# Bio-mimetic feature generators improve ML accuracy on limited data

Charles B. Delahunt [1,2] J. Nathan Kutz [1]

[1] Applied Mathematics, [2] Computational Neuroscience Center, U of Washington, Seattle, WA. *delahunt@uw.edu, kutz@uw.edu*

COMPUTATIONAL NEUROSCIENCE CENTER

APPLIED MATHEMATICS UNIVERSITY *of* WASHINGTON

## Introduction

Informative features are central to ML performance, more so when training data is limited. Biological neural networks excel at fast learning, and extract highly informative features.

The insect olfactory network learns new odors rapidly. It includes four motifs endemic in biological networks: (i) A competitive inhibition layer; (ii) a high-dimensional sparse plastic layer; (iii) sparse inter-layer connectivity; and (iv) Hebbian weight updates.

We deployed MothNet, a computational model of this network, as a front-end feature generator for standard ML classifiers, on vectorized MNIST and Omniglot data sets.

MothNet-generated features significantly improved ML performance vs baseline: relative reduction in test set error averaged 20% to 60%. MothNet features also strongly out-performed comparison feature generators including PCA, PLS, and NNs.

This bio-mimetic architecture encoded, and made accessible, new class-relevant information that was otherwise ignored by the ML methods alone. These results highlight the potential value of bio-inspired NNs as feature generators in the ML context.

## Neural Architecture

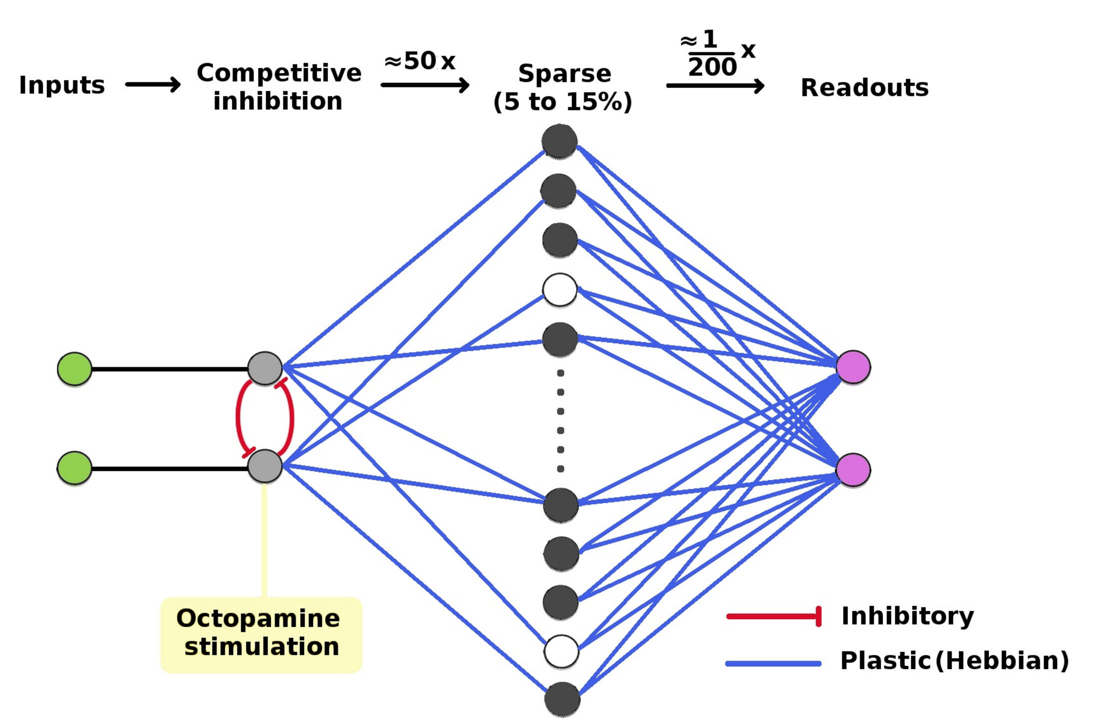


**Figure 1**: Inputs feed 1-to-1 into a pre-amp layer (Antenna Lobe) with ~60 noisy units and intra-layer competitive inhibition. This layer feeds-forward with sparse connectivity into a high-dimensional (~4000 units) but sparsely-active layer (Mushroom Body). The only plastic connections in the system are into and out of this sparse layer. This layer feeds-forward to readout units (RNs). Learning causes RN responses to the various classes to diversify.

## Results for MothNet features on vectorized MNIST

Downsampled, vectorized MNIST images give an 85-feature, 10-class, non-spatial dataset. Small training sets (1 to 100 digits per class, << 6000) constrain baseline ML accuracy. Given a training set *S*, experiments ran as follows:

1. Baseline models: Neural Nets (NNs), SVM, and Nearest Neighbors were trained on *S*, using pixel values as features.
2. Models with added MothNet features: (a) MothNet was trained, by time-evolving stochastic differential equations, on pixel values of *S*. This diversified the responses of its 10 Readout Neurons. (b) The ML models were retrained on *S*, using pixel values plus MothNet readouts. (c) Test set samples were run through the trained MothNet, then through the ML method using pixel values plus MothNet readouts.

MothNet-generated features increased ML accuracy in all cases (Fig. 2). Gains were significant for most cases with $|S| > 3$ samples per class ($p < 0.05$). Relative error reduction was 20% to 60% (Fig 2 inset).

The trained MothNet encoded class-relevant information in the pixel values ignored by the ML methods alone, and made it usable.

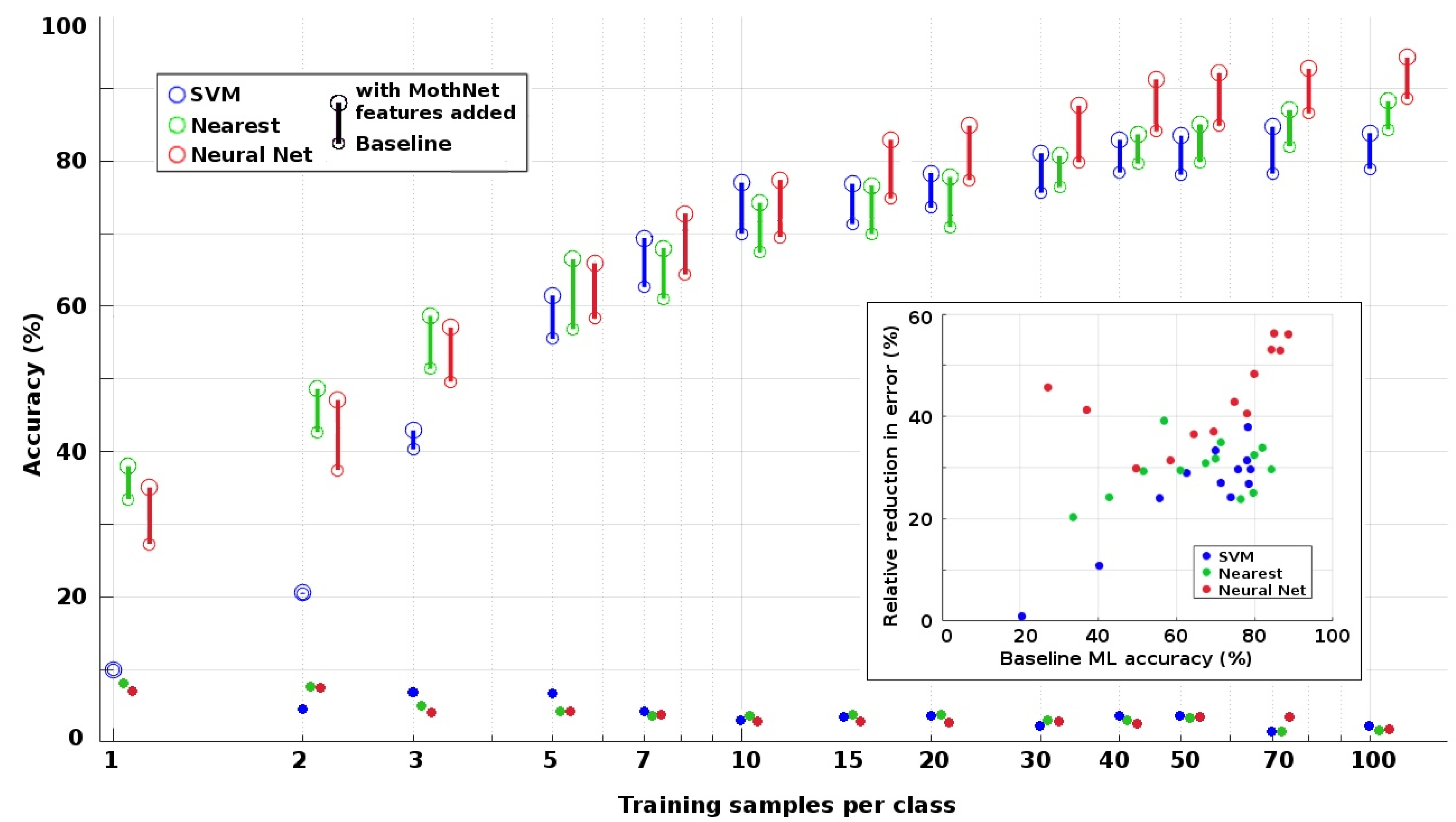


**Figure 2**: Trained accuracy of ML methods, with and without MothNet features, vs |S|. Baseline ML accuracies are small circles, accuracies with MothNet are larger circles, vertical bars show increase in accuracy due to MothNet features. Inset: Relative reduction in error as percentage. MothNet-generated features significantly improved ML accuracy. 13 runs per datum.

## Comparison to other methods

We compared MothNet to other feature generators: PCA, PLS (partial least squares), NNs, and pre-training NN weights on similar data. Features were generated as follows: for PCA and PLS, projection onto first 10 modes of *S*; for NNs, 10 readouts of an NN trained on *S*.

MothNet features substantially out-performed the comparison feature generators. Tables 1 - 3 compare the mean raw gains in accuracy due to each of the feature generators, for each baseline ML method (13 runs per datum).

This may be due to the biological motifs in MothNet's architecture, and to the Hebbian update method. For small |*S*|, these may allow MothNet to extract "orthogonal" class information, relative to feature extractors based on $L_2$ norms.

**Table 1**: SVM, mean raw gains due to feature generators (%).

| **F Gen** | $N$=1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCA | NA | **12.2** | -0.4 | -1.4 | 0.3 | 0.2 | 0.2 | -0.9 | 0.3 | -0.8 | -1.4 | -0.5 |
| PLS | NA | -14 | 4.2 | 3.5 | 1.5 | -0.2 | -2.6 | -4 | -5.4 | -5.3 | -5.1 | -5.5 |
| NN | NA | 6.8 | -1.3 | -3.7 | -2 | -0.9 | 1.7 | 0.5 | 4.3 | 4.1 | 4.9 | 4.9 |
| MothNet | NA | 0.8 | **6.5** | **10.7** | **11** | **10** | **7.8** | **6.3** | **7.2** | **6.9** | **8.3** | **6.2** |

**Table 2**: Nearest Neighbor, raw gains due to feature generators(%).

| **F Gen** | $N$=1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCA | -67 | 0.7 | 0.6 | 1.4 | 1.2 | 1.2 | 1.5 | 1 | 1.3 | 0.0 | 0.9 | 1.5 |
| PLS | NA | 1.4 | 0.6 | 1.6 | 2.1 | 1.5 | 1.1 | 1.9 | 1.2 | 0.4 | 0.9 | -0.1 |
| NN | -1.4 | 1.3 | 2.1 | 1.5 | 2.6 | 2.1 | 4.4 | 3.2 | 4.7 | 3.9 | 3.9 | 3.7 |
| MothNet | **13.6** | **13.9** | **14.2** | **16.9** | **11.5** | **10** | **9.6** | **10** | **5.6** | **6.6** | **6.1** | **4.7** |

**Table 3**: Neural Net, raw gains due to feature generators (%).

| **F Gen** | $N$=1 | 2 | 3 | 5 | 7 | 10 | 15 | 20 | 30 | 50 | 70 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCA | -57 | 0.2 | -0.8 | 1.2 | 2.6 | 1.7 | 0.3 | 1.3 | -0.3 | 0.2 | 0.3 | 0.2 |
| PLS | NA | 0.2 | 5.9 | 1.0 | 1.5 | 2.8 | -0.2 | 1.2 | 0.3 | 1.6 | 1.5 | 1.9 |
| preTrain | **15** | 4.2 | 5.8 | -3.1 | -1.1 | 0.2 | 1.3 | 1.5 | -3.4 | -0.4 | -4.7 | -1.1 |
| MothNet | 4 | **17** | **15** | **13.1** | **13** | **11.3** | **10.8** | **9.0** | **9.7** | **8.5** | **7.1** | **6.4** |

## Results on Omniglot

Omniglot is a character dataset with 20 samples per class. On vectorized Omniglot, MothNet features consistently boosted ML accuracy. Relative gains in accuracy were 5% to 20% for NNs and Near Neighbors, and over 50% for SVMs (Fig 4A). Relative reduction in error was 20% to 60% (Fig 4B).

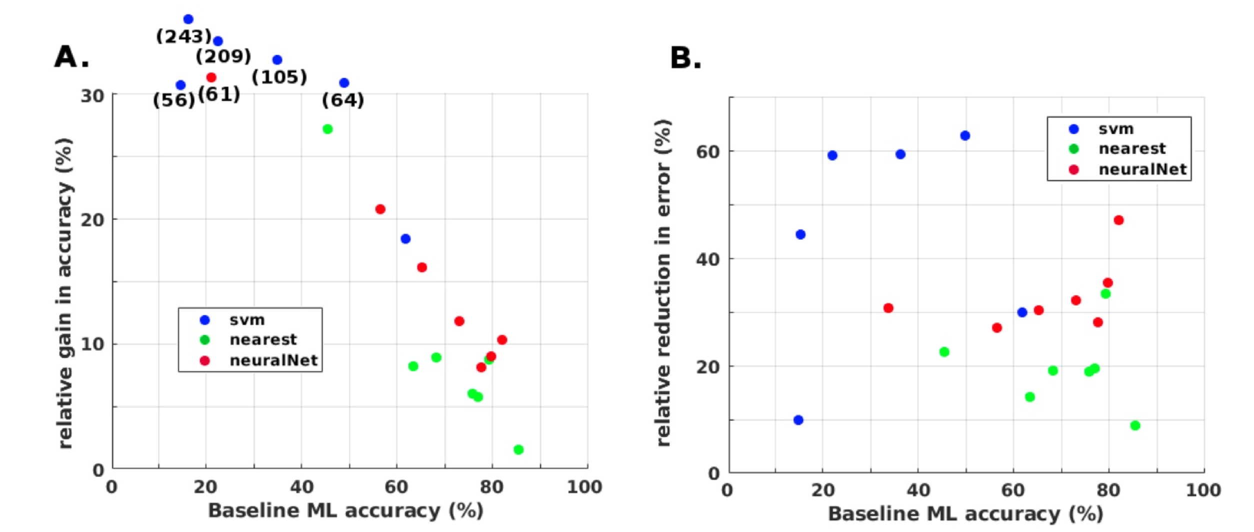


**Figure 4**: Relative change in test set accuracy due to MothNet features, vs baseline accuracy. A: accuracy gains. B: error reductions.

## Conclusion

Biological brains have strong feature extraction abilities. We deployed a bio-mimetic neural network, MothNet, as a feature generator to aid standard ML classifiers.

MothNet-generated features significantly increased ML classifier accuracy on vectorized MNIST and Omniglot. MothNet outperformed comparison feature generators such as PCA, PLS, NNs, and pre-training NN weights.

MothNet includes four key biological motifs: (i) Competitive inhibition, (ii) a high-dimensional, sparsely-firing layer, (iii) sparse inter-layer connectivity, and (iv) Hebbian updates.

These architectural elements extracted and encoded strong class-relevant information not accessed by the ML methods alone. The ML methods were then able to use this new, "orthogonal" class structure to significantly improve accuracy.

Our results indicate that bio-mimetic networks may hold value as feature generators in the ML context.

## References

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

Lake BM et alia: Human-level concept learning through probabilistic program induction. *Science*, 2015 (Omniglot)

Delahunt CB, Kutz JN: Putting a bug in ML: The moth olfactory network learns to read MNIST. *Neural Networks*, 2019

Delahunt CB, Kutz JN: Insect cyborgs: Bio-mimetic feature generators improve ML accuracy on limited data. *arXiv*, 2018

Matlab codebase: github/charlesDelahunt/PuttingABugInML

Python codebase for MothNet: github/meccaLeccaHi/pymoth, by Adam P. Jones

Our thanks to Blake Richards, who suggested these experiments.

*https://charlesdelahunt.github.io* *delahunt@uw.edu* *neurIPS 2019*