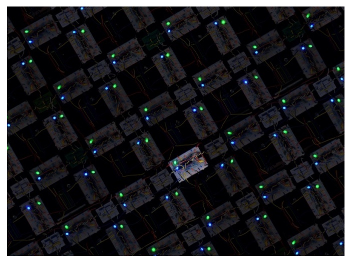
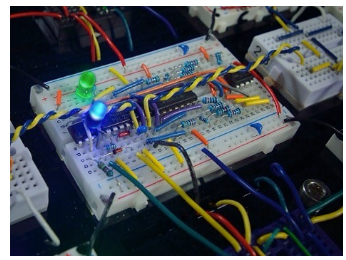
Autonomous Learning Metamaterials

Douglas J. Durian <djdurian@physics.upenn.edu>
University of Pennsylvania, Philadelphia PA

- Physical systems made of many copies of a repeat unit, each of which uses a *local learning rule* to collectively optimize a *global* cost function defining a task to be "computed" physically. Eg "*Contrastive Local Learning Networks*"

  Analog in-memory training for analog in-memory analog computing for control, metamaterials with complex functionality, AI,…





- Boulder School 2024
  - Lecture 1: learning systems and rules
  - Lecture 2: electronic realizations
  - Lecture 3: mechanical realizations

1

ACKNOWLEDGEMENTS

experiment

Sam Dillavou

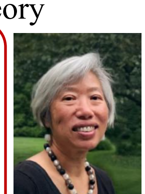
Jacob Wycoff

Benjamin Beyer

Lauren Altman

theory

Menachem Stern



Andrea Liu

2nd generation

Marc Miskin

Dinesh Jayaraman, Maggie Miller, Shivangi Misra, Tarunyaa Sivakumar, Cynthia Sung

University of Pennsylvania

LRSM

NSF UPenn MRSEC

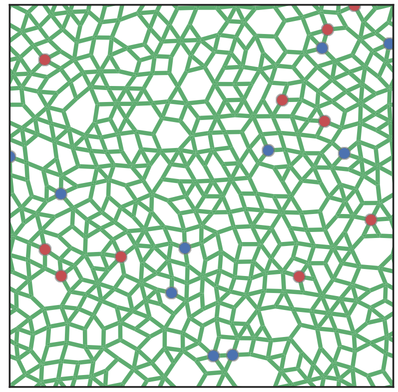
2

Response of Networks to Stimuli

"forward" problem for electrical resistors, fluidic pipes, mechanical springs,…

- Fixed boundary conditions: Apply voltage, pressure, current, force, displacements, etc. to designated ***"input"*** nodes or edges, and let the network equilibrate. What happens?
  - The unfixed ***physical degrees of freedom*** will relax and stop evolving once Kirchhoff's laws or force balance is satisfied (and power or elastic energy is minimized)
  - The final behavior of designated ***"output"*** nodes or edges depends on the BCs as well as the conductances / stiffnesses of *all* the edges
    - Theory: the outputs are predicted by solving a *global* optimization problem (hard, if elements are nonlinear)
    - ***Expt: the outputs are computed physically – quickly & for free***

- *inputs*
- *outputs*

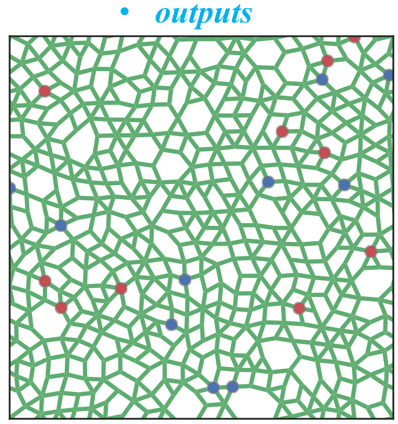


3

This is an analog physical computer…

- …if the output values represent the answer to some desired computational problem (big IF!)
  - A "physical neural network (PNN)"
  - Example tasks / functions:
    - Outputs are at $\{V_j\}$ when inputs are at $\{V_i\}$ ("allostery")
    - Outputs are a desired linear combo of inputs
    - Matrix multiplication
    - Classification: e.g inputs are grayscale pixel data and outputs indicate whether the image is a dog/cat/etc or which squiggles represent which letters or numbers
  - How to train?

- *inputs*
- *outputs*



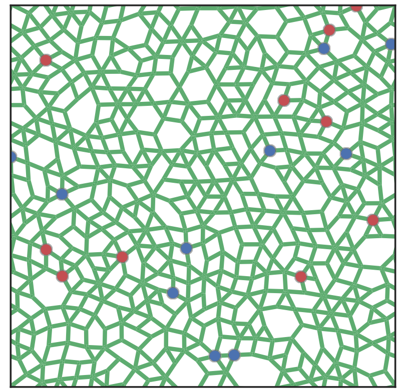
4

Learning degrees of freedom

- The conductances / stiffnesses of the edges (i.e. the "learning degrees of freedom") must be chosen according to the desired functionality.
- How can the network learn these parameters? This is a harder global optimization problem.
  - Theoretically:
    - Directly solve inverse problem {outputs} == {desired values} ???
    - Do gradient descent on (actual – desired)^2 ???

      Seems like machine learning, with analogous issues of over-parameterization, rugged landscape with many local minima,…
  - Experimentally:
    - Mimic theory, or do something else?

- *inputs*
- *outputs*



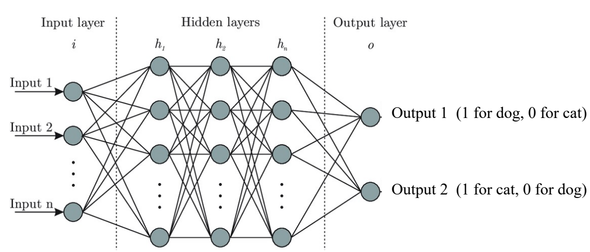
Take an extended detour through prior approaches…

5

Artificial/Digital Neural Networks

- Each node is weighted sum of nodes in prior layer (set to 0 if negative). Weights are the "learning parameters / learning degrees of freedom"

e.g. classification:



Use gradient descent to minimize a cost function that penalizes mistakes wrt training data

(*e.g.* by backpropagation and GPUs)

- The adjustment of each "neuron" depends on ***all*** other neurons in the layer (i.e. needs global info)
- Memory & computer power are needed, and both grow rapidly with network size

  AlphaGo: 12 layers of (19x19)x17 nodes; hardware 64 GPU + 19 CPU + 4 TPU cost \$25M

  ChatGPT: 96 layers with 12,288 hidden layer dimensions and 175 billion learning parameters

  (human brain: 86 billion neurons and 100-1000 trillion synapses)

6

Hybrid Physical Neural Networks

- E.g. crossbar array of memristors, trained by backprop on computer model

nature communications

Review article

https://doi.org/10.1038/s41467-024-45670-9

Hardware implementation of memristor-based artificial neural networks

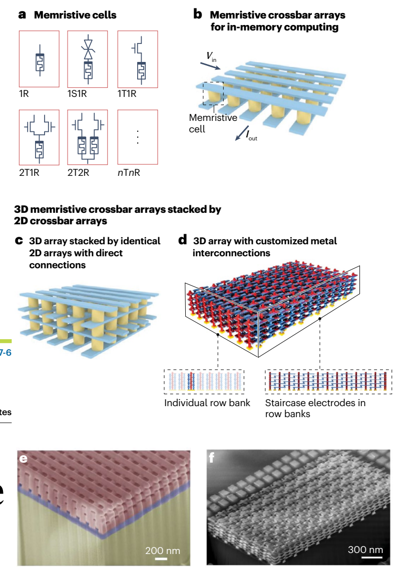
Received: 8 June 2023

Accepted: 1 February 2024

Published online: 04 March 2024

Check for updates

Fernando Aguirre[1,2], Abu Sebastian[3], Manuel Le Gallo[3], Wenhao Song[4], Tong Wang[4], J. Joshua Yang[4], Wei Lu[5], Meng-Fan Chang[6], Daniele Ielmini[7], Yuchao Yang[8], Adnan Mehonic[9], Anthony Kenyon[9], Marco A. Villena[1], Juan B. Roldán[10], Yuting Wu[5], Hung-Hsi Hsu[6], Nagarajan Raghavan[11], Jordi Suñé[2], Enrique Miranda[2], Ahmed Eltawil[12], Gianluca Setti[12], Kamilya Smagulova[12], Khaled N. Salama[12], Olga Krestinskaya[12], Xiaobing Yan[13], Kah-Wee Ang[14], Samarth Jain[14], Sifan Li[14], Osamah Alharbi[1], Sebastian Pazos[1] & Mario Lanza[1]



nature reviews electrical engineering

https://doi.org/10.1038/s44287-024-00037-6

Review article

Check for updates

Memristor-based hardware accelerators for artificial intelligence

Yi Huang[1], Takashi Ando[2], Abu Sebastian[3], Meng-Fan Chang[4], J. Joshua Yang[5] & Qiangfei Xia[1]

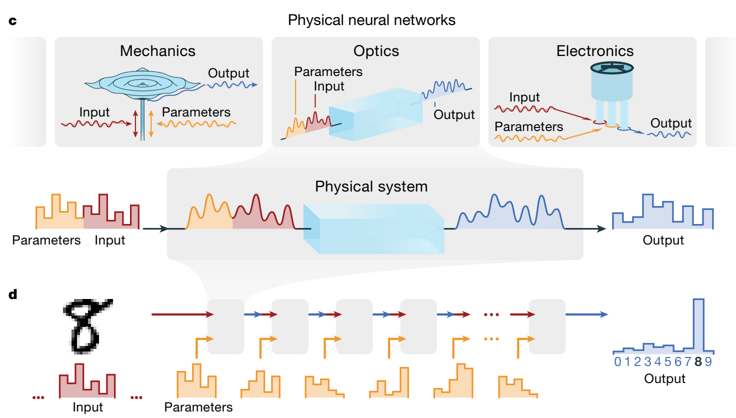
Hybrid Physical Neural Networks

- Input/output encoded in wave; trained by backprop on computer model

Deep physical neural networks trained with backpropagation

Logan G. Wright, Tatsuhiro Onodera, Martin M. Stein, Tianyu Wang, Darren T. Schachter, Zoey Hu & Peter L. McMahon

Nature 601, 549–555 (2022) | Cite this article



All-Physical Neural Networks

- small systems trained physically with *in-situ* backpropagation?

Research Article — Vol. 9, No. 3 / March 2021 / Photonics Research — B71

PHOTONICS Research (theory)

Backpropagation through nonlinear units for the all-optical training of neural networks

Xianxin Guo,[1,2,3,5,†] Thomas D. Barrett,[2,6,†] Zhiming M. Wang,[1,7] and A. I. Lvovsky[2,4,8]

Research Article — Vol. 5, No. 7 / July 2018 / Optica — 864

optica (theory)

Training of photonic neural networks through *in situ* backpropagation and gradient measurement

Tyler W. Hughes,[1] Momchil Minkov,[2] Yu Shi,[2] and Shanhui Fan[2,*]

Research Article — Vol. 9, No. 7 / July 2022 / Optica — 803

OPTICA

Hybrid training of optical neural networks

James Spall,[1,†] Xianxin Guo,[1,2,3,†] and A. I. Lvovsky[1,2,4]

arXiv > cs > arXiv:2404.15471 (theory)

Computer Science > Machine Learning

[Submitted on 23 Apr 2024]

Training all-mechanical neural networks for task learning through in situ backpropagation

Shuaifeng Li, Xiaoming Mao

NANOPHOTONICS — *Science* **380**, 398–404 (2023)

Experimentally realized in situ backpropagation for deep learning in photonic neural networks

Sunil Pai[1*]†, Zhanghao Sun[1], Tyler W. Hughes[2]‡, Taewon Park[1], Ben Bartlett[2]†, Ian A. D. Williamson[1]§, Momchil Minkov[1]‡, Maziyar Milanizadeh[3], Nathnael Abebe[1]#, Francesco Morichetti[3], Andrea Melloni[3], Shanhui Fan[1], Olav Solgaard[1], David A. B. Miller[1]

9

Physical Neural Networks

- These approaches promise great speed and energy efficiency for inference

- But they are hard to scale up large enough to compete with big ANNs
  - training by backprop requires global information (to update one neuron requires information about weights of all other neurons in the layer)

    "top down" by external agent with vast memory and computational power

  - reality gap between actual device and computer model used for training
  - need to externally read/write each edge, writing is often imperfectly done

Any lessons to be learned from *real* neural networks?

10

The human brain

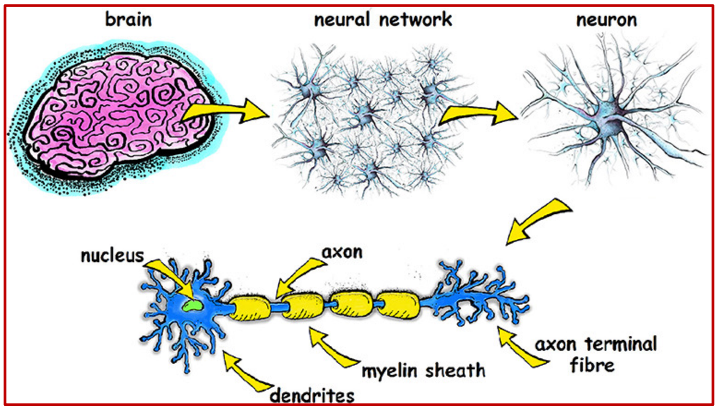
- 86 billion neurons and 100-1000 trillion synapses (very highly connected)

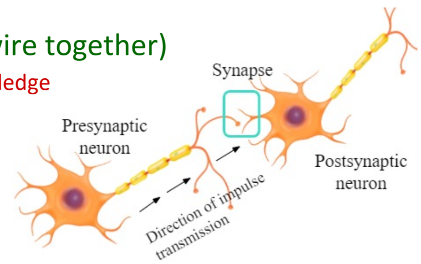
  analog & digital; slow and noisy compared to modern digital electronics

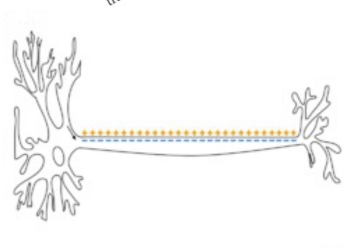
  – learning changes the conductances and connections

  – this is done by ***local rules*** (Hebbian: if fire together the wire together)

  "bottom up" with no external memory or computation or global knowledge

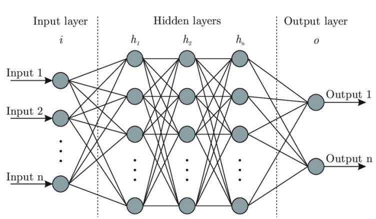






Real vs Artificial Neural Networks

- One is far more capable and energy-efficient than the other



- **Top down** gradient descent on cost function needing global network details
- Relatively narrow range of tasks
- Mostly feed forward
- Fragile wrt damage
- Costs a lot of energy
  - 200kJ per ChatGPT query
  - 10,000kJ per image generation

- **Bottom up** learning using local rules and local information
- Controls thoughts, memory, senses, motor skills, regulation…
- Very recurrent / highly connected
- Robust to damage
- Relatively energy efficient
  - 20 W = 1700kJ/day

Spiking Neural Networks

- Hardware that mimics networks of real neurons (neuromorphic computing)

Neuromorphics chips with O(10^4) neurons can now be microfabricated, eg:

566 IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 43, NO. 2, FEBRUARY 2008

A 128×128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor

Patrick Lichtsteiner, *Member, IEEE*, Christoph Posch, *Member, IEEE*, and Tobi Delbruck, *Senior Member, IEEE*

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-23342-2 OPEN

Check for updates

An electronic neuromorphic system for real-time detection of high frequency oscillations (HFO) in intracranial EEG

Mohammadali Sharifshazileh[1,2,3], Karla Burelo[1,2,3], Johannes Sarnthein[2], & Giacomo Indiveri[1]

Difficult to scale up truly large networks

13

With all this context, let's jump into the main topic…

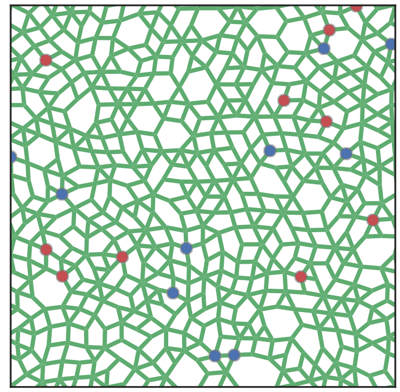
14

Contrastive Local Learning Networks

- Return to recurrent physical networks of conductors, pipes, springs,…

- Develop local learning rules based on contrast of behavior to different boundary conditions
  - "free" where output node values are measured
  - "clamped" where training data is imposed on output nodes

- *inputs*
- *outputs*



Learning should emerge from the bottom up, rather than be imposed from the top down. Being recurrent and bottom-up should bring some brain-like advantages

Bottom-up learning of complex functionality using local rules

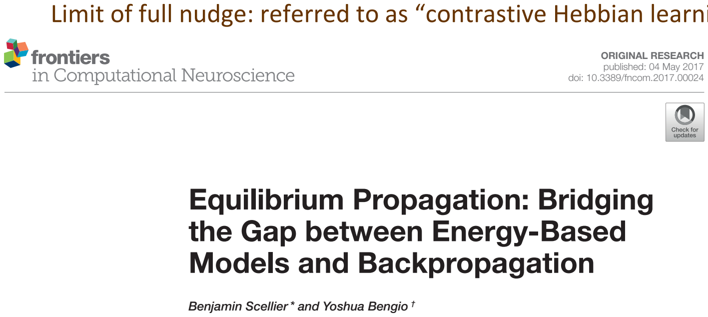
15

Equilibrium Propagation (2017)

- For voltage inputs, inject current into outputs in small proportion to error

  Limit of small nudge: becomes exact gradient decent on (actual-desired)^2 loss function

  Limit of full nudge: referred to as "contrastive Hebbian learning"



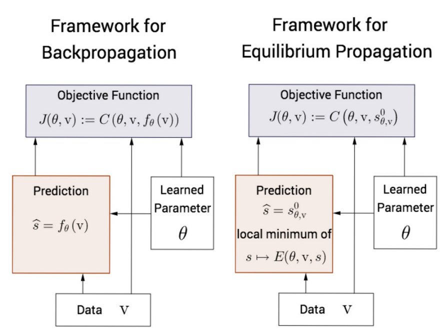
frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 04 May 2017
doi: 10.3389/fncom.2017.00024

Check for updates

Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation

*Benjamin Scellier * and Yoshua Bengio †*

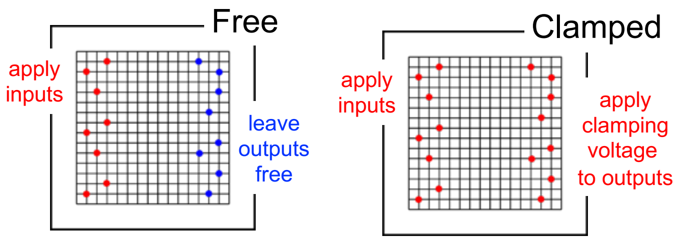


	Backprop	Equilibrium Prop	Contrastive Hebbian Learning
First Phase	Forward Pass	Free Phase	Free Phase (or Negative Phase)
Second Phase	Backward Pass	Weakly Clamped Phase	Clamped Phase (or Positive Phase)

16

Coupled Learning (2021)

- A scheme to adjust conductances based on behavior under **free** versus **clamped** boundary conditions (Stern, Hexner, Rocks, Liu, Phys Rev X 2021):



Menachem "Nachi" Stern

Nudge each output node closer to desired value (according to the training data):

$$V_{Clamped} = V_{Free} + \eta(V_{Desired} - V_{Free})$$

Learning Degrees of Freedom:
conductances k_j of each edge
Physical Degrees of Freedom:
current or pressure/voltage across each edge *("computed" for free, thanks to physics)*

Pop quiz (the crux!): Which circuit has higher dissipation rate, P, free or clamped?
Answer: P is greater in the clamped state *(thanks to physics)*

17

Coupled Learning Rule

- Traditional loss/cost function = (desired response – free response)^2 {>0}
  - must be squared to guarantee it's positive, for minimization by gradient descent
- New **contrast function** = dissipation rate difference, $\mathcal{P}^{clamped} - \mathcal{P}^{free}$ {>0}
  - positive due to optimization of energy functional over physical degrees of freedom for given BCs

Evolve the edge conductances to drive contrast function to zero:

$$\dot{k}_j \propto -\frac{\partial}{\partial k_j}\left[\mathcal{P}^{clamped} - \mathcal{P}^{free}\right]$$

$$= -\frac{\partial}{\partial k_j}\left[\sum_i (V_i^2 k_i)^{clamped} - \sum_i (V_i^2 k_i)^{free}\right]$$

$$= -\left[(V_j^2)^{clamped} - (V_j^2)^{free}\right]$$

This rule is LOCAL
{squaring $\mathcal{P}^{clamped} - \mathcal{P}^{free}$ gives global prefactor}

[Stern, Hexner, Rocks, Liu, Phys Rev X 2021]

18

Coupled Learning Rule

- Meant to be implemented in the lab
  - The voltage drops in under free/clamped conditions depend on conductances
  - Experiments don't take the gradient in 2nd line holding voltage drops constant
  - This issue to be addressed in Andrea's lecture and in homework problem

$$\dot{k}_j \propto -\frac{\partial}{\partial k_j}\left[\mathcal{P}^{clamped} - \mathcal{P}^{free}\right]$$

$$= -\frac{\partial}{\partial k_j}\left[\sum_i (V_i^2 k_i)^{clamped} - \sum_i (V_i^2 k_i)^{free}\right]$$

$$= -\left[(V_j^2)^{clamped} - (V_j^2)^{free}\right]$$

This rule is LOCAL
{squaring $\mathcal{P}^{clamped} - \mathcal{P}^{free}$ gives global prefactor}

[Stern, Hexner, Rocks, Liu, Phys Rev X 2021]

19

Coupled Learning, More Generally

- Conjugate physical variables aka *physical degrees of freedom* $\{p_i, q_i\}$ on/across/through each edge i, connected by *learning degrees of freedom* $\{L_{i1}, L_{i2}, \ldots\}$. E.g. {Δvoltage, current} and {resistance, resistor geometry}; {Δpressure, current} and {conductivity, channel geometry}; {force, deformation} and {spring constant, rest length, strut geometry}; {Δvoltage, charge} and {capacitance, plate geometry}; …
- Energy functional $u_i(p_i, q_i, L_{i1}, L_{i2}, \ldots)$ for each edge, e.g. power for current & fluid flow networks and energy for capacitor & elastic networks. Total energy or power $U = \sum u_i$ summed over all edges.
- Clamp: nudge designated output edges j from free-state values $\{p_{jf}\}$ toward target values $\{p_{jT}\}$ by $p_{jc} = \eta p_{jT} + (1-\eta)p_{jf}$ or ditto for q tasks.

Solve several networks and compare gradient descent vs local learning rules. Findings:

$$\frac{dL_i}{dt} = -\gamma \underbrace{\frac{\partial (U_c - U_f)}{\partial L_i}}_{\text{gradient descent}} \cong -\gamma\eta \begin{cases} \left(\frac{\partial (u_{ic}-u_{if})}{\partial L_i}\right)_p & p \text{ tasks} \\ \left(\frac{\partial (u_{ic}-u_{if})}{\partial L_i}\right)_q & q \text{ tasks} \end{cases}$$

gradient descent; local approximation; empirical

- Cost function $C = U_c - U_f$ has form $C \cong \eta^2 f(\{L_i\}; \text{inputs; task})$ as major η dependence (exactly near learning)
- Empirical factor of η is needed for local rule to approximate gradient descent
- Local approximation to global gradient descent becomes exact for near learning

20

Penalty function terminology

- Distinguish different global penalty functions to be minimized (driven to zero):

  Loss/Cost $\mathcal{L} = \sum(\text{desired output} - \text{free output})^2$ is traditional / most important

  Contrast $\mathcal{C} = U_c - U_f$ {clamped minus free energy or power of network, $U = \sum u_i$} is proxy

  Learn by gradient descent on learning degrees of freedom: $\frac{dL_i}{dt} = -\gamma \frac{\partial(\mathcal{L} \text{ or } \mathcal{C})}{\partial L_i}$

  Local Coupled Learning rule: approx. $\frac{dL_i}{dt} = -\gamma \frac{\partial \mathcal{C}}{\partial L_i}$ by $\frac{dL_i}{dt} \cong -\gamma\eta \begin{cases} \left(\frac{\partial(u_{ic}-u_{if})}{\partial L_i}\right)_p & p \text{ tasks} \\ \left(\frac{\partial(u_{ic}-u_{if})}{\partial L_i}\right)_q & q \text{ tasks} \end{cases}$

  Compare $\hat{n}_{loss} \cdot \hat{n}_{cost}$ and $\hat{n}_{loss} \cdot \hat{n}_{local}$ where $\hat{n} = \frac{d\vec{L}}{dt} / \left|\frac{d\vec{L}}{dt}\right|$ is unit vector in learning direction

21

c.f. Equilibrium Propagation

The difference is in the nudge.

E.g. for voltage in / voltage out tasks:

- **Equilibrium propagation:** nudge by clamping current ∝ (desired – free voltage)

  $I_C \propto \eta(V_D - V_F)$


frontiers in Computational Neuroscience

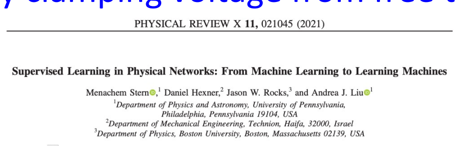
ORIGINAL RESEARCH
published: 04 May 2017
doi: 10.3389/fncom.2017.00024

Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation

*Benjamin Scellier * and Yoshua Bengio †*

- **Coupled learning:** nudge by clamping voltage from free toward desired

  $V_C = V_F + \eta(V_D - V_F)$


PHYSICAL REVIEW X **11,** 021045 (2021)

Supervised Learning in Physical Networks: From Machine Learning to Learning Machines

Menachem Stern,[1] Daniel Hexner,[2] Jason W. Rocks,[3] and Andrea J. Liu[1]

[1]*Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, Pennsylvania 19104, USA*
[2]*Department of Mechanical Engineering, Technion, Haifa, 32000, Israel*
[3]*Department of Physics, Boston University, Boston, Massachusetts 02139, USA*

(Received 10 November 2020; revised 2 April 2021; accepted 5 April 2021; published 28 May 2021)

22

Related local learning rules

- Coupled learning:

$$\dot{k}_j \propto -\frac{\partial}{\partial k_j}\left[P^{clamped} - P^{free}\right]$$

- Directed aging (generally cannot train to complex desired target outputs)

$$\dot{k}_j \propto -\frac{\partial}{\partial k_j}P^{clamped}$$ [Liu & Nagel; Pashine]

- Coupled learning with regularization (small ϵ biases toward low-power solns):

$$\dot{k}_j \propto -\frac{\partial}{\partial k_j}\left[\left(P^{clamped} - P^{free}\right) + \epsilon P^{free}\right]$$

[Stern, Dillavou, Jayaraman, Durian, Liu (APL Machine Learning 2024) – next lecture]

23

Other local learning rules

PHYSICAL REVIEW RESEARCH **5**, 023024 (2023)

Learning by non-interfering feedback chemical signaling in physical networks

Vidyesh Rao Anisetti[1,*] B. Scellier[2] and J. M. Schwarz[1,3]

PNAS RESEARCH ARTICLE | APPLIED PHYSICAL SCIENCES 2023 Vol. 120 No. 27 e2219558120

Learning to learn by using nonequilibrium training protocols for adaptable materials

Martin J. Falk[a,1], Jiayi Wu[a,1], Ayanna Matthews[b], Vedant Sachdeva[b], Nidhi Pashine[c], Margaret L. Gardel[a,d,e,f], Sidney R. Nagel[a,d], and Arvind Murugan[a,d,2]

Frequency Propagation: Multimechanism Learning in Nonlinear Physical Networks

Neural Computation **36, 596–620** (2024)

Vidyesh Rao Anisetti
vvaniset@syr.edu
Physics Department, Syracuse University, Syracuse, NY 13244 U.S.A.

Ananth Kandala
an.kandala@ufl.edu
Department of Physics, University of Florida, Gainesville, FL 32611, U.S.A.

Benjamin Scellier
benjamin.scellier@polytechnique.edu
Department of Mathematics, ETH Zürich, 8092 Zürich, Switzerland

J. M. Schwarz
jmschw02@syr.edu
Physics Department, Syracuse University, Syracuse, NY 13244 U.S.A., and Indian Creek Farm, Ithaca, NY 14850, U.S.A.

24

Reviews on training physical systems

Learning Without Neurons in Physical Systems

Annu. Rev. Condens. Matter Phys. 2023. 14:417–41

Menachem Stern[1] and Arvind Murugan[2]

[1]Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, Pennsylvania; email: nachis@sas.upenn.edu

[2]Department of Physics, University of Chicago, Chicago, Illinois; email: amurugan@uchicago.edu

Energy-Based Learning Algorithms for Analog Computing: A Comparative Study

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Benjamin Scellier	Maxence Ernoult	Jack Kendall	Suhas Kumar
Rain AI	Rain AI	Rain AI	Rain AI
benjamin@rain.ai	maxence@rain.ai	jack@rain.ai	suhas@rain.ai

arXiv > physics > arXiv:2406.03372

Physics > Applied Physics

[Submitted on 5 Jun 2024]

Training of Physical Neural Networks

Ali Momeni, Babak Rahmani, Benjamin Scellier, Logan G. Wright, Peter L. McMahon, Clara C. Wanjura, Yuhang Li, Anas Skalli, Natalia G. Berloff, Tatsuhiro Onodera, Ilker Oguz, Francesco Morichetti, Philipp del Hougne, Manuel Le Gallo, Abu Sebastian, Azalia Mirhoseini, Cheng Zhang, Danijela Marković, Daniel Brunner, Christophe Moser, Sylvain Gigan, Florian Marquardt, Aydogan Ozcan, Julie Grollier, Andrea J. Liu, Demetri Psaltis, Andrea Alù, Romain Fleury

25

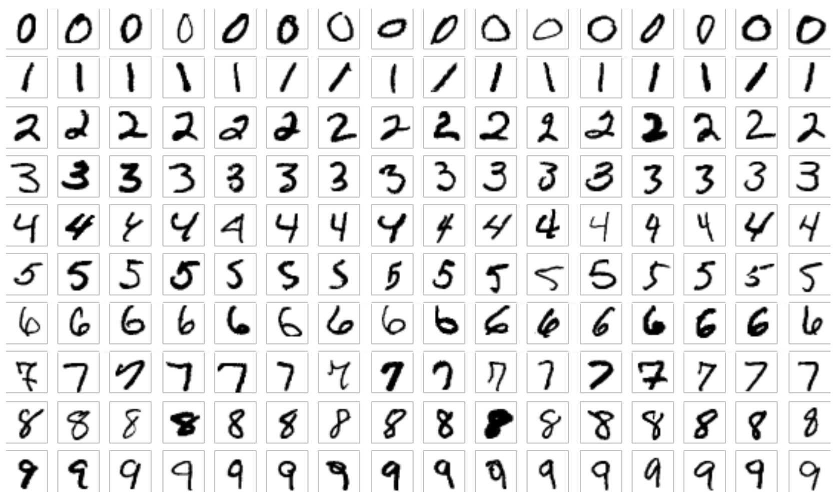
Use of local learning rules

Switch gears from developing rules to using them…

26

A classic problem for benchmarking

- Classify handwritten digits: MNIST database of 60,000 training images and 10,000 testing images (28x28 grayscale, https://yann.lecun.com/exdb/mnist/)

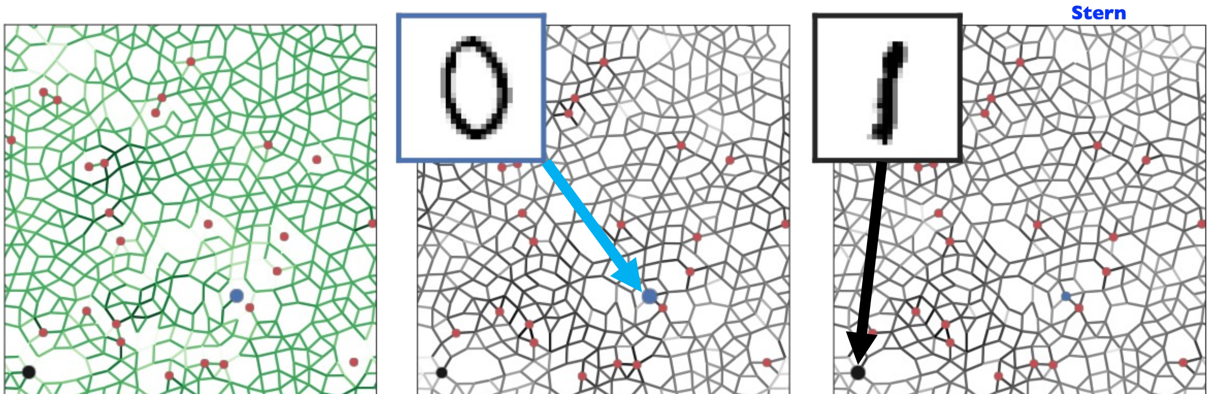


27

Coupled Learning in silico

- Successfully Classify zeros and ones from MNIST (5% test error)
  - Input nodes (red): top 25 principal components of images
  - Output nodes (blue, black): larger value indicates digit (0,1 respectively)

Menachem Stern



[Stern, Hexner, Rocks, Liu, Phys Rev X 2021]

28

Coupled Learning in laboratorium?

- Successful *in silico* demonstration is exciting & impressive, but…
  - requires CPU and memory storage both for training and for forward computation by solution of Kirchhoff's laws
- Really, want laboratory implementation that does not require CPU or memory storage during training or afterwards for "forward/inference" computation
  - In-memory analog training for in-memory analog compute…
  - Tremendous scaling advantage for large networks & complex tasks…
    - Tune in tomorrow!

29