Exploring the global dynamics of networks trained through equilibrium propagation.

Exploring the Global Dynamics of Networks Trained through Equilibrium Propagation

Gianluca Zoppo, Fernando Corinto, and Marco Gilli
Department of Electronics and Telecommunications - Politecnico di Torino - Torino - Italy
Email: gianluca.zoppo@polito.it, fernando.corinto@polito.it, marco.gilli@polito.it

***Abstract*—Equilibrium propagation is a learning technique conceived for training continuous-time recurrent neural networks. It offers some notable advantages when compared to conventional back-propagation-based algorithms and to classical design methods. From an implementation perspective, it demands only a single computational circuit. Theoretically, although it seeks to minimize a cost function, it exhibits similarities to spike-timing-dependent plasticity (STDP), rendering it, to a certain extent, biologically plausible. This paper explores the global dynamic behavior of continuous-time piecewise linear networks trained through equilibrium point propagation. We examine a network in which the target patterns are presented as external inputs rather than as initial conditions. We first show that the learning rules, which extend equilibrium propagation to gradient-like and non-symmetric networks, can be derived as a suitable approximation of Lagrangian optimization. Then, by studying a relatively simple but thoroughly significant case, we demonstrate that a detailed analysis of the equilibrium point distribution yields a profound understanding of the network's fundamental properties and provides a valuable tool for quantitatively evaluating the network's accuracy. Compared to classical synthesis techniques, our approach, where patterns are introduced as external inputs, in most cases, circumvents the impractical task of estimating the basins of attraction for sets of multiple equilibrium points. Furthermore, preliminary extensive simulations indicate that the primary dynamic features observed in relatively small networks closely resemble those ensuring the performance and accuracy of large-scale networks.**

I. Introduction

Within the context of biological systems, "learning" denotes the result of gradual changes occurring at the neuronal synaptic level. It is crucial to recognize that synaptic adjustments and overarching tasks operate on distinct planes - individual synapses lack awareness of the broader learning objectives. This prompts the existence of underlying principles governing localized synaptic modifications, which enable the system to accomplish the overall learning process. Nonetheless, these foundational rules remain widely undisclosed.

A groundbreaking local learning rule was introduced by Donald O. Hebb, back in 1962 [1]. Hebb's insight suggested that synaptic changes take place when there is a coincidental firing of pre- and post-synaptic neurons. Specifically, the connection between two neurons is strengthened if they both activate concurrently in response to external input, and this connection weakens if they fire separately. This seemingly straightforward rule, known as the Hebbian learning rule, holds profound significance. It is recognized for its potential to induce robust self-organizing effects, particularly within the context of relatively simple neural network models.

From an alternative standpoint, gradient descent and, primarily, back-propagation algorithms have served as the central tools for training diverse types of neural networks. Paired with the abundance of data and unparalleled computational resources, they form the foundation of the remarkable advancements in neural network applications, encompassing the most cutting-edge Artificial Intelligence applications.

In the framework of continuous-time neural networks, specifically focusing on continuous-time Hopfield networks, Almeida and Pineda independently introduced a recurrent back-propagation algorithm [2], [3]. The core idea is to force the network to converge onto a desired fixed-point attractor, starting from a given input and initial state, through an iterative process aimed at adjusting the synaptic weight matrix of a dynamic neural network. Similar to feed-forward neural networks, this is achieved by minimizing a specific loss function tied to the network parameters. The novelty lies in the approach of "back propagating" the error signal using an analog auxiliary network, involving a related differential equation which avoids the computation of the inverse of the Jacobian. Nevertheless, the need for a side network to propagate error derivatives makes this technique markedly different from emulating the computational processes of the brain. In general back-propagation algorithms are not considered to be biologically plausible, because experimental observation in neuroscience shows that the change of synaptic weights is primarily due to the timing difference between post- and pre-synaptic spikes (Spike-Timing-Dependent-Plasticity [4]).

In [5] Scellier and Bengio introduced a new learning framework for energy based models, named "Equilibrium Propagation" which requires only one computational circuit and, despite minimizing a cost function, may be interpreted as a STDP approach (see also [6]). By exploiting a rather different technique, partially based on linear algebra, some efficient methods were proposed for designing Hopfield-like continuous-time networks. In particular, Michel et al. developed a series of algorithms for synthesizing network with piecewise linear output functions, with a prescribed number of stable equilibrium points, that behaves as associative memories [7].

In this manuscript, we thoroughly analyze the global dynamic behavior of continuous-time networks trained using equilibrium point learning. We focus on networks featuring

piecewise linear outputs, where target patterns are presented as constant inputs. Initially, we show that the local learning rule presented in [8] for the non-symmetric case can be derived as a precise approximation of Lagrangian optimization. Subsequently, we delve into exploring the dynamic structure inherent in networks trained through equilibrium propagation. By concentrating on relatively straightforward yet notable scenarios that enable the direct computation of all stable equilibrium points, we systematically investigate the network's convergence for each distinct input pattern. This provides a deep understanding and an almost analytical assessment of the accuracy of equilibrium learning methodologies, also facilitating a thorough comparison with the outcomes obtained through exact Lagrangian optimization. Furthermore, when compared with conventional neural network synthesis, our approach, where patterns are introduced as external inputs, often mitigates the impractical challenge of estimating the basins of attraction for extensive sets of equilibrium points. Finally, preliminary simulations suggest that the principal dynamic characteristics observed in relatively small networks closely mirror those responsible for the performance and precision of larger-scale networks [9], [10], [11].

II. Neural Network Model

We consider a continuous-time recurrent neural network, composed of N cells and described by the following nonlinear differential equation in matrix form:

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},\mathbf{W},\mathbf{b},\mathbf{I}) &= -\mathbf{x}+\mathbf{W}g(\mathbf{x})+\mathbf{b}+\mathbf{I} \\ \\ \mathbf{y} &= g(\mathbf{x}) \end{cases} \tag{1}$$

where $\mathbf{x}$ represents the state, $\mathbf{I} \in R^N$ is the input vector, $\mathbf{W} \in R^{N,N}$ is the interconnecting weight matrix, and $\mathbf{b} \in R^N$ is the usual constant bias term. The neural output $\mathbf{y} = g(\mathbf{x}) = [g(x_1),...,g(x_N)]^T$ is defined through the following piecewise linear monotone increasing function $g(\cdot)$:

$$g(x_i) = \frac{1}{2}(|x_i+1| - |x_i-1|) \tag{2}$$

Equilibrium point learning and synthesis techniques aim to design a network that exhibits a prescribed set of equilibrium points. In general, the objective is to determine the weights W_{ij} and the bias b_i such that, for any initial condition $\mathbf{x}^0$ within a given set, the system output converges towards a specified target value $\mathbf{T}$. In practical applications, it is often challenging to specify initial conditions with accuracy. As an alternative approach, one can require that for each input within a given set, the system output converges to a predefined target value, regardless of the initial condition. This alternative approach not only makes implementation in physical systems more feasible but also simplifies the analysis of the network global dynamic behavior. In addition, the network weights can be efficiently implemented through memristor devices [12]. As shown in [13], the adjustable connections are implemented by using memristors as programmable resistors coupled in a differential pair configuration.

The problem can be addressed by minimizing a suitable cost function, $\mathscr{C}(\mathscr{T},\mathscr{Y}) = \sum_k \mathbf{C}[\mathbf{T}^k, g(\hat{\mathbf{x}}^k)]$ with $\mathscr{T} = \{\mathbf{T}^1,\ldots,\mathbf{T}^m\}$ and $\mathscr{Y} = \{g(\hat{\mathbf{x}}^1),\ldots g(\hat{\mathbf{x}}^m)\}$, which quantifies the discrepancy between the selected set of target patterns and the outputs at the corresponding equilibria $\hat{\mathbf{x}}^k$. Furthermore, in the expression of the cost function, it is convenient to replace the piecewise linear function $g(\cdot)$ with a more manageable smooth function, specifically $h(\cdot) = \tanh(\cdot)$.

The matrix weights and the bias terms can be accordingly updated, by iteratively moving in the direction of the steepest descent:

$$\begin{aligned} \Delta W_{ij} &= -\eta \frac{\partial \mathscr{C}}{\partial W_{ij}}(\mathscr{T},\mathscr{Y}) = -\eta \sum_{k=1}^{m} \frac{\partial \mathbf{C}}{\partial W_{ij}}[\mathbf{T}^k, h(\hat{\mathbf{x}}^k)] \\ \Delta b_i &= -\eta \frac{\partial \mathscr{C}}{\partial b_i}(\mathscr{T},\mathscr{Y}) = -\eta \sum_{k=1}^{m} \frac{\partial \mathbf{C}}{\partial b_i}[\mathbf{T}^k, h(\hat{\mathbf{x}}^k)] \end{aligned} \tag{3}$$

Due to the linearity of differentiation, without loss of generality, we can focus on the estimation of the gradient for a single pattern. According to the approach followed in [5] and [14], the optimization problem can be recast as follows:

$$\min_{\mathbf{W},\mathbf{b}} \mathbf{C}[\mathbf{T}, h(\mathbf{x})] \quad \text{subject to} \quad \mathbf{f}(\mathbf{x},\mathbf{W},\mathbf{b},\mathbf{I}) = \mathbf{0} \tag{4}$$

The constrained optimization problem defined in Eq. (4) can be readily addressed, by introducing a Lagrange multiplier $\boldsymbol{\lambda} \in \mathbb{R}^N$, and considering the Lagrangian function defined by:

$$\mathbf{L}(\mathbf{x},\boldsymbol{\lambda},\mathbf{W},\mathbf{b},\mathbf{T},\mathbf{I}) = \mathbf{C}[\mathbf{T}, h(\mathbf{x})] + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x},\mathbf{W},\mathbf{b},\mathbf{I}). \tag{5}$$

By keeping the entries of the weight matrix $\mathbf{W}$ and of the bias term $\mathbf{b}$ constant and solving for $(\boldsymbol{\lambda},\mathbf{x})$, we obtain:

$$\begin{cases} \nabla_{\boldsymbol{\lambda}} \mathbf{L} = \mathbf{f}(\mathbf{x},\mathbf{W},\mathbf{b},\mathbf{I}) = \mathbf{0} \quad \rightarrow \quad \mathbf{x} = \hat{\mathbf{x}} \text{ equilibrium of (1)} \\ \nabla_{\mathbf{x}} \mathbf{L} = H'(\mathbf{x}) \nabla_{h(\mathbf{x})} \mathbf{C}[\mathbf{T}, h(\mathbf{x})] + \mathbf{J}_{\mathbf{f}}^T(\mathbf{x})\,\boldsymbol{\lambda} = \mathbf{0} \end{cases} \tag{6}$$

where $H'(\hat{\mathbf{x}}) = \text{diag}[h'(\hat{x}_1),\ldots,h'(\hat{x}_n)]$, ∇ denotes the gradient, and $\mathbf{J}_{\mathbf{f}}(\mathbf{x})$ the Jacobian of function $\mathbf{f}$. The expression of $\boldsymbol{\lambda}$ is promptly derived:

$$\hat{\boldsymbol{\lambda}} = -[\mathbf{J}_{\mathbf{f}}^T(\hat{\mathbf{x}})]^{-1} H'(\hat{\mathbf{x}}) \nabla_{h(\hat{\mathbf{x}})} \mathbf{C}[\mathbf{T}, h(\hat{\mathbf{x}})] \tag{7}$$

The corresponding variation of the weight matrix and bias terms, due to the pattern $\mathbf{T}$ can accordingly be computed as:

$$\begin{aligned} \Delta \mathbf{W} &= -\eta \frac{\partial \mathbf{L}}{\partial \mathbf{W}} = -\eta\, \hat{\boldsymbol{\lambda}} \cdot [g(\hat{\mathbf{x}})]^T \\ \Delta \mathbf{b} &= -\eta \frac{\partial \mathbf{L}}{\partial \mathbf{b}} = -\eta\, \hat{\boldsymbol{\lambda}} \end{aligned} \tag{8}$$

The above expression formally holds for generic, even non symmetric networks, but requires the computation of the inverse of a Jacobian matrix and some additional matrix multiplications, which are not feasible for large systems.

An alternative and effective approach to calculating the Lagrange multiplier $\hat{\boldsymbol{\lambda}}$ involves examining the following augmented dynamical system derived from the equilibrium propagation approach, as introduced in [5]:

$$\frac{d\mathbf{x}^{\beta}}{dt} = \mathbf{f}^{\beta}(\mathbf{x}^{\beta},\mathbf{W},\mathbf{b},\mathbf{I}) = \mathbf{f}(\mathbf{x}^{\beta},\mathbf{W},\mathbf{b},\mathbf{I}) - \beta\nabla_{h(x^{\beta})}\mathbf{C}[\mathbf{T},h(\mathbf{x}^{\beta})] \tag{9}$$

where $\beta > 0$ is a forcing parameter. A fixed point $\hat{\mathbf{x}}^{\beta}$ of the new dynamical system satisfies

$$\mathbf{f}^{\beta}(\hat{\mathbf{x}}^{\beta},\mathbf{W},\mathbf{b},\mathbf{I}) = \mathbf{0}. \tag{10}$$

Since (10) is constant for all β, the total derivative with respect to β evaluated at $\beta = 0$ gives:

$$\mathbf{J}_{\mathbf{f}}(\hat{\mathbf{x}})\left.\frac{\partial\hat{\mathbf{x}}^{\beta}}{\partial\beta}\right|_{\beta=0} - \nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})] = 0 \tag{11}$$

where $\hat{\mathbf{x}}^{\beta}|_{\beta=0} = \hat{\mathbf{x}}$. By comparing (7) and (11) we obtain:

$$\hat{\boldsymbol{\lambda}} = -\mathbf{J}_{\mathbf{f}}^{T}(\hat{\mathbf{x}})^{-1}H'(\hat{\mathbf{x}})\mathbf{J}_{\mathbf{f}}(\hat{\mathbf{x}})\left.\frac{\partial\hat{\mathbf{x}}^{\beta}}{\partial\beta}\right|_{\beta=0}. \tag{12}$$

We shall show that certain advantageous geometric characteristics of Lagrangian multipliers, can be used to accurately approximate the gradient descent computation, without computing the Jacobian inverse, which is an unfeasible task. This also offers an alternative derivation of the local learning rules introduced in [8], which only relies on the Lagrangian multiplier approach. The results are based on the following proposition:

Proposition 1. *The following property hold:*

$$\nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})]^{T}\hat{\boldsymbol{\lambda}} = -\nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})]^{T}\left.\frac{\partial h(\hat{\mathbf{x}}^{\beta})}{\partial\beta}\right|_{\beta=0} \tag{13}$$

Proof. The above assertion can be proved by considering a system comprising equation (11) and the transpose of equation (7):

$$\begin{cases}\mathbf{J}_{\mathbf{f}}(\hat{\mathbf{x}})\left.\frac{\partial\hat{\mathbf{x}}^{\beta}}{\partial\beta}\right|_{\beta=0} - \nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})] = \mathbf{0}\\ \hat{\boldsymbol{\lambda}}^{T}\mathbf{J}_{\mathbf{f}}(\hat{\mathbf{x}}) + \nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})]^{T}H'(\hat{\mathbf{x}}) = \mathbf{0}^{T}\end{cases} \tag{14}$$

By multiplying the first equation with the row vector $\hat{\boldsymbol{\lambda}}^{T}$ it is obtained:

$$\begin{cases}\hat{\boldsymbol{\lambda}}^{T}\mathbf{J}_{\mathbf{f}}(\hat{\mathbf{x}})\left.\frac{\partial\hat{\mathbf{x}}^{\beta}}{\partial\beta}\right|_{\beta=0} - \hat{\boldsymbol{\lambda}}^{T}\nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})] = \mathbf{0}\\ \hat{\boldsymbol{\lambda}}^{T}\mathbf{J}_{\mathbf{f}}(\hat{\mathbf{x}}) + \nabla_{h(\mathbf{x})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})]^{T}H'(\hat{\mathbf{x}}) = \mathbf{0}\end{cases} \tag{15}$$

By substituting the second equation into the first and using the symmetry property of the dot product it follows:

$$\nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})]^{T}\hat{\boldsymbol{\lambda}} = -\nabla_{h(\hat{\mathbf{x}})}\mathbf{C}[\mathbf{T},h(\hat{\mathbf{x}})]^{T}H'(\hat{\mathbf{x}})\left.\frac{\partial\hat{\mathbf{x}}^{\beta}}{\partial\beta}\right|_{\beta=0} \tag{16}$$

which yields (13) and proves the proposition. ∎

Equation (13) shows that along the gradient's direction of the cost function $\mathbf{C}$ at $(\mathbf{T},h(\hat{\mathbf{x}}))$ the projection of the vector $-\left.\frac{\partial h(\hat{\mathbf{x}}^{\beta})}{\partial\beta}\right|_{\beta=0}$ equals the projection of the vector $\hat{\boldsymbol{\lambda}}$. By considering that, as a consequence of the Lagrangian method, the gradients of $\mathbf{C}$ and $\mathbf{L}$ have the same direction, it turns out that the information provided by Eq. (12) is somewhat redundant and one can consider $\hat{\boldsymbol{\lambda}} = -\left.\frac{\partial h(\hat{\mathbf{x}}^{\beta})}{\partial\beta}\right|_{\beta=0}$. Using (8) the update rule for the system can consequently be written without explicitly computing the Lagrangian multipliers and without requiring the matrix symmetry:

$$\begin{aligned}\Delta\mathbf{W} &= \eta\left(\left.\frac{\partial h(\hat{\mathbf{x}}^{\beta})}{\partial\beta}\right|_{\beta=0}\right)\cdot[g(\hat{\mathbf{x}})]^{T}\\ \Delta\mathbf{b} &= \eta\left(\left.\frac{\partial h(\hat{\mathbf{x}}^{\beta})}{\partial\beta}\right|_{\beta=0}\right)\end{aligned} \tag{17}$$

III. Analysis of the Global Dynamic Behavior

Analyzing the dynamics yields a comprehensive grasp of the equilibrium point learning process. To demonstrate the potential and utility of such an investigation, we will examine a network comprising ten cells, as initially described in [7]. We assume that the network should memorize five target patterns, α_i, $i = 1,\ldots 5$, which represents 50% of the network nodes, thereby rendering this system noteworthy for investigation:

$$\alpha_1=\begin{bmatrix}-1\\1\\-1\\1\\1\\1\\-1\\1\\1\\1\end{bmatrix}\ \alpha_2=\begin{bmatrix}1\\1\\-1\\-1\\1\\-1\\1\\-1\\1\\1\end{bmatrix}\ \alpha_3=\begin{bmatrix}-1\\1\\1\\1\\-1\\-1\\1\\-1\\-1\\-1\end{bmatrix}\ \alpha_4=\begin{bmatrix}1\\1\\-1\\1\\-1\\1\\-1\\1\\1\\1\end{bmatrix}\ \alpha_5=\begin{bmatrix}1\\-1\\-1\\-1\\1\\1\\1\\-1\\-1\\-1\end{bmatrix} \tag{18}$$

Though formally referred to a linear system operating on a closed hypercube, it is easily derived that the system studied in [7] can be described by (1), assuming that the input term is presented as an initial condition and consequently $\mathbf{I} = 0$. It turns out that the network synthesized in [7] exhibits 13 stable equilibrium points: five of them lie in saturation regions corresponding to the five stored patterns, i.e. $g(\hat{\mathbf{x}}) = \alpha_i$, but eight of them are spurious, and lie in regions, where at least one variable is not saturated. In addition there are 70 unstable equilibrium points. The presence of spurious equilibrium points shows that there are initial conditions, which do not converge to α_i. We may presume that the initial conditions with minimal hamming distance from the stored patterns converge to one of them, but this assertion cannot be proved without estimating the basin of attraction of the stable equilibrium points, which is a formidable and impractical task even for very simple networks (see [7] where a subset of the basins of attraction is determined for a network of only three cells).

If the network input is presented as an external input $\mathbf{I}$, a thorough examination of the equilibrium point distribution in the resulting system serves as a valuable means for evaluating the overall accuracy of the network. It is noteworthy that, due to the piecewise linear architecture of the system described by equation (1) and the relatively modest number of cells

Method	η	β	d_H = 1 from one target			$d_H = 1$ from two targets		
			C	*AC*	*IC*	*C*	*AC*	*IC*
Lagrangian Multipliers	0.3	–	46	0	0	2	0	0
	0.5	–	46	0	0	2	0	0
Modified Eq. Prop.	0.5	0.2	38	8	0	2	0	0
	2	0.2	30	16	0	2	0	0

Table I: *Network trained through the LM and the MEP approach. Equilibrium point analysis, considering inputs* $\mathbf{I}$ *with Hamming distance* $d_H = 1$ *from one or more target patterns (*46 *inputs with* $d_H = 1$ *from a single pattern* α_i *and 2 inputs with* $d_H = 1$ *from two patterns* α_i, α_j*). The numerical values have the following meaning. A) Network Inputs with* $d_H = 1$ *from a single pattern* α_i*: C - number of inputs for which the system exhibits only one stable equilibrium point such that* $\text{sgn}(\hat{\mathbf{x}}) = \alpha_i$*; PC - number of inputs for which the system exhibits more than one stable equilibrium point, such that for at least one of them* $\text{sgn}(\hat{\mathbf{x}}) = \alpha_i$*; IC - number of inputs for which the system does not exhibit equilibrium points such that* $\text{sgn}(\hat{\mathbf{x}}) = \alpha_i$*. B) Network inputs with* $d_H = 1$ *from two patterns* α_i, α_j*: C - number of inputs for which the system exhibits either one or two stable equilibrium points such that for all of them* $\text{sgn}(\hat{\mathbf{x}}) \in \{\alpha_i, \alpha_j\}$*; PC - number of inputs for which the system exhibits more than one stable equilibrium point, such that for at least one of them* $\text{sgn}(\hat{\mathbf{x}}) \in \{\alpha_i, \alpha_j\}$*; IC - number of inputs for which no equilibrium point satisfies* $\text{sgn}(\hat{\mathbf{x}}) \in \{\alpha_i, \alpha_j\}$.

Method	η	β	d_H = 2 from one target			$d_H = 2$ from two targets		
			C	*AC*	*IC*	*C*	*AC*	*IC*
Lagrangian Multipliers	0.3	–	135	23	21	15	2	5
	0.5	–	113	53	13	15	4	3
Modified Eq. Prop.	0.5	0.2	42	133	4	12	8	2
	2	0.2	66	108	5	14	6	2

Table II: *Network trained through the LM and the MEP approach. Equilibrium point analysis, considering inputs* $\mathbf{I}$ *with* $d_H = 2$ *from one of more patterns* α_i *(excluding two inputs that also have* $d_H = 1$ *from some target patterns, there are* 179 *inputs which exhibit* $d_H = 2$ *from a single pattern* α_i *and 22 inputs with* $d_H = 1$ *from two patterns* α_i, α_j*). The numerical values have the same meaning of Table I.*

involved, it is feasible to precisely compute all equilibria, whether stable or unstable. This computation is achieved by systematically inspecting the 3^{10} linear cells comprising the state space.

We have trained the network by exploiting both the Lagrangian multipliers (LM) approach synthesized by (7) and (8) and the modified equilibrium propagation (MEP) method described by (17). When applying these learning rules, the resulting matrix $\mathbf{W}$ may exhibit slight asymmetry. In most cases, the overall system remains stable, although this cannot be rigorously proved. To avoid this problem and make our analysis robust, we have symmetricized matrix $\mathbf{W}$ at each update, so to ensure the network's complete stability.

Through the investigation of the equilibria distribution, summarized in Tables I and II, it is observed that the proposed modified equilibrium propagation method demonstrates a notably high level of accuracy, closely approaching the results achieved through Lagrangian optimization.

Table I provides evidence that, when considering input patterns with a Hamming distance (d_H) of 1 from the target patterns α_i, the system trained through equilibrium propagation, typically exhibits a single stable equilibrium point corresponding to the correct output. As a result, it converges to the correct output from any initial condition. In some instances, a limited number of stable equilibrium points emerge, but they consistently include the correct output. Notably, no cases were found where the correct output was absent among the stable equilibria of the network. Moreover, extensive simulations revealed that when multiple equilibria exist, the basin of attraction associated with the correct output is substantially larger compared to others. Consequently, the overall accuracy of the network for inputs with $d_H = 1$ is nearly 100%.

Table II illustrates the scenario encountered when analyzing inputs with a Hamming distance (d_H) of 2 from the target patterns α_i. It is observed that in the vast majority of cases, the correct output is included within the stable equilibrium points, with occurrences where it is absent being exceedingly rare, amounting to less than 3.1% of the total cases (specifically, $7/223$). This outcome underscores the system's noteworthy accuracy.

Finally, a comparison with [7] shows that presenting the input patterns as external inputs, not only is more feasible for practical implementations, but also enables a comprehensive dynamic analysis, that in most cases does not require to estimate the basins of attraction of multiple equilibrium points.

IV. Conclusion

In this manuscript, we have thoroughly investigated the overall performances of continuous-time networks trained via equilibrium point propagation from a nonlinear dynamic perspective. We have focused on networks in which the target patterns are introduced as external inputs. In relatively straightforward yet highly significant cases, we have demonstrated that the network's performance can be precisely assessed through a detailed analysis of the equilibrium point distribution for each input pattern. Extensive simulations, that for lack of space are not reported in this paper, have revealed that the identical key dynamic properties observed in small networks offer profound insights into the overall behavior and the performances of large-scale networks.

V. Acknowledgment

This work was supported by the Ministero dell'Istruzione, dell'Universitaà della Ricerca (MIUR) under contract 2017LSCR4K-003.

REFERENCES

[1] D. O. Hebb, *The organization of behavior: A neuropsychological theory,* Science Edition, 1962.

[2] L. B, Almeida, "A learning rule for asynchronous perceptrons with feedback in a combinatorial environment," *Int. Conf. Neural Netw.*, vol. 2, pp. 609–618, 1987.

[3] F. J. Pineda, "Generalization of back propagation to recurrent and higher order neural networks," *Neural Inf. Process. Syst.*, pp. 602–611, 1988.

[4] W. Senn and J. P Pfister, "Spike-timing-dependent plasticity, Learning rules," *Encyclopedia of Computational Neuroscience*, Springer, 2015.

[5] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and back-propagation," *Front. Comput. Neurosci.*, vol. 11, p. 24, 2017.

[6] S. Scellier, "Equilibrium Propagation: A Physics-Grounded Theory of Computation and Learning" *Bulletin of the American Physical Society*, 2023.

[7] J. Li. A N. Michel, and W. Porod, "Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube," *IEEE Trans. Circuits Syst*, vol. 36, no. 11, pp. 1405-1422, 1989.

[8] B. Scellier, A. Goyal, J. Binas, T. Mesnard, and Y. Bengio, "Generalization of equilibrium propagation to vector field dynamics," https://arxiv.org/abs/1808.04873, 2018.

[9] J. Laydevant, M. Ernoult, D. Querlioz, and J. Grollier, "Training dynamical binary neural networks with equilibrium propagation," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4640-4649, 2021.

[10] Z. Ji, and W. Gross, "Towards efficient on-chip learning using equilibrium propagation" *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, 2020.

[11] S. Oh, J. An, S. Cho, R. Yoon, and K. S. Min, "Memristor Crossbar Circuits Implementing Equilibrium Propagation for On-Device Learning", *Micromachines*, 14(7), pp. 1367, 2023.

[12] G. Zoppo, F. Marrone, and F. Corinto, "Equilibrium propagation for memristor-based recurrent neural networks," *Front. Neurosci.*, 2020.

[13] G. Zoppo, F. Marrone, M. Bonnin, and F. Corinto, "Equilibrium propagation and (memristor-based) oscillatory neural networks", *2022 IEEE international symposium on circuits and systems (ISCAS)*, pp. 639-643, 2022.

[14] Y. Lecun, "A theoretical framework for back-propagation," *Proc. Connectionist Models Summer School*, pp. 21-28, 1988.