Learning a Hopfield Network in Latent Space with Equilibrium Propagation

Zhancun Mu
YuanPei College
Peking University
yhbylch@stu.pku.edu.cn

Abstract

In vision systems, the top-down process is a crucial part of the visual perception process [5]. Assuming that the observed data is generated by a low-dimensional latent variable by this top-down process, generative models are proposed to learn the distribution of natural images. Pang et al. [11] assume the latent vector follows an energy-based model (EBM) and leverage short-run Markov chain Monte Carlo (MCMC) to jointly learn the EBM and the top-down network. In this paper, we use Hopfield network as an EBM to model the prior probability of latent vector and train the model with equilibrium propagation (EP). We show that the EP algorithm, short-run MCMC and Constrastive Hebbian Learning (CHL) share strong similarities. The Hopfield network and the algorithm are more biologically plausible. And we believe this method opens more opportunities for usage of EP in generative models.

1 Introduction

Generative modeling aims to capture the underlying data distribution and generate new samples. Variational autoencoders (VAEs) [10] and generative adversarial networks (GANs) [4] have shown impressive results on images, texts and other modalities. Most existing models use feedforward neural networks and backpropagation for training. However, such architectures lack certain properties of biological neural networks and may be inefficient to implement in neuromorphic hardware [15].

As an alternative, energy-based models such as Gibbs models and Hopfield networks have gained interest [2, 11, 16]. Pang et al. [11] assume the prior distribution of the latent vector follows a Gibbs distribution and use short-run MCMC to jointly learn the Gibbs distribution and the top-down network. At the same time, as an equivalent of Gibbs distribution, Hopfield network is a more biologically plausible alternative [7, 8]. The recurrent connections and settling dynamics make it better suited for hardware implementations. Equilibrium propagation (EP) was introduced as an efficient and local learning rule for training Hopfield networks. While EP has shown promise for discriminative tasks, its application to generative modeling remains limited.

In this work, we propose learning a Hopfield network in latent space to model prior distribution of the latent vector using EP. The Hopfield network serves as an expressive prior on the latent variables. We show that short-run MCMC sampling for inference and learning shares similarities with EP and CHL. Avoiding the use of MCMC, our method is more biologically plausible, and opens opportunities for efficient neuromorphic implementations.

2 Preliminaries and Related Work

2.1 Continuous Hopfield Network

The Continuous Hopfield Network (CHN) is a recurrent neural network model whose dynamics are governed by an energy function. It was first introduced by [8].

Here we take the notation in [9].A CHN consists of input neurons $\mathbf{x}$, hidden neurons $\mathbf{h}$, and output neurons $\mathbf{y}$, interconnected through symmetric weights $\mathbf{W}$. The energy function is defined as:

$$E(\mathbf{s}) = \frac{1}{2}|\mathbf{s}|^2 - \frac{1}{2}\rho(\mathbf{s}^\top)\mathbf{W}\rho(\mathbf{s}) \tag{1}$$

where $\mathbf{s}$ is the state vector of all neurons, $\rho(\cdot)$ is a bounded nonlinear activation function, and $\mathbf{W}$ is the symmetric weight matrix.

The network dynamics follow gradient descent on the energy function:

$$\frac{d\mathbf{s}}{dt} = -\nabla E(\mathbf{s}) = -\mathbf{s} + \rho'(\mathbf{s}) \odot \mathbf{W}\rho(\mathbf{s}) \tag{2}$$

where $\odot$ is element-wise multiplication and ρ' is the derivative of $\rho(\cdot)$. This differential equation is typically integrated numerically using the update rule:

$$\mathbf{s}_{i+1} = (1 - \delta t)\mathbf{s}_i + \delta t \rho'(\mathbf{s}_i) \odot \mathbf{W}\rho(\mathbf{s}_i) \tag{3}$$

where δt is the step size. Starting from an initial state, this iterative update settles the network into an equilibrium energy minimum.

2.2 Equilibrium Propagation

Equilibrium Propagation (EP) is an alternative training method for energy-based recurrent neural networks like Continuous Hopfield Networks. It was proposed by Scellier and Bengio [14] as a more biologically plausible learning rule than backpropagation through time.

For supervised learning, EP first lets the network settle into a free equilibrium given input $\mathbf{x}$. This is the *free phase* $s_0^{\mathbf{x}}$. Then an extra term representing the loss $L(\mathbf{y}, \mathbf{y}_{target})$ between output $\mathbf{y}$ and target $\mathbf{y}_{target}$ is added to the energy function:

$$F = E + \beta L(\mathbf{y}, \mathbf{y}_{target}) \tag{4}$$

where E is the original energy, β is a small constant, and F is the *total energy*. Using F in place of E in the update rule nudges $\mathbf{y}$ towards $\mathbf{y}_{target}$. The equilibrium energy minimum of F is called the *weakly clamped equilibrium* $\mathbf{s}_\beta^{\mathbf{x}}$.

Each weight w is then updated based on the difference between partial derivatives of F at the two equilibria:

$$\Delta w \propto \frac{1}{\beta}\left(\frac{\partial F}{\partial w}\bigg|_{\mathbf{s}_\beta^{\mathbf{x}}} - \frac{\partial F}{\partial w}\bigg|_{\mathbf{s}_0^{\mathbf{x}}}\right) \tag{5}$$

This update approximates the gradient from backpropagation through time. EP has computational advantages as a local learning rule requiring only activations and symmetric weight updates.

By alternating between free and weakly clamped phases, EP can train CHNs to perform supervised learning tasks. Extensions to EP also allow optimizing hyperparameters like the learning rate.

2.3 Generative Models and Energy-Based Models

Generative models have been widely studied in artificial intelligence. The main assumption is that the observed data $\mathbf{I}$ is generated by a latent variable $\mathbf{z}$, i.e.

$$\mathbf{I} \sim G(\mathbf{z}) + \epsilon, \quad \epsilon \sim \mathcal{N}(\mu, \sigma^2) \tag{6}$$

In the training process of generative models, the main challenge is to tackle the intractable posterior distribution $p(\mathbf{z}|\mathbf{I})$. One approach is to run MCMC to sample from the posterior distribution. For example, in [6], they assume $\mathbf{z}$ comes from a Guassian distribution and use Langevin dynamics to sample from the posterior distribution. In [11], they assume $\mathbf{z}$ follows a Gibbs distribution and jointly optimize the Gibbs distribution and the top-down network after sampling from the Gibbs distribution. Another approach is to train a model to estimate the posterior distribution. For example, Kingma et al. [10] use a variational autoencoder (VAE) to estimate the posterior distribution and maximize the evidence lower bound (ELBO) to train the models. In [15], they replace VAE with a CHN and use EP to train the model, while our method aims to replace the MCMC sampling with Hopfield dynamics and learn a prior distribution of the latent vector. However, the usage of EP and CHN in the context of generative models is still limited.

3 Training a Hopfield Network with Equilibrium Propagation

Suppose n training samples $x_1, x_2, \ldots, x_n$ are observed. The optimization objective is to maximize the log-likelihood of the training data:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \log p(x_i; \theta) \tag{7}$$

Assuming the data is generated by latent factors z, the gradient of the parameters θ can be obtained according to:

$$\begin{aligned}
\nabla_\theta p(x;\theta) &= \frac{1}{p(x;\theta)} \frac{\partial}{\partial \theta} \int p(x,z;\theta) \mathrm{d}z \\
&= \frac{1}{p(x;\theta)} \int \left[\frac{\partial}{\partial \theta} \log p(x,z;\theta) \right] p(x,z;\theta) \mathrm{d}z \\
&= \int \left[\frac{\partial}{\partial \theta} \log p(x,z;\theta) \right] \frac{p(x,z;\theta)}{p(x;\theta)} \mathrm{d}z \\
&= \mathrm{E}_{p(z|x;\theta)} \left[\frac{\partial}{\partial \theta} p(x,z;\theta) \right] \\
&= \mathrm{E}_{p(z|x;\theta)} \left[\nabla_\theta (\log p_\alpha(z) + \log p_\beta(x|z)) \right]
\end{aligned} \tag{8}$$

Here $\theta = \{\alpha, \beta\}$ where α and β are the parameters of the prior distribution $p_\alpha(z)$ and the conditional distribution $p_\beta(x|z)$ respectively. A detailed derivation can be found in [11].

Therefore, for the prior model, the gradient is $\nabla_\alpha \mathrm{E}_{p(z|x;\theta)} \left[\log p_\alpha(z) \right]$ and for the generation model, the gradient is $\nabla_\beta \mathrm{E}_{p(z|x;\theta)} \left[\log p_\beta(x|z) \right]$. If the prior model takes the form of a Gibbs distribution:

$$p_\alpha(z) = \frac{1}{Z_\alpha} \exp\left(f_\alpha(z) \right) p_0(z), \tag{9}$$

the gradient of the prior model can be written as:

$$\nabla_\alpha \log p_\theta(x) = \mathrm{E}_{p(z|x;\theta)} \left[\nabla_\alpha f_\alpha(z) \right] - \mathrm{E}_{p_\alpha(z)} \left[\nabla_\alpha f_\alpha(z) \right] \tag{10}$$

Intuitively, α is updated to match the inferred distribution $p(z|x;\theta)$ with the current prior distribution $p_\alpha(z)$.

However, the inference of $p(z|x;\theta)$ is intractable. In [11], they use short-run MCMC to sample z from the posterior distribution. Inspired by the relation between Gibbs distribution and Hopfield network, MCMC and attractor dynamics [3, 7, 12], Eq. (10) and EP, we use Hopfield dynamics for sampling latent variable and Equilibrium Propagation for training the model.

Free phase: For the free phase, the Hopfield network is initialized randomly and the dynamics are run for a few steps to settle into an equilibrium. This is equivalent to the second term in Eq. (10).

Weakly clamped phase: For the weakly clamped phase, the Hopfield network dynamics is guided by the posterior distribution $p(z|x;\theta)$. The posterior density $p(z|x;\theta) = p(z,x;\theta)/p(x;\theta)$ is proportional to the joint density $p(z,x;\theta)$. Thus to generate z to match the posterior distribution, we can change the energy function in Eq. (4) to:

$$F = E + \beta \log p(z,x;\theta) \tag{11}$$

The dynamics according to Eq. (11) is highly similar to the Langevin inference dynamics in [11], i.e.

$$z_{\tau+1} = z_\tau + \frac{\delta^2}{2} \frac{\partial}{\partial z} \log p(z_\tau, x; \theta) + \delta U_\tau, U_\tau \sim N(0, I_d) \tag{12}$$

After reaching the equilibrium, the latent variable z can be used to update the Hopfield network according to Eq. (5) and the generation model according to Eq. (5). Therefore a unified framework for MCMC, EP, Hopfield network and generative models is established, which can be summarized in Tab. 1. A detailed linkage between EP and CHL can be seen in [14].

The algorithm is summarized in Alg. 1.

	Equilibrium Propagation	MCMC-based Generative Model	Constrastive Hebbian Learning	Our Framework
First Phase	Free Phase	Sample from Current Prior	Negative Phase	Free Phase
Second Phase	Weakly Clamped Phase	Sample from Posterior	Positive Phase	Weakly Clamped Phase

Table 1: Comparison between Equilibrium Propagation, MCMC-based Generative Model, Constrastive Hebbian Learning and Our Framework. The linkage between EP and CHL can be seen in [14].

Algorithm 1: Learning a Hopfield Network Prior in Latent Space with Equilibrium Propagation

input : Learning iterations T, initial parameters $\theta_0 = \{\alpha_0, \beta_0\}$, observed data $\{x_1, x_2, \ldots, x_n\}$, batch size B, learning rate for Hopfield network η_0, learning rate for generation model η_1, Hopfield dynamics $\{dt, \tau, n_{\text{HP}}, \xi\}$ where dt is the step size, τ is a time constant, n_{HP} is the number of Hopfield dynamics iterations and ξ is threshold to justify convergence.

output : $\theta_T = \{\alpha_T, \beta_T\}$.

for $i \leftarrow 0$ **to** $T - 1$ **do**

- **Mini-batch:** Sample a mini-batch of B samples from the training data $\{x_1, x_2, \ldots, x_n\}$;
- **Free phase:** Initialize the Hopfield network randomly and run the dynamics to settle into an equilibrium s_0^x;
- **Weakly clamped phase:** Update the energy function according to Eq. (11) and run the dynamics to settle into an equilibrium $s_{\eta_0}^x$;
- **Hopfield network update:** Update the Hopfield network parameters α according to Eq. (5);
- **Generation model update:** Update the generation model parameters β according to Eq. (8);

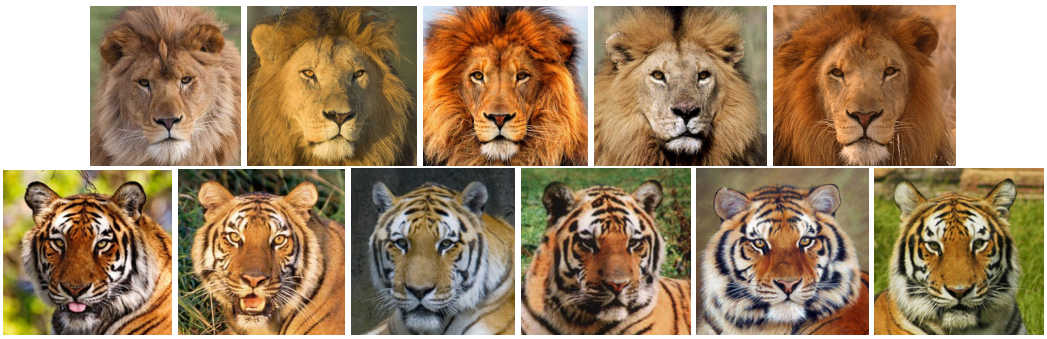


Figure 1: Lion-tiger images. Image credit: [6].

4 Experiments

Here we provide qualitive results of our method on a relatively small dataset. The dataset contains 11 images of 6 tigers and 5 lions, as used in [6] and can be found in Fig. 1.

As proved in [1], the capacity of a Hopfield network scales as $\frac{D}{4 \log_2 D}$ where D is number of neurons. Thus if there are 11 types of patterns, the number of neurons should be roughly larger than 400. Though it seems to be big, it is still much smaller than the dimension of the input image, which is $3 \times 128 \times 128 = 49152$. We learn a Hopfield network with 512 neurons and use a top-down ConvNet as the generation model. The results are shown in Fig. 2. The left column is the reconstructed images and the right column is systhesized by feeding a Guassian noise to the Hopfield network and using the equilibrium state for generation. The right column can be seen as visualization of the attractor states of the Hopfield network, projecting the latent space to the image space.

We examine the Hopfield network dynamics in the weakly clamped phase. With a well-trained Hopfield network, we expect the state of the Hopfield network to converge to the posterior distribution $p(z|x; \theta)$, which means the systhesized images should be more and more similar to the input image.

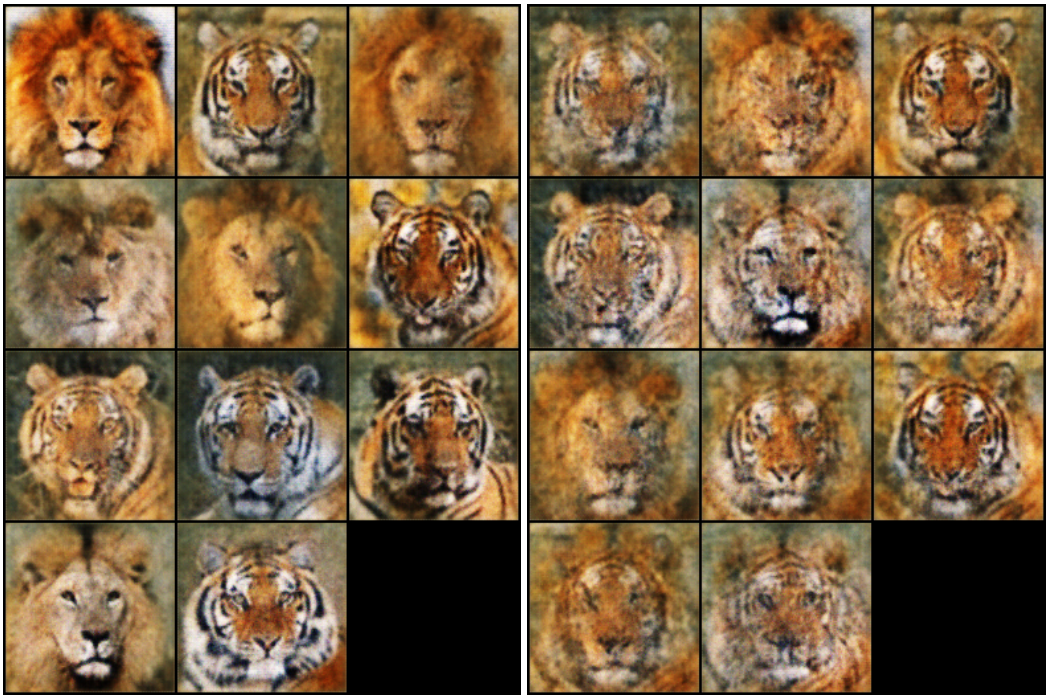


Figure 2: Reconstructed and sampled images after 1800 epochs.

Fig. 3 depicts the convergence process of the Hopfield network dynamics with 100 steps. We can see that the systhesized images converge fast from tiger to lion.

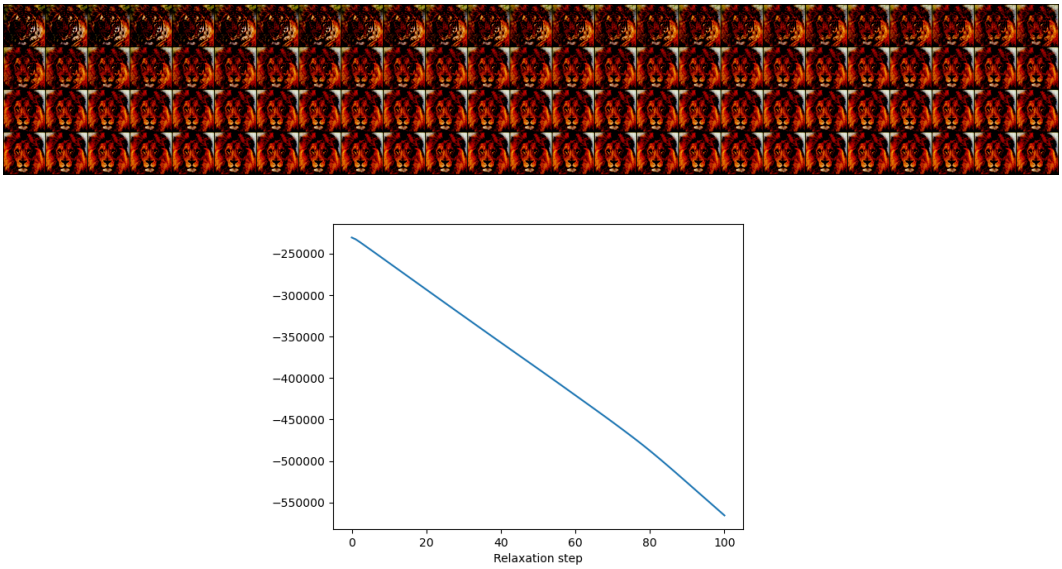


Figure 3: Hopfield network dynamics with 100 steps. The top row is the systhesized images and the bottom row is the energy of the Hopfield network.

5 Conclusion and Future Work

In this paper, we propose a method to learn a Hopfield network in latent space with equilibrium propagation. To the best of our knowledge, this is the first time that EP and CHN is used for prior modeling in generative models. We show an elegant and strong connection between EP, CHL, MCMC and Hopfield dynamics, demonstrating the potential of EP and CHN in generative models. The biologically plausible properties of EP and CHN make them more suitable for neuromorphic hardware implementations.

However, due to the limitation of computation resources, we only test our method on a relatively small dataset and the Hopfield network we designed is relatively small comparing to [15], leading to a small capacity of equilibrium states. Future work includes testing our method on larger datasets and designing a larger Hopfield network or using modern Hopfield networks such as [13]. Also, one shortcome of our method is that the generated images are not sharp enough, this can be improved by converting our method to more advanced generative models such as GANs [4].

References

[1] Yaser Abu-Mostafa and John S Jacques. Information capacity of the hopfield model. *IEEE Transactions on Information Theory*, 31(4):461–464, 1985. 4

[2] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985. 1

[3] DJ Amit and Modelling Brain Function. The world of attractor neural networks. *Modeling Brain Function*, 1989. 3

[4] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 1, 6

[5] Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350–363, 2013. 1

[6] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 2, 4

[7] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. 1, 3

[8] John J Hopfield. 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. 1

[9] Anthony Hu. Neural world models for computer vision, 2023. 2

[10] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. 1, 2

[11] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33: 21994–22008, 2020. 1, 2, 3

[12] Bruno Poucet and Etienne Save. Attractors in memory. *Science*, 308(5723):799–800, 2005. 3

[13] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020. 6

[14] Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017. 2, 3, 4

[15] Tom Van Der Meersch, Johannes Deleu, and Thomas Demeester. Training a hopfield variational autoencoder with equilibrium propagation. *arXiv preprint arXiv:2311.15047*, 2023. 1, 2, 6

[16] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27:107–126, 1998. 1