MULTILAYER RECURRENT NETWORK MODELS OF PRIMATE RETINAL GANGLION CELL RESPONSES

Eleanor Batty, Josh Merel *
Doctoral Program in Neurobiology & Behavior, Columbia University
erb2180@columbia.edu, jsmerel@gmail.com

Nora Brackbill *
Department of Physics, Stanford University
nbrack@stanford.edu

Alexander Heitman
Neurosciences Graduate Program, University of California, San Diego
alexkenheitman@gmail.com

Alexander Sher & Alan Litke
Santa Cruz Institute for Particle Physics, University of California, Santa Cruz
sashake3@ucsc.edu, Alan.Litke@cern.ch

E.J. Chichilnisky
Department of Neurosurgery and Hansen Experimental Physics Laboratory, Stanford University
ej@stanford.edu

Liam Paninski
Departments of Statistics and Neuroscience, Columbia University
liam@stat.columbia.edu

ABSTRACT

Developing accurate predictive models of sensory neurons is vital to understanding sensory processing and brain computations. The current standard approach to modeling neurons is to start with simple models and to incrementally add interpretable features. An alternative approach is to start with a more complex model that captures responses accurately, and then probe the fitted model structure to understand the neural computations. Here, we show that a multitask recurrent neural network (RNN) framework provides the flexibility necessary to model complex computations of neurons that cannot be captured by previous methods. Specifically, multilayer recurrent neural networks that share features across neurons outperform generalized linear models (GLMs) in predicting the spiking responses of parasol ganglion cells in the primate retina to natural images. The networks achieve good predictive performance given surprisingly small amounts of experimental training data. Additionally, we present a novel GLM-RNN hybrid model with separate spatial and temporal processing components which provides insights into the aspects of retinal processing better captured by the recurrent neural networks.

1 INTRODUCTION

Our understanding of sensory processing in the brain is most straightforwardly reflected in our ability to model the process by which stimuli presented at the sensory periphery are transformed into the spiking activity of populations of neurons. For decades, researchers have interrogated stimulus-response

*These authors contributed equally.

neural properties using simplified targeted stimuli, such as bars, spots, or gratings. While these types of stimuli uncovered many interesting aspects of visual computation, they have several limitations (Barlow & Levick, 1965). These stimuli may not fully drive important components of neural response, and modeling efforts have often assumed a quasi-linear mapping from stimulus to firing rate. Subsequent efforts to characterize cells relied on white noise stimulation and building models through reverse correlation (de Boer R & Kuyper, 1968; Marmarelis & Naka, 1972; Chichilnisky, 2001). A standard model used to relate white noise to spiking responses is the linear-nonlinear-Poisson (LN) or generalized linear model (GLM) which consists of a spatiotemporal linear filtering of the stimulus followed by a nonlinearity and probabilistic spike generation (Chichilnisky, 2001; Simoncelli et al., 2004; Schwartz et al., 2006). Although this family of models have advanced our understanding, they do not optimally capture neural responses, especially to natural scenes which can lead to more complex responses than white noise stimuli (David et al., 2004). Even in the retina, early in the visual processing stream, these commonly-used models capture retinal ganglion cell (RGC) responses to natural stimuli less accurately than to white noise (Heitman et al., 2016).

Recently, deep neural networks have been used to dramatically improve performance on a diverse array of machine learning tasks (Krizhevsky et al., 2012; LeCun et al., 2015). Furthermore, these networks bear a loose resemblance to real neural networks, and provide a sufficiently rich model class that can still be roughly constrained to match the biological architecture (Kriegeskorte, 2015). Most previous research at this intersection of neuroscience and artificial neural networks has focused on training networks on a certain task, such as object recognition, and then comparing the computations performed in different layers of the artificial network to those performed by real neurons (Yamins et al., 2014). Here we take a different approach: we fit multilayer models directly to the spiking responses of neurons, an approach that has not been explored in detail (but see (McIntosh et al., 2016) for some recent independent parallel developments).

2 Approach

We fit a range of models, detailed below, to spiking responses of primate RGCs. Our baseline comparisons are the GLM architectures that have been widely used to construct previous neural models (Pillow et al., 2008), though here we focus on individual neuronal responses (we leave modeling of correlations between neurons for future work). We focused on RNNs as a flexible framework in which to model more complex temporal and spatial nonlinearities. We also explored a number of network architectures involving features or weights shared across observed neurons. Given the complexity of the network architectures, we reasoned that sharing statistical strength across neurons by learning a shared feature space might improve predictive performance. This is conceptually a form of *multitask* learning - we are using a shared representation to achieve better generalization (Baxter, 2000). Motivated by previous research showing significant differences in the processing properties of the two cell types examined, ON and OFF parasol retinal ganglion cells, we fit separate models for each of these cell types (Chichilnisky & Kalmar, 2002).

3 Methods

3.1 Data collection

We fit spiking responses of OFF and ON parasol retinal ganglion cells to natural scenes. Recordings were performed on isolated retina using a large-scale multi-electrode recording system (Litke et al., 2004; Frechette et al., 2005; Field et al., 2007). A standard spike sorting algorithm was used to identify spikes from different cells from the voltage signals on each electrode during visual stimulation (Litke et al., 2004). We focus on two separate experiments (the same experimental procedure in two separate retinas) here; analyses of other datasets yielded similar results. Models were fit separately for the two experiments due to animal to animal variability in cell properties, such as receptive field size and firing rate. Almost all spike sorted cells were used for training (exp 1 = 118 OFF cells, 66 ON cells; exp 2 = 142 OFF cells, 103 ON cells): two cells were removed due to data quality issues (see sec 3.3). Performance metrics in this paper are reported for the same subset of cells used in a previous study (Heitman et al., 2016). These cells passed passed a manual screen for spike sorting accuracy, demonstrated stable light responses, and met a convergence criteria in prior linear-nonlinear modeling (exp 1 = 10 OFF cells, 18 ON cells; exp 2 = 65 OFF cells, 14 ON cells). The naturalistic movie

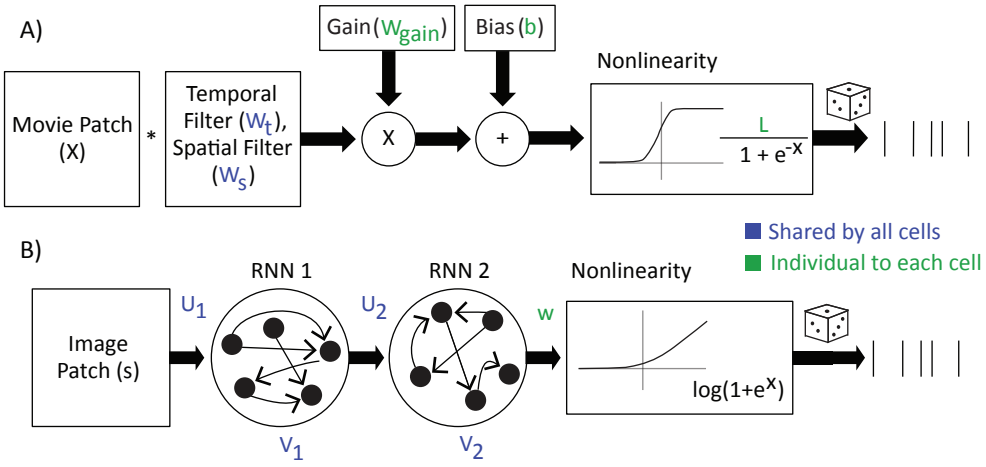

Figure 1: Example model architectures. (A) Shared LN model. The past few frames of the stimulus images are presented as inputs which are spatiotemporally filtered and passed through a nonlinearity to produce a firing rate, which drives a Poisson spiking process. (B) Two-layer RNN. The current frame of the stimulus feeds into a sequence of RNN layers (history dependence is implicit in the hidden unit activations) and a Poisson GLM draws weighted inputs from the activations of the hidden units of the last RNN layer and outputs predicted spike trains. Thus the last RNN layer represents a shared feature pool that all the RGCs can draw from.

stimulus consisted of images from the Van Hateren database shown for one second each, with spatial jitter based on eye movements during fixation by awake macaque monkeys (Z.M. Hafed and R.J. Krauzlis, personal communication), (van Hateren & van der Schaaf, 1998). An example stimulus can be found at `https://youtu.be/sG_18Uz_6OE`. 59 distinct natural scenes movies of length one minute (the training data) were interleaved with 59 repetitions of a 30 second movie (the test data). Interleaving ensured that the test movie repetitions spanned the same period of time as the training data and therefore experienced the same range of experimental conditions (in case of neural response drifts over time). The first 4 movies shown (2 training movies and 2 repetitions of the test movie) were excluded to avoid initial transients. Test metrics are reported for the last 29 seconds of the 30 second test movie for the same reason. For further details on the experimental set-up, data preprocessing, and visual stimulation, see Heitman et al. (2016).

3.2 Model Training

All models were implemented in Theano and trained on a combination of CPUs and GPUs (Theano Development Team, 2016). Training was performed using the Adam optimizer on the mean squared error (MSE) between predicted firing rate and true spikes (Kingma & Ba, 2014). We also experimented with optimizing a Poisson likelihood; this led to qualitatively similar results but occasionally less stable fits, so we focus on the MSE results here. All recurrent dynamics and temporal filters operated on time bins of 8.33 ms (the frame rate of the movie). Spike history terms and performance metrics were calculated for 0.833 ms bins. We used the same split of training and validation data for both experiments: 104 thirty-second movies as training data and 10 thirty-second movies as a held-out validation set.

During training, the performance on the held-out validation set is checked after every pass through the training data. After each iteration through the training data, if the model exhibits significantly better validation performance than our previous best, we reset the minimum number of iterations to be twice the current iteration number. If we make it through those iterations without another significant improvement, we stop. We train for a maximum of 150 epochs, where we define one epoch as one pass through all the training data. The model with the best validation performance is saved and used to assess test performance. All models with shared parameters were trained on a combined MSE over

all neurons and the parameters picked were those which minimized validation MSE for all neurons. For individual LNs/GLMs/RNNs, the validation MSE was minimized for each neuron separately.

3.3 RECEPTIVE FIELD CENTER ESTIMATION

In all models used in this paper, we estimate the receptive field (RF) center of each neuron in order to identify the appropriate portion of the image to use as input. We calculate a 250 ms long spike triggered average (STA) using reverse correlation of the neuron's spikes with a white noise stimulus. We reduce the noise in this STA by using a rank 1 approximation (singular value decomposition followed by reconstruction using the primary temporal and spatial components). We then smooth each frame of the STA via convolution with a Gaussian spatial filter. The center location is defined as the pixel location that has the maximum absolute magnitude over time. The center locations were visually assessed to check accuracy of the algorithm. Rare cases where the algorithm failed to identify the correct center indicated neurons that responded to very little of the image as their receptive field was more than half-way displaced out of the image. These two neurons (two Exp 1 ON cells) were removed from further analysis. If the receptive field center is close to the edge of the image, the image patch is padded with the average training stimulus value.

3.4 PERFORMANCE EVALUATION

To quantitatively evaluate the accuracy of model spike predictions, we used the fraction of explainable variance, which has been described in previous literature (Heitman et al., 2016). Average firing rates over time are obtained after generating spikes from the model in 0.833 ms bins and smoothing with a Gaussian temporal filter (SD=10ms). The fraction of variance is computed as

$$F(r, r_s) = 1 - \frac{\sum_t (r(t) - r_s(t))^2}{\sum_t (r(t) - \mu)^2} \tag{1}$$

where $r(t)$ is the smoothed recorded firing rate, $r_s(t)$ is the smoothed predicted firing rate, and μ is the average recorded rate. Finally, to account for the reproducibility of responses over repeated trials, we normalize by the fraction of variance captured by using the average firing rate on the odd (r_o) trials of the repeated test movie to predict responses on the even (r_e) trials:

$$FV = \frac{F(r, r_s)}{F(r_e, r_o)}. \tag{2}$$

4 MODEL ANALYSIS

4.1 NETWORK ARCHITECTURES

Individual LNs and GLMs: The linear-nonlinear model (LN) consists of a spatiotemporal filtering of the 31x31x30 movie patch (X_t, width by height by time) surrounding the estimated center of the neuron's receptive field plus a bias term (b), followed by a sigmoid nonlinearity (f), and Poisson spike generation to produce the responses r_t. The generalized linear model (GLM), given by

$$r_t \sim Poiss\left[f\left(\vec{w}_s^T (X_t \vec{w}_t) + b + \sum_i h_i r_{t-i} \right) \right], \tag{3}$$

has the same architecture with the addition of a post-spike history filter h before the nonlinearity f (Pillow et al., 2008). We used a rank 1 approximation of the full spatiotemporal filter (higher rank models did not significantly improve fits on a subset of examined neurons), resulting in a vectorized 31x31 spatial filter ($\vec{w}_s$) and a 30 bin temporal filter ($\vec{w}_t$) which spans 250 ms (Heitman et al., 2016). The post-spike history filter consists of a weighted sum of a basis of 20 raised cosines spanning approximately 100 ms (Pillow et al., 2008). The models with spike history were fit by initializing with the model fit without spike history. The filter either operates on the recorded spikes (training and validation) or the spikes generated by the model (testing). The nonlinearity is the logistic sigmoid: $f = L/(1 + \exp(-x))$, which has been shown to improve fitting over an exponential nonlinearity for modeling RGC responses to natural scenes (Heitman et al., 2016).

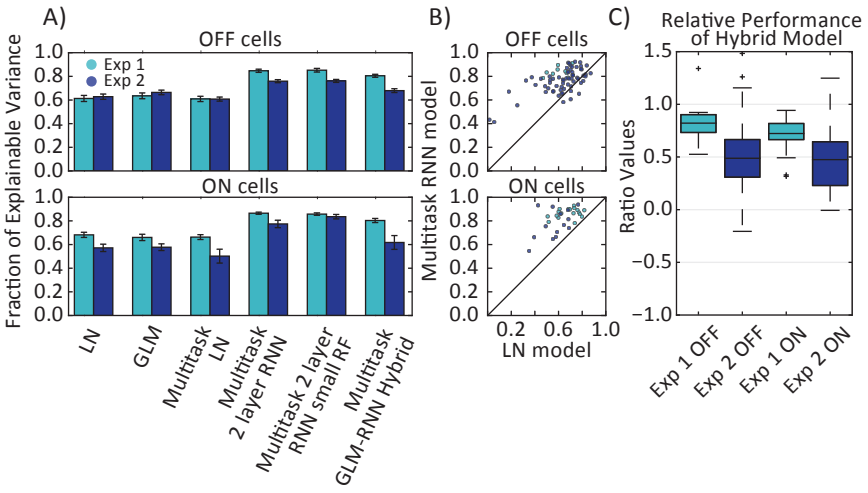

Figure 2: Model performance. (A) Mean $\pm$ std. err. of the fraction of explainable variance for criteria-passing subset of OFF and ON cells for various model architectures. (B) Scatter plots show individual neural performance from LN and RNN model; each dot corresponds to one cell. Negative FV values are shown on relevant axis as FV=0 (C) Hybrid model performance, quantified by the ratio between the multitask LN to multitask hybrid performance gap and the multitask LN to multitask RNN performance gap (one high outlier not pictured for both Exp 1 ON and Exp 2 ON)

Shared LN: In this model, the architecture is similar to the individual LNs but all cells of a given type (OFF or ON) share the same temporal and spatial filters (Figure 1A; note that the spatial filters are displaced to the RF center of each individual RGC). All other parameters are individually tuned for each observed neuron. There is an additional gain term that weights the output of the filtering individually for each observed neuron.

Two-layer RNN, 50 units: In this architecture, there are two recurrent neural network (RNN) layers between the image patch and Poisson neural unit:

$$\vec{h}^{(1)}_{j,t} = \max(0, U_1 \vec{s}_{j,t} + V_1 \vec{h}^{(1)}_{j,t-1} + \vec{c}) \tag{4}$$

$$\vec{h}^{(2)}_{j,t} = \max(0, U_2 \vec{h}^{(1)}_{j,t} + V_2 \vec{h}^{(2)}_{j,t-1} + \vec{d}) \tag{5}$$

$$r_{j,t} \sim Poiss\left[f(\vec{w}_j^T \vec{h}^{(2)}_{j,t} + b_j)\right]. \tag{6}$$

The activity of the 50 units in the first RNN layer at time t is given by $\vec{h}^{(1)}_{j,t}$ in Eqn. 4. These units are rectified linear, and receive input from the vectorized 31x31 image patch surrounding the center of neuron j's receptive field, $\vec{s}_{j,t}$, with weights U_1, along with input from the other units in the layer with weights V_1 and a bias $\vec{c}$. The output of the first RNN is then fed into a second RNN with similar architecture. The firing rate for each observed neuron in the final layer is then given by Eqn. 6, and is a weighted sum of the recurrent units plus a bias b_j, followed by a softplus nonlinearity $f = \log(1 + \exp(-x))$. Note that all parameters are shared across neurons except for the weights to the final layer and the final bias terms ($\vec{w}_j$ and b_j).

GLM-RNN Hybrid: The GLM-RNN hybrid model consists of a spatial filter followed by a two-layer RNN. The architecture resembles that of the full two-layer RNN with 50 units, except the input to the first layer is a scalar (post multiplication with the spatial filter) at each time step instead of the full image patch; thus the RNN in this model is responsible for shaping the temporal properties of the output, but does not affect spatial processing after the first linear spatial filtering stage. All weights

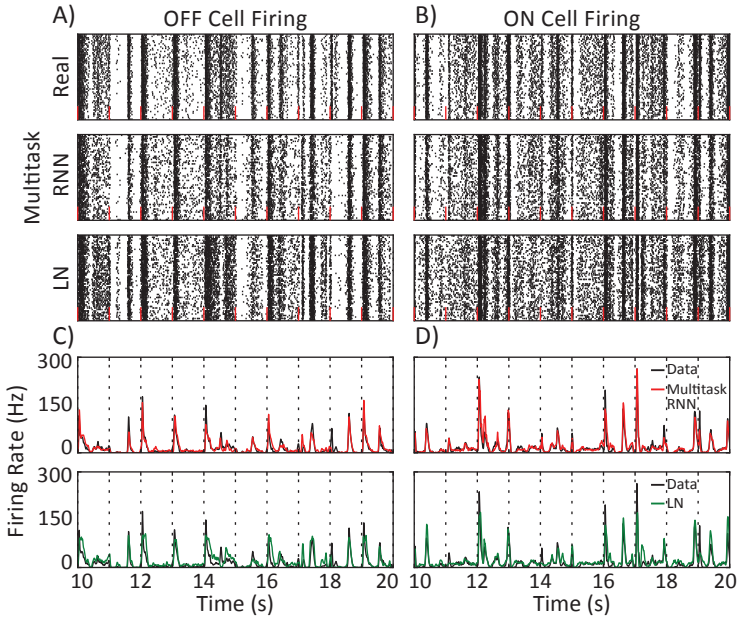

Figure 3: (A,B) Rasters showing spiking responses for 57 trials (each row corresponds to a single trial) for an OFF and ON cell from experiment 1 for 10 seconds of a novel natural scenes movie. Example cells chosen had near average difference between LN and RNN performance. Red ticks denote time at which one natural image was replaced by another. (C,D) Average predicted spikes over trials smoothed with Gaussian (SD = 10 ms) for same 10 seconds of the novel natural scenes movie show qualitative differences among models. Dotted vertical lines align with red ticks in (A,B).

are shared across neurons except for weights to the final layer ($\vec{w}_j$) and the final bias terms (b_j):

$$y_{j,t} = \vec{w}_s^T \vec{s}_{j,t} \tag{7}$$

$$\vec{h}_{j,t}^{(1)} = max(0, \vec{w_1} y_{j,t} + V_1 \vec{h}_{j,t-1}^{(1)} + \vec{c}) \tag{8}$$

$$\vec{h}_{j,t}^{(2)} = max(0, U_2 h_{j,t}^{(1)} + V_2 \vec{h}_{j,t-1}^{(2)} + \vec{d}) \tag{9}$$

$$r_{j,t} \sim Poiss\left[f(\vec{w}_j^T \vec{h}_t^{(2)} + b_j)\right]. \tag{10}$$

4.2 Model performance

RNNs of varying architectures consistently outperformed LNs and GLMs in predicting neural spiking responses to a novel natural scene movie for both OFF and ON parasol retinal ganglion cells in both experiments (Figure 2). A shared two-layer recurrent network consistently captures around 80% of the explainable variance across experiments and cell types. Other recurrent architectures (1-3 layer RNNs and a 2 layer LSTM) led to similar levels of performance (Supplementary Figure 6). The increase in performance according to the fraction of explainable variance metric was not an average effect: almost all neurons were significantly better predicted by the RNN (Figure 2B). A 2 layer RNN model with additional trained spike history filters outperformed GLMs and LNs according to a normalized log likelihood metric (Supplementary Figure 7).

Inspection of the mean predicted firing rate traces for LNs and RNNs in Figure 3 reveals that the recurrent network seems to be capturing the timing of firing more precisely. The LN often predicts a general increase in firing rate at the correct times, but the RNN captures the sudden increase in firing rate followed by decay which often occurs when the image changes. On the other hand, the LN models sometimes predict modest increases or decreases in firing rate that the recurrent nets miss.

Understanding why the recurrent models improve performance is a challenging task due to the black-box nature of deep networks. The first layer filters (U_1, from image patches to recurrent units) have an interpretable structure resembling traditional receptive fields expected in the retina

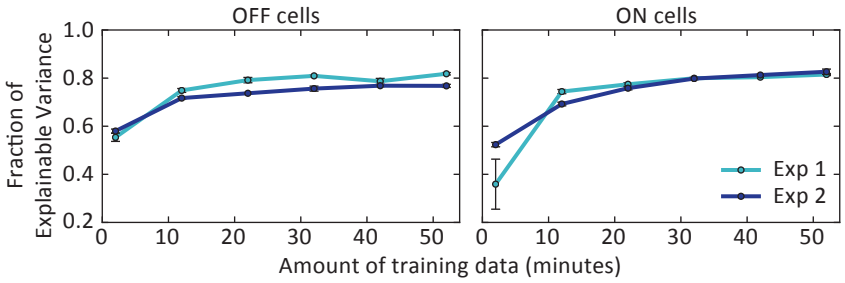

Figure 4: Model predictive performance on held-out data as a function of the amount of training data. Error bars show SEM over 3 iterations of the mean FV over all neurons

(Supplementary Figure 8). However, the computations performed by the recurrent units are difficult to tease apart, because the weights are less interpretable. Thus, instead of attempting a mechanistic explanation of the internals of the RNN, we focused on what additional captured information resulted in the improved RNN performance.

One possibility is that capturing nonlinear effects in parts of the image far from the receptive field center improved predictions (McIlwain, 1964; Passaglia et al., 2009). We restricted the size of the image patch surrounding each receptive field center from 31x31 to 15x15 (Supplementary Figure 9). Shared RNNs trained on the smaller image patch size did as well, or better, than those trained on the larger patch across almost all combinations of cell type and experiment. (We see a similar small improvement when training the LN models on the small patch.) Thus we concluded that long-range nonlinear spatial interactions do not contribute to the increased performance produced by the RNNs.

We also investigated whether nonlinear spatial interactions or nonlinear temporal processing primarily contributed to better predictions. To accomplish this, we constructed a GLM-RNN hybrid, described previously, in which a single spatial filter precedes a two-layer RNN - effectively allowing only temporal nonlinearities to be captured. This model improved prediction over the LNs and GLMs but did not reach full RNN performance. The amount by which this model closed the gap differed for different experiments and cell types. We quantified this by computing the difference between multitask RNN and multitask LN performance for each neuron and the difference between multitask hybrid and multitask LN performance. We divide the latter by the former (on a cell-by-cell basis) to obtain the ratios summarized in Figure 2C. The hybrid model closed greater than half of the gap on average between multitask LN and RNN performance, indicating that the richer temporal dynamics of the RNN model account for a large part of the difference between RNN and LN performance, though spatial nonlinearities play a role too.

5 Modest training data length suffices for good performance

Deep networks can be complex and often require large amounts of data to adequately train: convolutional neural networks used for object recognition are trained on over a million images (Krizhevsky et al., 2012). Standard neuroscience experiments yield limited data sets, so it is crucial to assess whether we have enough data to adequately fit our network architectures. We trained the RNN on varying amounts of data, and ran several different iterations of the network to explore variation over random initializations and randomly chosen training sets. These results are shown for both ON and OFF cells in Figure 4. Surprisingly small amounts of training data resulted in good predictive abilities. For larger amounts of training data, different iterations resulted in very similar mean fraction of variance values, indicating fairly robust fitting in these models. See Supplementary Figure 10 for further details.

6 Benefits of multitask framework

We investigated whether the multitask framework with shared parameters across neurons actually helps to improve predictive performance with reasonable amounts of experimental data. First, we quantified the benefits of parameter-sharing in the simple LN model. This is a highly constrained

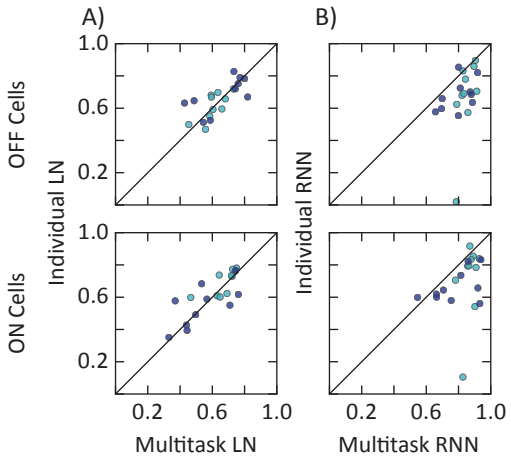

Figure 5: Shared vs individual fits for LN model (A) and RNN model (B). 10 OFF and 10 ON cells from each experiment are pictured (Light blue = exp 1, dark blue = exp 2). Negative FV values are pictured as FV = 0.

framework: every cell has the same spatial and temporal filter. The shared LN does not improve performance for most neurons (Figure 5A).

We expected the multitask framework to be more helpful applied to the RNN model because in this case we are sharing features but not all parameters across neurons. Indeed, the multitask RNN consistently outperformed RNNs trained individually on single neurons (Figure 5B); individually-trained RNNs also had much more variable losses than did the multitask-trained RNNs. In a realistic experimental setting with limited data, the multitask framework is a useful way to leverage all of the data collected for all neurons.

7 Conclusion

Using retinal neurons responding to natural scenes as an example, we showed that: using deep networks to model neural spiking responses can significantly improve prediction over current state-of-the-art models; sharing information across neurons in a multi-task framework leads to better and more stable predictions; and these models work well even given relatively small amounts of experimental data. We believe that the multitask RNN framework presented here will enable new, richer models of complex nonlinear spiking computations in other brain areas.

While one could argue that we have merely exchanged the black box of the brain for another black box, just having a more predictive model is an important tool for research: these predictive models of the primate retina can be used in retinal prosthetics research, to probe decoding, and as a first stage of processing in the modeling of higher visual areas. Additionally, the recurrent network is more accessible and available for experimentation and quantitative analysis. For example, the trained neural network models may guide choices for more accurate simpler models by identifying key computational features that are important to include. Training smaller models on the denoised compression of spiking data (the predicted firing rate) may help them to learn features they otherwise would not (Ba & Caruana, 2014). The deep network approach allows one to determine types of information important to the neuron without having to build an exact mechanistic model of how such information is incorporated, as demonstrated by our finding that both spatial and temporal nonlinearities are not fully captured by the standard pseudo-linear models. We hope in future work to gain a more thorough and quantitative understanding of the dynamics captured by the recurrent networks and to extend this approach to higher sensory areas.

ACKNOWLEDGMENTS

Funding for this research was provided by the National Science Foundation Graduate Research Fellowship Program under grant No. DGE-114747 (NB), Grant Number No. DGE-16-44869 (EB), the National Science Foundation IGERT Training Grant No. 0801700 (NB), the National Institutes of Health Grant EY017992 (EJC), NSF CRCNS IIS-1430239 (LP, EJC) and Google Faculty Research awards (LP, EJC); in addition, this work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DoI/IBC) contract number D16PC00003 (LP). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

REFERENCES

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in Neural Information Processing Systems*, 2014.

HB Barlow and William R Levick. The mechanism of directionally selective units in rabbit's retina. *The Journal of Physiology*, 178(3):477, 1965.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, 2000.

E. J. Chichilnisky and Rachel S. Kalmar. Functional asymmetries in on and off ganglion cells of primate retina. *The Journal of Neuroscience*, 22(7):2737–2747, 2002.

E.J. Chichilnisky. A simple white noise analysis of neuronal light responses. *Network: Computation in Neural Systems*, 12(2):199–213, 2001.

Stephen V. David, William E. Vinje, and Jack L. Gallant. Natural stimulus statistics alter the receptive field structure of v1 neurons. *The Journal of Neuroscience*, 24(31):6991–7006, 2004.

de Boer R and P Kuyper. Triggered correlation. *IEEE Transactions on Biomedical Engineering*, 15: 169–179, 1968.

Greg D. Field, Alexander Sher, Jeffrey L. Gauthier, Martin Greschner, Jonathon Shlens, Alan M. Litke, and E. J. Chichilnisky. Spatial properties and functional organization of small bistratified ganglion cells in primate retina. *The Journal of Neuroscience*, 27(48):13261–13272, 2007.

E. S. Frechette, A. Sher, M. I. Grivich, D. Petrusca, A. M. Litke, and E. J. Chichilnisky. Fidelity of the ensemble code for visual motion in primate retina. *Journal of Neurophysiology*, 94(1):119–135, 2005.

Alexander Heitman, Nora Brackbill, Martin Greschner, Alexander Sher, Alan M. Litke, and E.J. Chichilnisky. Testing pseudo-linear models of responses to natural scenes in primate retina. *bioRxiv*, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:11412.6980, 2014.

Nikolaus Kriegeskorte. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 1:417–446, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

A. M. Litke, N. Bezayiff, E. J. Chichilnisky, W. Cunningham, W. Dabrowski, A. A. Grillo, M. Grivich, P. Grybos, P. Hottowy, S. Kachiguine, R. S. Kalmar, K. Mathieson, D. Petrusca, M. Rahman, and A. Sher. What does the eye tell the brain?: Development of a system for the large-scale recording of retinal output activity. *IEEE Transactions on Nuclear Science*, 51(4):1434–1440, 2004.

P.Z. Marmarelis and K. Naka. White-noise analysis of a neuron chain: an application of the wiener theory. *Science*, 175:1276–1278, 1972.

James T. McIlwain. Receptive fields of optic tract axons and lateral geniculate cells: peripheral extent and barbiturate sensitivity. *Journal of Neurophysiology*, 27(6):1154–1173, 1964.

Lane McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen Baccus. Deep convolutional neural network models of the retinal response to natural scenes. In *Cosyne Abstracts*, Salt Lake City USA, 2016.

CL. Passaglia, Freeman DK., and Troy JB. Effects of remote stimulation of the modulated activity of cat retina. *Journal of Neuroscience*, 29, 2009.

Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, E.J. Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

Odelia Schwartz, Jonathan W. Pillow, Nicole C. Rust, and Eero P. Simoncelli. Spike-triggered neural characterization. *Journal of Vision*, 6(4):13, 2006.

Eero Simoncelli, Jonathan W. Pillow, Liam Paninski, and Odelia Schwartz. *Characterization of neural responses with stochastic stimuli*, pp. 327–338. MIT Press, 2004.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings. Biological sciences / The Royal Society*, 265(1394):359–366, March 1998.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

8 SUPPLEMENTARY FIGURES

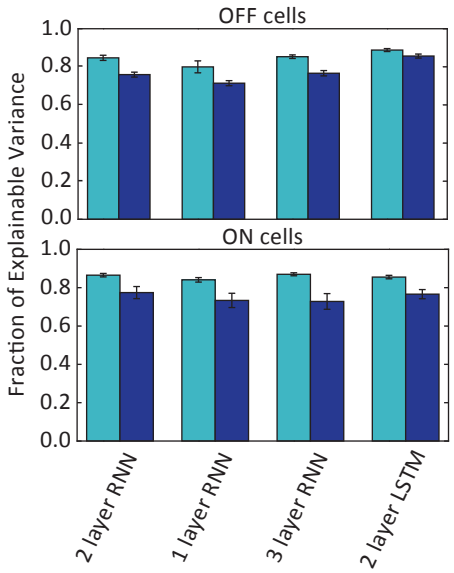

Figure 6: Multiple different types of RNN architectures lead to similar levels of performance (Light blue = exp 1, dark blue = exp 2). We compare (from left to right): 1) 2 layer RNN with 50 units/layer, 2) 1 layer RNN with 100 units, 3) 3 layer RNN with 33 units/layer, 4) LSTM architecture as detailed in Hochreiter & Schmidhuber (1997)

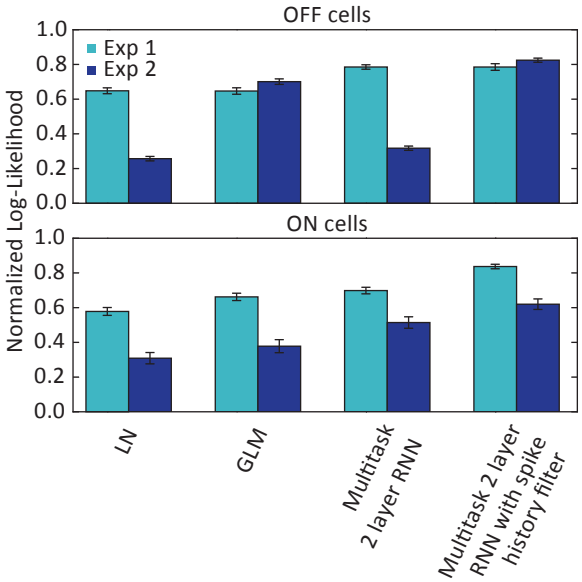

Figure 7: RNNs with added spike history filters outperform GLMs and LNs according to a normalized log-likelihood metric. RNNs without spike history filters underperformed GLMs in some cases: log-likelihood metrics are calculated using spike history filters generated by actual spikes so these filters can improve log-likelihood without improving the fraction of explainable variance. The parameters of a multitask 2 layer RNN trained with a sigmoid nonlinearity were held fixed while the parameters of the last layer, including spike history filters, were trained. The normalized LL term was computed as detailed in Heitman et al. (2016) except the ideal model was trained using MSE.

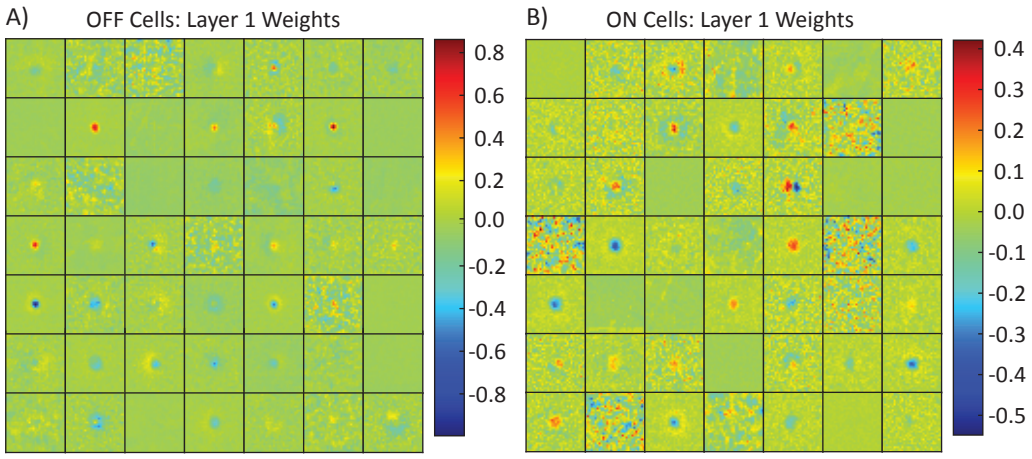

Figure 8: Filters from image to Layer 1 RNN units in 2 layer RNN. Interpretable structures, including OFF and ON centers and surrounds, are visible

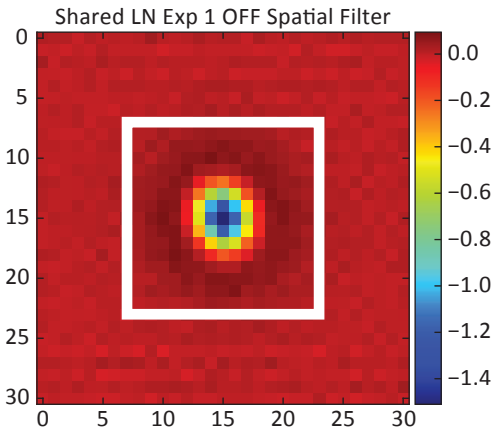

Figure 9: The 31x31 shared LN spatial filter is pictured. The white rectangle indicates the boundary of the smaller 15x15 patch

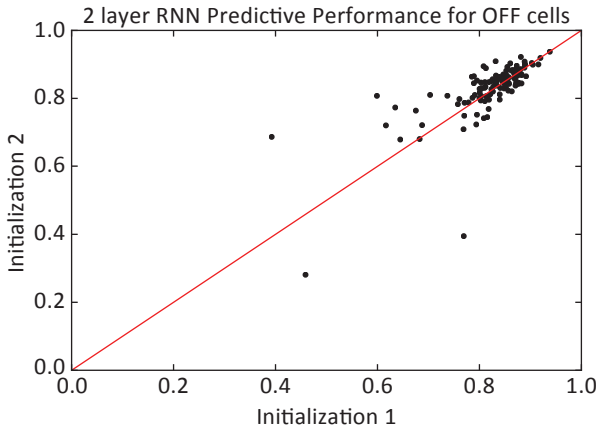

Figure 10: Comparison of two different initializations of a 2 layer RNN trained on Exp 1 OFF cells. Performance differs slightly for individual neurons but the resulting average FV values are very similar (0.819 vs 0.828)