Optimizing an Analog Neuron Circuit Design for Nonlinear Function Approximation

Alexander Neckar*, Terrence C. Stewart†, Ben V. Benjamin*, Kwabena Boahen**
{benvb, aneckar, boahen}@stanford.edu tcstewar@uwaterloo.ca
*Electrical Engineering, **Bioengineering and Electrical Engineering, Stanford University, Stanford, CA, U.S.A.
†Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON, Canada

***Abstract*—Silicon neurons designed using subthreshold analog-circuit techniques offer low power and compact area but are exponentially sensitive to threshold-voltage mismatch in transistors. The resulting heterogeneity in the neurons' responses, however, provides a diverse set of basis functions for smooth nonlinear function approximation. For low-order polynomials, neuron spiking thresholds ought to be distributed uniformly across the function's domain. This uniform distribution is difficult to achieve solely by sizing transistors to titrate mismatch. With too much mismatch, many neuron's thresholds fall outside the domain (i.e. they either always spike or remain silent). With too little mismatch, all their thresholds bunch up in the middle of the domain. Here, we present a silicon-neuron design methodology that minimizes overall area by optimizing transistor sizes in concert with a few locally-stored programmable bits to adjust each neuron's offset (and gain). We validated this methodology in a 28-nm mixed analog-digital CMOS process. Compared to relying on mismatch alone, augmentation with digital correction effectively reduced silicon area by 38%.**

***Keywords*—Neural engineering framework, silicon neurons, neuromorphic computing, mixed analog-digital circuits**

I. Neuromorphic Computing

Neuromorphic chips compute by using the heterogeneous input-output functions of their analog neurons as physical computational primitives [1]. Arbitrary computations may be mapped onto this physical substrate using the Neural Engineering Framework (NEF), which assigns encoding and decoding vectors to the neurons, grouped into functional units called ensembles [2]. Encoding vectors define how a vector of continuous signals is encoded in an ensemble's spiking activity. Decoding vectors define how a static or dynamic mathematical transformation of this vector is decoded from an ensemble's spiking activity. This transformation may be performed in a single step by combining decoding and encoding vectors to obtain synaptic weights that connect one ensemble directly to another or back to itself.

Arbitrary nonlinear functions can be approximated with linear weightings, $f(\mathbf{x}) = \sum_{i=1}^{N} a_i(\mathbf{x})\mathbf{d}_i$, of the neuronal *tuning curves*, $a_i(\mathbf{x})$, where the decoders, $\mathbf{d}_i$, are found by optimization methods tailored to analog neurons [3], [4]. Given an input vector $\mathbf{x}$ of dimension D, the NEF defines a tuning curve as $a_i(\mathbf{x}) = G_i(\alpha_i \mathbf{e}_i^{\mathrm{T}}\mathbf{x} + \beta_i)$, where a is the spike rate, G is the neuronal nonlinearity, $\mathbf{e}$ is the D-dimensional encoder (unit length), α is a constant gain, β is a constant bias, and i indexes the N neurons. The *space* of functions that can be linearly decoded from such an ensemble is spanned by left singular-vectors of the $Q \times N$ tuning-curve matrix (each of its N columns is a tuning curve sampled at Q values of $\mathbf{x}$). In particular, these vectors are an orthonormal basis for the function space. Therefore, the error in approximating them measures the quality of an ensemble's function approximation.

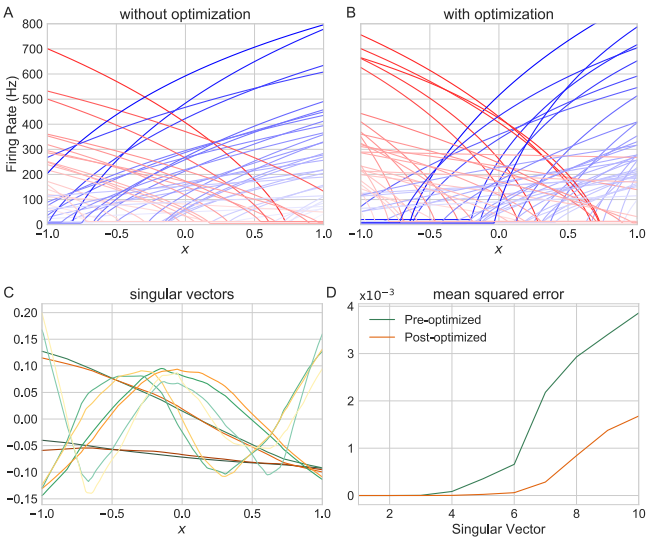

Fig. 1. A,B: Tuning curves of 64 simulated silicon-neurons before (A) and after (B) optimization. C: The ensemble's first five left singular-vectors before and after optimization. D: Optimization significantly lowers the mean squared error (MSE) when decoding the first ten singular vectors.

The threshold distribution of an ensemble's tuning-curves is a key determinant of how well it approximates functions. In particular, a uniform distribution across the domain is ideal for approximating smooth nonlinear functions (i.e. low-order polynomials). Such a distribution is difficult to achieve by relying on transistor-mismatch alone [5]-[11]. In this paper, we focus on optimizing the threshold distribution by titrating transistor mismatch and augmenting it with digital correction (Fig. 1). Section II describes our model of transistor-mismatch's effect on the tuning curves. Section III uses this model to characterize the baseline circuit (without optimization). Section IV describes our procedure for co-optimizing transistor-sizing and digital-correction; it also reports the resulting savings in silicon area. Section V concludes the paper.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

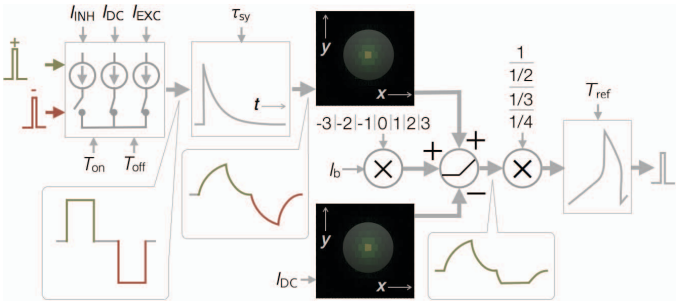

Fig. 2. The neuron is comprised of—from left to right—a pulse-extender, a lowpass temporal filter (synapse), a pair of lowpass spatial filters shared with the other neurons (diffusor network), a rectifier with programmable bias and gain, and a relaxation oscillator (soma).

II. Silicon-Neuron Response Curve

The neuron receives excitatory or inhibitory spikes, converts them into current pulses (pulse-extender), filters these pulses temporally (synapse) as well as spatially (diffusor network), and converts the resulting current into a spike train (soma) (Fig. 2). All these operations are subject to transistor-mismatch, which may be modeled by current-gain parameters (denoted by Λ_i for transistor i) that are log-normally distributed with median equal to one.

A. Pulse-Extender and Synapse

Our pulse-extender accepts an excitatory or inhibitory spike and, after a nominal delay $T_{\rm off}$, generates a current pulse, $I_{\rm P}(t)$, with nominal duration $T_{\rm on}$ and nominal amplitude $I_{\rm EXC} = I_{\rm DC} + \Delta I$ or $I_{\rm INH} = I_{\rm DC} - \Delta I$. Thus, if the spike arrives at time 0, $I_{\rm P}(t) = I_{\rm DC} \pm \Delta I$ from $\Lambda_{\rm off}T_{\rm off}$ to $\Lambda_{\rm off}T_{\rm off} + \Lambda_{\rm on}T_{\rm on}$, where $\Lambda_{\rm off}$ and $\Lambda_{\rm on}$ capture the overall mismatch between different pulse-extenders. Outside this interval, $I_{\rm P}(t)$ is nominally equal to $I_{\rm DC}$. When excitatory $(+)$ or inhibitory $(-)$ spikes arrive with interspike interval $T_{\rm spk}$, $I_{\rm P}(t)$'s mean value is $\frac{\Lambda_{\rm on}T_{\rm on}}{T_{\rm spk}}(I_{\rm DC} \pm \Delta I)$. It saturates at $\frac{\Lambda_{\rm on}T_{\rm on}}{\Lambda_{\rm on}T_{\rm on}+\Lambda_{\rm off}T_{\rm off}}(I_{\rm DC} \pm \Delta I)$ when $T_{\rm spk} < \Lambda_{\rm on}T_{\rm on} + \Lambda_{\rm off}T_{\rm off}$.

Our synapse's circuit-design is similar to [3], [12]. Accounting for transistor-mismatch, its output, $I_{\rm sy}(t)$, obeys

$$\tau_{\rm sy}\frac{dI_{\rm sy}}{dt} + (\Lambda_6 - \Lambda_5)I_{\rm sy}(t) = \frac{\Lambda_2\Lambda_4}{\Lambda_1\Lambda_3}\Lambda_5 I_{\rm P}(t) \qquad (1)$$

where $\tau_{\rm sy}$ is its time-constant. Λ_{1-4} correspond to $\rm M_{L1-4}$ in [3]'s Fig. 3, while Λ_5 and Λ_6, which have median 1 and 2, respectively, correspond to $\rm I_{\tau_2}$ and $\rm I_{\tau_1} + I_{\tau_2}$ in [3]'s Fig. 3. In steady-state, $I_{\rm sy}(t) = \frac{\Lambda_2\Lambda_4}{\Lambda_1\Lambda_3}\frac{\Lambda_5}{\Lambda_6-\Lambda_5}I_{\rm P}(t)$. This current is injected into the diffusor network. Each of the pulse-extender's and synapse's programmable parameters ($I_{\rm EXC}$, $I_{\rm INH}$, $I_{\rm DC}$, $T_{\rm on}$, $T_{\rm off}$, and $\tau_{\rm sy}$) share a common global bias.

B. Diffusor network

We use two diffusor networks, one for the synaptic output ($I_{{\rm sy},i}(t)$) and the other for a reference current ($I_{\rm DC}$). The diffusor network emulates current spread in a hexagonal resistive network and supports globally programmable horizontal (between neighboring nodes) and vertical (to ground) conductances [13], [14]. Each conductance is modeled by a single transistor, whose mismatch is captured by its current-gain parameter. We express the fraction $\zeta_{i,j}$ of the current injected into node i that flows to ground at node j in terms of these current-gains. Thus, the two networks' outputs at node j are $\zeta^{\rm sy}_{i,j}I_{{\rm sy},i}(t)$ and $\zeta^{\rm dc}_{i,j}I_{\rm DC}$, for $I_{{\rm sy},i}(t)$ and $I_{\rm DC}$, respectively.

C. Soma

The $j^{\rm th}$ soma's input current $J(t)$ is obtained from $\zeta^{\rm sy}_{i,j}I_{{\rm sy},i}(t)$ and $\zeta^{\rm dc}_{i,j}I_{\rm DC}$, the diffusor-networks' outputs, as follows. A programmable offset current—denoted by $\gamma^{\rm sy}_j I_{\rm b}$ and $\gamma^{\rm dc}_j I_{\rm b}$, respectively—is first added if desired; $I_{\rm b}$'s value is set globally. If added, $\gamma^{\rm sy}_j$ and $\gamma^{\rm dc}_j$ may have one of three values, described by $\gamma^{\rm sy}_j = \sum_{k=1}^{n}\Lambda_{{\rm P}k}$ for $n = 1, 2, 3$, where $\Lambda_{\rm P1-3}$ capture the switched current-sources' transistor mismatch; $\Lambda_{\rm N1-3}$ play a similar role for $\gamma^{\rm dc}_j$. After the difference between the resulting currents is rectified, a programmable gain, ξ_j, is applied. It may have one of four values, described by $\xi_j = \Lambda_{\rm G1}/(\sum_{k=1}^{n}\Lambda_{{\rm G}k})$ for $n = 1, 2, 3, 4$, where $\Lambda_{\rm G1-4}$ capture transistor-mismatch. Thus, we have

$$J(t) = \xi_j \max\left(\sum_i \zeta^{\rm sy}_{i,j}I_{{\rm sy},i}(t) - \zeta^{\rm dc}_{i,j}I_{\rm DC} + \gamma_j I_{\rm b}, I_0\right) \qquad (2)$$

where $\gamma_j = \gamma^{\rm sy}_j - \gamma^{\rm dc}_j$ and I_0 is the (transistor) leakage current.

Our soma's relaxation oscillator is a modified axon-hillock circuit [15]. Our modification adds a logarithmic dependence to the interspike interval's inversely proportional dependence on the steady-state input current J:

$$T_{\rm spk} = \frac{k_0}{\Lambda_{\rm a}J} + \frac{k_1}{\Lambda_{\rm a}J}\ln\left(\frac{1 + \frac{\Lambda_2\Lambda_{\rm d}}{\Lambda_{\rm a}\Lambda_{\rm c}}\frac{k_2}{J}}{1 + \frac{\Lambda_2\Lambda_{\rm d}}{\Lambda_{\rm a}\Lambda_{\rm c}}\frac{k_3}{J}}\right) \qquad (3)$$

The pulse-width, which serves as a refractory period, has a similar dependence on a bias current, $I_{\rm ref}$, fed to the input during this phase:

$$T_{\rm ref} = \frac{m_0}{\Lambda_1 I_{\rm ref}} + \frac{m_1}{\Lambda_1 I_{\rm ref}}\ln\left(\frac{1 + \frac{\Lambda_{\rm b}\Lambda_4}{\Lambda_1\Lambda_3}\frac{m_2}{I_{\rm ref}}}{1 + \frac{\Lambda_{\rm b}\Lambda_4}{\Lambda_1\Lambda_3}\frac{m_3}{I_{\rm ref}}}\right) \qquad (4)$$

$\Lambda_{\rm a-d}$ and Λ_{1-4} capture transistor-mismatch; k_{0-3} and m_{0-3} are constant circuit-design parameters.

III. Baseline Neuron Behavior

In order to present a continuous signal, $x(t)$, to an ensemble of silicon neurons, we convert it into a train of excitatory and inhibitory spikes. For $x(t) = 1$, $\eta f_{\rm max}$ eps (excitatory-spikes per second) will be fed into one group of synapses and $\eta f_{\rm max}$ ips (inhibitory-spikes per second) will be fed into another group of synapses. For $x(t) = -1$, the first group will receive $\eta f_{\rm max}$ ips and the other group will receive $\eta f_{\rm max}$ eps. These rates are linearly interpolated to $x(t) = 0$, where no synapses receive any spikes. The fraction $\eta \approx 0.6$ of $f_{\rm max} = 1/(T_{\rm on} + T_{\rm off})$ that we use is chosen to avoid saturation—even for outliers—as well as to allow $x(t) = \pm 1$ to be encoded by a mix of excitatory and inhibitory spikes (the case during decoding). For multiple continuous signals, $\mathbf{x}(t)$, this same

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

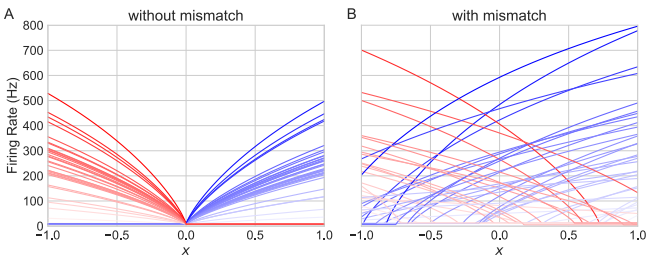

Fig. 3. Spike-rates of 64 simulated silicon-neurons for different inputs x with no mismatch (A) and with mismatch and baseline transistor sizes (B).

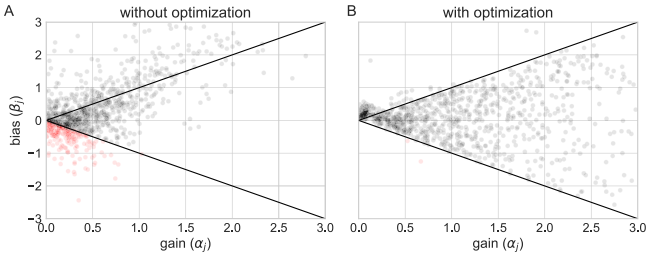

Fig. 4. Distribution of effective gains and biases across 16 ensembles with 64 simulated silicon-neurons each (1,024 samples in total). A: With baseline-sized, mismatched, transistors. B: After optimizing transistor-sizing and bias-bit and gain-bit settings. Neurons above the upper line always spike, neurons below the lower line (red) never spike, and neurons between the lines cross threshold somewhere in the $[-1, 1]$ range (good fraction).

process applies to each dimension, with each signal's spike-train targeting its own (non-overlapping) set of synapses.

These spikes are extended and filtered, producing current that the diffusor network spreads and feeds into each soma, similar to [5]. The j^{th} one's net positive input is

$$J = \xi_j \sum_i (\mathbf{1}_+ - \mathbf{1}_-)\zeta^{\text{sy}}_{i,j}\Delta J_i x + (\mathbf{1}_+ + \mathbf{1}_-)\zeta^{\Delta}_{i,j} I_{\text{DC}} + \gamma_j I_{\text{b}} \quad (5)$$

where $\mathbf{1}_+[i]$ and $\mathbf{1}_-[i]$ are indicator functions for the two groups of synapses, $\Delta J = \eta \Lambda_{\text{sy}} \Lambda_{\text{PG}} \Delta I$, and $\zeta^{\Delta} = \Lambda_{\text{sy}}\zeta^{\text{sy}} - \zeta^{\text{dc}}$. Here, $\Lambda_{\text{PG}} = \frac{\Lambda_{\text{on}} T_{\text{on}}}{\Lambda_{\text{on}} T_{\text{on}} + \Lambda_{\text{off}} T_{\text{off}}}$ and $\Lambda_{\text{sy}} = \frac{\Lambda_2 \Lambda_4}{\Lambda_1 \Lambda_3} \frac{\Lambda_5}{\Lambda_6 - \Lambda_5}$ are the gains of the pulse-extender and synapse, respectively.

A. Behavior Without Mismatch

To demonstrate that some mismatch is necessary for function-approximation, we use the silicon-neuron model (see Section II) to simulate 64 neurons receiving a single continuous signal, $x(t)$, with no transistor-mismatch (Fig. 3A). Differences among their responses arise solely from the different distances that synaptic input has to spread through the diffusor network to reach them. For $x(t) = 0$, they will not spike. For $x(t) > 0$, those closer to synapses receiving excitatory spikes will spike; the more positive $x(t)$ is, the more quickly they spike. For $x(t) < 0$, the second set of synapses will now be receiving excitatory spikes, and neurons closer to them will now spike. Without mismatch, all the neurons' tuning curves are scaled versions of each other. Consequently, the ensemble is nearly worthless for function-approximation.

B. Behavior With Mismatch

If we include transistor mismatch in our simulation, we obtain heterogeneous tuning-curves that are useful for approximating functions (Fig. 3B). The most useful neurons are those that start spiking somewhere inside the input range. Those that spike throughout the input range are not as useful. And those that do not spike at any input value are useless. While identifying these three groups of neurons is straightforward when the ensemble is presented with a single continuous signal, this task becomes unwieldy when it is presented with multiple continous signals (i.e. $\mathbf{x}(t)$ is multidimensional).

To easily identify the three groups of neurons, even for a multidimensional ensemble, we introduce the *cone plot*. This plot visualizes an ensemble's distribution of gains (α_j) and biases (β_j). To generate it, we compute the transformation needed to convert each tuning curve into an ideal tuning curve. If the ideal neuron's response to its input current J is $G_{\text{ideal}}(J)$, we find the j^{th} neuron's effective gain α_j and bias β_j such that its tuning curve $G_j(x) = G_{\text{ideal}}(\alpha_j \mathbf{e}_j^{\text{T}} \mathbf{x} + \beta_j)$. For $x(t)$ one-dimensional, $e = \pm 1$ (e.g., the neurons on the right ($+1$) and on the left (-1) in Fig. 3A). Referring back to J's expression, we find that

$$\alpha_j = \xi_j \left| \sum_i (\mathbf{1}_+ - \mathbf{1}_-)\zeta^{\text{sy}}_{i,j} \eta \Lambda_{\text{sy},i} \Lambda_{\text{PG},i} \Delta I \right| \quad (6)$$

$$\beta_j = \xi_j \left(\sum_i (\mathbf{1}_+ + \mathbf{1}_-)(\Lambda_{\text{sy},i}\zeta^{\text{sy}}_{i,j} - \zeta^{\text{dc}}_{i,j}) I_{\text{DC}} + \gamma_j I_{\text{b}} \right) \quad (7)$$

Notice that, for $e = +1$ or -1, the neuron spikes if $x > -\beta_j/\alpha_j$ or $x < \beta_j/\alpha_j$, respectively. Therefore, its threshold does not depend on ξ_j, the programmable gain.

The three groups of neurons fall neatly into the cone plot's three regions (Fig. 4A). For our baseline transistor-sizing (see below), only 48.2% of the ensemble is in the *good* group (i.e. $-1 < \beta_j/\alpha_j < 1$). β_j/α_j's spread—see the expressions above—arises mainly from the synapses' mismatched gains ($\Lambda_{\text{sy},i}$), which are determined by four transistors.[1] Therefore, we explored the trade-off between sizing-up these transistors versus adding switched-current sources to increase the number of programmable offset-current levels (γ_j) that we can choose from to push neurons above the cone down ($\gamma_j < 0$) and push those below the cone up ($\gamma_j > 0$).

IV. Optimization

To find the optimal trade-off between allocating area to the synapse's transistors or allocating it to switched-current sources, we swept these quantities simultaneously, measuring the good fraction of the ensemble for a variety of configurations. On one hand, quadrupling a transistor's channel-area only halves the standard deviation of the Gaussian underlying its current-gain's log-normal distribution [16]. On the other hand, with $n - 1$ equally-sized switched current-sources, $\log_2(n)$ bits programmed into SRAM can be translated into $n - 1$ levels of current (or gains).

[1]That is, Λ_2, Λ_3, Λ_5, & Λ_6. Λ_1 arises from a transistor in a global-bias circuit and Λ_4 is identical to preserve symmetry.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

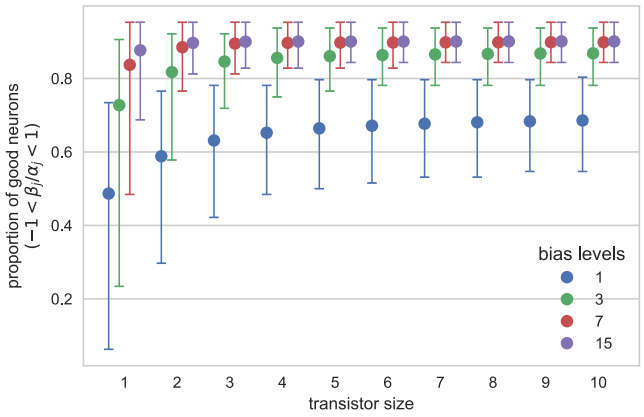

Fig. 5. Good fraction versus width of four key transistors (normalized to baseline). This fraction increases dramatically as the number of offset-current levels the optimization algorithm chooses from increases. Whiskers indicate the 2.5^{th} and 97.5^{th} percentile of the distribution of fractions of good neurons across 2,000 samples of 64-neuron ensembles.

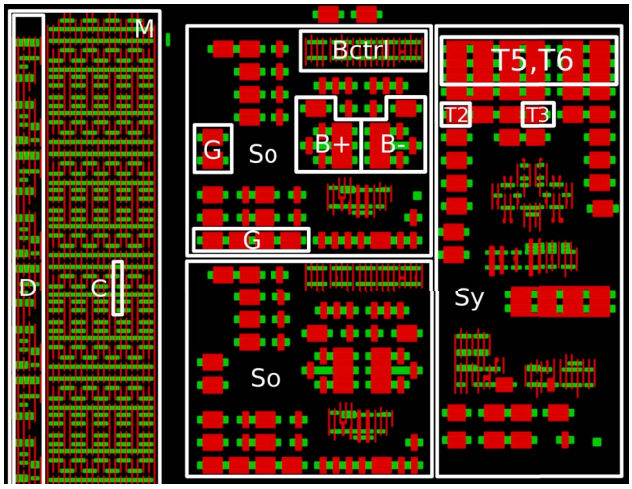

Fig. 6. Transistor Layout. M: Portion of SRAM showing 64 bits shared by eight somas and two synapses. The row decoder (D) and an 8T bit-cell (C) are outlined. So: Soma with programmable gain (G), bias (B+ & B-), and associated logic (Bctrl) outlined. Sy: Synapse with area increases related to Λ_5 & Λ_6 (T5, T6) and Λ_2 & Λ_3 (T2, T3) outlined. AER circuitry is not shown. Note the relative sizes of analog (thick-oxide) and digital (thin-oxide) transistors in this 28-nm process.

A. Transistor Sizing and Programmable Bias-Levels

We program γ_j, the j^{th} neuron's bias, as follows. Given $n-1$ levels to choose from, we select the one that yields $-1 < \beta_j/\alpha_j < 1$ (i.e. inside the cone). If more than one satisfy this requirement, we choose randomly among them. If none satisfy it, each level either satisfies $\beta_j/\alpha_j \leq -1$ (never spikes for $e = +1$) or $\beta_j/\alpha_j \geq 1$ (always spikes for $e = +1$). Of the former, all choices are useless. Of the latter, we choose the lowest level, which gives us access to the steepest and most nonlinear part of the tuning-curve. Since the neuron's programmable gain, ξ_j, does not affect the β_j/α_j ratio, we program it independently to obtain maximum spike-rates between 100 and 1,000 spike/s, choosing randomly if more than one gain-level satisfies this requirement.

By simultaneously optimizing the synapse's transistor-sizing and the number of programmable offset-current levels $(n-1)$, we found that more than seven levels (including zero) gives diminishing returns (Fig. 5). Furthermore, seven levels improve yield to such a degree that more than doubling the baseline-width ($W = 160$ nm) does not provide significant improvement; lengths are kept at baseline ($L = 450$ nm). Doubling the width left room to spare in the layout, as its area was dominated by capacitors, which were implemented in the metal layers. Thus, we were at liberty to quadruple the width of the two transistors that contribute to τ_{sy}'s mismatch (i.e. to Λ_5 & Λ_6). With all these optimizations, the yield increased from $Y = 0.482$ to $\tilde{Y} = 0.896$ (Fig. 4B). If we consider the distribution's 2.5^{th} percentile, instead of its mean, the yield increases from 0.062 to 0.812—even more dramatic.

B. Silicon Area Savings

We compute the effective area our optimized design needs to yield a fixed number of good neurons and compare it to that an unoptimized version of the design needs. In our fabricated layout, four somas share a synapse and sixteen neurons share a SRAM and an AER transceiver (Fig. 6). Hence, the optimized design's area per neuron (μm^2) is

$$\begin{aligned}\tilde{A}_{\text{neu}} &= A_{\text{soma}} + A_{\text{syn}}/4 + (A_{\text{SRAM}} + A_{\text{AER}})/16 \\ &= 27.7 + 42.2/4 + (78.8 + 566.5)/16 = 78.6 \end{aligned} \quad (8)$$

To compute the unoptimized design's area per neuron, we subtract the area sizing-up the synapse's transistors takes as well as the area programmable gain- and offset-circuitry takes. Four switched current-sources controlled by two bits implement four gains and six switched current-sources controlled by three bits implement six nonzero offsets. Hence, we have

$$\begin{aligned}A_{\text{neu}} &= \tilde{A}_{\text{neu}} - \Delta A_{\text{syn}}/4 - A_{\text{gain}} - A_{\text{bias}} - (2+3)A_{\text{bit}} \\ &= 78.6 - 1.15 - 6.22 - 5 \times 0.616 = 68.2 \end{aligned} \quad (9)$$

Thus, optimization added $10.4\ \mu\text{m}^2$.

The two versions' effective areas are $A_{\text{neu}}/Y = 141.5\ \mu\text{m}^2$ and $\tilde{A}_{\text{neu}}/\tilde{Y} = 87.7\ \mu\text{m}^2$, since we have to fabricate $1/Y$ neurons to get one good neuron. Thus, the effective-area saving is $S = (A_{\text{neu}}/Y - \tilde{A}_{\text{neu}}/\tilde{Y})/(A_{\text{neu}}/Y) = 0.38$.

V. Conclusion

We showed how the quality of a silicon-neuron ensemble's function-approximation can be enhanced by co-optimizing transistor-sizing (to titrate mismatch) and programmable offset-current levels (to rescue outliers). Our solution reduced silicon area by 38% while tightening the yield distribution considerably. Our approach is readily applicable to ensembles that receive multiple continuous signals, thus supporting multidimensional function approximation.

Acknowledgment

This work was supported by ONR grants N000141310419 and N000141512827.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

REFERENCES

[1] K. Boahen, "A Neuromorph's Prospectus", *Computing in Science & Engineering*, vol. 19, no. 2, pp. 14-28, 2017.

[2] C. Eliasmith and C. Anderson, *Neural Engineering: Computation, representation, and dynamics in neurobiological systems*, MIT Press, Cambridge MA, 2003.

[3] A. R. Voelker, B. V. Benjamin, T. C. Stewart, K. Boahen and C. Eliasmith, "Extending the neural engineering framework for nonideal silicon synapses", *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4, Baltimore MD, May 2017. doi: 10.1109/ISCAS.2017.8050810

[4] E. Kauderer-Abrams, A. Gilbert, A. Voelker, B. Benjamin, and T. C. Stewart, and K. Boahen, "A Population-Level Approach to Temperature Robustness in Neuromorphic Systems", *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4, Baltimore MD, May 2017. doi: 10.1109/ISCAS.2017.8050985

[5] S. Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Eliasmith, and K. Boahen, "Silicon Neurons that Compute", *ICANN 2012: 22nd International Conference on Artificial Neural Networks*, pp. 121-128, Lausanne, Switzerland, September 2012. doi: 10.1007/978-3-642-33269-2_16

[6] A. Basu, S. Shuo, H. Zhou, M. H. Lim, and G.-B. Huang, "Silicon spiking neurons for hardware implementation of extreme learning machines", *Neurocomputing*, vol. 102, pp. 125-134, February 2013.

[7] S. Menon, S. Fok, A. Neckar, O. Khatib and K. Boahen, "Controlling articulated robots in task-space with spiking silicon neurons", *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 181-186, Sao Paulo BRAZIL, August 2014, doi: 10.1109/BIOROB.2014.6913773

[8] F. Corradi, C. Eliasmith, and G. Indiveri, "Mapping arbitrary mathematical functions and dynamical systems to neuromorphic VLSI circuits for spike-based neural computation, *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 269-272, June 2014.

[9] O. Richter, R. F. Reinhart, S. Nease, J. Steil, and E. Chicca, "Device mismatch in a neuromorphic system implements random features for regression, *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, pp. 1-4, October 2015.

[10] C. S. Thakur, R. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A low power trainable neuromorphic integrated circuit that is tolerant to device mismatch", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 2, pp. 211-221, 2016.

[11] C. S. Thakur, R. Wang, T. J. Hamilton, R. Etienne-Cummings, J. Tapson and A. van Schaik, "An Analogue Neuromorphic Co-Processor That Utilizes Device Mismatch for Learning Applications", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. xx, no. yy, pp. 1-11, 2017. doi: 10.1109/TCSI.2017.2756878

[12] W. Himmelbauer and A. G. Andreou, "Log-domain circuits in subthreshold MOS, *Proceedings of the 40th IEEE Midwest Symposium on Circuits and Systems*, vol. 1. pp. 26-30, 1997.

[13] K. A. Boahen and A. G. Andreou, "A Contrast Sensitive Silicon Retina with Reciprocal Synapses", *Advances in Neural Information Processing Systems*, J. E. Moody and R. P. Lippmann, Eds., vol. 4, pp. 764-772, Morgan Kaufmann, San Mateo CA, 1992.

[14] A. G. Andreou and K. A. Boahen, "Translinear circuits in subthreshold MOS", *Analog Integr. Circuits Signal Process.*, vol. 9, no. 2, pp. 141-166, 1996.

[15] C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, Boston MA, 1989.

[16] A. Pavasović, A. G. Andreou, and C. R. Westgate, "Characterization of subthreshold MOS mismatch in transistors for VLSI systems", *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 8, no. 1, pp. 75-85, 1994. doi: 10.1007/BF02407112

978-1-5386-4881-0/18/$31.00 ©2018 IEEE