A Comparison of Absolute and Relative Neural Encoding Schemes in Addition and Subtraction Functional Subnetworks

Cody Scharzenberger(✉) and Alexander Hunt

Portland State University, Portland, OR 97201, USA
cscharz2@pdx.edu

Abstract. As neural networks have become increasingly prolific solutions to modern problems in science and engineering, there has been a congruent rise in the popularity of the numerical machine learning techniques used to design them. While numerical methods are highly generalizable, they also tend to produce unintuitive networks with inscrutable behavior. One solution to the problem of network interpretability is to use analytical design techniques, but these methods are relatively underdeveloped compared to their numerical alternatives. To increase the utilization of analytical techniques and eventually facilitate the symbiotic integration of both design strategies, it is necessary to improve the efficacy of analytical methods on fundamental function approximation tasks that can be used to perform more complex operations. Toward this end, this manuscript extends the design constraints of the addition and subtraction subnetworks of the functional subnetwork approach (FSA) to arbitrarily many inputs, and then derives new constraints for an alternative neural encoding/decoding scheme. This encoding/decoding scheme involves storing information in the activation ratio of a subnetwork's neurons, rather than directly in their membrane voltages. We show that our new "relative" encoding/decoding scheme has both qualitative and quantitative advantages compared to the existing "absolute" encoding/decoding scheme, including helping to mitigate saturation and improving approximation accuracy. Our relative encoding scheme will be extended to other functional subnetworks in future work to assess its advantages on more complex operations.

Keywords: Neural Encoding Schemes · Functional Subnetwork Approach · Analytical Network Design Methods

1 Introduction

Over the past decade, there has been an explosion of academic interest in neural networks, both in their capacity as biological computational units and as

Supported by NSF DBI 2015317 as part of the NSF/CIHR/DFG/FRQ/UKRI-MRC Next Generation Networks for Neuroscience Program.

F. Meder et al. (Eds.): Living Machines 2023, LNAI 14158, pp. 193–205, 2023.
https://doi.org/10.1007/978-3-031-39504-8_13

potential solutions to a plethora of problems across disparate fields of scientific inquiry. Accompanying the precipitous ascendance of neural network research, there has been an escalation in the quantity and quality of techniques used in their design and training. Since the advent of backpropagation [2] made it possible to train networks comprised of multiple layers of neurons, techniques such as stochastic gradient descent (SGD) [1] and its many variations (e.g., Adagrad [5], Adam [4], etc.) have cemented numerical methods as the dominant approach to tuning network parameters. While numerical methods are excellent in terms of their ease of application and scalability, especially given modern advances in graphical processing units (GPU) [3], their key limitation is that they tend to produce networks whose decision making and computational processes are inscrutable. This makes them inappropriate for designing networks for applications where transparency is essential, such as when building high fidelity biological models. Fortunately, some analytical techniques for designing interpretable neural networks do exist (e.g., the functional subnetwork approach (FSA) [6,7]), but these methodologies remain relatively underdeveloped and need additional investigation to facilitate their broader utilization. Among the many open questions concerning analytical techniques, one of particular interest is how best to encode information in these networks to facilitate their design and enhance their approximation accuracy. After all, different encoding schemes require different network design constraints and may be more or less appropriate depending on the application. Since the existing work of the FSA is currently limited to a single encoding scheme that represents information directly in the membrane voltages of the neurons, this work extends these techniques to a new encoding scheme: one that stores information in neural activation ratios. While potentially less intuitive, such an encoding scheme is prudent since it allows for the simple integration of multiple subnetworks that might other operate over significantly different representational domains.

1.1 Our Contribution

In this manuscript we derive simple, yet novel analytical design rules for creating functional subnetworks that encode information in neural activation ratios rather than directly in membrane voltages. Our analysis indicates that this "relative" information encoding scheme has a variety of qualitative advantages (e.g., saturation prevention, biological plausibility, etc.), while also improving the approximation accuracy of the resulting subnetworks. For the purpose of generating quantitative results, we use our unique design rules to build small functional subnetworks that perform addition and subtraction operations. We then compare the design rules and performance of our custom "relative" addition and subtraction subnetworks to the "absolute" variations of the FSA [6] to emphasize the advantages of our approach. In this way, the work discussed here serves as an alternative to the non-spiking FSA formulation, with advantages that make it easier to use for some problems while offering improved accuracy.

2 Background

The two fields of information that are required to understand our work on encoding schemes in functional subnetworks are those pertaining to neuron modeling and the existing non-spiking FSA design rules.

2.1 Neuron Model

We use the same rate-based LIF model as [6] to facilitate direct comparisons with their results. Suppose that we want to analyze the behavior of a system of $n \in \mathbb{N}$ neurons over some time domain $\mathcal{T} = [0, T_f]$ where $T_f \in \mathbb{R}_{>0}$ is the final time of interest. Then the membrane voltages of these neurons form a first order dynamical system comprised of $n \in \mathbb{N}$ state variables $U_i \in \mathbb{R}$ that satisfy

$$C_{m,i}\dot{U}_i = I_{leak,i} + I_{syn,i} + I_{app,i}, \tag{1}$$

$\forall i \in \mathbb{N}_{\leq n}$, where the leak and synaptic currents are defined as

$$I_{leak,i} = -G_{m,i}U_i, \tag{2}$$

$$I_{syn,i} = \sum_{j=1}^{n} g_{s,ij} \min\left(\max\left(\frac{U_j}{R_j}, 0\right), 1\right)(\Delta E_{s,ij} - U_i), \tag{3}$$

respectively, and the applied currents $I_{app,i} : \mathcal{T} \to \mathbb{R}$ are known functions of time. Throughout this work, the following definitions hold: $U_i \in \mathbb{R}$ is the membrane voltage of the ith neuron with respect to its resting potential, $C_{m,i} \in \mathbb{R}_{>0}$ is the membrane capacitance of the ith neuron, $G_{m,i} \in \mathbb{R}_{>0}$ is the membrane conductance of the ith neuron, $R_j \in \mathbb{R}_{>0}$ is the activation domain of the jth neuron with respect to its resting potential, $g_{s,ij} \in \mathbb{R}_{>0}$ is the maximum synaptic conductance from neuron j to neuron i, and $\Delta E_{s,ij} \in \mathbb{R}$ is the synaptic reversal potential from neuron j to neuron i with respect to neuron i's resting potential. Substituting the leak current Eq. (2) and synaptic current Eq. (3) into the dynamical system Eq. (1) we have the governing equation

$$C_{m,i}\dot{U}_i = -G_{m,i}U_i + \sum_{j=1}^{n} g_{s,ij} \min\left(\max\left(\frac{U_j}{R_j}, 0\right), 1\right)(\Delta E_{s,ij} - U_i) + I_{app,i}. \tag{4}$$

2.2 Functional Subnetwork Approach (FSA)

The functional subnetwork approach (FSA) refers to the analytical methods developed in [6] for designing subnetworks of non-spiking neurons to perform basic tasks, including: (1) signal transfer such as transmission and modulation; (2) arithmetic operations such as addition, subtraction, multiplication, and division; and (3) calculus operations such as differentiation and integration. One of the main attractions of this work lies in the fact that it combines *simple* neural architectures with *analytical* design rules constrained by biological limitations, a combination of features that ensures that the resulting subnetworks are both meaningful and interpretable. For a thorough explanation of the existing FSA design rules refer to [6].

3 Methodology

To begin our analysis of information encoding schemes in addition and subtraction functional subnetworks, we derive the analytical design rules that are necessary to create these subnetworks in the first place. In order to compare the absolute encoding scheme, which stores information in the membrane voltages of a network's neurons, and the relative encoding scheme, which stores information in the percent activation of a network's neurons, we derive the design rules of both approaches using arbitrarily many inputs. Since the absolute encoding scheme is the same as that used in [6], the design rules for our absolute addition and subtraction subnetworks simplify to those of [6] when each subnetwork is assumed to have only two inputs. Similarly, since addition is just a special case of subtraction, we exclusively derive analytical design rules for subtraction subnetworks, because these rules can be simplified to apply to addition subnetworks by removing the inhibitory synapses.

3.1 Subtraction Subnetwork Architecture and Equilibrium

Consider a system of $n \in \mathbb{N}$ neurons, with each of the first $n-1$ neurons connected to the final nth neuron via some combination of excitatory and inhibitory synapses as shown in Fig. 1a. Let $U_i^\star \in \mathbb{R}$ be the steady state membrane voltage of the ith neuron with respect to its resting potential $\forall i \in \mathbb{N}_{\leq n}$. Given this architecture, the steady state membrane voltage of the output neuron $U_n^\star$ can be written in terms of the steady state membrane voltages of the first $n-1$ input neurons by

$$U_n^\star = \frac{\sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{U_i^\star}{R_i}, 0\right), 1\right) \Delta E_{s,ni} + I_{app,n}}{G_{m,n} + \sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{U_i^\star}{R_i}, 0\right), 1\right)}. \tag{5}$$

Equation (5) describes the natural steady state behavior of our subtraction subnetwork given our chosen neuron model and architecture.

3.2 Absolute and Relative Notions of Subtraction

Given the baseline steady state behavior of our subtraction subnetwork in Eq. (5), we now consider how we want our subnetwork to behave for each of our two encoding schemes. For the absolute encoding scheme we want the membrane voltage of the output neuron to be the sum of the membrane voltages of the excitatory input neurons less that of the inhibitory input neurons, scaled by

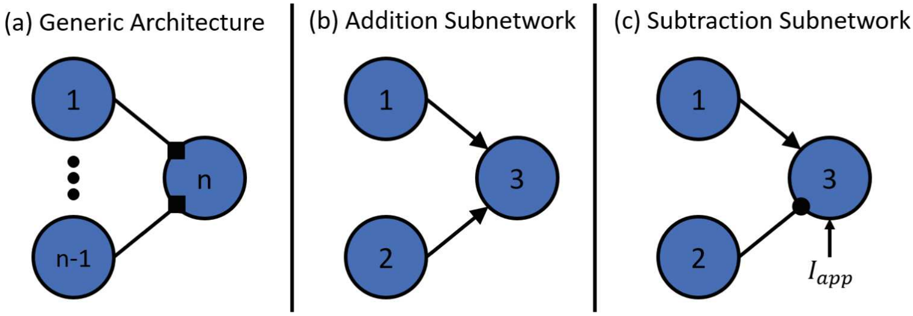

Fig. 1. (a) Generic subtraction subnetwork architecture. (b) Addition subnetwork example. (c) Subtraction subnetwork example. Triangular synapses are excitatory, circular synapses are inhibitory, and square synapses may be either.

some gain $c \in \mathbb{R}_{>0}$. Let $s_i \in \mathcal{S} = \{-1, 1\}$ be the sign associated with each input neuron $\forall i \in \mathbb{N}_{\leq n-1}$, where -1 is assigned to each inhibitory synapse and 1 is assigned to each excitatory synapse. In this case, the steady state membrane voltage of the output neuron should satisfy

$$U_n^{\star} = c \sum_{i=1}^{n-1} s_i U_i^{\star}. \tag{6}$$

While the absolute formulation in Eq. (6) is convenient for directly representing differences, it requires a potentially biologically unfeasible representational domain R_n at the output neuron. Consider an example where there are many more positive inputs than negative inputs in our subtraction subnetwork, such as when approximating an addition operation. When this happens, Eq. (6) requires that the representation domain R_n of the output neuron be the sum of the representational domains of the input neurons in order to prevent the output from saturating prematurely. Yet, as the number of input pathways grows, so too does the required output representational domain R_n, allowing for potentially impractically large values.

To address this problem, we can instead create a relative formulation for desired subtraction subnetwork behavior wherein the steady state activation ratio of the output neuron $\frac{U_n^{\star}}{R_n}$ is the average steady state activation ratio of the excitatory input neurons less that of the inhibitory input neurons, scaled by some gain $c \in \mathbb{R}_{>0}$. Let $n^+, n^- \in \mathbb{N}$ be the number of excitatory and inhibitory subtraction subnetwork inputs, respectively. Similarly, let $i_k^+ \in \mathcal{I}^+ = \{i \in \mathbb{N}_{\leq n-1} : s_i = 1\}$ and $i_k^- \in \mathcal{I}^- = \{i \in \mathbb{N}_{\leq n-1} : s_i = -1\}$ be the sets of indexes associated with excitatory and inhibitory input neurons, respectively. In this case, the steady state membrane voltage of the output neuron should satisfy

$$U_n^{\star} = cR_n \left(\frac{1}{n^+} \sum_{k=1}^{n^+} \frac{U_{i_k^+}^{\star}}{R_{i_k^+}} - \frac{1}{n^-} \sum_{k=1}^{n^-} \frac{U_{i_k^-}^{\star}}{R_{i_k^-}} \right). \tag{7}$$

This formulation ensures that the representational domain R_n of the output neuron can be selected as appropriate for the application.

3.3 Universal Subtraction Subnetwork Design

To make Eq. (5) have a similar structure to either Eq. (6) or Eq. (7), several common parameters must be set. Firstly, the external current applied to the output neuron $I_{app,n}$ should be zero in order to eliminate the constant offset from the numerator of Eq. (5). Likewise, the membrane conductance $G_{m,n}$ of the output neuron should be minimized to mitigate its influence on the denominator of Eq. (5), though it can not be completely eliminated and still allow the neuron to behave as a leaky integrator. Finally, in order to emphasize the impact of the membrane voltage terms $U_i^\star$ in the numerator of Eq. (5) compared to those in the denominator, we must maximize the synaptic reversal potentials $\Delta E_{s,ni}$ of the excitatory synapses and minimize those of the inhibitory synapses. While the synaptic reversal potentials $\Delta E_{s,ni}$ are in theory unbounded, they are in practice limited by biological constraints. As such, we use $\Delta E_{s,ni} = 194\,\text{mV}$, which describes calcium ion channels, for excitatory synapses, and $\Delta E_{s,ni} = -40\,\text{mV}$, which describes chloride channels, for inhibitory synapses.

3.4 Absolute Subtraction Subnetwork

For an absolute subtraction subnetwork, the membrane voltage of the nth neuron should satisfy Eq. (6). To achieve this, we start by substituting Eq. (5) into Eq. (6) and solving for the gain c to find

$$c = \frac{\sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{U_i^\star}{R_i}, 0\right), 1\right) \Delta E_{s,ni} + I_{app,n}}{\left(\sum_{i=1}^{n-1} s_i U_i^\star\right)\left(G_{m,n} + \sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{U_i^\star}{R_i}, 0\right), 1\right)\right)}. \tag{8}$$

There are numerous parameters in Eq. (8) that need to be set in order to achieve the desired gain c, most of which are already constrained by the principles discussed in Sect. 3.3. Among the remaining parameters, the most appropriate to adjust to achieve the desired gain c are the $n-1$ maximum synaptic conductances $g_{s,ni}$. However, since there are $n-1$ maximum synaptic conductances $g_{s,ni}$ and only one constraining equation (e.g., Eq. (8)), there are infinitely many sets of maximum synaptic conductances $g_{s,ni}$ that produce the desired gain c for a single steady state input membrane voltage point $\vec{U}^\star = \left[U_1^\star \cdots U_{n-1}^\star\right]^T \in \mathbb{R}^{n-1}$. To constrain the number of solutions, we choose $n-1$ steady state input membrane voltage points $\vec{U}_j^\star \in \mathbb{R}^{n-1}$ at which to enforce the gain relationship in Eq. (8), $\forall j \in \mathbb{N}_{\leq n-1}$. For notational convenience, let $\mathbf{U}^\star \in \mathbb{R}^{(n-1)\times(n-1)}$ be such that $\mathbf{U}^\star = \left[\vec{U}_1^\star \cdots \vec{U}_{n-1}^\star\right]$. Enforcing Eq. (8) at each of the $n-1$ steady state membrane voltage points $\vec{U}_j^\star$ that comprise the columns of $\mathbf{U}^\star$ yields the system of equations

$$c = \frac{\sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{\mathbf{U}^{\star}_{ij}}{R_i}, 0\right), 1\right) \Delta E_{s,ni} + I_{app,n}}{\left(\sum_{i=1}^{n-1} s_i \mathbf{U}^{\star}_{ij}\right)\left(G_{m,n} + \sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{\mathbf{U}^{\star}_{ij}}{R_i}, 0\right), 1\right)\right)}, \tag{9}$$

$\forall j \in \mathbb{N}_{\leq n-1}$. Rearranging Eq. (9) into matrix-vector form and isolating the maximum synaptic conductances $g_{s,ni}$ yields a linear system of equations

$$\mathbf{A}\vec{g}_{s,n} = \vec{b} \tag{10}$$

where $\mathbf{A} \in \mathbb{R}^{(n-1)\times(n-1)}$ such that $\forall i, j \in \mathbb{N}_{\leq n-1}$

$$\mathbf{A}_{ij} = \left(c \sum_{k=1}^{n-1} s_k \mathbf{U}^{\star}_{ki} - \Delta E_{s,nj}\right) \min\left(\max\left(\frac{\mathbf{U}^{\star}_{ji}}{R_j}, 0\right), 1\right), \tag{11}$$

$\vec{g}_{s,n} \in \mathbb{R}^{n-1}$ such that $\vec{g}_{s,n} = \left[g_{s,n1} \cdots g_{s,n(n-1)}\right]^T$, and $\vec{b} \in \mathbb{R}^{n-1}$ such that $\forall i \in \mathbb{N}_{\leq n-1}$

$$\vec{b}_i = I_{app,n} - cG_{m,n} \sum_{k=1}^{n-1} s_k \mathbf{U}^{\star}_{ki}. \tag{12}$$

While any linearly independent choice of $n-1$ steady state input membrane voltage points $\vec{U}^{\star}_j$ would be sufficient for computing a unique set of maximum synaptic conductances $g_{s,ni}$, certain choices are more insightful than others. For example, if we choose $\vec{U}^{\star}_j$ such that $\forall i, j \in \mathbb{N}_{\leq n-1}$

$$\mathbf{U}^{\star}_{ij} = \begin{cases} R_i, & i = j \\ 0, & i \neq j \end{cases} \tag{13}$$

then the system matrix in Eq. (10) becomes diagonal and the associated design requirement for each $g_{s,ni}$ simplifies substantially. To see this, substitute Eq. (13) into Eq. (10) such that the system becomes

$$\begin{bmatrix} cs_1 R_1 - \Delta E_{s,n1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & cs_{n-1} R_{n-1} - \Delta E_{s,n(n-1)} \end{bmatrix} \begin{bmatrix} g_{s,n1} \\ \vdots \\ g_{s,n(n-1)} \end{bmatrix} = \begin{bmatrix} I_{app,n} - cG_{m,n} s_1 R_1 \\ \vdots \\ I_{app,n} - cG_{m,n} s_{n-1} R_{n-1} \end{bmatrix}. \tag{14}$$

Solving the system of equations described by Eq. (14) for each $g_{s,nk}$ yields

$$g_{s,nk} = \frac{I_{app,n} - cs_k G_{m,n} R_k}{cs_k R_k - \Delta E_{s,nk}}, \tag{15}$$

$\forall k \in \mathbb{N}_{\leq n-1}$. The design requirement from Eq. (15) allows for absolute subtraction subnetworks to be designed with arbitrarily many input neurons. Note that choosing a different linearly independent set of steady state membrane voltage points $\vec{U}^{\star}_j$ at which to enforce the gain relationship Eq. 9 will yield a modified variation of the design requirement in Eq. 15 that prioritizes error minimization at those specific points.

3.5 Relative Subtraction Subnetwork

For a relative subtraction subnetwork, the membrane voltage of the nth neuron should satisfy Eq. (7). If we substitute Eq. (5) into Eq. (7) and solve for the gain c we find

$$c = \frac{\sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{U_i^\star}{R_i}, 0\right), 1\right) \Delta E_{s,ni} + I_{app,n}}{R_n \left(\frac{1}{n^+}\sum_{k=1}^{n^+} \frac{U_{i_k^+}^\star}{R_{i_k^+}} - \frac{1}{n^-}\sum_{k=1}^{n^-} \frac{U_{i_k^-}^\star}{R_{i_k^-}}\right)\left(G_{m,n} + \sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{U_i^\star}{R_i}, 0\right), 1\right)\right)}. \tag{16}$$

Following the same procedure as in Sect. 3.4, let $\vec{U}_j^\star = \left[U_{1,j}^\star \cdots U_{n-1,j}^\star\right]^T \in \mathbb{R}^{n-1}$, $\forall j \in \mathbb{N}_{\leq n-1}$ be the $n-1$ steady state membrane voltage points at which we want to achieve Eq. (16). Similarly, let $\mathbf{U}^\star \in \mathbb{R}^{(n-1)\times(n-1)}$ be such that $\mathbf{U}^\star = \left[\vec{U}_1^\star \cdots \vec{U}_{n-1}^\star\right]$. Enforcing Eq. (16) at each of the $n-1$ steady state membrane voltage points $\vec{U}_j^\star$ that comprise the columns of $\mathbf{U}^\star$ yields the system of equations

$$c = \frac{\sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{\mathbf{U}_{ij}^\star}{R_i}, 0\right), 1\right) \Delta E_{s,ni} + I_{app,n}}{\left(R_n \left(\frac{1}{n^+}\sum_{k=1}^{n^+} \frac{\mathbf{U}_{i_k^+ j}^\star}{R_{i_k^+}} - \frac{1}{n^-}\sum_{k=1}^{n^-} \frac{U_{i_k^- j}^\star}{R_{i_k^-}}\right)\right)\left(G_{m,n} + \sum_{i=1}^{n-1} g_{s,ni} \min\left(\max\left(\frac{\mathbf{U}_{ij}^\star}{R_i}, 0\right), 1\right)\right)}, \tag{17}$$

$\forall j \in \mathbb{N}_{\leq n-1}$. To compress the following notation, let

$$C_j = R_n \left(\frac{1}{n^+}\sum_{k=1}^{n^+} \frac{\mathbf{U}_{i_k^+ j}^\star}{R_{i_k^+}} - \frac{1}{n^-}\sum_{k=1}^{n^-} \frac{U_{i_k^- j}^\star}{R_{i_k^-}}\right), \tag{18}$$

$\forall j \in \mathbb{N}_{\leq n-1}$. Rearranging Eq. (17) into matrix-vector form and isolating the maximum synaptic conductances $g_{s,ni}$ yields a linear system of equations

$$\mathbf{A}\vec{g}_{s,n} = \vec{b} \tag{19}$$

where $\mathbf{A} \in \mathbb{R}^{(n-1)\times(n-1)}$ such that $\forall i, j \in \mathbb{N}_{\leq n-1}$

$$\mathbf{A}_{ij} = (cC_i - \Delta E_{s,nj}) \min\left(\max\left(\frac{\mathbf{U}_{ji}^\star}{R_j}, 0\right), 1\right), \tag{20}$$

$\vec{g}_{s,n} \in \mathbb{R}^{n-1}$ such that $\vec{g}_{s,n} = \left[g_{s,n1} \cdots g_{s,n(n-1)}\right]^T$, and $\vec{b} \in \mathbb{R}^{n-1}$ such that $\forall i \in \mathbb{N}_{\leq n-1}$

$$\vec{b}_i = I_{app,n} - cG_{m,n}C_i \tag{21}$$

Choosing the same $n-1$ steady state input membrane voltage points $\vec{U}_j^\star$ as in Eq. (13), the system matrix in Eq. (19) becomes diagonal. Such a choice for the $\vec{U}_j^\star$ simplifies the parameters C_j from Eq. (18) to

$$C_j = \frac{s_j R_n}{n_j^\pm}, \tag{22}$$

where $s_j \in \mathcal{S} = \{-1, 1\}$ is defined as

$$s_j = \begin{cases} -1, & j \in \mathcal{I}^- \\ 1, & j \in \mathcal{I}^+ \end{cases}, \tag{23}$$

and $n_j^{\pm} \in \mathcal{N} = \{n^-, n^+\}$ is defined as

$$n_j^{\pm} = \begin{cases} n^-, & j \in \mathcal{I}^- \\ n^+, & j \in \mathcal{I}^+ \end{cases}, \tag{24}$$

$\forall j \in \mathbb{N}_{\leq n-1}$. The associated system of equations in Eq. (10) then becomes

$$\begin{bmatrix} \frac{cs_1 R_n}{n_1^{\pm}} - \Delta E_{s,n1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{cs_{n-1} R_n}{n_{n-1}^{\pm}} - \Delta E_{s,n(n-1)} \end{bmatrix} \begin{bmatrix} g_{s,n1} \\ \vdots \\ g_{s,n(n-1)} \end{bmatrix} = \begin{bmatrix} I_{app,n} - \frac{cG_{m,n} s_1 R_n}{n_1^{\pm}} \\ \vdots \\ I_{app,n} - \frac{cG_{m,n} s_{n-1} R_n}{n_{n-1}^{\pm}} \end{bmatrix}. \tag{25}$$

Solving Eq. (25) for each $g_{s,nk}$ yields

$$g_{s,nk} = \frac{n_k^{\pm} I_{app,n} - cs_k G_{m,n} R_n}{cs_k R_n - n_k^{\pm} \Delta E_{s,nk}}, \tag{26}$$

$\forall k \in \mathbb{N}_{\leq n-1}$. The design requirement from Eq. (26) allows for relative subtraction subnetworks to be designed with arbitrarily many input neurons. As before, choosing a different linearly independent set of steady state membrane voltage points $\vec{U}_j^{\star}$ at which to enforce the gain relationship Eq. 17 will yield a modified variation of the design requirement in Eq. 26 that prioritizes error minimization at those specific points.

4 Results

After deriving the analytical design rules necessary to build addition and subtraction subnetworks for each information encoding scheme, it is possible to determine how these different approaches impact subnetwork approximation accuracy by applying these techniques to example subnetworks. Toward this end, we employ the design constraints from Sect. 3.3, as well as from Eq. (15) and Eq. (26), to build simple addition and subtraction subnetworks for each encoding scheme. The example addition and subtraction subnetworks that we consider here have the architectures shown in Figs. 1b, 1c, respectively. To ensure that the output neurons of our subtraction subnetworks stay positive throughout the entire input domain, we apply a constant current to the output neuron so that it is tonically excited to half of its maximum value in the absence of inputs.

The steady state results obtained from simulating the two addition subnetworks are represented graphically in Fig. 2, while those associated with the two subtraction subnetworks are displayed in Fig. 3. Both figures are divided into four sections, where plots (a) and (b) show the steady state response of the subnetworks when using the absolute and relative information encoding schemes, respectively; plot (c) shows the steady state approximation error associated with each encoding scheme; and plot (d) shows the difference in steady state approximation error between the two encoding schemes. Comparing plots (a) and (b) from Fig. 2, it is clear that the maximum approximation error for each encoding scheme occurs when the input neurons are maximally active. Plots (a) and (b) from Fig. 2 also indicate that the relative encoding scheme tends to have less error across the input domain than the absolute encoding scheme. This fact is confirmed by plot (c) from Fig. 2 where both the maximum and average approximation error associated with the relative encoding scheme are less than that of the absolute encoding scheme. Finally, plot (d) from Fig. 2 makes the difference in approximation accuracy explicit by showing that the approximation error for relative addition subnetwork never exceeds that of the absolute addition subnetwork.

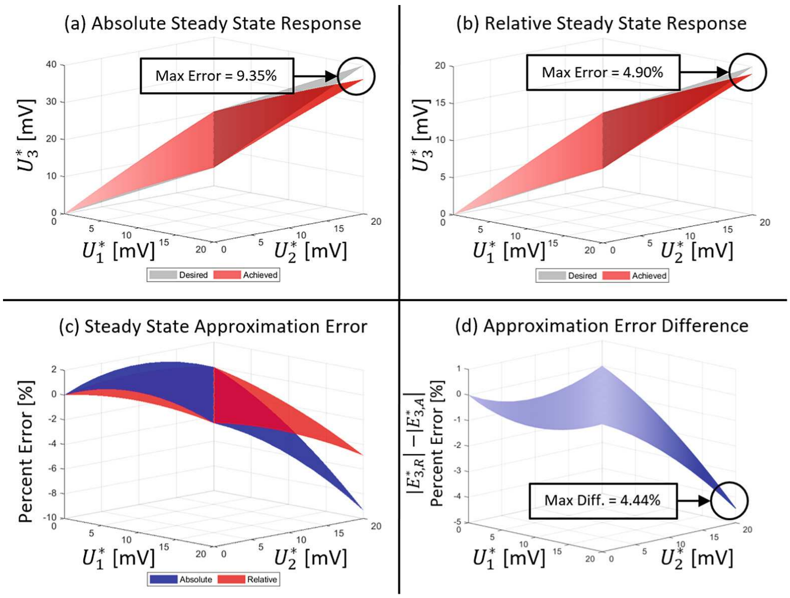

Fig. 2. Addition subnetwork information encoding scheme comparison. (a) Absolute steady state response. (b) Relative steady state response. (c) Steady state approximation error. (d) Difference in approximation error.

Figure 3 tells a similar story concerning the approximation accuracy of the subtraction subnetwork. In this case, plots (a) and (b) from Fig. 3 show that the maximum approximation error occurs for both encoding schemes when the first input neuron value is minimized and the second input neuron value is maximized. Similarly, plot (c) of Fig. 3 indicates that the approximation error of the relative subtraction subnetwork is typically less than that of the absolute subtraction subnetwork, except for when the first input neuron value is small. This relationship is confirmed in plot (d) of Fig. 3, wherein the relative subtraction subnetwork typically outperforms the absolute subtraction subnetwork, but not for the entire input domain.

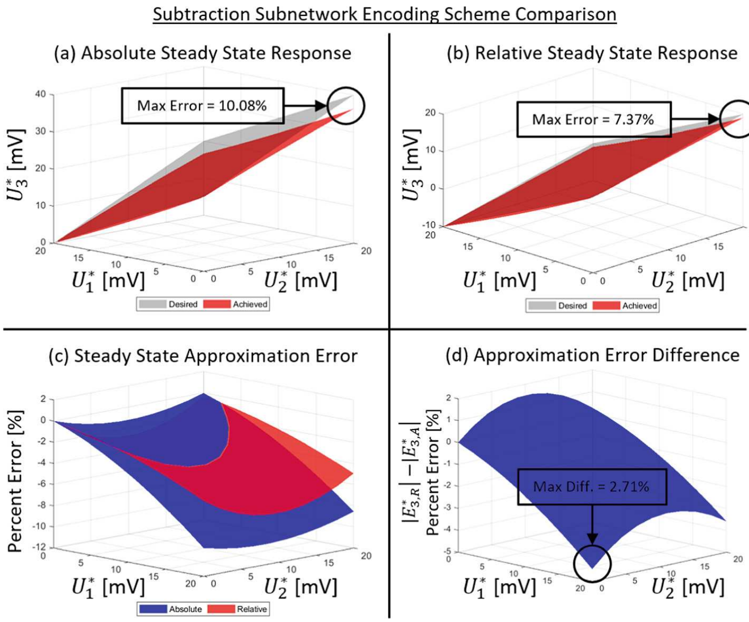

Fig. 3. Subtraction subnetwork information encoding scheme comparison. (a) Absolute steady state response. (b) Relative steady state response. (c) Steady state approximation error. (d) Difference in approximation error.

5 Discussion

Our results indicate that there are both qualitative and quantitative advantages to using a relative information encoding scheme compared to an absolute one. The design constraints derived in Sect. 3 allow the representation domain of the output neuron to be unconstrained when using a relative encoding scheme, meaning that a biologically realistic value may be selected by the designer without risking saturation for expected inputs. Although only addition and subtraction subnetworks are considered here, this feature may be even more salient in the context of certain mathematical operations, such as division, that tend toward infinity under certain input conditions. When this happens, an absolute subnetwork would be unable to represent the infinite output value, but a relative subnetwork would simply encode the result as a maximally active output. In addition to these qualitative benefits, the results shown in Figs. 2 and 3 demonstrate that the relative subnetworks tend to experience less approximation error, though the extent to which this is true is somewhat modest at about 4.44% for addition subnetworks and 2.71% for subtraction subnetworks. While differences in approximation error do vary across the input domain and the operation being approximated, the quantitative benefits that we have observed from using a relative encoding scheme are appreciable on average despite these variations. A logical extension of this work would be to apply our relative encoding scheme to other functional subnetworks in order to determine whether the qualitative network design benefits and quantitative reductions in approximation error are maintained in other contexts as we expect.

6 Conclusions

The functional subnetwork approach (FSA) provides a collection of analytical design tools for building neural networks that approximate basic mathematical operations. This work takes a step toward extending these pre-existing techniques by deriving a simple, yet novel method of encoding information in the activation ratio of the neurons that comprise such subnetworks. We showed that applying the relative encoding scheme presented in this work modestly improves approximation accuracy and allows the designer freedom to chose convenient representation domains. Since continuing to improve existing analytical neural network design techniques is necessary to bring their advantages to bear on a wider variety of modern scientific problems, it is our goal to continue to expand the generality and utility of these methods by applying them to new subnetworks and mathematical operations in future work.

Acknowledgements. The authors acknowledge support by NSF DBI 2015317 as part of the NSF/CIHR/DFG/FRQ/UKRI-MRC Next Generation Networks for Neuroscience Program.

References

1. Bottou, L.: Stochastic gradient descent tricks. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 421–436. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_25
2. Hecht-Nielsen, R.: Theory of the backpropagation neural network. In: Neural Networks for Perception, pp. 65–93. Elsevier (1992). https://linkinghub.elsevier.com/retrieve/pii/B9780127412528500108
3. Keckler, S.W., Dally, W.J., Khailany, B., Garland, M., Glasco, D.: GPUs and the future of parallel computing. IEEE Micro **31**(5), 7–17 (2011). http://ieeexplore.ieee.org/document/6045685/
4. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 [cs] (2017)

5. Mukkamala, M.C., Hein, M.: Variants of RMSProp and Adagrad with Logarithmic Regret Bounds (2017)
6. Szczecinski, N.S., Hunt, A.J., Quinn, R.D.: A functional subnetwork approach to designing synthetic nervous systems that control legged robot locomotion. Front. Neurorobot. **11** (2017). http://journal.frontiersin.org/article/10.3389/fnbot.2017.00037/full
7. Szczecinski, N.S., Quinn, R.D., Hunt, A.J.: Extending the functional subnetwork approach to a generalized linear integrate-and-fire neuron model. Front. Neurorobot. **14**, 577804 (2020). https://www.frontiersin.org/articles/10.3389/fnbot.2020.577804/full