CHARACTERIZATION OF NONLINEAR NEURON RESPONSES

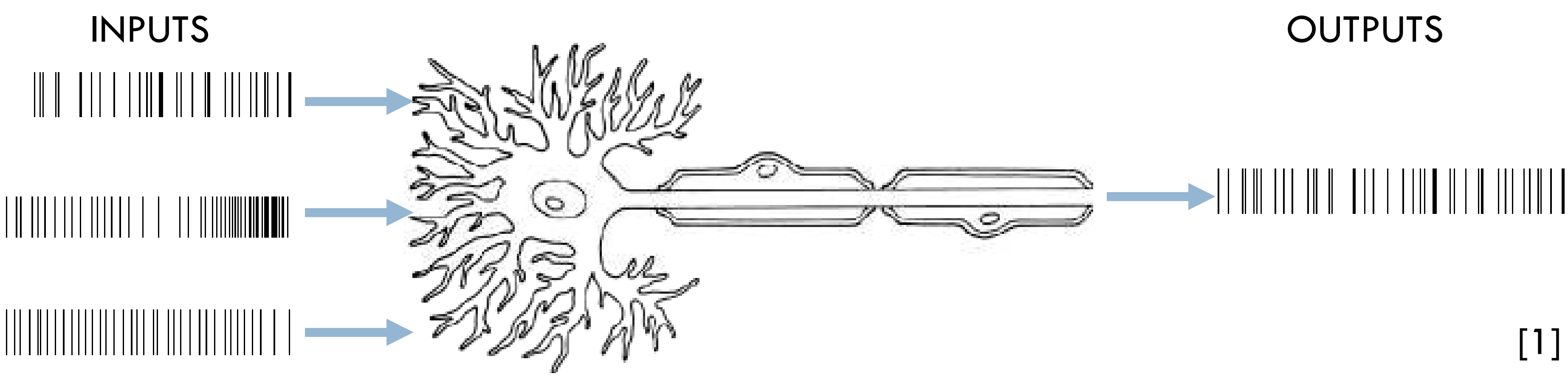
Matt Whiteway
whit8022@umd.edu

Dr. Daniel A. Butts
dab@umd.edu
Neuroscience and Cognitive Science (NACS)
Applied Mathematics and Scientific Computation (AMSC)
Biological Sciences Graduate Program (BISI)

AMSC 663 Project Proposal Presentation

Background

- The human brain contains 100 billion neurons
- These neurons process information nonlinearly, thus making them difficult to study



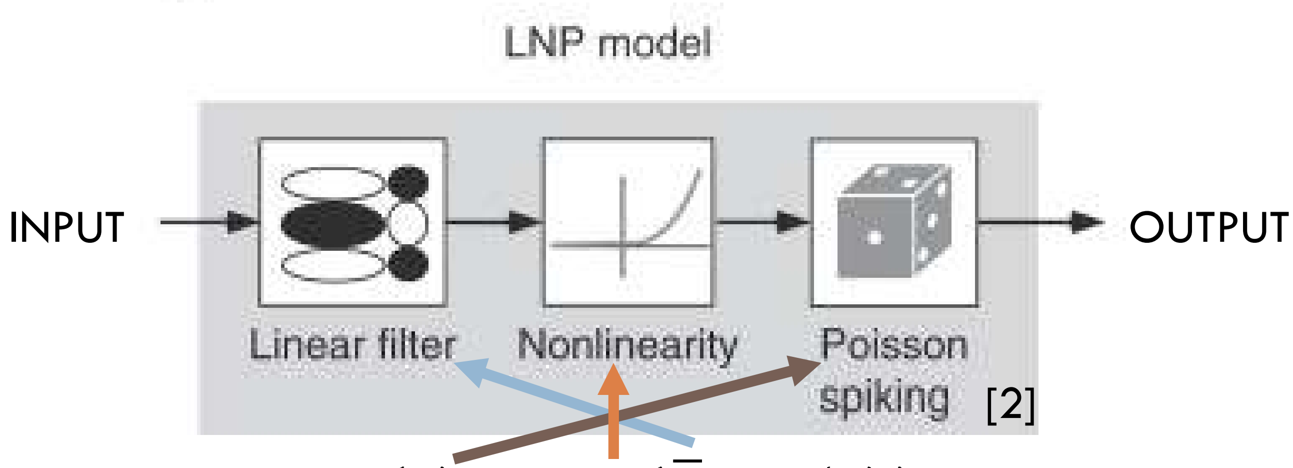
- Given the inputs and the outputs, how can we model the neuron's computation?

[1] http://msjensen.cehd.umn.edu/webanatomy_archive/Images/Histology/

The Models

- Many models of increasing complexity have been developed
- The models I will be implementing are based on statistics
  - Linear Models – Linear Nonlinear Poisson (LNP) Model
    1. LNP using Spike Triggered Average (STA)
    2. LNP using Maximum Likelihood Estimates – Generalized Linear Model (GLM)
    3. Spike Triggered Covariance (STC)
  - Nonlinear Models
    4. Generalized Quadratic Model (GQM)
    5. Nonlinear Input Model (NIM)

The Models - LNP



$$r(t) = F(\bar{k} \cdot \bar{s}(t))$$

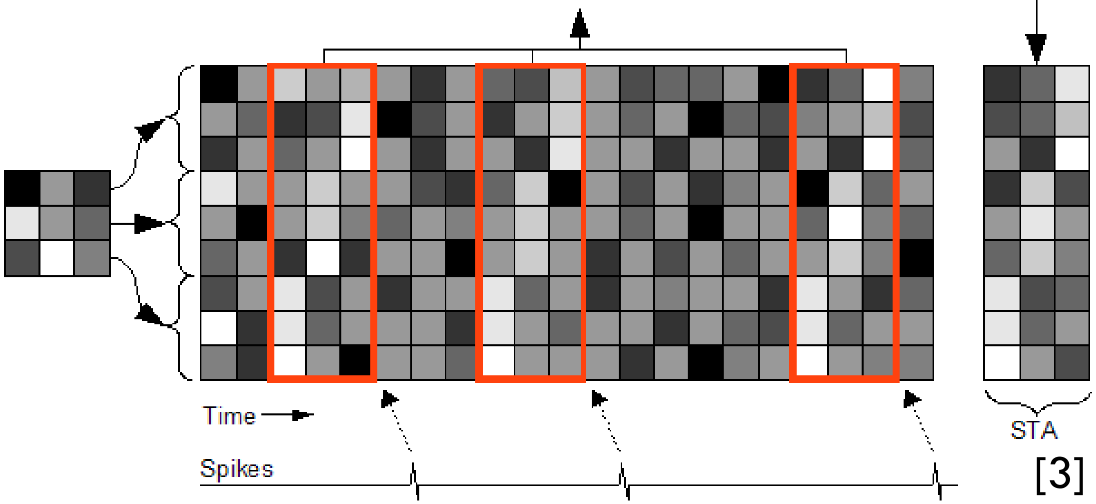
- Knowns
  - $\bar{s}(t)$ is the stimulus vector
  - Spike times
- Unknowns
  - $\bar{k}$ is a linear filter, defines the neuron's stimulus selectivity
  - F is a nonlinear function
  - $r(t)$ is the instantaneous rate parameter of an non-homogenous Poisson process

[2] http://www.sciencedirect.com/science/article/pii/S0079612306650310

The Models - LNP-STA[1]

- The STA is the average stimulus preceding a spike in the output, where N is the number of spikes and $\bar{s}_{spike}(t)$ is the set of stimuli preceding a spike

$$STA = \frac{1}{N}\sum_{n=1}^{N} \bar{s}_{spike}(t)$$



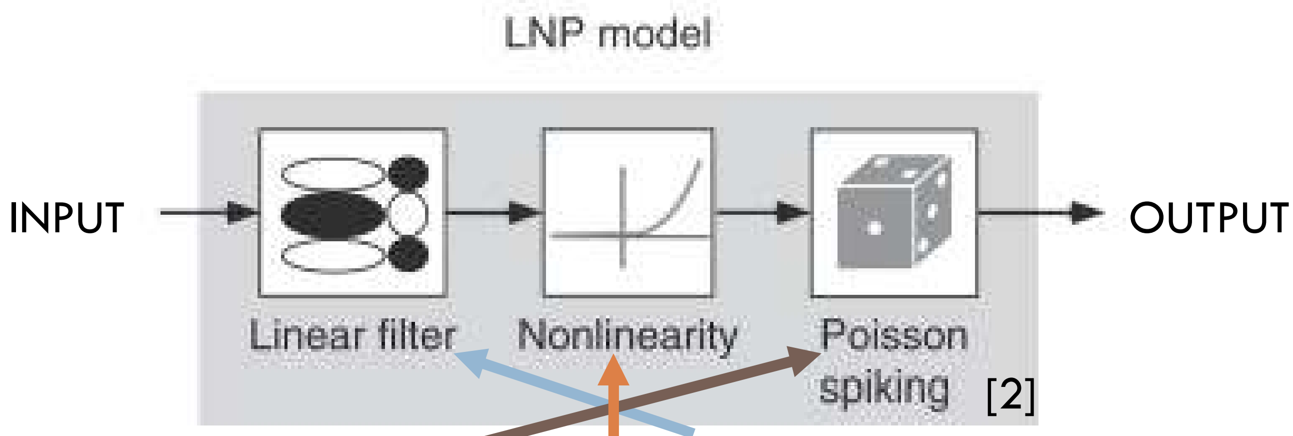
1. Chichilnisky, E.J. (2001) A simple white noise analysis of neuronal light responses.

[3] http://en.wikipedia.org/wiki/Spike-triggered_average

The Models - LNP-STA

- It can be shown that the STA is proportional to the linear filter $\bar{k}$
- Though it is possible to fit a parametric form of the nonlinear function, the more common approach is to bin the values of $\bar{k} \cdot \bar{s}(t)$ and plot average spike count for each bin based on the stimulus and spike data (histogram method).
- The Algorithm:
  - Calculate the STA to find $\bar{k}$
  - Use the stimulus and the filter for the histogram method to estimate F discretely

The Models - LNP



$$r(t) = F(\bar{k} \cdot \bar{s}(t))$$

- Knowns
  - $\bar{s}(t)$ is the stimulus vector
  - Spike times
- Unknowns
  - $\bar{k}$ is a linear filter, defines the neuron's stimulus selectivity
  - F is a nonlinear function
  - $r(t)$ is the instantaneous rate parameter of an non-homogenous Poisson process

[2] http://www.sciencedirect.com/science/article/pii/S0079612306650310

The Models - LNP-GLM[2]

- Now we will approximate the linear filter using the Maximum Likelihood Estimate (MLE)
- A likelihood function $\mathcal{L}(\theta)$ is the probability of an outcome Y given a probability density function with parameter θ
- The LNP model uses the Poisson distribution

$$P(Y|\theta) = \prod_t \frac{(r(t)\Delta)^{y_t}}{y_t!} e^{-r(t)\Delta}$$

where Y is the vector of spike counts binned at a resolution Δ

- We want to maximize a log-likelihood function

$$\mathcal{L} = \sum_{t=spike} log(r(t)) - \Delta \sum_t r(t)$$

2. Paninski, L. (2004) Maximum Likelihood estimation of cascade point-process neural encoding models.

The Models - LNP-GLM

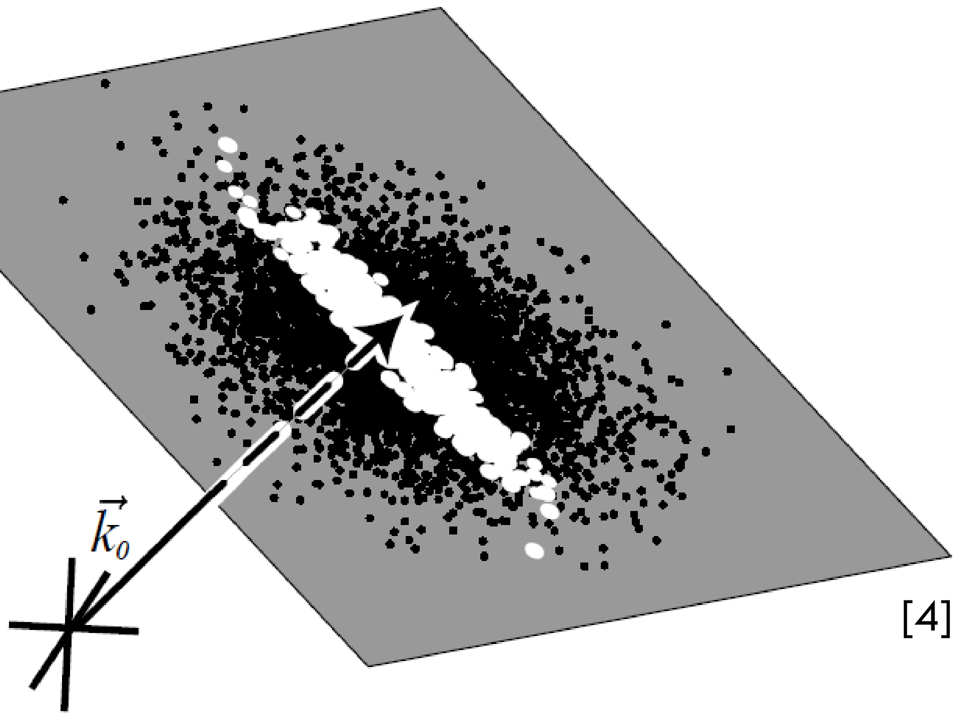
- The Maximum Likelihood Estimate maximizes the probability of spiking as a function of the model parameters when the actual data shows a spike, and minimizes it otherwise.
- Can employ likelihood optimization methods to obtain maximum likelihood estimates for linear filter $\bar{k}$
- If we make some assumptions about the form of the nonlinearity F, the likelihood function has no non-global local maxima – gradient ascent!
  - F(u) is convex in u
  - log(F(u)) is concave in u
- Regularizing estimated linear filter can reduce noise effects
  - Applies penalty for complexity

The Models - LNP-GLM

- The Algorithm:
  - Assume a form of *F* and get the corresponding log-likelihood function
  - Depending on the resulting function, find the MLE by analytical or numerical means

The Models - STC[3]

- Spike Triggered Covariance (STC) analysis is a way to identify a feature subspace that affects a neuron's response.
- Find excitatory and suppressive directions



[4]

3. Schwartz, O. et al. (2006) Spike-triggered neural characterization.
[4] http://books.nips.cc/papers/files/nips14/NS14.pdf

The Models - STC

$$STC = \frac{1}{N-1}\sum_{n=1}^{N}(\bar{s}_{spike}(t) - STA)(\bar{s}_{spike}(t) - STA)^T$$

$$C = \frac{1}{N-1}\sum_{n=1}^{N}\bar{s}_{spike}(t)\bar{s}_{spike}(t)^T$$

- The smallest eigenvalues associated with $STC - C$ correspond to inhibitory eigenvectors in the stimulus space; we will just use the smallest for a “suppressive” filter.

The Models - STC

- The model then becomes

$$r(t) = F(\bar{k}_e \cdot \bar{s}(t), \bar{k}_s \cdot \bar{s}(t))$$

- Knowns
  - $\bar{s}(t)$ is the stimulus vector
  - Spike times
- Unknowns
  - $\bar{k}_e$ is the excitatory filter
  - $\bar{k}_s$ is the suppressive filter
  - F is a nonlinear function
  - $r(t)$ is the instantaneous rate parameter of an non-homogenous Poisson process

The Models - STC

- Again, we could try to fit the nonlinearity parametrically, but using the histogram method is fine if we are just using two filters, with bin values coming from $\bar{k}_e \cdot \bar{s}(t)$ and $\bar{k}_s \cdot \bar{s}(t)$
- The Algorithm:
  - Calculate the STA to find $\bar{k}_e$
  - Calculate the eigenvectors/values of $STC - C$
  - Use eigenvector associated with smallest eigenvalue to find $\bar{k}_s$
  - Use the stimulus and the filter for the histogram method to estimate F discretely

The Models - Nonlinear Models

- We can extend these models even further by introducing nonlinearities in the input. The general form is:

$$r(t) = F\big(f_1(\bar{k}_1 \cdot \bar{s}(t)), \ldots, f_n(\bar{k}_n \cdot \bar{s}(t))\big)$$

  where the f_i's can be any nonlinear functions.
- Perhaps we should start with something easier first…
- Assuming that the input is a *sum* of one-dimensional nonlinearities makes parameter fitting much nicer

The Models - GQM[4]

- A particular choice of this sum of nonlinearities gives us the Generalized Quadratic Model:

$$r(t) = F\left(\frac{1}{2}\bar{s}(t)^T C \bar{s}(t) + \bar{b}^T \bar{s}(t) + a\right)$$

- Knowns
  - $\bar{s}(t)$ is the stimulus vector
  - Spike times
- Unknowns
  - matrix C
  - vector $\bar{b}$
  - scalar a
  - F is a nonlinear function
  - $r(t)$ is the instantaneous rate parameter of an non-homogenous Poisson process

4. Park, I., and Pillow, J. (2011) Bayesian Spike-Triggered Covariance Analysis.

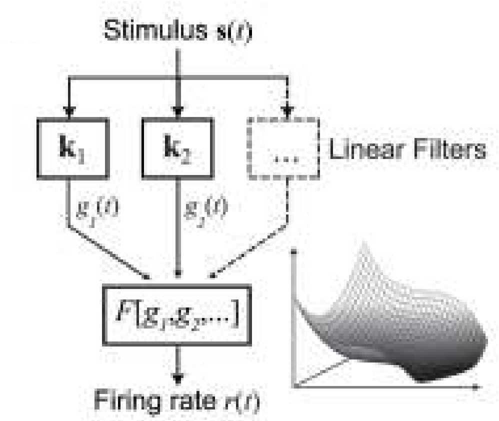
The Models - GQM

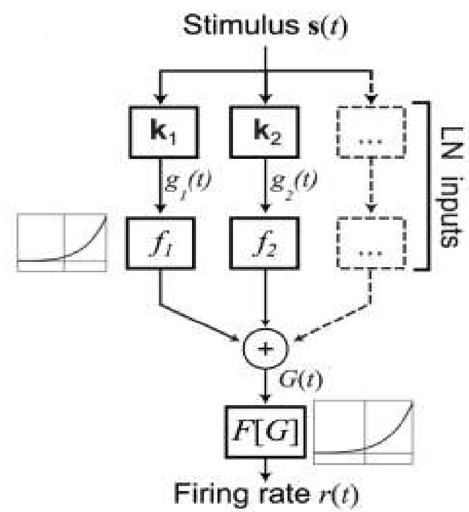
- Even though we are no longer dealing with the GLM, in practice if we choose a form of *F* that preserves concavity of the log-likelihood, estimating the parameters using MLE will tractable
- Functions commonly used are exp(u) and log(1+exp(u))
- The Algorithm:
  - Write down the log-likelihood function
  - Use optimization to find MLE, assuming one of the above forms for *F*

The Models - NIM[5]

LINEAR NONLINEAR MODEL

NONLINEAR INPUT MODEL [5]





$$r(t) = F(\bar{k}_1 \cdot \bar{s}(t), \bar{k}_2 \cdot \bar{s}(t), \ldots)$$

$$r(t) = F\big(f_1(\bar{k}_1 \cdot \bar{s}(t)), f_2(\bar{k}_2 \cdot \bar{s}(t) \ldots)\big)$$

- The Nonlinear Input Model (NIM) defines the neuron's processing as a sum of nonlinear inputs
- Allows for rectification of inputs due to spike-generation from the input neuron

5., [5] McFarland, J.M., Cui, Y., and Butts, D.A. (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs.

The Models - NIM

$$r(t) = F\left(\sum_i w_i f_i(\bar{k}_i \cdot \bar{s}(t))\right)$$

- Knowns
  - $\bar{s}(t)$ is the stimulus vector
  - Spike times
- Assumptions
  - $F(u) = \alpha \log[1 + e^{\beta(u-\theta)}]$
  - f_i's are rectified linear functions
- Unknowns
  - $\bar{k}_i$'s are the linear filters
  - w_i 's will be +/-1
  - f_i's are nonlinear functions
  - F is a nonlinear function
  - $r(t)$ is the instantaneous rate parameter of an non-homogenous Poisson process

The Models - NIM

- The Algorithm
  - Write down the log-likelihood function
  - Use optimization to find MLE, assuming one of the above forms for *F*
  - Regularization will allow us to incorporate prior knowledge about the parameters

Implementation and Data

- Parameter fitting will be implemented on an Intel Core 2 Duo with 3 GB of RAM
- Programming language will be MATLAB
- Data
  - Initial work will be done using data from a Lateral geniculate nucleus (LGN) neuron (visual system)
  - simulated data from other neurons in the visual system
  - Data can be found at http://www.clfs.umd.edu/biology/ntlab/NIM/

Validation

- Use stimulus data $\bar{s}(t)$ and response data $\bar{r}(t)$ to construct the model, i.e. find the model parameters
- Use stimulus data $\bar{s}(t)$ and the model to create new response data $\bar{r}_{model}(t)$
- Now use $\bar{s}(t)$ and $\bar{r}_{model}(t)$ to generate a new model
- Check to see how closely the parameters of the new model and the original model agree

Testing

- Use k-fold cross-validation on the log-likelihood of the model, LL_x, with the log-likelihood of the "null" model, LL_0, subtracted.
- Measure the 'predictive power' of the model by comparing the predicted model output to the measured output:
  - 'fraction of variance explained'

$$FVE = 1 - \frac{MSE(F)}{Var(\bar{r}_{measured}(t))}$$

- Use early models (i.e. LNP) to compare results of later models (i.e. NIM)

Project Schedule and Milestones

- PHASE I – September through December
  - Write project proposal (September)
  - Implement and validate the LNP model (October)
  - Develop software to validate models (October)
  - Implement and validate the GLM with regularization (November-December)
  - Complete mid-year progress report (December)
- PHASE II – January through May
  - Implement and validate the STC and GQM (January-February)
  - Implement and validate the NIM with linear rectified upstream functions (March)
  - Develop software to test all models (April)
  - Complete final report and presentation
- If time permits…
  - Fit the nonlinear upstream functions of the NIM using basis functions
  - Extend models to be used for more diverse stimulus types
  - Use the NIM to model small networks of neurons

Deliverables

- Implemented MATLAB code for all of the models (LNP-STA, LNP-GLM, STC, GQM, NIM)
- Documentation of code
- Results of validation and testing for all of the models
- Mid-year presentation
- Final report
- Final presentation

References

- Chichilnisky, E.J. (2001) A simple white noise analysis of neuronal light responses. *Network: Comput. Neural Syst.*, 12, 199-213.
- Schwartz, O., Chichilnisky, E. J., & Simoncelli, E. P. (2002). Characterizing neural gain control using spike-triggered covariance. *Advances in neural information processing systems*, *1*, 269-276.
- Paninski, L. (2004) Maximum Likelihood estimation of cascade point-process neural encoding models. *Network: Comput. Neural Syst.* ,15, 243-262.
- Schwartz, O. et al. (2006) Spike-triggered neural characterization. *Journal of Vision*, 6, 484-507.
- Paninski, L., Pillow, J., and Lewi, J. (2006) Statistical models for neural encoding, decoding, and optimal stimulus design.
- Park, I., and Pillow, J. (2011) Bayesian Spike-Triggered Covariance Analysis. *Adv. Neural Information Processing Systems* ,24, 1692-1700.
- Butts, D. A., Weng, C., Jin, J., Alonso, J. M., & Paninski, L. (2011). Temporal precision in the visual pathway through the interplay of excitation and stimulus-driven suppression. *The Journal of Neuroscience*, *31*(31), 11313-11327.
- McFarland, J.M., Cui, Y., and Butts, D.A. (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs. *PLoS Computational Biology*.

Figures

1. http://msjensen.cehd.umn.edu/webanatomy_archive/Images/Histology/
2. http://www.sciencedirect.com/science/article/pii/S0079612306650310
3. http://en.wikipedia.org/wiki/Spike-triggered_average
4. http://books.nips.cc/papers/files/nips14/NS14.pdf
5. McFarland, J.M., Cui, Y., and Butts, D.A. (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs. PLoS Computational Biology.