Equivalence of Equilibrium Propagation and Recurrent Backpropagation

Benjamin Scellier and Yoshua Bengio[*]
MILA, Université de Montréal

June 15, 2018

Abstract

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 which 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 back-propagation. The work presented in this paper also 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. At prediction time, the network relaxes to a fixed point corresponding to a local minimum of the energy function. This corresponds to the first phase of the algorithm. The goal of learning is that of minimizing the cost at the fixed point, called objective.

In section 3 we present *Recurrent Backpropagation* [Almeida, 1987, Pineda, 1987], a two-phase learning algorithm which computes the gradient of the objective function. In the second phase of Recurrent Backpropagation, an iterative procedure computes error derivatives.

In section 4 we present *Equilibrium Propagation* [Scellier and Bengio, 2017], another two-phase algorithm which computes the gradient of the objective. In the second phase of Equilibrium Propagation the dynamics of the network is slightly perturbed and the network starts a second relaxation phase towards a second but nearby fixed point which 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.

In section 5 (the main contribution of the present 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*.

[*] Y.B. is also a Senior Fellow of CIFAR

2 Machine Learning Setting

We consider the supervised setting in which we want to predict a *target* y given an *input* x. 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 $E_\theta(\mathrm{x}, s)$ and $C_\theta(\mathrm{y}, s)$ which we call *energy function* and *cost function* respectively. In most of the paper, 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 towards low-energy configurations by following the gradient of the energy function:

$$\frac{ds}{dt} = -\frac{\partial E_\theta}{\partial s}(s). \tag{1}$$

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

$$\frac{\partial E_\theta}{\partial s}\left(s_\theta^0\right) = 0. \tag{2}$$

Since the dynamics in Eq. 1 only depends on the input x (through $E_\theta(\mathrm{x}, s)$) but not on the target y, we call this relaxation phase the *free phase*, and the energy minimum s_θ^0 is called *free fixed point*.

The goal of learning is to adjust θ so as to minimize the cost value of the fixed point. We introduce the *objective function* (for a single data sample (x, y))

$$J(\theta) := C_\theta\left(s_\theta^0\right). \tag{3}$$

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], Pineda [1987] have introduced an algorithm called Recurrent Backpropagation, which we present in section 3. In Scellier and Bengio [2017] we proposed another algorithm - at first sight very different - which we present in section 4. In section 5 we will show that there is actually a profound connection between these two algorithms.

3 Recurrent Back-Propagation

In this section we present *Recurrent Backpropagation*, an algorithm introduced by Almeida [1987], Pineda [1987] which computes the gradient of J (Eq. 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 (Eq. 1). A direct derivation based on the adjoint method can also be found in LeCun et al. [1988].

3.1 Projected Cost Function

Let $S_\theta^0(\mathrm{s}, t)$ 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 (Eq. 1). In the theory of dynamical systems $S_\theta^0(\mathrm{s}, t)$ is called the *flow*. We introduce the *projected cost function*

$$L_\theta(\mathrm{s}, t) := C_\theta\left(S_\theta^0(\mathrm{s}, t)\right). \tag{4}$$

This is the cost of the state projected a duration t in the future, when the networks starts from s and follows the free dynamics. For fixed s the process $\left(L_\theta(\mathrm{s}, t)\right)_{t \geq 0}$ represents the successive cost values taken by the state of the network along the free dynamics when it starts from the initial state s. For $t = 0$, the projected cost is simply the cost of the current state: $L_\theta(\mathrm{s}, 0) = C_\theta(\mathrm{s})$. As $t \to \infty$ the dynamics converges to the fixed point $S_\theta^0(\mathrm{s}, t) \to s_\theta^0$, so the projected cost converges to the objective $L_\theta(\mathrm{s}, t) \to J(\theta)$. Under mild regularity conditions on the functions $E_\theta(s)$ and $C_\theta(s)$, the gradient of the projected cost function converges to the gradient of the objective function as $t \to \infty$, i.e.

$$\frac{\partial L_\theta}{\partial \theta}(\mathrm{s}, t) \to \frac{\partial J}{\partial \theta}(\theta). \tag{5}$$

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

3.2 Process of Error Derivatives

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

$$\overline{S}_t := \frac{\partial L_\theta}{\partial s}\left(s_\theta^0, t\right), \qquad t \geq 0, \tag{6}$$

$$\overline{\Theta}_t := \frac{\partial L_\theta}{\partial \theta}\left(s_\theta^0, t\right), \qquad t \geq 0. \tag{7}$$

[1] In general, the fixed point defined by Eq. 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 (Eq. 1), and so does the objective function of Eq. 3. However, for ease of presentation, we shall avoid delving into these mathematical details here.

The processes $\overline{S}_t$ and $\overline{\Theta}_t$ take values in the state space (space of the state variable s) and parameter space (space of the parameter variable θ) respectively. 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*

$$\overline{S}_0 = \frac{\partial C_\theta}{\partial s}\left(s_\theta^0\right), \tag{8}$$

$$\overline{\Theta}_0 = \frac{\partial C_\theta}{\partial \theta}\left(s_\theta^0\right), \tag{9}$$

$$\frac{d}{dt}\overline{S}_t = -\frac{\partial^2 E_\theta}{\partial s^2}\left(s_\theta^0\right) \cdot \overline{S}_t, \tag{10}$$

$$\frac{d}{dt}\overline{\Theta}_t = -\frac{\partial^2 E_\theta}{\partial \theta \partial s}\left(s_\theta^0\right) \cdot \overline{S}_t. \tag{11}$$

Theorem 1 offers us a two-phase method to compute the gradient $\frac{\partial J}{\partial \theta}(\theta)$. In the first phase, the state variable s follows the free dynamics (Eq. 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, $\overline{S}_t$ and $\overline{\Theta}_t$ are computed iteratively for increasing values of t thanks to Eq. 8, Eq. 9, Eq. 10 and Eq. 11. As a consequence of Eq. 5, the desired gradient $\frac{\partial J}{\partial \theta}(\theta)$ is obtained as $t \to \infty$ since

$$\overline{\Theta}_t \to \frac{\partial J}{\partial \theta}(\theta). \tag{12}$$

From the point of view of biological plausibility, the requirement to run a new 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. This issue is adressed in sections 4 and 5.

4 Equilibrium Propagation

In this section, we present Equilibrium Propagation [Scellier and Bengio, 2017], another algorithm which computes the gradient of the objective function J (Eq. 3). At first sight, Equilibrium Propagation and Recurrent Backpropagation share 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*

$$E_\theta^\beta(s) := E_\theta(s) + \beta\, C_\theta(s), \tag{13}$$

where $\beta \geq 0$ is a scalar which we call *influence parameter*. The free dynamics (Eq. 1) is then replaced by the *augmented dynamics*

$$\frac{ds}{dt} = -\frac{\partial E_\theta^\beta}{\partial s}(s) \tag{14}$$

$$= -\frac{\partial E_\theta}{\partial s}(s) - \beta \frac{\partial C_\theta}{\partial s}(s). \tag{15}$$

When $\beta > 0$, in addition to the usual term $-\frac{\partial E_\theta}{\partial s}(s)$, an additional term $-\beta \frac{\partial C_\theta}{\partial s}(s)$ nudges s towards configurations that have lower cost values. As $t \to \infty$ the augmented dynamics converges to a fixed point s_θ^β, i.e. an energy minimum of E_θ^β characterized by

$$\frac{\partial E_\theta^\beta}{\partial s}\left(s_\theta^\beta\right) = 0. \tag{16}$$

Unlike the free fixed point s_θ^0 which only depends on x (through $E_\theta(\mathrm{x}, s)$) but not on y, the fixed point s_θ^β also depends on y (through $C_\theta(\mathrm{y}, s)$).

4.2 Gradient Formula

The Equilibrium Propagation algorithm estimates the gradient $\frac{\partial J}{\partial \theta}(\theta)$ based on measures 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 thanks to the formula*

$$\frac{\partial J}{\partial \theta}(\theta) = \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(s_\theta^0\right)\right). \tag{17}$$

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 (Eq. 1). 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 thanks to the formula in Eq. 17.

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 in 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 we show that, at every moment t in the nudged phase, the temporal derivative $\frac{ds}{dt}$ encodes the error derivative of Eq. 6.

5 Temporal Derivatives Code for Error Derivatives

In this section we are interested in 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 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 (Eq. 1). Similarly we define $S_\theta^\beta(\mathrm{s}, t)$ for any value of β when the network follows the augmented dynamics (Eq. 14).

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

$$\widetilde{S}_t := - \lim_{\beta \to 0} \frac{1}{\beta} \frac{\partial S_\theta^\beta}{\partial t}\left(s_\theta^0, t\right), \qquad (18)$$

$$\widetilde{\Theta}_t := \lim_{\beta \to 0} \frac{1}{\beta} \left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(S_\theta^\beta\left(s_\theta^0, t\right)\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(s_\theta^0\right) \right). \qquad (19)$$

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 processes of temporal derivatives and error derivatives are equal: for every* $t \geq 0$ *we have*

$$\overline{S}_t = \widetilde{S}_t, \qquad (20)$$

$$\overline{\Theta}_t = \widetilde{\Theta}_t. \qquad (21)$$

In essence, Eq. 20 says that in the second phase of Equilibrium Propagation, the temporal derivative $\frac{ds}{dt}$ (rescaled by $\frac{1}{\beta}$) *encodes* the error derivatives of Eq. 6.

Also, note that the formula of Eq. 21 entails the gradient formula of Equilibrium Propagation (Theorem 2). As $t \to \infty$ in Eq. 21, one gets the formula of Eq. 17. Interestingly, Eq. 21 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, namely 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 which Recurrent Backpropagation computes iteratively. Thereby, 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.

Furthermore, we have shown that halting the second phase before convergence in Equilibrium Propagation is equivalent to Truncated Recurrent Backpropagation. This provides a new justification for saving time by stopping the second phase of Equilibrium Propagation early.

Acknowledgments

The authors would like to thank NSERC, CIFAR, Samsung and Canada Research Chairs for funding.

References

L. B. Almeida. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. volume 2, pages 609–618, San Diego 1987, 1987. IEEE, New York.

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

G. E. Hinton and J. L. McClelland. Learning representations by recirculation. In D. Z. Anderson, editor, *Neural Information Processing Systems*, pages 358–366. American Institute of Physics, 1988.

Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

F. J. Pineda. Generalization of back-propagation to recurrent neural networks. 59:2229–2232, 1987.

B. Scellier and Y. Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11, 2017.

Appendix

A Recurrent Backpropagation - Proof

Proof of Theorem 1. First of all, by definition of L (Eq. 4) we have $L_\theta(\mathrm{s}, 0) = C_\theta(\mathrm{s})$. Therefore the initial conditions (Eq. 8 and Eq. 9) are satisfied:

$$\overline{S}_0 = \frac{\partial L_\theta}{\partial s}\left(s_\theta^0, 0\right) = \frac{\partial C_\theta}{\partial s}\left(s_\theta^0\right) \tag{22}$$

and

$$\overline{\Theta}_0 = \frac{\partial L_\theta}{\partial \theta}\left(s_\theta^0, 0\right) = \frac{\partial C_\theta}{\partial \theta}\left(s_\theta^0\right). \tag{23}$$

It remains to show Eq. 10 and Eq. 11. We omit temporarily to write 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 [2]

$$\frac{\partial L}{\partial t}(\mathrm{s}, t) + \frac{\partial L}{\partial s}(\mathrm{s}, t) \cdot \frac{\partial E}{\partial s}(\mathrm{s}) = 0. \tag{24}$$

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

$$L\left(S^0(\mathrm{s}, u), t\right) = L(\mathrm{s}, t + u). \tag{25}$$

The derivatives of the right-hand side of Eq. 25 with respect to t and u are clearly equal:

$$\frac{d}{dt} L(\mathrm{s}, t + u) = \frac{d}{du} L(\mathrm{s}, t + u). \tag{26}$$

Therefore the derivatives of the left-hand side of Eq. 25 are equal too:

$$\begin{aligned} \frac{\partial L}{\partial t}\left(S^0(\mathrm{s}, u), t\right) &= \frac{d}{du} L\left(S^0(\mathrm{s}, u), t\right) && (27) \\ &= -\frac{\partial L}{\partial s}\left(S^0(\mathrm{s}, u), t\right) \cdot \frac{\partial E}{\partial s}\left(S^0(\mathrm{s}, u)\right). && (28) \end{aligned}$$

Here we have used the differential equation of motion (Eq. 1). Evaluating this expression for $u = 0$ we get Eq. 24.
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 Eq. 10. Differentiating Eq. 24 with respect to s, we get

$$\frac{\partial^2 L}{\partial t \partial s}(\mathrm{s}, t) + \frac{\partial^2 L}{\partial s^2}(\mathrm{s}, t) \cdot \frac{\partial E}{\partial s}(\mathrm{s}) + \frac{\partial L}{\partial s}(\mathrm{s}, t) \cdot \frac{\partial^2 E}{\partial s^2}(\mathrm{s}) = 0. \tag{29}$$

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

$$\frac{d}{dt} \frac{\partial L}{\partial s}\left(s^0, t\right) = -\frac{\partial^2 E}{\partial s^2}\left(s^0\right) \cdot \frac{\partial L}{\partial s}\left(s^0, t\right). \tag{30}$$

Therefore $\overline{S}_t = \frac{\partial L}{\partial s}\left(s^0, t\right)$ satisfies Eq. 10.
We prove Eq. 11 similarly. Differentiating Eq. 24 with respect to θ, we get

$$\begin{aligned} \frac{\partial^2 L_\theta}{\partial t \partial \theta}(\mathrm{s}, t) + \frac{\partial^2 L_\theta}{\partial s \partial \theta}(\mathrm{s}, t) \cdot \frac{\partial E_\theta}{\partial s}(\mathrm{s}) & \\ + \frac{\partial L_\theta}{\partial s}(\mathrm{s}, t) \cdot \frac{\partial^2 E_\theta}{\partial s \partial \theta}(\mathrm{s}) &= 0. && (31) \end{aligned}$$

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

$$\frac{d}{dt} \frac{\partial L_\theta}{\partial \theta}\left(s_\theta^0, t\right) = -\frac{\partial^2 E_\theta}{\partial \theta \partial s}\left(s_\theta^0\right) \cdot \frac{\partial L_\theta}{\partial s}\left(s_\theta^0, t\right). \tag{32}$$

Hence the result. □

B Equilibrium Propagation - Proof

In this Appendix we prove Theorem 2. The same proof was already 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"

$$\frac{\partial J}{\partial \theta}(\theta) = \lim_{\beta \to 0} \frac{1}{\beta} \left(\frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right) - \frac{\partial E_\theta^0}{\partial \theta}\left(s_\theta^0\right) \right). \tag{33}$$

The gradient formula Eq. 33 is a particular case of the following formula [3], when evaluated at the point $\beta = 0$:

$$\frac{d}{d\theta} \frac{\partial E_\theta^\beta}{\partial \beta}\left(s_\theta^\beta\right) = \frac{d}{d\beta} \frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right). \tag{34}$$

[2] Eq. 24 is the Kolmogorov backward equation for deterministic processes.

[3] 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 Eq. 33, it is sufficient to prove Eq. 34.

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

$$\frac{d}{d\theta}\frac{d}{d\beta}E_\theta^\beta\left(s_\theta^\beta\right) = \frac{d}{d\beta}\frac{d}{d\theta}E_\theta^\beta\left(s_\theta^\beta\right). \tag{35}$$

Second, by the chain rule of differentiation we have

$$\begin{aligned}\frac{d}{d\beta}E_\theta^\beta\left(s_\theta^\beta\right) &= \frac{\partial E_\theta^\beta}{\partial \beta}\left(s_\theta^\beta\right) + \frac{\partial E_\theta^\beta}{\partial s}\left(s_\theta^\beta\right)\cdot\frac{\partial s_\theta^\beta}{\partial \beta} \qquad (36)\\ &= \frac{\partial E_\theta^\beta}{\partial \beta}\left(s_\theta^\beta\right). \qquad (37)\end{aligned}$$

Here we have used the fixed point condition

$$\frac{\partial E_\theta^\beta}{\partial s}\left(s_\theta^\beta\right) = 0. \tag{38}$$

Similarly we have

$$\frac{d}{d\theta}E_\theta^\beta\left(s_\theta^\beta\right) = \frac{\partial E_\theta^\beta}{\partial \theta}\left(s_\theta^\beta\right). \tag{39}$$

Plugging Eq. 37 and Eq. 39 in Eq. 35, we get Eq. 34. Hence the result. □

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 Eq. 8, Eq. 9, Eq. 10 and Eq. 11 (Theorem 1). We will conclude by using the uniqueness of the solution to the differential equation with initial condition. First of all, note that

$$\begin{aligned}&\left.\frac{\partial^2 S_\theta^\beta}{\partial\beta\partial t}\right|_{\beta=0}\left(s_\theta^0, t\right)\\ &= \lim_{\beta\to 0}\frac{1}{\beta}\left(\frac{\partial S_\theta^\beta}{\partial t}\left(s_\theta^0,t\right) - \frac{\partial S_\theta^0}{\partial t}\left(s_\theta^0,t\right)\right) \qquad (40)\\ &= \lim_{\beta\to 0}\frac{1}{\beta}\frac{\partial S_\theta^\beta}{\partial t}\left(s_\theta^0,t\right). \qquad (41)\end{aligned}$$

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

$$\begin{aligned}&\left.\frac{d}{d\beta}\right|_{\beta=0}\frac{\partial E_\theta^\beta}{\partial\theta}\left(S_\theta^\beta\left(s_\theta^0,t\right)\right)\\ &= \lim_{\beta\to 0}\frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial\theta}\left(S_\theta^\beta\left(s_\theta^0,t\right)\right) - \frac{\partial E_\theta^0}{\partial\theta}\left(S_\theta^0\left(s_\theta^0,t\right)\right)\right) \qquad (42)\\ &= \lim_{\beta\to 0}\frac{1}{\beta}\left(\frac{\partial E_\theta^\beta}{\partial\theta}\left(S_\theta^\beta\left(s_\theta^0,t\right)\right) - \frac{\partial E_\theta}{\partial\theta}\left(s_\theta^0\right)\right). \qquad (43)\end{aligned}$$

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

$$\widetilde{S}_t = -\left.\frac{\partial^2 S_\theta^\beta}{\partial\beta\partial t}\right|_{\beta=0}\left(s_\theta^0,t\right), \qquad \forall t \geq 0, \tag{44}$$

$$\widetilde{\Theta}_t = \left.\frac{d}{d\beta}\right|_{\beta=0}\frac{\partial E_\theta^\beta}{\partial\theta}\left(S_\theta^\beta\left(s_\theta^0,t\right)\right), \qquad \forall t \geq 0. \tag{45}$$

Now we prove that $\widetilde{S}_t$ is the solution of Eq. 8 and Eq. 10. We omit to write 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

$$\frac{\partial S^\beta}{\partial t}\left(s^0,t\right) = -\frac{\partial E^\beta}{\partial s}\left(S^\beta\left(s^0,t\right)\right). \tag{46}$$

with initial condition $S^\beta\left(s^0,0\right) = s^0$. Differentiating Eq. 46 with respect to β, we get

$$\begin{aligned}\frac{d}{dt}\frac{\partial S^\beta}{\partial\beta}\left(s^0,t\right) = &-\frac{\partial^2 E^\beta}{\partial s\partial\beta}\left(S^\beta\left(s^0,t\right)\right)\\ &-\frac{\partial^2 E^\beta}{\partial s^2}\left(S^\beta\left(s^0,t\right)\right)\cdot\frac{\partial S^\beta}{\partial\beta}\left(s^0,t\right).\end{aligned} \tag{47}$$

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

$$\begin{aligned}\frac{d}{dt}\left.\frac{\partial S^\beta}{\partial\beta}\right|_{\beta=0}\left(s^0,t\right) = &-\frac{\partial C}{\partial s}\left(s^0\right)\\ &-\frac{\partial^2 E}{\partial s^2}\left(s^0\right)\cdot\left.\frac{\partial S^\beta}{\partial\beta}\right|_{\beta=0}\left(s^0,t\right).\end{aligned} \tag{48}$$

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

$$\frac{\partial S^\beta}{\partial\beta}\left(s^0,0\right) = 0. \tag{49}$$

Therefore, evaluating Eq. 48 at $t = 0$, we get the initial condition (Eq. 8)

$$\widetilde{S}_0 = - \left. \frac{\partial^2 S^\beta}{\partial t \partial \beta} \right|_{\beta=0} \left(s^0, 0\right) = \frac{\partial C}{\partial s} \left(s^0\right). \tag{50}$$

Moreover, differentiating Eq. 48 with respect to time we get

$$\frac{d}{dt} \left. \frac{\partial^2 S^\beta}{\partial t \partial \beta} \right|_{\beta=0} \left(s^0, t\right) = -\frac{\partial^2 E}{\partial s^2} \left(s^0\right) \cdot \left. \frac{\partial^2 S^\beta}{\partial t \partial \beta} \right|_{\beta=0} \left(s^0, t\right). \tag{51}$$

Hence Eq. 10:

$$\frac{d}{dt} \widetilde{S}_t = -\frac{\partial^2 E}{\partial s^2} \left(s^0\right) \cdot \widetilde{S}_t. \tag{52}$$

Now we prove the result for $\widetilde{\Theta}_t$ (Eq. 9 and Eq. 11). 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:

$$\begin{aligned} \frac{d}{d\beta} \frac{\partial E_\theta^\beta}{\partial \theta} \left(S_\theta^\beta \left(s_\theta^0, t\right)\right) = & \frac{\partial E_\theta^\beta}{\partial \theta \partial \beta} \left(S_\theta^\beta \left(s_\theta^0, t\right)\right) \\ & + \frac{\partial E_\theta^\beta}{\partial \theta \partial s} \left(S_\theta^\beta \left(s_\theta^0, t\right)\right) \cdot \frac{\partial S_\theta^\beta}{\partial \beta} \left(s_\theta^0, t\right). \end{aligned} \tag{53}$$

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

$$\begin{aligned} \left. \frac{d}{d\beta} \right|_{\beta=0} \frac{\partial E_\theta^\beta}{\partial \theta} \left(S_\theta^\beta \left(s_\theta^0, t\right)\right) = & \frac{\partial C_\theta}{\partial \theta} \left(s_\theta^0\right) \\ & + \frac{\partial E_\theta}{\partial \theta \partial s} \left(s_\theta^0\right) \cdot \left. \frac{\partial S_\theta^\beta}{\partial \beta} \right|_{\beta=0} \left(s_\theta^0, t\right). \end{aligned} \tag{54}$$

Evaluating Eq. 54 at time $t = 0$ and using Eq. 49 we get the initial condition (Eq. 9)

$$\widetilde{\Theta}_0 = \left. \frac{d}{d\beta} \right|_{\beta=0} \frac{\partial E_\theta^\beta}{\partial \theta} \left(S_\theta^\beta \left(s_\theta^0, 0\right)\right) = \frac{\partial C_\theta}{\partial \theta} \left(s_\theta^0\right). \tag{55}$$

Moreover, differentiating Eq. 54 with respect to time we get

$$\frac{d}{dt} \left. \frac{d}{d\beta} \right|_{\beta=0} \frac{\partial E_\theta^\beta}{\partial \theta} \left(S_\theta^\beta \left(s_\theta^0, t\right)\right) = \frac{\partial E_\theta}{\partial \theta \partial s} \left(s_\theta^0\right) \cdot \left. \frac{\partial^2 S_\theta^\beta}{\partial t \partial \beta} \right|_{\beta=0} \left(s_\theta^0, t\right). \tag{56}$$

Hence Eq. 11:

$$\frac{d}{dt} \widetilde{\Theta}_t = -\frac{\partial E_\theta}{\partial \theta \partial s} \left(s_\theta^0\right) \cdot \widetilde{S}_t. \tag{57}$$

This completes the proof. □