Models of Neuron Coding in Retinal Ganglion Cells and Clustering by Receptive Field

Kevin FEIGELIS (feigelis@stanford.edu), Claire HEBERT (chebert@stanford.edu), Brett LARSEN (bwlarsen@stanford.edu)

BACKGROUND

One of the fundamental questions in neuroscience today is understanding the functional relationship between stimuli and the firing of neurons, often referred to as the neuron coding problem. [1] We will specifically consider how machine learning can help us learn models of the retina and its response to visual stimuli. [2]

As seen in Figure 1, the retina in general consists of three layers: photoreceptors (P) which transduce photons into biological signals; an intermediate network of neurons consisting of what are called bipolar cells (B), horizonatal cells (H) and retina amacrine cells (A); and finally retinal ganglion cells (G) which transfer the visual information to the brain. [3] The aim of this project is to learn a model $p(y|x)$ of how spike trains y from the ganglion cells will respond to various stimuli x and then use this model to predict future spike trains for new stimuli (we are unconcerned about modeling the explicit structure of the neurons between the photoreceptors and the ganglion cells).

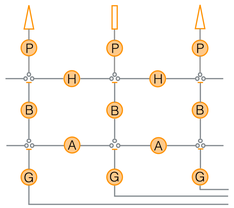

Figure 1: Diagram of the structure of retinal neurons (Redrawn from [3]).

DATA AND FEATURES

The data taken in the Baccus lab was acquired by surgically removing a salamander retina and stimulating it with three different types of signals: high-contrast Gaussian noise and natural scenes. [2] While the photoreceptors receive these signals, the output of the retinal ganglion cells are simultaneously recorded.

When working with recorded spike trains, our first step is to bin the spikes into some small time interval which we call τ, in this case 10 ms. Predicting the spike train then reduces to predicting the number of spikes that occur in each bin. Because we assume the system is causal, our features will be the tensor of movie images (2 space dimensions, 1 time dimension) in a time window before the point we want to predict, say 350 ms. We denote x_t to be the stimuli vector in this window proceeding time t. [4]

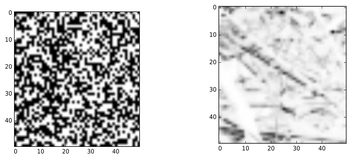

Figure 2: Sample high-contrast Gaussian noise (left), natural scene (right).

POISSON MODELS

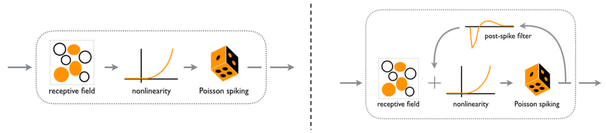

Figure 3: LNP Model (left) and GLM (right). (Redrawn from [4]).

We will split the probabilistic relationship between the stimuli and the retinal responses $p(y|x)$ into two parts: a probabilistic model of neuron spiking (a Poisson process) and a functional relationship between all the factors that we think affect neuron spiking (stimuli, spike train history, refractoriness) and the intensity λ in the Poisson process. [4]

$$p(\mathbf{y}|\theta) = \prod_t \frac{(\lambda_t \tau)^y}{y!} \exp(-\lambda_t \tau), \quad \ell(\theta) = \sum_t y_t \log \lambda_t - \tau \sum_t \lambda_t + c$$

We need a model of how the intensities λ_t relate to the stimuli observed by the photoreceptors. [4] For this, we will incorporate two insights from biological research. First, each ganglion cell has a receptive field, or region in space that it responds to stimuli. [3] Second, neurons exhibit refractoriness (less likely to respond immediately after firing). [1]

1. **Linear-Nonlinear Poisson (LNP) Model**

$$\lambda_t = f(\mathbf{k} \cdot x_\mathbf{t})$$

In the LNP model we assume that the receptive field $\mathbf{k}$ of the neuron acts linearly on the stimuli vector before being passed through a static nonlinearity f to give us the spiking rate.

2. **Generalized Linear Model (GLM)**

$$\lambda_t = f(\mathbf{k} \cdot x_\mathbf{t} + \mathbf{h} \cdot y_\mathbf{t})$$

The GLM is identical to the LNP model described in the previous section except that we incorporate the idea that spiking rates are also affected by the spiking history of the neuron due to properties like refactoriness, modelled by the linear post-spiking filter $\mathbf{h}$.

DEEP NETWORK

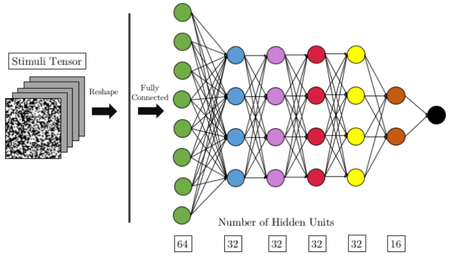

Figure 4: Deep network architecture designed in TensorFlow. ReLU neurons were used with dropout between most hidden layers. Weights initialized to 10^{-3}; optimized using ADAM with a learning rate of 10^{-4}.

CLUSTERING RESULTS

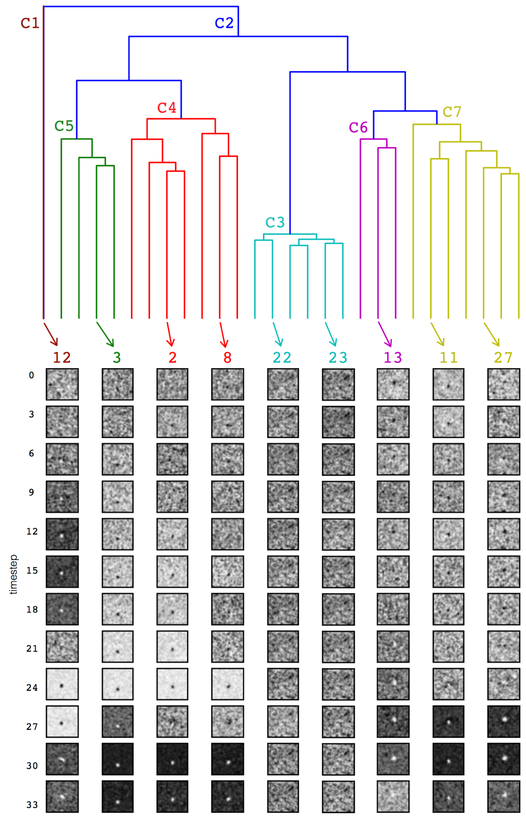

Figure 5: Clustering of the neurons by learned space-time receptive field using hierarchical clustering. Samples of the learned receptive field over time show the different types of neuronal behavior observed in each cluster.

REFERENCES

[1] K. Doya, S. Ishii, A. Pouget, and R. P. N. Rao, *Bayesian brain: Probabilistic approaches to neural coding*. Cambridge, MA: MIT Press, 2007.
[2] S. Baccus, "Baccus lab: Department of neurobiology," https://sites.stanford.edu/baccuslab/.
[3] S. A. Baccus, "Timing and computation in inner retinal circuitry," Annu. Rev. Physiol. , vol. 69, pp. 271-290, 2007.
[4] J. Shlens, "Notes on generalized linear models of neurons," arXiv preprint arXiv:1404.1999 , 2014.
[5] T. Gollisch and M. Meister, "Eye smarter than scientists believed: neural computations in circuits of the retina," Neuron , vol. 65, no. 2, pp. 150-164, 2010.
[6] L. McIntosh, N. Maheswaranathan, A. Nayebi, S. Ganguli, and S. Baccus, "Deep learning models of the retinal response to natural scenes," NIPS , 2016.

REGRESSION RESULTS

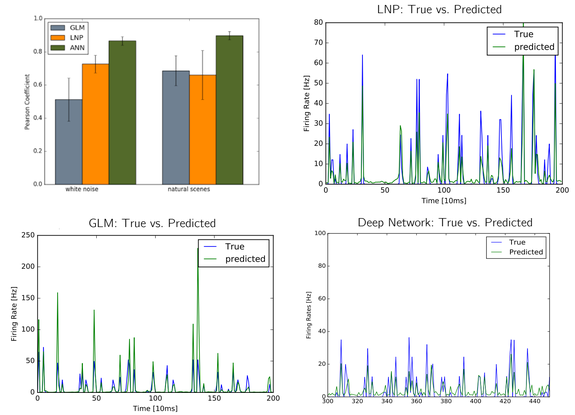

Figure 6: Statistics across all neurons (upper left); sample predictions for the LNP (upper right), GLM (lower left), and deep net (lower right).

The plots above show sample predicted values from each of our three models on different trials (thus, the true data is different for each). To give an idea of the overall performance of the model, we have compiled the average Pearson correlation coefficient (typical measure in neuroscience of similarity between time series) across all trials and neurons. For each trial, 70% of the data is used for training and 30% of the data for testing. The first column is for models trained and tested on the white noise and the second column is for models tested and trained on natural scenes. Note that for the deep network the coefficient was calculated based on 5 neurons only due to time constraints.

FUTURE WORK

The immediate next step for the project would be to further examine the idea of using neural networks to predict spike trains. Because of their natural application to image processing, we plan to examine the performance of ConvNets on predicting spike trains based on exciting results from work in Surya Ganguli's lab at Stanford [6]. As part of the project, we coded a basic ConvNet using TensorFlow, but it was too computationally intensive to run in the time remaining in the project and often such networks require significant fine-tuning. In addition, it would be interesting to see how the learned filters generalize from one type of stimuli to another, i.e. how the learned receptive filters from the white noise would perform on predicting spike trains from natural scenes.

ACKNOWLEDGEMENTS

The group would like to thank Steve Baccus's lab for the use of the retinal ganlion cell data as well as Niru Maheswaranathan and Lane McIntosh in Surya Ganguli's lab for many useful discussions and advice.