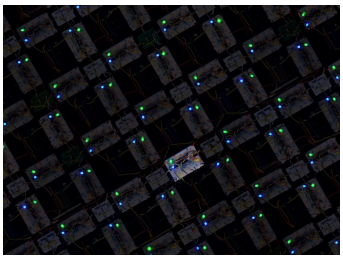
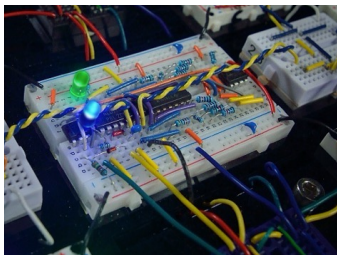
Autonomous Learning Metamaterials

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,…

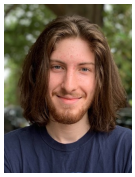
- Boulder School 2024
  - Lecture 1: learning systems and rules
  - Lecture 2: electronic realizations
  - Lecture 3: mechanical realizations

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

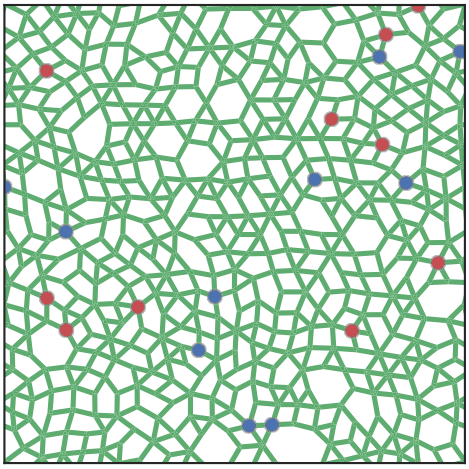
NSF UPenn MRSEC

2

Contrastive Local Learning Networks

- Can we build edges where all the conductances independently self-adjust using the local coupled learning rule so that **output** voltages evolve to the desired function of **input** voltages?

- **inputs**
- **outputs**



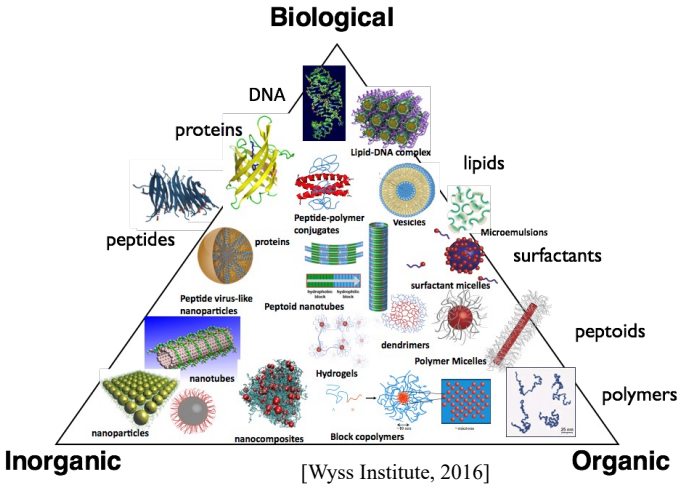
Example tasks/functions:
(1) Outputs are at $\{V_j\}$ when inputs are at $\{V_i\}$
(2) Outputs are a desired linear combo of inputs
(3) Classification: e.g inputs are grayscale pixel data and outputs indicate whether the image is a dog or a cat

Bottom-up learning of complex functionality using local rules in a fully recurrent network

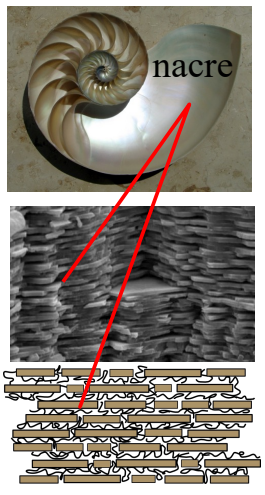
3

Bottom-up/self-assembly of structure

- Nature has built complex structures without a blueprint or oversight for eons
- Soft/living matter scientists/engineers have done this for decades



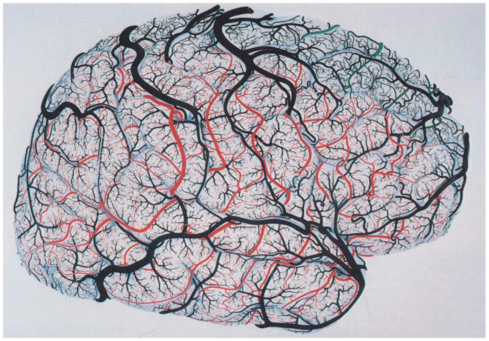
[Wyss Institute, 2016]

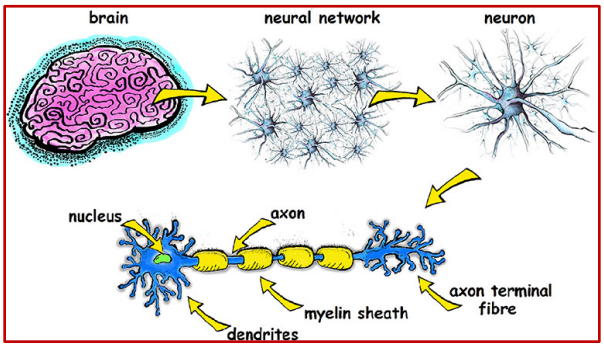


4

Bottom-up/self-learning of function

- Only prior examples are networks found in nature, and they use local rules
  - Eg vascular network in brain: conductivities self-adjust to redirect blood flow where needed
  - Eg neural network in brain: connections & conductances self-adjust to learn





Despite being slow and noisy, brains are far more capable and energy-efficient than artificial neural networks, which are trained top-down using external CPU and memory.

5

Coupled Learning Rule - recap

- Traditional loss 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, $P^{clamped} - P^{free}$ {>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 & designed to be implemented in the lab

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

6

Physical Implementation:

- Digital variable resistors are readily available, eg:
  - Click up/down by voltage pulse to proper pin

$$R = \delta R \times \{1, 2, 3, \ldots, or\ 128\}$$

  - If resistance rather than conductance is the learning degree of freedom, then the coupled learning rule follows from clamped-free difference of sum of P = IV = V^2/R as

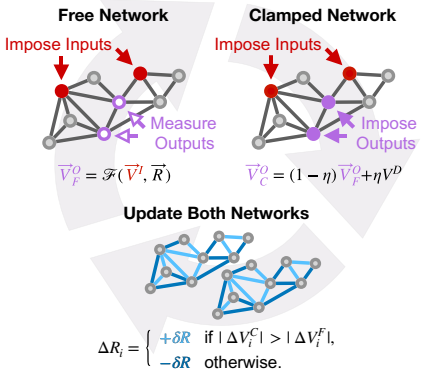
$$\dot{R}_i = -\gamma \frac{\partial}{\partial R_i}\left(\sum \frac{V_{jc}^2}{R_j} - \frac{V_{jf}^2}{R_j}\right) \cong \frac{\gamma\eta}{R_i^2}\left(V_{ic}^2 - V_{if}^2\right)$$

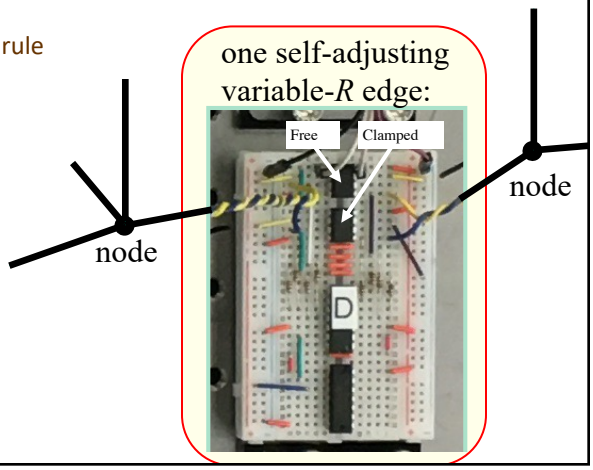
  - Easy enough to buy and connect into a network with chosen architecture
- But how to compare voltage drops of resistor i in free and clamped states and update R_i ***without memory or central processor?***
  - My answer: build identical twins…

7

Twin Network Trick

- Build repeat units (edges) consisting of two variable resistors kept at same resistances, with circuitry to change their value in unison on clock signal
- Connect the edges together in some chosen architecture, giving twin networks
  - One runs free BCs, the other runs clamped BCs
  - Circuitry performs an approximate coupled learning rule



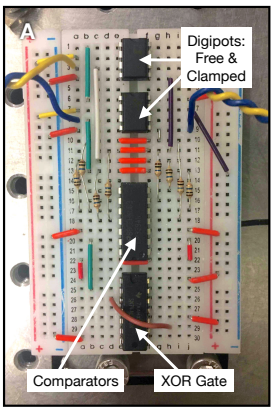


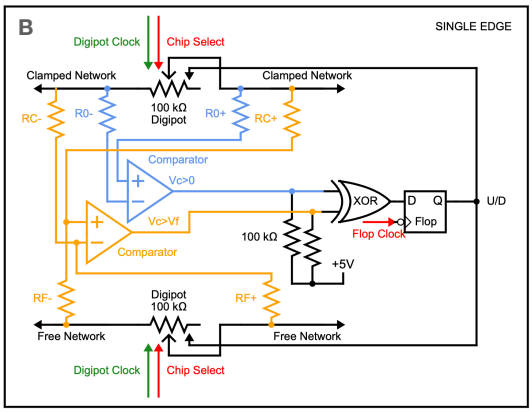
8

Edge circuitry

- On clock signal, compare voltages and move resistance up or down one δR unit

  approximate implementation of the coupled learning rule





Sam Dillavou

$$\Delta R_i = \frac{\gamma}{R_i^2}\left([\Delta V_i^C]^2 - [\Delta V_i^F]^2\right) \longrightarrow \begin{cases} +\delta R & \text{if } |\Delta V_i^C| > |\Delta V_i^F| \\ -\delta R & \text{otherwise.} \end{cases}$$

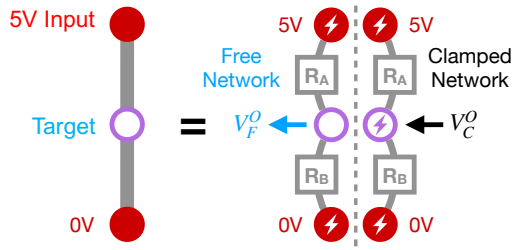
9

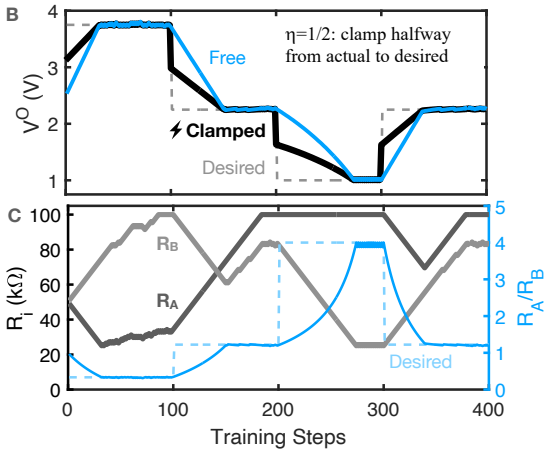
Warm-Up with a Voltage Divider

- Toy 2-edge network with continuum of known of solutions ($R_A/R_B = V_I/V_O - 1$)

  {over-parameterized per ML practice: more learning degrees of freedom (2) than constraints (1)}

  - Target output voltage is switched every 100 steps (from 3.75V to 2.5V to 1V to 2.5V)
  - Training to target is achieved in $\mathcal{O}(50)$ steps
  - After learning, bias gives drift at constant R_A/R_B



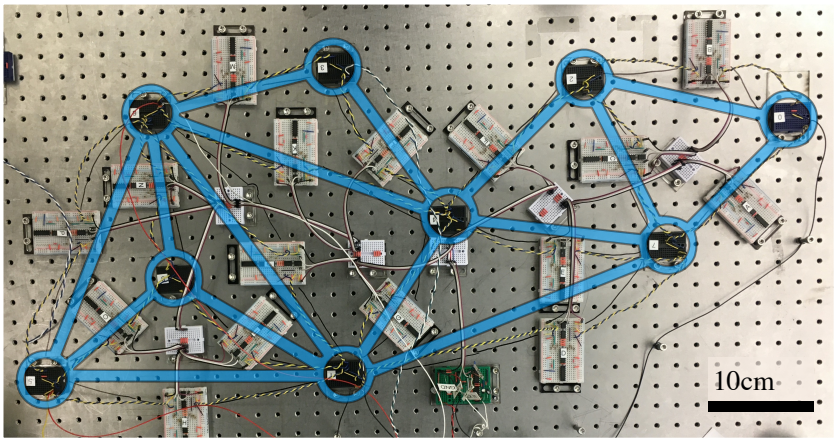


[Dillavou, Stern, Liu, Durian (Phys. Rev. Applied 2022)]

10

1st Contrastive Local Learning Network

- A 9-node electrical network of 16 identically-constructed repeat units (edges)



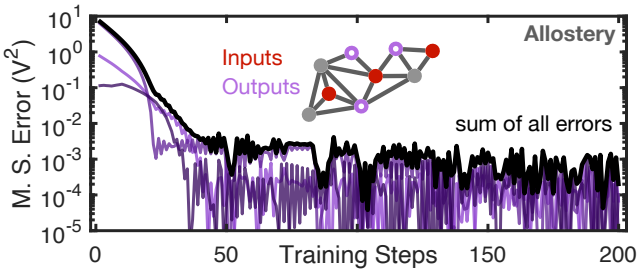
[Dillavou, Stern, Liu, Durian (Phys. Rev. Applied 2022)]

11

Bio- and math-inspired tasks

- "Allostery"
  - Train for three desired outputs to be 3V when three inputs are at 0V, 1V, 5V

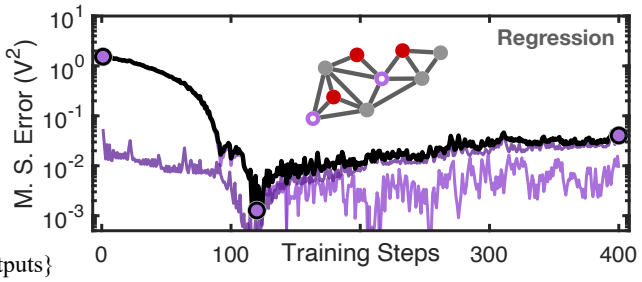
{training data = test data}



- Linear Regression
  - Train for two desired outputs to be a linear combination of two inputs

$$V_1^D = 0.15V_1^I + 0.20V_2^I$$
$$V_2^D = 0.25V_1^I + 0.10V_2^I$$

{training data: 400 inputs/outputs; test data: 20 inputs/outputs}



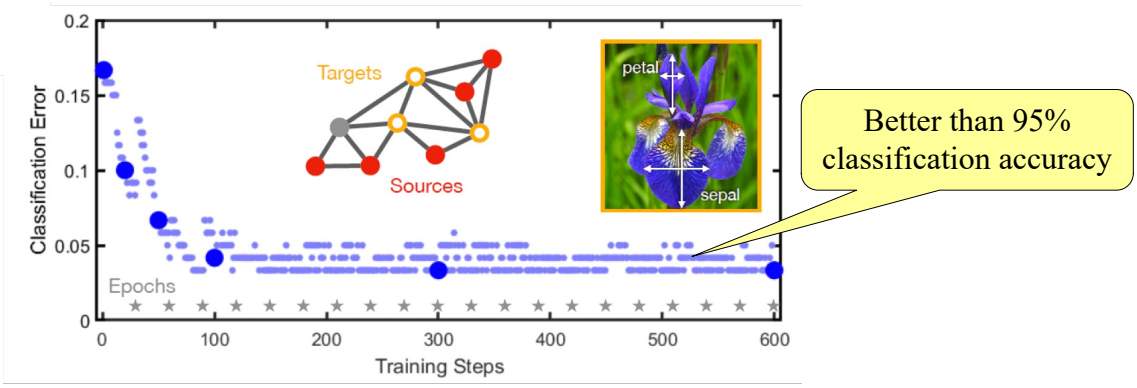
12

Classification of 3 Species of Iris

- Dataset: 4 measurements x 50 flowers x 3 species

  {10 of each species for training + 40 of each species for testing}
- Five input nodes: these four measurement plus one ground
- Three target nodes: L_2 norm of outputs from <input> for each species

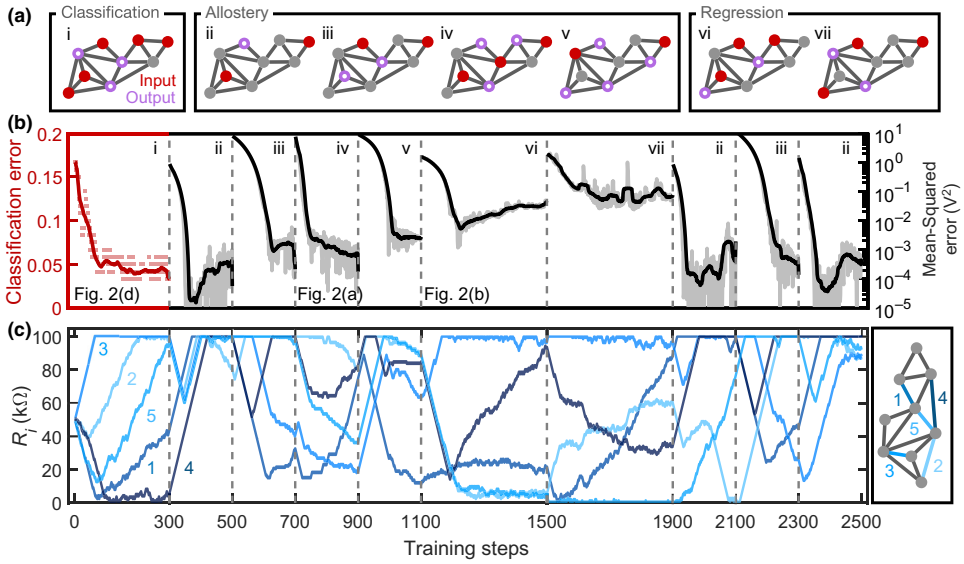
  {reset target every epoch = 30 training steps}



13

Flexible and retrainable

- One task after another, with different input/output node selections:



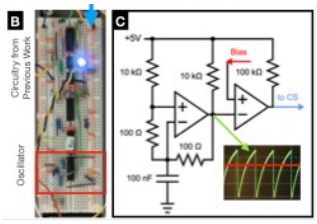
14

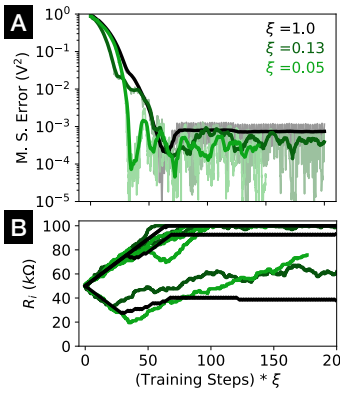
Variation: Asynchronous Learning

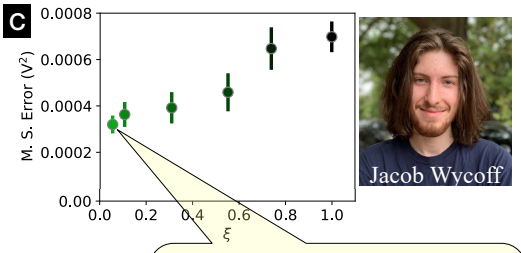
- Resistors in our original network adjust simultaneously, on clock signal

  Unlike neurons in brain

- Train by updating only a fraction ξ of edges, randomly chosen, each clock cycle:







Desynchronization improves performance! (cf stochastic grad. desc.)

[Wycoff, Dillavou, Stern, Liu, Durian (J Chem Phys 2022)]

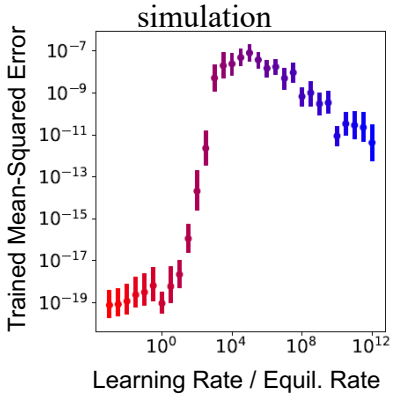
15

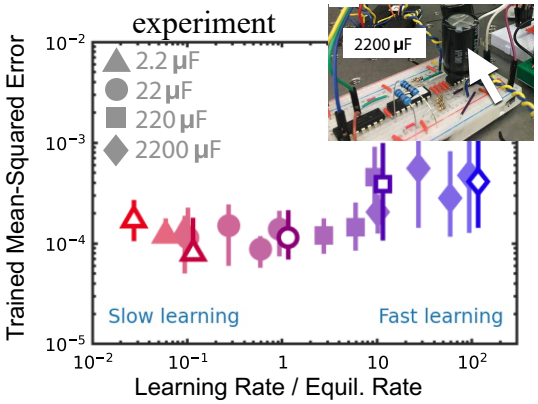
Variation: Out-of-Equilibrium Learning

- Currents in original network equilibrate much faster than resistances change

  This isn't always true in the brain (short term synaptic plasticity, fast behavior changes, macroscopic remodeling)

- We can train faster than network can equilibrate, without much accuracy loss:





[Stern, Dillavou, Miskin, Durian, Liu, (Phys Rev Res 2022)]

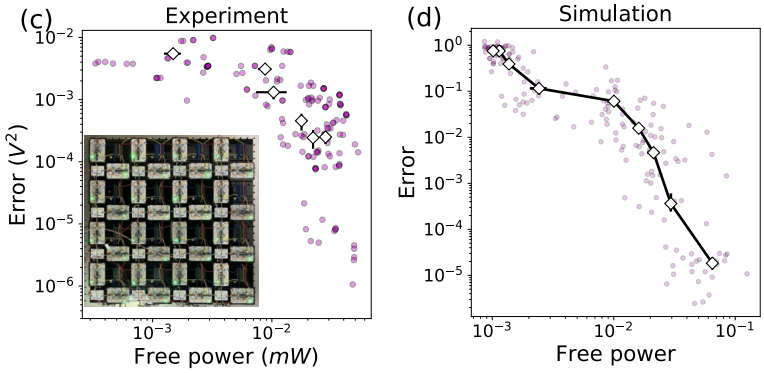
16

Variation: Power optimization

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

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

tradeoff: as power goes down, error goes up



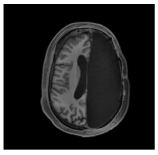
[Stern, Dillavou, Jayaraman, Durian, Liu (APL Machine Learning 2024)]

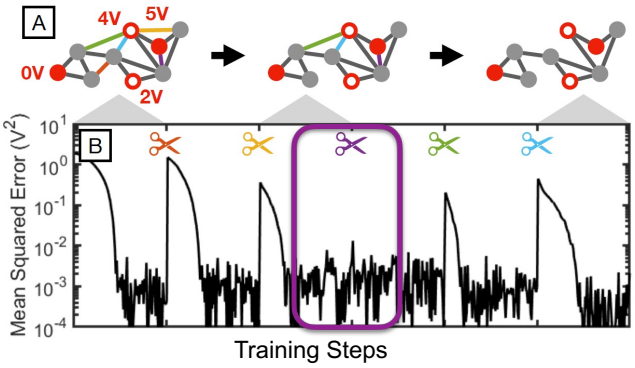
17

Important feature: Robust to damage

- Computers are fragile

- Brains are robust





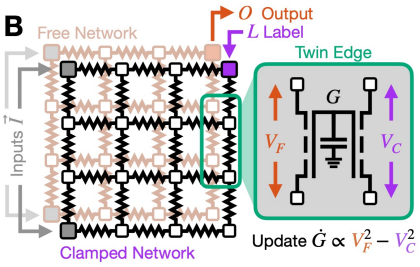
- So are our networks!
  - Easily retrained, like the brain, even after massive damage
  - Often/usually totally unaffected
  - ***Hence manufacturable at scale***, by contrast with other neuromorphic hardware

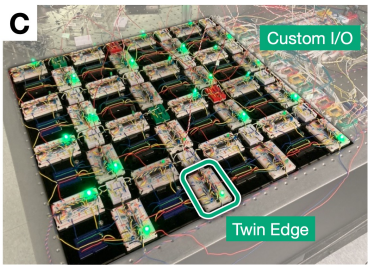
18

Continuously-variable conductances

- 2nd gen. made with transistors (gate voltage G is learning degree-of-freedom)

  {implements actual coupled learning rule}





32-edge 16-node network
(with or without periodic BCs)


Sam Dillavou


Benjamin Beyer


Marc Miskin

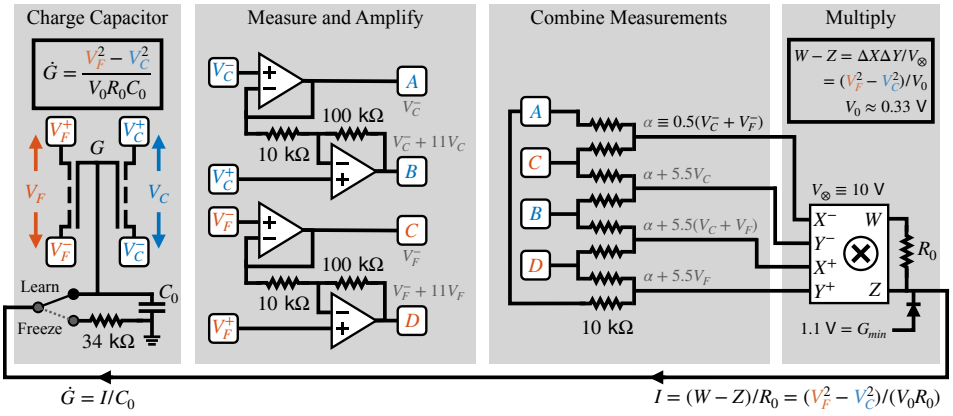
[Dillavou, Beyer, Liu, Miskin, Durian (PNAS 2024)]

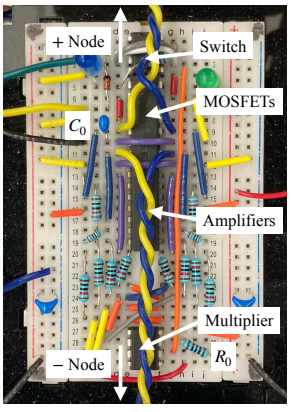
19

Edge Circuitry

- Run current to gate capacitor in proportion to ($V_F^2 - V_C^2$)

  Implements $\dot{k}_j \propto -\left[(V_j^2)^{clamped} - (V_j^2)^{free}\right]$

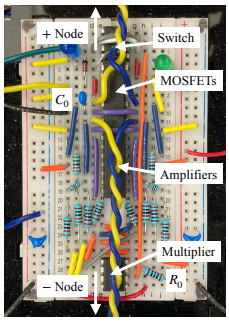


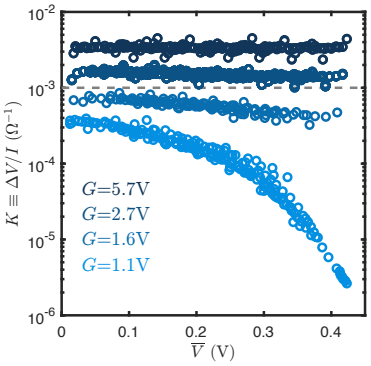


20

IV Characteristics

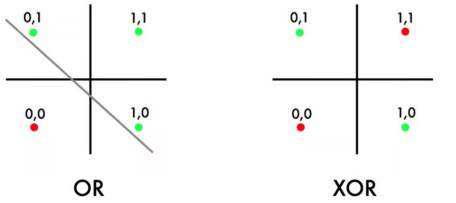
- Linear at high conductivity (high gate voltage), nonlinear at low conductivity:





Nonlinear → more functionality?

e.g it can learn XOR?

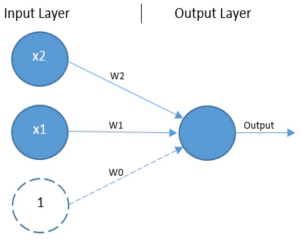


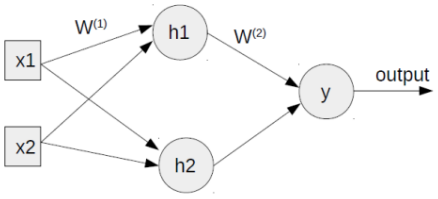
- NB: XOR is a hard task!
  - As computation: outputs are not a linear combination of inputs
  - As classification: cannot draw a through input space to separate the TRUE=1 from FALSE=0 labels

21

Aside on XOR

- Minsky & Papert (1969) proved that perceptrons (two-layer feedforward ANNs) cannot solve the XOR problem. This was the first of many failures that began an "AI winter" lasting 30+ years.
  - The trick is to add "depth" in terms of hidden layers between inputs and outputs
  - Source of nonlinearity is "activation function" such as {Heaviside, sigmoid, tanh, ReLU,…} acting on weighted sums, e.g. h1 = σ(x1 W1 + x2 W2)
  - Minimum ANN for XOR is 2x2x1, although this isn't overparameterized enough to easily find weights by gradient descent





22

Further aside…

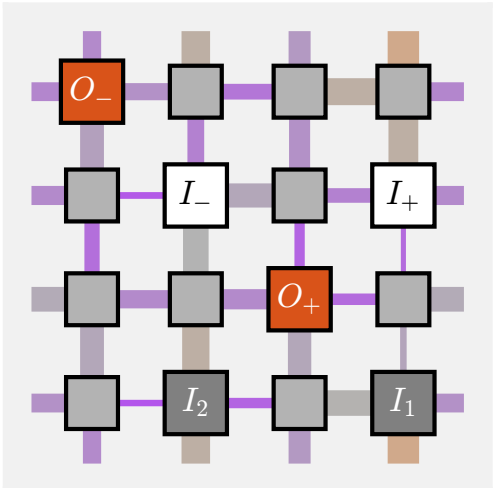
- Overparameterization is generally needed for gradient descent to avoid getting stuck in local minima
- Regularization is needed to prevent overfitting
- Double-descent phenomenon [Rocks & Mehta (Phys. Rev. Res. 2022)]
  - Optimization of the learning circuits will involve analogous phenomena…
  - Expect interesting/complex behavior in the dynamics of learning…

23

Network architecture & programming

- Square lattice with 32 edges, 16 nodes
- Periodic boundary conditions 😊

Fixed $\{I_-=0.11V, I_+=0.33V\}$
Variable inputs $\{I_1, I_2\}$
Differential output $O = O_+ - O_-$



24

XOR can be learned:

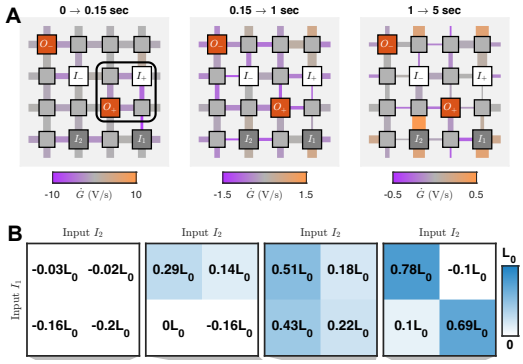
- Takes a while to "figure out" how to take advantage of edge nonlinearities

Fixed {I_-=0.11V, I_+=0.33V}
Variable inputs {I_1, I_2}
Differential output O = O_+-O_-

	I_1 = 0 V	I_1 = I_0
I_2 = I_0	$L = L_0$	L = 0 V
I_2 = 0 V	L = 0 V	$L = L_0$

Table 1. XOR Training Data / Truth Table

Here $I_0 = 0.45$ V, and $L_0 = -0.087$ V



NB: simpler / lower-order modes are learned first

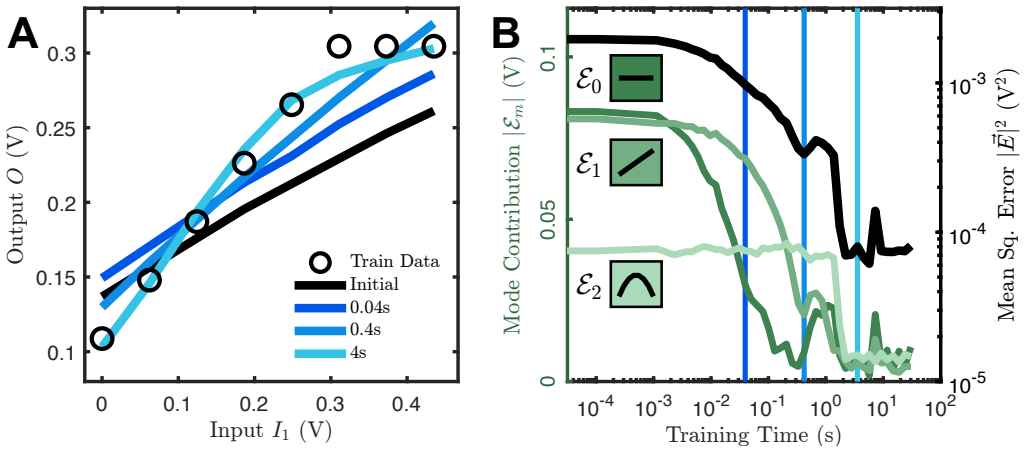
[Dillavou, Beyer, Liu, Miskin, Durian (PNAS 2024)]

25

Nonlinear regression & learning dynamics

Same architecture and input/output choices

- Learns in order of complexity: first the mean, then the slope, then the curvature

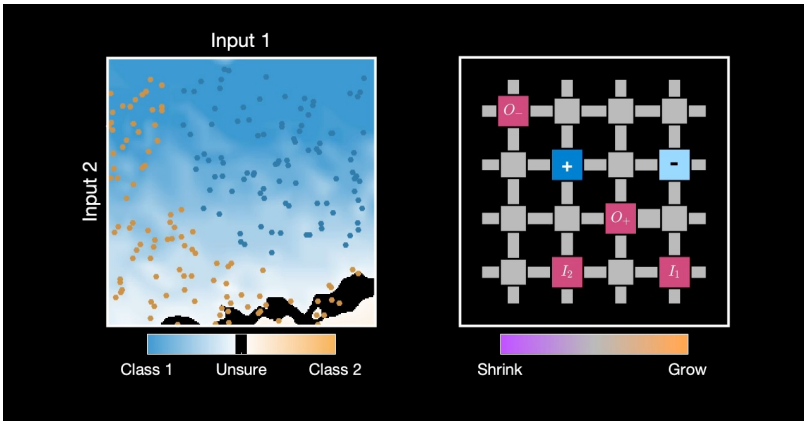


[Dillavou, Beyer, Liu, Miskin, Durian (PNAS 2024)]

26

Nonlinear classification & learning dynamics

- 2 analog inputs (I_1, I_2); 2 fixed voltages (+,-); 1 analog output ($O = O_+ - O_-$)
- Training data: dots
- Classification: background color (decision boundary in black, O = 0 volts)



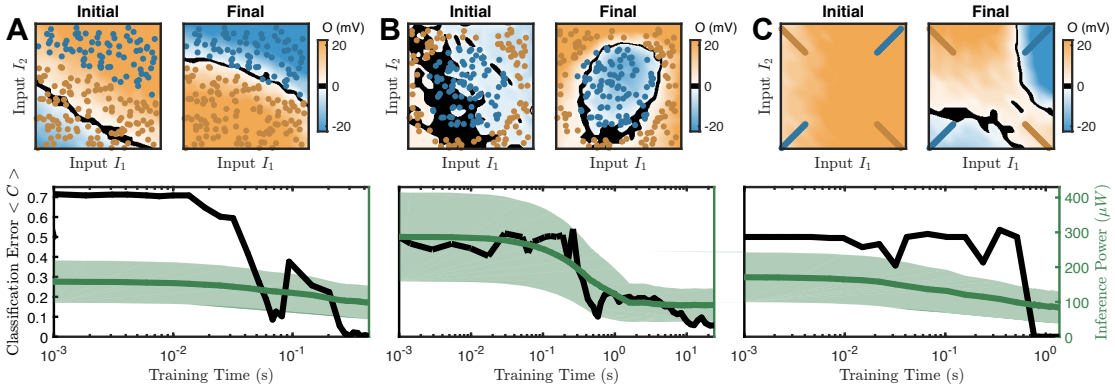
Square network with PBCs

thickness = conductivity
color = Δconductivity

[Dillavou, Beyer, Liu, Miskin, Durian (PNAS 2024)]

27

More nonlinear classification tasks



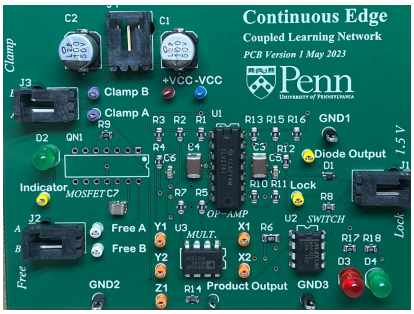
NB: Example (C) is XOR

28

Progression to large-scale networks

1. Breadboarding of analog / 2nd generation (done)

2. Printed circuit boards with 2nd generation analog circuitry
   - A few boards have been received and tested



Lauren Altman

Marc Miskin

& Tarunyaa Sivakumar

3. Make O(200)-edge networks
4. If all is well, microfabricate O(10^7)-edge networks on chips…

29

A new hardware compute platform?

- Our dreams:
  - massively scalable:
    - 10^9 edges per chip, all updating in parallel
    - cf 10^{12} parameters in large-language models
  - fast and energy efficient:
    - 10^{-18} J/parameter
    - cf most efficient Qualcomm AI chip: 10^{-12} J/parameter
    - cf gen.2 on breadboards: 10^{-11} J/parameter
- Eventually disrupt current artificial neural network paradigm for AI?
- Or at least find niche applications where the energy cost must be minimized, or the task be must adapted to changing conditions?
  - eg edge computing in sensors,…

30

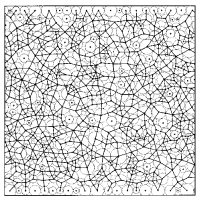
Mechanical networks?

- Low Re fluidic networks are isomorphic to linear resistor networks
  - Voltage ⟷ pressure, current ⟷ current, conductance ⟷ conductance
- Tune in to Lecture 3 for contrastive local learning spring networks
  - harder for theory/simulation than electronics
    - Voltage and current are scalars, but force and displacement are vectors
    - Force balance is a nonlinear problem even with linear/Hookean springs

31

For me, this harkens back to jamming

- Network of contacts and forces in particulate systems (grains, bubbles, cells…)

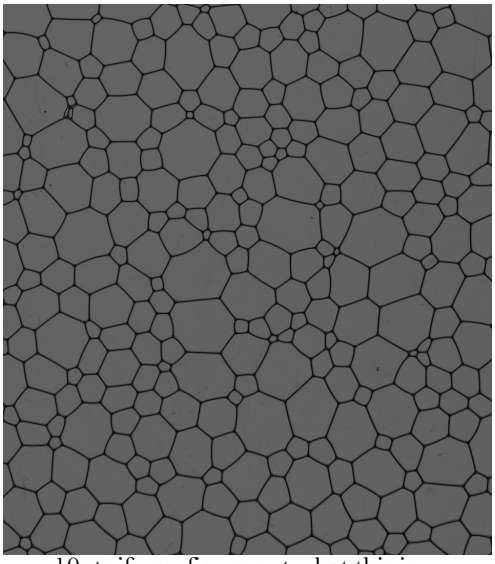


DJD 1995+…
& Boulder School 2002

- An active mechanical system from my lab:

  Note the convergent flow towards center

Anthony Chieco



10pts if you figure out what this is

32