Project proposal:

Statistical models of visual neurons

Anna Sotnikova
asotniko@math.umd.edu

Project Advisor: Prof. Daniel A. Butts
dab@umd.edu

Department of Biology

Abstract

Studying visual neurons may explain how a human brain performs image analysis. The main question is what mathematical function can represent the processes occurring in the brain. Nowadays there is a variety of models, which are trying to reproduce them. The estimation of precision of such models is the difference between the predicted output (firing rate) and the real experimental data. In this project we will be implementing 5 models, which predict neurons' firing rates. The implementation of the models also requires parameters fitting, validation and testing procedures.

October, 6 2016

1 Project Background

Here is a brief introduction to the eye's structure. For this project we are focusing on the feed-forward processing model. In this model, the input goes to retina first, which is the back side of the eye, then to lateral geniculate nucleus(LGN) and then to the primary visual cortex(V1). We are not considering steps after V1 phase due to their complexity and will work only with retina, LGN, and V1 data. Input data can be viewed as a 2D matrix of light intensity that varies in time (so it's a sequence of 2D matrices, each corresponding to intensity of light as a function of XY coordinates and time). Input is called stimulus and denoted by letter s. Output data can be viewed as a probability of spikes at a given time. Spike is a defined as a delta function. Spike frequency depends on the experiment. Typically, stimuli are displayed between 16-120 Hz. [meaning 62.5 ms down to 8.3 ms]. Output is called firing rate and denoted by letter r.

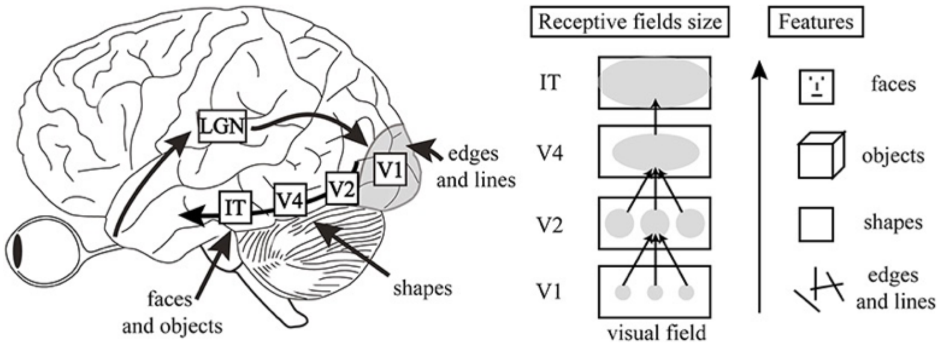

Fig. 1 The feed-forward processing model

First important experiment in the area of information processing in the visual system was done by David Hubel and Torsen Wisel in 1958. They anesthetized a cat, fixed its head, and put electrodes into the primary visual cortex area of the cat's brain. David H. Hubel and Torsten N. Wiesel projected images of a light bar on a screen and were changing it's position and angle.

The main goal of the experiment was to understand how neurons extract a complex information from the pattern of light cast on retina and construct an image. Their experiment revealed how visual neurons encoded image features. Also they discovered that if one neurons fires at some specific position of the light bar, this does not guarantee the other neuron will react similarly.

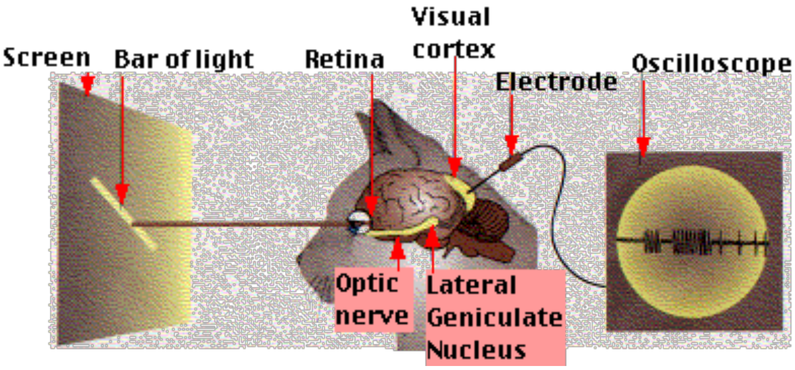

Fig. 2 David Hubel and Torsen Wisel experiment
source http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/V/VisualProcessing.html

Many researches are trying to understand what is the mathematical description of the processes, which happen in the brain during "image processing". What kind of function can describe relation between the input (stimulus) and the output (firing rate)? Such function may have linear, or quadratic , or cascade form. Cascade form means repetition of combinations of linear and non-linear functions. It is unlikely that there exist a simple function that does this. We therefore must split this function into a set of other functions with optimally distributed properties (for instance, linear filter + non-linear estimator function) and this choice of functions will be a model. If we make our model too complicated, it might indeed cover all the observations, but it might require too complicated parameters fitting procedure. Therefore we must consider maximally simple models.

2 Project Objective

For the project, I will be implementing all three mentioned above types of models.

1. Linear models
   - Spike Triggered Average (STA)
   - Generalized Linear Model (GLM)
2. Quadratic models
   - Spike Triggered Covariance (STC)
   - Generalized Quadratic Model (GQM)
3. Cascade model
   - Nonlinear Input Model (NIM)

STA and STC models do not require parameters fitting. The rest models require parameters fitting, which is explained in sections devoted to the models' description. The general idea of parameters fitting is that we need to find such a model, which will be suitable for different sample data rather than catch all details of a particular data set. This might cause overfitting, when we a have a model with more described parameters than needed. Validation and testing procedures are explained in the section 3. Synthetic and real data sets will be used for the validation and testing.

2.1 STA Model

In the spike triggered average(STA) model the output is performed by a linear combination of stimulus parameters. The STA is represented by the formula (1)

(1)
$$\phi_{sta} = \frac{1}{N} \sum_t n(t)(\boldsymbol{s}(t) - \bar{\boldsymbol{s}})$$

where N is the total number of spikes, n(t) is the number of spikes at time t, which might be equal 1 or 0 (spike occurred or not, respectively), s is a stimulus and

(2)
$$\bar{\boldsymbol{s}} = \frac{1}{M} \sum_t \boldsymbol{s}(t)$$

which is the average stimulus with M equals to the total number of stimuli per experiment. The average stimulus was obtained by averaging stimuli, which precede the spike.[6]

STA estimates the linear stage, which corresponds to one filter. [1] STA might be used for fitting data collected from retina cells, lateral geniculate nucleus, and primary visual cortex. However, this model very often does not fit fully because neural responses are mostly non-linear. The most general interpretation of STA is the receptive field of a neuron, which defines the preferred stimulus for the neuron. [1]

2.2 STC Model

The STA model analyzes changes in the spike-triggered stimulus's mean for estimating linear part of linear non-linear model. However, it corresponds only to the single direction of the stimulus. The Spike Triggered Covariance(STC) is used when we need to predict probability of a spike along more than one direction.

(3)
$$\phi_{stc} = \frac{1}{N-1} \sum_t n(t)(\boldsymbol{s}(t) - \phi_{sta}) \cdot (\boldsymbol{s}(t) - \phi_{sta})^T$$

where N is the total number of spikes, n(t) is the number of spikes at time t, which might be equal 1 or 0 (spike occurred or not, respectively), s is a stimulus and STA is defined by formula (1).[6]

STC gives a quadratic model for neural responses and as well as STA cannot fit data completely. That is why it is often used as starting point for estimation of another model. Geometrical idea of STC is that we are looking for such directions along which the variance of spike-triggered stimulus differs from the raw stimulus. STC model determines excitatory or inhibitory properties of neurons' responses. Excitatory property is defined by the increase in variance, and inhibitory property is defined by the decrease in variance.[5]

2.3 GLM model

The idea of the Generalized Linear Model(GLM) is represented by the fig. 3, when we apply linear filter(or linear receptive field) to the stimulus. Next step is adding to the new term spike history term and applying non-linear function. This gives us firing rate us output, which is used for spikes predicting. [3]

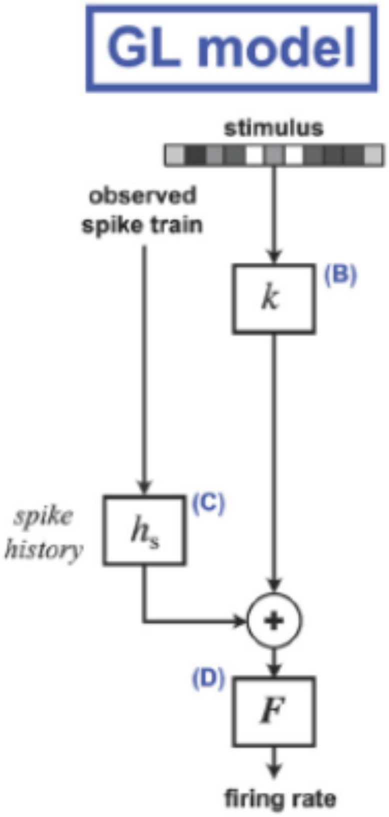

Fig. 3 Generalized linear model. (B) - apply filter to the stimulus. (C) - add spike-history term and summarize (B) and (C). (D) - apply non-linear function to the sum and get the firing rate.
source : reference [3]

The firing rate of the GLM is defined by the formula

(4) $$r(t) = F\Bigg(\boldsymbol{s}(t) \cdot \boldsymbol{k} + h_s \cdot \boldsymbol{R}(t) + \boldsymbol{b}\Bigg)$$

where   h_s and are R(t) terms related to the spiking history, b shifts spiking non-linearity.These are linear terms and are fit simultaneously with the [linear] filter coefficients. F is spiking non-linearity, s is a stimulus and k is a linear filter.

Based on Paninski work [4], we assume that spiking non-linearity is

(5) $$F(g) = log(1 + e^g)$$

The GLM includes the spiking history part because this pat might affect the ability of a neuron to react on a stimulus. All neurons have some refractory period that might not allow neuron to fire shortly after a previous spike even for the “right stimulus”. [6]

This model can be used to fit non-linear inputs.

2.3.1 GLM model parameters fitting

For using GLM we need to fit a linear filter k and shift term b. Shift term b is approximately estimated based on the data. For estimating linear filter k two approaches might be used: STA model or maximum log likelihood. The second one is more powerful.

For this approach we assume that the neuron’s spikes are described by Poisson distribution, then log likelihood [4]

(6) $$LL = \sum_t (R_{obs}(t) log(r(t)) - r(t))$$

where R_obs is the binned spike times and r(t) is defined by formula (4).
In other words, in each time bin we observe some number of spikes. Usually, it is 0, sometimes 1, and it might be greater than 1 for a large bin size.

Next step is to take the derivative of LL with respect to k and from (4) and (6) get

(7) $$\frac{\partial LL}{\partial k_m} = \sum_t (\frac{R_{obs}(t)}{r(t)} - 1)F'(G(t))s_m(t)$$

where s_m*th* is the m*th* element of a stimulus at time t, F is (5) and G(t) is defined as

(8) $$G(t) = \boldsymbol{s}(t) \cdot \boldsymbol{k} + h_{spk} \cdot \boldsymbol{R}(t) + \boldsymbol{b}$$

Then using Gradient ascend method we need to find maximum, and the result will be the optimal filter. [4]

Gradient ascend method is the first-order optimization algorithm. In order to find a local maximum of a function, one need to take steps from a point of an initial guess proportional to the positive gradient of a function. [7]

The gradient ascend method is defined as

(9) $$\boldsymbol{b} = \boldsymbol{a} + \gamma \nabla F(\boldsymbol{a})$$

where a is a point of an initial guess, gamma is a step size (typically small), and b is a new point in a direction of positive gradient. Gradient is defined by the formula (7)

The algorithm contains three steps: 1 - take some initial guess point a, 2 - find a new point b by formula(9), 3 -while the norm of the difference between the value of the function into b and the value of function into a is greater than some defined threshold, make a equals b and repeat step 2.

2.4 GQM model

The Generalized Quadratic Model (GQM) is a probabilistic generalization of STA/STC analysis and of the second-order Wiener-Volterra expansion [2].

(10) $$r(t) = F\left(\boldsymbol{k}_L \boldsymbol{s} + \sum_{i=1}^{M} w_i (\boldsymbol{k}_i \cdot \boldsymbol{s})^2\right)$$

where k_L is a linear filter and k_i are M squared filters, omega is a weight parameter, which can be +1 or - 1, which defines excitatory or inhibitory properties of a filter. F is (5), and s is a stimulus.

GQM is good for quadratic approximations of neuronal responses. [2]

2.4.1 GQM model parameters fitting

As the first stage of the fitting procedure this model requires k_L and k_i fitting. We use formula (6) again but with r(t) defined by (10).
Then take the derivative with respect to k_l and find maximum using gradient-based method.

(11) $$\frac{\partial LL}{\partial k_{L,m}} = \sum_t (\frac{R_{obs}(t)}{r(t)} - 1)F'(G(t))s_m(t)$$

where s_m*th* is the m*th* element of a stimulus at time t, F is (5) and G(t) is defined as

(12) $$G(t) = \boldsymbol{k}_L \boldsymbol{s} + \sum_{i=1}^{M} w_i(\boldsymbol{k}_i \cdot \boldsymbol{s})^2$$

Next step is taking the derivative of (6) with respect to k_i and finding maximum using gradient -based method.

(13) $$\frac{\partial LL}{\partial k_{i,m}} = \sum_t (\frac{R_{obs}(t)}{r(t)} - 1)F'(G(t))2k_{i,m}w_i s_m^2(t)$$

where s_m*th* is the m*th* element of a stimulus at time t , F is (5) and G(t) is defined by (12)

Formula for the gradient-based method is (9), the gradient functions are (11) and (13)

The last stage of the fitting procedure is defining how many k_i filter are required for the model's fitting and their properties (excitatory or inhibitory). In other words, we need to understand how many k_i filters do we need and what are the values of omega. This will be done by checking combinations of different types of filters(excitatory or inhibitory) and finding the maximal value. For instance, the best choice of filters for synthetic data of retinal cells with which I will be working is one linear and two quadratic filters [2]. Two filters span the relevant stimulus subspace, and the third one gives the best quadratic approximation to the input data.

2.5 NIM model

The Non-Linear Input Model (NIM) is represented by the following steps: apply some number n of squared filters k, then apply upstream nonlinearity function f, which is defined as rectified function. Next step is adding input weights (excitatory or inhibitory), and finally apply spiking non-linearity function, which is defined by (6). [2]

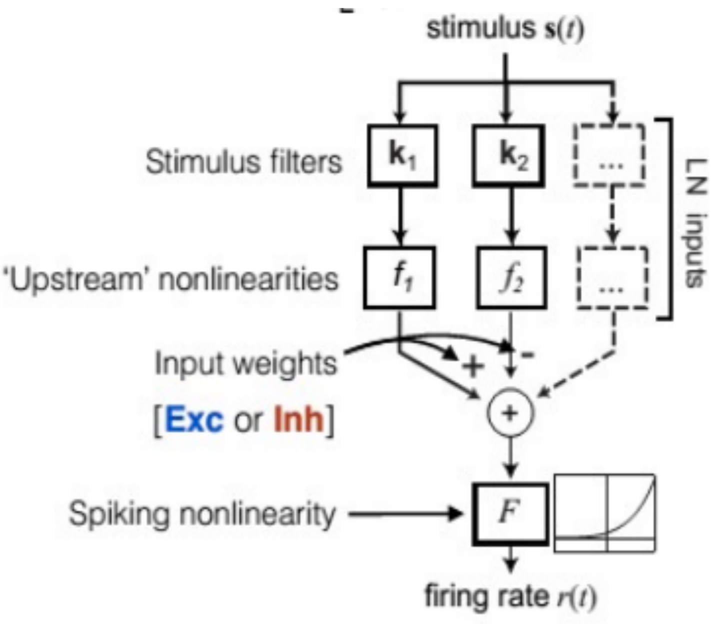

Fig. 4 NIM scheme
source: reference [2]

The firing rate for NIM is given by

(14) $$r(t) = F\left(\sum_{i=1} w_i f_i(\boldsymbol{k}_i \cdot \boldsymbol{s}(t))\right)$$

where F is defined by (5), f is upstream nonlinearities defined by (15), omega is weight parameter, k’s are filters, F is (5) and s is a stimulus. [2]

2.5.1 NIM model parameters fitting

This model requires squared filters fitting. We fix upstream nonlinearity as a rectified function.

(15) $$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{otherwise} \end{cases}$$

Take the derivative of LL (7) with respect to k_i and find maximum using gradient-based method (9).

(16) $$\frac{\partial LL}{\partial k_{i,m}} = \sum_t (\frac{R_{obs}(t)}{r(t)} - 1) F'(G(t)) w_i s_m(t)$$

where s m*th* is the m*th* element of a stimulus at time t, F is (5) and G(t) is defined as

(17) $$G(t) = \sum_{i=1} g_i(t) = \sum_{i=1} w_i f_i(\boldsymbol{k}_i \cdot \boldsymbol{s}(t))$$

The next step is defining how many k_i filter are required for the model's fitting and their properties (excitatory or inhibitory). This will be done as for GQM. [2]

3 Validation and testing

There are two stages for validation and testing. The first one is running all 5 codes for 5 models for synthetic data, which was used for Dr. Butts paper of 2013 [2]. The outputs of the codes will be plotted and compared with results in the paper [2] fig. 5, which will determine the validity of the codes. Moreover, the output for the synthetic data is predictable, and there is no unexpected noise, which might spoil the picture. Synthetic data is written for V1 cells and retinal cells.

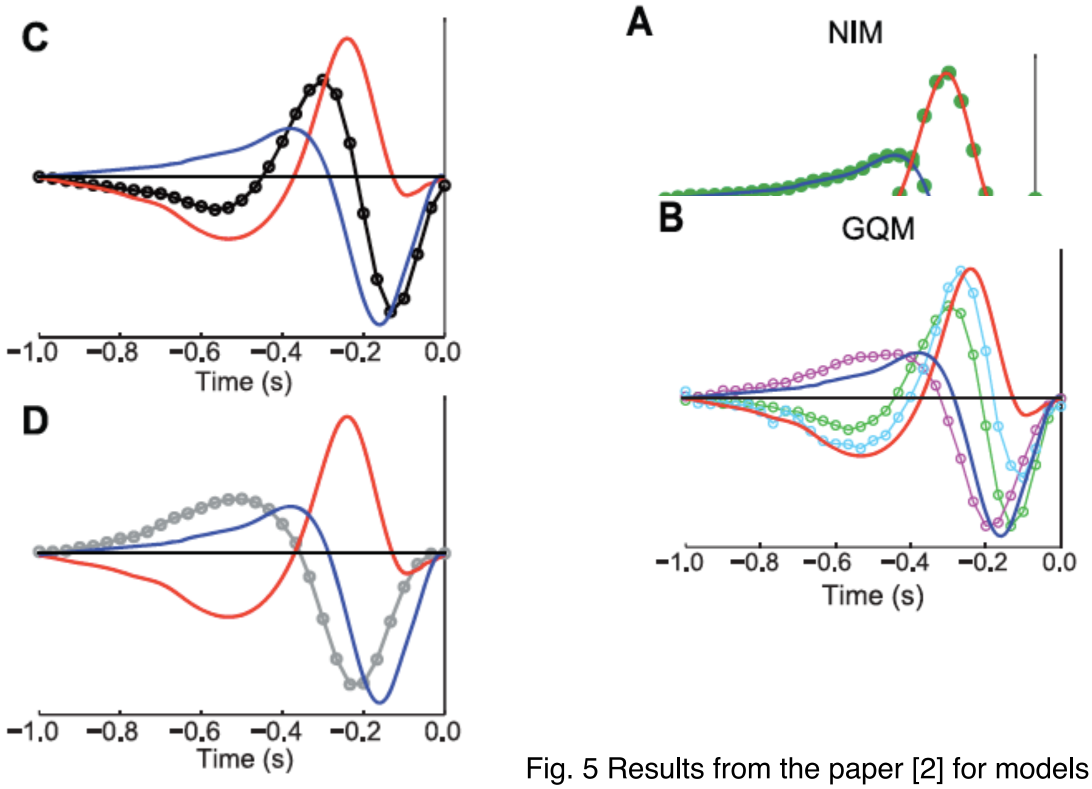

Fig. 5 Results from the paper [2] for models STA,STC, GQM, and NIM

source: reference [2]

C - the black line is STA, which represents an average of On(red) and Off(blue) filters.
D - the grey line is STC, which represents the mixture of On and Off filters.
A - green dots are NIM, which fully captures On and Off filters.
B - magenta and light blue lines are squared filters, the green line is a linear filter. Red line is On filter and blue line is Off filter.

The second stage is applying the cross validation procedure for the set of the real data collected from LGN cells of three cats. This data was used in Dr. Butts 2011 paper [3]. We will use 80% of the data for the training purposes and 20% of the data for the testing purposes. The output of the testing part will be compared with the real data.

For both stages respective plots will be provided. All data sets include vectors of stimuli of the size around 5e4x1, vectors of spike times of the size around 8e4x1 in units of seconds, and time interval of the stimulus updates.

4 Timeline

<u>October - mid November</u>

- Implement STA and STC models

- Test models on synthetic data set and validate models on LGN data set

November - December

- Implement GLM
- Test model on synthetic data set and validate model on LGN data set

January - March

- Implement GQM and NIM
- Test models on synthetic data set and validate models on LGN data set

April - May

- Collect results and prepare final report

5 Deliverables

- Matlab code for all 5 models
- List of models' fitted parameters
- Reports and presentations
  - Project proposal report and presentation
  - Mid-year progress presentation
  - Final paper and presentation

6 References

[1] Chichilnisky EJ (2001) A simple white noise analysis of neuronal light responses. Network 12:199 –213.
[2] McFarland JM, Cui Y, Butts DA (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs. PLoS Computational Biology 9(7): e1003142.
[3] Butts DA, Weng C, Jin JZ, Alonso JM, Paninski L (2011) Temporal precision in the visual pathway through the interplay of excitation and stimulus-driven suppression. J. Neurosci. 31: 11313-27.
[4] Paninski L (2004) Maximum likelihood estimation of cascade point-process neural encoding models. Network 15: 243-262.
[5] Simoncelli EP, Pillow J, Paninski L, Schwartz O (2004) Characterization of neural responses with stochastic stimuli. In: The cognitive neurosciences (Gazzaniga M, ed), pp 327–338. Cambridge, MA: MIT.
[6] Aljadeff J, Lansdell BJ, Fairhall AL, and Kleinfeld D (2016) Analysis of Neuronal Spike Trains, Deconstructed. Neuron, Volume 91, Issue 2, Pages 221–259
[7] Gradient ascend method https://en.wikipedia.org/wiki/Gradient_descent