Flow networks as learning machines

Menachem Stern*
Department of Physics and Astronomy
University of Pennsylvania

Daniel Hexner
Department of Mechanical Engineering
Technion, Israel

Jason W. Rocks
Department of Physics
Boston University

Andrea J. Liu
Department of Physics and Astronomy
University of Pennsylvania

Abstract

Materials and machines are often designed with particular goals in mind, so that they exhibit desired responses to given forces. Here we explore an alternative approach, physical *coupled learning*. In this paradigm, a physical network such as a flow network consisting of pipes conveying fluid, adapts the pipe conductances to obtain a desired pressure response at target nodes in response to pressures applied at source nodes, similar to supervised machine learning. Instead of obtaining the desired pressure response by minimizing some loss function, coupled learning introduces *physically plausible* learning rules. By physically plausible we mean (1) that the conductance of each pipe responds only to properties of the flow through that pipe, such as the pressure drop, and that (2) the learning rule itself should not contain explicit information about the desired response. We demonstrate how disordered flow networks can learn to distinguish handwritten digits. These results suggest the feasibility of a new class of smart metamaterials that can adapt in-situ to users' needs.

1 Introduction

Engineered systems and materials are typically designed for particular properties or functions [1]. The design process often involves numerous trial and error iterations, where the system is repeatedly tested for the desired functionality [2], modified and then tested again. A second class of strategies is based on *learning*, where a system adjusts to display desired functionality given external signals (training examples). One class of methods, "global supervised learning," is ubiquitous for problems such as classification [3, 4]. These methods involve optimizing a global cost function that depends on all of the microscopic details of the system. Such methods have been used to design physical flow and elastic networks with desired functions such as allostery [5, 6]. In this physical context, such learning methods were dubbed *tuning*, since modifying the learning degrees of freedom (e.g. pipe conductances in flow networks) requires external intervention.

Natural systems (e.g. the brain) develop desired functions using a different learning framework that uses *local* learning rules. Such learning is *autonomous*, requiring no external designer for evaluation of the system and its subsequent modification. This learning approach is particularly relevant in physical networks such as flow networks. The microscopic elements of such networks cannot perform computations and do not encode information about the desired functionality *a priori*.

Here we consider training flow networks to achieve tailored responses of "target" nodes to external constraints applied at "source" nodes, e.g. when pressures corresponding to MNIST images are

*nachis@sas.upenn.edu

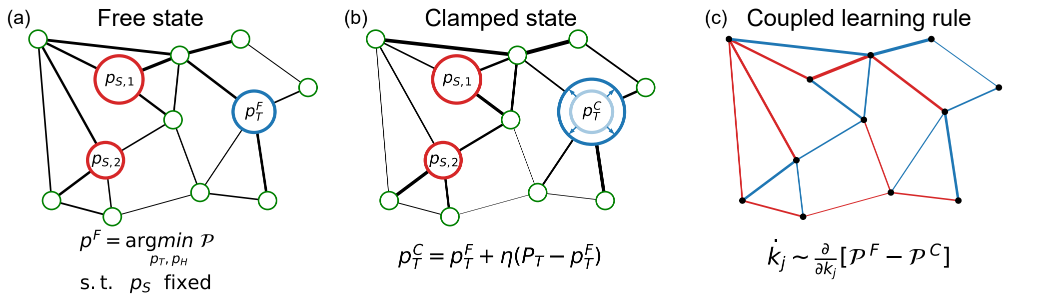

Figure 1: Coupled learning in flow networks. a) In the free state, node pressures are constrained such that source nodes (red) have specific pressure values $\boldsymbol{p}_S$. The target node pressures $\boldsymbol{p}_T$ and the dissipated power at all pipes $\mathcal{P}_j$ attain their steady state values due to the physical processes in the network. b) In the clamped phase, we further constrain the target node pressures to be slightly closer to their desired values compared to the free phase. c) A local learning rule modifying pipe conductance according to its flow response is proportional to the difference in dissipated power between the free and clamped states.

applied to source nodes, we require target nodes to have desired pressures corresponding to a specific digit.

We propose a general framework, "coupled supervised learning," for deriving the local learning rules for physical networks such as flow networks. These local learning rules specify how learning degrees of freedom (conductances of pipes) respond to local conditions (e.g. the current through each pipe). The proposed learning rules lead to results similar to those obtained by minimizing a cost function.

Coupled learning is inspired by advances in neuroscience and computer science [7, 8, 9, 10, 11]. The learning rules are based on the difference in response between two steady states of the system, one in which only source constraints are applied (*free state*), the other where source and target constraints are applied simultaneously (*clamped state*). We demonstrate how coupled learning works for disordered flow networks, and test our learning framework on a standard classification problem, distinguishing handwritten digits. Trained flow networks classify digits with high accuracy, on par with simple machine learning algorithms.

2 Coupled learning in flow networks

A flow network is defined by N nodes carrying pressure values p_μ. Nodes are connected by pipes characterized by conductances k_j; these conductances are the *learning degrees of freedom*, whose modification enables the network to adapt to a desired function. Assuming each pipe is directed from nodes μ_1 to μ_2, the current in the pipe is given by $I_j = k_j(p_{\mu_1} - p_{\mu_2}) \equiv k_j \Delta p_j$. If boundary conditions are applied to the network (e.g. fixed pressure values at some nodes) the network finds a flow steady state, in which the total dissipated power $\mathcal{P} = 0.5 \sum_j k_j \Delta p_j^2$ is minimized by varying the pressures at unconstrained nodes.

We now define a generic learning task. Divide the node pressures $\{p_\mu\}$ into three types, source nodes p_S, target nodes p_T, and "hidden" nodes p_H. The target node pressures are to have a set of values $\{P_T\}$ when the source node pressures are constrained to $\{P_S\}$. A generic disordered flow network does not possess this function, so design strategies are needed to find values for the pipe conductances $\{k_j\}$ that achieve the desired task.

For the network to learn, we allow the pipe conductance values $\{k_j\}$ to vary depending on the physical state of the network $\{p_\mu\}$. We focus on local learning rules, where k_j in each pipe j only changes in response to the current in that pipe. Following ideas from Hebbian contrastive learning [7] and equilibrium propagation [10], we define two special network states. In the *free* state p_μ^F only the source nodes p_S are constrained to their values P_S, allowing p_T, p_H to obtain their steady state (Fig. 1a). The *clamped* state p_μ^C is the state where both the source and target node pressures p_S, p_T are constrained to P_S and p_T^C, respectively, so that only the remaining (hidden) nodes p_H are allowed to change to find a steady state. The values of the dissipated power in the resulting steady states are denoted $\mathcal{P}^F$ and $\mathcal{P}^C$ for the free and clamped states, respectively.

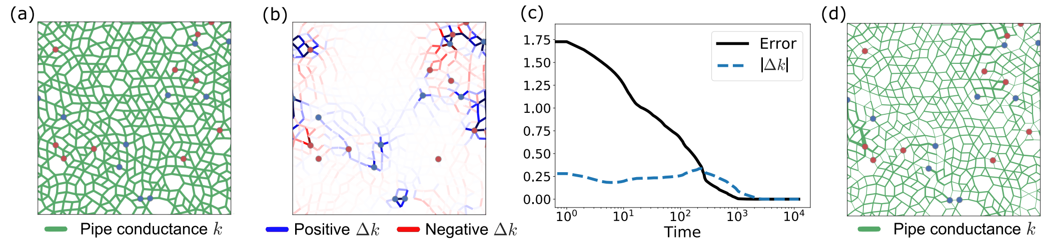

Figure 2: Training a flow network with coupled learning. (a) Untrained disordered flow network ($N = 512$ nodes) with uniform pipe conductance $k_i = 1$ (uniform edge thickness). Red and blue dots correspond to source and target nodes with dot sizes indicating the magnitudes of the source pressures $\{P_S\}$ and *desired* target pressures $\{P_T\}$. (b) In each step of the learning process, the conductance values are modified using Eq. 2, according to the difference in flow between the free and clamped states. This process is applied iteratively. (c) During training of a network, the pressure values of the target nodes approach the desired values, as indicated by the shrinking error (solid line); as the error diminishes, conductance modification Δk (dashed line) vanishes as well. (d) After training, the network conductance values are considerably changed compared to the initial network shown in (a).

We introduce a trainer (supervisor) that nudges the target node pressures slightly away from their free state p_T^F values by clamping them at

$$p_T^C = p_T^F + \eta[P_T - p_T^F] \tag{1}$$

where $\eta \ll 1$ (Fig. 1b). The trainer imposes pressures on the target nodes that are a small step closer to the desired response P_T. We then propose a learning rule for the pipe conductance values (Fig. 1c):

$$\dot{k}_j = \alpha\eta^{-1}\frac{\partial}{\partial k_j}\{\mathcal{P}^F - \mathcal{P}^C\} = \frac{1}{2}\alpha\eta^{-1}\{[\Delta p_j^F]^2 - [\Delta p_j^C]^2\}, \tag{2}$$

where α is a scalar *learning rate*. Note that the derivative of the physically minimized function (power dissipation $\mathcal{P}$) is taken with respect to the *learning* degrees of freedom $\{k_j\}$. The simplest implementation of coupled learning is to iteratively apply Eq. 2. We focus on learning in the quasi-static limit where we completely relax the node pressures to their steady state at each iteration.

The learning rule of Eq. 2 is manifestly local, as the conductance of a pipe k_j changes only due to the flow at that pipe. Such local learning rules may conceivably be implemented in physical pipes for which the conductance (radius) of the pipe is controlled by the current in it. Furthermore, the network is not required to encode information about the desired function *a priori*. This information is supplied by the actions of the external supervisor, slightly nudging the target node pressures towards the desired state at every iteration. These properties of coupled learning stand in contrast to tuning algorithms based on optimization of global cost functions.

We test coupled learning on disordered flow networks (size $N = 512$). A network is initialized with uniform pipe conductance $k_j = 1$ (Fig. 2(a)). We pick 10 nodes randomly as source nodes and apply source pressures P_S. Source pressures are sampled from a Gaussian distribution $\mathcal{N}(0, 1)$. We also choose 10 target nodes, with desired pressures at target nodes, $\{P_T\}$, sampled from a Gaussian distribution, $\mathcal{N}(0, 0.2\sum P_S)$. We compute the dissipated power $\{\mathcal{P}_j^F\}$ for each edge j of the network in the free state by solving the linear set of equations corresponding to Kirchhoff's law at every node [6]. To compare the output p_T^F of the network to the desired response P_T, we use the standard mean squared error function $C = \frac{1}{2}\sum_T [p_T^F - P_T]^2$.

In the clamped state we nudge the target nodes towards their desired value (Eq. 1, $\eta = 10^{-3}$) and similarly compute the dissipated power $\{\mathcal{P}_j^C\}$. After obtaining the clamped state, we update the conductance values (Eq. 2) with $\alpha = 5 \cdot 10^{-4}$. Fig. 2(b) shows the change of conductance for each edge at the first iteration of learning. This constitutes one step of the training process; at the end of each step, we compute the error function C (Fig. 2(c)). The difference between the obtained targets and the desired ones decreases by orders of magnitude during training, reaching machine precision. We find that the magnitude of changes in the conductance vector, $|\Delta k|$, also decreases towards zero during training. This shows that the learning process is adaptive–it slows down as it approaches

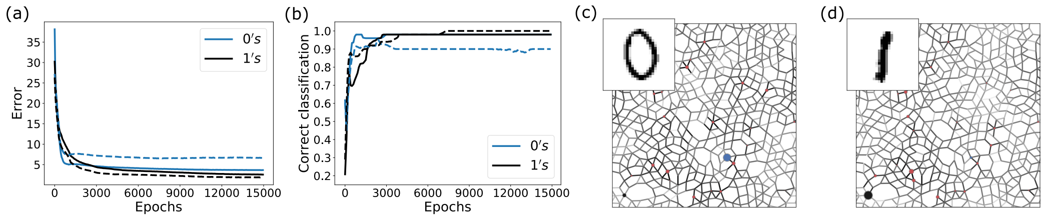

Figure 3: Classification of MNIST digits (0's & 1's). a) Cost values for the digits 0 (blue) and 1 (black) pressures. Training error is indicated by full lines and test error in dashed lines. b) Accuracy of classification (fraction of correct predictions) for digits 0 and 1. c-d) Response of the network when presented with the image on the top-left. Flow power is indicated by edge brightness, while target node pressures are indicated with blue (0) and black (1) dots. When a 0 image is shown, the 0 target node has high pressure and the 1 target node has low pressure (and vice versa).

good solutions. The final trained network is displayed in Fig. 2(d), with edge thicknesses indicating conductance. The pipes of the trained network have changed considerably compared to the initial one shown in Fig. 2(a), with some pipes effectively removed (conductances near zero).

3 Supervised classification with flow networks

We train flow networks to distinguish between images of handwritten digits (0 and 1). We pick 50 images of each digit from the MNIST database [12] as a training set, and 50 images for test sets. Instead of using pixel values, we carry out a Principal Component Analysis of all MNIST images, and train the network with the top 25 principal components of the training set images. Source pressures are given by these principal components of the training images. We pick 2 target nodes, one for each digit. The network is trained so that when an image of '0' is shown, the associated '0' target node has high pressure ($p_{'0'} = 1$), and the '1' node has no pressure ($p_{'1'} = 0$). The target pressures are reversed when an example of the digit '1' is chosen. At each iteration, the network is presented with a single image-label pair, sampled at random from the training set. A training epoch is defined as the time required for the network to be presented with 100 training examples.

The error in pressure values is shown in Fig. 3(a) for each digit. Coupled learning reduces both training and test errors. For discrete digit predictions, we say that the predicted label is given by the target node having the larger pressure. We find that the classification accuracy of the network improves dramatically during training (Fig. 3b), with training (test) accuracy reaching 98% (95%). A logistic regression model trained on the same data yielded a training accuracy of 100% and a test accuracy of 98%. Note that we did not tune hyperparameters to achieve the listed accuracies; such tuning might well improve the performance of our algorithm. In Fig. 3c-d, network response is shown for two select input images. When the source nodes are constrained with pressure values corresponding to an image of a '0', the '0' target node has high pressure (blue) while the '1' target node has low pressure. The opposite response occurs when an image of '1' is applied as input.

4 Concluding remarks

In this work we introduced coupled learning, a class of learning rules born of contrastive Hebbian learning and equilibrium propagation [7, 10], and applied it to flow networks. Coupled learning rules are physically plausible; they can be implemented in realistic materials and networks, allowing them to autonomously learn from external signals.

Such learning machines may be trained in-situ, so that it is not necessary to know the network geometry or topology, or even any information about the physical or learning degrees of freedom. This approach should be particularly valuable for experimental systems, which are difficult to characterize in full microscopic detail. As long as the proper learning rules are implemented, the systems can be trained simply by applying the proper boundary conditions. As an additional benefit, this learning approach is scalable–it can be applied to arbitrarily large systems. We hope the results shown here will inspire further research of new classes of physical learning machines, capable of autonomously adapting to perform new tasks and able to generalize to diverse inputs.

While revising our paper, we found that a similar approach for the case of flow/resistor networks has been introduced in Ref. [13].

Broader Impact

The confluence of ideas from neuroscience and machine learning has contributed immensely to our fundamental understanding of the nature of learning. We study whether these ideas can be exported to real world physical networks, to construct “learning machines", able to autonomously adapt to external influence in order to gain desired functionality. Such machines are essentially rudimentary brains, composed of simple mechanical parts. The simplicity of these machines may allow us to study fundamental questions about learning, such as basic trade-offs that underlie physical learning algorithms. Moreover, if physical learning machines are realized, they can be trained in-situ by users, rather than a specialist designer. Adaptable learning machines are thus expected to be particularly useful if desired physical tasks are either not known in advance, or defined by examples of use.

Acknowledgments and Disclosure of Funding

This research was supported by the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Materials Sciences and Engineering under Awards DE- FG02-05ER46199 and DE-SC0020963 (MS), and the Simons Foundation for the collaboration “Cracking the Glass Problem” award #454945 to AJL, as well as Investigator award #327939 (AJL). DH wished to thank the Israel Science Foundation (grant 2385/20).

References

[1] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[2] Stefan H Thomke. Managing experimentation in the design of new products. *Management Science*, 44(6):743–762, 1998.

[3] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[4] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre GR Day, Clint Richardson, Charles K Fisher, and David J Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports*, 2019.

[5] Jason W Rocks, Nidhi Pashine, Irmgard Bischofberger, Carl P Goodrich, Andrea J Liu, and Sidney R Nagel. Designing allostery-inspired response in mechanical networks. *Proceedings of the National Academy of Sciences*, 114(10):2520–2525, 2017.

[6] Jason W Rocks, Henrik Ronellenfitsch, Andrea J Liu, Sidney R Nagel, and Eleni Katifori. Limits of multifunctionality in tunable networks. *Proceedings of the National Academy of Sciences*, 116(7):2506–2511, 2019.

[7] Javier R Movellan. Contrastive hebbian learning in the continuous hopfield model. In *Connectionist models*, pages 10–17. Elsevier, 1991.

[8] Pierre Baldi and Peter Sadowski. A theory of local learning, the learning channel, and the optimality of backpropagation. *Neural Networks*, 83:51–74, 2016.

[9] Yoshua Bengio and Asja Fischer. Early inference in energy-based models approximates back-propagation. *arXiv preprint arXiv:1510.02777*, 2015.

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

[11] Sergey Bartunov, Adam Santoro, Blake Richards, Luke Marris, Geoffrey E Hinton, and Timothy Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Advances in Neural Information Processing Systems*, pages 9368–9378, 2018.

[12] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[13] Jack Kendall, Ross Pantone, Kalpana Manickavasagam, Yoshua Bengio, and Benjamin Scellier. Training end-to-end analog neural networks with equilibrium propagation. *arXiv preprint arXiv:2006.01981*, 2020.