Compact Modeling of Nanocluster Functionality as a Higher-Order Neuron

Celestine P. Lawrence

Abstract—Disordered nanoclusters with multielectrode input–output functionality had recently been experimentally realized with energy-efficient and emergent computational capacity, and thus an interconnected network of several such nanoclusters had been proposed to realize artificial neural networks. To aid that end, here we show that nanocluster functionality can be fit to the simplest dendritic neuron model (DNM), where the only form of nonlinearity is due to multiplicative interactions. This work brings into the spotlight higher-order neural networks (known for their efficient encoding of geometric invariances) to serve as an explainable baseline model of nano-networks against which experimentalists can compare more sophisticated models (deep neural networks or physics-based models such as the lin-min network introduced here) and provides ground for designing novel approximate hardware and a statistical mechanics analysis of the learning performance of interconnected nanoclusters versus perceptrons (where neurons output a nonlinear function of the weighted sum of their inputs). A network with just ten higher-order neurons is shown to achieve a classification accuracy of more than 96% on the MNIST benchmark for handwritten digit recognition (which required 100 times more neurons in three-layer perceptrons).

Index Terms—Approximate computing, compact model, dendritic neuron model (DNM), higher-order neural network, lin-min network, nanocluster.

I. INTRODUCTION

COMPACT modeling is the art of characterizing electronic device functionality in terms of physics-based equations, numerical tables, or empirical equations [1]. One trades off accuracy in return for compactness, so it becomes feasible to analyze circuits composed of a large number of devices. It is of special relevance for designing novel nanoscale artificial neural networks by approximate computing [2].

Here, we derive an empirical compact model for a disordered nanocluster, in terms of a higher-order neuron, based on previously done seven-input one-output voltage–current measurements [3, Fig. 6.3, run#1 to be specific]. To check if there exists a mechanistic interpretation for the higher-order neuron model to describe multielectrode nanocluster functionality, the experimental data is also fit to a physics-based model. The higher-order neuron model is faster to evaluate than physics-based device models and sets the stage for unconventional designs of neuromorphic systems, that go beyond three-electrode transistor and two-electrode memristor units [4], [5]. For example, previous work [6] on estimating the accuracy of nanocluster networks for handwritten digit recognition is improved here by using a higher-order neuron model instead of a deep neural network model.

The modeling and analysis are done using Wolfram Mathematica [8] and available at DOI:10.5281/zenodo.6677810.

A. Nanocluster Functionality

Nanocluster functionality, in general, is dictated by nonlinear conduction phenomena in disordered media. The first experimental result to demonstrate complex and robust functionality in nanoclusters was on a cluster of gold nanoparticles measured at ultralow temperatures (≈300 mK), where six control voltages were optimized by a genetic algorithm to realize two-voltage-input one-current-output Boolean logic gates [9]. Next, similar results were obtained at liquid-nitrogen temperature (77 K) and room temperature on a nanocluster of dopant atoms [10]. Here, we build a compact model of such nanocluster functionality (dopant atom network at 77 K) over binary 7-D input space.

The nanocluster is surrounded by electrodes (see Fig. 1), of which N = 7 input electrodes are configured to be at a voltage of either 0 or 1 V, and an output electrode is connected to a transimpedance amplifier to measure a current. The set of voltages is denoted by a binary-coded configuration number. For example, the binary voltage configuration V_{7:1} = 0100010 is

![Fig. 1. Schematic of the nanocluster with seven-input voltages and one-output current. Dashed lines are some hypothetical multiplicative and weighted sum functionality.](image-url)
denoted by the decimal number 34. The measured output current \( C \) (pA) for every one of the \( 2^N = 128 \) configurations (plotted in blue, Fig. 2) is imported from [3], where it was checked to be reproducible and time-stable across different runs with a standard deviation below 50 pA.

### B. Model Fitting

We aim to fit a compact model \( F \) for our binary \( N \)-input and analog one-output nanocluster with \( P \ll 2^N \) \( w \)-parameters such that \( F(V,w_{1:P}) \approx C[V] \forall V \in \{0,1\}^N \). To validate the model, we will do a random 50–50 split of the configuration set \( \{0,1\}^N \) into \( V_{\text{train}} \) and \( V_{\text{test}} \), obtain \( w \)-parameters by minimizing the training-error \( \Sigma_{V_{\text{train}}} (F(V,w_{1:P}) - C[V])^2 \), standardize the values of \( F \), \( C \) across \( V \in V_{\text{test}} \) to zero-mean unit-variance values \( \hat{F} \), \( \hat{C} \), and finally check if the test correlation \( \Sigma_{V_{\text{test}}} \hat{F}(V,w_{1:P})\hat{C}[V] > 0.8 \) (alluding to Pareto’s principle).

If \( F(V,w_{1:P}) \) can be expressed as \( G_{1:P}(V) \cdot w_{1:P} \) for some (nonlinear) projection \( G : \{0,1\}^N \rightarrow \mathbb{R}^P \), then the training error is minimized linear algebraically at \( w_{1:P} = (G_{\text{train}}:1:P)^{(-1)} \cdot C_{\text{train}} \), where the matrix pseudo-inverse \( A^{(-1)} \equiv (A^T A)^{-1}A^T \). It is trivial to fit the training data with a sufficiently large (\( |V_{\text{train}}| \gg P \gg N \)) and expressive projection \( G \), but the model would generalize to the test data only if the nanocluster functionality were compactable. We find that a natural compact model does exist in the form of a dendritic neuron model (DNM)!

### C. Dendritic Neuron Model

The general form of a DNM [11] is

\[
y = \sigma_0(\sum_{j=1}^D \Pi_{i=1}^N \sigma_{ij}(x_i))
\]

where an input \( x_{1:D} \) is transformed to an output \( y \) by four layers starting with a synaptic function \( \sigma_{ij} \), followed by a dendritic multiplication \( \Pi \), a membrane summation \( \Sigma \) over \( D \) dendrites, and finally a soma activation function \( \sigma_0 \). Note that multiplicative interactions are not exploited by most deep-learning methods, due to present-day hardware limitations. Moreover, a dendritic topology is in itself no guarantee for multiplicative interactions, as known from prior work on metallic microwires and nanowires [12], [13]. Thus, to mark a functional distinction, we will call DNM \( s \) with multiplicative interactions (and not necessarily bioscale dendritic connectivity) as DNM.

The simplest class of DNM \( s \) is a two-factor weighted sum of the form

\[
y = \sum_{k=1}^N \sum_{l=1}^N w_{kl}x_kx_l.
\]

Thus, nanocluster functionality can be expressed using a compact model of the form

\[
y = F(V = x_{1:N},w_{1:N^2}) = G_{1,N^2}(x_{1:N}) \cdot w_{1:N^2}
\]

where

\[
G_p(x_{1:N}) = x_{p\%N}x_{\lfloor p/N \rfloor}
\]

and \( p\%N \equiv (p - 1 \mod N) + 1 \).

### II. RESULTS AND DISCUSSION

In Fig. 2, we show for randomly chosen training data, the output currents from the fit compact model (in red) in comparison to the measured data (in blue). We repeat such a fitting for 100 random test-train splits (results summarized in Table I) and find that the correlation between the compact model and measured currents is consistently very high and outperforms a weighted sum model of the form \( F(V = x_{1:N},w_{1:N}) = x_{1:N} \cdot w_{1:N} \).

The attentive reader might have noticed that due to the symmetry of multiplicative interactions, the number of parameters \( P \) can be nearly halved without any loss of model accuracy. \( P \) can further be reduced by discarding parameters that fluctuate wildly across different randomizations of the training set, without much loss in accuracy (because we did

---

**TABLE I**

<table>
<thead>
<tr>
<th>Model</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted sum</td>
<td>0.81 ± 0.03</td>
<td>0.78 ± 0.04</td>
</tr>
<tr>
<td>2-factor weighted sum</td>
<td>0.97 ± 0.01</td>
<td>0.92 ± 0.02</td>
</tr>
</tbody>
</table>

(means ± standard-deviation) over 100 random test-train splits.
not yet apply any form of regularization in the fitting method). So, while there remains a wide scope for refining our compact model, it should not distract us from our clear and surprising (for some) finding that the simplest DNM can be sufficient to explain nanocluster functionality over binary inputs. This finding lays a foundation for future work in the statistical mechanics of learning [14], continuous input models [15], and higher-order neural networks [16].

A. Comparison to a Physics-Based Model

To check for a mechanistic interpretation of the nanocluster functionality, we fit the entire measurement data to a physics-based model (described in Appendix A) and obtain a correlation of 0.99 (higher than a correlation of 0.96 obtained by fitting to the higher-order neuron model). This result is a quasi-validation for compact modeling nanocluster physics, because, in the absence of additional experimental data, multiple physics-based models can be fit to the same functionality.

B. Applications

Note that, sophisticated models of similar nanocluster functionality over a continuous 7-D input space had been built by deep-learning methods [15]. While such continuous models help in extracting practically meaningful functionality out of a single nanocluster, when attempting to engineer functionality out of a large interconnection of these nanoclusters, a discrete model may be sufficient (and even desirable due to their tractability). For example, we can apply tools developed to incrementally replace resilient areas of artificial neural networks with approximately functional devices to increase energy efficiency [17] and to study the impact of device non-idealities on neural network training [18]. For a demo application (check Appendix B for details), we trained a network with just ten higher-order neurons (and 70 weighted-sum units) to achieve a test accuracy of 96.5% on the MNIST handwritten digit recognition dataset. This estimate is remarkably more resource-efficient than three-layer Rosenblatt perceptrons that employed 8000 thresholded weighted-sum six-input units for 95.5% test accuracy [19], and comparable to the 96.7% test accuracy obtained by a quadratic classifier over 40 principal components [20]. The next step could be to explore deeper networks of higher-order neurons for object recognition [21].

C. Novelty of the Main Result

Finally, it helps to reiterate that the existence of a compact model, although quite expected for conventional electronic devices, is not a priori certain for unconventional nanoelectronic devices with rich multi-input interactions. In another context, when developing nanoelectronic devices for cryptographic functionality [22], there should exist no compact models.

III. Conclusion

Further experiments or simulations (such as Monte Carlo [23], [24] and mean-field [3, Sec. 2.2.2] methods) are warranted to find and exploit the compactness (or lack thereof) of functionality in disordered nanoclusters made out of different recipes of materials, device structures, and manufacturing processes.

APPENDIX A

The electron flow in a nanocluster can be described by a master equation system [25], where the transition rates \( \Gamma_{ij} \) from one state \( i \) of electron occupancy to another \( j \) can be idealized [3, eq. (2.9) and Sec. 2.2] as a piecewise linear function of the voltage inputs. Now assuming: 1) the state transition graph is a simple cycle; 2) the mean flow in a cycle is approximately equal to the minimum flow in a cycle; and 3) the inputs operate in a region such that the piecewise linear state transition rates can be cast as a minimum over linear functions [26]; we can express the output current as

\[
y = \min_{k=1}^{M} \left( b_k + \sum_{l=1}^{N} w_{kl} x_l \right)
\]

which can be seen as a neural network with a linear layer of \( M \) units feeding to an output neuron computing the minimum function. We can call this a lin-min network.

We used a lin-min network with \( M = 8 \) and fit to the experimental data (zero-mean unit-variance standardized, to aid the learning algorithm) using a validation ratio \( = 0.2 \). The default optimizer (Adam) took 11 s to run on a four-core CPU.

APPENDIX B

The MNIST dataset (http://yann.lecun.com/exdb/mnist/) contains grayscale images (28 × 28 pixels) of handwritten digits from 0 to 9 (60k train + 10k test images). The image matrix is flattened and zero-mean unit-variance standardized (to aid the learning algorithm) to a 784-input array and connected to a linear layer of 70 units, of which ten sections (of seven units) are each connected to a higher-order neuron (that is two-factor weighted sum model fit to the output shown in Fig. 2 across all input configurations of a nanocluster). The ten higher-order neurons are connected to a softmax layer to complete the neural network for ten-digit classification, and the learning of its 784 × 70 weights and 70 biases was done on a four-core CPU for a runtime of 2 min, using the default optimizer (Adam).

ACKNOWLEDGMENT

The author would like to thank Herbert Jaeger and anonymous reviewers for their comments on earlier versions of this manuscript.

REFERENCES


