Computational Neuroscience – Project 3

Group 3

Contributions:

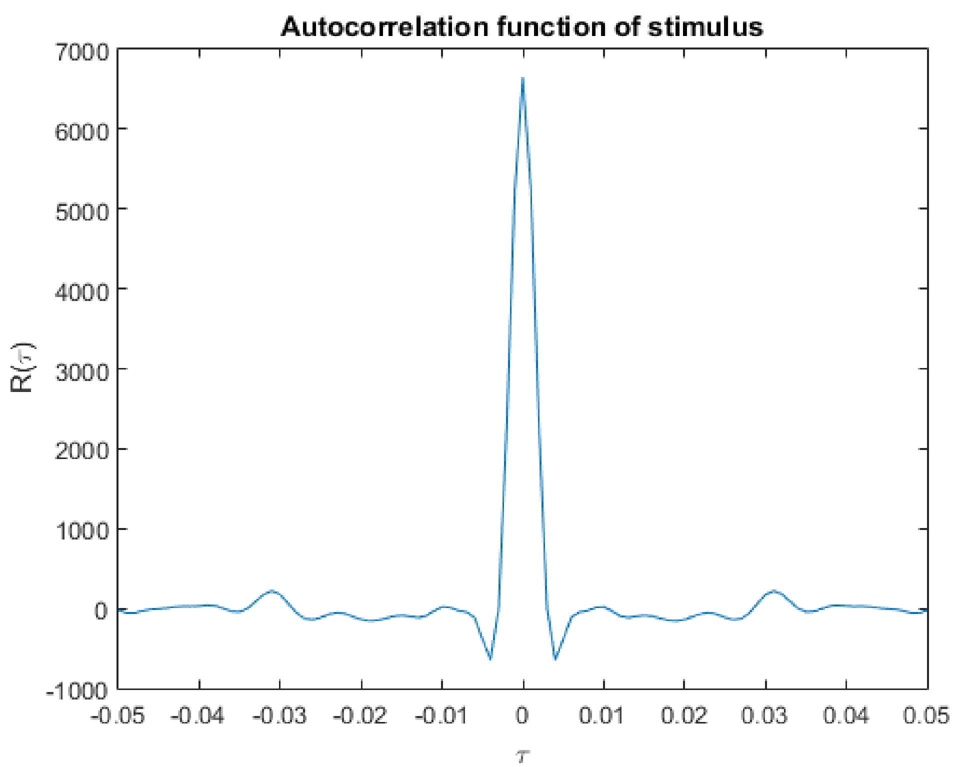
- Part 1 – Aditya, Sai Krishna
- Part 2 – Aditya, Bhumika
- Part 3 – Aditya, Vijay
- Part 4 – Aditya, Manash, Sai Krishna
- Part 5 – Aditya, Manash
- Part 6 – Aditya
- Part 7 – Aditya

1. Stimulus Nature

While deriving the relation between the filter function and the STA, we were getting:

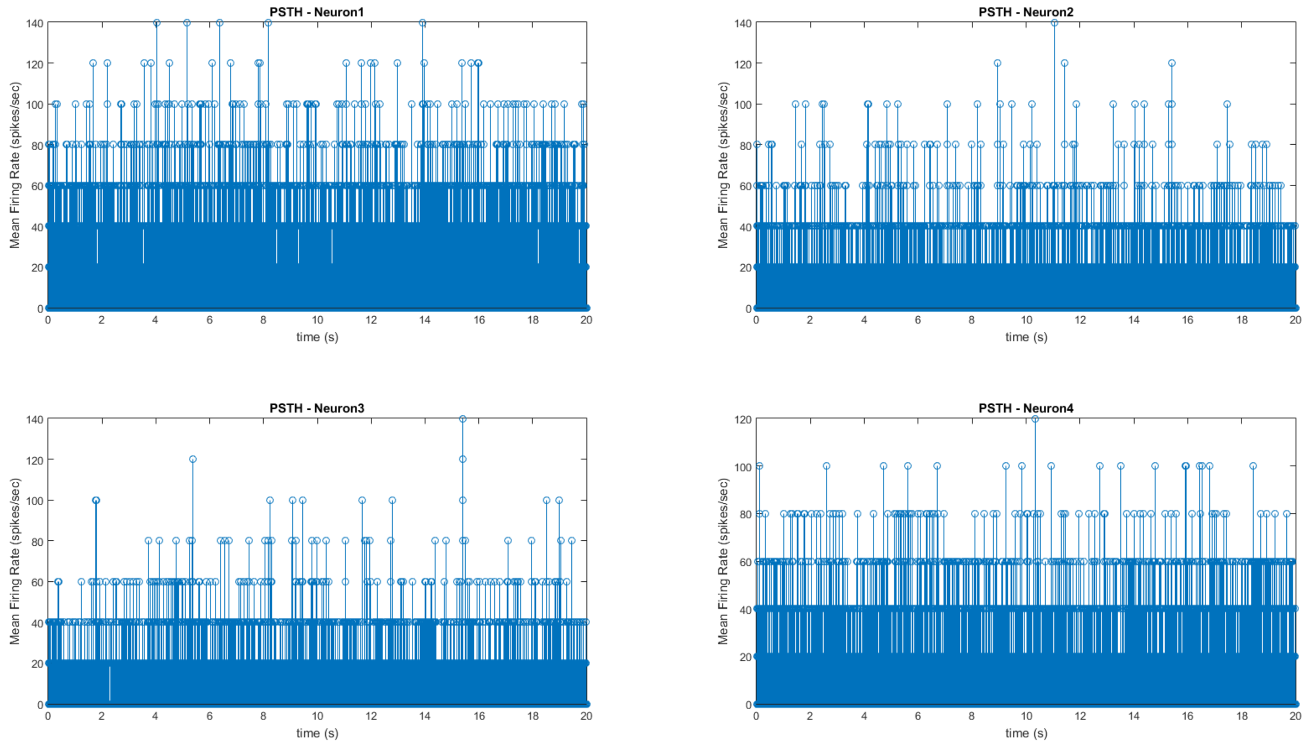
$$R_{YX}(\tau) = h(\tau) * R_{XX}(\tau)$$

This gives a direct proportionality between R_{YX} and h only if R_{XX} is in the form of a delta function, which is only true for Gaussian distributed stimulus. For the general case, we have to perform pre-whitening to first convert the stimulus to white noise. This will be indeed the requirement in our case, as the autocorrelation of the stimulus is as follows:



2. PSTH and mean firing rate

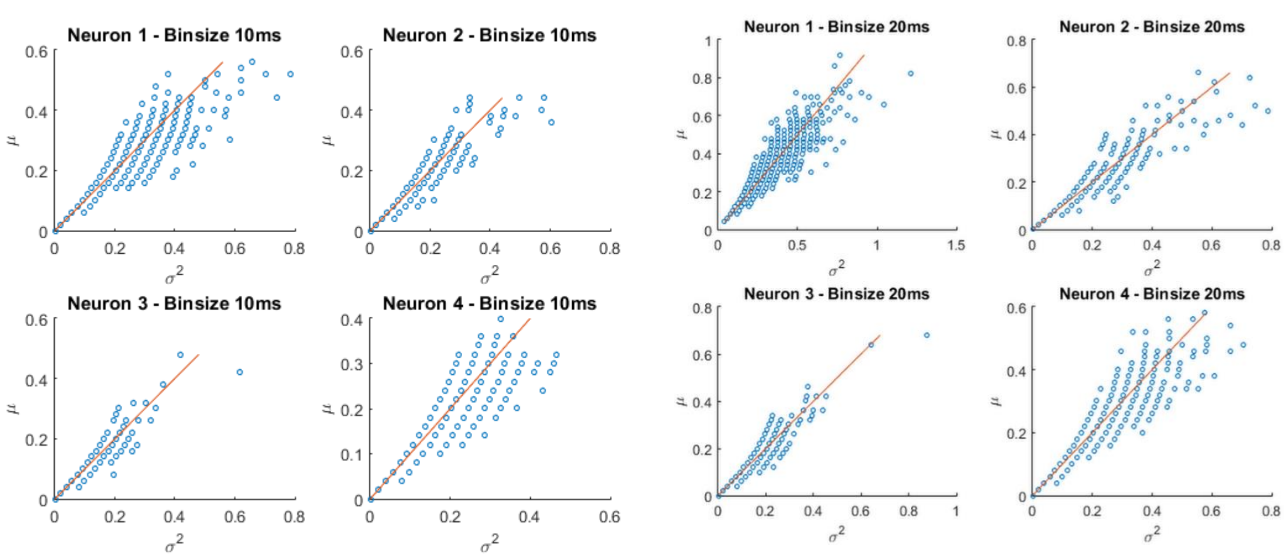
The Peri-Stimulus Time Histogram (PSTH) is a frequentist indicator of the mean firing rate (which is λ(t) in the case of a Poisson Process). Here are the plots for the same:

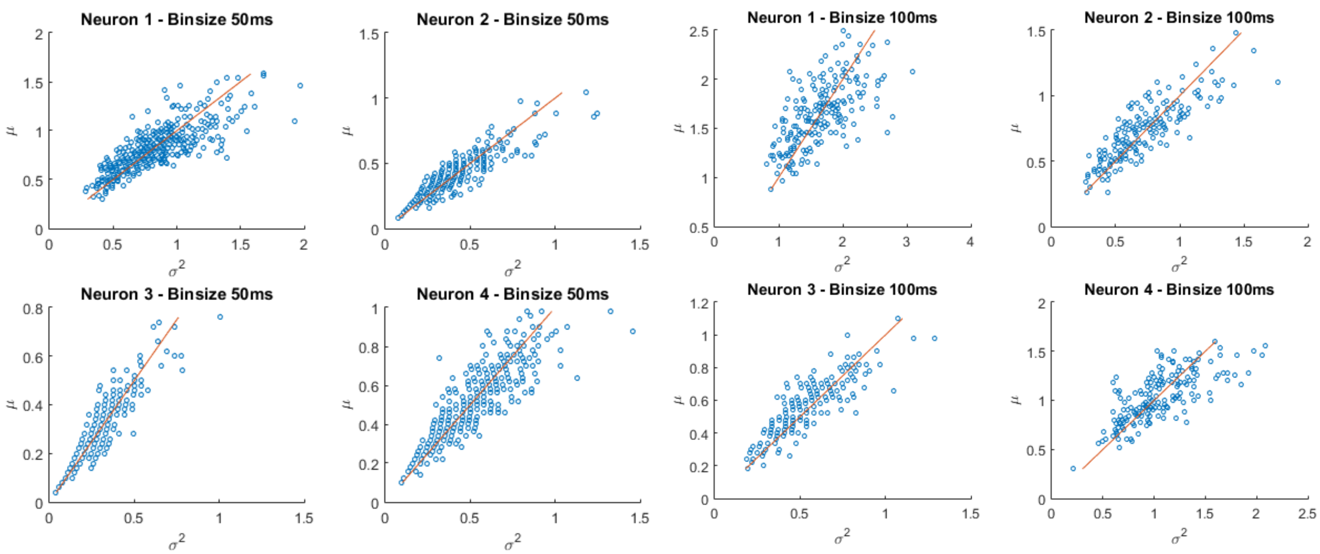


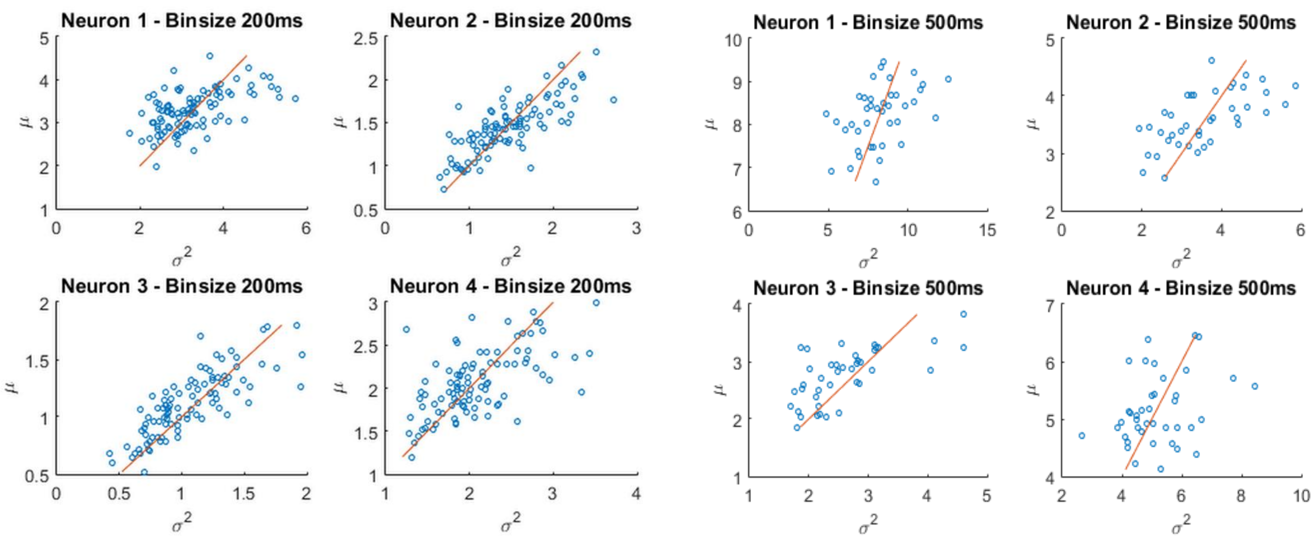
For interpreting the result, we have plotted the mean rate, rather than the normalized values.

3. Scatter plots and Poisson determination

For determining whether the point process is Poisson, we plot the variance vs. mean graph of the distribution of spikes for various bin sizes. This should be a straight line if a Poisson distribution is followed.





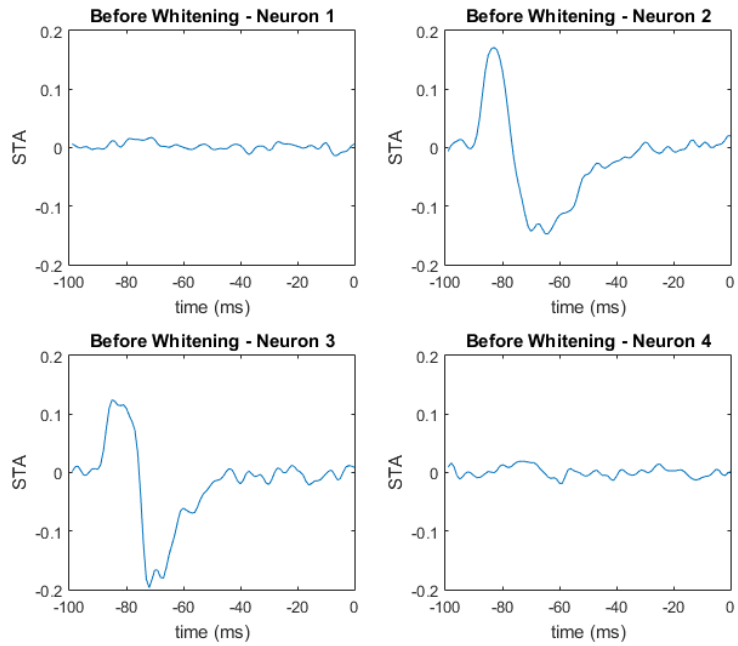


Bin size (ms)	Neurons following Poisson pro
10	2,3
20	1,3
50	1,2,3,4
100	1,2,3
200	2,3
500	NA

We see that neurons 2 and 3 consistently show a Poisson behaviour.

4. Spike Triggered Average and Pre-Whitening

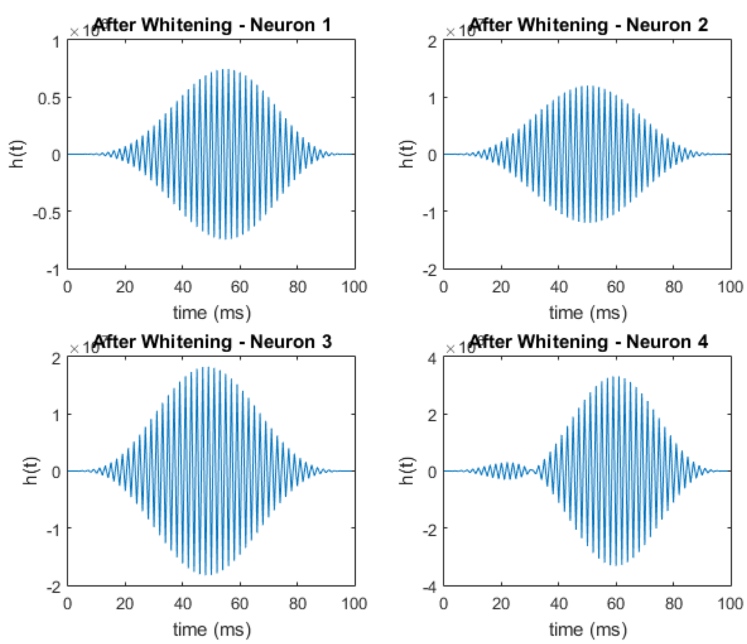
The STA is found by averaging out all the stimulus waveforms corresponding to output spikes for a 100ms before the spike.



If the stimulus were Gaussian distributed, we could have just reversed the STA in time to get the filter function h(t) (sans a constant factor). However, in our case, we need to perform pre-whitening and we have:

$$h = {C_{SS}}^{-1} C_{SR}$$

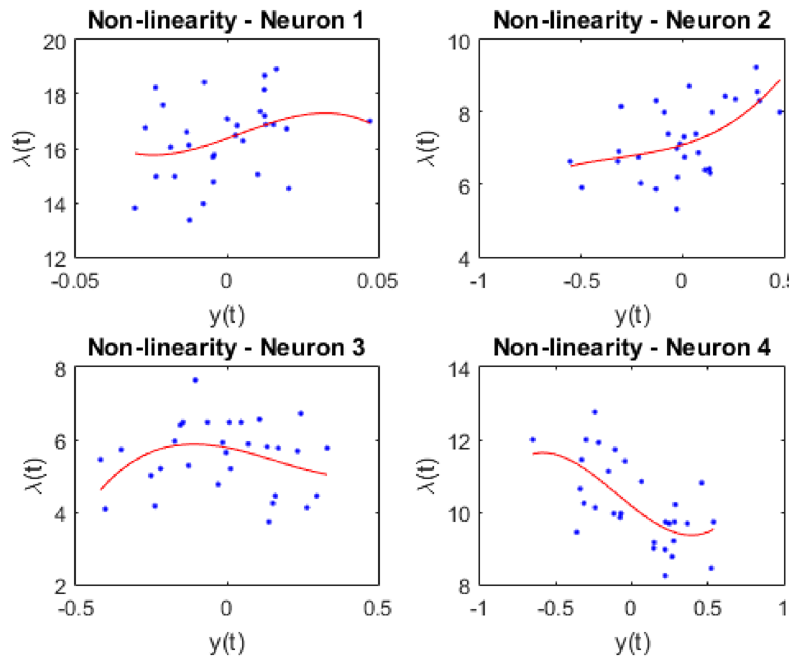
For calculating C_{SS}, we took segments of stimulus from 1:100, 2:101 …. 14901:15000. Then, for each segment, we calculated ss^T. Then the expectation of this matrix was taken over all the possible 100ms segments of stimulus s.



These filters perform low-pass filtering, as will be seen in part 6. Based on the autocorrelation result in part 1, pre-whitening has been done to the stimulus to correct for the non-Gaussianity.

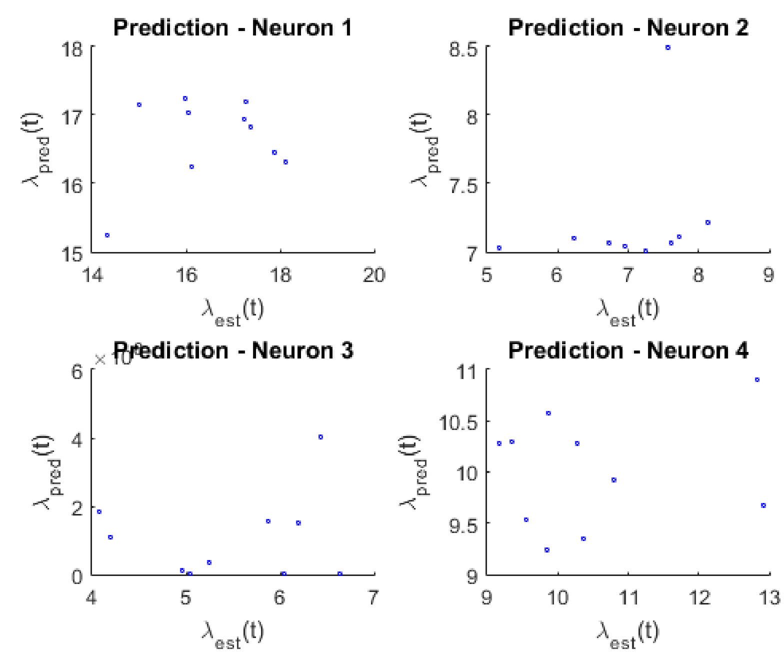
5. Determining the output non-linearity

For determining output non-linearity f(y), we simply use our x(t) (Stimulus) and h(t) obtained in the last part to get y(t). Then, a scatter plot between the PSTH (which is an ***estimate for λ(t) for neuron's 2 & 3 only***) and fitting it with a function will help us estimate the non-linearity. Note that in order to do this, we will have to bin the y(t) and λ(t) values.

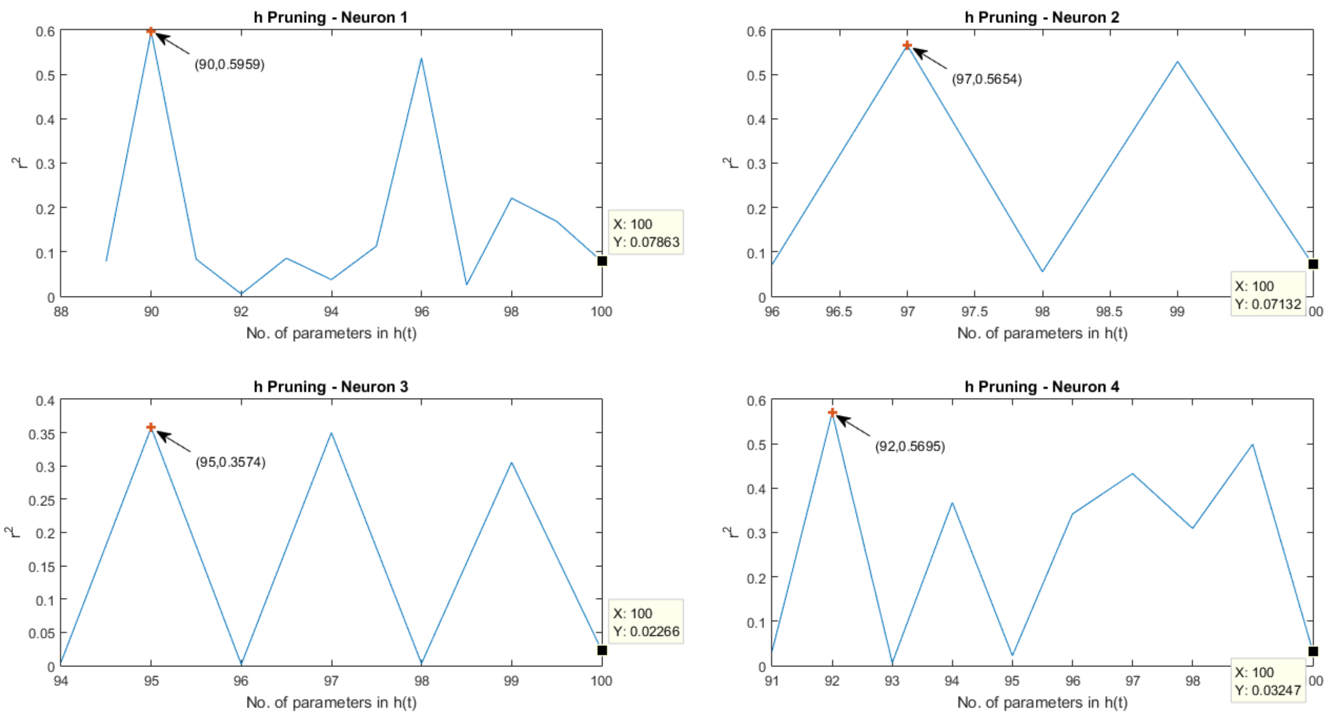


6. Prediction performance and filter pruning

For prediction, we use the estimated non-linearity and apply it to the last 5 seconds of the data to find the predicted λ(t), which is plotted against the estimated λ(t). This should ideally be a straight line and the *square of the correlation coefficient* (r^2) is a good measure of our prediction performance.

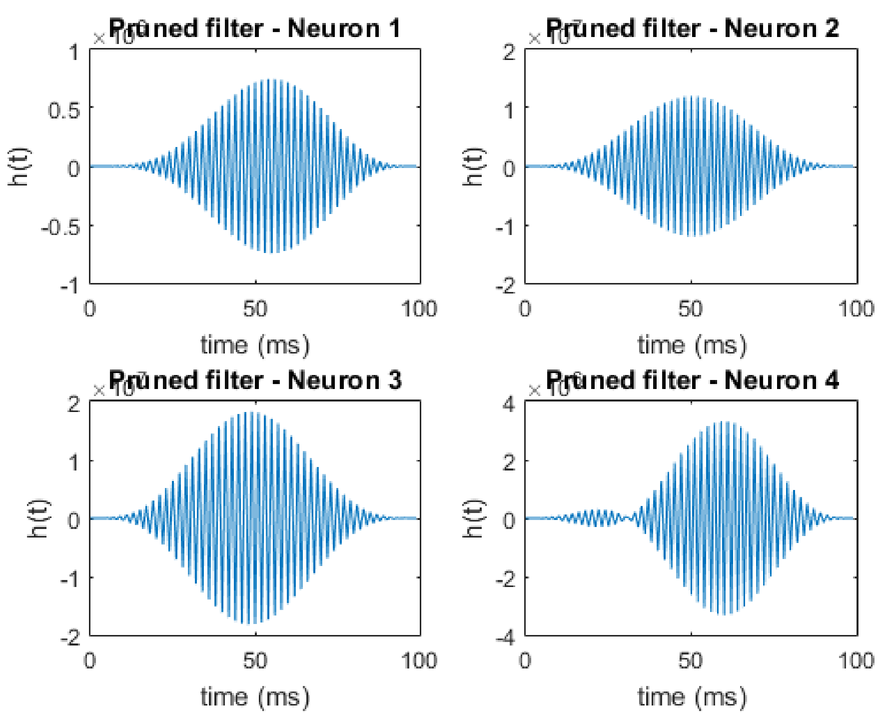


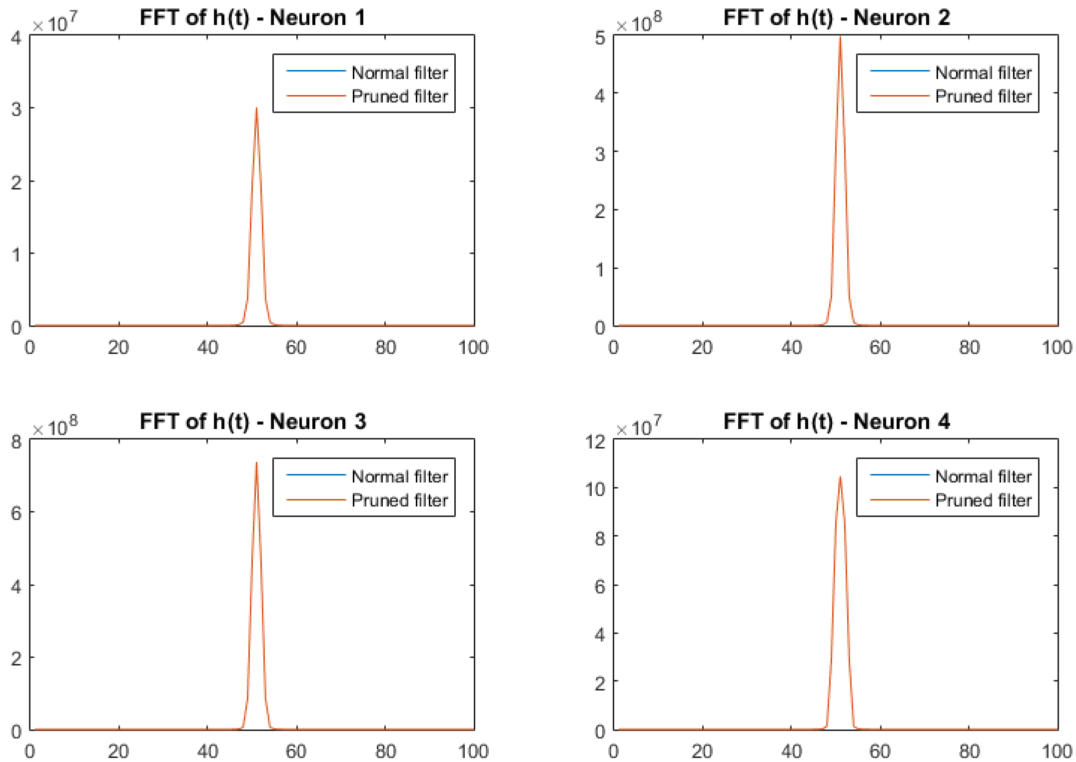
Now, the filter function obtained may not necessarily be giving the best prediction performance. So, we resort to filter parameter pruning, where we set the filter parameters to 0 one by one and see if there is any improvement in performance.



We see significant improvement in the r^2 value by simple pruning of the filter parameters. This method helps us improve prediction performance by mitigating errors in filter function estimation.

Here are the pruned filter functions and their corresponding frequency domain representations:





7. <u>Discrimination based on VP-SDM (Option B)</u>

For the problem of discriminating a set of stimuli (indexed by random variable X), we subject the neurons to the stimuli multiple times and get the recordings of the spike trains. In our case, we have 8 stimuli and 50 spike trains for each. Now, using a distance metric, which is the Victor and Purpura Spike Distance Metric in our case, we determine the distance of any one particular spike train $\mathbf{S_{i,j}}$ with other spike trains in the same stimulus set (i=1:8 and j=1:50). The mean of this distance is taken to get $\mathbf{d(S_{i,j}-S_i)}$. Similarly, for the same spike train, we calculate this mean distance for all S_i's.

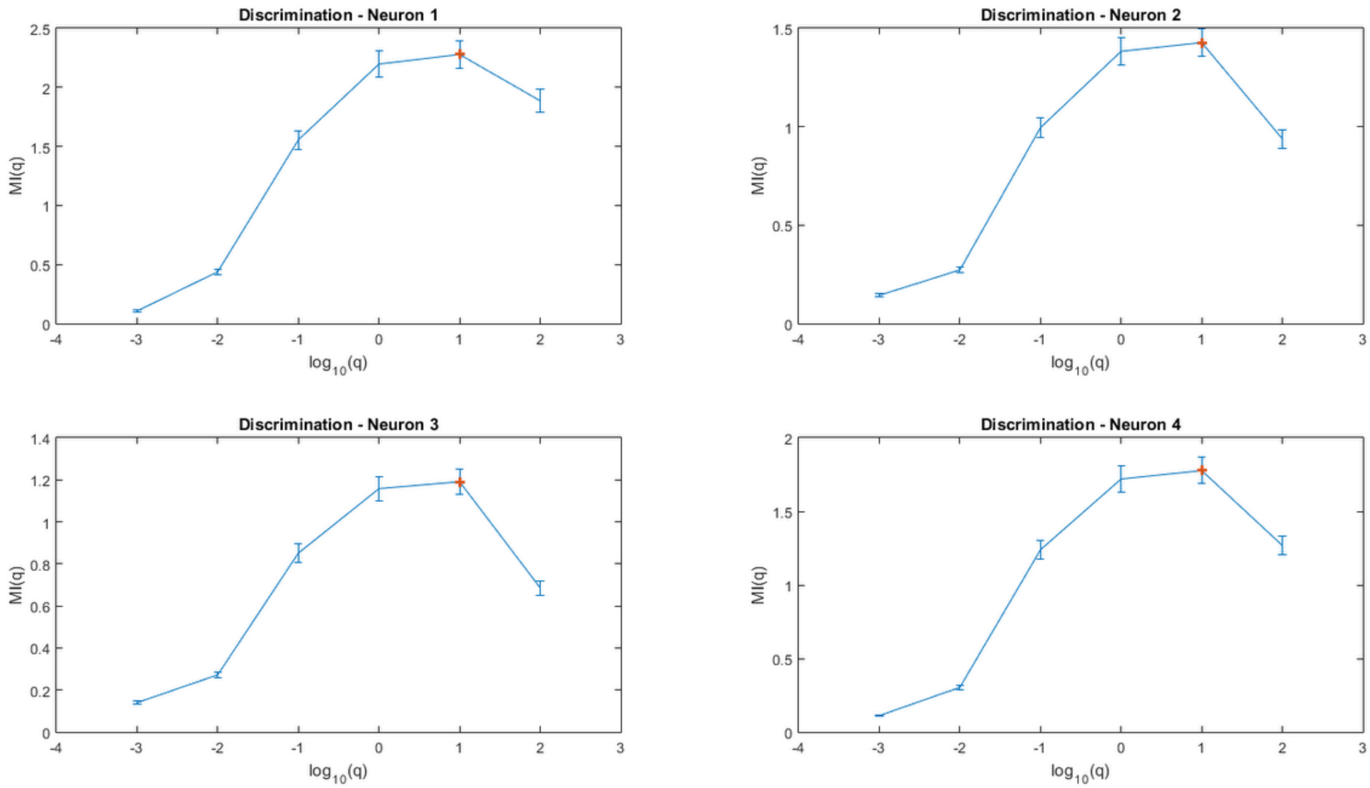
<u>Decision rule</u>: $\mathbf{k = argmin_i\ (d(S_{i,j}-S_i))}$

k is the index of the decision made by our detector (random variable Y), which has the same sample space as X. In essence, each (i,k) pair contributes to an increment in the **confusion matrix** between X and Y.

The above process is repeated for all the 8x50 spike trains to get the confusion matrix. From this, we use the entries of $P_{Y/X}(y/x)$ to obtain the mutual information I(X;Y). The assumption made is that *all the stimuli are equally likely*.

This mutual information, for a given cost in the VP-SDM, is an indicator of how well the stimuli are discriminated by the neuron. We make this value stimulus independent by repeating the experiment for 8 different stimuli a 100 times.

For each neuron, a plot of $\mathbf{I_q(X;Y)}$ is made (with 90% confidence)], which gives us optimal q values for discrimination. Since the cost for shifting (q) is inversely related with time, 1/q is a measure of the optimal timescale of neuron operation for discrimination.



We see that **$\log_{10}(q)=1$ ie. q=10** is the optimal cost for all the neurons 1-4. The actual value can be found by taking finer gradations. Therefore, the optimal timescale of neuron operation is about 1/10s, ie. of the order of **10ms**.

Now, in the cases where STA has good prediction performance, we say that the 100ms average history is good enough to ‘encode’ the stimulus. In other words, the neuron only needs to remember 100ms worth of data. In this part, we have taken a circuitous approach to show that the optimal timescale of operation for decoding (discriminating) different stimuli is of the order of ~10ms. Both of these are related as the timescale at which the neuron operates also decides the amount of memory that the filter function has.