Published in final edited form as:

Conf Proc IEEE Eng Med Biol Soc. 2018 July ; 2018: 6129–6132. doi:10.1109/EMBC.2018.8513616.

Modeling Nonlinear Synaptic Dynamics: A Laguerre-Volterra Network Framework for Improved Computational Efficiency in Large Scale Simulations

Eric Y. Hu [IEEE Student Member],

Department of Biomedical Engineering, University of Southern California, Los Angeles, USA
(ehu@usc.edu)

Gene Yu [IEEE Student Member],

Department of Biomedical Engineering, University of Southern California, Los Angeles, USA
(geneyu@usc.edu)

Dong Song [IEEE Member],

Department of Biomedical Engineering, University of Southern California, Los Angeles, USA
(dsong@usc.edu)

Jean-Marie C. Bouteiller [IEEE Member], and

Department of Biomedical Engineering, University of Southern California, Los Angeles, USA
(jbouteil@usc.edu)

Theodore W. Berger [IEEE Fellow]

Department of Biomedical Engineering, University of Southern California, Los Angeles, USA
(berger@usc.edu)

Abstract

Synapses are key components in signal transmission in the brain, often exhibiting complex non-linear dynamics. Yet, they are often crudely modelled as linear exponential equations in large-scale neuron network simulations. Mechanistic models that use detailed channel receptor kinetics more closely replicate the nonlinear dynamics observed at synapses, but use of such models are generally restricted to small scale simulations due to their computational complexity. Previously, we have developed an “input-output” (IO) synapse model using the Volterra functional series to estimate nonlinear synaptic dynamics. Here, we present an improvement on the IO synapse model using the Laguerre-Volterra network (LVN) framework. We demonstrate that utilization of the LVN framework helps reduce memory requirements and improves the simulation speed in comparison to the previous iteration of the IO synapse model. We present results that demonstrate the accuracy, memory efficiency, and speed of the LVN model that can be extended to simulations with large numbers of synapses. Our efforts enable complex nonlinear synaptic dynamics to be modelled in large-scale network models, allowing us to explore how synaptic activity may influence network behavior and affects memory, learning, and neurodegenerative diseases.

I. INTRODUCTION

Synaptic transmission between neurons is a critical process essential for neuron network activity and lies at the foundation of learning and memory. Yet in large network models modeling neuronal activity, incorporating nonlinear synaptic dynamics is difficult to achieve without drastically increasing the computational resource requirements. Our most recent developments focus on developing nonlinear synapse models to be feasibly modeled in multi-scale simulations with minimal impact on computational load. Previously, we presented in detail the input-output synapse model, highlighting its abilities to accurately predict the response of a mechanistic model[1]. The expected follow-up to the IO model development, then, is its implementation into large scale simulations. However, upon implementation into the large-scale model, it became clear that the IO model structure still had limitations impacting computational efficiency. We present these limitations and propose a solution to address them in this manuscript.

Our past IO synapse model uses the Volterra functional series as its framework, with Laguerre equations as the basis functions [2]. The advantage that the Volterra functional series provides over the kinetic rate models is that the series can predict a nonlinear response based on linear equations. This is in contrast to the kinetic rate models, which use nonlinear rate equations to solve for the different state compartments represented in the model[3]. However, the number of linear equations required in the IO model increases exponentially with respect to the order of the IO model. This means for highly nonlinear IO models, the number of equations becomes prohibitively high. To address this issue, the basic structure of the IO model itself must be changed.

A variant of the Volterra functional series using Laguerre basis functions is the Laguerre-Volterra network (LVN) model, which integrates the previous technique with neural network modeling [4]. The LVN model is a parsimonious model which highlights mainly the more prominent nonlinear changes in the system being estimated. This allows the model to be scaled linearly with respect to increasing nonlinear complexity of the system. However, the LVN model has typically been limited by the training procedure and large parameter search space. We herein present a modified protocol of the simulated annealing procedure from [4]. We validate the LVN model with the mechanistic model to show the model is accurate. Finally, we benchmark the performance of the original IO synapse model and LVN synapse model in large-scale simulations.

II. METHODS

A. LAGUERRE-VOLTERRA NETWORK MODEL

The LVN model is a nonparametric method that integrates the Laguerre estimation technique with a connection-based neural network approach [4]. In the first section, we describe the structure of the LVN prediction model; the following session describes the training protocol for the model's associated weights and coefficients.

Laguerre-Volterra Network Structure—A schematic of the Laguerre-Volterra Network (LVN) model is shown in Figure 1. The LVN model uses an artificial neural network

structure where the inputs to each of the individual units is the signal input convolved with a number of Laguerre basis functions. The Laguerre basis functions and their derivations had been described previously in the Laguerre expansion technique [1], [2], [5]. Briefly, the Laguerre basis functions are a set of orthonormal functions with parameter p that controls the exponential decay of the basis function set. In the LVN model we developed, we use two basis function sets with two p values. This is to consider both short-term and long-term changes to the system, similar to the previous Laguerre estimation technique used for the IO synapse model. Each basis function set contains 3 Laguerre basis functions; this gives us a total of 6 basis functions per unit in the LVN model.

The equations which describe the LVN model is as follows:

$$z_h(n) = \sum_{j=0}^{B-1} w_{j,h} L_j(n)$$

$$y_h(n) = \sum_{q=0}^Q c_{q,h} z_h^q(n)$$

For each hidden unit h , there is a hidden parameter z which constitutes the summation of all the basis functions convolved with the input L , each multiplied by an optimized weight function w_j . Then, z is taken to the power of each order value q up to the designated order of the model Q ; each of these are scaled by a coefficient C_q and summed together to produce the output of that particular unit, y_h . All the outputs of all the units are summed together to produce the predicted response y . For the LVN model, the number of hidden units H and the order of each unit q can be varied; we chose the values of H and q to be 6 and 4, respectively, to ensure our model is accurate and structurally efficient.

Parameter Estimation of the LVN Model—In [4], the weights w , coefficients c , and decay value p were estimated using a simulated annealing algorithm. We, too, employed simulated annealing to train the LVN IO synapse model; however, we include adjustments which help reduce the parameter space to be estimated during simulated annealing. By reducing the parameter space, we expedite the training process. Our first modification is an assumption that the basis function decay value p has been optimized already during the development of our initial IO synapse model through Laguerre expansion technique; we therefore use the p values from our previous model for the LVN model as well. Our second modification is to use the weights and hidden value z to estimate coefficients through pseudoinverse, rather than including the parameters in simulated annealing. Given the training data set y_{actual} we estimate the coefficients of the hidden units by taking the pseudoinverse of the hidden parameter matrix $\mathbf{Z}$:

$$c = y_{actual} \cdot \text{pinv}(\mathbf{Z})$$

$$y_{est} = c \cdot Z$$

Z is a matrix of the hidden parameters of each order from all the hidden units, and c is an array of the coefficients for all the hidden units. This process is essentially identical to the coefficient estimation when using the Laguerre expansion technique of the original IO synapse model; the difference is that the pseudoinverted matrix is the hidden parameter values instead of the convolved basis functions.

The remaining weights are estimated with a simulated annealing process similar to that described by [4]. The algorithm is described here:

1. Initializing simulated annealing training parameters. Input training data includes randomized input trains of both 2 Hz and 10 Hz, 500 events each for a total of 1000 events. The training output response is the AMPAr or NMDAr kinetic model response to the said randomized input. Initial weights are set to 0. In simulated annealing, there are two types of iterative processes: local iterations, where the temperature parameter does not change, and global iterations, where temperature is reduced for each iteration. The number of iterations should be large enough to reach error difference stability by the end of training, but not too large or else the process may take an excessive amount of time. The number of global iterations is set to 150. One global iteration contains 60 local iterations, which results in approximately 3 days of training. This gives a total number of 9000 iterations for training. Initial temperature is set to 100 with a cooling rate of 0.99. Decay parameter values are determined previously from the Laguerre estimation technique. We use two decay values per model, where half of the hidden units use one decay value and the other use the other decay value. They are set to 0.5 and 0.008 for AMPA LVN model, and 0.054 and 0.003 for the NMDA LVN model.
2. Weight change process. A single weight chosen randomly from all the hidden units is either added or subtracted by the value of the weight step size, set to 0.1.
3. Estimation of Z and coefficients. The Laguerre basis functions are convolved with the training input data, then multiplied by their respective weights. The hidden parameter z is determined by the summation of the weighted and convolved basis function. A Z matrix with q number of z vectors are calculated for each hidden unit, where q is the order of the hidden weight model. These z vectors refer to the 1st to q th order of the model, and each such vector is the z parameter taken to the power of its respective order. The pseudoinverse of the Z matrix is calculated and multiplied by the training output data to estimate the coefficient values. Coefficients are multiplied with the Z matrix give the estimated output response; the estimate is compared with the training output to get the error difference.
4. The error difference determines whether the weight change from the current iteration should be changed or not. If the current error is less than the error from

the previous iteration, the weight change is kept. If the error is larger than the previous iteration's error, a probability value is calculated based on the difference between the current and previous error divided by the temperature:

$$p = e^{-\left(\frac{\text{error}_{\text{current}} - \text{error}_{\text{prev}}}{T}\right)}$$

The probability is used to determine whether the weight change should be kept or not.

5. If the end of the local iteration is also the end of the global iteration, the temperature parameter is reduced by multiplying the current temperature with the cooling rate. Then another iteration is started by repeating steps 2–4.

Because the process is random, multiple simulated annealing training procedures are run concurrently on different nodes on a high performance computing cluster. At the end of training, the procedure that results in the lowest error difference has the weights selected for the new LVN model.

B. Integration Methods of the IO Synapse Model

Two integration methods are generally used for ODEs within the NEURON platform, depending on the nonlinearity of the rate equations. The 'derivimplicit' solver is used for models with nonlinear ODEs, such as the mechanistic synapse model, whereas the 'cnexp' solver is used for linear ODEs such as the exponential synapse model. While the IO synapse model response is nonlinear, the ODEs which the IO model is composed of is linear, therefore the model solver was switched from 'derivimplicit' to 'cnexp' to improve simulation time performance. Consequently, the response of the IO synapse model had no difference at all when the solver was changed.

III. RESULTS

Trained LVN IO Synapse Model Accurately Estimates the Response Based on the Mechanistic Model

The weights of the LVN model for the AMPA receptor conductance and the NMDA receptor open state probability were estimated by using the simulated annealing protocol described in the Methods section. For each model, we ran the simulated annealing protocol seven times, in parallel; the resulting training normalized root mean square error (NRMSE) for each simulated annealing protocol had a range between 4.1% to 5.39% for the AMPA LVN models, and 5% to 5.91% for the NMDA LVN models. We chose the weights with the lowest training error for each respective receptor LVN model (4.1% for AMPAr, 5% for NMDAr). In validation of the AMPAr LVN model, the NRMSE was 3.39% for 2 Hz input response and 5.19% for 10 Hz randomized input train (RIT) response; the NMDAr LVN model had NRMSE of 6.6% and 9.24% for 2 Hz and 10 RIT response, respectively (Figure 2). The low error highlights the accuracy of the predicted response from the LVN model.

Simulation Time Benchmarking of the IO Synapse Models

The simulation time of different synaptic models was measured on the large scale model, with the number of synapses ranging from 500 to 20,000 synapses on a single granule cell neuron model for 1 second of simulated time (Figure 3). Input to all synapses was a randomized poisson train of 3 Hz. The mechanistic model requires the most simulation time out of all the synapse models tested, needing a total of approximately 50,000 seconds to finish a simulation with 20,000 synapses. The exponential synapse requires the least amount of simulation time, needing only 850 seconds to run a simulation of 20,000 synapses. We tested the IO synapse model using LVN at 4th order, as well as the IO synapse model using the Volterra functional series at 3rd order and 4th order, using two different solvers and compared the performance between all synapse models. Out of all the IO synapse models compared, the LVN 4th order model using the cnexp solver had the best simulation time, needing approximately 6200 seconds to run. This amount is approximately 8 times faster than the mechanistic synapse model, yet still 8 times slower than the exponential synapse model. Nevertheless, the IO LVN model using the cnexp solver has significantly improved compared to the 40,000 second simulation time of the initial IO synapse model using the derivimplicit solver. These results demonstrate the capabilities of the new model structure and its effects on simulation efficiency.

Memory Requirements for Large-Scale Simulation of Synapse Models

Large memory requirements for synapse models impose limitations to the number of synapse that can be instantiated in a large-scale simulation. To determine the memory usage for different synapse models, we measured the amount of memory required during the setup of the simulation for each synapse model, with synapse numbers ranging from 10 to 30,000 synapses (Figure 4). In the memory usage tests, the original IO 4th order model had the most amount of memory usage, requiring approximately 400 megabytes of memory for a simulation with 30,000 synapses. The exponential synapse model requires the least amount of memory at around 30 megabytes. The mechanistic synapse model and the IO-LVN synapse model have comparable results, with the IO-LVN synapse model having about 20% less memory requirements than the mechanistic model (60 mb vs 75 mb, respectably). The improved memory efficiency of the IO-LVN synapse model can allow for larger scale simulations to be run with less concern for memory limitations of the computer.

IV. CONCLUSION

Here we have presented an LVN implementation of the IO synapse model and benchmarked its computational efficiency. Developments in the near future involve the implementation of the IO model into a large scale network simulation with thousands of neurons, and millions of synapses. These methodological advances will allow us to better understand the effects of complex nonlinear synaptic dynamics on neuron communication, learning, memory, as well as neurological diseases.

Acknowledgments

This work was supported in part by National Institute of Biomedical Imaging and BioEngineering (NIBIB) grant P41 EB001978 and U01 GM104604.

V. REFERENCES

- [1]. Hu EY, Bouteiller J-MC, Song D, Baudry M, and Berger TW, "Volterra representation enables modeling of complex synaptic nonlinear dynamics in large-scale simulations," *Front. Comput. Neurosci.*, vol. 9, 9 2015.
- [2]. Berger T and Song D, "The neurobiological basis of cognition: Identification by multi-input, multioutput nonlinear dynamic modeling," *Proc. ...*, vol. 98, no. 3, 2010.
- [3]. Bouteiller J-MC, Allam SL, Hu EY, Greget R, Ambert N, Keller AF, Bischoff S, Baudry M, and Berger TW, "Integrated multiscale modeling of the nervous system: predicting changes in hippocampal network activity by a positive AMPA receptor modulator.," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 10, pp. 3008–11, 10 2011. [PubMed: 21642035]
- [4]. Geng K and Marmarelis VZ, "Methodology of Recurrent Laguerre-Volterra Network for Modeling Nonlinear Dynamic Systems," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–13, 2016.
- [5]. Ghaderi VS, Allam SL, Ambert N, Bouteiller J-MC, Choma J, and Berger TW, "Modeling neuron-glia interactions: from parametric model to neuromorphic hardware.," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2011, pp. 3581–4, 1 2011.

Figure 1. A diagram of the Laguerre-Volterra Network structure.

A hidden unit consists of basis functions weighted and summed together to form the hidden parameter z , which is then used to calculate linear and nonlinear contributions of the unit. The outputs of multiple hidden units are summed to produce the predicted response of the system.

Figure 2. Laguerre-Volterra Network models trained and validated based on the AMPAr and NMDAr mechanistic model outputs.

The predicted response of the LVN models compared to the mechanistic models during validation are displayed here. The normalized root mean square error of the AMPAr models at validation were 3.39% for 2 Hz RIT and 5.19% for 10 Hz RIT input; For NMDAr models, the difference was 6.6% and 9.24% for 2 Hz and 10 Hz RIT input, respectively.

Figure 3. Simulation runtime of a large scale neuron model based on number and type of synapses.

The type of synapse models benchmarked were the exponential synapse model (black), mechanistic synapse model (purple), and Input-Output synapse models with differing structures and solver type. In (A), the simulation runtime is plotted (in minutes) with respect to number of synapses. (B) describes the ratio of runtime of the mechanistic or IO synapse over the runtime of the exponential synapse. The mechanistic synapse model can be up to 70 times slower than the exponential synapse, whereas the LVN-IO synapse model using the cnexp solver performs better at being approximately 8 times slower than the exponential synapse.

Figure 4. Memory requirements for different synapse types in a large scale neuron simulation. For each synapse model type, the number of synapses was varied from 10 to 30,000 and the memory usage was plotted. The original IO synapse model had the most memory consumption (in orange), while the exponential model required the least memory consumption (blue). The LVN model is the second lowest in memory consumption when compared to all other synapse models, showing significant reduction over the original IO synapse model by over 70%.