Equilibrium Propagation for Complete Directed Neural Networks

Matilde Tristany Farinha
matilde.tristany@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2019

Abstract

Artificial neural networks are one of the most successful approaches to supervised learning. However, their most successful learning algorithm, backpropagation, is considered biologically implausible. Many believe that the next generation of artificial neural networks should be built upon a better understanding of biological learning. Therefore, in this work, we contribute to the topic of biologically plausible learning algorithms for artificial neural networks by building upon the equilibrium propagation model: we introduce a novel neuronal dynamics and learning rule for arbitrary network architectures; we developed sparsity-inducing methods, able to prune connections deemed irrelevant; and we provide a dynamical-systems characterization of these models, based on Lyapunov theory.

Keywords: Equilibrium Propagation, Neural Networks, Machine Learning, Biological Learning, Dynamical Systems

1. Introduction

Nowadays, many state-of-the-art approaches to *supervised learning* rely on *artificial neural networks* (ANNs). *Backpropagation* (BP) [**?**], the most successful algorithm to train ANNs [9], is considered bio-implausible since: (*i*) it lacks local error representation; (*ii*) it uses distinct forward and backward information passes; (*iii*) it requires symmetric feedback weights [6]. To help bridging the gap between biological and machine learning, it is thus crucial to find alternatives to BP-based algorithms that encompass properties of biological neural networks.

The *equilibrium propagation* (EP [14]) model adopts a local learning rule and uses just one kind of computation both for the feedforward and feedback information passes, therefore being a more bio-plausible alternative to BP. However, EP has some bio-unrealistic aspects: (*i*) it assumes symmetric feedback connections; (*ii*) it has only been tested on layered architectures; and (*iii*) it does not promote sparse distributed networks [13]. We tackle these problems by introducing the *DirEcted EP* (DEEP) model, which: (*i*) has asymmetric feedback connections; (*ii*) allows for arbitrary network architectures; and (*iii*) promotes sparse distributed networks. DEEP assumes the network is a complete directed graph and its training algorithm actively removes the presumably expendable connections through a sparsity-inducing term. We also establish sufficient conditions for convergence of the neuronal dynamics of DEEP's inference phase.

The remainder of this work is organized as follows. In Section 2, state-of-the-art EP-like models are discussed. Section 3 presents and analyzes the DEEP model. In Section 4, DEEP is experimentally evaluated and the results are analyzed. Finally, Section 5 concludes the paper and discusses future work.

2. Related Work

The original EP is an energy-based model, consisting of a multi-layered continuous Hopfield network of recurrently connected neurons with symmetric feedback weights [14]. The idea is that if the brain behaves similarly to EP, then neurons perform computations by collectively attempting to reach an equilibrium state of a dynamical system, captured as the minimum of an energy function. The performance of EP was analyzed on the MNIST classification task [10], being reported a validation error lying "between 2% and 3%" [14].

Several extensions and adaptations of the original EP model have been proposed. For instance, an asymmetric version of EP (asymmetric feedback connections) was proposed [15]. However, with a complete graph architecture, we observed experimentally that the model is sometimes unable to learn. A bidirectional-EP that works both as generative and discriminative has also been proposed but, although it provides an insightful extension of EP [8], it does not solve the weight-transport problem [5]. Another EP adaptation considers spiking neurons, therefore extending EP's bio-plausibility [12]. Additionally, EP has been extended to convolutional architectures, for which the lowest error rates among EP-like models on the MNIST classification task [10] has been reported – "approximately 1%" [4]. However, this did not improve EP regarding bio-plausibility, since convolutional architectures are considered bio-implausible due to their extensive weight sharing [3]. In a recently submitted paper, which is still under review, an EP-like model was proposed where the learning rule, besides being spatially local, is also temporally local [2]. This temporal locality comes naturally by performing weight updates after each state update in the second phase. Thus, without altering the essence of the learning rule itself, but rather the training algorithmic scheme, the learning rule becomes temporally local.

3. Model

3.1. Background: Equilibrium Propagation (EP)

The structure of EP is completely specified by the following:

- A state vector $\boldsymbol{s}(t) = [s_j(t)]_{j=1}^{N} \in [0,1]^N$, containing the neurons' activities, that is their "firing rates". The state vector includes subvectors that correspond to input, $\boldsymbol{x} \in [0,1]^P$, hidden, $\boldsymbol{h}(t)$, and output, $\hat{\boldsymbol{y}}(t)$, neurons;
- A weighted adjacency matrix $\boldsymbol{W} = [W_{ij}]_{i,j=1}^{N}$, containing the weights of the connection between every two neurons in the network;
- A bias vector $\boldsymbol{b} = [b_j]_{j=1}^{N}$, where $b_j = 0$ for every input neuron j;
- A set of continuous-time differential equations defining its dynamics.

Henceforth, the time dependency of the state variable is omitted to simplify notation. Defining $\theta = (\boldsymbol{W}, \boldsymbol{b})$ and having the input neuron's fixed, the neuronal dynamics is dictated by the energy function[1]

$$F_\theta(\boldsymbol{s}, \boldsymbol{y}, \beta) = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s} - \frac{1}{2}\boldsymbol{s}^T\boldsymbol{W}\boldsymbol{s} - \boldsymbol{s}^T\boldsymbol{b} + \beta C_\theta(\hat{\boldsymbol{y}}, \boldsymbol{y}), \tag{1}$$

(recall that $\hat{\boldsymbol{y}}$ is a subvector of $\boldsymbol{s}$), where $\boldsymbol{y}$ is the vector of target/desired outputs, $C_\theta(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is the cost (e.g., mean squared error – MSE) and β controls how much the cost influences the dynamics.

The algorithm has two distinct phases, dictated by different neuronal dynamics based on the gradient of $F_\theta(\boldsymbol{s}, \boldsymbol{y}, \beta)$ with respect to $\boldsymbol{s}$, for $\beta = 0$ (first phase) and $\beta \neq 0$ (second phase). Therefore, we obtain (where $\dot{s}_j = ds_j/dt$)

$$\dot{s}_j = \sum_{i=1}^{N} W_{ji}s_i + b_j - s_j - \beta\big(s_j - y_j\big)\mathbb{1}_{\hat{\boldsymbol{y}}}(s_j), \tag{2}$$

where $\mathbb{1}_{\hat{\boldsymbol{y}}}(s_j)$ is nonzero if and only if neuron j is an output neuron. In the first and second phases, the network settles to equilibrium states, denoted $\boldsymbol{s}^0$ and $\boldsymbol{s}^\beta$, respectively. The loss function is defined as the cost in $\boldsymbol{s}^0$: $J_\theta(\boldsymbol{s}^0, \boldsymbol{y}) = C_\theta(\boldsymbol{s}^0, \boldsymbol{y})$.

EP takes advantage of the weight symmetry to rewrite the bio-inspired weight dynamics as $\dot{\boldsymbol{W}} \propto d(s_i(t)s_j(t))/dt$ [17]. Integrating throughout the path from $\boldsymbol{s}^0$ to $\boldsymbol{s}^\beta$, a contrastive Hebbian-like learning rule is obtained: $\Delta W_{ij} \propto s_i^\beta s_j^\beta - s_i^0 s_j^0$, which coincides with the loss gradients, up to a factor $1/\beta$. When used for inference, the activities of the inputs neurons are fixed and the network is evolved to equilibrium $\boldsymbol{s}^0$, from which the output is read at the corresponding output neurons.

3.2. DirEcted Equilibrium Propagation (DEEP)

The DEEP model is based on EP, but less restricted and with asymmetric feedback connections. Its neuronal dynamics is dictated by vector fields, rather than the gradients of an energy function, which account for the weighted incoming and outgoing connections. The novel neuronal dynamics is

$$\dot{s}_j = \sum_{i=1}^{N} W_{ji}s_i + b_j - s_j\sum_{i=1}^{N} W_{ij} - \beta\big(s_j - y_j\big)\mathbb{1}_{\hat{\boldsymbol{y}}}(s_j). \tag{3}$$

The dynamics of the first and second phases are given by Equation (3) with $\beta = 0$ and $\beta \neq 0$, respectively. The learning rule proposed is obtained by numerical integration of the weight dynamics $\dot{\boldsymbol{W}} \propto s_i(t)\dot{s}_j(t)$ [17] in the path from $\boldsymbol{s}^0$ to $\boldsymbol{s}^\beta$ in M_β-steps (the time derivatives $\dot{\boldsymbol{s}}$ are approximated by backward differences):

$$\Delta W_{ij} \propto \frac{1}{M_\beta}\sum_{m=M_0+1}^{M_0+M_\beta} s_i(m)\big(s_j(m) - s_j(m-1)\big). \tag{4}$$

Sparsity-inducing ℓ_1 regularization is added to this weight update to promote sparsity which is further enforced when the weights are below a certain threshold. Specifically, for each weight $W_{ij} \in [-\lambda, \lambda]$, with $\lambda > 0$, is randomly removed from the network with probability p_{ij} given by a Boltzmann distribution defined across the incoming weights of neuron j:

$$p_{ij} = e^{-|W_{ij}|/T} \Big/ \sum_{k=0}^{N} e^{-|W_{kj}|/T}. \tag{5}$$

[1] Notation slightly differs from the original paper [14] because each state update is bounded by the hard-sigmoid function.

In this context, the temperature (T) represents how likely it is for stronger connections to be deemed irrelevant.

Algorithm 1 has the pseudo-code of a sparsity-inducing training epoch of the DEEP model. The probability p_θ represents the probabilities calculated by Equation (5). When performing inference, the algorithm runs up to 2. (the first phase), and the predictions are read in the output neurons.

Algorithm 1: Pseudo-code for a sparsity-inducing training epoch of the DEEP model.

```
input: $\boldsymbol{x}$, $\boldsymbol{y}$, $\theta$, $M_0$, $M_\beta$, $\beta$, $dt$, $dt_\theta$, $\gamma_\theta$, $\lambda$, $T$
output: $\theta = (\boldsymbol{W}, \boldsymbol{b})$
  1. (initialization) $\boldsymbol{x} = \boldsymbol{x}^{(l)}$; $\boldsymbol{h}, \hat{\boldsymbol{y}} \sim$ Unif$(0,1)$
  2. (1st phase) for $m = 0$ to $M_0 - 1$ do:
         $\boldsymbol{s}(m+1) \longleftarrow \rho\big(\boldsymbol{s}(m) + dt\Delta_0 \boldsymbol{s}(m)\big)$
  3. store $\boldsymbol{s}^{path}(0) = \boldsymbol{s}(M_0)$
  4. (2nd phase) for $m = M_0$ to $M_\beta - 1$ do:
         $\boldsymbol{s}(m+1) \longleftarrow \big(\boldsymbol{s}(m) + dt\Delta_\beta \boldsymbol{s}(m)\big)$
  5. store $\boldsymbol{s}^{path}(m+1-M_0) = \boldsymbol{s}(m+1)$
  6. ($\theta$ update)
       (a) $\theta \longleftarrow \theta + dt_\theta \Delta\theta - \gamma_\theta sign(\theta)$
       (b) calculate $p_\theta$
       (b) randomly sample $p \sim$ Unif$(0,1)$
       (c) if $\theta \in [-\lambda, \lambda]$ and $p_\theta > p$, then $\theta = 0$
```

3.3. Analytic Properties

3.3.1 Time-invariant Sum of Firing Rates

In the absence of external stimulus (no fixed input or bias neurons), the dynamics in Eq.(3) preserves the sum of firing-rates across time (*i.e.*, $\sum_j \dot{s}_j(t) = 0,\ t \in \mathbb{R}_0^+$). This property is bioplausible as it has been reported that, in the absence of external sensory stimulus or motor activity, the grand mean firing rate of the hippocampal neurons remains constant [7].

3.3.2 A General Stability Test

Let us consider the time-invariant nonlinear dynamical system for state $\boldsymbol{s} \in \mathbb{R}^N$,

$$\dot{\boldsymbol{s}}(t) = f(\boldsymbol{s}(t)), \tag{6}$$

where f is a nonlinear continuously differentiable function.

Theorem 3.1 (Sufficient conditions for stability verification of time-invariant nonlinear dynamical systems)**.** *Let $\boldsymbol{s}^*$ be an equilibrium state of the system represented by Equation (6). Define the Jacobian matrix of f evaluated at $\boldsymbol{s}^*$ as $\boldsymbol{J} \in \mathbb{R}^{N\times N}$, and $R_j = \sum_{i=1, i\neq j}^{N} |J_{ji}|, \forall j$. If, for $j \in \{1, \ldots, N\}$, the following two conditions are satisfied:*

1. *$J_{jj} < 0$, and*
2. *$R_j < |J_{jj}|$,*

then $\boldsymbol{s}^*$ is locally asymptotically stable.

Proof. Sufficient conditions for the local asymptotic stability of $\boldsymbol{s}^*$ can be obtained by leveraging Gergschorin's circle theorem and nonlinear control analysis tools on the stability of the linear map on $\boldsymbol{s}^*$. Linearizing the system in Equation (6) around $\boldsymbol{s}^*$, yields $\dot{\boldsymbol{s}} \approx \boldsymbol{J}(\boldsymbol{s} - \boldsymbol{s}^*)$, where $\boldsymbol{J}$ is the Jacobian matrix of $f(\boldsymbol{s}^*(t))$. By the Lyapunov's indirect method [11], to establish the local asymptotic stability of $\boldsymbol{s}^*$, it must be verified that all eigenvalues of $\boldsymbol{J}$ have strictly negative real parts. Calculating the eigenvalues of the $\boldsymbol{J}$ can be very complicated and computationally expensive, especially if the matrix is of high dimension. An alternative, is using the Gershgorin circles theorem [16] to analyze the location of the real parts of the eigenvalues. This theorem guarantees that every eigenvalue of $\boldsymbol{J}$ lies within at least one of the Gershgorin circles, so by computing every Gershgorin circle and verifying the location of the eigenvalues, the conditions of Theorem 3.1 are obtained. □

The sufficient conditions in Theorem 3.1 can be particularized to establish the local asymptotic stability of the equilibrium state reached during inference, $\boldsymbol{s}^0$. This can be done for the inference algorithms of both DEEP and EP models. During inference, the neuronal dynamics of the DEEP model (Equation (3), $\beta = 0$), and of the original and asymmetric EP models (Equation (2), $\beta = 0$), are particular cases of the time-invariant nonlinear dynamical system in Equation (6).

Corollary 3.1.1 (DEEP inference sufficient conditions for stability verification)**.** *Let $\boldsymbol{s}^0$ be an equilibrium state with respect to the dynamics given by Equation (3) with $\beta = 0$. If, for $j \in \{P+1, \ldots, N\}$, the following two conditions are satisfied:*

1. *$\sum\limits_{i=1}^{N} W_{ji} > 0$, and*
2. *$\sum\limits_{i=P+1}^{N} |W_{ij}| < \left|\sum\limits_{i=1}^{N} W_{ji}\right|$,*

then $\boldsymbol{s}^0$ is locally asymptotically stable.

Corollary 3.1.2 (Original and asymmetric EP inference sufficient conditions for stability verification)**.** *Let $\boldsymbol{s}^0$ be an equilibrium state with respect to the dynamics given by Equation (2) with $\beta = 0$. If condition*

$$\max_{j=P+1,\ldots,N} \left\{ \sum_{i=P+1}^{N} |W_{ij}| \right\} < 1$$

is satisfied, then $\boldsymbol{s}^0$ is locally asymptotically stable.

4. Experiments

DEEP is constrained by the curse of dimensionality due to its complex search space, so its performance is analyzed for simple tasks such as learning logical operations. For these tasks, the architecture considered is a 8-neuron complete directed graph (2 input, 5 hidden and 1 output). Although DEEP can learn XOR with 1 hidden neuron and AND and OR with none, 5 hidden neurons are used so that sparsity is perceivable when using the sparsity-inducing method.

Figure 1 illustrates how, when learning XOR, the MSE converges to zero most of the times but not always, and less frequently than when learning AND or OR; thus, highlighting how XOR is harder to learn. Additionally, DEEP is also trained with the sparsity-inducing method introduced before. We observed that, with this method, the most relevant connections are strengthened and, while for simple tasks all the expendable connections are removed, for more complex tasks only few are removed (see Figure 2).

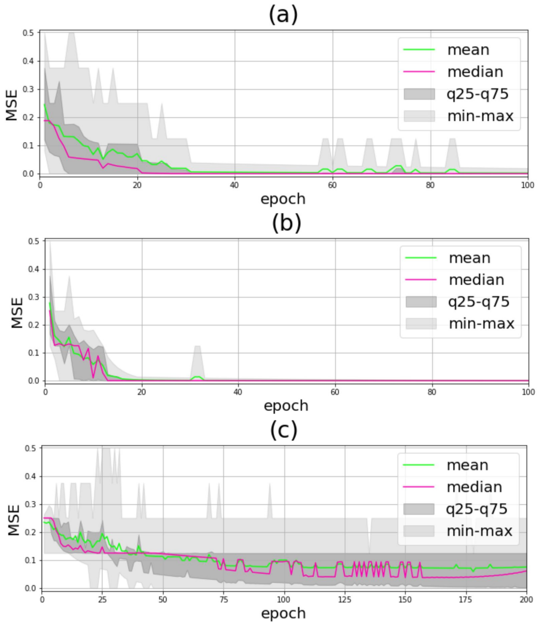

Figure 1: MSE convergence during the first phase of 10 independently trained models when learning the following logical operations: (a) AND, (b) OR, and (c) XOR.

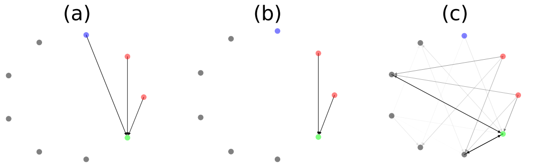

Figure 2: Sparse networks trained for learning the following logical operations: (a) AND (93.75% sparse), (b) OR (95.83% sparse), and (c) XOR (62.50% sparse). The colors of the neurons refer to their type: bias in blue, input in red, hidden in grey, output in green; the connections' opacity is proportional to their strength.

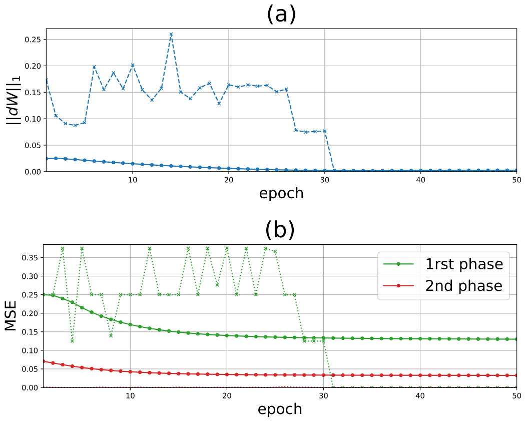

Figure 3: Comparative performance between the asymmetric EP (solid lines) and the DEEP (dotted lines) when learning the logical operation XOR: (a) neuronal dynamics convergence; and (b) MSE convergence.

The performance of DEEP and asymmetric EP [15] is compared with the same architecture (complete directed graph) and hyperparameters. Asymmetric EP fails to learn XOR and, for AND and OR, we observed that the MSE converged slower than DEEP, which took approximately half the number of epochs, and not always to zero (see Figure 3). We identified that the reason for DEEP's worse performance might be due to the fact that its first phase neuronal dynamics does not converge when learning this task. As the "leakage" of the activity of each neuron depends on the sum of outgoing weights, which can be negative, then this "leakage" can become a "source".

The dynamics is driven by the vector fields, which are fully determined by the weights and the neurons' activities. When doing inference (θ^* is fixed), the important factor is the "direction" in which the neurons' activities are being pulled, which is determined by the sign and relative magnitude of the neurons' incoming and outgoing weights. On that note, we performed inference with $\alpha\theta^*$, with $\alpha > 0$ a small constant, and verified that the model is still accurate.

DEEP's training algorithm was also tested with two variations of the learning rule originally introduced (Equation (4)). The first variation of the learning rule adds temporal locality to the already spatially local learning rule [2]. The second variation restricts the proposed learning rule to its sign. In this setting, the learning rule can only transport its sign and not its magnitude. For both variations of the learning rule, we observed that the model was still able to learn AND, OR, and XOR. However, for XOR, the model's convergence was less stable, in the sense that it did not converge to the correct output as often as it did when using the originally proposed learning rule.

Recall that the conditions stated in Corollaries

3.1.1 and 3.1.2 guaranteed the local asymptotic stability of the equilibrium state obtained during inference, s^0, for DEEP and EP, respectively. To verify how accurately these conditions can predict the models' performance, we experimentally tested trained models to see if they satisfied them. We verified that the DEEP models did not satisfy the two conditions in Corollary 3.1.1 and that the asymmetric EP models satisfied the condition in Corollary 3.1.2 only for small-sized networks. Note that this does not contradict the theory established by the corollaries because those conditions are just sufficient. It also does not render the models useless because we experimentally observed their convergence.

5. Conclusions and Future Work

In this work, aiming at more bio-plausible (artificial) neural networks, we extended the *equilibrium propagation* (EP) framework to a more bio-plausible model by generalizing the architecture to a complete directed graph, and introducing: new neuronal dynamics; a modified learning rule; and a sparsity-inducing method – the DirEcted EP (DEEP). Simulation results suggest that DEEP is able to learn logic operations that previous versions are unable. We also verified that the DEEP model could learn when its training algorithms is adapted so that the learning rule is temporally local [2]; and when the learning rule itself is adapted so that it can only transport its sign and not its magnitude. We supported our results with theoretical sufficient conditions to attain local asymptotic stability during inference.

As DEEP is defined by a continuous-time dynamical system, it provides an interesting line of research for algorithms that can be efficiently implemented with neuromorphic hardware [1]. Moreover, due to its unrestricted architecture, DEEP could be used as a network design tool: the optimized, possibly minimalist, structure of the trained networks could be used as an initial architecture for other learning algorithms.

It would also be interesting to study whether DEEP could be adapted to a spiking neural network (as in [12]). Lastly, DEEP has only been tested to learn logic operations, but, due to its recurrent nature, an interesting avenue for future research would be verifying whether DEEP could be used for sequence prediction problems.

References

[1] S. Ambrogio, P. Narayanan, H. Tsai, R. Shelby, I. Boybat, C. Di Nolfo, S. Sidler, M. Giordano, M. Bodini, N. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, and W. Burr. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature*, 558(7708):60–67, 2018.

[2] Anonymous. Equilibrium propagation with continual weight updates. In *ICLR*, 2020.

[3] S. Bartunov, A. Santoro, B. Richards, L. Marris, G. Hinton, and T. Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *NeurIPS*, 2018.

[4] M. Ernoult, J. Grollier, D. Querlioz, Y. Bengio, and B. Scellier. Updates of equilibrium prop match gradients of backprop through time in an RNN with static input. In *NeurIPS*, 2019.

[5] S. Grossberg. Competitive learning: from interactive activation to adaptive resonance. *Cognitive Science*, 11(1):23–63, 1987.

[6] J. Guerguiev, T. Lillicrap, and B. Richards. Towards deep learning with segregated dendrites. *eLife*, 6(e22901), 2017.

[7] H. Hirase, X. Leinekugel, A. Czurkó, J. Csicsvari, and G. Buzsáki. Firing rates of hippocampal neurons are preserved during subsequent sleep episodes and modified by novel awake experience. *PNAS*, 98(16):9386–9390, 2001.

[8] A. Khan. Bidirectional learning in recurrent neural networks using equilibrium propagation (master thesis), 2018.

[9] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[10] Y. LeCun and C. Cortes. MNIST handwritten digit database, 2010.

[11] A. Lyapunov. The general problem of the stability of motion. *Automatica (translated and edited by Francis & Taylor)*, 31(2):353–356, 1992.

[12] P. O'Connor, E. Gavves, and M. Welling. Training a spiking neural network with equilibrium propagation. *JMLR*, 89:1516–1523, 2019.

[13] R. O'Reilly. Six principles for biologically based computational models of cortical cognition. *Trends in Cognitive Sciences*, 2(11):455–462, 1998.

[14] B. Scellier and Y. Bengio. Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, 11(24), 2017.

[15] B. Scellier, A. Goyal, J. Binas, T. Mesnard, and Y. Bengio. Generalization of equilibrium propagation to vector field dynamics. *Arxiv:1808.04873v1*, 2018.

[16] P. Uronen and E. Jutila. Stability via the theorem of Gershgorin. *International Journal of Control*, 16(6):1057–1061, 1972.

[17] X. Xie and H. Seung. Spike-based learning rules and stabilization of persistent neural activity. In *NIPS*, 1999.