# Week 26/01 -02/02

## Todo

* Explore L+N behavior
  + Define properly 1st layer, introducing amplitudes and widths
  + see other weights distribution
* Finish (and write) behavior of covariance matri
  + First layer define parametrically and behavior of spectrum
  + other distribution for W
  + read Babad Sompolinski paper
* Retinal data:
  + talk with roland
  + relook at Felix paper + Botond paper
* Salinas paper + noise?

## PCA and Covariance Matrix

Let's consider a discretization of our stimulus x in points. We consider the matrix s.t. at each colum we have the response vector at the stimulus x: .   
These responses are multiplied by the random synaptic weights W to produce the responses of the second layer to the same stimuli: V=WU. The rows of V can be read now as the discretization of the tuning curve of the N neurons. As said before, these tuning curves are the realization of N independend gaussian processes of mean 0 and covariance matrix K. The covariance matrix can be computed as the overlap between the tuning curves. Indeed, for a single tuning curve:

$$K\_{xx'} = <v\_i(x)v\_i(x')>\_W = <\sum\_{jj'}w\_{ij}w\_{ij'} u\_j(x) u\_j'(x')>\\
= \sum\_{j}u\_j(x)u\_j(x')\qquad\text{(1)}$$

If tuning curves of the first layer are translational invariant . Assuming L large enough, we can replace the sum with an integral, obtaining the correlation function of the gaussian process

$$K(x,x') = \sum\_j u(j-x)u(j-x') \simeq \frac{1}{L}\int dc\_j u(x-c\_j)u(x'-c\_j)\\
= K(|x-x'|,\sigma^2)\qquad\text{(2)}$$

In our original formulation, where ,

Now let's consider the N x n matrix V. Let's suppose that is mean centered (for simplicity).We can read the dimensionality of the coding manifold described in the activity space by the N neurons, computing the eigenvalues of the matrix It's element are the correlation between the tuning curves of the N neurons, that is  
 . For example, it is possible to relate the behavior of the eigenvalues to the actual dimensionality of the underlying manifold . A method proposed by Gao et al. , for example, propose the principal ratio

( measure of how many eigenvalues matters). Anyway, in our case computing the eigenvalue of such a matrix is hard. The entries are . Nevertheless, the eigenvalues are the same of the sample correlation matrix . Its entries are precisely the similarity of the response vector of the second layer to different stimuli : . This follow a Wishart distribution, of expected value the covariance matrix of the gaussian process K. Strictly, this is true only if . Otherwise the N bound the number of eigenvalues different form 0 to be n-N. Taking the limit of N large enough to capture the eigenspecturm of the true covariance matrix, the eigenspectrum of approach the on of K.

If the covariance function is translational invariant, its spectrum can be found simply taking the fourier transform. Example, if ,

,that is the case of gaussian tuning functions, then the eigenvalues scale as . We can obtain a power law eigenspectrum simply using a covariance function whose fourier transform is a power law. N=1000

Finite size effect.  
Studying the eigenvalue distribution of the matrix through random matrix theory is hard. If the covariance matrix is diagonal, then the eigenvalues follow the marchenko pastour distribution.

Distribution of widths  
Let's try to extend the previous theory to an heterogeneous layer of different widths. To mantain the translational invariance of tuning curves, we assume having a layer of L neurons arranged on the line with centers distributed equally spaced in positions (). On each center, we associate neurons of different charactersitics. In our case, we will choose gaussian tuning function with different gain and different widths:

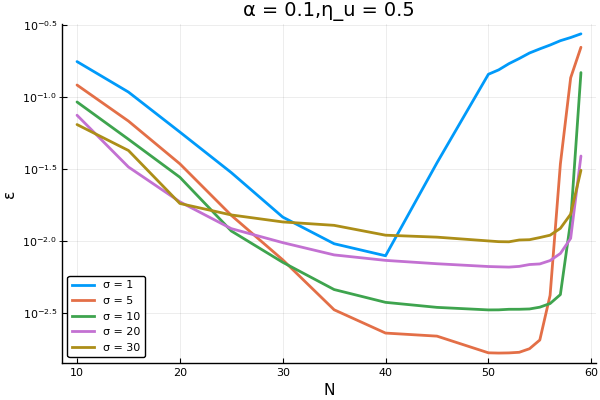
The kernel function become now where we integrated over j. The eigenspectrum can be found using fourier decomposition, and we obtain . Therefore, if we want to obtain a certain eigenspectrum , we have to solve the equation:

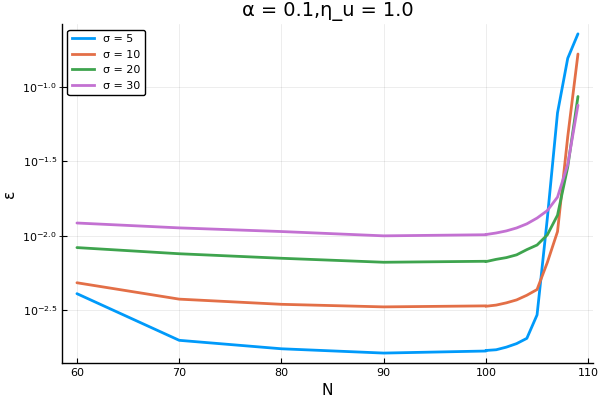
Writing the first layer tuning curves as , exploiting the convolution

## L+N

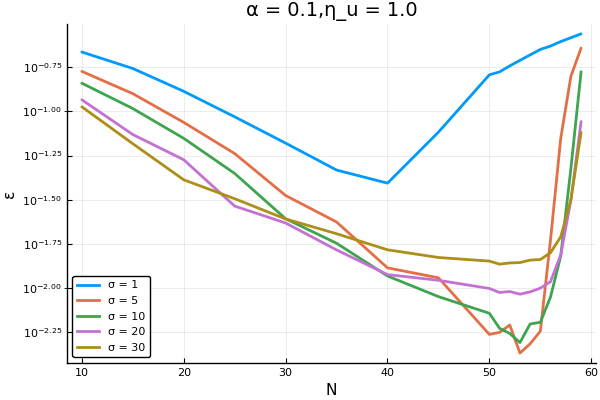
Let's fix a total cost C for the number of neurons: .   
Since and (actually the system is insensible to the mean of the firing rate), we can simply redefine

and use as the ratio between the cost of adding one neuron to the second in terms of neurons of the first one. Since L>>N, we will consider . Es setting will correspond to adding one N if we lower L of 10 units. Imaging all the neurons in the first layer to be the same (with the same noise), we can compute the tradeoff between L and N. Intuitively, increasing N should be advantegeous until we hit the limit set by the first layer tuning curves, that is , where is the variance of the noise of the first layer. This noise will correlate the noise in the second layer, giving a noise correlation matrix where ./ is intended by elementwise division.  
Having many parameters to explore, we make a selection. We fix to be low, and moderately high (high limit on info coding of first layer).

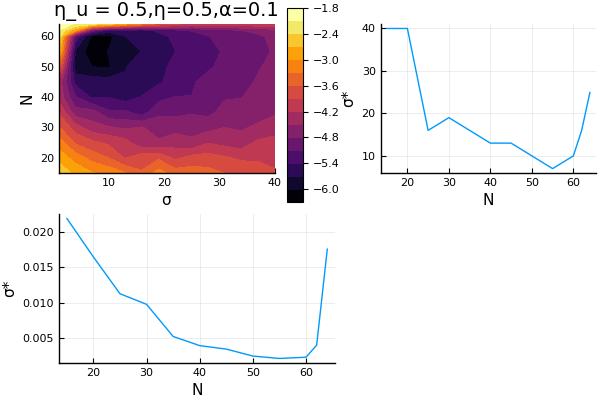
  
LvsN for starting value of L=500. For high we have an higher bound



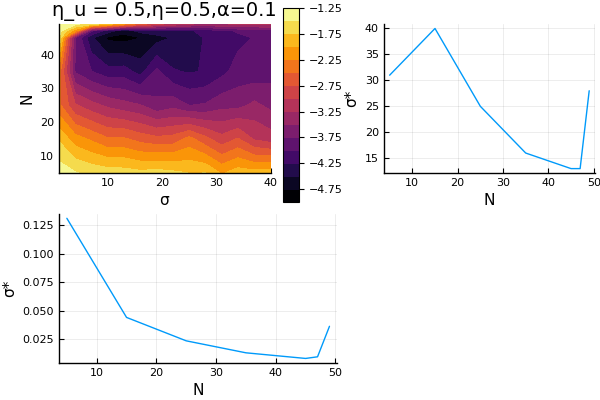
wi

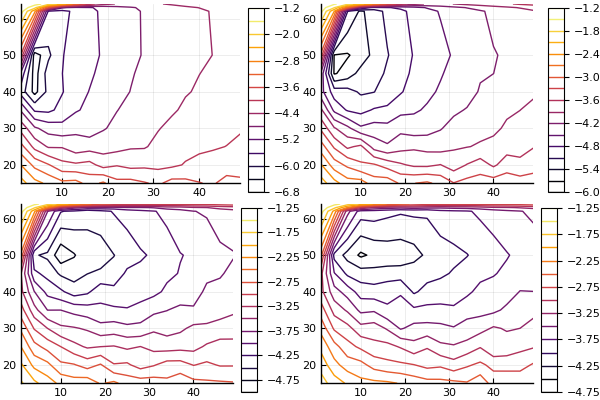


How does the optimal sigma change? We can try to explore this situation keeping fixed



Increasing the cost, we move the optimal sigma to higher values.



Increasing the noise on the first layer, we simply move the optimal sigma to higher values: for fixed value of : 

# Week 03/02 - 09/02

* Covariance matrix: look at distribution of performance
  + Energy function?
  + Longer gradient descent?
  + Read methods of bethge + variance of posterior (try analytics)
* Implement working memory model

## Retina and smooth pursuit

Look for phenomena related to smooth pursuit that can be investigated similarly to Franke's paper.  
Idea: retinal motion for analog conditions and see which effects are due to retinal signals and which  
to motor control (upstream process). E.g. inhanced sensitivity

## PCA and Covariance Matrix

Centers distributed equally spaced in positions (). On each center, we associate neurons with different tuning prop-erties. In our case, we will choose gaussian tuning function with different gain and different widths:

The kernel function become now where we integrated over j,assuming dense centers distribution. The eigenspectrum can be found using fourier decomposition: and we obtain . Therefore, if we want to obtain a certain eigenspectrum , we have to solve the equation:

If we have enough neuron and independent synapse, the spectrum is the same on the second layer.

### Finite m approximation

### Large m approximation

We can approximate the sum with an integral, rewriting and defining we obtain that the condition is This is analog to the problem considered in Harris' paper. If we assume a power law dependence of , we obtain the well known integral

where are determined by the probability distribution.

Otherwise, is a mess. Even if synapse are idnependent from neuron to neuron: The covariance function is now . If

## L vs N

Why the N remain constant while change? Actually we have a very flat minimum in that zone.

This is due to the fact that in that regime, increasing L increase the information contained in the first layer, while increasing N increase the ability of the second layer to read it out. We can not write this balance analytically, but numerically computing the fisher information it clearly comes out.

## Optimize over a set of

We should choose a set of such taht we minimiwe the total error function. Solving this optimization problem is dififcult. Recall that we have to minimize the RMSE of the second layer. This can be written as a loss function

Where our extimate, that will be a function of the noise is the average over the posterior distribution:

since the noisy response is assumed to be gaussianly distributed Note that we are interested only in the case where the is not trivial, therefore, we cannot use the limit of N very large. Otherwise, in that case we could simply minimize the fisher information (this is valid unluckly for lots of computations).  
We know from numerical simulations that our function has a non trivial minimum when N is low.

# Week 10/02-16/02

### Todo

* Optimization for different
  + minimize over different and check the symmetry of the problem if the amplitude is the same
  + write multiamplitudes function in such a way that they are differentiable
  + check with numerics that breaking the symmetry (allowing different weights) break triviality of energy landscape
* Implement working memory model
  + Implement gaussian bump
  + Write using SDE in Julia: how to implement time varying noise process?
  + Implement working meomry
* Summer schools
  + https://ws2.mbl.edu/studentapp/studentapp.asp?courseid=MCN Woodshole
  + https://www.fens.org/Training/CAJAL-programme/CAJAL-courses-2020/CN-2020/Application-form---CCN-2020/ Lisbon
* If cosine tuning minimize motor errors, conciliate with a loss function for random projections

### Error

Problem: we want to minimize the error function that is defined stochastically.  
The error function indeed is the the Mean Square Error. For a given realization of the network, we have a noise model

The best estimator is given by the MMSE estimator, that is the average over the posterior:

One could do some analytics and show that the loss function can be described as

Where

The problem of computing the error can be reduced to the computation

For low N we also have to average over the distribution of synaptic matrix W, since the minimum can strongly depends from this. The problem lies in the high dimensional distribution of r.

A possible solution could be the following:  
define the loss to be the montecarlo extimate of the RMSE:

$$\mathcal{L} = \epsilon(\sigma; W) = \frac{1}{N\_t} \sum\_t \int\_0^1 dx (\hat{x}(r^t) - x)^2\\
\hat{x}(r^t) = \frac{\sum\_m x\_m h\_m(r^t)}{\sum\_m h\_m(r^t)}\\
h\_m(r^t) = p(r^t|x\_m) = exp(r^t v(x\_m)/\eta) \\
r^t = v(x) + \eta^t \qquad v\_i(x) = \frac{1}{Z\_i}\sum\_j w\_{ij}u\_j(x)\qquad\text{(17)}$$

This function is fully differentiable numerically .

Then, the problem could be solved with SGD at each trial. At each step of the algorithm

In 1D it seems to give good results, at least comparable with grid search method.

# Weeks 17/02 - 29/02

## Todo

* Summer schools application
* Sharpee and Sompolinsky paper
* Rewrite the code
  + solve the problem of differentiation with uncostrained nulbe rof neurons/population: how to differentiate steprange
  + switch to train in flux?
* Write code for differentiation
  + random configuration
  + minimum in unconstrianed space

## 2 widths

With two widths, if we keep the same normalization coefficient, the resulting tuning curves  
are simply the superposition of two independent gaussian processes. Indexing as

Usually the solution is quite trivial due to the symmetry of the problem.

To break the symmetry we should require that

Question: which constraints we should imose?

Define two ensembles of populations of neurons. Respectively made up by neurons.  
The total number of neurons is . Each population is allowed to have a gaussian tuning function where the centers of each population are arranged to tile the space uniformly with We impose a constraint on the total neural activity of the first layer in the following way

$$\int dx \sum\_j^{L\_1} u^{(1)}\_j(x) + \sum\_j^{L\_2} u^{(2)}\_j(x) = R \\
L\_1A\_1\sigma\_1 + L\_2\sigma\_2A\_2 = R\qquad\text{(21)}$$

If we would have imposed the same constraint on the square of the tuning curves, we would have obtained

### Alternative constraint

Until now we imposed that for each neuron's tuning curves, the variance of the responses is =1. This constraint strongly depend from the realization of the random weights. An alternative way would be imposing that ON AVERAGE the standard deviation of the response is constant. Let's define , ,

Until now we imposed

$$\int dx [v\_i(x) - \int dx v\_i(x)]^2 = \int dx (\sum\_j w\_j u \_j(x))^2 - \alpha^2 \sum\_jw\_jw\_{j'} = 1 \\
\sum\_j w\_jw\_{j'} \int dx u\_j(x)u\_{j'}(x) - \alpha^2 \sum\_jw\_jw\_{j'} = 1\qquad\text{(23)}$$

This of cours eimply that A is dependent from the realization of the random weights for each tuning curve. An alternative is imposing that the average over the tuning curves is constant, that is, if

For a single tuning curve, this mean that

$$A^2(\sqrt{\pi\sigma^2} - 2\pi\sigma^2) = 1 \\
A^2 = \frac{1}{\sqrt{\pi}\sigma - 2\pi\sigma^2}\qquad\text{(25)}$$

Of course this can be done in the case where there is no non linearity between the 1st and second layer. This normalization works( pay attention to finite size effects)  
Using vonMises, we obtain that . In this case

We can extend this normalization including more sub population in the second layer. Since every random weight is independent, the equations become, defining and

$$\frac{L\_1}{L}(\beta\_1 - \alpha\_1^2) + \frac{L\_2}{L}(\beta\_2 - \alpha\_2^2) = 1 \\
L\_1A\_1^2(\sqrt{\pi\sigma\_1^2} - 2\pi\sigma\_1^2) + L\_2 A\_2^2(\sqrt{\pi\sigma\_2^2} - 2\pi\sigma\_2^2) =L\qquad\text{(26)}$$

We can recognize the term , and the equatio can be rewritten

The problem is now in fixing .

Ultimately we have to fix a parameter which is the ratio between the amplitudes. Every other constraint can be rreduced to this.

# March 2020

## Todo

* Summer schools application
* Book Basel
* 2 widths problem
  + Make it fully differentiable and optmize over number of neurons
  + rewrite gaussian process with the proposed normalization
  + learnability of the function
  + Go on with the analytics
  + Numerical simulation in a intelligent way to show optimality in some cinfigurations
  + check network decoder, write it and write code for error curves
* wrtie global error, check with one sigma and then extend
* Random moving objects

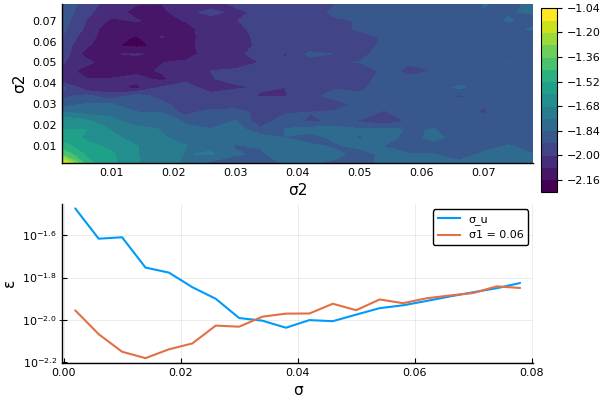
## 2 width problem

Summary:imposign the same constraint of before, we write the following equation for the first layer tunign curves.

an intuitive relationship could be relate the number of neurons in each population to their width , but still we have to imose a relationship between and . Introducing a constraint on the first layer does not add anything more, since still we will fix the sum of the neural cost to an arbirtary value (before it was sigma dependent),; which is equivalent to fix the ratio between the amplitudes.

Doing: simulation varying and the ratio between . question: does exist a point that does better then the single width? At which ratio?

We did the following: we vary , for a fixed unbalanced population of neurons. We also varied the ratio between their amplitude from 0.1 to 3. We aim to find   
1) if there is any configuration that achieve a better error (strongly)with respect to the single width.  
2) at which ratio of the amplitudes and at which pair of width.  
Surprisingly we find that the configuration that achieve the best error is at and when is relatively large and is small.



At the top is show the heatmap of the error for different width pairs, for the optimal c. At the bottom, the error is plotted when is held fixed at the high value and we vary

Problem: is a very strong result? Try to vary for fdfferent network realization and see if the minimum is always less, and not solely due to the high variance of the error in the regime of global errors.It seems to be very weak, as taken ON AVERAGE. In the sense that the same values of parameters are not always optimal for all the networks.  
Viceversa, it seems that, with every network realization, it exist a configuration with two widths that improve the error.

### Analytical intuition?

We can write the tuning curves of the second layer neurons as a superposition of two gaussian processes

where the covariance functions are given by

For simplicity we will assume . We denote as

the total variance of the response is given by , while the constraint impose that

The contribution coming from the large error, when is greater than ,can be estimated in the following way:

$$\varepsilon\_{g2}^2 = \frac{1}{6\sigma\_2^2} erfc(\sqrt{N"})\qquad\text{(30)}$$

where $N" = \frac{VN}{2(1+\sigma\_\eta^2)}$ and

This resemble the approximate scaling of the global errors of a single width. We also have another type of error, comprised between the two correlation functions. The typical size of this kind of error squared will be of order . To estimate its probability we can make the following assumptions. Let's say that if , we can indicate . That is we approximate every point inside the larger correlation length as constant, plus a perturbation that will be Gaussian with variance . The argument of the error function become therefore

using the rough approximation for the error function

Now, even if we assume that the probability of local error is almost 1, we have still that the probability of global errors are approximated and with a proportionality constant that may change depending from lot of things.

We made an extensive number of exmperiments, and overall they give very ambiguous results. We operate in the folowing regime: we compute the error for a , increasing gradually c, the contributions of , trying to lower the local errors. What we obtain is incredibly variegate and depends from the realization of the network.

fig:double_width_mhigh1double_width_mlows1double_width_mhigh1double_width_mlows1double_width_lows1

The problem was that the minimum is very noisy and therefore changing simply random weights can improve the situation or worsen it.

# April 2020

## TODO

* 2 widths problem
  + Make it fully differentiable and optmize over number of neurons
  + rewrite gaussian process with the proposed normalization
  + learnability of the function
  + Go on with the analytics
  + Numerical simulation in a intelligent way to show optimality in some cinfigurations
  + check network decoder, write it and write code for error curves
* Random moving objects: start to think a project
* Finish the paper

### Machine Learning view of our problem

A 1D latent variable is supposed to generate a set of N responses , described by a random realziation of a gaussian process. The covariance function is assumed to be s.e., s.t. noise free

.

We access only a set of noisy measurements of , namely . How we can perform regression on this noisy response to learn the decoding function , knowing a set of p training point . How the rapidity of learning depends from ?

Generally it is possible to relate the learning curve to the spectrum of the covariance function for data without input noise. How input noise affect the learnability of the function?

Our aim is to learn the function , possibly generalizing to noisy variations of the input.   
Let's consdier a feedforward architecture with an infinitely wide layer, minimizing the loss function through gradient descent is equivalent to kernel regression with the neural tangent kernel (NTK). Now, we would like to relate the content in frequency of , and possibly the input noise affecting the measurements, to the generalization error

# May 2020

## TODO

* Multiscale problem
* Decoder
* Random moving objects: start to think a project
* Write a table of comparison between different approaches of coding w tuning curves

## Multiscale problem

We analyzed the problem of encoding the 1D variable with 2 widths . We related the amplitude of one tuning curves to the other . We then fixed the variance of each tuning curve to be EXACTLY 1. We split evenly the number of cells between the two population . We also replicated the connecvtivity matrix, such that , such that when sending , , for all pairs . In order to see if introducing a lower width we reduce the error, we did the following simulations. For a given , we compute the error curve in function of when . In order to restrict the hige spae of parameters to explore, we did the following analysis. We fixed and we computed the error curve, finding . We then fixed a and we analyzed how the error curves evolve changing .   
To see: how evolve. A predictable result is that it increases with c to counter the effect of low . We would like also to say something about the error at the optimal width. Generally, if the error at the optimal width decrease or not, depends from the specfic realization of the random matrix. Anyway, it is possbile average over it. If the average show that there exist an improvement, than is a good sign. Since we are considering the MSE and generally it si very low, we worked on the following measure: , that is how the addition of improve the optimal error. The following plot show how the optimal and the error evolve introducing a , for 4 different networks.



We can see the different wrt the case of very low fig:

## Coding with tuning curves

Wrt to classical measures, Yarrow and Butts introduce Stimulus Specific Information which is defined by

. is also called response specific information and depends on the specific response

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Paper | Stimulus | Encoding function | Noise model | Loss function | Constraints and pop size | Parameters explored | Results | Methods Comments |
| Tuning curves:to sharpen or broaden. Zhang-Sejnowski-1999 |  | Radially symmetric with constant gain with centers uniformly distributed | Generalized Poisson + Noise correlation impact | J | Fixed peak activity and neurons/preferred position such that allow continuous approximation | Width | Fisher Info scale as | Sum over neurons -> sum over stimuli -> integral |
| Brunel-Nadal 1999 |  |  |  |  |  |  | Relationship between FI and lower bound to MI |  |
| Statistically efficient estimation,Pouget -1998 , (Partially NarrowvsWide-Pouget 1999, PopCodes-Pouget Rev 2000) |  | Von Mises | Poisson and iid gaussian | Peak of activity of a RNN | Weights of RNN set equal to tuning curves of input(or at least same form) | Difference in amplitude and width between RRN and Input | Show that a plausible RNN act as a ML decoder | Dependence of the estimator from the difference in width between decoder and ecndoe |
| F and Shannon Info in finite neural pop, Yarrow 2012,Tuning Curves, neuronal variability-Butts, Goldman 2006 |  | Von Mises (fixed amplitude) | Multivariate gaussian with variance proportional to firing rate . Three correlations: diagonal, expdecay, uniform | MI (SSI , , J ) Discrimination tasks | Fixed width and peak | Population size. Background firing rate, noise level-time of integration | MI and both grow with logN even at small popsizes. Their difference is consistent at small population sizes. Difference between MI and J optimal coding at different noise regimes (from flank to peak coding). Pop size determine the transition from peak to flank coding | Numerical estimation of MI |
| The infleunce of PopSize , Yarrow 2015 | , not uniform priors | VonMises, Montonic | Poisson | SSI,FI |  |  |  |  |
| Optimal neural rate coding leads to bimodal firing rate distributions-Betghe 2003 |  | Fisher Optimized (parabolic and step function) | Poisson | MSE | Max and min firjg rate | Integration time | Discrete to contniuous optimality reducing time window | Numerical integration for MSE |
| When Fisher Information fail, Bethge 2002 (Reassessing Optimal codes... -Berens 2012) |  | Monomodal (several considered) | Poisson | MSE | Max firing rate and others | Width, integration time , also tuning function | Optimal width depend on time of integration FIsher optimal tuning function can be non optimal wrt MSE Analysis of fisher optimal codes | Numerical integrtion of MSE and analytical computation of FI for different tuning curves |
| Tuning curve sharpening for orientation selectivity, Series |  | Feedforward model with recurrent conections of retina lng-V1 | Poisson | Information |  | Sharpening vs non sharpening | Sharpening introduce correlation that are not beneficial |  |
| How Behavioral Constraints May Determine Optimal Sensory Representations,Salinas 2006 |  | gaussians and monotonic, heterogeneity allowed | Gaussian with variacnce dependent from the mean | Approximation of downstream motor function |  | Width and shape for different function approximation | Shape of optimal tc depends on behavioral constraints |  |
| Efficient coding heterogeneous population, Simoncelli 2014 | with non uniform prior | Monomodal (no specification) | Poisson | J (and related quantities, but not MSE) | Mean total firing rate (gain), number of neurons (density) |  | Density proprotional to prior, constant gain | Define a convolutional population and then transform it according to some change of parameters coupling density and width. FIx fisher info for the convolutional population (that, implicitlty, depends on the width, here is the hack) |
| Implications of Neuronal Diversity on Population Coding- Sompolinski 2006 |  | VonMises + fluctuation in the parameters | Multivariate gaussian | J |  |  |  |  |
| Error-based analysis of optimal tuning functions, Meir 2010 | , gaussian prior | Gaussian with homogenoeus width (or diagonal covariance matrix in the multivariate case) | Poisson | MSE |  | Width, integration time, variance of the prio | Existance of optimal width that decrease with increasing integration time | Write MMSE as a gaussian estimator allow to write MSE in a closed form. Still it is assumed that the number of neurons and width is "sufficiently high" to cover the stimulus space |