Adversarial Training of Neural Encoding Models on Population Spike Trains

Poornima Ramesh, Mohamad Atayi, Jakob H. Macke
Computational Neuroengineering, Technical University of Munich, Germany

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.

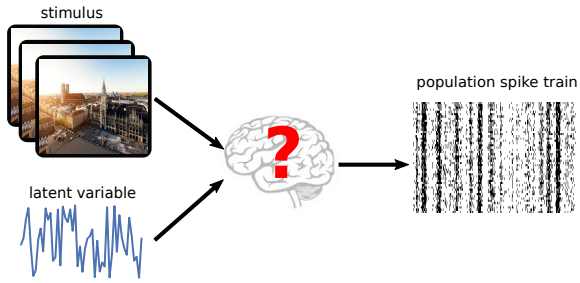
Keywords: population coding; GANs; generative adversarial networks; visual system; data analysis

Introduction

Modern recording methods make it possible to record neural activity from many neurons, revealing 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 focus on approximating the 'signal' (i.e. the mapping of stimuli to average neural responses), neglecting the statistical structure of neural variability. Probabilistic encoding models (Pillow et al., 2008), and in particular latent variable models (e.g. Archer, Koster, Pillow, and Macke (2014)) can, in principle, also capture shared variability. 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 is statistically matched to the empirical data, a problem that can be exacerbated by the approximations necessary for training in the presence of latent variables.

Generative adversarial networks (GANs) (Goodfellow et al., 2014) provide an alternative approach to fitting the parameters of probabilistic models. In adversarial training, the objective is to find parameters which match the statistics of the empirical data, using a pair of competing neural networks – a generator and a discriminator. The generator network maps the distribution of some input random variable onto the empirical data distribution to try and fool the discriminator network. The discriminator network attempts to classify input data as samples from the true data distribution or from the generator. This approach has been used extensively and successfully to produce realistic images (Brock, Donahue, & Simonyan, 2018) and for text generation (Yu, Zhang, Wang, & Yu, 2016).

Recently, Molano-Mazon, Onken, Piasini, and Panzeri (2018) trained a generative model of spike trains, and Arakaki, Barello, and Ahmadian (2019), rate models of neural populations, using GANs. However, to the best of our knowledge, adversarial training has not yet been used to train neural encoding models of population spike trains, i.e. models which produce discrete outputs and aim to capture both the dependence of firing rates on external inputs and shared variability.



Our goal: Neural encoding models that capture both stimulus and response variability in neural population activity, and which yield synthetic data matching the statistics of experimentally observed data.

We propose to use *conditional* GANs (Mirza & Osindero, 2014) 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 generative model, which is not possible for models with a discrete sampling step, and hence, requires the application of gradient estimators. While most studies of discrete GANs use biased gradient estimators based on the concrete relaxation (Maddison, Mnih, & Teh, 2016), we use REBAR (Tucker, Mnih, Maddison, Lawson, & Sohl-Dickstein, 2017) to obtain unbiased gradients for adversarial training of neural encoding models. We demonstrate our approach by fitting a convolutional neural network model with shared noise sources to multi-electrode recordings from V1 (Smith & Kohn, 2008).

Methods

Generative Adversarial Networks 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,

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

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

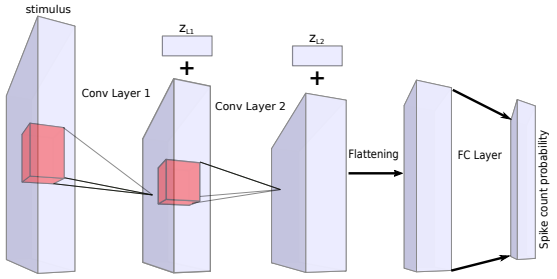


Figure 1: Generator with added noise z_{L1} and z_{L2}

The discriminator $\mathcal{D}$ is a neural network parametrized by ϕ (Fig. 2) which receives both the stimulus and the corresponding (simulated or experimental) spike trains. The discriminator 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 (rather than simulated).

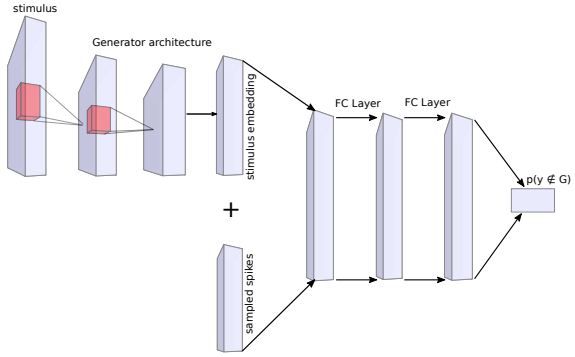


Figure 2: Discriminator

GAN Training The objective is to find the Nash equilibrium of a minimax 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 (Arjovsky & Bottou, 2017). We used the cross-entropy objective as in equation 3, but constrained the discriminator gradients using spectral normalisation (Miyato, Kataoka, Koyama, & Yoshida, 2018), 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. To overcome this problem, we use REBAR (Tucker et al., 2017) to estimate gradients. Compared to gradient estimators previously used to train discrete GANs (Molano-Mazon et al., 2018; Kusner & Hernández-Lobato, 2016), REBAR provides unbiased gradients, and we found that it led to better performance in practice.

The REBAR gradient estimator combines concrete relaxation (Maddison et al., 2016) and the REINFORCE gradient estimator (Williams, 1992). Concrete relaxation approximates the binary variables as continuous values y_{relax} which are close to 0 and 1. This allows backpropagation through the sampling step, but leads to biased gradients. The REINFORCE gradient estimator, on the other hand, provides unbiased but high-variance gradients using the log-derivative trick,

$$\frac{\partial}{\partial\theta}\hat{\mathcal{L}}(\theta,\phi) = \mathbb{E}_{y\sim p_\theta(y)}\left[\mathcal{L}(\mathcal{D}_\phi(y))\frac{\partial}{\partial\theta}\log\, p_\theta(y)\right] \tag{4}$$

REBAR calculates the REINFORCE gradient and uses the relaxed outputs y_{relax} as a control variate (Fig. 3),

$$\frac{\partial}{\partial\theta}\hat{\mathcal{L}}(\theta,\phi) = \mathbb{E}_{y\sim p_\theta(y)}\left[\left(\mathcal{L}(\mathcal{D}_\phi(y) - \eta\mathcal{L}(\mathcal{D}_\phi(y_{\text{relax}}|y))\right)\right.$$
$$\left.\frac{\partial}{\partial\theta}\log\, p_\theta(y)\right] + \eta\frac{\partial}{\partial\theta}\mathbb{E}_{y_{\text{relax}}\sim p_\theta(y_{\text{relax}})}[\mathcal{L}(\mathcal{D}_\phi(y_{\text{relax}}))] \tag{5}$$

Hence, the REBAR gradient is unbiased, has a lower variance compared to REINFORCE, and allows us to estimate gradients despite the discrete sampling step (Fig. 3).

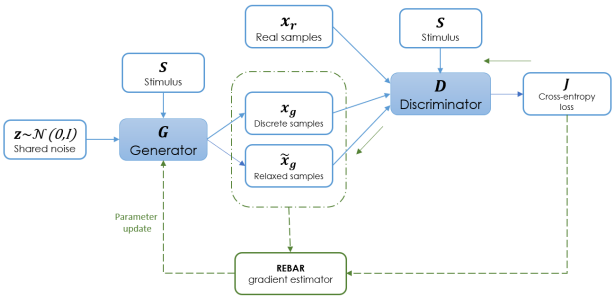


Figure 3: REBAR gradient estimator

Architecture and dataset We fit our models to a dataset from Kohn and Smith (2016), consisting of 69 cells recorded from macaque primary visual cortex, 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.

For the generator, which received 10 consecutive movie frames of size 27 x 27 pixels as input, we used a 3-layer CNN architecture similar to that of Kindel, Christensen, and Zylberberg (2017) (Fig. 1, layers 1+2: convolutional with 16 and 32

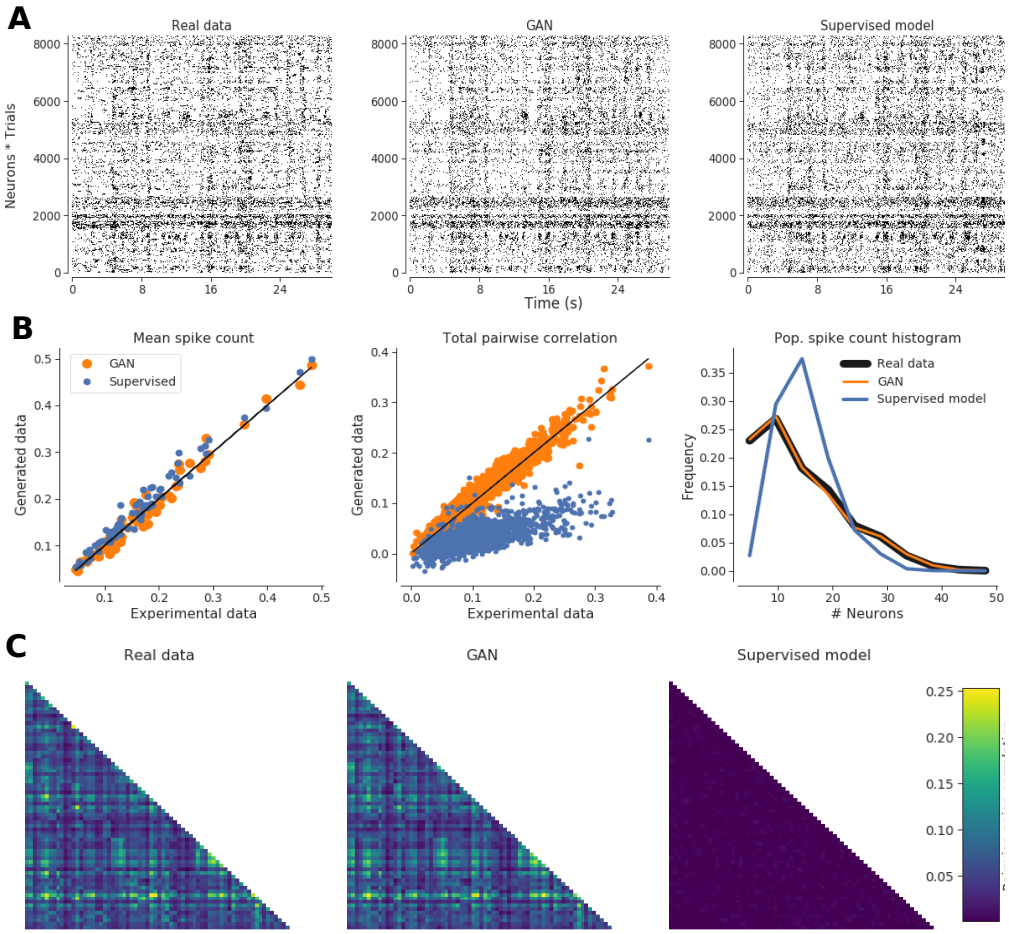


Figure 4: (A) Spike train rasters for the experimental data, GAN generator and supervised model. (B) Firing rates (per bin, left) and total pairwise correlation (middle) of the model trained with supervised learning (blue) and the GAN generator (orange) versus experimental data. Right: population spike count histogram for experimental data (black), supervised model (blue) and the GAN generator (orange). (C) Pairwise noise correlation matrix for data and models.

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. 2), and 5 subsequent fully connected ReLU layers.

Training We trained the two networks in parallel for 15k epochs, each consisting of 2 discriminator updates 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.

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 (Kohn & Smith, 2016). Cells in V1 typically have a nonlinear stimulus selectivity, as well as richly structured shared variability(Cohen & Kohn, 2011; Smith & Kohn, 2008). We fit the networks using adversarial training, as described above. For comparison, we also 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.

On the training data, both approaches were able to reproduce the gross structure in the spike train rasters (Fig. 4A) and accurately capture the firing rates (here: spike-probabilities per bin, Fig. 4B left). However, the supervised model did not accurately reproduce total pairwise correlations between the

neurons, as it has no model of correlated variability (Fig. 4B center). In addition, the histogram of population spike counts for data generated from the supervised model is substantially different from that of the real data (Fig. 4B right). The GAN generator, on the other hand, was able to capture the total correlation between most pairs of neurons in the dataset, with the addition of just a few shared noise parameters. The GAN-model was also able to accurately capture the matrix of pairwise noise-correlations (Fig. 4C). In contrast, as the supervised model has no model of shared variability, its noise-correlations are constrained to be 0.

Discussion

We here showed how adversarial training of simple 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 the REBAR gradient estimator to train conditional GANs on discrete spike trains and used spectral normalisation to stabilise training. We found this approach to be more effective than previous approaches to training discrete GANs. However, training of discrete GANs remains sensitive to the architecture of the discriminator, as well as hyper-parameter settings.

Adversarial training could also 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 (Schneidman, Berry, Segev, & Bialek, 2006). Similarly, this approach could also be extended to capture temporal features in neural population data (Macke et al., 2011) 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, that can 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.

Acknowledgements We thank all members of CNE, especially David Greenberg and Artur Speiser, for feedback and discussions. We thank Matthew Smith and Adam Kohn for sharing data via CRCNS.org. Funding was provided by the DFG through SFB 1233 (276693517).

References

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

Archer, E. W., Koster, U., Pillow, J. W., & Macke, J. H. (2014). Low-dimensional models of neural population activity in sensory cortical circuits. In *NIPS* (pp. 343–351).

Arjovsky, M., & Bottou, L. (2017, Jan). Towards Principled Methods for Training Generative Adversarial Networks. *arXiv e-prints*, arXiv:1701.04862.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Cohen, M. R., & Kohn, A. (2011). Measuring and interpreting neuronal correlations. *Nature neuroscience*, *14*(7), 811.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative Adversarial Nets. *arXiv*, *9781107015*, 1–315.

Kindel, W. F., Christensen, E. D., & Zylberberg, J. (2017). Using deep learning to reveal the neural code for images in primary visual cortex. *arXiv*, 1–9.

Kohn, A., & Smith, M. A. (2016). *Utah array extracellular recordings of spontaneous and visually evoked activity from anesthetized macaque primary visual cortex (v1)*. CRCNS.org. doi: http://dx.doi.org/10.6080/K0NC5Z4X

Kusner, M. J., & Hernández-Lobato, J. M. (2016). GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. *ArXiv*, 1–6.

Macke, J. H., Büsing, L., Cunningham, J. P., Yu, B. M., Shenoy, K., & Sahani, M. (2011). Empirical models of spiking in neural populations. In *Nips* (pp. 1350–1358).

Maddison, C. J., Mnih, A., & Teh, Y. W. (2016). The concrete distribution: a continuous relaxation of discrete random variables. *ArXiv*.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv:1411.1784*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *CoRR*, *abs/1802.05957*.

Molano-Mazon, M., Onken, A., Piasini, E., & Panzeri, S. (2018). Synthesizing realistic neural population activity patterns using Generative Adversarial Networks. *ICLR*.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., & Simoncelli, E. P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, *454*(7207), 995-9.

Schneidman, E., Berry, M. J., Segev, R., & Bialek, W. (2006). Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, *440*(7087), 1007–12.

Smith, M. A., & Kohn, A. (2008). Spatial and temporal scales of neuronal correlation in primary visual cortex. *Journal of Neuroscience*, *28*(48), 12591–12603.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, D., & Sohl-Dickstein, J. (2017). REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *NIPS* (pp. 1–17).

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

Yu, L., Zhang, W., Wang, J., & Yu, Y. (2016). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *ArXiv*.