Putting a bug in machine learning: a moth brain learns to read MNIST

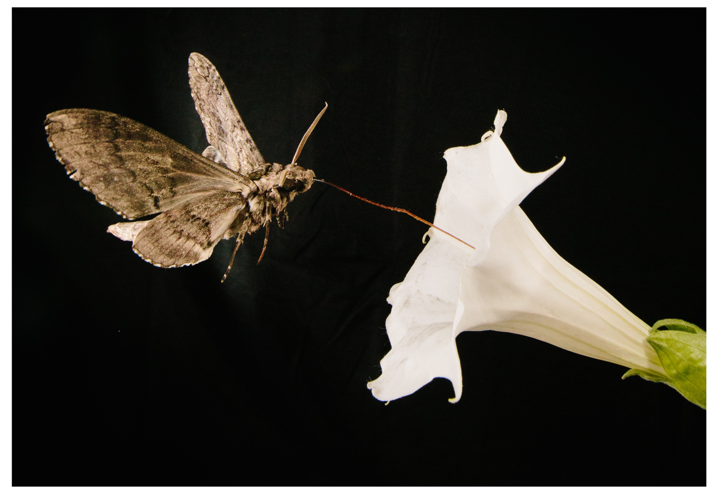
APPLIED MATHEMATICS
DEPARTMENT OF BIOLOGY
ELECTRICAL ENGINEERING
UNIVERSITY of WASHINGTON

Charles B. Delahunt [1] **Jeffrey A. Riffell** [2] **J. Nathan Kutz** [3]

[1] Electrical Engineering [2] Biology [3] Applied Mathematics *delahunt@uw.edu, jriffell@uw.edu, kutz@uw.edu*

Introduction

We seek to characterize the learning tools used in biological neural networks, in order to port them to the machine learning context. We focus on the regime of very few training samples.



The Moth Olfactory Network is among the simplest biological neural systems that can learn.

We developed a computational model of the Moth Olfactory Network, and set it to learn the MNIST digits. The moth brain learns to read given very few (1 to 10) training samples per class. In this regime the moth out-performs standard ML methods (Fig 5).

Our experiments elucidate biological mechanisms for fast learning that rely on competitive inhibition, sparsity, and Hebbian plasticity. These represent a novel, alternative toolkit for building neural nets.

Neural Architecture

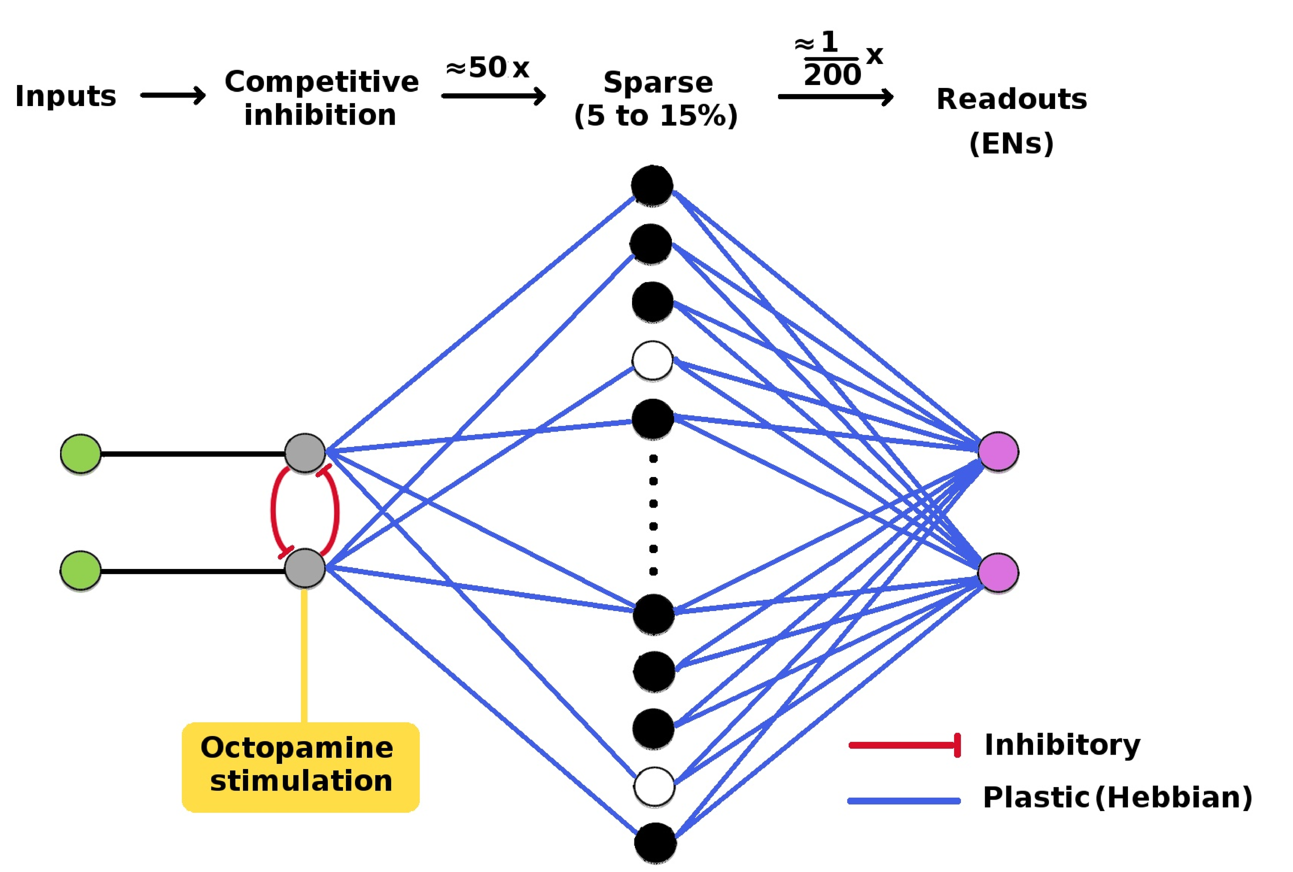


Figure 1: Inputs feed 1-to-1 into a pre-amp layer (Antenna Lobe, AL) with ~60 noisy units and intra-layer competitive inhibition. The AL feeds-forward with sparse connectivity into a high-dimensional (~4000 units) but sparsely-active layer (Mushroom Body, MB). The only plastic connections in the system are into and out of this sparse layer. Connection maps and weights are largely random. The MB feeds-forward to readout units (Extrinsic Neurons, ENs). Learning occurs when EN responses to different classes diversify.

Simulations

Our AL-MB network design follows the known biophysics. Neurons are integrate-and-fire units with coupled ODEs for firing rates (inset a, b) with an added term for octopamine. Plasticity is Hebbian, ie "fire together, wire together" (inset c). The firing rates are evolved in time as stochastic differential equations (inset b). The model was calibrated to *in vivo* firing rate data from moths exposed to odors and octopamine (ie learning).

a) $\tau \frac{df}{dt} = -f + s(\Sigma \mathbf{w}_i \mathbf{u}_i) = -f + s(\mathbf{w} \cdot \mathbf{u})$, where

$\mathbf{w}$ = connection weights;
$\mathbf{u}$ = upstream neuron FRs;
$s()$ is a sigmoid function or similar.

b) $\tau \frac{d\mathbf{P}}{dt} = -\mathbf{P} + \mathbf{s}(\widetilde{\mathbf{P}}) + \mathbf{dW}$ where

$\mathbf{W}(t)$ = brownian motion process;
$\widetilde{\mathbf{P}} = -(1 - \gamma M^{OP})*M^{LP}*\mathbf{u}^L + (1 + M^{OP})*M^{RP}*\mathbf{u}^R$;
M^{OP} = octopamine→PN weight matrix (diagonal);
M^{LP} = LN→PN weight matrix;
M^{RP} = RN→PN weight matrix (diagonal);
$\mathbf{u}^L$ = LN FRs;
$\mathbf{u}^R$ = RN FRs;
γ = scaling factor for effects on inhibition.

c) $\Delta w_{ab}(t) = \gamma f_a(t) f_b(t)$
$\Delta w_{ab}(t) = \delta w_{ab}(t)$, if $f_a(t) f_b(t) = 0$.

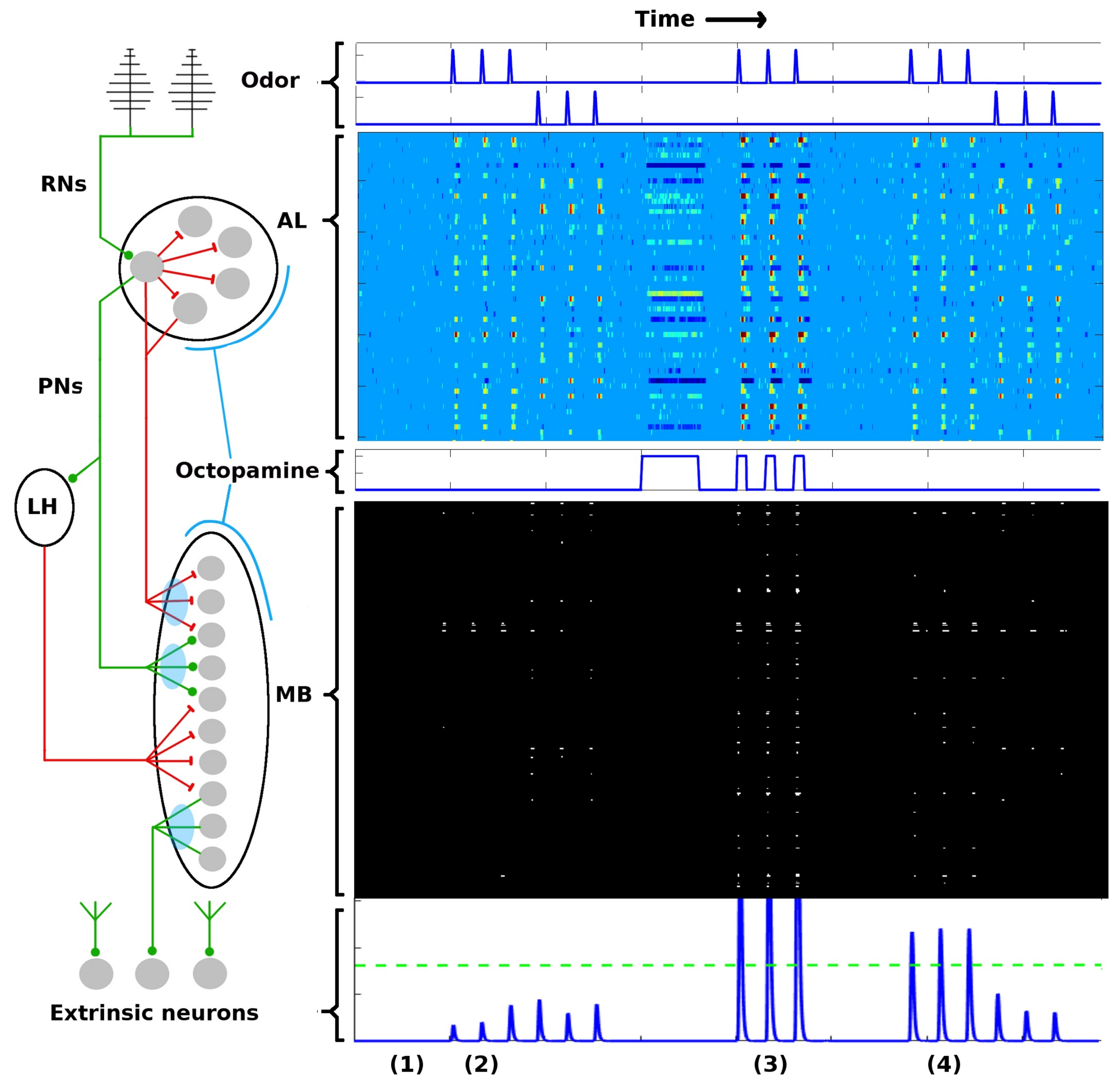


Figure 2: Neural firing rate heatmap of a learning simulation, showing neuron timecourses from each network, time axes aligned vertically. Timecourse events:
(1) No odor: All regions are silent. (2) Two odors are delivered, 3 doses each: AL, MB, and ENs display odor-specific responses. (3) Training on the first odor (with octopamine): All regions respond strongly. (4) The odors are re-applied: The AL returns to its pre-trained activity since it is not plastic. In contrast, the MB and EN are now more responsive to the trained odor, crossing an action threshold (green dotted line). Response to control odor is unchanged.

Learning

The actual moth brain handles about 60 features. We created an MNIST-like task by selecting a subset of pixels from downsampled images, giving 85 pixels-as-features (Fig 3).

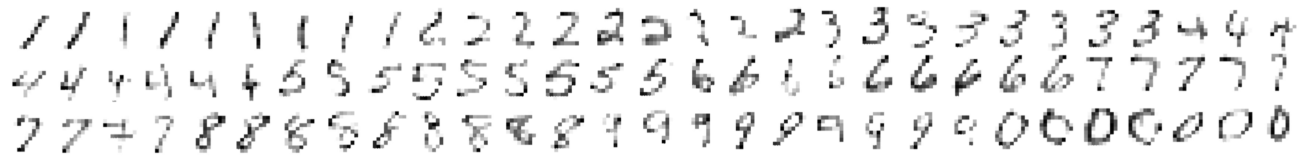


Figure 3: Digits from the downsampled MNIST dataset.

For learning, one EN is assigned to each digit class. Training consistently and robustly diversifies EN response (Fig 4). Moths randomly generated from template attain over 75% median accuracy given 10 training samples per class.

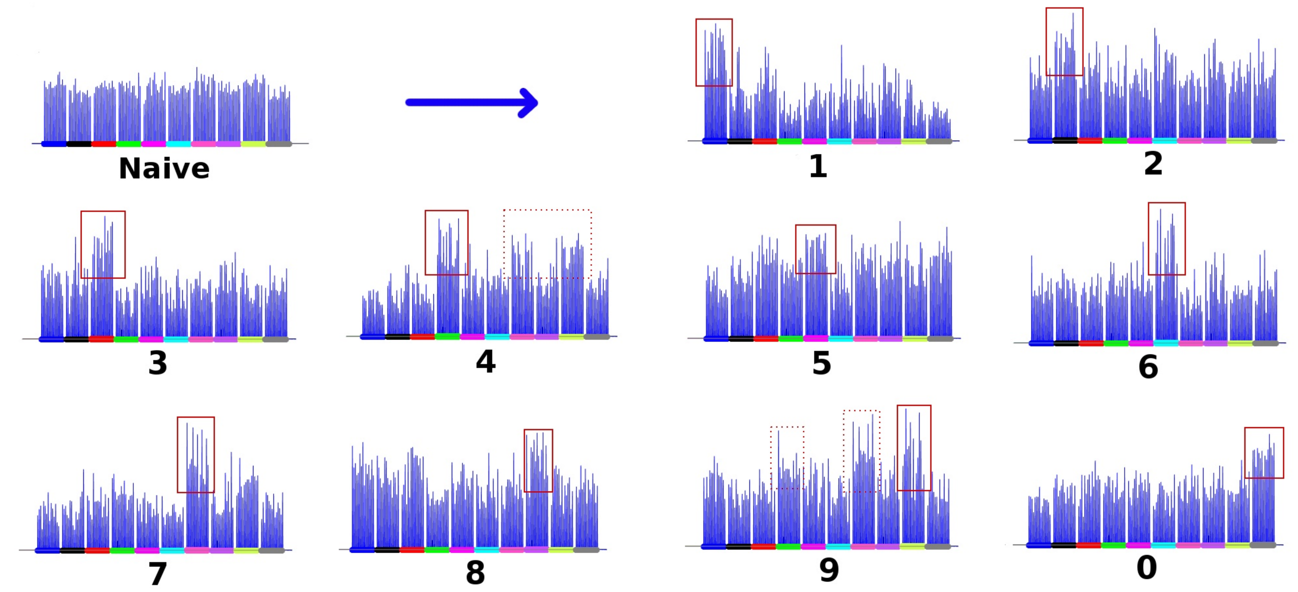


Figure 4: Pre- and post-training EN time courses (normalized) for a typical moth. Each timecourse shows an EN_i's response to 150 digits (15 ones, then 15 twos, etc). At top left is a naive response (all ENs similar). Other subplots show the trained ENs (targeted classes are framed in red).

Bug vs machine

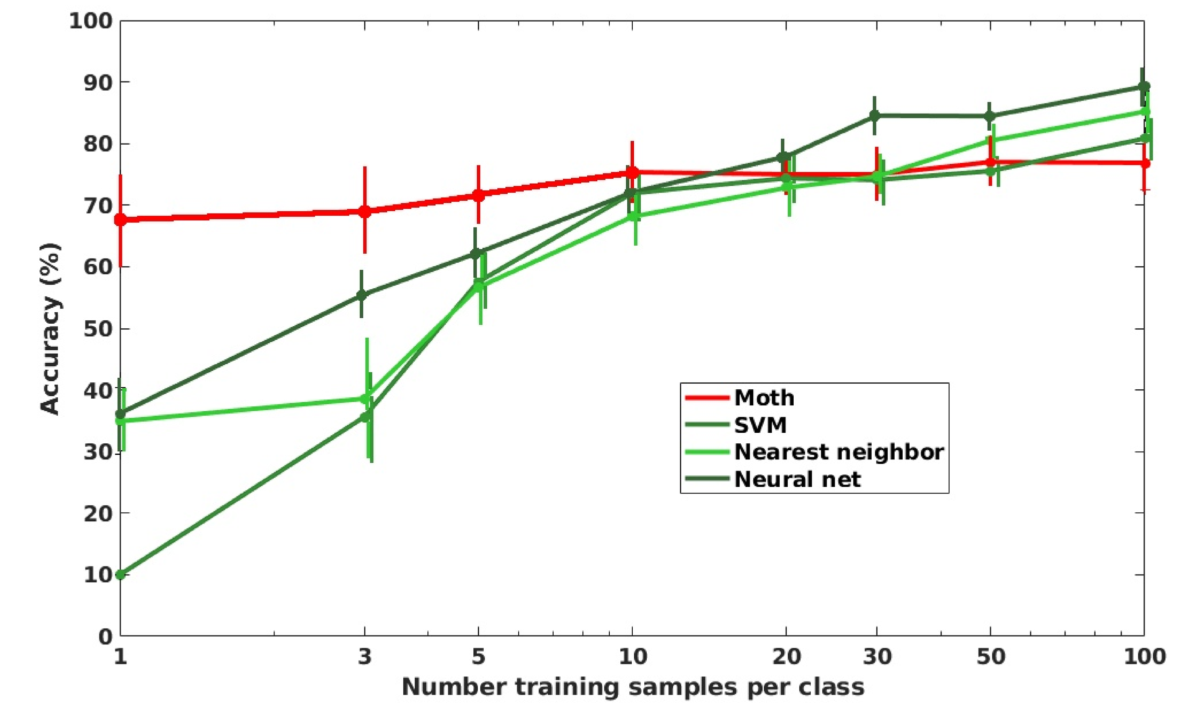


Figure 5: Comparison of various classifiers. The moth out-performs SVM, nearest neighbors, and a neural net (one hidden layer) at very rapid learning (10 or less training digits per class). At 100 digits per class, the moth falls behind. Mean +/- std dev (medians are slightly higher), *N* = 13.

Sparsity focuses Hebbian growth

High-dimensional, sparse neural layers are a widespread motif in biological NNs. In the moth, sparsity in the Mushroom Body controls noise and thus focuses Hebbian growth: Hebbian growth is an *AND* gate. Sparsity enforces silence in one neuron or the other, preventing synaptic growth from non-relevant signals.

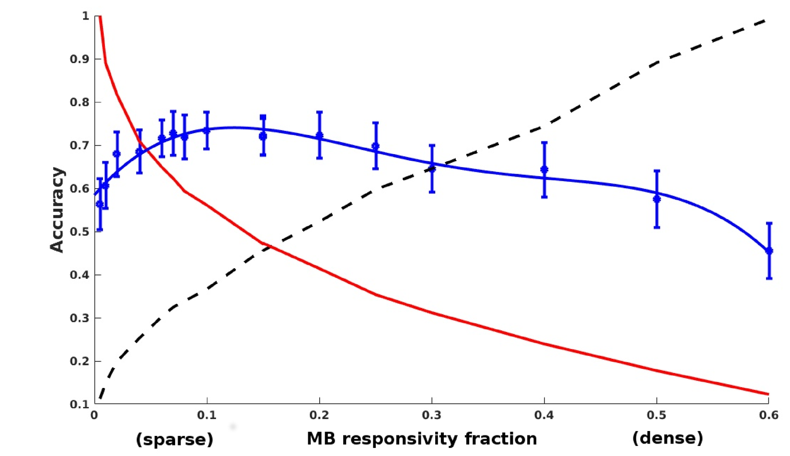


Figure 6: Optimal accuracy (blue domed curve) occurs at 5-20%, as in biological systems. This gives a compromise between high learning focus and high intra-class signal-to-noise ratio (SNR). Red curve = mean separation of trained vs control (learning focus). Black curve = mean intra-class SNR. (Learning focus and SNR are scaled for plotting.)

Discussion

To learn, the moth olfactory network uses just a few core tools: A noisy pre-amp network with competitive inhibition; Hebbian plasticity regulated by a high-dimensional sparse layer; and generalized (global) stimulation during training.

These biological tools are well-suited for combination into larger, deeper neural nets, just as convolutional kernels, etc, are combined to build current DNNs.

The moth is on the bottom rung of the ladder of biological learning complexity. Yet it is a strong rapid-learner, and in fact out-performs standard ML methods.

The ability of this simplest of biological NNs, and the proven success and variety of biological NNs, argue for the potential benefit of porting biological toolkits to ML tasks.

References

Martin JP et al: The neurobiology of insect olfaction: Sensory processing in a comparative context. Science, 2011

Dayan P, Abbott LF: Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT press, 2005

Higham DJ: An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations. SIAM Rev, 2001

Ganguli S, Sompolinsky H: Compressed Sensing, Sparsity, and Dimensionality in Neuronal Information Processing and Data Analysis. Ann Rev of Neurosci, 2012

LeCun Y, Cortes C: MNIST handwritten digit database, 2010. http://yann.lecun.com/exdb/mnist/

Moth photo: World Nature News

Delahunt CB, Riffell JA, Kutz JN: Biological Mechanisms for Learning: A Computational Model of Olfactory Learning in the Manduca sexta Moth, with Applications to Neural Nets. Draft on arXiv, 2018

Delahunt CB, Kutz JN: Putting a bug in ML: The moth olfactory network learns to read MNIST. Draft on arXiv, 2018

Full Matlab codebase: github/charlesDelahunt/PuttingABugInML

 delahunt@uw.edu