatives, which recurrent backpropagation computes iteratively. Our work shows that one does not require a special computational path for the computation of the error derivatives in the second phase; the same information is available in the temporal derivatives of the neural activities. Furthermore we show that in equilibrium propagation, halting the second phase before convergence to the second fixed point is equivalent to truncated recurrent backpropagation.

2 Machine Learning Setting

We consider the supervised setting in which we want to predict a target y given an input x. The pair (x, y) is a data point. The model is a network specified by a state variable s and a parameter variable θ. The dynamics of the network are determined by two differentiable scalar functions, $E_\theta(x, s)$ and $C_\theta(y, s)$, which we call *energy function* and *cost function*, respectively. In most of the letter, to simplify the notations, we omit the dependence on x and y and simply write $E_\theta(s)$ and $C_\theta(s)$. Furthermore, we write $\frac{\partial E_\theta}{\partial \theta}(s)$ and $\frac{\partial E_\theta}{\partial s}(s)$ the partial derivatives of $(\theta, s) \mapsto E_\theta(s)$ with respect to θ and s, respectively. Similarly $\frac{\partial C_\theta}{\partial \theta}(s)$ and $\frac{\partial C_\theta}{\partial s}(s)$ denote the partial derivatives of $(\theta, s) \mapsto C_\theta(s)$.

The state variable s is assumed to move spontaneously toward low-energy configurations by following the gradient of the energy function:

$$\frac{ds}{dt} = -\frac{\partial E_\theta}{\partial s}(s). \tag{2.1}$$

The state s eventually settles to a minimum of the energy function, written s_θ^0 and characterized by[1]

$$\frac{\partial E_\theta}{\partial s}\left(s_\theta^0\right) = 0. \tag{2.2}$$

Since the dynamics in equation 2.1 depends on only the input x (through $E_\theta(x, s)$) and not on the target y, we call this relaxation phase the *free phase*, and the energy minimum s_θ^0 is called the *free fixed point*.

The goal of learning is that of finding θ such that the cost at the fixed point $C_\theta\left(s_\theta^0\right)$ is minimal.[2] We introduce the objective function (for a single data point (x, y)):

$$J(\theta) := C_\theta\left(s_\theta^0\right). \tag{2.3}$$

Note the distinction between the cost function and the objective function: the cost function $C_\theta(s)$ is defined for any state s, whereas the objective function $J(\theta)$ is the cost at the fixed point.

Several methods have been proposed to compute the gradient of J with respect to θ. Early work by Almeida (1987) and Pineda (1987) introduced the recurrent backpropagation algorithm, which we present in section 3. In Scellier and Bengio (2017) we proposed another algorithm—at first sight very different. We present it in section 4. In section 5 we show that there is actually a profound connection between these two algorithms.

2.1 Example: Hopfield Model. In this section we propose particular forms for the energy function $E_\theta(x, s)$ and the cost function $C_\theta(y, s)$ to ease understanding. Nevertheless, the theory presented in this letter is general and does not rely on the particular forms of the functions $E_\theta(x, s)$ and $C_\theta(y, s)$ chosen here.

[1] In general, the fixed point defined by equation 2.2 is not unique unless further assumptions are made on $E_\theta(s)$ (e.g., convexity). The fixed point depends on the initial state of the dynamics (see equation 2.1) and so does the objective function of equation 2.3. However, for ease of presentation, we avoid delving into these mathematical details here.

[2] In this expression, both the cost function $C_\theta(s)$ and the fixed-point s_θ^0 depend on θ. $C_\theta(s)$ directly depends on θ, whereas s_θ^0 indirectly depends on θ through $E_\theta(s)$ (see equation 2.2).

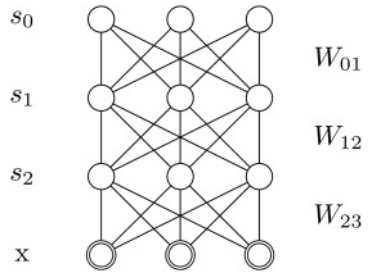

Figure 1: Graph of the network. Input x is clamped. State variable s includes hidden layers s_2 and s_1, and output layer s_0 (layer where the prediction is read). Output layer s_0 has the same dimension as target y.

Recall that we consider the supervised setting where we must predict a target y given an input x. To illustrate the idea, we consider the case where the neurons of the network are split in layers s_0, s_1, and s_2, as in Figure 1.[3] In this setting, the state variable s is the set of layers $s = (s_0, s_1, s_2)$. Each of the layers of neurons s_0, s_1, and s_2 is a vector whose coordinates are real numbers representing the membrane voltages of the neurons. The output layer s_0 corresponds to the layer where the prediction is read and has the same dimension as the target y. Furthermore ρ is a deterministic function (nonlinear activation) that maps a neuron's voltage onto its firing rate. We commit a small abuse of notation and denote $\rho(s_i)$ the vector of firing rates of the neurons in layer s_i; here the function ρ is applied elementwise to the coordinates of the vector s_i. Therefore, the vector $\rho(s_i)$ has the same dimension as s_i. Finally, the parameter variable θ is the set of (bidirectional) weight matrices between the layers $\theta = (W_{01}, W_{12}, W_{23})$.

We consider the following modified Hopfield energy function:

$$\begin{aligned} E_\theta(x, s) = \frac{1}{2}\left(\|s_0\|^2 + \|s_1\|^2 + \|s_2\|^2\right) \\ -\rho(s_0)^T \cdot W_{01} \cdot \rho(s_1) \\ -\rho(s_1)^T \cdot W_{12} \cdot \rho(s_2) \\ -\rho(s_2)^T \cdot W_{23} \cdot \rho(x). \end{aligned} \tag{2.4}$$

With this choice of energy function, the dynamics (see equation 2.1) translate into a form of leaky integration neural dynamics with symmetric

[3] We choose to number the layers in increasing order from output to input, in the sense of propagation of error signals (see section 4).

connections:[4]

$$\frac{ds_0}{dt} = \rho'(s_0)^T \odot W_{01} \cdot \rho(s_1) - s_0, \tag{2.5}$$

$$\frac{ds_1}{dt} = \rho'(s_1)^T \odot \left(W_{12} \cdot \rho(s_2) + W_{01}^T \cdot \rho(s_0)\right) - s_1, \tag{2.6}$$

$$\frac{ds_2}{dt} = \rho'(s_2)^T \odot \left(W_{23} \cdot \rho(\mathrm{x}) + W_{12}^T \cdot \rho(s_1)\right) - s_2. \tag{2.7}$$

Here again the derivative of the function ρ (denoted ρ') is applied elementwise to the coordinates of the vectors s_0, s_1, and s_2, and the notation $\odot$ is used to mean element-wise multiplication.[5]

Finally we consider the quadratic cost function,[6]

$$C_\theta(\mathrm{y}, s) = \frac{1}{2}\|\mathrm{y} - s_0\|^2, \tag{2.8}$$

which measures the discrepancy between the output layer s_0 and the target y.

Note that the results established in this letter hold for any energy function $E_\theta(s)$ and any cost function $C_\theta(s)$, and are not limited to the Hopfield energy and the quadratic cost (see equations 2.4 and 2.8).

3 Recurrent Backpropagation

In this section, we present recurrent backpropagation, an algorithm introduced by Almeida (1987) and Pineda (1987) that computes the gradient of J (see equation 2.3). The original algorithm was described in the discrete-time setting and for a general state-to-state dynamics. Here we present it in the continuous-time setting in the particular case of a gradient dynamics (see equation 2.1). A direct derivation based on the adjoint method can also be found in LeCun, Touresky, Hinton, and Sejnowski (1988).

3.1 Projected Cost Function. Let $S_\theta^0(s, t)$ denote the state of the network at time $t \geq 0$ when it starts from an initial state s at time $t = 0$ and follows the free dynamics (see equation 2.1). In the theory of dynamical systems, $S_\theta^0(s, t)$ is called the *flow*. We introduce the projected cost function:

$$L_\theta(\mathrm{s}, t) := C_\theta\left(S_\theta^0(s, t)\right). \tag{3.1}$$

[4] The case without the constraint of symmetric connections is studied in Scellier, Goyal, Binas, Mesnard, and Bengio (2018).

[5] Given two vectors $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$, their product element by element is $a \odot b = (a_1 b_1, \ldots, a_n b_n)$.

[6] In this specific example the cost function $C_\theta(\mathrm{y}, s)$ does not depend on θ, s_1 and s_2.

This is the cost of the state projected a duration t in the future, when the network starts from s and follows the free dynamics. For fixed s, the process $(L_\theta(s,t))_{t\geq 0}$ represents the successive cost values taken by the state of the network, along the free dynamics when it starts from the initial state s. Notable cases include these:

- For $t = 0$, the projected cost is simply the cost of the current state $L_\theta(\mathrm{s}, 0) = C_\theta(s)$.
- As $t \to \infty$, the projected cost converges to the objective $L_\theta(s,t) \to J(\theta)$.

The second property comes from the fact that the dynamics converges to the fixed point, $S^0_\theta(s,t) \to s^0_\theta$ as $t \to \infty$. Under mild regularity conditions on $E_\theta(s)$ and $C_\theta(s)$, the gradient of the projected cost function converges to the gradient of the objective function in the limit of infinite duration,

$$\frac{\partial L_\theta}{\partial \theta}(\mathrm{s}, t) \to \frac{\partial J}{\partial \theta}(\theta), \tag{3.2}$$

as $t \to \infty$. Therefore, if we can compute $\frac{\partial L_\theta}{\partial \theta}(s,t)$ for a particular value of s and for any $t \geq 0$, we can obtain the desired gradient $\frac{\partial J}{\partial \theta}(\theta)$ by letting $t \to \infty$. We will show next that this is what recurrent backpropagation does in the case where the initial state s is the fixed point s^0_θ.

3.2 Process of Error Derivatives. We introduce the process of error derivatives $(\overline{S}_t, \overline{\Theta}_t)_{t\geq 0}$, defined as

$$\overline{S}_t := \frac{\partial L_\theta}{\partial s}\left(s^0_\theta, t\right), \qquad t \geq 0, \tag{3.3}$$

$$\overline{\Theta}_t := \frac{\partial L_\theta}{\partial \theta}\left(s^0_\theta, t\right), \qquad t \geq 0. \tag{3.4}$$

The process $\overline{S}_t$ takes values in the state space (space of the state variable s), and the process $\overline{\Theta}_t$ takes values in the parameter space (space of the parameter variable θ).[7] The recurrent backpropagation algorithm computes $\overline{S}_t$ and $\overline{\Theta}_t$ iteratively for increasing values of t.

Theorem 1 *(recurrent backpropagation). The process of error derivatives* $(\overline{S}_t, \overline{\Theta}_t)$ *satisfies*

$$\overline{S}_0 = \frac{\partial C_\theta}{\partial s}\left(s^0_\theta\right), \tag{3.5}$$

[7] The quantity $\overline{\Theta}_t = \frac{\partial L_\theta}{\partial \theta}\left(s^0_\theta, t\right)$ represents the partial derivative of $L_\theta(\mathrm{s}, t)$ with respect to θ, evaluated at the fixed point $\mathrm{s} = s^0_\theta$. This does not include the differentiation path through the fixed point s^0_θ.

$$\overline{\Theta}_0 = \frac{\partial C_\theta}{\partial \theta}\left(s_\theta^0\right), \tag{3.6}$$

$$\frac{d}{dt}\overline{S}_t = -\frac{\partial^2 E_\theta}{\partial s^2}\left(s_\theta^0\right) \cdot \overline{S}_t, \tag{3.7}$$

$$\frac{d}{dt}\overline{\Theta}_t = -\frac{\partial^2 E_\theta}{\partial \theta \partial s}\left(s_\theta^0\right) \cdot \overline{S}_t. \tag{3.8}$$

Theorem 1, proved in appendix A, offers us a two-phase method to compute the gradient $\frac{\partial J}{\partial \theta}(\theta)$. In the first phase (or free phase), the state variable s follows the free dynamics (see equation 2.1) and relaxes to the fixed-point s_θ^0. Reaching this fixed point is necessary for evaluating the Hessian $\frac{\partial^2 E_\theta}{\partial s^2}\left(s_\theta^0\right)$, which is required in the second phase. In the second phase, one computes $\overline{S}_t$ and $\overline{\Theta}_t$ iteratively for increasing values of t using equations 3.5 to 3.8. We obtain the desired gradient in the limit $t \to \infty$, as a consequence of equation 3.2:

$$\overline{\Theta}_t \to \frac{\partial J}{\partial \theta}(\theta). \tag{3.9}$$

Note that the Hessian $\frac{\partial^2 E_\theta}{\partial s^2}\left(s_\theta^0\right)$ is positive definite since s_θ^0 is an energy minimum. Therefore, equation 3.7 guarantees that $\frac{\partial L_\theta}{\partial s}\left(s_\theta^0, t\right) \to 0$ as $t \to \infty$, in agreement with the fact that $J(\theta)$ is (locally) insensitive to the initial state ($s = s_\theta^0$ in our case).

From the point of view of biological plausibility, the requirement to run the dynamics for $\overline{S}_t$ and $\overline{\Theta}_t$ to compute the gradient $\frac{\partial J}{\partial \theta}(\theta)$ is not satisfying. It is not clear what the quantities $\overline{S}_t$ and $\overline{\Theta}_t$ would represent in a biological network. We address this issue in sections 4 and 5.

4 Equilibrium Propagation

In this section, we present equilibrium propagation (Scellier & Bengio, 2017), another algorithm that computes the gradient of the objective function J, equation 2.3. At first sight, equilibrium propagation and recurrent backpropagation have little in common. However, in section 5, we will show a profound connection between these algorithms.

4.1 Augmented Energy Function. The central idea of equilibrium propagation is to introduce the augmented energy function,

$$E_\theta^\beta(s) := E_\theta(s) + \beta\, C_\theta(s), \tag{4.1}$$

where $\beta \geq 0$ is a scalar that we call *influence parameter*. The free dynamics (see equation 2.1) is then replaced by the augmented dynamics:

$$\frac{ds}{dt} = -\frac{\partial E_\theta^\beta}{\partial s}(s). \tag{4.2}$$

The state variable now follows the dynamics $\frac{ds}{dt} = -\frac{\partial E_\theta}{\partial s}(s) - \beta \frac{\partial C_\theta}{\partial s}(s)$. When $\beta > 0$, in addition to the usual term $-\frac{\partial E_\theta}{\partial s}(s)$, a term $-\beta \frac{\partial C_\theta}{\partial s}(s)$ nudges s toward configurations that have lower cost values. In the case of the model described in section 2.1 with the quadratic cost function (see equation 2.8), the new term $-\beta \frac{\partial C_\theta}{\partial s}(s)$ is the vector of $\dim(s)$ whose component on s_0 is $\beta\left(\mathrm{y} - s_0\right)$ and whose components on s_1 and s_2 are zero. Thus, the new term takes the form of a force that nudges the output layer s_0 toward the target y. Unlike the free dynamics, which depends only on x (through $E_\theta(\mathrm{x}, s)$), not on y, the augmented dynamics also depends on y (through $C_\theta(y, s)$).

Note that the free dynamics corresponds to the value $\beta = 0$. We then generalize the notion of fixed point for any value of β. The augmented dynamics converges to the fixed-point s_θ^β, an energy minimum of E_θ^β, characterized by

$$\frac{\partial E_\theta^\beta}{\partial s}\left(s_\theta^\beta\right) = 0. \tag{4.3}$$

Theorem 2 shows that the gradient $\frac{\partial J}{\partial \theta}(\theta)$ can be estimated based on measurements at the fixed points s_θ^0 and s_θ^β.

Theorem 2 *(equilibrium propagation). The gradient of the objective function with respect to θ can be estimated using the formula*

$$\frac{\partial J}{\partial \theta}(\theta) = \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(s_\theta^0\right)\right). \tag{4.4}$$

A proof of theorem 2 is given in appendix B. Note that theorem 2 is also a consequence of the more general formula of equation 5.5 (theorem 3), established in the next section.

Theorem 2 offers another way to estimate the gradient of $J(\theta)$. As in recurrent backpropagation, in the first phase (or free phase), the network follows the free dynamics (see equation 2.1). This is equivalent to saying that the network follows the augmented dynamics (see equation 4.2) when the value of β is set to 0. The network relaxes to the free fixed point s_θ^0, where $\frac{\partial E_\theta}{\partial \theta}\left(s_\theta^0\right)$ is measured. In the second phase, which we call *nudged phase*, the influence parameter takes on a small, positive value $\beta \gtrsim 0$, and the network relaxes to a new but nearby fixed point s_θ^β where $\frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right)$ is measured. The gradient of the objective function is estimated using the formula in equation 4.4.

In the case of the modified Hopfield energy (see equation 2.4), the components of $\frac{\partial E_\theta}{\partial \theta}(s)$ are $\frac{\partial E_\theta}{\partial W_{01}}(s)$, $\frac{\partial E_\theta}{\partial W_{12}}(s)$, and $\frac{\partial E_\theta}{\partial W_{23}}(s)$. For instance, $\frac{\partial E_\theta}{\partial W_{01}}(s) = -\rho(s_0) \cdot \rho(s_1)^T$ is a matrix of size $\dim(s_0) \times \dim(s_1)$ whose entries can be measured locally at each synapse based on the presynaptic activity and postsynaptic activity. Thus, the learning rule of equation 4.4 is a kind of contrastive Hebbian learning rule at the free and nudged fixed points.

At the beginning of the second phase, the network is initially at the free fixed point s_θ^0. When the influence parameter takes on a small, positive value $\beta \gtrsim 0$, the novel term $-\beta \frac{\partial C_\theta}{\partial s}(s)$ in the dynamics of the state variable perturbs the system. This perturbation propagates into the layers of the network until convergence to the new fixed point s_θ^β.

In the next section, we go beyond the analysis of fixed points and show that at every moment t in the nudged phase, the temporal derivative $\frac{ds}{dt}$ encodes the error derivative of equation 3.3.

5 Temporal Derivatives Code for Error Derivatives

Theorem 2 shows that the gradient of J can be estimated based on the free and nudged fixed points only. In this section, we study the dynamics of the network in the second phase, from the free fixed point to the nudged fixed point. Recall that $S_\theta^0(\mathrm{s}, t)$ is the flow of the dynamical system (see equation 2.1), that is, the state of the network at time $t \geq 0$ when it starts from an initial state s at time $t = 0$ and follows the free dynamics. Similarly, we define $S_\theta^\beta(\mathrm{s}, t)$ for any value of β when the network follows the augmented dynamics (see equation 4.2).

In equilibrium propagation, the state of the network at the beginning of the nudged phase is the free fixed-point s_θ^0. We choose as origin of time $t = 0$ the moment when the second phase starts: the network is in the state s_θ^0, and the influence parameter takes on a small, positive value $\beta \gtrsim 0$. With our notations, the state of the network after a duration t in the nudged phase is $S_\theta^\beta\left(s_\theta^0, t\right)$. As $t \to \infty$, the network's state converges to the nudged fixed point $S_\theta^\beta\left(s_\theta^0, t\right) \to s_\theta^\beta$.

5.1 Process of Temporal Derivatives. Now we are ready to introduce the process of temporal derivatives $(\widetilde{S}_t, \widetilde{\Theta}_t)_{t \geq 0}$, defined by

$$\widetilde{S}_t := -\lim_{\beta \to 0} \frac{1}{\beta} \frac{\partial S_\theta^\beta}{\partial t}\left(s_\theta^0, t\right), \tag{5.1}$$

$$\widetilde{\Theta}_t := \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(s_\theta^0\right)\right). \tag{5.2}$$

Like $\overline{S}_t$ and $\overline{\Theta}_t$, the processes $\widetilde{S}_t$ and $\widetilde{\Theta}_t$ take values in the state space and parameter space, respectively.

The process $\widetilde{S}_t$ is simply the temporal derivative $\frac{ds}{dt}$ in the second phase, rescaled by $\frac{1}{\beta}$ (so that its value does not depend on the particular choice of $\beta \gtrsim 0$).

Theorem 3 *(temporal derivatives as error derivatives). The process of error derivatives* $(\overline{S}_t, \overline{\Theta}_t)$ *and the process of temporal derivatives* $(\widetilde{S}_t, \widetilde{\Theta}_t)$ *are equal:*

$$\forall t \geq 0, \qquad \overline{S}_t = \widetilde{S}_t, \qquad \overline{\Theta}_t = \widetilde{\Theta}_t, \tag{5.3}$$

or, using explicit forms,

$$\frac{\partial L_\theta}{\partial s}\left(s_\theta^0, t\right) = -\lim_{\beta \to 0} \frac{1}{\beta} \frac{\partial S_\theta^\beta}{\partial t}\left(s_\theta^0, t\right), \tag{5.4}$$

$$\frac{\partial L_\theta}{\partial \theta}\left(s_\theta^0, t\right) = \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(s_\theta^0\right)\right). \tag{5.5}$$

Theorem 3 is proved in appendix C. In essence, equation 5.4 says that in the second phase of equilibrium propagation, the temporal derivative $\frac{ds}{dt}$ (rescaled by $\frac{1}{\beta}$) encodes the error derivative (equation 3.3).

Here is an interpretation of equation 5.4. Suppose that the network is initially at the fixed point $\mathrm{s} = s_\theta^0$. Consider the cost $L_\theta\left(s_\theta^0 + \Delta s, t\right)$, a duration t in the future if one moved the initial state $\mathrm{s} = s_\theta^0$ by a small step Δs. The goal is to find the direction Δs, which minimizes $L_\theta\left(s_\theta^0 + \Delta s, t\right)$. The naive approach by trial and error is neither biologically plausible nor efficient. Equation 5.4 tells us that there is a physically realistic way of finding such a direction Δs in one attempt. This direction is encoded in the temporal derivative $\frac{ds}{dt}$ at time t after starting the nudged phase.

Note that as $t \to \infty$, both sides of equation 5.4 converge to 0. This is a consequence of equation 3.7 and the fact that the Hessian $\frac{\partial^2 E_\theta}{\partial s^2}\left(s_\theta^0\right)$ is positive definite (since s_θ^0 is an energy minimum), as already mentioned in section 3. Intuitively, the right-hand side converges to 0 because $S_\theta^\beta\left(s_\theta^0, t\right)$ converges smoothly to the nudged fixed point s_θ^β. As for the left-hand side, when t is large, $L_\theta\left(s, t\right)$ is close to the cost of the energy minimum and thus has little sensitivity to the initial state s.

Finally, as $t \to \infty$ in equation 5.5, one recovers the gradient formula of equilibrium propagation (theorem 2). Interestingly, equation 5.5 shows that in equilibrium propagation, halting the second phase before convergence to the nudged fixed point corresponds to truncated recurrent backpropagation.

6 Conclusion

Our work establishes a close connection between two algorithms for fixed-point recurrent networks: recurrent backpropagation and equilibrium propagation. The temporal derivatives of the neural activities in the second phase of equilibrium propagation are equal to the error derivatives that recurrent backpropagation computes iteratively. Moreover, we have shown that halting the second phase before convergence in equilibrium propagation is equivalent to truncated recurrent backpropagation. Our work supports the hypothesis that in biological networks, temporal changes in neural activities may represent error signals for supervised learning from a machine learning perspective.

One important drawback of the theory presented here is that it assumes the existence of an energy function. In the case of the Hopfield energy, this implies symmetric connections between neurons. However, the analysis presented here can be generalized to dynamics that do not involve energy functions. This is the subject of Scellier et al. (2018). Another concern is the fact that our algorithm is rate based, whereas biological neurons emit spikes. Ideally we would like a theory applicable to spiking networks. Finally, the assumption of the existence of specialized output neurons (s_0 here) would need to be relaxed too.

From a practical point of view, another issue is that the time needed to converge to the first fixed point was experimentally found to grow exponentially with the number of layers in Scellier and Bengio (2017). Although equation 5.5 provides a new justification for saving time by stopping the second phase early, our algorithm (as well as recurrent backpropagation) still requires convergence to the free fixed point in the first phase.

Appendix A: Recurrent Backpropagation: Proof

Proof of Theorem 1. First, by definition of L, equation 3.1, we have $L_\theta(\mathrm{s}, 0) = C_\theta(\mathrm{s})$. Therefore, the initial conditions, equations 3.5 and 3.6, are satisfied:

$$\overline{S}_0 = \frac{\partial L_\theta}{\partial s}\left(s_\theta^0, 0\right) = \frac{\partial C_\theta}{\partial s}\left(s_\theta^0\right) \tag{A.1}$$

and

$$\overline{\Theta}_0 = \frac{\partial L_\theta}{\partial \theta}\left(s_\theta^0, 0\right) = \frac{\partial C_\theta}{\partial \theta}\left(s_\theta^0\right). \tag{A.2}$$

It remains to show equations 3.7 and 3.8. Temporarily, we omit writing the dependence in θ to keep notations simple. As a preliminary result, we show that for any initial state s and time t, we have[8]

$$\frac{\partial L}{\partial t}(s,t) + \frac{\partial L}{\partial s}(s,t) \cdot \frac{\partial E}{\partial s}(s) = 0. \tag{A.3}$$

To this end, note that (by definition of L and S^0) we have for any t and u,

$$L\left(S^0(s,u),t\right) = L(s,t+u). \tag{A.4}$$

The derivatives of the right-hand side of equation A.4 with respect to t and u are clearly equal:

$$\frac{d}{dt}L(s,t+u) = \frac{d}{du}L(s,t+u). \tag{A.5}$$

Therefore the derivatives of the left-hand side of equation A.4 are equal too:

$$\frac{\partial L}{\partial t}\left(S^0(s,u),t\right) = \frac{d}{du}L\left(S^0(\mathrm{s},u),t\right) \tag{A.6}$$

$$= -\frac{\partial L}{\partial s}\left(S^0(s,u),t\right) \cdot \frac{\partial E}{\partial s}\left(S^0(s,u)\right). \tag{A.7}$$

Here we have used the differential equation of motion, equation 2.1. Evaluating this expression for $u = 0$, we get equation A.3.

Now we are ready to show that $\overline{S}_t = \frac{\partial L}{\partial s}\left(s^0,t\right)$ satisfies the differential equation in equation 3.7. Differentiating equation A.3 with respect to s, we get

$$\frac{\partial^2 L}{\partial t \partial s}(s,t) + \frac{\partial^2 L}{\partial s^2}(s,t) \cdot \frac{\partial E}{\partial s}(s) + \frac{\partial L}{\partial s}(s,t) \cdot \frac{\partial^2 E}{\partial s^2}(s) = 0. \tag{A.8}$$

Evaluating this expression at the fixed point $\mathrm{s} = s^0$ and using the fixed-point condition $\frac{\partial E}{\partial s}\left(s^0\right) = 0$, we get

$$\frac{d}{dt}\frac{\partial L}{\partial s}\left(s^0,t\right) = -\frac{\partial^2 E}{\partial s^2}\left(s^0\right) \cdot \frac{\partial L}{\partial s}\left(s^0,t\right). \tag{A.9}$$

Therefore $\overline{S}_t = \frac{\partial L}{\partial s}\left(s^0,t\right)$ satisfies equation 3.7.

[8] Equation A.3 is the Kolmogorov backward equation for deterministic processes.

We prove equation 3.8 similarly. Differentiating equation A.3 with respect to θ, we get

$$\frac{\partial^2 L_\theta}{\partial t \partial \theta}(s,t) + \frac{\partial^2 L_\theta}{\partial s \partial \theta}(s,t) \cdot \frac{\partial E_\theta}{\partial s}(s) + \frac{\partial L_\theta}{\partial s}(s,t) \cdot \frac{\partial^2 E_\theta}{\partial s \partial \theta}(s) = 0. \tag{A.10}$$

Evaluating this expression at the fixed point $s = s_\theta^0$, we get

$$\frac{d}{dt}\frac{\partial L_\theta}{\partial \theta}\left(s_\theta^0, t\right) = -\frac{\partial^2 E_\theta}{\partial \theta \partial s}\left(s_\theta^0\right) \cdot \frac{\partial L_\theta}{\partial s}\left(s_\theta^0, t\right). \tag{A.11}$$

Hence the result. □

Appendix B: Equilibrium Propagation: Proof

In this appendix, we prove theorem 2. The same proof is provided in Scellier and Bengio (2017).

Since the data point (x, y) does not play any role, its dependence is omitted in the notations. We assume that the energy function $E_\theta(s)$ and the cost function $C_\theta(s)$ (and thus the augmented energy function $E_\theta^\beta(s)$) are twice differentiable and that the conditions of the implicit function theorem are satisfied so that the fixed point s_θ^β is a continuously differentiable function of (θ, β).

Proof of Theorem 2. Recall that we want to show the gradient formula:

$$\frac{\partial J}{\partial \theta}(\theta) = \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(s_\theta^0\right)\right). \tag{B.1}$$

The gradient formula, equation B.1, is a particular case of the following formula,[9] when evaluated at the point $\beta = 0$:

$$\frac{d}{d\theta}\frac{\partial E_\theta^\beta}{\partial \beta}\left(s_\theta^\beta\right) = \frac{d}{d\beta}\frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right). \tag{B.2}$$

[9] The notations $\frac{\partial E_\theta^\beta}{\partial \theta}$ and $\frac{\partial E_\theta^\beta}{\partial \beta}$ are used to mean the partial derivatives with respect to the arguments of E_θ^β, whereas $\frac{d}{d\theta}$ and $\frac{d}{d\beta}$ represent the total derivatives with respect to θ and β, respectively (which include the differentiation path through s_θ^β). The total derivative $\frac{d}{d\theta}$ (resp. $\frac{d}{d\beta}$) is performed for fixed β (resp. fixed θ).

Therefore, in order to prove equation B.1, it is sufficient to prove equation B.2.

First, the cross-derivatives of $(\theta, \beta) \mapsto E_\theta^\beta\left(s_\theta^\beta\right)$ are equal:

$$\frac{d}{d\theta}\frac{d}{d\beta}E_\theta^\beta\left(s_\theta^\beta\right) = \frac{d}{d\beta}\frac{d}{d\theta}E_\theta^\beta\left(s_\theta^\beta\right). \tag{B.3}$$

Second, by the chain rule of differentiation, we have

$$\frac{d}{d\beta}E_\theta^\beta\left(s_\theta^\beta\right) = \frac{\partial E_\theta^\beta}{\partial \beta}\left(s_\theta^\beta\right) + \frac{\partial E_\theta^\beta}{\partial s}\left(s_\theta^\beta\right)\cdot\frac{\partial s_\theta^\beta}{\partial \beta} \tag{B.4}$$

$$= \frac{\partial E_\theta^\beta}{\partial \beta}\left(s_\theta^\beta\right). \tag{B.5}$$

Here we have used the fixed-point condition,

$$\frac{\partial E_\theta^\beta}{\partial s}\left(s_\theta^\beta\right) = 0. \tag{B.6}$$

Similarly we have

$$\frac{d}{d\theta}E_\theta^\beta\left(s_\theta^\beta\right) = \frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right). \tag{B.7}$$

Plugging equations B.5 and B.7 in B.3, we get equation B.2. Hence the result. □

Appendix C: Temporal Derivatives Code for Error Derivatives: Proof

Proof of Theorem 3. In order to prove theorem 3, we have to show that the process $(\widetilde{S}_t, \widetilde{\Theta}_t)$ satisfies the same differential equations as $(\overline{S}_t, \overline{\Theta}_t)$, namely, equations 3.5 to 3.8 (see theorem 1). We conclude by using the uniqueness of the solution to the differential equation with initial condition.

First, note that

$$\left.\frac{\partial^2 S_\theta^\beta}{\partial \beta \partial t}\right|_{\beta=0}\left(s_\theta^0, t\right)$$

$$= \lim_{\beta\to 0}\frac{1}{\beta}\left(\frac{\partial S_\theta^\beta}{\partial t}\left(s_\theta^0, t\right) - \frac{\partial S_\theta^0}{\partial t}\left(s_\theta^0, t\right)\right) \tag{C.1}$$

$$= \lim_{\beta\to 0}\frac{1}{\beta}\frac{\partial S_\theta^\beta}{\partial t}\left(s_\theta^0, t\right). \tag{C.2}$$

The latter equality comes from the fact that $S_\theta^0\left(s_\theta^0, t\right) = s_\theta^0$ for every $t \geq 0$, implying that $\frac{\partial S_\theta^0}{\partial t}\left(s_\theta^0, t\right) = 0$ at every moment $t \geq 0$. Furthermore,

$$\begin{aligned}
&\left.\frac{d}{d\beta}\right|_{\beta=0} \frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) \\
&= \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(S_\theta^0\left(s_\theta^0, t\right)\right)\right) \qquad \text{(C.3)} \\
&= \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) - \frac{\partial E_\theta}{\partial \theta}\left(s_\theta^0\right)\right). \qquad \text{(C.4)}
\end{aligned}$$

Again, the latter equality comes from the fact that $S_\theta^0\left(s_\theta^0, t\right) = s_\theta^0$ for every $t \geq 0$. Therefore,

$$\widetilde{S}_t = -\left.\frac{\partial^2 S_\theta^\beta}{\partial \beta \partial t}\right|_{\beta=0}\left(s_\theta^0, t\right), \qquad \forall t \geq 0, \qquad \text{(C.5)}$$

$$\widetilde{\Theta}_t = \left.\frac{d}{d\beta}\right|_{\beta=0} \frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right), \qquad \forall t \geq 0. \qquad \text{(C.6)}$$

Now we prove that $\widetilde{S}_t$ is the solution of equations 3.5 and 3.7. We omit writing the dependence in θ to keep notations simple. The process $\left(S^\beta\left(s^0, t\right)\right)_{t \geq 0}$ is the solution of the differential equation,

$$\frac{\partial S^\beta}{\partial t}\left(s^0, t\right) = -\frac{\partial E^\beta}{\partial s}\left(S^\beta\left(s^0, t\right)\right), \qquad \text{(C.7)}$$

with initial condition $S^\beta\left(s^0, 0\right) = s^0$. Differentiating equation C.7 with respect to β, we get

$$\begin{aligned}
\frac{d}{dt}\frac{\partial S^\beta}{\partial \beta}\left(s^0, t\right) = &-\frac{\partial^2 E^\beta}{\partial s \partial \beta}\left(S^\beta\left(s^0, t\right)\right) \\
&-\frac{\partial^2 E^\beta}{\partial s^2}\left(S^\beta\left(s^0, t\right)\right) \cdot \frac{\partial S^\beta}{\partial \beta}\left(s^0, t\right). \qquad \text{(C.8)}
\end{aligned}$$

Evaluating at $\beta = 0$ and using the fact that $S^0\left(s^0, t\right) = s^0$, we get

$$\begin{aligned}
\frac{d}{dt}\left.\frac{\partial S^\beta}{\partial \beta}\right|_{\beta=0}\left(s^0, t\right) = &-\frac{\partial C}{\partial s}\left(s^0\right) \\
&-\frac{\partial^2 E}{\partial s^2}\left(s^0\right) \cdot \left.\frac{\partial S^\beta}{\partial \beta}\right|_{\beta=0}\left(s^0, t\right). \qquad \text{(C.9)}
\end{aligned}$$

Since at time $t = 0$, the initial state of the network $S^\beta\left(s^0, 0\right) = s^0$ is independent of β, we have

$$\frac{\partial S^\beta}{\partial \beta}\left(s^0, 0\right) = 0. \tag{C.10}$$

Therefore, evaluating equation 4.9 at $t = 0$, we get the initial condition, equation 3.5:

$$\widetilde{S}_0 = -\left.\frac{\partial^2 S^\beta}{\partial t \partial \beta}\right|_{\beta=0}\left(s^0, 0\right) = \frac{\partial C}{\partial s}\left(s^0\right). \tag{C.11}$$

Moreover, differentiating equation C.9 with respect to time, we get

$$\frac{d}{dt}\left.\frac{\partial^2 S^\beta}{\partial t \partial \beta}\right|_{\beta=0}\left(s^0, t\right) = -\frac{\partial^2 E}{\partial s^2}\left(s^0\right) \cdot \left.\frac{\partial^2 S^\beta}{\partial t \partial \beta}\right|_{\beta=0}\left(s^0, t\right). \tag{C.12}$$

Hence, equation 3.7:

$$\frac{d}{dt}\widetilde{S}_t = -\frac{\partial^2 E}{\partial s^2}\left(s^0\right) \cdot \widetilde{S}_t. \tag{C.13}$$

Now we prove the result for $\widetilde{\Theta}_t$ (see equations 3.6 and 3.8). First, we differentiate $\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right)$ with respect to β:

$$\begin{aligned}\frac{d}{d\beta}\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) = &\frac{\partial E_\theta^\beta}{\partial \theta \partial \beta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) \\ &+ \frac{\partial E_\theta^\beta}{\partial \theta \partial s}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) \cdot \frac{\partial S_\theta^\beta}{\partial \beta}\left(s_\theta^0, t\right).\end{aligned} \tag{C.14}$$

Again we evaluate at $\beta = 0$ and use the fact that $S_\theta^0\left(s_\theta^0, t\right) = s_\theta^0$. We get

$$\begin{aligned}\left.\frac{d}{d\beta}\right|_{\beta=0}\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) = &\frac{\partial C_\theta}{\partial \theta}\left(s_\theta^0\right) \\ &+ \frac{\partial E_\theta}{\partial \theta \partial s}\left(s_\theta^0\right) \cdot \left.\frac{\partial S_\theta^\beta}{\partial \beta}\right|_{\beta=0}\left(s_\theta^0, t\right).\end{aligned} \tag{C.15}$$

Evaluating equation C.15 at time $t = 0$ and using equation C.10, we get the initial condition, equation 3.6:

$$\widetilde{\Theta}_0 = \left.\frac{d}{d\beta}\right|_{\beta=0} \frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, 0\right)\right) = \frac{\partial C_\theta}{\partial \theta}\left(s_\theta^0\right). \tag{C.16}$$

Moreover, differentiating equation C.15 with respect to time, we get

$$\frac{d}{dt}\left.\frac{d}{d\beta}\right|_{\beta=0} \frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) = \frac{\partial E_\theta}{\partial \theta \partial s}\left(s_\theta^0\right) \cdot \left.\frac{\partial^2 S_\theta^\beta}{\partial t \partial \beta}\right|_{\beta=0} \left(s_\theta^0, t\right). \tag{C.17}$$

Hence, equation 3.8:

$$\frac{d}{dt}\widetilde{\Theta}_t = -\frac{\partial E_\theta}{\partial \theta \partial s}\left(s_\theta^0\right) \cdot \widetilde{S}_t. \tag{C.18}$$

This completes the proof. □

Acknowledgments

We thank Jonathan Binas for feedback and discussions, as well as NSERC, CIFAR, Samsung, and Canada Research Chairs for funding.

References

Almeida, L. B. (1987). A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In *Proceedings of the First International Conference on Neural Networks* (vol. 2, pp. 609–618). Piscataway, NJ: IEEE.

Cohen, M. A., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, *5*, 815–826.

Crick, F. (1989). The recent excitement about neural networks. *Nature*, *337*(6203), 129–132.

Hinton, G. E., & McClelland, J. L. (1988). Learning representations by recirculation. In D. Z. Anderson (Ed.), *Neural information processing systems* (pp. 358–366). College Park, MD: American Institute of Physics.

Hopfield, J. J. (1984). Neurons with graded responses have collective computational properties like those of two-state neurons. *PNAS*, *81*, 3088–3092.

LeCun, Y., Touresky, D., Hinton, G., & Sejnowski, T. (1988). A theoretical framework for back-propagation. In *Proceedings of the 1988 Connectionist Models Summer School* (pp. 21–28). San Mateo, CA: Morgan Kaufmann.

Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, *59*, 2229–2232.

Scellier, B., & Bengio, Y. (2017). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, *11*.

Scellier, B., Goyal, A., Binas, J., Mesnard, T., & Bengio, Y. (2018). *Generalization of equilibrium propagation to vector field dynamics*. arXiv:1808.04873.

Received March 14, 2018; accepted October 15, 2018.