Energy-Based Learning Algorithms: A Comparative Study

Anonymous Authors[1]

Abstract

This work compares seven energy-based learning algorithms, namely contrastive learning (CL), equilibrium propagation (EP), coupled learning (CpL) and different variants of these algorithms depending on the type of perturbation used. The algorithms are compared on deep convolutional Hopfield networks (DCHNs) and evaluated on five vision tasks (MNIST, Fashion-MNIST, SVHN, CIFAR-10 and CIFAR-100). The results reveal that while all algorithms perform similarly on the simplest task (MNIST), differences in performance become evident as task complexity increases. Perhaps surprisingly, we find that negative perturbations yield significantly better results than positive ones, and the centered variant of EP emerges as the top-performing algorithm. Lastly, we report new state-of-the-art DCHN simulations on all five datasets (both in terms of speed and accuracy), achieving a 13.5x speedup compared to Laborieux et al. (2021).

1. Introduction

Prior to the dominance of backpropagation-based machine learning, an alternative 'energy-based' learning (EBL) approach was introduced by Hopfield, Hinton and co-others, in which the model dynamics is governed through an energy function (Hopfield, 1984; Hinton et al., 1984). One of the earliest EBL algorithms to train such energy-based models was constrastive learning (CL)(Anderson & Peterson, 1987; Movellan, 1991; Baldi & Pineda, 1991). In the machine learning literature, interest in EBL algorithms has remained limited (LeCun et al., 2006) due to the widespread success of deep neural networks (DNNs) trained by backpropagation on GPUs. However, EBL algorithms have more recently revived interest as a promising learning framework for *analog* learning machines (Kendall et al., 2020; Stern et al., 2021). Small-scale EBL-trained variable resistor networks have been built (Dillavou et al., 2022; 2023), projecting a possible 10,000× improvement in energy consumption compared to GPU-based training of deep neural networks (Yi et al., 2023).

In recent years, several variants of CL have been proposed, such as equilibrium propagation (EP) (Scellier & Bengio, 2017), the centered variant of EP (Laborieux et al., 2021) and coupled learning (CpL) (Stern et al., 2021). These algorithms are often evaluated on different models and different datasets without being compared to CL and to one another. Due to the lack of explicit comparison between these algorithms, they are typically clustered under the 'contrastive learning' banner (Luczak et al., 2022; Stern & Murugan, 2023; Dillavou et al., 2022; Peterson & Lavin, 2022) with many follow-up works indifferently selecting among these algorithms (Ernoult et al., 2019; Dillavou et al., 2022; Watfa et al., 2022; Kiraz et al., 2022; Wycoff et al., 2022; Stern et al., 2022; Yi et al., 2023; Dillavou et al., 2023). One of the main issues for direct comparisons is the slow simulation speed of EBL algorithms. Due to this slowness, EBL algorithms have often been used to train very small networks on very small datasets (by deep learning standards). [1]

In this work, we aim to explicitly compare seven EBL algorithms, including the four above-mentioned and three new ones. Our comparative study is conducted on deep convolutional Hopfield networks (DCHNs), an energy-based model for which simulation times are not prohibitive (Ernoult et al., 2019) and previously demonstrated to scale to more advanced tasks such as CIFAR-10 (Laborieux et al., 2021) and Imagenet 32x32 (Laborieux & Zenke, 2022). Our study consists in training DCHNs with each algorithm on five vision tasks (MNIST, Fashion-MNIST, SVHN, CIFAR-10 and CIFAR-100). We observe notable behavioural differences between algorithms as task difficulty increases. We also introduce a novel energy minimization procedure for DCHNs that significantly accelerates simulation speed (13.5x speedup compared to Laborieux et al. (2021)), yield-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]For example, Kendall et al. (2020) use SPICE to simulate the training of a one-hidden-layer network (with 100 'hidden nodes') on MNIST, which takes one week for only ten epochs of training. Similarly, Stern et al. (2021) simulate the training of disordered networks of up to 2048 nodes on a subset of 200 images of the MNIST dataset.

ing state-of-the-art DCHN simulations both in terms of performance and speed.

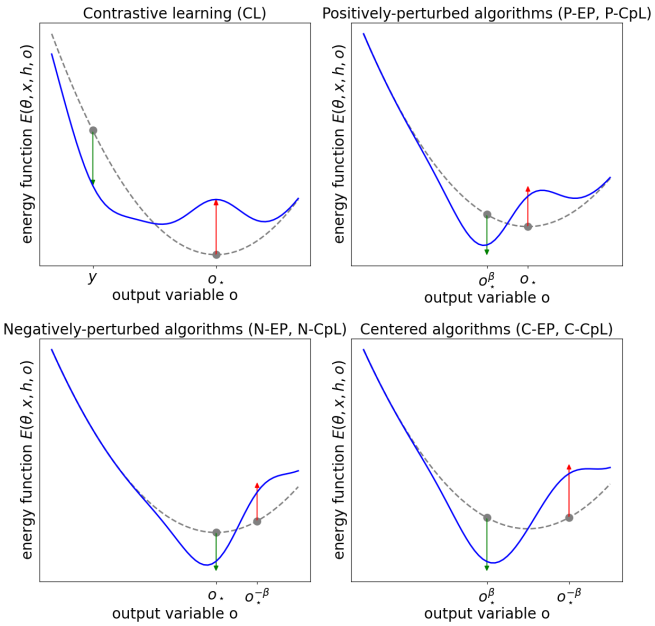

Figure 1. Cartoon illustrating the seven energy-based learning (EBL) algorithms: contrastive learning (CL), positively-perturbed algorithms (P-EP and P-CpL), negatively-perturbed algorithms (N-EP and N-CpL) and centered algorithms (C-EP and C-CpL). The desired output is y. The model prediction is $o_\star$ (i.e. the output configuration minimizing the energy function). The strength of the perturbation is β. A positive perturbation pulls the model output ($o_\star^\beta$) towards y. A negative perturbation pushes the model output ($o_\star^{-\beta}$) away from y. Arrows indicate the weight update: green (resp. red) arrows decrease (resp. increase) the energy value of the corresponding configuration.

2. Energy-based learning algorithms

In the setting of classification, an energy-based model is composed of an input variable (x), a parameter variable (θ) a hidden variable (h) and an output variable (o). A scalar function E called *energy function* assigns to each tuple (θ, x, h, o) a real number $E(\theta, x, h, o)$. Given θ and x, among all possible configurations (h, o), the effective configuration of the model is the equilibrium state, denoted $(h_\star, o_\star)$ and *implicitly* defined as a minimum of the energy function,

$$(h_\star, o_\star) := \underset{(h,o)}{\arg\min}\ E(\theta, x, h, o). \tag{1}$$

The equilibrium value of output variables, $o_\star$, represents a prediction of the model, which we also denote $o(\theta, x) = o_\star$ to emphasize its dependence on the input x and the model parameter θ. The goal of learning is to adjust θ so that, for any input x, the output $o(\theta, x)$ coincides with a desired output y (the label associated to x).

2.1. Contrastive learning

Contrastive learning (CL) is the earliest energy-based learning (EBL) algorithm (Movellan, 1991). The CL algorithm proceeds in two phases. In the first phase, input variables x are clamped, while the hidden and output variables are free to stabilize to their equilibrium value $(h_\star, o_\star)$ as in (1). In the second phase, the output variables o are clamped to the desired output y (i.e., a *strong* perturbation), and the hidden variables h are free to stabilize to a second energy minimum, denoted $h_\star^+$, characterized by

$$h_\star^+ := \underset{h}{\arg\min}\ E(\theta, x, h, y). \tag{2}$$

The contrastive learning rule for the model parameters reads

$$\Delta^{\mathrm{CL}}\theta = \eta \left(\frac{\partial E}{\partial \theta}(\theta, x, h_\star, o_\star) - \frac{\partial E}{\partial \theta}(\theta, x, h_\star^+, y) \right), \tag{3}$$

where η is a learning rate.

2.2. Equilibrium propagation

Equilibrium propagation (EP) is another EBL algorithm (Scellier & Bengio, 2017). One notable difference in EP is that one explicitly introduces a cost function $C(o, y)$ to measure the discrepancy between the output o and desired output y. EP is a variant of CL which also consists of contrasting two states. The first phase of EP is the same as in CL. In the second phase of EP however, the output variables are only slightly perturbed (or *nudged*) rather than clamped to y. This is achieved by augmenting the model's energy by a term $\beta C(o, y)$, where $\beta \in \mathbb{R}$ is a scalar – the *nudging parameter*. The model settles to another equilibrium state, the perturbed state, characterized by

$$(h_\star^\beta, o_\star^\beta) = \underset{(h,o)}{\arg\min}\ \left[E(\theta, x, h, o) + \beta C(o, y)\right]. \tag{4}$$

The learning rule of EP is similar to CL:

$$\Delta^{\mathrm{EP}}\theta = \frac{\eta}{\beta} \left(\frac{\partial E}{\partial \theta}(\theta, x, h_\star, o_\star) - \frac{\partial E}{\partial \theta}(\theta, x, h_\star^\beta, o_\star^\beta) \right). \tag{5}$$

The EP learning rule (5) comes in two variants depending on whether $\beta > 0$ or $\beta < 0$. In this work, we refer to the variant with $\beta > 0$ as *positively-perturbed* EP (P-EP) and the variant with $\beta < 0$ as *negatively-perturbed* EP (N-EP). We also consider the centered variant of EP (C-EP) introduced by Laborieux et al. (2021) whose learning rule reads

$$\Delta\theta = \frac{\eta}{2\beta} \left(\frac{\partial E}{\partial \theta}(\theta, x, h_\star^{-\beta}, o_\star^{-\beta}) - \frac{\partial E}{\partial \theta}(\theta, x, h_\star^\beta, o_\star^\beta) \right). \tag{6}$$

2.3. Coupled learning

Coupled learning (CpL) is another variant of CL, also proceeding in two phases (Stern et al., 2021). The first phase of CpL is identical to the one of CL. In the second phase of CpL, output variables are clamped to a weighted mean of $o_\star$ and y, and the hidden variables are allowed to settle to equilibrium. The weighted mean of outputs, denoted o_C^β, is parameterized by a parameter $\beta \in \mathbb{R} \setminus \{0\}$ as follows:

$$h_C^\beta := \arg\min_h E(\theta, x, h, o_C^\beta), \quad o_C^\beta := (1-\beta)o_\star + \beta y. \tag{7}$$

Similarly to CL and EP, the learning rule of CpL reads

$$\Delta^{\mathrm{CpL}}\theta = \frac{\eta}{\beta}\left(\frac{\partial E}{\partial \theta}(\theta, x, h_\star, o_\star) - \frac{\partial E}{\partial \theta}\left(\theta, x, h_C^\beta, o_C^\beta\right)\right). \tag{8}$$

In particular, for $\beta = 1$, CpL boils down to CL. In their original formulation, Stern et al. (2021) use $\beta > 0$; here we refer to this version as *positively-perturbed* CpL (P-CpL). We also introduce *negatively-perturbed* CpL (N-CpL, with $\beta < 0$) and *centered* CpL (C-CpL):

$$\Delta\theta = \frac{\eta}{2\beta}\left(\frac{\partial E}{\partial \theta}\left(\theta, x, h_C^{-\beta}, o_C^{-\beta}\right) - \frac{\partial E}{\partial \theta}\left(\theta, x, h_C^\beta, o_C^\beta\right)\right). \tag{9}$$

2.4. Theoretical results

Theorem 1 (Contrastive learning)**.** *The contrastive learning rule* (3) *performs one step of gradient descent on the so-called contrastive function J, i.e.* $\Delta^{\mathrm{CL}}\theta = -\eta\frac{\partial J}{\partial \theta}(\theta, x, y)$, *where*

$$J(\theta, x, y) := E\left(\theta, x, h_\star^+, y\right) - E\left(\theta, x, h_\star, o_\star\right). \tag{10}$$

Theorem 1 is proved in Movellan (1991); Baldi & Pineda (1991). However, it is not clear that the contrastive function J has the desirable properties of an objective function from a machine learning perspective. The equilibrium propagation (EP) learning rules have better properties ; the results below are adapted from Scellier et al. (2022).

Theorem 2 (P-EP and N-EP)**.** *There exists a function $\mathcal{L}_\beta$ on which the learning rule* (5) *performs one step of gradient descent, i.e.* $\Delta^{\mathrm{EP}}\theta = -\eta\frac{\partial \mathcal{L}_\beta}{\partial \theta}(\theta, x, y)$. *This function is a lower bound of the 'true' cost function if $\beta > 0$ (P-EP), and an upper bound if $\beta < 0$ (N-EP), i.e.*

$$\mathcal{L}_\beta(\theta, x, y) \leq C\left(o(\theta, x), y\right) \leq \mathcal{L}_{-\beta}(\theta, x, y), \qquad \forall \beta > 0, \tag{11}$$

where $o(\theta, x) := o_\star$ as defined in (1). *Furthermore, $\mathcal{L}_\beta$ approximates the true cost function when $\beta \to 0$ as*

$$\mathcal{L}_\beta(\theta, x, y) = C\left(o(\theta, x), y\right) + O(\beta). \tag{12}$$

Theorem 3 (Centered EP)**.** *There exists a function $\mathcal{L}_{-\beta,+\beta}$ on which the learning rule* (6) *of C-EP performs one step of gradient descent, i.e.* $\Delta^{\mathrm{C-EP}}\theta = -\eta\frac{\partial \mathcal{L}_{-\beta,+\beta}}{\partial \theta}(\theta, x, y)$. *Furthermore, this loss function for C-EP approximates the 'true' cost function up to $O(\beta^2)$, i.e.*

$$\mathcal{L}_{-\beta,+\beta}(\theta, x, y) = C\left(o(\theta, x), y\right) + O(\beta^2). \tag{13}$$

The analysis of coupled learning (P-CpL, N-CpL and C-CpL) is more complicated. The coupled learning rules (8) and (9) provably do not perform gradient descent on the mean squared error (MSE). In fact, there exist situations where the MSE increases under one step of the coupled learning rule, for any value of β and any value of η.

3. Deep convolutional Hopfield networks

To compare the EBL algorithms presented in section 2, we consider the EBL model of Laborieux et al. (2021), which we call here *deep convolutional Hopfield network* (DCHN) model. The network has an input layer, four hidden layers, and an output layer of M units, where M is the number of categories for the classification task considered. Successive layers are interconnected by convolutional interactions with kernel size 3×3, padding 1, and max pooling. Except for the last hidden layer and the output layer, which are interconnected by a dense interaction.

One novelty in our DCHN simulations is that we use an energy minimization procedure based on 'asynchronous updates' so as to compute the steady states (free state and perturbed states) required by the learning algorithms. This is in contrast with the 'synchronous update' procedure used in other works (Ernoult et al., 2019; Laborieux et al., 2021; Laydevant et al., 2021; Laborieux & Zenke, 2022). We find experimentally that our asynchronous procedure requires fewer steps to converge than the synchronous one.

4. Simulations

4.1. Comparative study of EBL algorithms

We compare with simulations the seven EBL algorithms of section 2 on the deep convolutional Hopfield network (DCHN) of section 3. To do this, we train a DCHN on MNIST, Fashion-MNIST, SVHN, CIFAR-10 and CIFAR-100 using each of the seven EBL algorithms. For each simulation, the DCHN is trained for 100 epochs. Each run is performed on a single A100 GPU. A run on MNIST and FashionMNIST takes 3 hours 30 minutes ; a run on SVHN takes 4 hours 45 minutes ; and a run on CIFAR-10 and CIFAR-100 takes 3 hours. All these simulations are performed with the same network using the same initialization scheme and the same hyperparameters.

We draw several lessons from Table 1.

Table 1. Results obtained by training the deep convolutional Hopfield network (DCHN) of Section 3 with the seven EBL algorithms of Section 2. We also report two baselines: truncated backpropagation (TBP) and recurrent backpropagation (RBP) (Almeida, 1987; Pineda, 1987). For each of these 45 experiments, we perform three runs and report the mean test error rate in %.

	MNIST	F-MNIST	SVHN	C-10	C-100
TBP	0.42	6.12	3.76	10.1	33.4
RBP	0.44	6.28	3.87	10.7	34.4
CL	0.61	10.10	6.1	31.4	71.4
P-EP	1.66	90.00	83.9	72.6	89.4
N-EP	**0.42**	**6.22**	80.4	11.9	44.7
C-EP	0.44	6.47	**3.51**	**11.1**	**37.0**
P-CpL	0.66	64.70	40.1	46.9	77.9
N-CpL	0.50	6.86	80.4	13.5	51.9
C-CpL	0.44	6.91	4.23	14.9	46.5

Algorithms perform alike on MNIST. Little difference is observed in the test performance of the algorithms on MNIST, ranging from 0.42% to 0.66% test error rate for six of the seven EBL algorithms.

Negative perturbations yield better results than positive ones. On Fashion-MNIST, CIFAR-10 and CIFAR-100, the N-EP and N-CpL algorithms (employing negative perturbations) perform significantly better than CL, P-EP and P-CpL (employing positive perturbations). This can be partly explained by Theorem 2, which shows that N-EP optimizes an upper bound of the cost function, whereas P-EP optimizes a lower bound. [2]

Two-sided perturbations yield better results than one-sided perturbations. While there is little difference between centered algorithms (C-EP and C-CpL) and negatively-perturbed algorithms (N-EP and N-CpL) on MNIST, FashionMNIST and CIFAR-10, the centered algorithms significantly improve the test error rate on CIFAR-100 and unlock training on SVHN. This can be partly explained by Theorems 2 and 3, which show that the loss function of C-EP better approximates the cost function (up to $O(\beta^2)$) than the loss function $\mathcal{L}_{-\beta}$ of N-EP (up to $O(\beta)$).

The EP perturbation method yields better results than the one of CpL. C-EP outperforms C-CpL on all tasks, sometimes by a significant margin (CIFAR-10 and CIFAR-100). Similarly, N-EP outperforms N-CpL on all tasks. [3]

Weak positive perturbations perform worse than strong ones. P-EP fails on most datasets (Fashion-MNIST, SVHN, CIFAR-10 and CIFAR-100). P-CpL performs better than P-EP on all tasks, but also yields poor results in general. It is noteworthy that CL, which employs a strong positive perturbation, performs better than P-CpL and P-EP (which employ weak positive perturbations) on *all tasks*, sometimes by a very large margin (Fashion-MNIST and SVHN).

4.2. State-of-the-art DCHN simulations (performance and speed)

The comparative study conducted in the previous subsection highlights C-EP as the best EBL algorithm among those considered in this work. Using C-EP, we then perform additional simulations on MNIST, CIFAR-10 and CIFAR-100, where we optimize the hyperparameters of training (weight initialization, initial learning rates, number of iterations, value of the nudging parameter and weight decay) to yield the best performance. We report the results in Table 2.

Table 2. We achieve state-of-the-art results with C-EP-trained DCHNs on MNIST, CIFAR-10 and CIFAR-100. For each dataset, we report the test error rate averaged over 3 runs, after 100 epochs of training, and after 300 epochs, and we compare with the existing literature on deep convolutional Hopfield networks (DCHNs)

	MNIST	C-10	C-100
Ernoult et al. (2019)	1.02		
Laborieux et al. (2021)		11.68	
Laydevant et al. (2021)	0.85	13.78	
Luczak et al. (2022)		20.03	
Laborieux & Zenke (2022)		11.4	38.4
This work (100 epochs)	**0.44**	10.40	34.2
This work (300 epochs)		**9.70**	**31.6**

We achieve better simulation results than existing works on DCHNs on all three datasets, both in terms of performance and speed. For instance, our 100-epoch simulations on CIFAR-10 take 3 hours 18 minutes, which is 7 times faster than those reported in (Laborieux & Zenke, 2022) (1 day), and 36 times faster than those reported in Laydevant et al. (2021) (5 days), and our 300-epoch simulations on CIFAR-10 yield 9.70% test error rate, which is significantly lower than (Laborieux & Zenke, 2022) (11.4%). Since no earlier work on DCHNs has performed simulations on Fashion-MNIST and SVHN, our results reported in Table 1 are state-of-the-art on these datasets as well.

Our important speedup comes from our novel energy-

[2] We note that on SVHN, the results obtained with N-EP and N-CpL are much worse than CL. However, further simulations with a different weight initialization scheme show that N-EP and N-CpL generally perform better than CL, P-EP and P-CpL, supporting our conclusion.

[3] We note however that P-EP performs worse than P-CpL on all tasks.

minimization procedure based on "asynchronous updates", combined with 60 iterations at inference (free phase) and the use of 16-bit precision ; in comparison, Laborieux et al. (2021) used "synchronous updates" with 250 iterations and 32-bit precision. These changes result in a 13.5x speedup on the same device (a A100 GPU) without degrading the performance.

5. Conclusion

Our comparative study of energy-based learning (EBL) algorithms delivers a few key take-aways: 1) while all EBL algorithms work equally well on MNIST, more difficult tasks magnify small algorithmic differences, 2) negative perturbations yield better results than positive ones, 3) two-sided (centered) perturbations perform better than one-sided perturbations, and 4) the perturbation technique of equilibrium propagation yields better results than the one of coupled learning. Our results highlight the centered variant of equilibrium propagation (C-EP) as the best EBL algorithm among those considered in the present work.

Our work also establishes new state-of-the-art results for deep convolutional Hopfield networks (DCHNs) on all five datasets, both in terms of performance (accuracy) and speed. In particular, thanks to the use of a novel "asynchronous" energy-minimization procedure for DCHNs, we manage to reduce the number of iterations required to converge to equilibrium from 250 to 60. Combined with the use of 16-bit precision (instead of 32-bit), this leads our simulations to be 13.5 times faster than those of Laborieux et al. (2021) when run on the same hardware. We hope that our important simulation speedup will help foster more interest and research in EBL algorithms.

Ultimately, while we have conducted our comparative study on *simulations* of Hopfield networks, the potential of EBL algorithms is for the training of *analog* hardware. We believe that our findings can inform the design of analog learning machines (Dillavou et al., 2022; Yi et al., 2023; Dillavou et al., 2023), and simulations of these machines (Kendall et al., 2020; Stern et al., 2021). Finally, our theoretical insights (Section 2.4) also apply to novel EBL algorithms (Scellier et al., 2022; Anisetti et al., 2022; Laborieux & Zenke, 2022; Hexner, 2023), and more generally to bi-level optimization problems (Zucchet & Sacramento, 2022), including the training of Lagrangian systems (Kendall, 2021; Scellier, 2021), meta-learning (Zucchet et al., 2022), as well as predictive coding (Millidge et al., 2022). We think that our insights and experimental findings can also guide the development of better learning algorithms in these settings as well.

References

Almeida, L. B. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In *Proceedings, 1st First International Conference on Neural Networks*, volume 2, pp. 609–618. IEEE, 1987.

Anderson, J. R. and Peterson, C. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1: 995–1019, 1987.

Anisetti, V. R., Kandala, A., Scellier, B., and Schwarz, J. Frequency propagation: Multi-mechanism learning in nonlinear physical networks. *arXiv preprint arXiv:2208.08862*, 2022.

Baldi, P. and Pineda, F. Contrastive learning and neural oscillations. *Neural Computation*, 3(4):526–545, 1991.

Dillavou, S., Stern, M., Liu, A. J., and Durian, D. J. Demonstration of decentralized physics-driven learning. *Physical Review Applied*, 18(1):014040, 2022.

Dillavou, S., Beyer, B., Stern, M., Miskin, M. Z., Liu, A. J., and Durian, D. J. Circuits that train themselves: decentralized, physics-driven learning. In *AI and Optical Data Sciences IV*, volume 12438, pp. 115–117. SPIE, 2023.

Ernoult, M., Grollier, J., Querlioz, D., Bengio, Y., and Scellier, B. Updates of equilibrium prop match gradients of backprop through time in an rnn with static input. In *Advances in Neural Information Processing Systems*, pp. 7079–7089, 2019.

Hexner, D. Training precise stress patterns. *Soft Matter*, 19 (11):2120–2126, 2023.

Hinton, G. E., Sejnowski, T. J., and Ackley, D. H. *Boltzmann machines: Constraint satisfaction networks that learn*. Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA, 1984.

Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, 81(10): 3088–3092, 1984.

Kendall, J. A gradient estimator for time-varying electrical networks with non-linear dissipation. *arXiv preprint arXiv:2103.05636*, 2021.

Kendall, J., Pantone, R., Manickavasagam, K., Bengio, Y., and Scellier, B. Training end-to-end analog neural networks with equilibrium propagation. *arXiv preprint arXiv:2006.01981*, 2020.

Kiraz, F. Z., Pham, D.-K. G., and Desgreys, P. Impacts of Feedback Current Value and Learning Rate on Equilibrium Propagation Performance. In *2022 20th IEEE*

Interregional NEWCAS Conference (NEWCAS), pp. 519–523, Quebec City, Canada, June 2022. IEEE. doi: 10.1109/NEWCAS52662.2022.9842178. URL https://hal.telecom-paris.fr/hal-03779416.

Laborieux, A. and Zenke, F. Holomorphic equilibrium propagation computes exact gradients through finite size oscillations. *arXiv preprint arXiv:2209.00530*, 2022.

Laborieux, A., Ernoult, M., Scellier, B., Bengio, Y., Grollier, J., and Querlioz, D. Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias. *Frontiers in neuroscience*, 15:129, 2021.

Laydevant, J., Ernoult, M., Querlioz, D., and Grollier, J. Training dynamical binary neural networks with equilibrium propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4640–4649, 2021.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Luczak, A., McNaughton, B. L., and Kubo, Y. Neurons learn by predicting future activity. *Nature Machine Intelligence*, 4(1):62–72, 2022.

Millidge, B., Song, Y., Salvatori, T., Lukasiewicz, T., and Bogacz, R. Backpropagation at the infinitesimal inference limit of energy-based models: Unifying predictive coding, equilibrium propagation, and contrastive hebbian learning. *arXiv preprint arXiv:2206.02629*, 2022.

Movellan, J. R. Contrastive hebbian learning in the continuous hopfield model. In *Connectionist Models*, pp. 10–17. Elsevier, 1991.

Peterson, E. and Lavin, A. Physical computing for materials acceleration platforms. *arXiv preprint arXiv:2208.08566*, 2022.

Pineda, F. J. Generalization of back-propagation to recurrent neural networks. *Physical review letters*, 59(19):2229, 1987.

Scellier, B. *A deep learning theory for neural networks grounded in physics*. PhD thesis, Université de Montréal, 2021. URL https://papyrus.bib.umontreal.ca/xmlui/handle/1866/25593.

Scellier, B. and Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Scellier, B., Mishra, S., Bengio, Y., and Ollivier, Y. Agnostic physics-driven deep learning. *arXiv preprint arXiv:2205.15021*, 2022.

Stern, M. and Murugan, A. Learning without neurons in physical systems. *Annual Review of Condensed Matter Physics*, 14:417–441, 2023.

Stern, M., Hexner, D., Rocks, J. W., and Liu, A. J. Supervised learning in physical networks: From machine learning to learning machines. *Physical Review X*, 11(2): 021045, 2021.

Stern, M., Dillavou, S., Miskin, M. Z., Durian, D. J., and Liu, A. J. Physical learning beyond the quasistatic limit. *Physical Review Research*, 4(2):L022037, 2022.

Watfa, M., Garcia-Ortiz, A., and Sassatelli, G. Energy-based analog neural network framework. In *2022 IEEE 35th International System-on-Chip Conference (SOCC)*, pp. 1–6. IEEE, 2022.

Wycoff, J., Dillavou, S., Stern, M., Liu, A., and Durian, D. Desynchronous learning in a physics-driven learning network. *The Journal of Chemical Physics*, 156(14): 144903, 2022.

Yi, S.-i., Kendall, J. D., Williams, R. S., and Kumar, S. Activity-difference training of deep neural networks using memristor crossbars. *Nature Electronics*, 6(1):45–51, 2023.

Zucchet, N. and Sacramento, J. Beyond backpropagation: bilevel optimization through implicit differentiation and equilibrium propagation. *Neural Computation*, 34(12): 2309–2346, 2022.

Zucchet, N., Schug, S., Von Oswald, J., Zhao, D., and Sacramento, J. A contrastive rule for meta-learning. *Advances in Neural Information Processing Systems*, 35: 25921–25936, 2022.