2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Efficient Point-Process Modeling of Spiking Neurons for Neuroprosthesis

Weihan Li[1†], Cunle Qian[1†], Yu Qi[2*], Yiwen Wang[3], Yueming Wang[4] and Gang Pan[1*]

Abstract— **Neuroprosthesis refers to implantable medical devices which can replace injured biological functions in the brain. One of the core problems in neuroprosthesis study is to construct a neural signal transformation model from one cortical area to another. Since the brain encodes and transmits information in spike trains, spiking neural network (SNN) can be an ideal choice for neuroprosthesis modeling. This paper proposes a spiking neuron point-process model (SNPM), which receives spike times as input, and is capable of modeling nonlinear interactions between cortical areas. The proposed SNPM can be implemented on neuromorphic chips for low-energy computing, thus has potential for clinical applications. Experiments show that SNPM can accurately reconstruct functional relationships from PMd (dorsal premotor cortex) to M1 (primary motor cortex) areas.**

I. Introduction

Different functional areas in our brain communicate with each other by synaptic connections, but injures of these connections could cut off the transregional brain communication and lead to cognitive functional losses. One possible solution to repair such losses is to design a neuroprosthesis, which relinks the damaged connections by modeling the spike transformation of neural populations [1] [2].

An optimal neuroprosthesis should have the following properties: (1) It should take spike times as its inputs to best preserve the information in spike trains [3]; (2) It should be computationally efficient so as to meet the stringent power budget of the intracranial implantation; (3) It should be biologically plausible from the perspective of the brain.

Considering these properties, spiking neural network (SNN) can be a primary candidate. SNN not only guarantees biologically plausible, but also allows efficient temporal information coding and promises considerable computing power savings with neuromorphic chips [4] [5] [6].

*The corresponding authors are Yu Qi and Gang Pan. This work was supported by the Key Research and Development Program of Zhejiang Province in China (2020C03004), and partly supported by the grants from the National Key Research and Development Program of China (2018YFA0701400), and Natural Science Foundation of China (61906166, U1909202), and the Zhejiang Lab (2019KE0AD01).

†Equal contribution.

[1]Weihan Li, Cunle Qian and Gang Pan are with the College of Computer Science; Gang Pan is also with The First Affiliated Hospital, College of Medicine, Zhejiang University, and Key Laboratory for Biomedical Engineering of Ministry of Education, Zhejiang University. (e-mail:weihanli@zju.edu.cn, qiancunle@zju.edu.cn, gpan@zju.edu.cn)

[2]Yu Qi is with the MOE Frontier Science Center for Brain Science and Brain-machine Integration, and the College of Computer Science and Technology, Hangzhou, China. (e-mail:qiyu@zju.edu.cn)

[3]Yiwen Wang is with the Department of Electronic and Computer Engineering and the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Kowloon, Hongkong SAR, China. (e-mail:eewangyw@ust.hk)

[4]Yueming Wang is with the Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, China. (e-mail:ymingwang@zju.edu.cn)

Efforts have been made in modeling the transformation between the brain areas. Song et al. [1] proposed a computation model to capture the nonlinearity transformation from hippocampal CA3 to CA1, and achieved high prediction accuracy on modeling the CA1 output spike distribution based on CA3 input spike trains. However, the estimation of this model becomes computationally complex when the size of input data increases. They [7] then developed a sparse model coefficients representation to optimize the estimation with large-scale binned spike trains. Qian et al. [8] proposed a staged point-process model (SPM), which can theoretically approximate arbitrary spike transformation nonlinearity. Compared with Song's model, SPM requires less parameters to estimate, however, it needs to take several times of re-training to obtain the best parameter initialization, thus still has low computational efficiency. From the perspective of neuroprosthesis with SNN, Dethier et al. [9] designed a SNN based Kalman decoder on ultra-low power neuromorphic chips, which decodes spike trains from motor brain regions to control signals for a prosthetic arm. Nevertheless, this decoder receives spike rates calculated in 50ms bin width, which may affect accuracy due to missing details of temporal information.

In this work, we propose a spiking neuron point-process to capture the nonlinearity of spike transformation. Specifically, we construct a SNN version of staged point-process model (SPM), which receives historical spike times rather than binned spike train as inputs. In doing so, our biologically plausible model could have a more stable performance on parameters estimation than original SPM [8], and prevents lossy compression of binned spike train as well as reducing massive heat dissipation on specialized hardware.

We evaluate the spiking neuron point-process model (SNPM) with both synthetic nonlinear spike data and real neuron data from PMd (dorsal premotor cortex) to M1 (primary motor cortex). Our approach is evaluated and compared with the original SPM, generalized Laguerre Volterra model (GLVM) [7] and generalized linear model (GLM) [10]. The rest of this paper is organized as follows: section II describes the methods of the whole neural point-process model, section III introduces the results of SNPM, and section IV presents the conclusion and future work.

II. Methods

The methods consist of two parts. First of all, we illustrate the preprocessing of recorded spike train. After that, we describe the steps and theoretical analysis of the conversion from SPM to SNPM. The overall framework of spiking neuron point-process model (SNPM) is shown in Figure 1.

A. Spike Train Preprocessing

Spike train is a type of events that occur in the domain of continuous time, where information is encoded by the firing times of action potential. Currently, most decoding or computation models discretize the point-process of spike train [10], and set "1" to the given discrete time interval if there exists spikes, otherwise set '0', which means nothing happens. However, such binarization may lead to lossy compression. Consider a discrete time interval with large time width, there is a chance that multiple spikes exist in this interval. That is, spike train may degrade to spike firing rate, thus the details will be lost. On the other hand, a discrete time interval with small time width also causes temporal jitter [3] and brings high computational complexity to the model due to the high dimensional spike data.

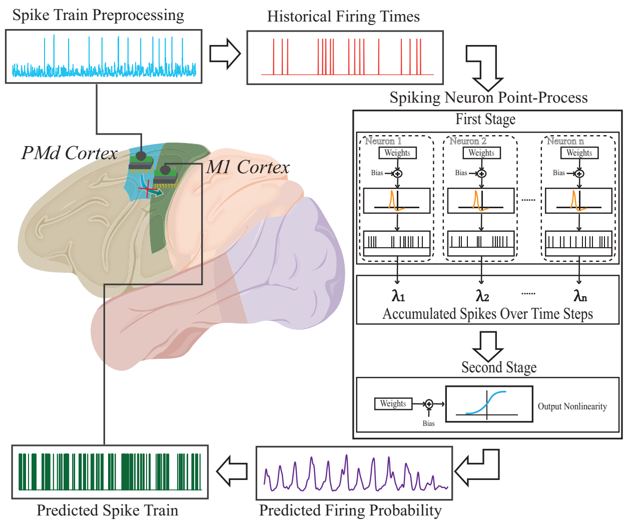

Fig. 1. The framework of spiking neuron point-process model (SNPM). We directly obtain spike times by preprocessing recorded PMd spike trains. Then we use a two-stage spiking neuron point-process to predict the firing probability of output M1 neurons. Finally, we generate output spike trains based on predicted firing probabilities.

An alternative approach is to take firing times as inputs, since those time intervals with no spikes usually encode little information, which instead almost exists in the spike firing times [3] [11]. Given the time point t, let y^t denotes the spike train of output neuron at time t and $(\tau_1, ..., \tau_h, ..., \tau_t)$ be the input historical firing times with τ_h represents the time interval between firing time t_h and previous firing time t_{h-1}, our goal is to model the output neuron's firing probability Pr_y^t and the final output spike train y^t is generated by setting a threshold on Pr_y^t.

Technically, the output spike y^t is determined by all previous historical firing times since spiking neuron point-process model considers the output neuron train as a point-process. However, each spike event at different time point obviously has unequal influence on the current output spike. Given this insight, we use exponential decay to capturing the temporal evolution, which is similar to a postsynaptic current generated by synaptic models

$$x^t = exp(-\frac{(\tau_1, ..., \tau_h)}{\alpha}) \tag{1}$$

where x^t is the input history spike times and $\alpha = 100$ denotes the decay parameter.

B. Spiking Neuron Point-Process

The approach to construct a spiking neuron point-process is taking the parameters of a pre-trained staged point-process model (SPM) and mapping them to an equivalent spiking neural network (SNN) as well as converting all the neurons in SPM to spiking neurons. In dong so, our model is more biologically plausible considering this spikes-in/out application with temporal coding scheme.

Besides, previous study with TrueNorth [12] indicate that a power dissipation of only a few hundred mW is achieved with more than a million neurons implemented, which also implies our model's ability to low-energy computing on these neuromorphic platforms.

We start by giving a brief view of SPM [8]. SPM can be considered as a cascade structure with two linear-nonlinear (a linear combination plus a static nonlinear function) stages and an inhomogeneous Bernoulli process, which follows the idea of two-stages ANN and allows arbitrary nonlinear mapping between the spike of input neurons and output neuron [13]. Using statistical learning theory under the point-process framework, we can probabilistically derive the staged point-process model

$$\begin{aligned} \text{first stage: } & \lambda_z^j = \sigma(\sum_{i=1}^{N}\sum_{h=1}^{H} x_i^h W_{i,h}^j + W_0^j) \\ \text{second stage: } & Pr_y = f(\sum_{j=1}^{N_z} \theta_j \lambda_z^j + \theta_0) \end{aligned} \tag{2}$$

where λ_z and Pr_y are the firing probability of hidden neurons and output neuron, and $\sigma()$ represents the ReLU activation function as well as $f()$ denotes the softmax function.

To properly estimate parameters $W_{i,h}^j$, W_0^j, θ_j and θ_0, we choose cross-entropy loss $L = -\sum_{y=1}^{N_y} P_y^* \log(Pr_y)$ as the log-likelihood function, and use scaled conjugate gradient algorithm [14] to maximize it. Since current SPM receives historical firing times instead of binary spike values as inputs, the optimization surface of log-likelihood function becomes smoother [15] and the training has a more stable performance on different initial parameters settings than original SPM. That is, a higher computational efficiency is achieved when it comes to real application of an implanted neuroprosthesis.

After obtaining a pre-trained SPM, the problem becomes how to guarantee the firing of spiking neurons should exactly match the activation of SPM's neurons. Our core idea is followed by [16] where a link between the transfer function of a integrate-and-fire (IF) spiking neuron to the activation of ReLU is established.

Considering a two-layer SNN, every IF spiking neuron has a membrane potential $V_i^l(t)$, which integrates its input current at each time point. Once $V_i^l(t)$ reaches threshold V_{th}, a spike will be generated and the value of $V_i^l(t)$ will be reset to zero. Meanwhile, note that if given the time steps T, we can calculate the firing rate of each IF spiking neuron i as $r_i^l(t) = (\sum_{k=1}^{t} H_i^l(t))/t$, where $H_i^l(t)$ is the Heaviside step function representing the producing of a spike at time t.

As in [17], there exists a relationship between the firing rate $r_i^1(t)$ of first layer's neurons in SNN and the activation a_i^1 of the hidden units in SPM

$$r_i^1(t) = a_i^1 \frac{1}{\Delta t} \frac{V_{th}}{V_{th} + \epsilon_i^1} - \frac{V_i^1(t)}{t(V_{th} + \epsilon_i^1)} \tag{3}$$

where ϵ_i^1 is the exceeding amount of ith neuron's membrane potential when producing a spike and Δt denotes the time interval size among T time steps.

Based on equation (3), it's clear that the firing rate is proportional to the hidden units' activation, but reduced by an additive error term. That is, we can approximate ReLU's activation by accumulating spikes over all time steps and ReLU itself can be viewed as a firing rate approximation of an IF spiking neuron model without refractory period.

Additionally, in order to avoid approximation errors of too high or too low firing, weight normalization [18] is also introduced. Given all possible activations that may occur as an input to the next layer, we scale all the weights by the maximum possible input. Denoting the scale factor as the max sum of weights from last layer's neurons among all current layer's neurons $\mu^l = max_{i=1}^{N^l}(\sum_{j=1}^{N^{l-1}} max(w_{ij}^l))$, then weights w^l and biases b^l are normalized to $w^l = w^l/\mu^l$ and $b^l = b^l/\mu^l$.

In summary, the steps for converting staged point-process model to spiking neuron point-process are as follows

- Directly map the weights from the staged point-process model to a spiking neuron point-process with IF neurons.
- Find a reasonable time steps T to approximate ReLU's activation.
- Use weight normalization to obtain better accuracy and faster convergence.

III. Results

We evaluate our model on synthetic nonlinear spike data and real neural data recorded from PMd to M1 cortex. We compare to staged point-process model (SPM), generalized Laguerre Volterra model (GLVM) and generalized linear model (GLM). We measure the ability of nonlinearity modeling by DTR-KS test.

A. Assessing Goodness-of-Fit

The goodness-of-fit is defined by the DTR-KS [19], where the maximal distance (denoted by D) between the DTR-KS curve and the $45°$ line is calculated for model evaluation. In order to evaluate the general spike prediction performance across different output neurons, the 95% confidence bound is considered and the distance-bound ratio (DBR) is given by

$$DBR = \frac{D}{1.36}\sqrt{C_{spike}} \tag{4}$$

where C_{spike} is the actual spike count of the given output spike observation. The lower the DBR is, the better performance the model has.

B. Performance with Synthetic Spike Data

First of all, the spike probability of input synthetic spike data is defined as a sinusoidal function $p_x^k = \alpha_1 sin(\beta_1 t^k) + \gamma_1$, where $\alpha_1 = 0.3$, $\beta_1 = 0.01$ are the amplitude and the frequency of the sinusoidal function, with $\gamma_1 = 0.3$ be the background firing probability. Binned input and output spike are generated as a Bernoulli random variable with the probability p_x^k.

Next, as for the nonlinear output spike train, we use a sinusoidal function $g()$ to convert the input spike probability into output spike probability, since it cannot be fully approximated by the superposition of the lower ordered polynomial functions: $p_y^k = g(p_x^k) = \alpha_2 sin(\beta_2 p_x^k + \gamma_2) + \delta$, where $\alpha_2 = 0.25$ denotes amplitude, $\beta_2 = 15.71$ and $\gamma_2 = -1.571$ come together to determine the frequency of interactions that arise from different input signals, with $\delta = 0.25$ refers to the background spike probability. An example of synthetic inputs and outputs is shown in Figure 2.

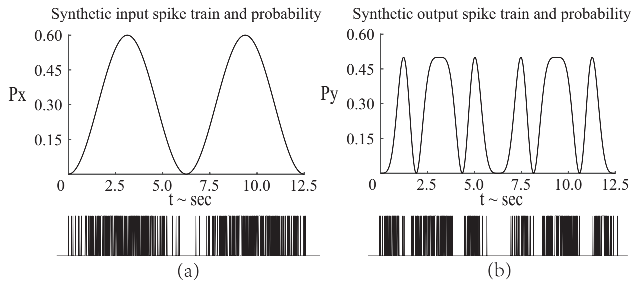

Fig. 2. (a) and (b) are the synthetic input and output spike trains as well as spike probabilities

The training, validation and test set each makes up 60%, 10% and 30% of synthetic spike data and we tune all models on the validation set. As for SNPM, the number of input historical firing times are 20, and for the rest of three models, the length of input binary spikes is all set to 20. Besides, the number of hidden units in SNPM and SPM are also both set to 20.

To analyze the behaviour of the models, Figure 3 indicates the prediction results of four models (SNPM, SPM, GLM and GLVM) on test synthetic data, where black curve in the background is the output spike probability of nonlinear transformation. As a result, SPM and SNPM both match the three peaks of synthetic output probability, while GLM and GLVM fail to cover the left and right peaks.

C. Performance with cortical signals

We use a real spike train data recorded from PMd to M1 cortex area of a rhesus macaque monkey in a behavioral experiment. The monkey was trained to control a joystick and perform a four-direction center-out task. Raw neural data were recorded by a Cerebus Data Acquisition System (Blackrock Mi crosystems, Salt Lake City, UT, U.S.A.) with

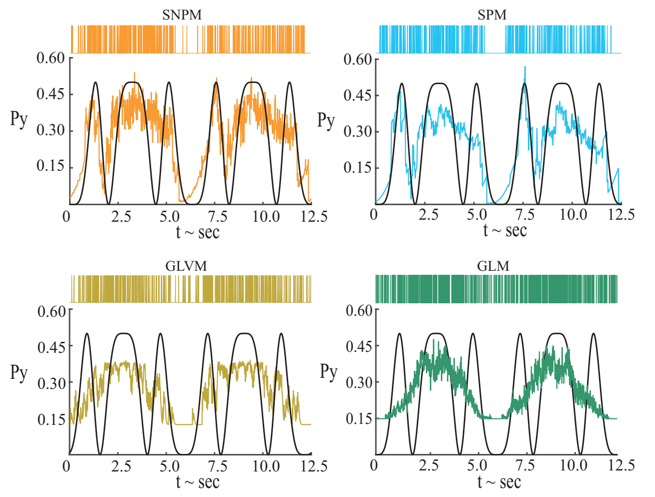

Fig. 3. Predicted firing trains and probability of SNPM, SPM, GLVM and GLM with synthetic output spike probability as the background.

a sampling rate of 30 kHz. The bin size of the spike train was set to be 10 msec such that most intervals will have just one spike [20]. The animal-handling procedures were approved by the Animal Care Committee at Zhejiang University, China, abiding strictly by the Guide for Care and Use of Laboratory Animals (China Ministry of Health). Further details of this data can be found in [20] [21]. Since not all recorded neural activities are related to the movement task [21], spike trains from 54 M1 neurons and their corresponding top 10 PMd neurons are used in this study.

The settings for SNPM and SPM are the same as section III-B, but we choose 6 previous historical firing times as inputs. The predicted spike probabilities of 19_{th} M1 neuron's test data for four models are shown in Figure 4, compared with the actual M1 spike. To give a clear view of predicted probability, the results are smoothed by a normalized Gaussian kernel, and we can see both GLM and GLVM do not match the actual spike rate at all, while SPM and SNPM partially predict the actual spike times.

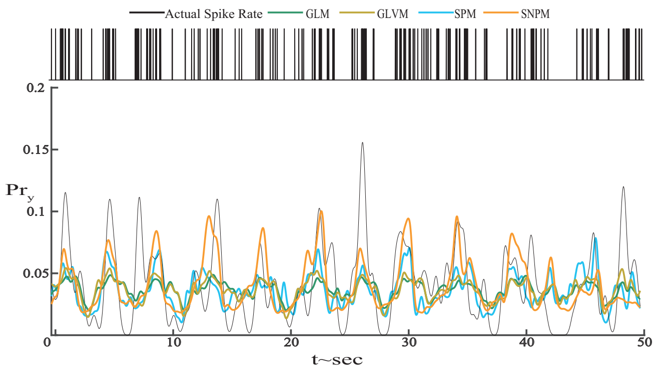

Fig. 4. Comparison of the predicted firing probabilities on the 19_{th} M1 neuron and its actual spike.

Furthermore, we aim to assess the goodness-of-fit. We first propose visualizations of the results of DTR-KS test on both synthetic data and 19_{th} M1 neuron with the DTR-KS curve within the 95% confidence bounds in Figure 5 (other M1 neurons have different 95% confidence bounds, so their DTR-KS curve cannot be averaged and presented in one figure). The averaged DBRs of four models on two types of data are shown in Table 1.

According to Figure 5, we can see that, for both synthetic and real neuron's test data, the DTR-KS curve of the SNPM is close to the 45° black solid line, which further validates that the SNPM statistically models the nonlinear spike transformation well.

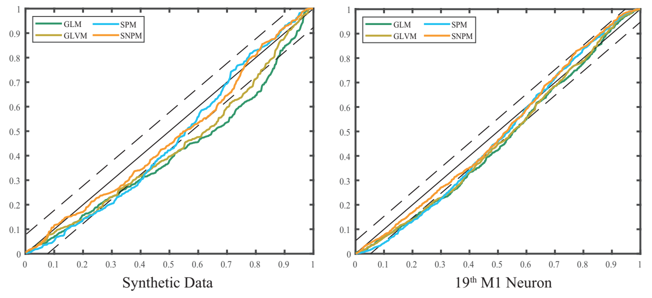

Fig. 5. DTR-KS plots for synthetic test data and the 19_{th} M1 neuron's test data.

Table 1 presents the DBR of the GLM, GLVM, SPM and SNMP on two types of data, where the DBR of real neural data is averaged on all 54 M1 neurons. The results indicate that the DBR of SNPM is much lower than those of GLVM and GLM, and a little bit lower than those of the SPM, which is caused by SPM's unstable performance on parameter setting (local minimum). Thus, SNPM has the most stable performance over SPM with respect to parameter initialisation, and achieves the better goodness-of-fit over GLVM and GLM on modeling nonlinear dynamics.

TABLE I
DISTANCE-BOUND RATIO (DBR) OF FOUR MODELS ON PREDICTED SPIKES

	SNPM	SPM	GLVM	GLM
Synthetic Data	0.89±0.10	0.92±0.14	1.59±0.11	1.90±0.06
Real Data	1.11±0.32	1.22±0.38	1.36±0.55	1.39±0.57

IV. CONCLUSIONS

A real-time application of neuroprosthesis not only requires a accurate nonlinear transformation model but also meets the strict power constraints. The predicted spikes on simulated data and real data both indicate that our spiking neuron point-process is a promising spike train transformation modeling approach for real applications of neuroprosthesis. In the future, the ability to allow online learning of our model is deserved to be explored.

REFERENCES

[1] Dong Song, Rosa HM Chan, Vasilis Z Marmarelis, Robert E Hampson, Sam A Deadwyler, and Theodore W Berger. Nonlinear dynamic modeling of spike train transformations for hippocampal-cortical prostheses. *IEEE Transactions on Biomedical Engineering*, 54(6):1053–1066, 2007.

[2] Theodore W Berger, Robert E Hampson, Dong Song, Anushka Goonawardena, Vasilis Z Marmarelis, and Sam A Deadwyler. A cortical neural prosthesis for restoring and enhancing memory. *Journal of neural engineering*, 8(4):046017, 2011.

[3] Il Memming Park, Sohan Seth, Antonio RC Paiva, Lin Li, and Jose C Principe. Kernel methods on spike train space for neuroscience: a tutorial. *IEEE Signal Processing Magazine*, 30(4):149–160, 2013.

[4] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[5] Ming Li, Haibo Ruan, Yu Qi, Tiantian Guo, Ping Wang, and Gang Pan. Odor recognition with a spiking neural network for bioelectronic nose. *Sensors*, 19(5):993, 2019.

[6] Ziru Wang, Jiawen Liu, Yongqiang Ma, Badong Chen, Nanning Zheng, and Pengju Ren. Perturbation of spike timing benefits neural network performance on similarity search. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[7] Dong Song, Brian S Robinson, Robert E Hampson, Vasilis Z Marmarelis, Sam A Deadwyler, and Theodore W Berger. Sparse large-scale nonlinear dynamical modeling of human hippocampus for memory prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(2):272–280, 2016.

[8] Cunle Qian, Xuyun Sun, Shaomin Zhang, Dong Xing, Hongbao Li, Xiaoxiang Zheng, Gang Pan, and Yiwen Wang. Nonlinear modeling of neural interaction for spike prediction using the staged point-process model. *Neural computation*, 30(12):3189–3226, 2018.

[9] Julie Dethier, Paul Nuyujukian, Stephen I Ryu, Krishna V Shenoy, and Kwabena Boahen. Design and validation of a real-time spiking-neural-network decoder for brain–machine interfaces. *Journal of neural engineering*, 10(3):036008, 2013.

[10] Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

[11] Yu Qi, Bin Liu, Yueming Wang, and Gang Pan. Dynamic ensemble modeling approach to nonstationary neural decoding in brain-computer interfaces. *arXiv preprint arXiv:1911.00714*, 2019.

[12] Steven K Esser, Paul A Merolla, John V Arthur, Andrew S Cassidy, Rathinakumar Appuswamy, Alexander Andreopoulos, David J Berg, Jeffrey L McKinstry, Timothy Melano, Davis R Barch, et al. Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the national academy of sciences*, 113(41):11441–11446, 2016.

[13] Věra Krková. Kolmogorov's theorem is relevant. *Neural computation*, 3(4):617–622, 1991.

[14] Martin Fodslette Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.

[15] Bogdan M Wilamowski and Hao Yu. Improved computation for levenberg–marquardt training. *IEEE transactions on neural networks*, 21(6):930–937, 2010.

[16] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015.

[17] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

[18] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, pages 1–8. ieee, 2015.

[19] Robert Haslinger, Gordon Pipa, and Emery Brown. Discrete time rescaling theorem: determining goodness of fit for discrete time statistical models of neural spiking. *Neural computation*, 22(10):2477–2506, 2010.

[20] Yiwen Wang, António RC Paiva, José C Príncipe, and Justin C Sanchez. Sequential monte carlo point-process estimation of kinematics from neural spiking activity for brain-machine interfaces. *Neural computation*, 21(10):2894–2930, 2009.

[21] Kai Xu, Yiwen Wang, Yueming Wang, Fang Wang, Yaoyao Hao, Shaomin Zhang, Qiaosheng Zhang, Weidong Chen, and Xiaoxiang Zheng. Local-learning-based neuron selection for grasping gesture prediction in motor brain machine interfaces. *Journal of neural engineering*, 10(2):026008, 2013.