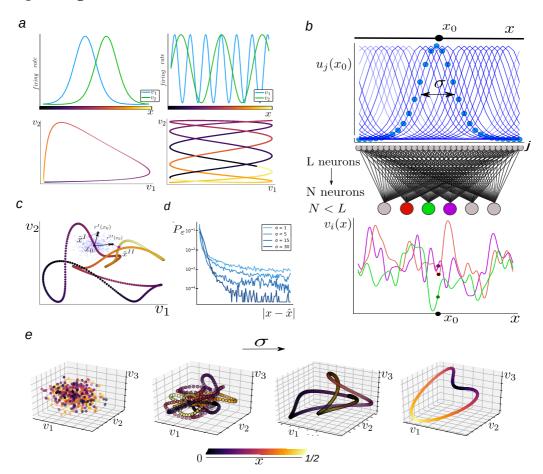
Structure

Introduction

- Tuning curves are often assumed to have a regular shape. Many studies are made analyzing populations of neurons which are responsive in a localized region of the stimulus space. (E.g. DS cells, Place Cells, ecc...)
- Nevertheless data shows that in deeper regions of the brain, neurons does
 not always have a clear selectivity profile. A first example are grid cells.
 Another example is given by tuning curves which are not consistent with
 parametric shapes: data from Lalazar Abbott shows that the brain may
 employ random connectivity to transmit information from one region to the
 other, giving rise to irregular tuning curves.
- Intrinsic difference between localized tuning curves and multi-peaked ones: ambiguity at single cell level, efficient use of phase space.
- We study the coding properties of a population of neurons whose tuning curves arise from random connections with a classical sensory layer. Balance between local accuracy and global noise robustness.
- Explain figure 1a



Results

Example with Discrete Stimuli?

Plots: L vs N?

Random Feedforward Network - 1D

We extend the previous reasoning to a more biologically relevant architecture, and this allow us to use continuous stimuli and explain the randomness of firing rates. We introduce a 2 layer feedforward network.

The first layer is a sensory layer, that encode a low dimensional stimulus into an high dimensional representation. Then, it transmit this information to a smaller layer with a random all to all connectivity.

Increasing σ we go from a random uncorrelated code to a smooth one (Fig.1 e)

Global and Local Errors

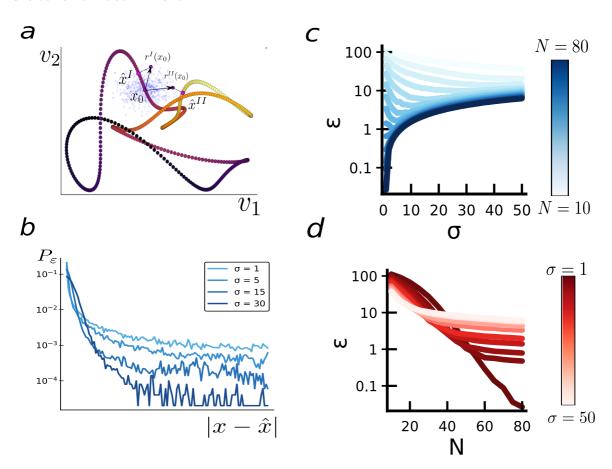
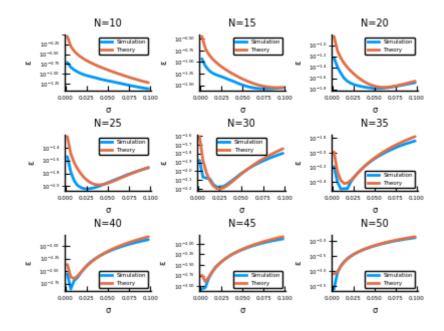


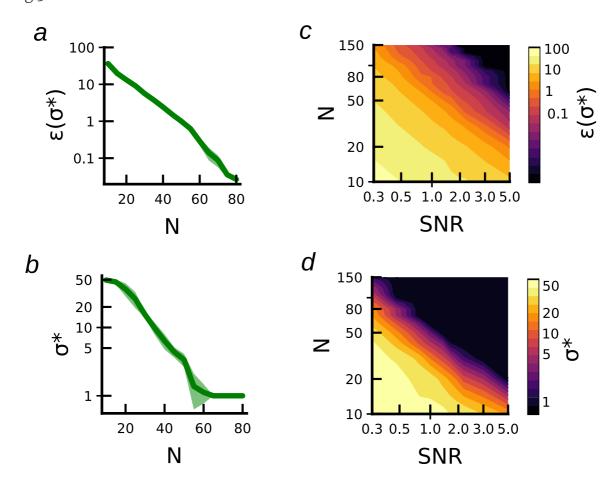
Fig2. Explain the concept of balance between the errors. Two types of error: global and local. Tuning σ we change the relative contribution (Fig.2 a-b)

Insert formulas from methods + fitting of the theoretical formula?



Scaling of optimal parameters with N and η .

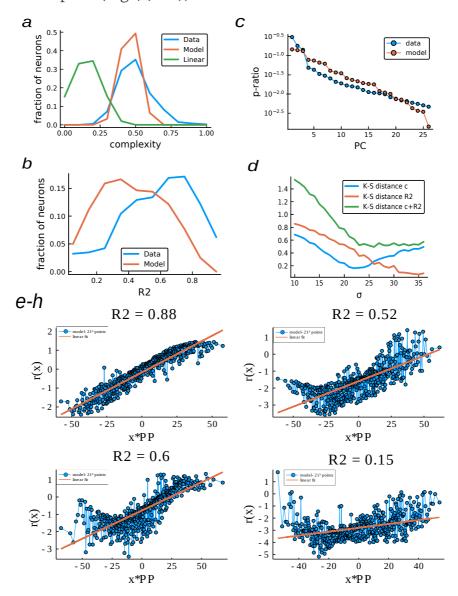
Fig.3



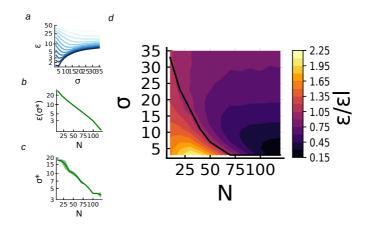
Extension to Multiple Dimensions (?)

Data Analysis

We know that a similar version of the model has been used to fit the data from Lalazar Abbott. We first test which parts of the data the simplest version of the model is able to explain (Fig.4 (a-d))

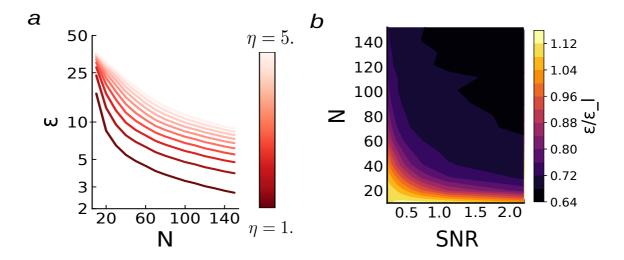


Then, we do the same analysis of before, looking for the scaling of the optimal σ . In this case we have also a theoretical prediction for the parametric shape of the tuning curves, so we can compare how much is advantageous for the system having random tuning curves or linear ones.



Finally, real data shows an high variance of the noise magnitude (insert data analysis?).

We fix the σ_t to be the parameter that better fit the data, and we vary the noise level, showing that the size of the population required for doing better than a linear code is quite big and comparable with the number of input a single muscle receive.



Decoder

Until now we used an optimal decoder. It exist a network implementation of this decoder that achieve the same result. It requires to know the true responses. Can we actually LEARN this type of decoder, especially in presence of global errors?

If we would have to learn the decoder, what is the optimal parameter for the encoding?

Methods

Continuous stimulus $x\in[0,1]$, array of traslational invariant tuning curves, with $c_j=\frac1L,\frac2L,\dots,1$

$$u_j(x) = Aexp(-\frac{(x-c_j)^2}{2\sigma^2}) \tag{1}$$

If translational invariance, we have vonmises tuning curves that has to defined on an interval 0-2pi. A general form is:

$$u_j(x) = Aexp(\frac{1}{(2\pi\sigma)^2}cos(2\pi(x-c_j)))$$
 (2)

The second layer mix the tuning curves with random synaptic weights

$$v_i(x) = \frac{1}{Z_i} \sum_{j}^{L} w_{ij} u_j(x) \tag{3}$$

choosing $w_{ij} \sim \mathcal{N}(0, \frac{1}{L})$, each tuning curve of the second layer is the realization of a gaussian process, where each pairs of response of single tuning curves

Global and Local errors

Let $\mathbf{r}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) + \eta$ the noisy response to the stimulus x, for a given realizatio n of W. We want to compute the Mean Squre Error

$$\varepsilon = \int dx dr p(\mathbf{r}|\mathbf{x})(\hat{\mathbf{x}}(\mathbf{r}) - \mathbf{x})^2$$

We can split the error function in two contributions :local and global. We will denote as the probability that r will cause a local error:

 $p(r_l|x)=p(r|x$ s. t. $(\hat{x}-x)<\sigma$), the same with the global error. We can write the error as the sum of two contributions, using the shortcut $\varepsilon(r_l,x)=(\hat{x}(r)-x)^2$

$$arepsilon = \int dr dx p(r_l|x) arepsilon(r_l) + \int dr dx p(r_g|x) arepsilon(r_g)$$

Now, for the local error we know that, using an efficient estimator, the error will be gaussianly distributed with a variance equal to the CRAO bound. Using the mean field approximation that the fisher information is the same for every point of the curve, we obtain: $(\hat{x}(r_l) - x) \sim \mathcal{N}(0, 1/J)$. Therefore we can write the local error as

$$\varepsilon_l = P(\varepsilon_l) \frac{1}{J} \tag{5}$$

where we defined the average probability of having a local error $\int dr dx p(r_l|x)$. For N very large, this will be the only contribution.

For the global error we can do the same reasoning, assuming that the error is uniformly distributed between $[\sigma, 1]$ independently from x.

If PBC, otherwise will be something slightly different since we have to keep into account that point in the middle has a lower error. Anyway, what matters is that when we integrate, and $\sigma << 1 \int d\Delta x_g p(\Delta x_g) \Delta x_g^2 = \bar{\varepsilon}_g^2$ is of order 1.

Then the contribution to the global error is given by

$$\varepsilon_g = P(\varepsilon_g)\bar{\varepsilon}_g^2 \tag{6}$$

where $P(\varepsilon_g)=\int dx dr p(r_g|x)$ is the average over the stimuli of the probability that r , given x, will cause a global error. Let's now fix x'. If we imagine having discrete stimuli and uncorrelated, we have an error when r if closer to v(x') than to v(x). The probability that this happen is (we denote as $|\cdot|^2$ the squared norm)

$$p(\varepsilon_{xx'}) = p(|r - v(x')|^2 < |r - v(x)|^2) = p(\sum_i (v(x) - v(x')_i^2 - 2(v(x) - v(x')_i \eta_i < 0))$$
 (7)

The probability of an error, given x, is the union probability of the events $p(\bigcup_{x'} p(\varepsilon_{xx'}))$. This is the probability that exist at least one x'. For a given realization of the network, this is hard to compute. Instead, we can average over the distribution of W and obtain

$$p(\varepsilon_g) = \langle \langle 1 - \Pi_{x'} (1 - p(\varepsilon_{xx'})) \rangle_x \rangle_W \tag{8}$$

We can assume that the probability of an error of a fixed x with any x' is i.i.d. over the distribution of W and we can substitute the average of the product with the product of the average (1st approximation):

$$p(\varepsilon_g) = 1 - (1 - \langle p(\varepsilon_{xx'}) \rangle_W))^n \tag{9}$$

We can compute the average of the probability of error between uncorrelated stimuli. Indeed, assuming the variance of the response is 1, the difference between uncorrelated stimuli is normally distributed $v(x)-v(x')=\tilde{v}\sim\mathcal{N}(0,2)$. We have to compute

$$< p(arepsilon_{xx'})>_W = p(\sum_i ilde{v}_i^2 - 2 ilde{v}_i\eta_i < 0)$$
 (10)

The distribution of $d=\sum_i \tilde{v}_i^2-2\tilde{v}_i\eta_i$ is a combination of chi squared distributions (for η =0 is a standard chi square distribution). We can compute its mean E[d]=2N and its variance $Var[d]=8N(1+\eta)$. (This is correct).

2nd approximation:

We can obtain a pessimistic estimation of the probability of error, assuming that d is distributed gaussianly with the same mean and variance.

Therefore the probability that d is less than 0 is $< p(\varepsilon_{xx'}) > = 0.5 erfc(\sqrt{\frac{N}{4(1+\eta)}})$.

This does not work.

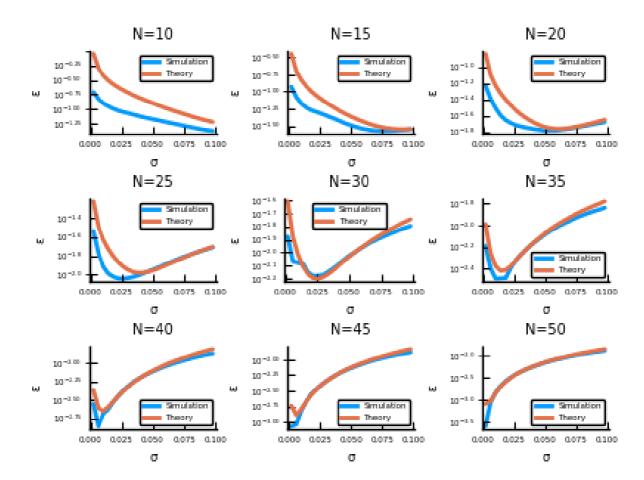
The distribution has a skewness higher than 0, and this procedure overestimate the probability of error. Indeed from numerical simulation, the behavior is well captured by the cdf of a gaussian with HALF variance

$$< p(arepsilon_{xx'})> = 0.5 erfc(\sqrt{rac{N}{2(1+\eta)}})$$
 . (This is surprisingly good)

Now, let's estimate the number n. This is the number of possible points that, fixed an x, can cause a global error. In the discrete case with uncorrelated stimuli, this is of course the number of stimuli L-1. If we start to correlate the stimuli, we can imagine to group the stimuli within a correlation length and consider a discret set of $\frac{1}{\sigma}$ uncorrelated cluster of points. If N is sufficiently high, we can use the 3rd approximation and obtain for the global error

$$p(\varepsilon_g) = \frac{1}{\sigma} \frac{1}{2} erfc(\sqrt{\frac{N}{2(1+\eta)}})$$
 (11)

This last formula is surprisingly robust and does not require any proportionality constant.



Discussion