

Degenerate solution networks (DSNs) for theoretical neuroscience

Sean Bittner

November 8, 2018

Theoretical neuroscientists design and test mathematical models of neural activity, assessing a models quality by its ability to replicate experimental results. This is done with rapid turnover, where proposed models are frequently rejected and sometimes promoted when they accurately postdict experimental findings [1]. A challenge in this scientific paradigm is the indeterminacy of model parameterizations; different parameterizations of these models accurately reflect established experiments, but make disparate predictions about uncharacterized properties of neurobiological activity. Oftentimes, the parameter space of these models is examined by extensive simulation. When both data and a reliable posterior inference method are available, probability can be assigned to each parameterization based on data likelihood and some prior assumptions. Even in these cases, it is unclear how the structure of the posterior distribution relies on the various properties of the data. We propose a new machine learning methodology for learning probabilistic models of the full degenerate solution space of generative models given statistical descriptions of the model behavior. The (Aim 1) development of degenerate solution networks (DSNs) will enable (Aim 2) characterization of learning algorithm biases, (Aim 3) behavioral model comparison, and (Aim 4) behavioral model revision.

Aim 1: Develop DSNs and assess scalability on classes of models and behaviors.

DSNs learn the full distribution of model parameterizations that yield a model behavior of interest. Consider model parameterization z and data x generated from some generative model of interest with known sampling procedure, and likelihood $p(x | z)$, which may or may not be known (as is common for models in theoretical neuroscience). DSNs learn a distribution on parameters z , that yields a behavior of interest $\mathcal{B}_i : E_{z \sim q_\theta} [E_{x \sim p(x|z)} [T_i(x)]] = \mu_i$. Since the family of $p(z | \mathcal{B})$ is generally unknown, DSNs make deep generative flexible approximations $q_\theta(z)$ as in [2]. In deep generative models, 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:

$$z = f_\theta(w) \tag{1}$$

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 the full degenerate solution space of a generative model $p(x | z)$ for a given behavior \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. In most settings (especially those relevant to theoretical neuroscience) the likelihood of the behavior with respect to the model parameters $p(T(x) | z)$ is unknown or intractable, requiring an alternative to stochastic gradient variational bayes (SGVB) [3] or black box variational inference (BBVI) [4]. As long as the desired behavior of the model can be statistically characterized as some expectation of sufficient statistics that are differentiable with respect to the model parameters z (and therefore the deep generative parameters θ), we can run the corresponding augmented LaGrangian procedure for the constrained maximum entropy optimization [5]. This is done by optimizing the following objective for a behavior of interest:

$$\begin{aligned} q_\theta^*(z) = \operatorname{argmax}_{q_\theta \in Q} H(q_\theta(z)) \\ \text{s.t. } E_{z \sim q_\theta} [E_{x \sim p(x|z)} [T(x)]] = \mu \end{aligned} \tag{2}$$

A convenient property of maximum entropy optimizations is their exponential family form solutions, which yield a convenient diagnostic for assessing global maximum convergence [6]. The tractability

of this optimization will be assessed using this diagnostic with regard to a mixture of factors: dimensionality of the parameter space, the complexity of the model, and the behavior of interest.

Aim 2: Learn solution spaces of dynamic computations.

Recurrent neural networks (RNNs) are often trained to execute computations in order to perform some experimental task with the intention of comparing the trained system’s activity with that measured in the brain [7]. Statements about the particular algorithm or computational strategy implemented by the RNN are often made by analyzing linearized dynamics around fixed points [8]. There are a variety of methods used to train RNNs [9, 10, 11, 12], and how these learning methods bias the learned connectivities (and potentially the implemented algorithm) within the broader solution space remains poorly understood. Relying on recent work describing the statistical properties of low-rank RNNs [13], we can train DSNs to learn maximally expansive low dimensional distributions of parameterizations of network connectivities that solve tasks such as noisy stimulus detection, context-dependent discrimination, and evidence integration. To check for a particular bias in a training method, we can formulate a structure-indicating test statistic that captures this bias. Then, the distribution of this test statistic over randomness in the training procedure can be compared to its distribution in the full solution space obtained with DSNs.

This will be our first use of DSNs to generate maximally random distributions of test statistics under a model with a particular behavioral specification. One can think of DSNs as a tool to obtain full distributions of summary statistics of models given constraints on the model behavior. Here, we will use DSNs to test whether some learning algorithms bias RNN connectivity within the broader solution space.

Aim 3: Behavioral model comparison.

Bayesian statistical analyses allow us to make inferences about the rules that govern the data we have and will have, which is commonly framed as inductive reasoning. As Gelman and Shalizi remind us [14], model checking procedures comparing test statistics of the data and posterior-predictive distribution facilitate model rejection through hypothetico-deductive reasoning. Akin to the idea of model-checking in Bayesian statistics, with DSNs, we can compare full distributions of a test statistics of models $\mathcal{M}_{p,\mathcal{B}}$ indexed by generative model p and produced behavior \mathcal{B} to the same generative model producing different behaviors (compare $\mathcal{M}_{p,\mathcal{B}_1}$ to $\mathcal{M}_{p,\mathcal{B}_2}$), another model producing the same behavior (compare $\mathcal{M}_{p_1,\mathcal{B}}$ to $\mathcal{M}_{p_2,\mathcal{B}}$), and a different model producing a different behavior (compare $\mathcal{M}_{p_1,\mathcal{B}_1}$ to $\mathcal{M}_{p_2,\mathcal{B}_2}$). Critically, none of the listed types of model comparisons directly relies on collected data as is typical in traditional statistical thinking. While the behaviors we are interested in are likely motivated by data collected through experimentation, the data itself is unnecessary for executing a hypothesis test. To demonstrate how powerful this methodology can be, we will use a single analysis facilitated by DSNs to make an affirmative statement about the relationship of noise robustness and tangling in RNNs, compared to several analyses which made a weaker statement in our previous work [15].

Aim 4: Behavioral model revision.

Access to the full probabilistic degenerate solution space of a theoretical model that yields accurate postdictions can inform model revision in several ways. Say we have behaviors \mathcal{B}_1 and \mathcal{B}_2 which are well established experimentally. We have some model $p(x|z; \phi)$ with parameters z and hyperparameters governing model selection ϕ , which is capable of producing behaviors \mathcal{B}_1 and \mathcal{B}_2 independently. We break down $\phi = [\phi_c, \phi_{nc}]$ into continuous and non-continuous sets of model-selection parameters, respectfully. For example, ϕ_{nc} may govern some choice of model architecture. If we learn a DSN for $\mathcal{M}_{p,\mathcal{B}_1}$ and find there is no support of $q_\theta^*(z)$ that yields \mathcal{B}_2 , we effectively realize that our model is wrong. Our first step for model revision can be to choose some subset of $\phi_{c,opt} \subseteq \phi_c$, and concatenate it to z so that $\tilde{z} = [z, \phi_{c,opt}]$ with $\tilde{\phi} = [\phi_{c,fixed}, \phi_{nc}]$, and learn

a new DSN $\mathcal{M}_{\tilde{p}, \mathcal{B}_1}$ for generative model $\tilde{p}(x | \tilde{z}, \tilde{\phi})$. Then, we can look for $\phi_{c,opt}$ that yield \mathcal{B}_2 indicating the necessary model revision. Alternatively, insights from the probabilistic degenerate solution space can give theorists with specialty knowledge about $p(x|z; \phi)$ information on how to change ϕ in general. We will demonstrate how probabilistic degenerate solution spaces of a V1 circuit model reproducing findings from [16] can be used to prescribe actions for model revision.

References

- [1] Larry F Abbott. Theoretical neuroscience rising. *Neuron*, 60(3):489–495, 2008.
- [2] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.
- [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] Gabriel Loaiza-Ganem, Yuanjun Gao, and John P Cunningham. Maximum entropy flow networks. *arXiv preprint arXiv:1701.03504*, 2017.
- [6] Sean Bittner and John Cunningham. Learning exponential families. (*In review, AI Stats*), ?(?):?–?, 2018.
- [7] David Sussillo. Neural circuits as computational dynamical systems. *Current opinion in neurobiology*, 25:156–163, 2014.
- [8] David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.
- [9] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [10] David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.
- [11] James Martens and Ilya Sutskever. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040. Citeseer, 2011.
- [12] Brian DePasquale, Christopher J Cueva, Kanaka Rajan, LF Abbott, et al. full-force: A target-based method for training recurrent networks. *PloS one*, 13(2):e0191527, 2018.
- [13] Francesca Mastrogiuseppe and Srdjan Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.
- [14] Andrew Gelman and Cosma Rohilla Shalizi. Philosophy and the practice of bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38, 2013.
- [15] Abigail A Russo, Sean R Bittner, Sean M Perkins, Jeffrey S Seely, Brian M London, Antonio H Lara, Andrew Miri, Najja J Marshall, Adam Kohn, Thomas M Jessell, et al. Motor cortex embeds muscle-like commands in an untangled population response. *Neuron*, 97(4):953–966, 2018.

- [16] Mario Dipoppa, Adam Ranson, Michael Krumin, Marius Pachitariu, Matteo Carandini, and Kenneth D Harris. Vision and locomotion shape the interactions between neuron types in mouse visual cortex. *Neuron*, 98(3):602–615, 2018.