Degenerate parametric distributions of a model of superior colliculus performing flexible routing

Sean Bittner

March 8, 2019

1 Introduction

Our collaborators (Duan et al. 2019, in prep.), have compelling data suggesting that superior colliculus (SC) is involved in flexible routing of information. They have designed a nonlinear dynamical model of SC, and seek to characterize the parameterizations that reproduce the emergent properties of their dataset. They have already performed a simulate-then-characterize based analysis that has provided substantial insight. We will explore how degenerate solution networks (DSNs) can provide deeper insight about the model through the full distribution of model parameterizations, which reproduces the experimental findings.

In the manuscript, satisfactory model parameterizations are found from many initializations. Statements about the sign of connectivity parameters (the excitatory or inhibitory nature of these projections) are made based on counts from the 373 total parameterizations (e.g. PRO $\leftrightarrow$ ANTI same hemisphere are almost always inhibitory, or PRO $\leftrightarrow$ ANTI cross hemisphere are almost always excitatory). The goal here with DSNs is to use machine learning to learn a proabilitistic mapping from parameter space to task performance (and other properties) in a way that is optimized to find all possible parameterizations. Once a DSN is optimized, we can effectively sample arbitarily many times from this degenerate parameteric distribution. This allows greater statistical power for statements on the probability of weight sign, shur mode contribution, or even implications on behavior in less-probable regions of parameter space (e.g. excitatory PRO $\leftrightarrow$ ANTI same hemisphere connections). Since DSNs are optimized to identify the full (maximum entropy) distribution of parameterizations, they may reveal new, unexpected solutions, whose mechanistic implementation of the flexible routing computation may be scientifically relevant.

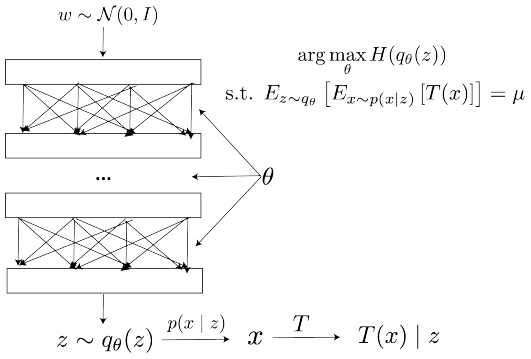
2 Method: Degenerate solution networks (DSNs)

Theoretical neuroscientists seek to design models or find parameterizations of models that yield emergent properties of behavior. These emergent properties (or high-level descriptions) of behavior contrast with say a collection of raw data points from an experiment. We have a host of methods from Bayesian machine learning that learn posteriors conditioning on a collection of experimentally collected data points. But, how do we do inference conditioned on statistically defined emergent properties of interest? We have developed a novel machine learning methodology for training degenerate solution networks (DSNs), which learn the full (i.e. maximum entropy) distribution of model parameterizations that yield a statistically defined behavior of interest. DSNs are trained to emit samples of maximum entropy distributions of model parameters, along with the density of such samples, and are thusly termed "density" networks. DSNs were designed to be a tool for theoretical neuroscientists enabling exploratory analyses, and scientific hypothesis testing with models.

Consider model parameterization z and data x generated from the model $p(x \mid z)$ with known sampling procedure. To train a DSN, the actual likelihood of this model may or may not be known. For example, in the nonlinear dynamical SC model, there is a known sampling procedure for simulating activity given the circuit parameterization, yet an explicit likelihood function for the generated

Sean Bittner
srb2201@columbia.edu

Figure 1: Degenerate solution network



neural activity is unavailable due to the complex nonlinear dynamics. In deep generative density networks, a simple random variable $w \sim p_0$ is mapped deterministically via a function f_θ parameterized by a neural network to the support of the distribution of interest where $z = f_\theta(w)$. Given a model $p(x \mid z)$ and some behavior of interest $\mathcal{B} : E_{z\sim q_\theta}\left[E_{x\sim p(x|z)}\left[T(x)\right]\right] = \mu$, DSNs are trained by optimizing the deep generative parameters θ to find the optimal approximation q^*_θ within the deep generative variational family Q to $p(z \mid \mathcal{B})$. This procedure is loosely equivalent to variational inference (VI) using a deep generative variational family with respect to the likelihood of the mean sufficient statistics rather than the data itself [1, 2]. In most settings (especially those relevant to theoretical neuroscience) the likelihood of the behavior with respect to the model parameters $p(T(x) \mid z)$ is unknown or intractable, requiring an alternative to stochastic gradient variational bayes [3] or black box variational inference [4]. Instead, DSNs are optimized with the following objective for a given model and statistical constraints on its produced activity:

$$\begin{aligned} q^*_\theta(z) = &\underset{q_\theta \in Q}{\operatorname{argmax}}\, H(q_\theta(z)) \\ &\text{s.t. } E_{z\sim q_\theta}\left[E_{x\sim p(x|z)}\left[T(x)\right]\right] = \mu \end{aligned} \tag{1}$$

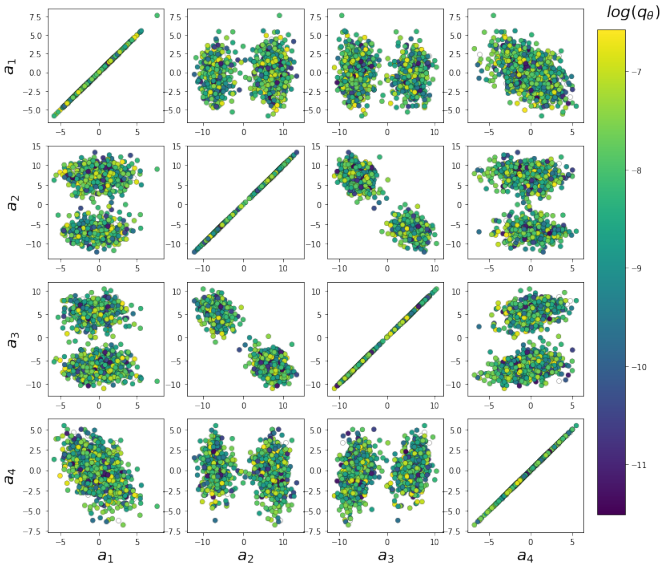
For the SC model, the means and variances of steady state responses across conditions would be reflected as first and second moment constraints on those sufficient statistics in $\mathcal{B}$. We use an augmented Lagrangian [5] constrained optimization procedure to train the DSN.

2.1 Example: 2-D linear system

To gain intuition for DSNs, consider degenerate parameterizations of two-dimensional linear dynamical systems, $\tau\dot{x} = Ax$ with $A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ that produce a band of oscillations. To train a DSN to learn the maximally entropic distribution of real entries of the dynamics matrix $z = [a_1, a_2, a_3, a_4]$ that yield a band of oscillations, $T(x)$ is chosen to contain the first- and second-moments of the oscillatory frequency ω and the the primary growth/decay factor c of the oscillating system. To learn the distribution of real entries of A that yield a c around zero with variance 1.0, and oscillations at 1 Hz with variance 0.025, the behavior of DSN is constrained to:

$$\mu = E\begin{bmatrix} c \\ \omega \\ c^2 \\ \omega^2 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 1.0 \\ 1.0 \\ 1.025 \end{bmatrix} \tag{2}$$

Figure 2: Pairplot of oscillating 2D linear system degenerate parameterization distribution.



We trained a DSN to learn the degenerate linear system parameterization (Fig. 2). Even this relatively simple system has nontrivial (though intuitively sensible) structure in the parameter distribution. Indeed, more subtle model-behavior combinations will have even more complexity, further motivating DSNs.

3 Experimental motivation

This section briefly summarizes my understanding of the experimental motivation of SC model. If I have misunderstood anything, or missed something important, please let us know!

Task: Cue indicates Pro or Anti condition. Following a delay after the cue, the mouse is supposed to lick right/left if the light stimulus is on the right/left in the Pro condition, and left/right in the Anti condition.

Electrophysiological recordings from SC and PFC indicate a much stronger encoding of task context in SC than PFC, and a significantly earlier representation of choice than PFC. This is discordant with the prevailing understanding that PFC sends the output of flexible routing to SC.

SC neurons can be succinctly sorted into two groups – cue neurons and delay/choice neurons by the period (before and after cue removal, respectively) of the task in which they strongly encode task context. Furthermore, these delay/choice neurons have relatively distorted representations of task context on error trials.

There was a surprisingly high fidelity match between Pro-tuned neurons and Contra-response-tuned neurons, and between Anti-tuned neurons and Ipsi-response-tuned neurons.

During bilateral SC silencing experiments, performance was unnaffected during task cue and choice period silencing. In delay period inactivation experiments, only performance in the Anti condition was reduced significantly.

4 SC model

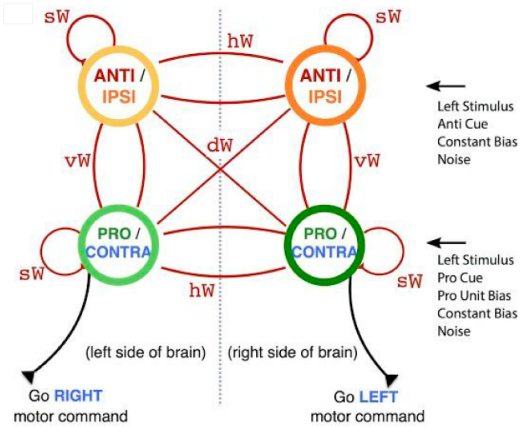
The authors designed a nonlinear dynamical model of SC (Fig. 3) and found parameterizations which yielded a high-level description of these experimental results.

There are four total units: two in each hemisphere corresponding to the PRO/CONTRA and ANTI/IPSI populations. Each unit had an external (V_i) and internal (U_i) variable related by

$$V_i(t) = \eta(t)\left(\frac{1}{2}\tanh\left(\frac{U_i(t)-\theta}{\beta}\right)+\frac{1}{2}\right) \quad (3)$$

$\theta = 0.05$ and $\beta = 0.5$ control the position and shape of the nonlinearity, repsectively, and $\eta(t)$ is the optogenetic inactivation function.

Figure 3: SC model from (Duan et al. 2019).



We can order the elements of V_i and U_i into vectors v and u with elements

$$v = \begin{bmatrix} V_{LP} \\ V_{LA} \\ V_{RA} \\ V_{RP} \end{bmatrix} \qquad u = \begin{bmatrix} U_{LP} \\ U_{LA} \\ U_{RA} \\ U_{RP} \end{bmatrix} \tag{4}$$

The internal variables follow dynamics:

$$\tau \frac{\partial u}{\partial t} = -u + Wv + I + \sigma \partial W \tag{5}$$

with time constant $\tau = 0.09s$ and gaussian noise $\sigma \partial W$ controlled by the magnitude of σ. The weight matrix has 8 parameters sW_P, sW_A, vW_{PA}, vW_{AP}, hW_P, hW_A, dW_{PA}, and dW_{AP}, related to the depiction in Fig. 2:

$$W = \begin{bmatrix} sW_P & vW_{PA} & dW_{PA} & hW_P \\ vW_{AP} & sW_A & hW_A & dW_{AP} \\ dW_{AP} & hW_P & sW_A & vW_{AP} \\ hW_A & dW_{PA} & vW_{PA} & sW_P \end{bmatrix} \tag{6}$$

The input is a sum of five constant scalar parameters:

$$I = I_{\text{constant}} + I_{\text{pro-bias}} + I_{\text{rule}} + I_{\text{choice-period}} + I_{\text{light}} \tag{7}$$

5 Next steps using DSNs

5.1. Free parameters

Our eventual goal is to train a DSN to learn a max-ent distribution on

$$z = \begin{bmatrix} \theta \\ \beta \\ \sigma \\ sW_P \\ sW_A \\ vW_{PA} \\ vW_{AP} \\ hW_P \\ hW_A \\ dW_P \\ dW_A \\ \tau \\ I_x \\ .. \end{bmatrix}^\top \tag{8}$$

producing some emergent property. To do an initial proof of concept, we'll start by fixing many of these parameters to something reasonable, and learning a DSN on say 4-8 variables of interest for tractability of interpretation/validation in the initial stages. I think it makes sense to start with

$$z = \begin{bmatrix} sW_P \\ sW_A \\ dW_P \\ dW_A \end{bmatrix} \tag{9}$$

or

$$z = \begin{bmatrix} sW_P \\ sW_A \\ vW_{PA} \\ vW_{AP} \\ hW_P \\ hW_A \\ dW_P \\ dW_A \end{bmatrix} \tag{10}$$

and fix the rest of the parameters to something reasonable. Is there a better/more interesting first set of parameters to work with?

5.2. Statistically specifying the behavior – task responses

We are interested in finding model parameterizations, which yield high accuracy in Pro trials across no inacitviation (NI), delay period inactivation (DI), and choice period inactivation (CI). Additionally, we want high accuracy on Anti trials during NI and CI, but low accuracy during DI.

In the manuscript, this was done by training SC models for many parameterizations, and keeping track of the parameterizations that yielded the task performance just described. This procedure gives us a set of satisfactory parameterizations, which can be analyzed and examined to gain an understanding of the parameterizations that produce these results and the different possibilities of response types.

With DSNs, we can use machine learning to find a maximally expansive distribution of such model parameterizations that produce the desired task responses. This distribution on parameterizations gives us additional insight to the structure of the degenerate space of parameters that produces those task responses. DSNs facilitate complementary analyses for understanding the implications of all SC models that produce these responses.

5.1 Training a DSN on the SC model and results

Let $V_\alpha, \text{ss} = V_\alpha(t = 1.85)$ be the activity of neuron type $\alpha \in [LP, LA, RA, RP]$ (ss means steady-state), where we have a 1.2 second cue + delay period and a 0.6s choice period. We want $p_{a,b}$ probability of success ($a \in [P, A]$ and $b \in [NI, DI, CI]$), with strong responses – high or low activity when either succeeding or failing. In the trained networks of Duan et al. 2019, this either high-or-low response type is encouraged by using term C_2 of the cost function (methods of Duan et al. 2019). With DSNs, we can require that the responses approximately have the properties of Bernoulli variables (which are all or none with some rate).

For a given parameter vector sample z_i from the DSN, let's consider M frozen noise realizations $\sigma\partial W_j \sim \mathcal{N}(0, \sigma) \in \mathcal{R}^T$, where $T = \frac{1.8}{\partial t = 0.024}$. To encourage the steady state responses of the left and right PRO/CONTRA neurons across the various task conditions to behave as Bernoulli variables, we can ask that for the pro condition with stimulus $s \in [L, R]$:

$$E_{\sigma\partial W}\left[V_{LP,\text{ss}} \mid s = L, b, z_i\right] = \frac{1}{M}\sum_{j=1}^{M} V_{LP,\text{ss}}(s = L, b, z_i, \sigma\partial W_j) = p_{P,b}$$

$$E_{\sigma\partial W}\left[V_{LP,\text{ss}} \mid s = R, b, z_i\right] = 1 - p_{P,b}$$

$$E_{\sigma\partial W}\left[V_{RP,\text{ss}} \mid s = L, b, z_i\right] = 1 - p_{P,b}$$

$$E_{\sigma\partial W}\left[V_{RP,\text{ss}} \mid s = R, b, z_i\right] = p_{P,b}$$

$Var_{\sigma\partial W}(V_{LP,\text{ss}} \mid s = L, b, z_i) = Var_{\sigma\partial W}(V_{LP,\text{ss}} \mid s = R, b, z_i) = Var_{\sigma\partial W}(V_{RP,\text{ss}} \mid s = L, b, z_i) = Var_{\sigma\partial W}(V_{RP,\text{ss}} \mid s = R, b, z_i) = p_{P,b}(1 - p_{P,b})$

and the analagous equations for the ANTI condition. As a reminder a Bernoulli random variable $x \in [0, 1]$ with parameter p has $E[x] = p$ and $Var(x) = p(1-p)$.

DSNs enforce statistics in expectation of samples $z_i \sim q_\theta(z_i)$ from the DSN. So we will end up imposing constraints on the vector:

$$E_{z\sim q_\theta(z)} \begin{bmatrix} E_{\sigma\partial W}\left[V_{LP,\text{ss}} \mid s = L, b, z\right] \\ E_{\sigma\partial W}\left[V_{LP,\text{ss}} \mid s = R, b, z\right] \\ E_{\sigma\partial W}\left[V_{RP,\text{ss}} \mid s = L, b, z\right] \\ E_{\sigma\partial W}\left[V_{RP,\text{ss}} \mid s = R, b, z\right] \\ Var_{\sigma\partial W}\left[V_{LP,\text{ss}} \mid s = L, b, z\right] \\ Var_{\sigma\partial W}\left[V_{LP,\text{ss}} \mid s = R, b, z\right] \\ Var_{\sigma\partial W}\left[V_{RP,\text{ss}} \mid s = L, b, z\right] \\ Var_{\sigma\partial W}\left[V_{RP,\text{ss}} \mid s = R, b, z\right] \\ \cdots \end{bmatrix} = \begin{bmatrix} p_{P,b} \\ 1 - p_{P,b} \\ 1 - p_{P,b} \\ p_{P,b} \\ p_{P,b}(1 - p_{P,b}) \\ p_{P,b}(1 - p_{P,b}) \\ p_{P,b}(1 - p_{P,b}) \\ p_{P,b}(1 - p_{P,b}) \\ \cdots \end{bmatrix} \tag{11}$$

We can additionally control the variance across samples z of these statistics using second moment constraints in addition to the first moment.

5.2 Goal of DSN modeling

Having access to the DSN trained on the SC model and task responses will allow us to make affirmative statements about the model with the desired task performance. For example, most SC models that produced the appropriate task responses in the manuscript had inhibitory ANTI-PRO connections within hemisphere. A DSN would allow us to assess frequency of these connections being excitatory or inhibitory in the full parameter space. Additionally we could examine differences in the remaining parameters/generated activity when those ANTI-PRO within-hemisphere connections are either excitatory or inhibitory. Similarly, one could examine the probability of E vs I connectivity between ANTI and PRO in opposite hemispheres, or the sign of the Schur-decomposition mode contributions. These are some suggestions of many possible analyses that is enabled by having access to the degenerate parameteric distribution. We should certainly discuss what tests or statements you would like to make moving forward. Additionally, DSNs (optimized to learn maximum entropy distributions) may identify new parameter regimes for solutions that may be scientifically relevant.

6 Github

https://cunningham-lab.github.io/dsn/

References

[1] Gabriel Loaiza-Ganem, Yuanjun Gao, and John P Cunningham. Maximum entropy flow networks. *arXiv preprint arXiv:1701.03504*, 2017.

[2] Sean Bittner and John Cunningham. Learning exponential families. *(In Prep)*, ?(?):?–?, 2019.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[4] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[5] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.