Adversarial Training of Neural Encoding Models on Population Spike Trains

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural population responses to sensory stimuli can exhibit both nonlinear stimulus-dependence and richly structured shared variability. Here, we show how adversarial training can be used to optimize neural encoding models to capture both the deterministic and stochastic components of neural population data. To account for the discrete nature of neural spike trains, we use the REBAR method to estimate unbiased gradients for adversarial optimization of neural encoding models. We illustrate our approach on population recordings from primary visual cortex. We show that adding latent noise-sources to a convolutional neural network yields a model which captures both the stimulus-dependence and noise correlations of the population activity.

1 Introduction

Neural population activity contains both nonlinear stimulus-dependence and richly structured neural variability. An important challenge for neural encoding models is to generate spike trains that match the statistics of experimentally measured neural population spike trains. Such synthetic spike trains could be used to explore limitations of a model, or as realistic inputs for simulation or stimulation experiments. Most encoding models either focus on modelling the relationship between stimuli and mean-firing rates e.g. [1]), or on the statistics of correlated variability ('noise correlations') e.g. [2]. They are typically fit with likelihood-based approaches (e.g. maximum likelihood estimation MLE, or variational methods for latent variable models). While this approach is very flexible and powerful, it has mostly been applied to simple models of variability (e.g. Gaussian inputs). Furthermore, MLE-based models are not guaranteed to yield synthetic data that statistically matches empirical data, particularly in the presence of latent variables.
Generative adversarial networks (GANs) [3] are an alternative to fitting the parameters of probabilistic models. In adversarial training, the objective is to find parameters which match the statistics of empirical data, using a pair of competing neural networks – a generator and discriminator. The generator maps the distribution of some input random variable onto the empirical data distribution to try and fool the discriminator. The discriminator attempts to classify input data as samples from the true data distribution or from the generator. This approach has been used extensively to produce realistic images [4] and for text generation [5]. Recently, Molano-Mazon et al. [6] trained a generative model of spike trains, and Arakaki et al. [7], rate models of neural populations, using GANs. However, to the best of our knowledge, adversarial training has not yet been used to train models which produce discrete outputs and aim to capture both the dependence of firing rates on external inputs and shared variability.

We propose to use conditional GANs for training neural encoding models, as an alternative to likelihood-based approaches. A key difficulty in using GANs for neural population data is the discrete nature of neural spike trains: Adversarial training requires calculation of gradients through the

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

generative model, which is not possible for models with a discrete sampling step, and hence, requires the application of gradient estimators. We demonstrate our approach by fitting a convolutional neural network model with shared noise sources to multi-electrode recordings from V1[8].

2 Methods

We want to train a GAN, conditioned on the visual input, to generate multivariate binary spike counts y which match the statistics of empirical data. We model binary spike trains (i.e. each bin t, neuron n and trial i corresponds to an independent draw from a Bernoulli distribution) conditioned on the stimulus x and latent variable z which induces shared variability. We use a convolutional neural network (CNN) f with parameters θ to capture the mapping from x and z to the firing rate λ (Fig. 1), with a sigmoid nonlinearity σ in the last layer,

$$\lambda = \sigma(f(z, x, \theta)) \tag{1}$$

$$y|\lambda \sim \text{Bernoulli}(\lambda). \tag{2}$$

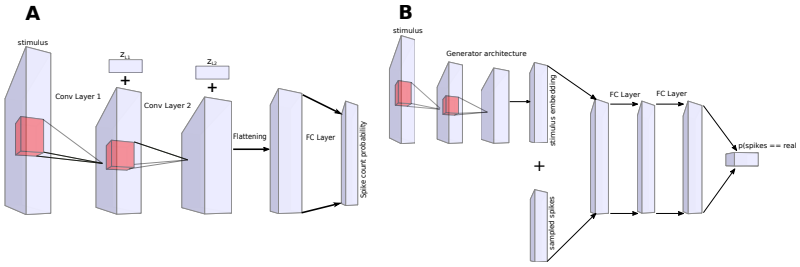

Figure 1: (A) Generator with added noise z_{L1} and z_{L2} (B) Discriminator

The discriminator $\mathcal{D}$ (Fig. 1) receives both the stimulus and the corresponding (simulated or experimental) spike trains. It uses a CNN (similar in architecture to the generator) to embed the stimulus, and combines it with the spike train via fully connected layers. For each timebin and trial, $\mathcal{D}$ outputs the probability of the input spike train being real.

GAN Training The objective is to find the Nash equilibrium of a minmax game between the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$,

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{y \sim p(y)}[\log\ \mathcal{D}(y|x)] + \mathbb{E}_{z \sim q(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)|x)). \tag{3}$$

Due to this objective, GANs are notoriously challenging to train as the training algorithm is sensitive to the gradients with respect to the discriminator parameters. We used the cross-entropy objective as in equation 3, but constrained the discriminator gradients using spectral normalisation [9], and employed gradient norm clipping for the generator gradients.

Dealing with discrete data Obtaining the gradients for the generator requires backpropagation through both generator and discriminator networks. Most applications of GANs have been on continuous data. However, spikes are discrete and thus, the generator has a discrete sampling step which blocks backpropagation. Previous attempts to overcome this problem include using concrete relaxation[10], REINFORCE [11] and REBAR [12] to estimate gradients.
Concrete relaxation approximates the binary variables as continuous values which are close to 0 and 1. This allows backpropagation through the sampling step, but leads to biased gradients. The REINFORCE gradient estimator provides unbiased but high-variance gradients using the log-derivative trick. The REBAR gradient estimator combines concrete relaxation and REINFORCE by using the relaxed outputs as a control variate to the REINFORCE gradient. We found that both REINFORCE and REBAR performed better than concrete relaxation for our GANs. However, we only present results for REBAR here.

Architecture and dataset We fit the GANs to a dataset from Kohn and Smith, consisting of 69 cells recorded from macaque V1, while animals watched a 30s movie repeated 120 times. The movie consisted of 750 frames of size 320 x 320 pixels, which we downsampled to 27 x 27 pixels. We binned the spikes at a 40ms resolution, and binarized the resulting spike trains. Since only 5% of the

spike counts in each bin were non-binary after re-binning, we assumed that the binarization would not significantly alter the results.

For the generator, which receives 10 consecutive movie frames of size 27 x 27 pixels as input, we used a 3-layer CNN architecture similar to Kindel et al. (Fig. 1A, layers 1+2: convolutional with 16 and 32 filters, size 7 by 7, each followed by a MaxPool layer with kernel size 3 and stride 2, followed by LeakyRELUs with slope 0.2.) The final layer of the CNN was a fully connected layer with units equal to the number of neurons in the dataset. To capture the stimulus-independent variability shared between the neurons, we added Gaussian white noise to the units of the convolutional layers. The noise was shared between the units of these layers, multiplied by a separate weight for each unit. The discriminator network consisted of a CNN embedding for the stimulus, similar in structure to the generator, but without the added shared noise (see Fig. 1B), and 5 fully connected ReLU layers.

Training We trained the two networks in parallel for 15k epochs, each consisting of 2 discriminator and 1 generator update. With batch size 50, we used ADAM with learning rate 0.0001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimise the network parameters. The first 650 timebins were used for training the networks and the last 100 timebins for validation. All hyper-parameters were set by hand.

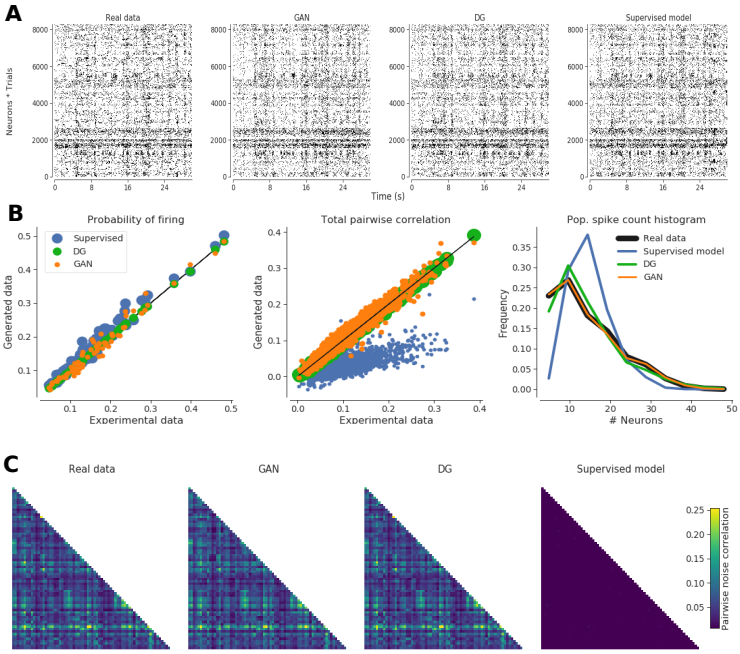

Figure 2: (A) Spike train rasters for data and models. (B) Firing rates (per bin, left) and total pairwise correlation (middle) of supervised (blue), DG (green) and GAN generator(orange) model versus data. Right: population spike count histogram for data (black), supervised (blue), DG (green) and GAN generator(orange) model. (C) Pairwise noise correlation matrix for data and models.

3 Results

We fit a 3-layer CNN generative model of binary spike counts to neural population data recorded in the V1 area of macaque visual cortex [8]. We fit the networks using adversarial training, as described above. For comparison, we fit a CNN with a similar architecture to the GAN generator – but without the shared noise layers – to the same dataset, using supervised learning, i.e. by optimizing the cross-entropy between predicted firing probabilities and experimentally observed spike trains. We also fit a dichotomised Gaussian (DG) model[14], which explicitly represents shared variability via the covariance matrix of a multi-variate Gaussian distribution.

On the training data, all approaches reproduced the gross structure in the spike train rasters (Fig. 2A) and accurately capture the firing rates (here: spike-probabilities per bin, Fig. 2B left). However, the supervised model did not accurately reproduce total pairwise correlations between the neurons (Fig. 2B center), since its noise-correlations 2C) are constrained to be 0. Thus, the histogram of population spike counts for data generated from the supervised model is also substantially different

from that of the real data (Fig. 2B right). The DG model accurately captured the total correlations and the pairwise noise-correlation matrix since it overfits to the PSTH and the SNR of the spike trains. However, it did not accurately capture the population spike histogram, as it only models second order correlation between neurons, while the population spike histogram also depends on higher order correlations. In contrast, the GAN generator, with the addition of just a few shared noise parameters to the supervised model, was able to accurately capture the total correlation, the population spike histogram and pairwise noise-correlation matrix.

4 Discussion

We here showed how adversarial training of conditional generative models that produce discrete outputs (i.e. neural spike trains) can be used to generate data that matches the distribution of spike trains recorded in-vivo, and in particular, its firing rates and correlations. We used unbiased gradient estimators to train conditional GANs on discrete spike trains and spectral normalisation to stabilise training. However, training of discrete GANs remains sensitive to the architecture of the discriminator, as well as hyper-parameter settings.

Adversarial training could be used to capture higher-order structure in neural data, and could be combined with discriminators that target certain statistics of the data that might be of particular interest, in a spirit similar to maximum entropy models [15]. Similarly, this approach could also be extended to capture temporal features in neural population data [16] such as spike-history dependence or adaptation effects. Since we condition the discriminator on the input stimulus, adversarial training could be used for transfer learning across multiple datasets. Generative models trained this way to produce realistic spike trains to various input stimuli, may be used to probe the range of spiking behaviour in a neural population under different kinds of stimulus or noise perturbations.

References

J A Nelder and R W M Wedderburn. Generalized Linear Models. *J. R. Statist. Soc. A.*, 135(3):370–384, 1972. ISSN 0162-1459.

Evan W Archer, Urs Koster, Jonathan W Pillow, and Jakob H Macke. Low-dimensional models of neural population activity in sensory cortical circuits. In *NIPS*, pages 343–351, 2014.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *arXiv*, 9781107015:1–315, 2014. ISSN 01420615.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *ArXiv*, 2016. ISSN 1525-822X.

Manuel Molano-Mazon, Arno Onken, Eugenio Piasini, and Stefano Panzeri. Synthesizing realistic neural population activity patterns using Generative Adversarial Networks. *ICLR*, 2018.

Takafumi Arakaki, G. Barello, and Yashar Ahmadian. Inferring neural circuit structure from datasets of heterogeneous tuning curves. *PLoS Computational Biology*, 15(4), 2019.

A. Kohn and M. A. Smith. *Utah array extracellular recordings of spontaneous and visually evoked activity from anesthetized macaque primary visual cortex (V1)*. CRCNS.org, 2016.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: a continuous relaxation of discrete random variables. *ArXiv*, 2016. ISSN 1528-4336.

Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, (8):229–256, 1992. ISSN 18736513.

George Tucker, Andriy Mnih, Chris J. Maddison, Dieterich Lawson, and Jascha Sohl-Dickstein. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *NIPS*, pages 1–17, 2017.

William F. Kindel, Elijah D. Christensen, and Joel Zylberberg. Using deep learning to reveal the neural code for images in primary visual cortex. *arXiv*, pages 1–9, 2017.

Dmitry R. Lyamzin, Jakob H. Macke, and Nicholas A. Lesica. Modeling Population Spike Trains with Specified Time-Varying Spike Rates, Trial-to-Trial Variability, and Pairwise Signal and Noise Correlations. *Frontiers in Computational Neuroscience*, 4(November):1–11, 2010. ISSN 1662-5188.

Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–12, 2006. ISSN 1476-4687.

Jakob H Macke, L Büsing, John P Cunningham, B M Yu, KV Shenoy, and M Sahani. Empirical models of spiking in neural populations. In *NIPS*, pages 1350–1358, 2011.