From Optimization to Learning in Physical Ising Machines

Jianan Wu

03/06/2026

Deep Neural Networks

- Inference:

$$s_l = f_l\left(f_{l-1}\left(\cdots f_1(x)\right)\right)$$

- Training:

$$\theta \leftarrow \theta - \eta \nabla_\theta L(\theta)$$

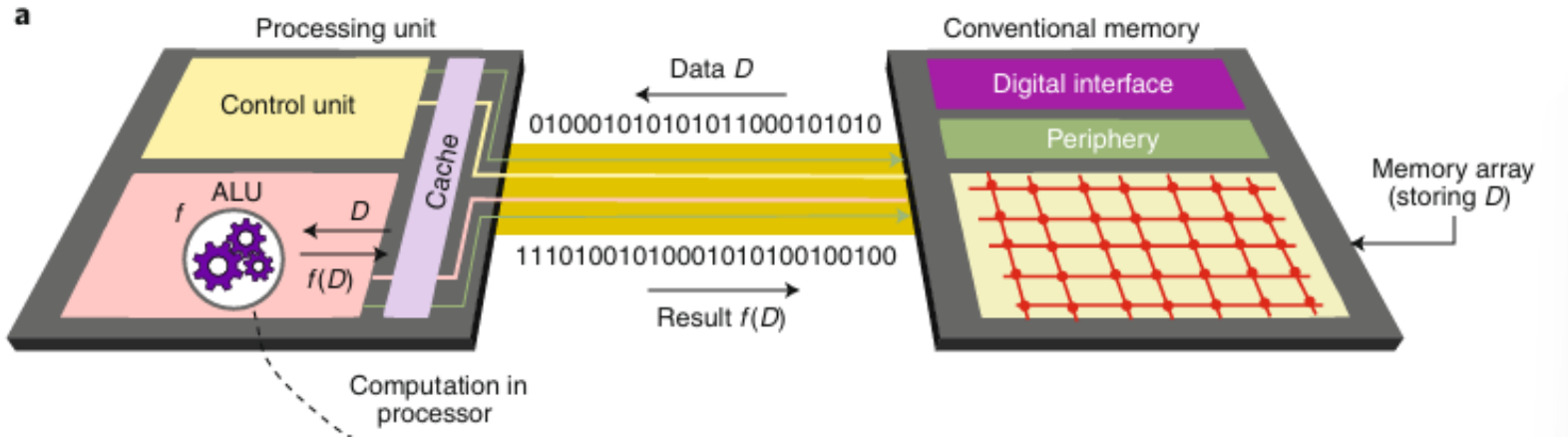
$$\nabla_\theta L = \frac{\partial L}{\partial s_l}\frac{\partial s_l}{\partial s_{l-1}}\cdots\frac{\partial s_1}{\partial \theta}$$

Computational implications

- Full forward propagation
- Full backward pass to propagate gradients
- All intermediate weights and activations must be stored

The Memory Wall

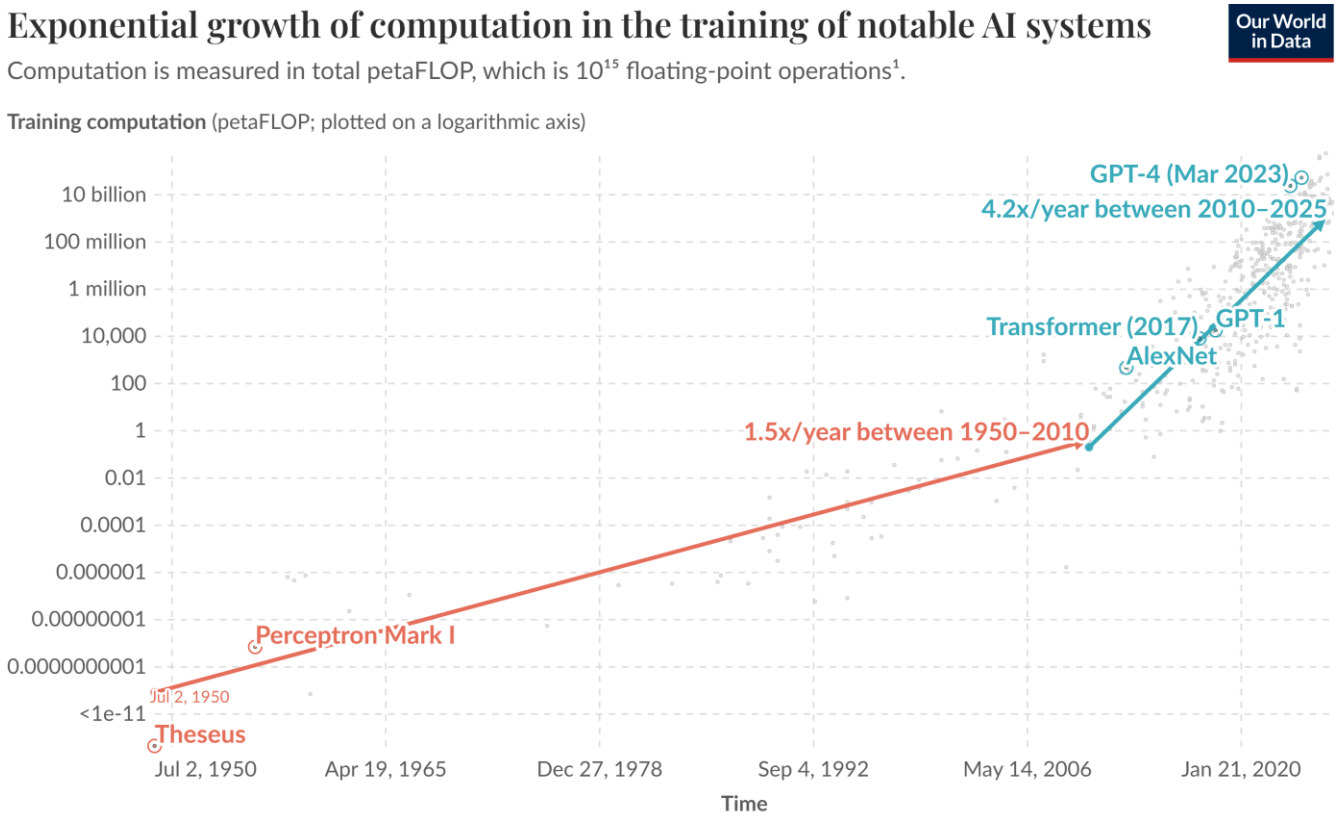
- In conventional computer architectures, the processor and memory are separate.
- During neural network computation, weights and activations must be moved back and forth between them.
- This repeated data movement creates a speed bottleneck and consumes a large amount of energy.



[2] Nat. Nanotechnol (Sebastian, 2020).

AI Scaling Crisis

- Modern AI progress is tightly coupled to compute scaling.
- Training compute has grown exponentially.
- Question: Is compute scaling sustainable in energy, cost and hardware efficiency?



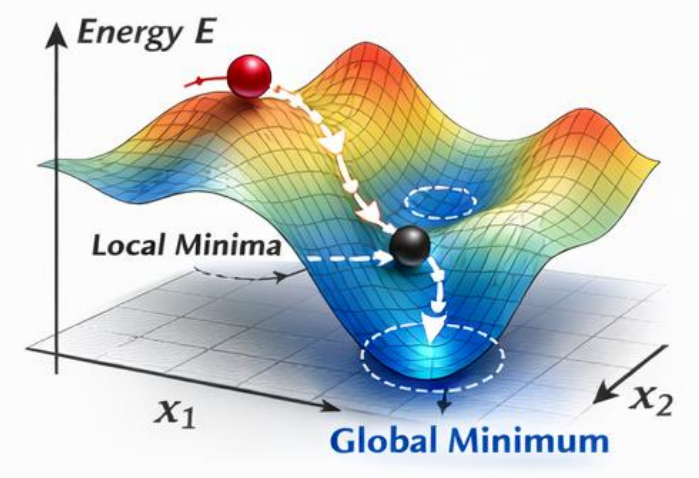
[1] Our World in Data (Samborska, 2025).

Can physics perform calculation directly?

Physics Performs Optimization Naturally

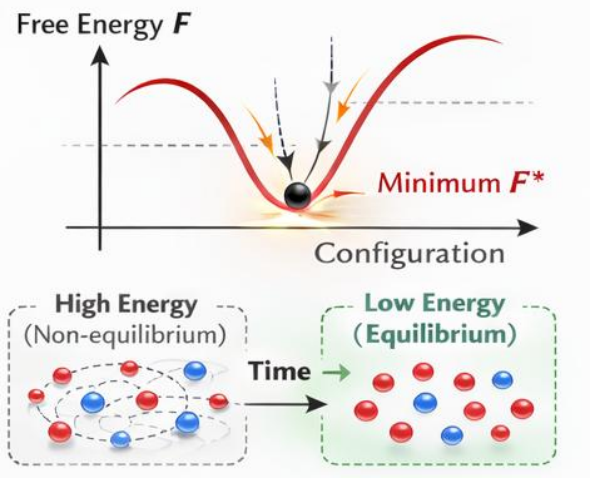
- **Physical systems naturally evolve toward lower energy states.**

1. Ball Rolling Down



System → lower E

2. Thermodynamic Equilibrium

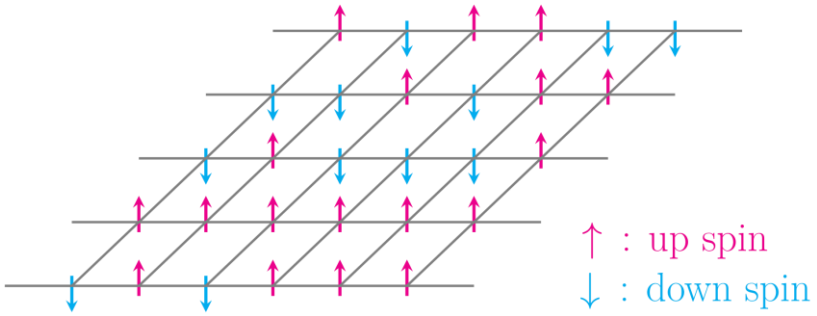


System → Equilibrium State

*If we **encode a problem** into an energy function E(s), physics will solve it !* → *The Ising Model*

The Ising Model: A Physical Energy Function

- The Ising model was introduced in statistical physics to study ferromagnetism in magnetic materials.
- It describes a system of interacting spins.



https://en.wikipedia.org/wiki/Ising_model

- Each node represents a spin variable σ_i
- Spins take binary values $\sigma_i \in \{-1,+1\}$
- Pink arrows: spin up (+1)
- Blue arrows: spin down (-1)
- Neighboring spins interact through couplings J_{ij}
- Each spin can be influenced by a local bias h_i

- The energy of the system is defined by

$$E(\sigma) = -\frac{1}{2}\sum_{i \neq j} J_{ij}\,\sigma_i \sigma_j - \sum_i h_i\ \sigma_i$$

- J_{ij} : pairwise couplings
- h_i : local bias
- σ_i : spins

*If we encode an optimization problem into **J** and **h**, the physical system will search for low-energy spin configuration.*

Spin Dynamics

Spin dynamics follow gradient descent on energy:

$$\frac{ds}{dt} = -\nabla E(s)$$

Compute how the energy evolves

- By the chain rule:

$$\frac{dE}{dt} = \nabla E(s)^{\top} \frac{ds}{dt} = -|\nabla E(s)|^2 \leq 0$$

At steady state:

$$\frac{ds}{dt} = -\nabla E(\mathrm{s}) = 0$$

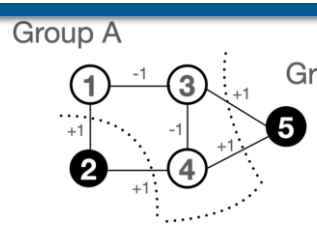
Therefore:

- Energy decreases monotonically
- System converges to local/global minima

Problems Encoded by the Ising Model

1. Max-Cut:

Partition graph to maximize total weight of edges between the two groups .



[3] Nat. Scientific Reports (Onizawa, 2024).

- Let spins $\sigma_i \in \{-1, +1\}$ represent the two partitions.
- An edge (i, j) is cut when spins are different:

$$\text{Cut indicator} = \frac{1 - \sigma_i \sigma_j}{2}$$

- Max-Cut objective:

$$\max \sum_{(i,j)\in E} W_{ij} \frac{1 - \sigma_i \sigma_j}{2}$$

- Ignoring the constant term:

$$\max \text{Cut} \Leftrightarrow \min \sum_{(i,j)\in E} \frac{1}{2} W_{ij} \sigma_i \sigma_j$$

- Ising Energy:

$$H(\sigma) = -\frac{1}{2} \sum_{(i,j)\in E} J_{ij}\, \sigma_i \sigma_j \quad , \quad J_{ij} = -W_{ij}$$

2. Network Community Detection:

Identify highly connected communities

https://www.analyticsvidhya.com/blog/2020/04/community-detection-graphs-networks/

Example: Social networks

Connections may not be directly given. They can be inferred from user features such as:

- interaction frequency
- mutual friends
- shared interests

Each user has a feature vector x_i.

Feature similarity defines the coupling strength:

$$J_{ij} = sim(x_i, x_j)$$

Encode as Ising Coupling:

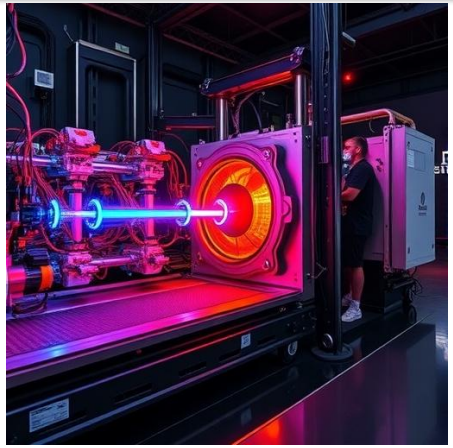
$$H(\sigma) = -\frac{1}{2} \sum_{(i,j)\in E} J_{ij}\, \sigma_i \sigma_j$$

From Ising Model to Ising Hardware

Platform	Spin	Coupling	Advantages	Limitations
Quantum Ising Machine	qubit state	quantum interaction	quantum tunneling (potentially escape local minima)	cryogenic system, limited connectivity
Coherent Ising Machine	optical phase	optical interference and feedback	massive parallel optical dynamics	complex feedback control
CMOS Ising Machine	voltage / ring-oscillator phase	current/ transmission gate	Scalable CMOS implementation	noise, mismatch

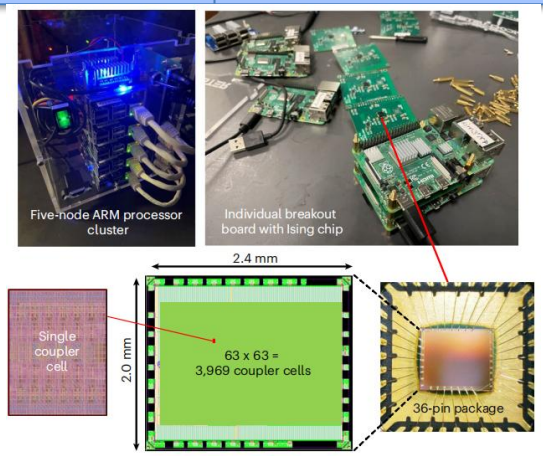
Quantum Ising Machine

https://thequantuminsider.com/2025/05/20/d-wave-announces-general-availability-of-advantage2-quantum-computer/



Coherent Ising Machine

https://bioengineer.org/versatile-large-scale-coherent-ising-machine-advances-across-spectrum/



CMOS Ising Machine

[4] Nature Electronics (Cılasun, 2025)

Can Ising machine learn?

How do we train an energy-based physical system?
— Equilibrium Propagation

Physical Neural Networks

Network Dynamics

- Neurons evolve according to energy minimization:

$$\frac{ds}{dt} = -\nabla E(s)$$

  - s: continuous neuron states
  - E: system energy

Energy Minimization as Computation

- Energy decreases during system evolution
- System relaxes toward an energy minimum
- Computation emerges from physical relaxation

Conventional vs Physical Computation

Conventional:

- Layer-by-layer computation
- memory movement

Physical:

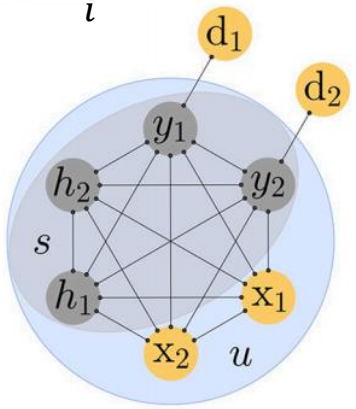
- Dynamical system evolution

Continuous Hopfield Network

The network is defined by an energy function:

$$E(u) = \frac{1}{2}\sum_i u_i^2 - \frac{1}{2}\sum_{i \neq j} W_{ij}\,\rho(u_i)\rho(u_j) - \sum_i b_i \rho(u_i)$$

- u_i: continuous neuron state
- $\rho(u)$: activation function
- $W_{ij} = W_{ji}$:symmetric weights
- b_i: biases



u = {x, h, y}
s = {h, y}
x: input data
(are clamped)
h: hidden neurons
y: output neurons
d: targets
θ ={W, b}

[5] Front. Comput. Neurosci. (Scellier, 2017).

Prediction as an Equilibrium (local/ global minimum)

- For an input x, define prediction as: $s_\theta^0 = \arg\min_s E\,(x, s; \theta)$
- Prediction is implicit

Learning Through Equilibrium

- Cost function: $C := \frac{1}{2}(y - d)^2$
- Learning requires differentiating through equilibrium

The Differentiation Problem

We want:

$$\frac{\partial C(s_\theta^0, x)}{\partial \theta}$$

Using the chain rule:

$$\frac{\partial C(s_\theta^0, x)}{\partial \theta} = \frac{\partial C(s_\theta^0, x)}{\partial s}\frac{ds_\theta^0}{d\theta}$$

But s_θ^0 satisfies the equilibrium condition:

$$\frac{\partial E}{\partial s}\left(\theta, x, s_\theta^0\right) = 0$$

Differentiating this w.r.t. θ gives:

$$\frac{\partial^2 E}{\partial s^2}\frac{ds_\theta^0}{d\theta} + \frac{\partial^2 E}{\partial s \partial \theta} = 0$$

Thus:

$$\frac{ds_\theta^0}{d\theta} = -\left(\frac{\partial^2 E}{\partial s^2}\right)^{-1}\frac{\partial^2 E}{\partial s \partial \theta} \quad \rightarrow \quad \frac{\partial C(s_\theta^0, x)}{\partial \theta} = -\frac{\partial C(s_\theta^0, x)}{\partial s}\left(\frac{\partial^2 E}{\partial s^2}\right)^{-1}\frac{\partial^2 E}{\partial s \partial \theta}$$

Problems:

1. Hessian inverse appears.
2. This is non-local and expensive.

Equilibrium Propagation (EP)

- **Define total energy**:

$$F(\theta, x, s, \beta) = E(\theta, x, s) + \beta C(s)$$

  - β is a small nudging parameter (This introduces supervision directly into energy) .
- **Free phase** ($\beta = 0$): System converges to s_θ^0, which corresponds to standard inference.

$$\frac{\partial F}{\partial s}(\theta, x, s_\theta^0, 0) = \frac{\partial E}{\partial s}(\theta, x, s_\theta^0) = 0$$

- **Nudge phase** ($\beta \neq 0$): System converges to s_θ^β

$$\frac{\partial F\left(\theta, x, s_\theta^\beta, \beta\right)}{\partial s} = \frac{\partial E\left(\theta, x, s_\theta^\beta\right)}{\partial s} + \beta \frac{\partial C\left(s_\theta^\beta\right)}{\partial s} = 0$$

  - $\beta > 0$: outputs are pushed toward targets
  - $\beta < 0$: outputs are pushed away from targets
- The gradient of cost function at equilibrium satisfies:

$$\frac{\partial \mathrm{C}(s_\theta^0)}{\partial \theta} = \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial F}{\partial \theta}\left(\theta, x, \beta, s_\theta^\beta\right) - \frac{\partial F}{\partial \theta}(\theta, x, 0, s_\theta^0)\right) = \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E}{\partial \theta}\left(\theta, x, \beta, s_\theta^\beta\right) - \frac{\partial E}{\partial \theta}(\theta, x, 0, s_\theta^0)\right)$$

  - **Gradient equals difference of two equilibria**
  - **No Hessian inverse**
  - **No backward pass**

Proof of EP

Total energy:

$$F\left(\theta, x, \beta, s_\theta^\beta\right) = E\left(\theta, x, s_\theta^\beta\right) + \beta C\left(s_\theta^\beta, x\right)$$

Equilibrium Condition:

$$\frac{\partial F}{\partial s}\left(\theta, x, \beta, s_\theta^\beta\right) = 0$$

Cross-Derivative Symmetry:

$$\left(\frac{d^2F}{d\theta d\beta}\right)^T = \frac{d^2F}{d\beta d\theta}$$

Applying the chain rule:

$$\frac{dF}{d\beta} = \frac{\partial F}{\partial \beta} + \frac{\partial F}{\partial s}\frac{\partial s_\theta^\beta}{\partial \beta}$$

At equilibrium:

$$\frac{\partial F}{\partial s} = 0$$

So

$$\frac{dF}{d\beta} = \frac{\partial F}{\partial \beta}$$

Similarly,

$$\frac{dF}{d\theta} = \frac{\partial F}{\partial \theta}$$

Substituting into the cross-derivative relation gives

$$\left(\frac{d}{d\theta}\frac{\partial F}{\partial \beta}\right)^T = \frac{d}{d\beta}\frac{\partial F}{\partial \theta}$$

Also,

$$\frac{d}{d\theta}\frac{\partial F}{\partial \beta}\left(\theta, x, 0, s_\theta^0\right) = \frac{\partial C\left(s_\theta^0, x\right)}{\partial \theta}$$

$$\frac{d}{d\beta}\frac{\partial F}{\partial \theta}\left(\theta, x, 0, s_\theta^0\right) = \lim_{\beta \to 0}\frac{1}{\beta}\left(\frac{\partial F}{\partial \theta}\left(\theta, x, \beta, s_\theta^\beta\right) - \frac{\partial F}{\partial \theta}\left(\theta, x, 0, s_\theta^0\right)\right)$$

Gradient:

$$\frac{\partial C\left(s_\theta^0, x\right)}{\partial \theta} = \lim_{\beta \to 0}\frac{1}{\beta}\left(\frac{\partial F}{\partial \theta}\left(\theta, x, \beta, s_\theta^\beta\right) - \frac{\partial F}{\partial \theta}\left(\theta, x, 0, s_\theta^0\right)\right)$$

Weight Update (Hopfield Case)

- For the Hopfield energy:

$$E(u) = \frac{1}{2}\sum_i u_i^2 - \frac{1}{2}\sum_{i \neq j} W_{ij}\,\rho(u_i)\rho\left(u_j\right) - \sum_i b_i \rho(u_i)$$

Thus:

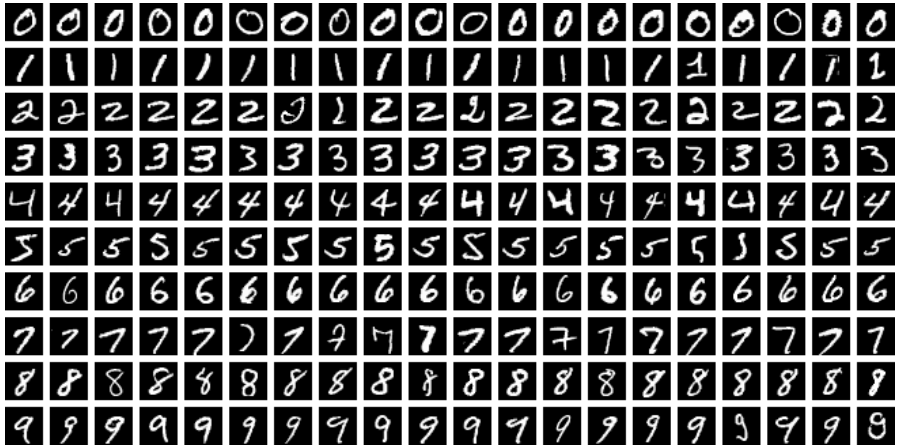
$$\Delta W_{ij} \propto -\frac{\partial \mathrm{C}(s_\theta^0, x)}{\partial W_{ij}} = \frac{1}{\beta}\left(\rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right) - \rho\left(u_i^0\right)\rho\left(u_j^0\right)\right)$$

$$\Delta b_i \propto -\frac{\partial \mathrm{C}(s_\theta^0, x)}{\partial b_i} = \frac{1}{\beta}\left(\rho\left(u_i^\beta\right) - \rho\left(u_i^0\right)\right)$$

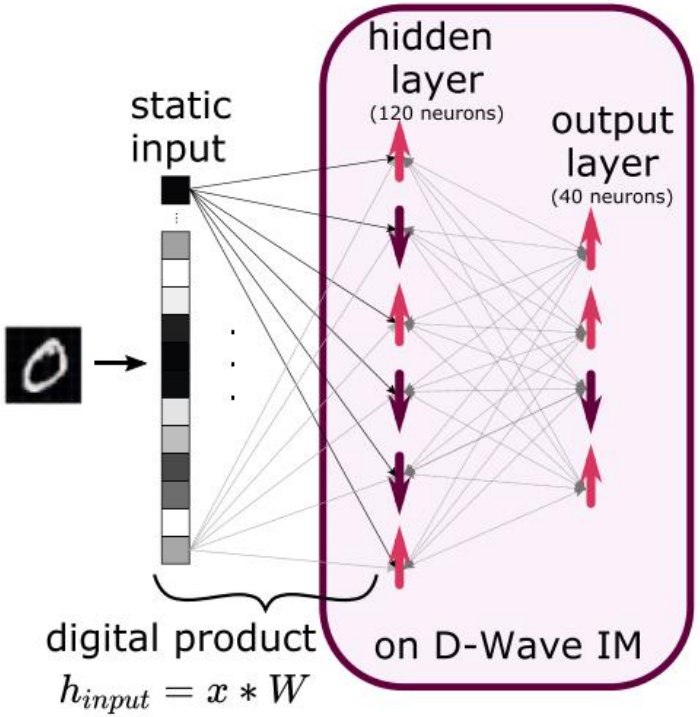
- Fully local update rule!

Training a D-Wave Ising Machine with EP

Example: MNIST handwritten digit recognition



https://en.wikipedia.org/wiki/MNIST_database



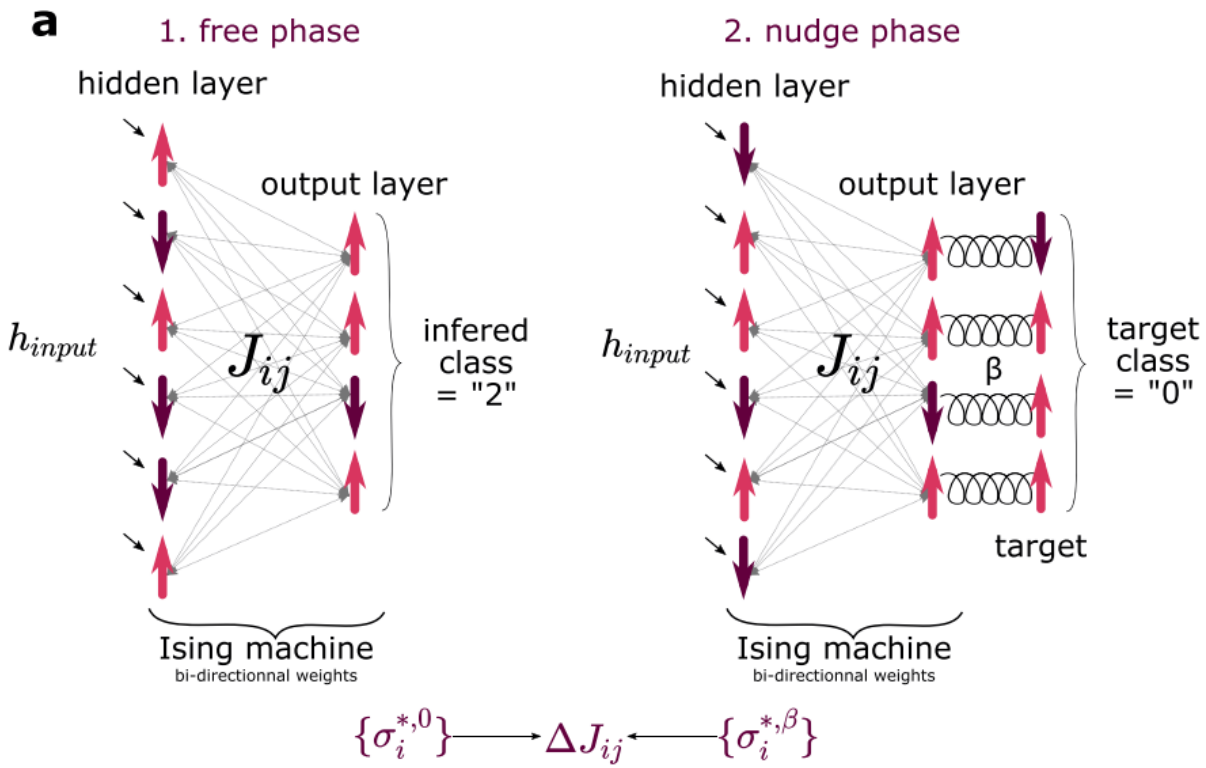
[6] Nature Communications (Laydevant,2024).

Fully connected neuron network on the Ising machine

- First, the product of the input vector (an MNIST image) and the first weight matrix is computed in software. The result is a vector of small constant bias fields that are directly applied to the hidden spins on the Ising machine.
- The hidden and the output layer are implemented on the Ising Machine.

Training a D-Wave Ising Machine with EP

Free phase and nudge phase of the EP algorithm applied to an Ising system.



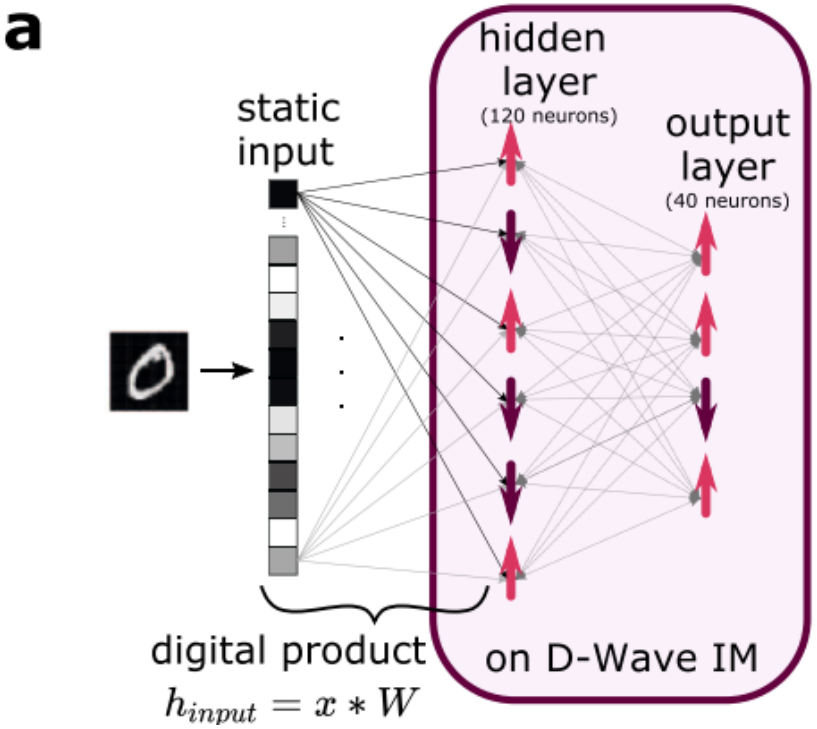
[6] Nature Communications (Laydevant,2024).

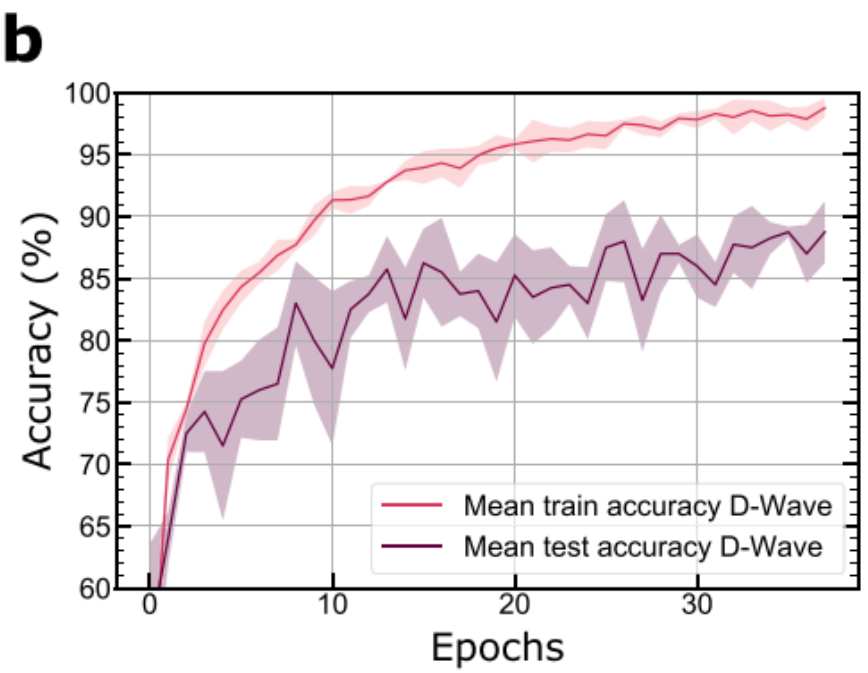
- For both phases, the input is fed to the Ising machine through bias fields with a strength that depends on the input.
- The steady spins states obtained at equilibrium after the free and the nudge phases can be directly measured to compute the parameters updates.

Training a D-Wave Ising Machine with EP

Training on D-Wave:

Train and test accuracy vs. number of epochs.





Challenges of Training Ising Machine with EP

No Damping → Difficulty in Destabilizing

- In continuous Hopfield networks, the dynamics include a leaky term ($-s_i$), which drives neurons out of saturation toward their resting state.

$$\textbf{\textit{Hopfield Energy}}: E(u) = \frac{1}{2}\sum_i u_i^2 - \frac{1}{2}\sum_{i\neq j} W_{ij}\,\rho(u_i)\rho(u_j) - \sum_i b_i \rho(u_i)$$

$$-\frac{\partial E}{\partial s_i} = \rho'(s_i)\left(\sum_{j\neq i} W_{ij}\ \rho(u_j) + b_i\right) - s_i$$

- Ising energy has no explicit damping; after annealing the spin configuration can become rigid (frozen) and small perturbations may fail to shift the equilibrium.

$$\textbf{\textit{Ising Energy}}: E(\sigma) = -\frac{1}{2}\sum_{i\neq j} J_{ij}\,\sigma_i\sigma_j - \sum_i h_i\ \sigma_i$$

$$\langle \sigma_i\sigma_j \rangle_\beta = \langle \sigma_i\sigma_j \rangle_0 \Rightarrow \Delta J_{ij} = 0$$

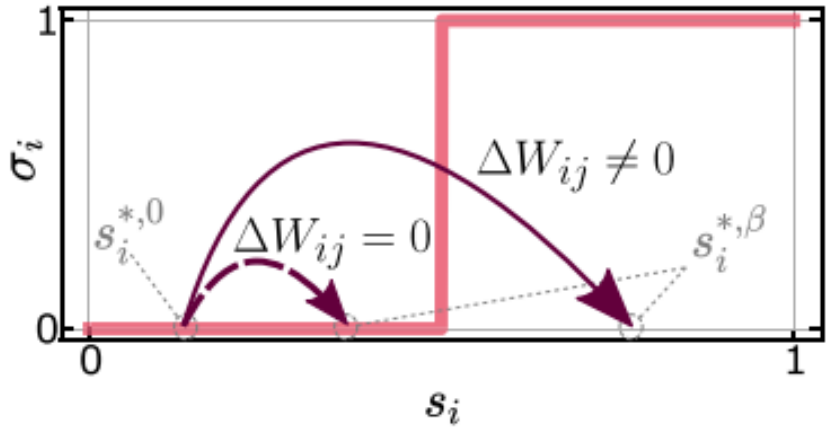
Challenges of Training Ising Machine with EP

Binary Activations vs Continuous Activations

- EP relies on a small-perturbation (linear-response) assumption.
- In continuous activations, a nonzero $\rho'(s_i)$ allows small nudges to induce small state changes.

$$\frac{ds_i}{dt} = -\frac{\partial E}{\partial s_i} = \rho'(s_i)\left(\sum_{j\neq i} W_{ij}\ \rho(u_j) + b_i\right) - s_i$$

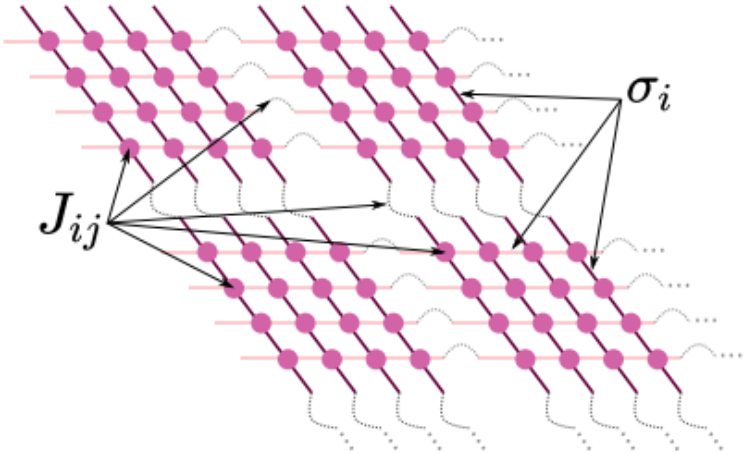
- For binary spins, the response is inherently discrete; small perturbations may produce **no change**, while larger perturbations may induce **abrupt flips**. This breaks the linear-response intuition underlying EP.



Challenges of Training Ising Machine with EP

Sparse / Planar Connectivity → Embedding Overhead （D-Wave)

- Many Ising hardware platforms exhibit sparse (often planar) connectivity.
- Fully-connected logical models therefore require minor embedding.
  - minor embedding: one logical spin is represented by a chain of physical spins strongly positively coupled.
- This introduces overhead in number of spins and robustness.



(D-wave)

[6] Nature Communications (Laydevant,2024).

Nudging in Ising Networks

Because Ising systems have no damping and the spins are binary, the response to nudge can be problematic.

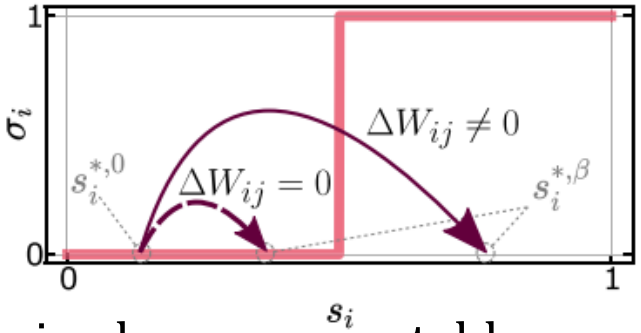
Nudge Dilemma

For small nudge (β):

Network does not change its state → no learning

For large nudge (β):

Abrupt output switching → strong global state change → Learning becomes unstable



Remedies

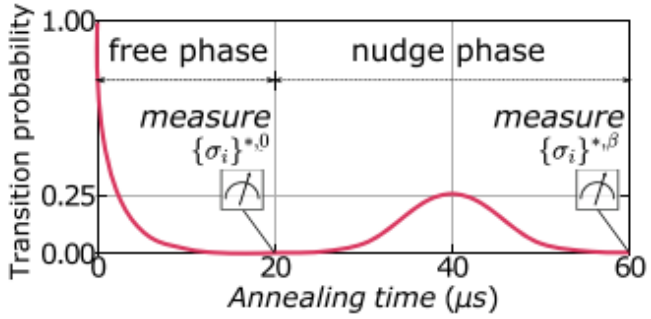
Reverse Annealing:

Reverse annealing briefly increases the transition probability to enable local exploration, and then reduces it back to zero as the system settles.

Multiple Output Units per Class:

Represent each class with several spins→

increase flip events→ smoother error propagation

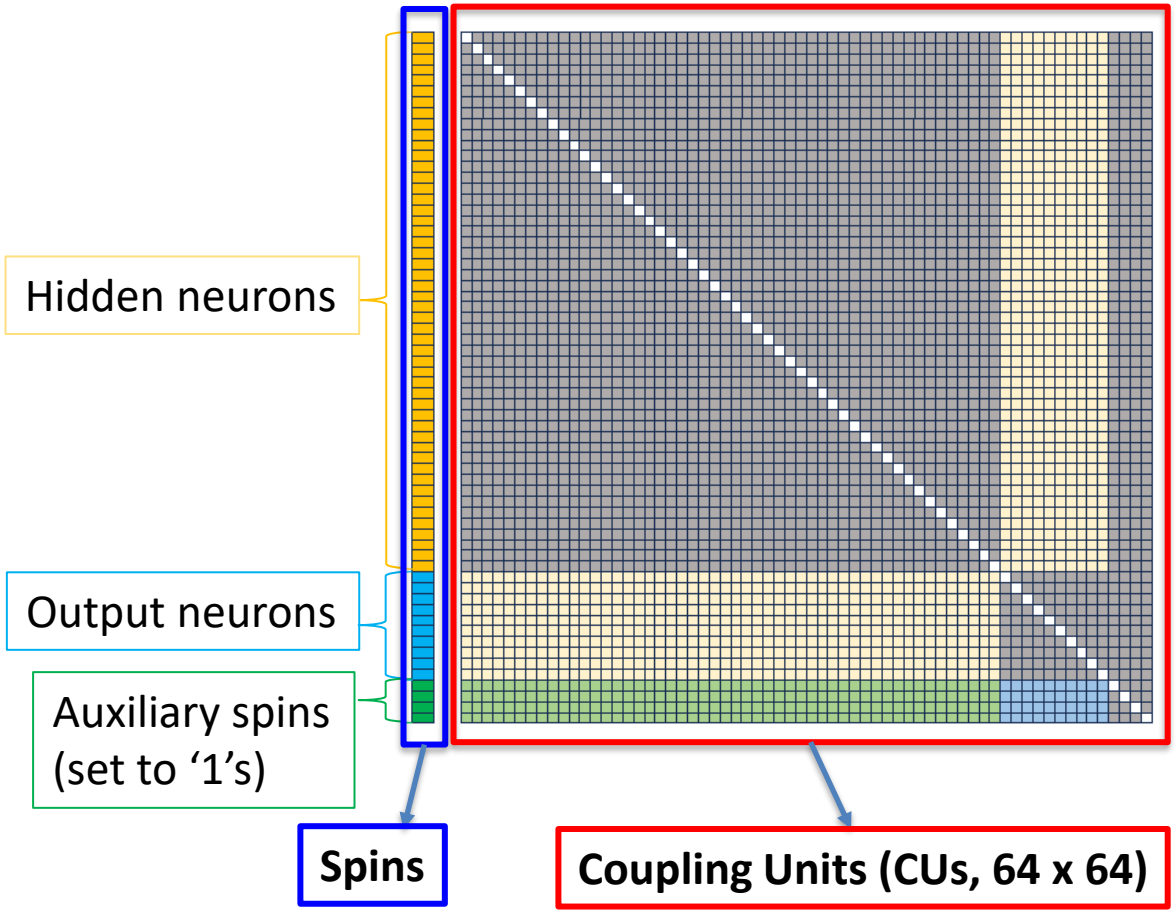


[6] Nature Communications (Laydevant,2024).

Training Proposed All-to-All CMOS Ising Machine with EP

Training 64-Spin All-to-All CMOS Ising Machine with EP

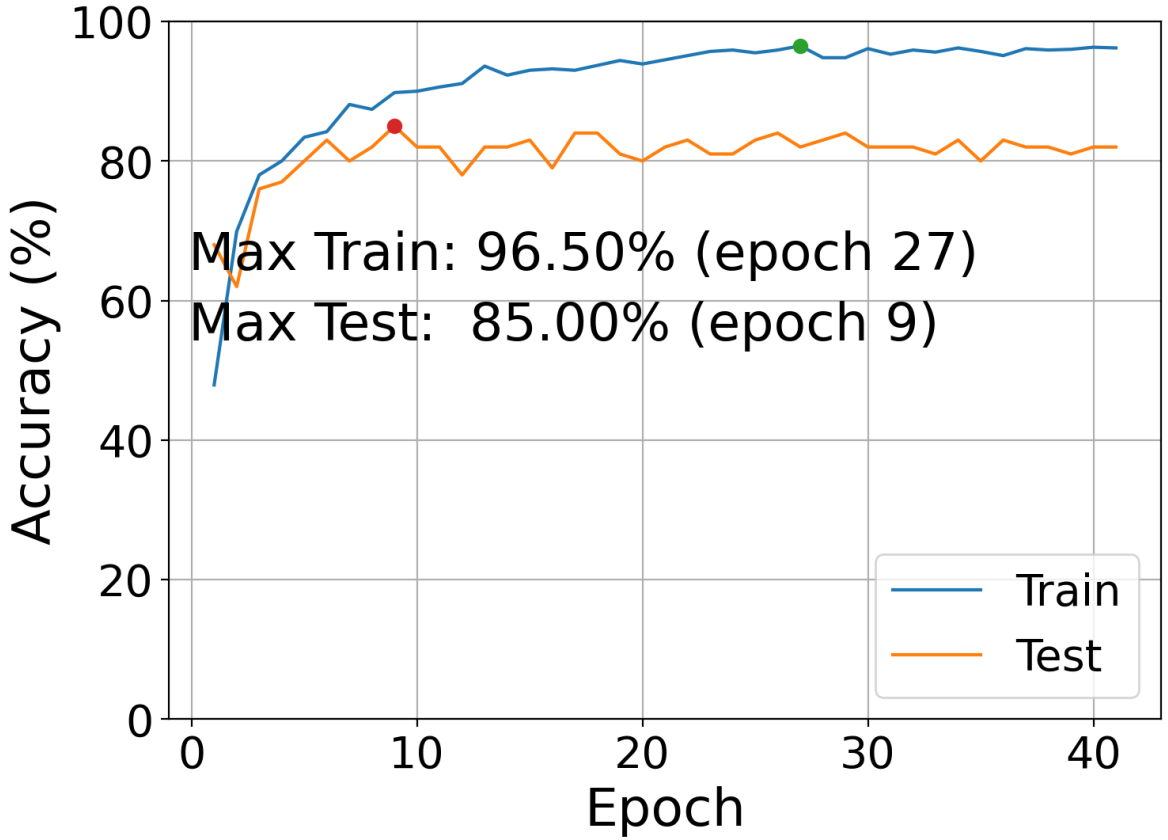
Ising Energy: $E(\sigma) = -\frac{1}{2}\sum_{i \neq j} J_{ij}\,\sigma_i\sigma_j - \sum_i h_i\,\sigma_i$



- Disabled CUs
- J_{ij}
- $h_{input} + h_{bias}$
- $h_{bias} - \beta\sigma_{\text{target}}$

$$\beta = \begin{cases} 0 & \text{, free phase} \\ const & \text{, nudge phase} \end{cases}$$

Training 64-Spin All-to-All CMOS Ising Machine with EP

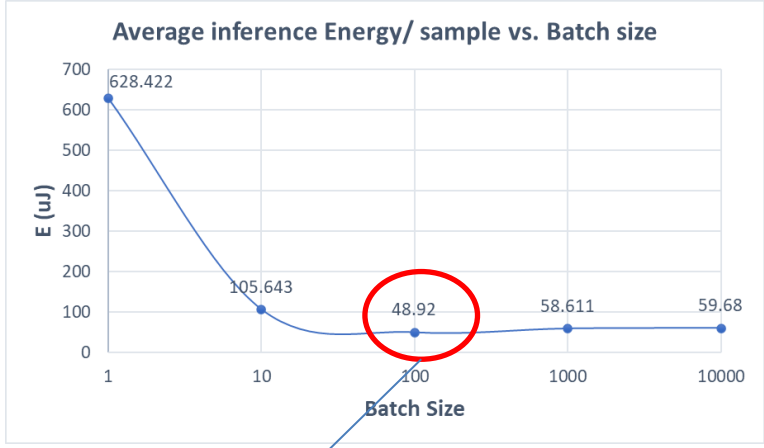


Comparison of Energy Consumption

CPU energy consumption were recorded using **Intel Performance Counter Monitor (PCM)** on the **Intel i5-14400** desktop.

```
class SimpleFC(nn.Module):
    def __init__(self):
        super().__init__()
        self.net = nn.Sequential(
            nn.Flatten(),
            nn.Linear(28*28, 50),
            nn.ReLU(),
            nn.Linear(50, 10)
        )
```

Count the energy consumption of this layer during inference.



Inference energy/sample in proposed **Ising Machine**:

Power × Time to reach equilibrium ≈ 12 mW ×3 μs = **0.036 μJ**

The software implementation consumes ~**1359** times more energy than the proposed Ising Machine.

Summary

Ising Machines for Optimization

- Optimization problems can be encoded as energy minimization:

$$E(\sigma) = -\frac{1}{2}\sum_{i,j} J_{ij}\,\sigma_i\sigma_j - \sum_i h_i\,\sigma_i$$

Equilibrium Propagation for Learning

- Gradient can be computed from two equilibria:

$$\Delta W_{ij} \propto \frac{1}{\beta}\left(\rho\left(u_i^{\beta}\right)\rho\left(u_j^{\beta}\right) - \rho(u_i^0)\rho(u_j^0)\right)$$

$$\Delta b_i \propto \frac{1}{\beta}\left(\rho\left(u_i^{\beta}\right) - \rho(u_i^0)\right)$$

- In Ising machines: $\rho(u_i) \in \{-1, +1\}$

Key Advantages

- **Fast computation** through physical relaxation
- **Energy-efficient** hardware, since computation occurs through the physical dynamics of the system with minimal data movement.

References

[1] Samborska, V. Scaling up: how increasing inputs has made artificial intelligence more capable. Our World in Data https://ourworldindata.org/scaling-up-ai (2025).
[2] Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R. & Eleftheriou, E. Memory devices and applications for in-memory computing. Nat. Nanotechnol. 15, 529–544 (2020).
[3] Onizawa, N. and Hanyu, T., 2024. Enhanced convergence in p-bit based simulated annealing with partial deactivation for large-scale combinatorial optimization problems. Scientific Reports, 14(1), p.1339.
[4] Cılasun, H., Moy, W., Zeng, Z., Islam, T., Lo, H., Vanasse, A., Tan, M., Anees, M., S, R., Kumar, A. and Sapatnekar, S.S., 2025. A coupled-oscillator-based Ising chip for combinatorial optimization. Nature Electronics, 8(6), pp.537-546.
[5] Scellier, B. & Bengio, Y. Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation. Front. Comput. Neurosci. 11, (2017).
[6] Laydevant, J., Marković, D. and Grollier, J., 2024. Training an Ising machine with equilibrium propagation. Nature Communications, 15(1), p.3671.

Thanks for Your Listening!