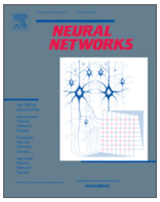
Neural Networks 131 (2020) 231–241

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Synthesis of recurrent neural dynamics for monotone inclusion with application to Bayesian inference[☆]

Peng Yi[a,b,*], ShiNung Ching[c]

[a] Department of Control Science and Engineering, Tongji University, Shanghai, China
[b] Shanghai Research Institute of Intelligent Science and Technology, Tongji University, Shanghai, China
[c] Department of Electrical and Systems Engineering, Washington University in St. Louis, USA

ARTICLE INFO

Article history:
Received 23 February 2020
Received in revised form 6 July 2020
Accepted 31 July 2020
Available online 12 August 2020

Keywords:
Recurrent neural networks
Poisson spiking neuron
Monotone inclusion
Normative approach
Network synthesis
Bayesian casual inference

ABSTRACT

We propose a top-down approach to construct recurrent neural circuit dynamics for the mathematical problem of monotone inclusion (MoI). MoI in a general optimization framework that encompasses a wide range of contemporary problems, including Bayesian inference and Markov decision making. We show that in a recurrent neural circuit/network with Poisson neurons, each neuron's firing curve can be understood as a proximal operator of a local objective function, while the overall circuit dynamics constitutes an operator-splitting system of ordinary differential equations whose equilibrium point corresponds to the solution of the MoI problem. Our analysis thus establishes that neural circuits are a substrate for solving a broad class of computational tasks. In this regard, we provide an explicit synthesis procedure for building neural circuits for specific MoI problems and demonstrate it for the specific case of Bayesian inference and sparse neural coding.

1. Introduction

Understanding the computational strategies embedded within neural circuits is a central question in neuroscience (Maass, 1997). The normative, or top-down approach to this problem presupposes that neural circuits act in a way that optimizes a certain objective function, e.g., minimization of free energy (Friston, 2010). One can synthesize a set of neural dynamics that enacts the optimization objective, then assess whether the synthesized dynamics are compatible with actual neural biophysics. This paper falls within this paradigm, with the goal of examining how a general-purpose mathematical objective – monotone inclusion, which underpins many ubiquitous functions – might be achieved through a generic set of biologically interpretable spiking dynamics. Deriving such an understanding would not only provide new hypotheses regarding neural function, but would also provide leverage on the problem of configuring neuromorphic computational architectures (Schuman et al., 2017).

Our work follows a wide thread of research that studies how different computational tasks might be completed by recurrent, spiking neural networks (SNNs). Recurrent neural dynamics has been investigated with the help of ODE methods, see Hu, Yi, and Zou (2020), Huang, Cao, Wen, and Yang (2016), Kudu (2018), Li, Liu, Li and Tisdell (2019) and Song, Fei, Cao, and Huang (2019) for the recent advances. In recent years, some scholars have dealt with recurrent neural dynamics with impulse inputs and time-delay. Some recent progress can be found in Chen, Zhang, Cao, and Huang (2020), Huang, Long, and Cao (2020), Huang, Zhang, Cao and Hu (2019), Huang, Zhang and Huang (2019), Li, Huang and Ji (2019), Zhang and Huang (2020) and Zhou, Wan, Huang, and Yang (2020). We focus specifically on the ubiquitous leaky integrate-and-fire (LIF) neural dynamics. Networks of LIF neurons have been configured to solve Bayesian causal inference (Moreno-Bote & Drugowitsch, 2015) by showing that the LIF dynamics can approximate a firing rate ODE for solving quadratic optimization. Similarly, Tang, Lin, and Davies (2017) show that a SNN with modified LIF spiking dynamics can approximate a locally competitive ordinary differential equation (ODE) to solve a LASSO (least absolute shrinkage and selection operator) problem, a schema that has been translated to LIF-based neuromorphic hardware. Other demonstrations of functions enacted via LIF-SNNs include L_1 minimization (Chou, Chung, & Lu, 2018), constrained quadratic optimization (Gangopadhyay & Chakrabartty, 2017), predictive

[☆] This work was partially supported by AFOSR 15RT0189 and NSF CMMI 1653589, from the US Air Force Office of Scientific Research, United States and the US National Science Foundation, respectively. This work was also sponsored by Shanghai Sailing Program, China (NO. 20YF1453000), the Fundamental Research Funds for the Central Universities, China (NO. 22120200048) and the Key research and development projects of the Ministry of Science and Technology of China (NO. 2018YFB1305304). The work of P. Yi was done when P. Yi was with Washington University in St. Louis.

[*] Correspondence to: NO. 17, Chuanhe Road 55, Pudong, Shanghai, China.
E-mail addresses: yipeng@tongji.edu.cn (P. Yi), shinung@wustl.edu (S. Ching).

https://doi.org/10.1016/j.neunet.2020.07.037
0893-6080/© 2020 Elsevier Ltd. All rights reserved.

coding and control (Huang & Ching, 2019; Pérez, Cabrera, Castillo, & Velasco, 2018) and sparse communication (Yoon, 2016; Zambrano, Nusselder, Steven Scholte, & Bohté, 2018). Another model class, SNNs with generalized linear response voltage dynamics and Poisson/Bernoulli spiking, has also been studied in the context of various computational tasks. For example, Friedrich and Lengyel (2016) configure an SNN with Poisson firing model to solve an infinite-horizon Markov decision problem, showing that the synthesized dynamics closely resemble experimental observations. Hao, Huang, Dong, and Xu (2020), Jang, Simeone, Gardner, and Grüning (2018), Rezende, Wierstra, and Gerstner (2011) and Taherkhani et al. (2020) investigate the learning of a probabilistic temporal sequence model with Poisson/Bernoulli SNN.

The case-by-case successes are suggestive of a general schema by which neural dynamics might be synthesized or configured. We posit the mathematical problem of monotone inclusion (MoI) as one such schema. MoI encompasses a range of important functions, such as quadratic optimization, Markov decision-making, non-cooperative games and as we will later demonstrate, Bayesian inference, Bauschke, Combettes, et al. (0000). The splitting algorithms developed for MoI are an efficient approach for mathematical programming (Beck & Teboulle, 2009; Combettes & Pesquet, 2011; Kafashan & Ching, 2017). Our premise is that neural circuits may embody a similar computational strategy for solving these decision making problems.

In particular, we study the widely adopted SNN model wherein neurons exhibit generalized linear response membrane voltage dynamics and Poisson firing rates. Using a mean-field approximation, we show that the differential–algebraic system of SNN equations has an equilibrium point that is a solution of an MoI. Hence, the SNN dynamics essentially enacts the well-known operator splitting algorithm in optimization and signal processing, Bauschke et al. (0000), Beck and Teboulle (2009) and Combettes and Pesquet (2011). This realization allows systematic top-down synthesis of SNNs to finding zeros of structured MoI problems.

The specific novel contributions of this paper are:

(1) Using the MoI/operator splitting formalism, we show that the nonlinear Poisson firing curves of individual neurons amount to proximal operators of different local regularizers. The observation that activation functions can enact a proximal operator has recently been made in the context of feedforward networks (Combettes & Pesquet, 2020), but our work does so in a recurrent setting with a well-defined and generic computational objective. Indeed, relating SNN dynamics to operator splitting ODEs has, to the best of our knowledge, not been previously discussed in the literature.

(2) We propose a systematic approach for configuring SNNs to complete specific MoI problems. We demonstrate this by configuring an SNN to solve linearly constrained quadratic optimization using a Lagrangian saddle point formulation. This is in contrast to existing work that does not include such constraints (Barrett, Denève, & Machens, 2013; Moreno-Bote & Drugowitsch, 2015; Tang et al., 2017).

(3) We demonstrate the proposed SNN methodology by numerical studies of a particularly relevant MoI problem: Bayesian causal inference. With an independent prior, the configured SNN dynamics resembles the well-studied locally competitive algorithm for sparse coding or LASSO in Rozell, Johnson, Baraniuk, and Olshausen (2008) and Tang (2016). However, by arriving at these dynamics from a top-down MoI synthesis we are able to ascribe functional significance to individual neuronal dynamics, e.g., relating Poisson firing curves directly to parameters of the Bayesian prior. In addition, we consider a causal inference with a structural prior by configuring SNN to solve linearly constrained quadratic programming.

The remainder of the paper is organized as follows. Section 2 uses a reverse-engineering analysis to show that a recurrent SNN can be treated as an ODE for solving MoI. Section 3 gives a top-down synthesis to hard-wire a recurrent SNN for computational tasks. Section 4 illustrates the proposed methodology with Bayesian causal inference and numerical simulations. Conclusions are provided in Section 5.

2. Top-down analysis of the recurrent SNN dynamics

Our first result involves reverse engineering a typical recurrent SNN as an operator-splitting-based ODE system that solves MoI. Relevant mathematical notation and background regarding MoI are found in Appendix.

2.1. Recurrent SNN dynamics

Membrane voltage dynamics: Following Gerstner, Kistler, Naud, and Paninski (2014), Jolivet, Rauch, Lüscher, and Gerstner (2006) and Pillow, Paninski, Uzzell, Simoncelli, and Chichilnisky (2005), we consider a recurrent spiking neural network as a group of interconnected neurons $\mathcal{N} = \{1, \ldots, N\}$, each of which has dynamics:

$$\tau_m \frac{du_i}{dt} = -u_i(t) + \sum_{j \in \mathcal{N}} w_{ij} \int_0^\infty X_j(t-\tau)\epsilon(\tau)d\tau - \eta_i X_i(t) + I_i^{ext} \quad (1)$$

Here, u_i is the membrane voltage of neuron i, and τ_m is the membrane time constant. w_{ij} is the synaptic weight from neuron j to neuron i (self-connections are allowed). $X_i(t) = \sum_s \delta(t - t_i^s)$ is the spiking train of neuron i represented as a sum of Dirac δ-functions. And $\epsilon(\tau)$ is the postsynaptic impulsive response kernel with the form of $\epsilon(\tau) = \epsilon_0 \exp(-\frac{\tau}{\tau_s})$ for $\tau \geq 0$. τ_s is the synaptic time constant and $\epsilon_0 = \tau_s^{-1}$ ensures normalization to $\int \epsilon(t)dt = 1$. The after-hyperpolarization is modeled as an instantaneous current pulse with a negative sign and magnitude $\eta_i \geq 0$. I_i^{ext} is the external input current. The physical units of the variables and constants are omitted since we consider only the theoretical properties of the dynamics.

Poisson firing curve: The spiking train of neuron i can be modeled as an inhomogeneous Poisson point process with the instantaneous firing rate $\lambda_i(t)$ as a function of membrane voltage. In other words, the number of spikes in a fixed short interval Δt follows $\mathbb{P}\{\sum_{t' \in [t, t+\Delta t]} \{\delta(t' - t_i^s) \neq 0\} = n\} = \frac{[\lambda_i(t)\Delta t]^n}{n!} e^{-\lambda_i(t)\Delta t}$. The firing curve for neuron i is a nonlinear mapping from its membrane voltage to the instantaneous firing rate: $\lambda_i(u) : u(t) \mapsto \lambda_i(t)$. A typical firing curve that has been observed in biological neurons is the rectified linear with a silent zone(SReLU):

$$\lambda_i(t) = k_i[u_i(t) - \theta_i]_+ \quad (2)$$

where $k_i \geq 0$ is the amplification factor and $\theta_i \geq 0$ is the voltage threshold. $[x]_+ = x$ if $x \geq 0$, and $[x]_+ = 0$, otherwise. A figure of (2) is shown in Fig. 1.

2.2. Mean-field approximation and equilibrium point analysis

An analytical challenge associated with the above formulation comes from the non-continuity of the membrane dynamics and stochasticity of the spikes. However, since the membrane time-scale is much slower than the synapse response time-scale, i.e., $\tau_m \gg \tau_s$, and the neurons are interacting in a dense population, the network membrane voltage dynamic can be well approximated with a smooth ordinary differential equation (ODE) from a mean-field analysis (see, e.g., Friedrich & Lengyel, 2016).

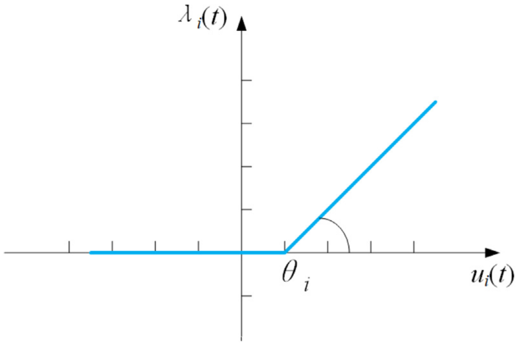


Fig. 1. The firing curve of the rectified linear with a silent zone.

That is the overall SNN dynamics can be approximated as follows:

$$\begin{aligned} &\tau_m \dot{u} = -u + W\lambda - diag\{\eta\}\lambda + I^{ext}, \\ &\lambda_i(t) = k_i[u_i(t) - \theta_i]_+, i = 1, \dots, N \end{aligned} \tag{3}$$

where $u = col(u_1, \dots, u_N) \in \mathbb{R}^N$, $W = [w_{ij}] \in \mathbb{R}^{N\times N}$, $\lambda = col(\lambda_1, \dots, \lambda_N)$, $\eta = col(\eta_1, \dots, \eta_N)$ and $I^{ext} = col(I_1^{ext}, \dots, I_N^{ext})$.

The equilibrium point/steady state (u^*, λ^*) of (3) satisfies the following equation:

$$\begin{aligned} &u^* = W\lambda^* - diag\{\eta\}\lambda^* + I^{ext}, \\ &\lambda_i^* = k_i[\textstyle\sum_{j\in\mathcal{N}} w_{ij}\lambda_j^* - \eta_i\lambda_i^* + I_i^{ext} - \theta_i]_+ \end{aligned} \tag{4}$$

In other words, when the recurrent SNN converges to its equilibrium point, the steady firing rate $\lambda_i^*, i \in \mathcal{N}$ is a fixed point solution of a group of nonlinear equations in (4).

2.3. Equilibrium analysis from MoI perspective

We aim to understand what type of computational task can be fulfilled by the fixed point equation (4). We show that (4) is closely related to the operator splitting methods in optimization and signal processing (Bauschke et al., 0000; Combettes & Pesquet, 2011).

Definition 2.1 (*Proximal Operator*). The proximal operator of a function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$prox_f(x) : x \in \mathbb{R}^n \mapsto \arg\min_{y\in\mathbb{R}^n} f(y) + \frac{1}{2}\|y - x\|_2^2. \tag{5}$$

Note, especially, that the SReLU curve in (2) is a proximal operator:

Lemma 2.2. *The SReLU operator $x \mapsto k[x - \theta]_+$ for $x \in \mathbb{R}$ with $k \in \mathbb{R}_+, \theta \in \mathbb{R}_+$ can be regarded as the proximal operator of*

$$r(x) = \frac{1}{2}\rho x^2 + \varrho x + \iota_{\mathbb{R}_+}(x). \tag{6}$$

with $\rho \geq 0, \varrho \geq 0$ that $k = \frac{1}{1+\rho}$ and $\theta = \varrho$. Here, $\iota_{\mathbb{R}_+}(x)$ is the indicator function that takes zeros when $x \geq 0$ and takes ∞ when $x < 0$.

Proof. $r(x)$ in (6) is a lower semi-continuous convex function by definition. From the definition of proximal operator, $prox_r(x) = \arg\min_{y\in\mathbb{R}} \frac{1}{2}\rho y^2 + \varrho y + \iota_{\mathbb{R}_+}(y) + \frac{1}{2}(y - x)^2$. Suppose $y^* = prox_r(x)$, and then at y^* we have

$$0 \in \rho y^* + \varrho + N_{\mathbb{R}_+}(y^*) + (y^* - x). \tag{7}$$

with $N_{\mathbb{R}_+}$ as the normal cone of $\mathbb{R}_+$. Therefore, we can rewrite the above equation as $0 \in y^* + N_{\mathbb{R}_+}(y^*) + \frac{\varrho}{\rho+1} - \frac{1}{\rho+1}x$ or

$$\frac{1}{1+\rho}x - \frac{\varrho}{\rho+1} \in (\mathrm{Id} + N_{\mathbb{R}_+})(y^*) \tag{8}$$

Therefore, $y^* = (\mathrm{Id} + N_{\mathbb{R}_+})^{-1}(\frac{1}{1+\rho}x - \frac{\varrho}{1+\rho}) = [\frac{1}{1+\rho}x - \frac{\varrho}{1+\rho}]_+$. Since $1 + \rho \geq 0$, we have $y^* = \frac{1}{1+\rho}[x - \varrho]_+$. Therefore, taking $k = \frac{1}{1+\rho}$ and $\theta = \varrho$, we have $k[x - \theta]_+ = Prox_r(x)$. □

Hence, the firing curve of neuron i, $\lambda_i = k_i[u_i - \theta_i]_+$ in (2) can be treated as a proximal operator $\lambda_i = Prox_{R_i}(u_i)$ with

$$R_i : u_i \mapsto \frac{1}{2}(1/k_i - 1)u_i^2 + \theta_i u_i + \iota_{\mathbb{R}_+}(u_i). \tag{9}$$

Remark 2.3. $\rho = (1/k_i - 1) \geq 0$ or equivalently $0 < k_i \leq 1$ is needed to ensure that $R_i(u_i)$ in (9) is a convex function. This always can be achieved by scaling the units of all physical variables. Therefore, we will assume $0 < k_i \leq 1, i \in \mathcal{N}$ for the firing curve in (2).

Given the membrane dynamics (1) and firing curve (2) with fixed parameters, we denote

$$R_i : \lambda_i \mapsto \frac{1}{2}\rho_i\lambda_i^2 + \varrho_i\lambda_i + \iota_{\mathbb{R}_+}(\lambda_i), R(\lambda) = \sum_{i=1}^{N} R_i(\lambda_i) \tag{10}$$

with $\rho_i = 1/k_i - 1$ and $\varrho_i = \theta_i$. Then we define two operators $\mathfrak{A}$ and $\mathfrak{B}$ as following.

$$\mathfrak{A} : \lambda \mapsto (I + diag\{\eta\})\lambda - W\lambda - I^{ext}, \mathfrak{B} : \lambda \mapsto \partial R(\lambda). \tag{11}$$

With assuming $0 < k_i \leq 1, i \in \mathcal{N}$, $R(\lambda)$ in (10) is a convex function, and $\mathfrak{B}$ is (strongly) monotone. Since $R(\lambda)$ is a sum of decomposable functions, we know $\partial R(\lambda) = \prod_{i\in N} \partial R_i(\lambda_i)$. Therefore, $Prox_R(z) = (\mathrm{Id}+\partial R)^{-1}(z) = \prod_{i\in N}(\mathrm{Id}+\partial R_i)^{-1}(z_i) = \prod_{i\in N} Prox_{R_i}(z_i)$. With (4), at the equilibrium point λ^*

$$\lambda_i^* = k_i[\lambda_i^* - ((1 + \eta_i)\lambda_i^* - \sum_{j\in\mathcal{N}} w_{ij}\lambda_j^* - I_i^{ext}) - \theta_i]_+, \forall i \in \mathcal{N}.$$

We put $z_i = \lambda_i^* - ((1 + \eta_i)\lambda_i^* - \sum_{j\in\mathcal{N}} w_{ij}\lambda_j^* - I_i^{ext})$, then at the equilibrium point $\lambda_i^* = Prox_{R_i}(z_i)$. Therefore, the fixed point equation (4) can be reformulated as

$$\begin{aligned} \lambda^* &= Prox_R\Big[\lambda^* - \Big((I + diag\{\eta\})\lambda^* - W\lambda^* - I^{ext}\Big)\Big] \\ &= (\mathrm{Id} + \mathfrak{B})^{-1}(\mathrm{Id} - \mathfrak{A})\lambda^* \end{aligned}$$

In other words, $(\mathrm{Id} - \mathfrak{A})\lambda^* \in (\mathrm{Id} + \mathfrak{B})\lambda^*$ since $\mathfrak{B}$ is a maximally monotone operator as a sub-differential of a convex function. Therefore, the equilibrium firing rate λ^* is a solution to the following generalized equation

$$\mathbf{0} \in (\mathfrak{A} + \mathfrak{B})(\lambda). \tag{12}$$

Thus, the nonlinear fixed point equation (4) is essentially a generalized inclusion problem.

The operator $\mathfrak{B}$ is decomposable and can be interpreted as a combination of each neuron's local objective. Then, the nonlinear firing curve can treated as the proximal operator of each local regularizer, i.e., $(\mathrm{Id} + \mathfrak{B}_i)^{-1}$. The operator $\mathfrak{A}$ is a linear operator specified by the interaction among neurons. To understand the computational versatility of the formulation (12), we will impose a further structure on the operators.

Definition 2.4 (*Monotone Operator and Monotone Inclusion*). An operator $M : x \mapsto Mx$ is monotone if $\langle x-y, Mx-My\rangle \geq 0, \forall x, y \in dom(M)$. Finding a point x that $\mathbf{0} \in Mx$ is called a monotone inclusion(MoI).

The operator $\mathfrak{B}$ is monotone since it is the subgradient of a convex function. When $\mathfrak{A}$ is also monotone, the equilibrium point λ^* is a solution to a MoI (12) with the operator being a sum of two monotone operators. The monotonicity of $\mathfrak{A}$ is dependent on W, the synaptic weights of the network. Thus, we have the following:

Proposition 1. *Assume a recurrent SNN with membrane dynamics (1) and firing curve (2) with fixed parameters. Suppose $0 < k_i \leq 1, i \in \mathcal{N}$ and the synaptic weights W ensures that $\mathfrak{A}$ in (11) is monotone. Then, at the equilibrium point of (1)–(2), the steady firing rate λ^* is a solution to MoI (12) as the sum of operator $\mathfrak{A}$ and $\mathfrak{B}$ in (11).*

Remark 2.5. The above analysis assumes a fixed SReLU firing curve (2). However, we can also reverse engineer other firing curves as the proximal operator of different local objectives (Combettes & Pesquet, 2011 give a table of common functions and their proximal operators, and the composition rules of proximal operator, like translation, scaling and conjugation). Moreover, Proposition 2.4 in Combettes and Pesquet (2007) shows that any function ϕ defined from $\mathbb{R}$ to $\mathbb{R}$ that is non-expansive (1-Lipschitz) and increasing is a proximal operator of a function on $\mathbb{R}$. The recent works (Combettes & Pesquet, 2019, 2020) also give more examples of relating various firing curves (also known as activation functions) to proximal operators of different regularizers, for the purposes of deep learning. However, there are still substantial difference with (Combettes & Pesquet, 2020), since Combettes and Pesquet (2020) only consider the static property of activation functions and treats the cascading of layered **feed-forward neurons** as an iteration process for solving variational inequalities. This work relates the dynamics of a **recurrent neural network** as an ODE solver for monotone inclusion.

2.4. Neural dynamics as an ODE that solves MoI

While we have shown that the fixed point of the SNN dynamics is the solution of an MoI, we have not yet concluded that dynamics converge to this equilibrium. To establish convergence, an additional property is required as follows:

Definition 2.6 (*Strongly Monotonicity*)**.** An operator $M : x \mapsto Mx$ is $\upsilon-$ strongly monotone if $\langle x - y, Mx - My\rangle \geq \upsilon\|x - y\|, \forall x, y \in dom(M)$. M is $\alpha-$inverse strongly monotone if $\langle x-y, Mx-My\rangle \geq \alpha\|Mx - My\|, \forall x, y \in dom(M)$.

Remark 2.7. Intuitively speaking, monotonicity can be treated as a type of rotational invariance of angles, while (inverse-)strongly monotonicity is a requirement on the scaling property of the operator, i.e., the operator will not shrink the distance of two vectors too much.

For a MoI, $\mathbf{0} \in M(x) + \partial\Phi(x)$ with operator M being α-inverse strongly monotone and Φ being a convex function, the following Newton-type differential–algebraic dynamics is proposed in (Abbas & Attouch, 2015),

$$\dot{u} = -u + \lambda - \mu M\lambda \tag{13}$$

$$\lambda = (\mathrm{Id} + \mu\partial\Phi)^{-1}(u) = Prox_{\mu\Phi}(u) \tag{14}$$

Note that (13)–(14) are differential–algebraic dynamics, and the existence of solution in the sense of ODE trajectory (the Cauchy problem) has been discussed in Abbas and Attouch (2015) and Attouch and Svaiter (2011), where the sufficient condition on the global existence and uniqueness of solution/trajectory is given in both Section 1 of Abbas and Attouch (2015) and Section 2 of Attouch and Svaiter (2011). It is shown that λ will converge to a zero of $M + \partial\Phi$ with (13)–(14) if μ is $0 \leq \mu \leq \frac{1}{2\alpha}$ (see Theorem 1.8 of Abbas & Attouch, 2015). Since (13) and (14) use different strategies to handle operators, (13)–(14) is called an operator-splitting ODE for solving MoI.

Thus, the mean-field approximated neural dynamics in (3) amounts to the operator-splitting ODE (13)–(14), modulo a proper parameter configuration.

Theorem 2.8. *Assume a recurrent SNN with mean-field dynamics (3) and fixed parameters, $W, I^{ext}, k, \theta, \eta$. Suppose $0 < k_i \leq 1, i \in \mathcal{N}$ and suppose the synaptic weight W ensures that $\mathfrak{A}$ in (11) is monotone. If there exists μ and α such that $0 \leq \mu \leq \frac{1}{2\alpha}$ and*

$$M : \lambda \mapsto \frac{1}{\mu}((I + diag\{\eta\} - W)\lambda - I_{ext}) \tag{15}$$

is α-inverse strongly monotone, then the recurrent neural dynamics (3) converges to its equilibrium point, and the steady firing rate λ^* is the zero of $\mathfrak{A} + \mathfrak{B}$ in (11).

Proof. With Proposition 1, the equilibrium point of the dynamics (3) is a zero of $\mathfrak{A} + \mathfrak{B}$ in (11). To show the convergence, we will rewrite the neural dynamics (3) as the (13)–(14) by properly setting M and Φ.

Since we assume $\tau_m \gg \tau_s$ in the mean-field dynamics (3), τ_m can be properly scaled to be 1. Therefore, with M in (15) and given any fixed $\mu \geq 0$ the right hand of (3) can be written as

$$-u + W\lambda - diag\{\eta\}\lambda + I^{ext} = -u + \lambda - \mu M\lambda \tag{16}$$

Denote $\Phi(\lambda) = \frac{1}{\mu}R(\lambda)$ with $R(\lambda)$ in (10). We already know that $\partial\Phi(\lambda) = \frac{1}{\mu}\partial R(\lambda)$. Therefore,

$$(\mathrm{Id} + \mu\partial\Phi)^{-1} = (\mathrm{Id} + \mu\frac{1}{\mu}\partial R)^{-1} = (\mathrm{Id} + \mathfrak{B})^{-1},$$

with $\mathfrak{B}$ in (11). We have already shown in Proposition 1 that $\lambda_i(t) = k_i[u_i(t) - \theta_i]_+, i = 1, \ldots, N$ is the same as $\lambda = (\mathrm{Id} + \mathfrak{B})^{-1}(u)$. Therefore, the neural dynamics (3) can be written as (13)–(14) with M in (15) and $\Phi = \frac{1}{\mu}R(\lambda)$.

When there exist μ and α such that $0 \leq \mu \leq \frac{1}{2\alpha}$ and M in (15) is α-inverse strongly monotone, the neural dynamics (3) converges to a zero of $M + \partial\Phi$ by Theorem 1.8 of Abbas and Attouch (2015). Since $M = \frac{1}{\mu}\mathfrak{A}$ and $\partial\Phi = \frac{1}{\mu}\mathfrak{B}$, $M + \partial\Phi$ has the same zeros as $\mathfrak{A} + \mathfrak{B}$. Note that $\mathfrak{B}$ is strongly monotone as the sub-differential of a strongly convex function $R(\lambda)$ in (10), therefore, the zero of $\mathfrak{A}+\mathfrak{B}$ is unique. Hence, the neural dynamics (3) converges to its equilibrium as the zero of $\mathfrak{A} + \mathfrak{B}$ in (11). □

Remark 2.9. It may be apparent that the MoI: $\mathbf{0} \in (\mathfrak{A}+\mathfrak{B})(x)$ can be solved by alternative, gradient-based dynamics. For example, the differential inclusion $\dot{x} \in \mathfrak{A}x+\mathfrak{B}x$ converges to $zer(\mathfrak{A}+\mathfrak{B})$ with a strongly monotonicity assumption on $\mathfrak{A}$ and $\mathfrak{B}$ (see Attouch, Cabot, & Czarnecki, 2018). The proximal gradient dynamics $\dot{x} = (\mathrm{Id}+\eta\mathfrak{B})^{-1}(\mathrm{Id}-\eta\mathfrak{A})x-x$ is also shown to converge to $zer(\mathfrak{A}+\mathfrak{B})$ with a monotone $\mathfrak{B}$ and an $\alpha-$inverse strongly monotone $\mathfrak{A}$ and proper step-size η (see Boţ & Csetnek, 2018). However, the differential–algebraic system of (1)–(2) cannot be mapped to the above two dynamics.

It is important to note that Theorem 2.8 relies on a sufficient condition to guarantee the convergence (strong monotonicity). Such an assumption is derived purely from a mathematical viewpoint, while whether the biological synaptic weights could ensure such an assumption still needs experimental validation. However, as we will later see in our simulations, this assumption could be relaxed in specific settings.

3. Top-down synthesis of spiking networks

Equipped with the above analysis, we are now able to synthesize SNNs for a variety of decision-making problems. We proceed to demonstrate this for the case of linearly constrained quadratic programming (QP).

3.1. Synthesis of SNNs for MoI

Consider a computational problem (e.g., Markov decision making) that can be solved by means of MoI:

$$\mathbf{0} \in (\mathfrak{A} + \mathfrak{B})\lambda. \tag{17}$$

Here, the decision variable is $\lambda \in \mathbb{R}^N$, and the operator $\mathfrak{A}$ is assumed to be monotone and linear $\mathfrak{A} : \lambda \in \mathbb{R}^N \rightarrow Q\lambda + c$ with $Q \in \mathbb{R}^{N\times N}$ and $c \in \mathbb{R}^N$. The operator $\mathfrak{B}$ has a Cartesian product form as $\mathfrak{B} : \lambda \mapsto \prod_{i=1}^N \mathfrak{B}_i(\lambda_i)$. We assume each $\mathfrak{B}_i(\lambda_i)$ is monotone and has a closed form resolvent $R_{\mathfrak{B}_i} = (\mathrm{Id} + \mathfrak{B}_i)^{-1}$.

Clearly, by Theorem 2.8, we can configure a SNN with its mean-field dynamics to solve the MoI in (17). In particular:

Corollary 3.1. *Suppose the operator $\mathfrak{A}$ in (17) is $\alpha-$inverse strongly monotone. Then, we can set a parameter ϵ that $0 \leq \epsilon \leq \frac{1}{2\alpha}$, and configure a SNN with its mean-field dynamics as follows. The membrane voltage dynamics (1) has the parameters:*

$$W - diag\{\eta\} = I - \epsilon * Q, I^{ext} = -\epsilon c, \tag{18}$$

with $\tau_m \gg \tau_s$ to ensure the validity of the mean-field approximation (3). The firing curve of each neuron i is set as

$$\lambda_i(t) = (\mathrm{Id} + \epsilon\mathfrak{B}_i)^{-1}(u_i(t)). \tag{19}$$

Then, the firing rate $\lambda_i(t), i \in \mathcal{N}$ of the configured SNN converges to a zero of MoI in (17).

Again, it is important to note that the condition on strong monotonicity is sufficient but not necessary. We will see this come into play in the subsequent section.

3.2. Synthesis of SNN for constrained quadratic optimization

Quadratic optimization (or, programming (QP)) is important for a variety of decision making problems (Nocedal & Wright, 2006), and can be used as a basic building block for solving general optimization problems, Boggs and Tolle (1995) and Gill, Murray, and Saunders (2005).

We can configure an SNN to solve a QP as follows:

$$\min_{x\in\mathbb{R}^N_+} \quad \frac{1}{2}x^T Ex + q^T x \quad s.t.\ Ax \leq b \tag{20}$$

with $E \in \mathbb{R}^{N\times N}$ being positive definite, $A \in \mathbb{R}^{m\times N}$, $q \in \mathbb{R}^m$ and $m < N$. We consider its Lagrangian function with a multiplier $\mu \in \mathbb{R}^m_+$

$$L(x, \mu) = \frac{1}{2}x^T Ex + q^T x + \mu^T(Ax - b) + \iota_{\mathbb{R}^N_+}(x) + \iota_{\mathbb{R}^m_+}(\mu). \tag{21}$$

According to the Lagrangian duality, the optimal solution with the optimal multiplier (x^*, μ^*) is a saddle point of the Lagrangian function $L(x, \mu)$ and should satisfy the following KKT condition

$$\begin{aligned} \mathbf{0} &\in Ex + q + A^T\mu + \partial\iota_{\mathbb{R}^N_+}(x) \\ \mathbf{0} &\in -Ax + b + \partial\iota_{\mathbb{R}^m_+}(\mu) \end{aligned}$$

By putting the decision variable x and the multiplier μ together as $\lambda = col(x, \mu)$, then the above KKT condition can be written as an MoI of the form (17):

$$\mathfrak{A} : \lambda \mapsto \begin{pmatrix} E & A^T \\ -A & \mathbf{0} \end{pmatrix} \lambda + \begin{pmatrix} q \\ b \end{pmatrix}, \mathfrak{B} : \lambda \mapsto \partial\iota_{\mathbb{R}^{N+m}_+}(\lambda) \tag{22}$$

The operator $\mathfrak{A}$ in (22) cannot be directly shown to be inverse strongly monotone. However, notice that the MoI in (22) shares a similar structure to (17). Thus, we can still configure an SNN with (18)–(19) by choosing a small step-size ϵ. Specifically, letting the firing rate of neurons be x and μ and with $\epsilon \ll 1$ and a fixed $\eta = col(\eta_x, \eta_\mu)$, we can set the parameters W, I^{ext} as

$$\begin{aligned} W &= \begin{pmatrix} I + diag\{\eta_x\} - \epsilon * E & -\epsilon * A^T \\ \epsilon * A & I + diag\{\eta_\mu\} \end{pmatrix}, \\ I^{ext} &= \begin{pmatrix} -\epsilon * q \\ -\epsilon * b \end{pmatrix} \end{aligned} \tag{23}$$

Moreover, we obtain the firing curve of each neuron as $\lambda_j(t) = [u_j]_+$. While our analytical guarantees of convergence no longer apply (due to aforementioned lack of strong monotonicity), we nonetheless find empirically that the configured SNN has its firing rate converge to optimal primal–dual solution of the QP (20). We will demonstrate this in the simulation Section 4.

4. Neural Bayesian causal inference

In this section, we illustrate the proposed synthesis methodology for the problem of maximum *a posteriori* (MAP) Bayesian causal inference, a fundamental task that could be a building block for many cognitive functions, such as object recognition or odors detection when the number of potential sensory stimuli is enormous (Knill & Richards, 1996; Tenenbaum, Griffiths, & Kemp, 2006). We can configure the synaptic interaction strengths of a recurrent neural network to perform Bayesian inference without noisy samplings. Our approach is consistent with a recent suggestion in Dold et al. (2019) that spiking neural networks may have no need for noise to perform sampling-based Bayesian inference. The experimental results of pilot examples show how the theoretical results could explain the functionality of neural networks.

The goal of MAP inference is to infer a latent cause of a given observation. Specifically, we assume a probabilistic generative model of the latent cause $\lambda \in \mathbb{R}^N$ and the observation $y \in \mathbb{R}^m$ as $p(y, \lambda) = p(y|\lambda)p(\lambda)$. Once $\mathbf{y}$ is observed, the MAP problem becomes:

$$\max_{\lambda\in\mathbb{R}^N} \log p(\lambda|\mathbf{y}) \propto \log p(\mathbf{y}|\lambda) + \log p(\lambda). \tag{24}$$

We consider a linear-Gaussian likelihood: $y = Q\lambda + \epsilon$. Here $Q = [q_1, \ldots, q_N] \in R^{m\times N}$ is assumed to be known or has been learned, while each $q_i \in \mathbb{R}^m$ is called a feature vector. Each component of ϵ is an i.i.d. Gaussian noise with zero mean and unit variance. Hence, in (24) we have $\log p(\mathbf{y}|\lambda) \propto -\frac{1}{2}(\mathbf{y} - Q\lambda)^T(\mathbf{y} - Q\lambda)$. It remains to specify the prior $p(\lambda)$, for which we consider two examples, below.

4.1. Independent sparsity prior

First, we consider the following heavy tailed prior that promotes sparsity:

$$p(\lambda) \propto \prod_{i=1}^N \exp(-\varrho_i|\lambda_i| - \frac{\rho_i}{2}\lambda_i^2)H(\lambda_i). \tag{25}$$

Here $H(x)$ is the Heaviside function with $H(x) = 1$ if $x \geq 0$ and $H(x) = 0$ otherwise, which imposes that each coordinate of the latent cause is nonnegative. The parameters $\varrho_i > 0$ and $\rho_i > 0$ weight Laplacian and Gaussian contributions to the prior and serve to regularize the sparsity and strength of the ith cause, respectively.

Then, combined with the likelihood, the causal inference problem is

$$\min_{\lambda\in\mathbb{R}^N}, 1/2(\mathbf{y} - Q\lambda)^T(\mathbf{y} - Q\lambda) + \sum_{i=1}^N R_i(\lambda_i) \tag{26}$$

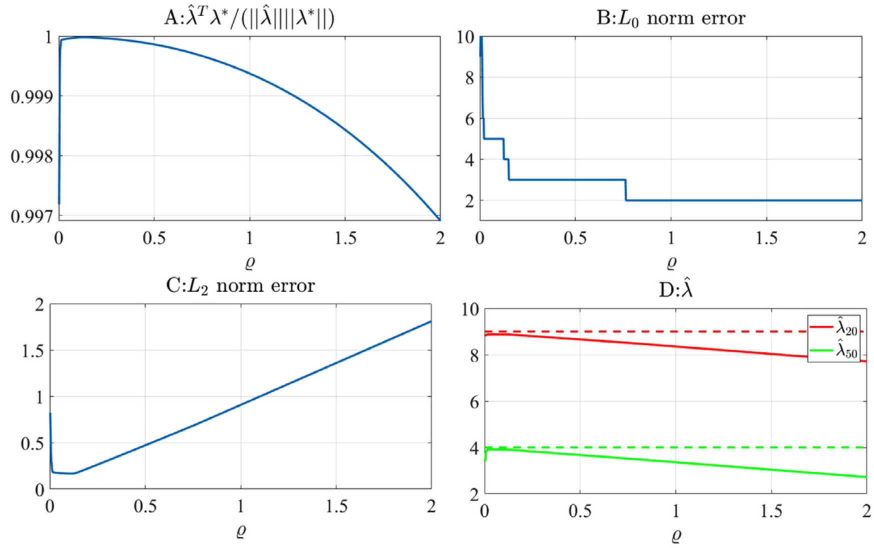


Fig. 2. The tuning of the inferred latent cause $\hat{\lambda}$ with sparsity prior parameter ϱ. A: The tuning of the angular error between λ^* and $\hat{\lambda}$, $\hat{\lambda}^T\lambda^*/(\|\hat{\lambda}\|\|\lambda^*\|)$. B: The tuning of L_0 norm error, $\|\lambda^* - \hat{\lambda}\|_0$. C: The tuning of L_2 norm error, $\|\lambda^* - \hat{\lambda}\|_2$. D: The tuning of the stationary firing rate of $\hat{\lambda}_{20}$ and $\hat{\lambda}_{50}$.

with $R_i(\lambda_i)$ defined in (10). Note that for consistency we turn the maximization in (24) to minimization. The optimization problem (26) and its variants have been widely studied in neural coding (where it is often referred to as sparse coding Balavoine, Romberg, & Rozell, 2012; Kafashan & Ching, 2017; Olshausen & Field, 1997; Rozell et al., 2008; Tang, 2016; Tang et al., 2017), signal processing and image science (e.g., Beck & Teboulle, 2009).

Solving (26) is equivalent to finding the solution to the MoI, $\mathbf{0} \in (\mathfrak{A} + \mathfrak{B})\lambda$ with

$$\mathfrak{A} : \lambda \mapsto Q^TQ\lambda - Q^T\mathbf{y};\ \mathfrak{B} : \lambda \mapsto \prod_{i=1}^{N} \partial R_i(\lambda_i).$$

Therefore, we can configure a recurrent SNN with Corollary 3.1 to solve this problem. We set $\tau_m = 1$, $\eta = \mathbf{1}$ and $\epsilon = 1$, then $W = 2I - Q^TQ$ and $I^{ext} = Q^T\mathbf{y}$. Then, with (3) and Lemma 2.2, the mean-field dynamics of the configured network is

$$\dot{u} = -u + \lambda - Q^TQ\lambda + Q^T\mathbf{y} \tag{27}$$

$$\lambda_i(t) = \frac{1}{1+\rho_i}[u_i(t) - \varrho_i]_+, i = 1, \ldots, N \tag{28}$$

Thus, starting with MoI as a general top-down objective, we arrive at a recurrent network with properties that are interpretable from a neural circuit perspective. For example, the parameter ϱ_i appears in (28) as an activation threshold (which promotes sparsity). Indeed, the overall derived dynamics resembles the locally competitive algorithm (LCA) studied in the context of sparse coding (e.g., Balavoine et al., 2012; Rozell et al., 2008). The LCA is built on physiological and functional intuition and then analyzed for convergence using a post-hoc Lyapunov approach. Here, we have shown that the sparse coding is a specific case of MoI and as such LCA-like dynamics can be synthesized directly with implicit convergence guarantees.

Example 4.1. Consider $\lambda \in \mathbb{R}^{100}$ and $\mathbf{y} \in \mathbb{R}^{10}$. The feature vectors $Q = [q_1, \ldots, q_{100}]$ and an observation $\mathbf{y}$ are randomly generated as follows. Each element of Q is first independently and uniformly drawn from $[0, 1]$, and then each $q_i \in \mathbb{R}^{10}$ is normalized to 1, $\|q_i\|_2 = 1$. The feature vectors are over-complete and linearly dependent. $\mathbf{y}$ is generated as $\mathbf{y} = 9 * q_{20} + 4 * q_{50} + \varepsilon$. Here each element of ε is an i.i.d. Gaussian noise with zero mean and variance of 0.001. The true latent cause $\lambda^* \in \mathbb{R}^{100}$ is a sparse high dimensional vector with $\lambda^*_{20} = 9$, $\lambda^*_{50} = 4$ and all other coordinates are zero. Then we configure the SNN to infer the latent cause with (27)–(28).

Firstly, we study how the inferred latent cause $\hat{\lambda}$ varies with different sparsity prior ϱ_i or the threshold in (28). We take a fixed ρ_i and set $k_i = \frac{1}{1+\rho_i} = \frac{1}{1+0.001}$ for all neurons. We let $\varrho_i = \varrho$ across all neurons, and tune ϱ to change from 0.002 to 2 in steps of 0.002. We run the configured SNN and take the stationary firing rate as the inferred $\hat{\lambda}$. Figs. 2 and 3 show how the hand-wired SNN performs with ϱ. Fig. 2A shows how the angular error between λ^* and $\hat{\lambda}$, i.e., $\frac{\hat{\lambda}^T\lambda^*}{\|\hat{\lambda}\|\|\lambda^*\|}$ varies with ϱ. Fig. 2B shows how the number of nonzero elements of $\lambda^* - \hat{\lambda}$, i.e., $\|\lambda^* - \hat{\lambda}\|_0$ varies with ϱ. It indicates that $\|\lambda^* - \hat{\lambda}\|_0$ decreases as we increase sparsity prior ϱ or the threshold. After a certain threshold ($\varrho = 0.2$ in Fig. 2B), the inferred latent cause $\hat{\lambda}$ has the same nonzero coordinates as the true latent cause λ^*, implying that the configured network discriminates the identity of the true latent cause, validating the effectiveness of the proposed methods in Bayesian inference. Fig. 2C shows how L_2 norm error, $\|\lambda^* - \hat{\lambda}\|_2$ tunes with ϱ. Fig. 2D shows how the firing rate of $\hat{\lambda}_{20}$ and $\hat{\lambda}_{50}$ tune with ϱ. Therefore, Fig. 2 shows that there is a performance tradeoff with respect to ϱ, involving the identity and strength of the inferred latent cause. Fig. 3 gives the firing rate trajectories when ϱ is set to $0.002, 0.02, 1, 2$, respectively, to evaluate the SNN transient response (from zero initial condition) with respect to ϱ. It shows that a larger ϱ (a stronger thresholding nonlinearity) corresponds to faster convergence. The studies examine how the algorithm parameter influences the inference results.

4.2. Tolerance to noise

Next, we study the robustness of the configured dynamics to noise. Since (27)–(28) is a mean-field approximation of a Poisson spiking dynamics, there exist approximation errors in λ on the right of (27). Further, due to the Poisson spiking randomness, the threshold parameter ϱ might also be corrupted. The synaptic weights might drift due to plasticity or stochasticity in molecular dynamics of neurotransmitters. Hence, we simulate the following SNN dynamics:

$$\dot{u} = -u + \lambda - (Q^TQ + \varepsilon_1(t))(\lambda + \varepsilon_2(t)) + Q^T\mathbf{y} \tag{29}$$

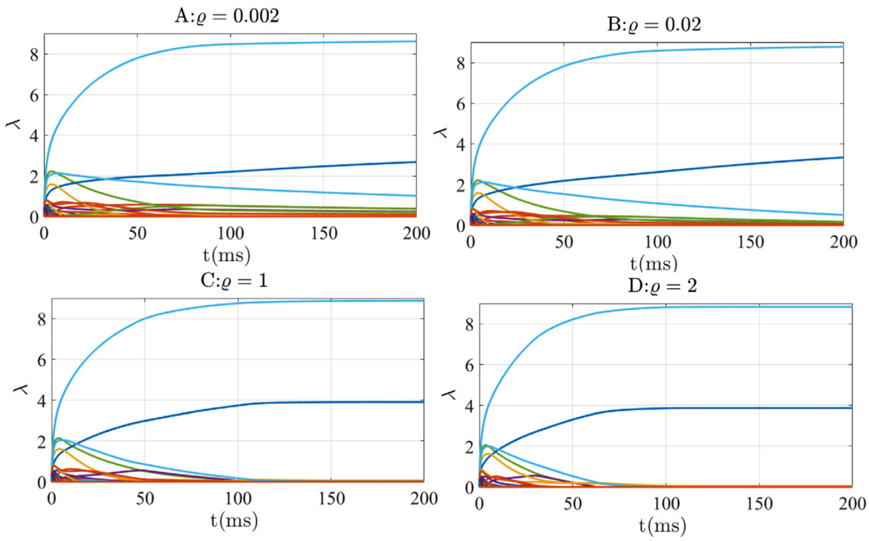


Fig. 3. The firing rate transition trajectories vary with four different sparsity prior parameter ϱ.

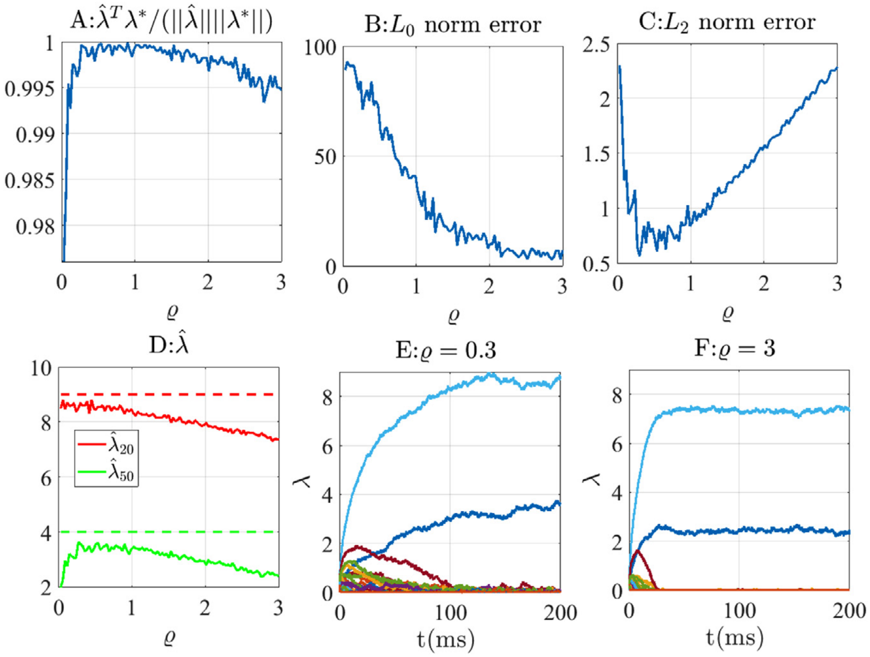


Fig. 4. The tuning of neural dynamics with sparsity prior parameter ϱ subject to random noises. A: The tuning of the angular error between λ^* and $\hat{\lambda}$, $\hat{\lambda}^T\lambda^*/(\|\hat{\lambda}\|\|\lambda^*\|)$. B: The tuning of L_0 norm error, $\|\lambda^* - \hat{\lambda}\|_0$. C: The tuning of L_2 norm error, $\|\lambda^* - \hat{\lambda}\|_2$. D: The tuning of stationary firing rate of $\hat{\lambda}_{20}$ and $\hat{\lambda}_{50}$. E: The firing rate trajectories when $\varrho = 0.3$. F: The firing rate trajectories when $\varrho = 3$.

$$\lambda_i(t) = \frac{1}{1+\rho_i}[u_i(t) - (\varrho_i + \varepsilon_3(t))]_+, i = 1, \ldots, N \tag{30}$$

$\varepsilon_1(t)$, $\varepsilon_2(t)$, $\varepsilon_3(t)$ are time-varying noises matrix or vectors with proper dimensions, while each element of them is independently drawn from $[-0.05, 0.05]$. The inference problem has the same parameter setting as before. We let $\varrho_i = \varrho$ across all neurons, and tune ϱ to change from 0.03 to 3 in steps of 0.03. Here, we take the average of the last one percentage of elapsed time as the inferred latent cause $\hat{\lambda}$. Fig. 4 shows how the configured neural dynamics varies with ϱ for this setup. From Fig. 4A and B, a larger ϱ leads to a network that is more robust to noise with a better inference performance. In fact, Fig. 4B shows that the inferred latent cause is almost nonzero in each coordinate when ϱ is small ($\varrho = 0.03$), but is a sparse vector with the same nonzero coordinates as the true latent cause when ϱ is large ($\varrho = 3$). However, Fig. 4C and D show that the inference of the latent causes strength deteriorates as ϱ increases. Fig. 4E and F show that a larger ϱ will not only speed up the transient response but can also suppress trajectory variance due to the noise. Finally, comparing Figs. 2–3 and 4, the SNN can still discriminate the identities of latent causes with properly chosen ϱ. The simulation example demonstrates that the synthesized neural dynamics based on monotone operator theory enjoys certain robustness with respect to noises and parameter uncertainties, validating the effectiveness of the proposed method.

4.3. Structural prior

There could be additional structural prior information, describing the interaction among different latent causes. For example, when inferring a human face image from a linear composition model, we know some components must appear together (e.g., eyes, nose and mouth.) or some components cannot appear at the same time (e.g., more than two eyes). The structural prior information could be modeled as a linear inequality constraint

$$A\lambda - b \leq \mathbf{0}. \tag{31}$$

with $A = [a_1, \dots, a_J]^T \in \mathbb{R}^{J\times N}$ and $b \in \mathbb{R}^J$. Incorporation of such a structural prior information properly in the inference is desirable since it limits the search space. To be consistent with the previous Bayesian framework, we change the independent prior in (25) to

$$p(\lambda) \propto \prod_{i=1}^{N} \exp(-\varrho_i|\lambda_i| - \frac{\rho_i}{2}\lambda_i^2)H(\lambda_i) \times \prod_{j=1}^{J} H(b_j - a_j^T\lambda) \tag{32}$$

Then, the MAP inference problem in (26) becomes:

$$\min_{\lambda\in\mathbb{R}^N}, \frac{1}{2}(\mathbf{y} - Q\lambda)^T(\mathbf{y} - Q\lambda) + \sum_{i=1}^{N} R_i(\lambda_i), s.t., (31) \tag{33}$$

Note that (33) deviates from the sparse coding problem and thus the LCA algorithm does not directly apply here. However, from the MoI perspective, we can configure a SNN using the method in Section 3.2. The configured network has a total of $N + J$ neurons with firing rates $\lambda = col(\lambda^c, \lambda^\mu) \in \mathbb{R}^{N+J}$ and corresponding membrane voltages $u = col(u^c, u^\mu)$. We let $\lambda^c \in \mathbb{R}^N$ stand for the inferred latent causes, and let λ^μ stand for the multipliers of the linear constraints. We set $\tau_m = 1$, $\eta = \mathbf{1}$ and $\epsilon = 1$. Then with (3) and Lemma 2.2, the mean-field dynamics of the configured SNN is

$$\dot{u}^c = -u^c + \lambda^c - Q^TQ\lambda^c - A^T\lambda^\mu + Q^T\mathbf{y} \tag{34}$$

$$\dot{u}^\mu = -u^\mu + \lambda^\mu + A\lambda^c - b \tag{35}$$

$$\lambda_i^c(t) = \frac{1}{1+\rho_i}[u_i^c(t) - \varrho_i]_+, i = 1, \dots, N \tag{36}$$

$$\lambda_j^\mu(t) = [u_j^\mu(t)]_+, j = 1, \dots, J \tag{37}$$

Remark 4.2. The structural prior can help to prune the latent space in posterior inference, which is believed to be used by neural networks for efficiently performing cognition tasks. However, how to synthesize a recurrent neural network to perform inference with structural prior is not known yet. With the proposed methods, we can easily synthesize the mean-field neural dynamics like (34)–(37) to accommodate more complicated structural prior in Bayesian inference, which is not achieved by other neural dynamics synthesis methods, such as Moreno-Bote and Drugowitsch (2015), Tang (2016), and Tang et al. (2017). In fact, the proposed monotone operator framework can be treated as a generalization of the synthesis method in Moreno-Bote and Drugowitsch (2015), Tang (2016) and Tang et al. (2017), since monotone inclusion can cover general decision making problems other than optimization problems. With the synthesis methods of this work, more practical decision making problems can be solved with a synthesized recurrent neural dynamics, such as the Nash game, the saddle point problem, the Markovian decision process, etc.

Example 4.3. As in Example 4.1, we take $N = 100, m = 10$. We generate feature vectors $Q = [q_1, \dots, q_{100}]$ and a observation $\mathbf{y}$ as follows. We first generate $\{q_1, \dots, q_{50}\}$ with each element of the feature vector being independently and uniformly drawn from $[0, 1]$. And then we set $q_i = q_{i-50} + \varepsilon_i$ for $i = 51, \dots, 100$ with each coordinate of ε_i being an i.i.d. Gaussian noise of zero mean and variance of 0.001. Then each feature vector is normalized to 1, $\|q_i\|_2 = 1$. Hence, the feature vector q_i is highly correlated with q_{i+50} for $i = 1, \dots, 50$. Then, $\mathbf{y}$ is generated as $\mathbf{y} = 9 * q_{20} + 4 * q_{50} + \varepsilon$ with each element of ε being an i.i.d. Gaussian noise with zero mean and variance of 0.001. The true latent cause $\lambda^* \in \mathbb{R}^{100}$ is a sparse high dimensional vector with $\lambda_{20}^* = 9$, $\lambda_{50}^* = 4$ and all other coordinates being zeros.

We incorporate additional structural information (31) by setting $A = [a_1, a_2]^T \in \mathbb{R}^{2\times 100}$ and $b = (b_1, b_2)^T \in \mathbb{R}^2$ with $a_1 = col(-\mathbf{1}_{50}, \mathbf{0}_{50})$ and $a_2 = col(\mathbf{0}_{50}, \mathbf{1}_{50})$. This structural prior imposes that the sum of first 50 latent causes is larger than $-b_1$ while the sum of last 50 latent causes is smaller than b_2. We let b to be $b^1 = (-5, 4)^T$, $b^2 = (-8, 3)^T$, $b^3 = (-11, 2)^T$, respectively. This could be treated as a refinement of structural prior when b changes from b^1 to b^2 and finally to b^3, during which the true latent cause gets closer to the boundary of the set specified by $A\lambda \leq b$. For each given b, we set $\rho_i = \rho = 0.001$ and $\varrho_i = \varrho$ for all neurons. We tune ϱ to change from 0.05 to 5 to study the interplay of the constraint parameter b and the sparsity parameter ϱ. We configure an SNN with (34)–(37) for each parameter setting. The stationary firing rate is denoted as $\hat{\lambda} = col(\hat{\lambda}^c, \lambda^\mu)$ with $\hat{\lambda}^c$ as inferred latent cause and $\hat{\lambda}^\mu$ as optimal multiplier. For comparison, we also configure an SNN using (27)–(28) (i.e., without structural prior). We set the same $\rho_i = \rho = 0.001$ and tune $\varrho_i = \varrho$ the same as before.

Fig. 5 compares the inference error with structural prior (32) and with independent prior (25) as a function of ϱ. Fig. 6 gives the inferred 20th, 50th, 70th, and 100th latent causes with (32) or (25) when tuning ϱ. Both Fig. 5A and B show that with the increase of ϱ the inferred latent cause vector becomes sparser and gets closer to the true latent cause in term of L_0 norm error. However, Fig. 5B indicates that with only the independent prior we cannot distinguish q_{20}, q_{50} from q_{70}, q_{100}, therefore, the L_0 norm error $\|\lambda^* - \hat{\lambda}\|_0$ can only decreases to 4. This is also evident in Fig. 6D, where all $\hat{\lambda}_{20}$ $\hat{\lambda}_{50}$, $\hat{\lambda}_{70}$, $\hat{\lambda}_{100}$ are nonzero when inferring with (25). The above comparative analysis validates that for practical casual inference problem with subtle features, the incorporation of a structural prior can significantly improve the inference fidelity. Moreover, Fig. 5A shows that incorporating (31) can further decrease the L_0 norm error with suitable choice of ϱ. Combining Fig. 6B and C, we see $\hat{\lambda}_{70}^c$ and $\hat{\lambda}_{100}^c$ will decrease to zero and $\|\lambda^* - \hat{\lambda}^c\|_0$ can decrease to 2 with b^2 and b^3 if ϱ is large enough, implying the identities of nonzero latent causes are correctly discriminated. Fig. 5A also shows that a more refined prior with b^3 leads to a better inference performance than a coarse prior with b^2 or b^1. Fig. 5C and D show the tuning of L_2 norm error with ϱ when inferring with (32) and (25), respectively. Fig. 5D shows that $\|\lambda^* - \hat{\lambda}\|$ increases as ϱ get larger when only an independent prior (25) is used. Fig. 5C shows that $\|\lambda^* - \hat{\lambda}^c\|$ can be bounded if (31) is also incorporated, which can also be found in Fig. 6A,B, and C. Moreover, combining Figs. 5C and 6C, we see the $\|\lambda^* - \hat{\lambda}^c\|$ is non-increasing as ϱ gets larger with a refined parameter b^3. We conclude that with a proper ϱ the structural prior (31) can help to improve the inference of both the identity and strength of the latent causes.

Fig. 7 shows how the multiplier λ^μ tunes with different constraint parameter b and sparsity parameter ϱ when inferring with prior (32). Fig. 7A, B and C give the tuning of the steady state multipliers $\hat{\lambda}^\mu$ with ϱ when b is b^1, b^2 and b^3, respectively. When $b = b^1$ the true latent cause λ^* is far away from the boundary of the constraint set specified by (31), and Fig. 7A indicates that (31) does not influence the inference in this case. Fig. 7B and C show that (31) starts to play a role in the inference only if ϱ is large enough. Since the threshold nonlinearity in (36) makes the inferred latent cause sparser as ϱ gets larger, the values of λ_{20}, λ_{50}, λ_{70} and λ_{100} has to increase to explain the observation. However, when b equals to b^2 or b^3, the true latent cause λ^* is

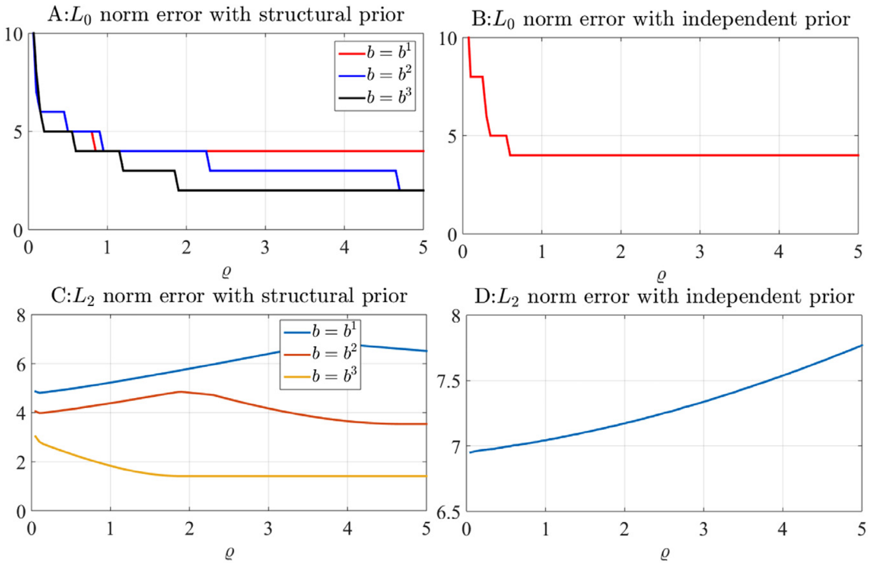


Fig. 5. The comparison of inference results with structural prior (32) and with independent prior (25) when tuning the sparsity parameter ϱ. A: The tuning of L_0 norm error with ϱ, $\|\lambda^* - \hat{\lambda}^c\|_0$, with (32) and different b. B: The tuning of L_0 norm error with ϱ, $\|\lambda^* - \hat{\lambda}\|_0$, with (25). C: The tuning of L_2 norm error with ϱ, $\|\lambda^* - \hat{\lambda}^c\|_2$, with (32) and different b. D: The tuning of L_2 norm error with ϱ, $\|\lambda^* - \hat{\lambda}\|_2$, with (25).

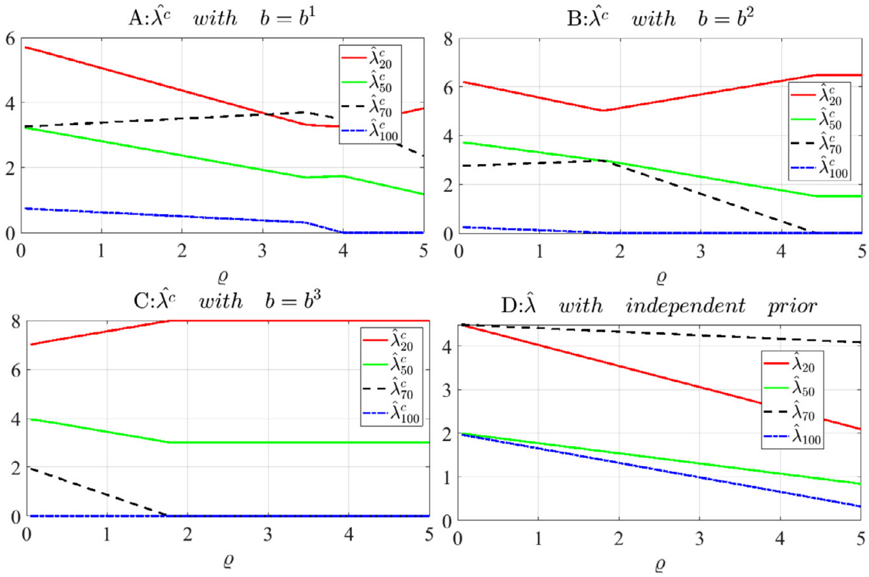


Fig. 6. The stationary firing rate as the inferred $\hat{\lambda}^c_{20}$ $\hat{\lambda}^c_{50}$, $\hat{\lambda}^c_{70}$, $\hat{\lambda}^c_{100}$ with different constraint parameters when tuning the sparsity parameter ϱ. (A), (B), (C) are the inferred causes with structural prior (32) when tuning ϱ with b^1, b^2 and b^3, respectively. D: The inferred causes with independent prior (25) when tuning ϱ.

closer to the boundary of the constraint set specified by (31). Once (31) is violated, the multiplier λ^μ becomes nonzero and starts to help the network to only increase the correct causes to explain away the observation. Fig. 7D, E, F give the transition trajectories of the firing rates for λ^μ with $\varrho = 5$ when b is b^1, b^2 and b^3, respectively. They show that the firing rates standing for multiplier λ^μ can converge to a stationary equilibrium quickly.

Fig. 8 shows transient firing rates for λ^c with different combination of structural prior parameter b and sparsity prior parameter ϱ.

By comparing the rows of Fig. 8 (for example Fig. 8C and I), the dynamics converges to equilibrium faster with a larger sparsity parameter ϱ. This implies that the threshold nonlinearity in (36) can promote inference sparsity and accelerate convergence. By comparing the columns of Fig. 8 (for example Fig. 8G and I), we see with a more refined structural prior b, the transient is prolonged, but converges to a more accurate inference of latent causes. This is because it takes time for the multiplier of the structural prior to evolve before affecting the inference. The figure shows the convergence of the configured dynamics with different parameters.

5. Conclusion

We provided a normative theory regarding the ability of recurrent neural circuits to embed the problem of monotone inclusion, which encompasses a wide range of optimization problems. We show that in a rate-based Poisson SNN, each neuron's firing (activation) curve can be treated as a proximal operator of a local regularizer and hence the overall dynamics amounts to an operator-splitting type ODE solver for MoI. We are then able to give a top-down approach to synthesize SNN for MoI as a

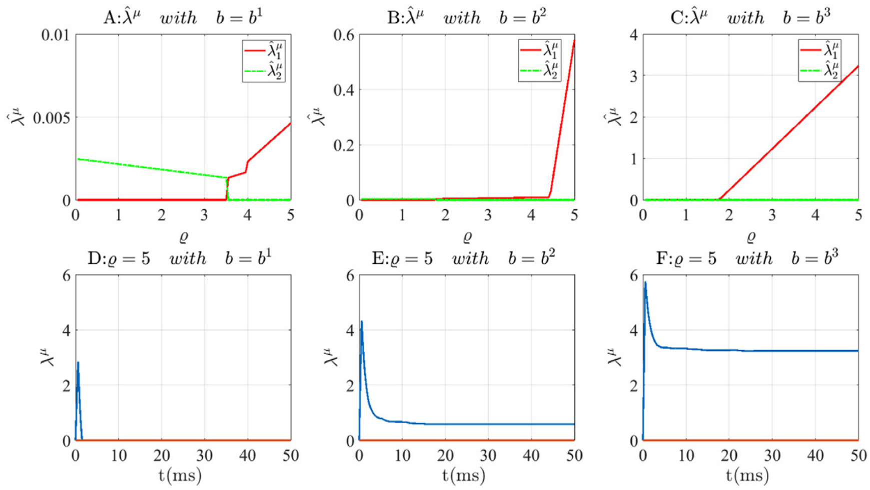


Fig. 7. The steady state and transition behavior of the multiplier λ^μ in (34)–(37) with different b and ϱ. (A), (B) and (C) are the tuning of steady $\hat{\lambda}^\mu$ with ϱ when $b=b^1$, $b=b^2$, $b=b^3$, respectively. (D), (E), and (F) are the transition trajectories of λ^μ with $\varrho=5$ when $b=b^1$, $b=b^2$, $b=b^3$, respectively.

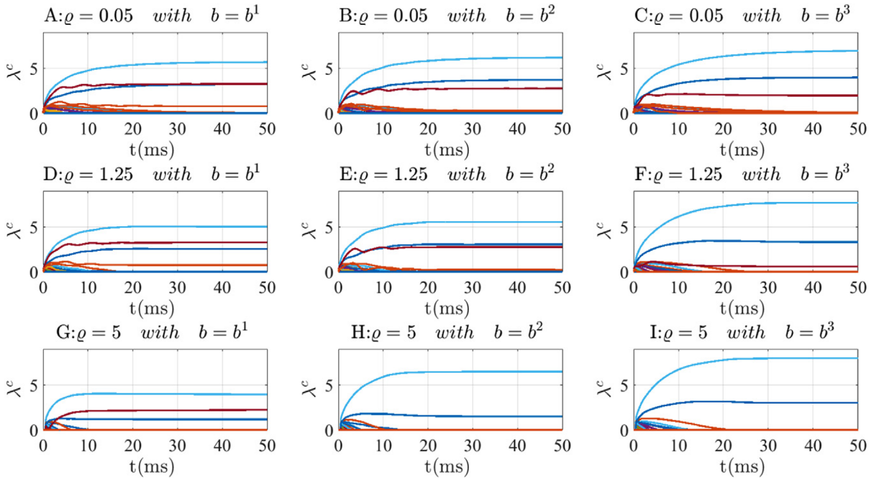


Fig. 8. The transition trajectories of the neural firing rates for λ^c by dynamics (34)–(37) with a different combination of constraint prior b and sparsity prior ϱ.

surrogate for salient functions. To illustrate this, we performed numerical studies in the context of Bayesian causal inference, including treatment of a structural prior that falls outside of the scope of existing works on sparse neural coding.

The key contribution of our work is a perspective on the unifying framework to understand the ability of neural circuits to carry out optimization. Indeed, the MoI framework could be used to synthesize networks for other problems of interest in neuroscience and neural computation, including Nash equilibrium calculation and Markov decision problems. As well, the ease of our synthesis procedure makes it potentially appealing for neuromorphic instantiations.

There remain several open questions regarding out results, including more detailed validation of the fidelity of the mean-field approximation, as well as convergence and rate analysis under a weaker assumption. As well, it is of obvious interest to consider the possibility of treating more complicated nonlinear problems. For this, one intriguing possibility is to successively approximate a nonlinear problem with linear approximations using a process analogous to synaptic plasticity, thus accommodate the normative theory of this work to the adaption and plasticity of biological neurons.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Monotone operator and fixed points

Let $\mathfrak{A}:\mathbb{R}^m\to 2^{\mathbb{R}^m}$ be a set-valued operator. Denote Id as the identity operator, i.e, $\text{Id}(x)=x$. The graph of $\mathfrak{A}$ is $gra\mathfrak{A}=\{(x,u)|u\in\mathfrak{A}x\}$, then the inverse of $\mathfrak{A}$ is defined through its graph as $gra\mathfrak{A}^{-1}=\{(u,x)|(x,u)\in gra\mathfrak{A}\}$. The zero set of $\mathfrak{A}$ is $zer\mathfrak{A}=\{x\in\mathbb{R}^m|\mathbf{0}\in\mathfrak{A}x\}$. $\mathfrak{A}$ is called monotone if $\forall(x,u),\forall(y,v)\in gra\mathfrak{A}$, we have $\langle x-y,u-v\rangle\geq 0$. Define the resolvent of $\mathfrak{A}$ as $R_{\mathfrak{A}}=(\text{Id}+\mathfrak{A})^{-1}$. For a proper *lower semi-continuous convex* (l.s.c.) function f, its sub-differential operator ∂f is $x\mapsto\{g|f(y)\geq f(x)+\langle g,y-x\rangle,\forall y\}$. ∂f is a monotone operator. Then $Prox_f=R_{\partial f}$ is called the proximal operator of f, i.e., $Prox_f:x\mapsto\arg\min_{u\in domf}f(u)+\frac{1}{2}\|u-x\|_2^2$. Define the indicator function of Ω as $\iota_\Omega(x)=0$ if $x\in\Omega$ and $\iota_\Omega(x)=\infty$ if $x\notin\Omega$. For a closed convex set Ω, ι_Ω is a proper l.s.c. function. $\partial\iota_\Omega$ is also the normal cone operator of Ω, i.e., $N_\Omega(x)$, where $N_\Omega(x)=\{v|\langle v,y-x\rangle\leq 0,\forall y\in\Omega\}$ and $domN_\Omega=\Omega$. For a single-valued operator $T:\mathbb{R}^m\to\mathbb{R}^m$, a point $x\in\mathbb{R}^m$ is a fixed point of T if $Tx=x$, and the set of fixed points of T is denoted as $FixT$. Assume $\mathfrak{A}$ is single-valued and $\mathfrak{B}$ is monotone, then $zer(\mathfrak{A}+\mathfrak{B}) = FixR_{\mathfrak{B}} \circ (\mathrm{Id} - \mathfrak{A})$.

References

Abbas, Boushra, & Attouch, Hedy (2015). Dynamical systems and forward–backward algorithms associated with the sum of a convex subdifferential and a monotone cocoercive operator. *Optimization*, *64*(10), 2223–2252.

Attouch, Hedy, Cabot, Alexandre, & Czarnecki, Marc-Olivier (2018). Asymptotic behavior of nonautonomous monotone and subgradient evolution equations. *Transactions of the American Mathematical Society*, *370*(2), 755–790.

Attouch, H., & Svaiter, B. F. (2011). A continuous dynamical newton-like approach to solving monotone inclusions. *SIAM Journal on Control and Optimization*, *49*(2), 574–598.

Balavoine, Aurèle, Romberg, Justin, & Rozell, Christopher J. (2012). Convergence and rate analysis of neural networks for sparse approximation. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(9), 1377–1389.

Barrett, David G., Denève, Sophie, & Machens, Christian K. (2013). Firing rate predictions in optimal balanced networks. In *Advances in neural information processing systems* (pp. 1538–1546).

Bauschke, Heinz H., Combettes, Patrick L., et al. *Convex analysis and monotone operator theory in Hilbert spaces, Vol. 408*. Springer.

Beck, Amir, & Teboulle, Marc (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, *2*(1), 183–202.

Boggs, Paul T., & Tolle, Jon W. (1995). Sequential quadratic programming. *Acta Numerica*, *4*, 1–51.

Boţ, Radu Ioan, & Csetnek, Ernö Robert (2018). Convergence rates for forward–backward dynamical systems associated with strongly monotone inclusions. *Journal of Mathematical Analysis and Applications*, *457*(2), 1135–1152.

Chen, Dingyuan, Zhang, Weiwei, Cao, Jinde, & Huang, Chuangxia (2020). Fixed time synchronization of delayed quaternion-valued memristor-based neural networks. *Advances in Difference Equations*, *2020*(1), 1–16.

Chou, Chi-Ning, Chung, Kai-Min, & Lu, Chi-Jen (2018). On the algorithmic power of spiking neural networks. arXiv preprint arXiv:1803.10375.

Combettes, Patrick L., & Pesquet, Jean-Christophe (2007). Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM Journal on Optimization*, *18*(4), 1351–1376.

Combettes, Patrick L., & Pesquet, Jean-Christophe (2011). Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering* (pp. 185–212). Springer.

Combettes, Patrick L., & Pesquet, Jean-Christophe (2019). Lipschitz Certificates for neural network structures driven by averaged activation operators. arXiv preprint arXiv:1903.01014.

Combettes, Patrick L., & Pesquet, Jean-Christophe (2020). Deep neural network structures solving variational inequalities. *Set-Valued and Variational Analysis*, 1–28.

Dold, Dominik, Bytschok, Ilja, Kungl, Akos F, Baumbach, Andreas, Breitwieser, Oliver, Senn, Walter, et al. (2019). Stochasticity from function why the bayesian brain may need no noise. *Neural Networks*, *119*, 200–213.

Friedrich, Johannes, & Lengyel, Máté (2016). Goal-directed decision making with spiking neurons. *Journal of Neuroscience*, *36*(5), 1529–1546.

Friston, Karl (2010). The free-energy principle: a unified brain theory?. *Nature Reviews Neuroscience*, *11*(2), 127–138.

Gangopadhyay, Ahana, & Chakrabartty, Shantanu (2017). Spiking, bursting, and population dynamics in a network of growth transform neurons. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(6), 2379–2391.

Gerstner, Wulfram, Kistler, Werner M., Naud, Richard, & Paninski, Liam (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Gill, Philip E., Murray, Walter, & Saunders, Michael A. (2005). Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Review*, *47*(1), 99–131.

Hao, Yunzhe, Huang, Xuhui, Dong, Meng, & Xu, Bo (2020). A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule. *Neural Networks*, *121*, 387–395.

Hu, Haijun, Yi, Taishan, & Zou, Xingfu (2020). On spatial-temporal dynamics of a fisher-kpp equation with a shifting environment. *Proceedings of the Americal Mathematical Society*, *148*(1), 213–221.

Huang, Chuangxia, Cao, Jie, Wen, Fenghua, & Yang, Xiaoguang (2016). Stability analysis of sir model with distributed delay on complex networks. *PLOS ONE*, *11*(8), 1–22.

Huang, Fuqiang, & Ching, ShiNung (2019). Spiking networks as efficient distributed controllers. *Biological Cybernetics*, *113*(1–2), 179–190.

Huang, Chuangxia, Long, Xin, & Cao, Jinde (2020). Stability of antiperiodic recurrent neural networks with multiproportional delays. *Mathematical Methods in the Applied Sciences*, *43*(9), 6093–6102.

Huang, Chuangxia, Zhang, Hua, Cao, Jinde, & Hu, Haijun (2019). Stability and hopf bifurcation of a delayed preycpredator model with disease in the predator. *International Journal of Bifurcation and Chaos*, *29*(07), Article 1950091.

Huang, Chuangxia, Zhang, Hua, & Huang, Lihong (2019). Almost periodicity analysis for a delayed nicholson's blowflies model with nonlinear density-dependent mortality term. *Communications on Pure & Applied Analysis*, *18*(6), 3337–3349.

Jang, Hyeryung, Simeone, Osvaldo, Gardner, Brian, & Grüning, André (2018). Spiking neural networks: A stochastic signal processing perspective. arXiv preprint arXiv:1812.03929.

Jolivet, Renaud, Rauch, Alexander, Lüscher, Hans-Rudolf, & Gerstner, Wulfram (2006). Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of Computational Neuroscience*, *21*(1), 35–49.

Kafashan, MohammadMehdi, & Ching, ShiNung (2017). Recurrent networks with soft-thresholding nonlinearities for lightweight coding. *Neural Networks*, *94*, 212–219.

Knill, David C., & Richards, Whitman (1996). *Perception as Bayesian inference*. Cambridge University Press.

Kudu, Mustafa (2018). A parameter uniform difference scheme for the parameterized singularly perturbed problem with integral boundary condition. *Advances in Difference Equations*, *2018*(1), 1–12.

Li, Wenjie, Huang, Lihong, & Ji, Jinchen (2019). Periodic solution and its stability of a delayed beddington-deangelis type predator-prey system with discontinuous control strategy. *Mathematical Methods in the Applied Sciences*, *42*(13), 4498–4515.

Li, Xiuwen, Liu, Zhenhai, Li, Jing, & Tisdell, Chris (2019). Existence and controllability for nonlinear fractional control systems with damping in hilbert spaces. *Acta Mathematica Scientia*, *39*(1), 229–242.

Maass, Wolfgang (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671.

Moreno-Bote, Rubén, & Drugowitsch, Jan (2015). Causal inference and explaining away in a spiking network. *Scientific Reports*, *5*, 17531.

Nocedal, Jorge, & Wright, Stephen (2006). *Numerical optimization*. Springer Science & Business Media.

Olshausen, Bruno A., & Field, David J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1?. *Vision Research*, *37*(23), 3311–3325.

Pérez, Javier, Cabrera, Juan A., Castillo, Juan J., & Velasco, Juan M. (2018). Bio-inspired spiking neural network for nonlinear systems control. *Neural Networks*, *104*, 15–25.

Pillow, Jonathan W., Paninski, Liam, Uzzell, Valerie J., Simoncelli, Eero P., & Chichilnisky, E. J. (2005). Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, *25*(47), 11003–11013.

Rezende, Danilo J., Wierstra, Daan, & Gerstner, Wulfram (2011). Variational learning for recurrent spiking networks. In *Advances in neural information processing systems* (pp. 136–144).

Rozell, Christopher J., Johnson, Don H., Baraniuk, Richard G., & Olshausen, Bruno A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural Computation*, *20*(10), 2526–2563.

Schuman, Catherine D., Potok, Thomas E., Patton, Robert M., Birdwell, J. Douglas, Dean, Mark E., Rose, Garrett S., et al. (2017). A survey of neuromorphic computing and neural networks in hardware. arXiv preprint arXiv:1705.06963.

Song, Chao, Fei, Shumin, Cao, Jinde, & Huang, Chuangxia (2019). Robust synchronization of fractional-order uncertain chaotic systems based on output feedback sliding mode control. *Mathematics*, *7*(7), 599.

Taherkhani, Aboozar, Belatreche, Ammar, Li, Yuhua, Cosma, Georgina, Maguire, Liam P., & McGinnity, T. Martin (2020). A review of learning in biologically plausible spiking neural networks. *Neural Networks*, *122*, 253–272.

Tang, Ping Tak Peter (2016). Convergence of lca flows to (c) lasso solutions. arXiv preprint arXiv:1603.01644.

Tang, Ping Tak Peter, Lin, Tsung-Han, & Davies, Mike (2017). Sparse coding by spiking neural networks: Convergence theory and computational results. arXiv preprint arXiv:1705.05475.

Tenenbaum, Joshua B., Griffiths, Thomas L., & Kemp, Charles (2006). Theory-based bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, *10*(7), 309–318.

Yoon, Young C. (2016). Lif and simplified srm neurons encode signals into spikes via a form of asynchronous pulse sigma–delta modulation. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(5), 1192–1205.

Zambrano, Davide, Nusselder, Roeland, Steven Scholte, H., & Bohté, Sander M. (2018). Sparse computation in adaptive spiking neural networks. *Frontiers in Neuroscience*, *12*.

Zhang, Jian, & Huang, Chuangxia (2020). Dynamics analysis on a class of delayed neural networks involving inertial terms. *Advances in Difference Equations*, *2020*(1), 1–12.

Zhou, Ya, Wan, Xiaoxiao, Huang, Chuangxia, & Yang, Xinsong (2020). Finite-time stochastic synchronization of dynamic networks with nonlinear coupling strength via quantized intermittent control. *Applied Mathematics and Computation*, *376*(C).