Neural Coding

What is Neural Code - Techniques for recording from brain

- tools for discover how brain rep info

- models that express relation,

etc

Recording from ~~the~~ brain

fMRI - Recording someone's brain as they perform task

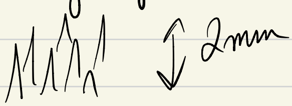
*Good for finding approx location (avg many neurons)

Faster: EEG, calc. electric field, faster + avg too

Reading Out ~~the~~ Neural Code

· multielectrode array

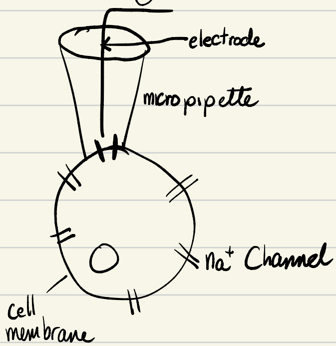
- good for when we can put layer over brain

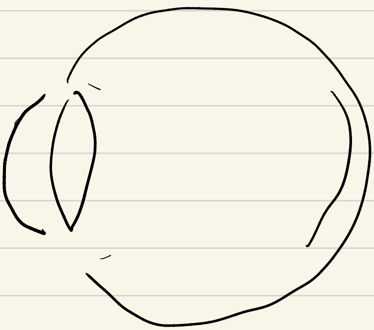


- calcium Imaging

↳ Calcium records neural activity since used in process.

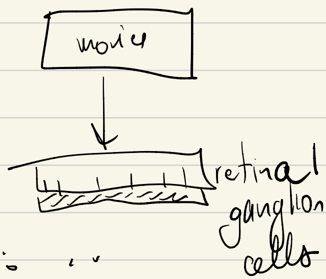
Looking inside single cells

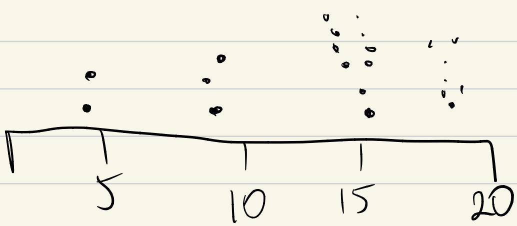




converts light
signal to electrical

model:





raster Plot

cells fire consistently at app times

suggests that ~~the~~ cell is responding ~~to the~~ particular feature of the stimulus that occurs at this time

Encoding: how does a stimulus cause a pattern of response
- building quasi-mechanistic models

Decoding: what do ~~these~~ responses ~~tell~~ us about ~~the~~ stimulus
- how do we reconstruct what the brain is doing

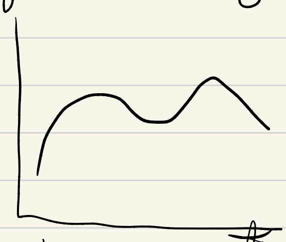
encoding: P(response | stimulus)
"being shown"
decoding: P(stimulus | response)

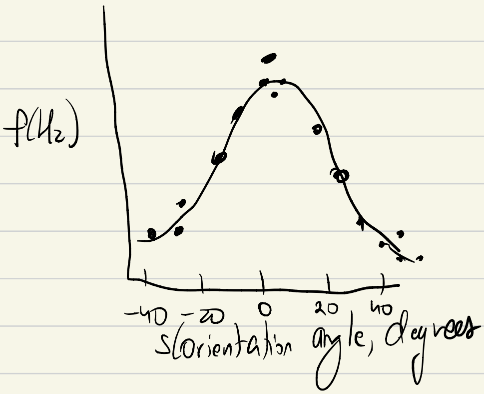
Q: What is the stimulus? What is the

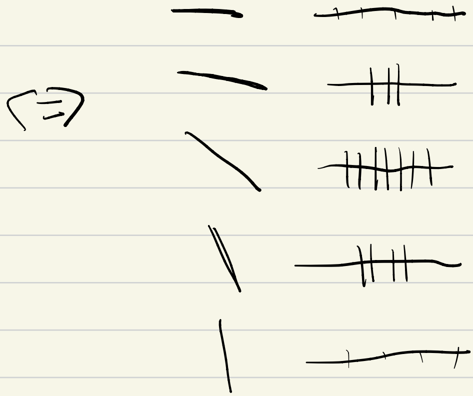
Neural Representation of Information

(avg proto rate of spike)

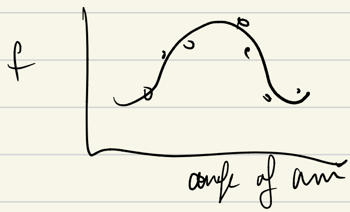
Neural Response







Monkey making arm movement



Simple Models

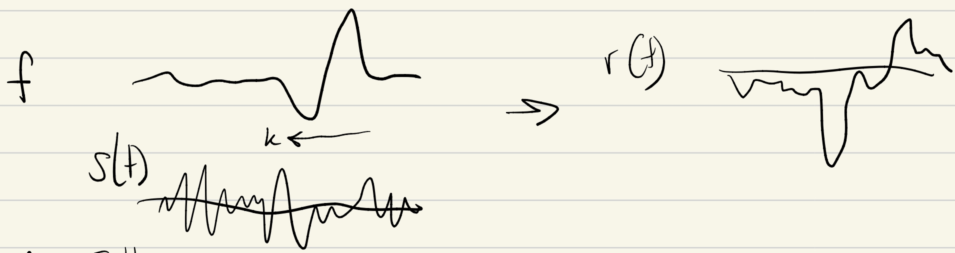
$P(\text{response} \mid \text{stimulus}) \to r(t)$ given a stimulus

for now, take response as spike in single neuron

$$r(t) = \phi s(t) - \phi s(t-\tau)$$

$$s(t) \to r(t)$$

· Temporal filtering



Linear Filtering:

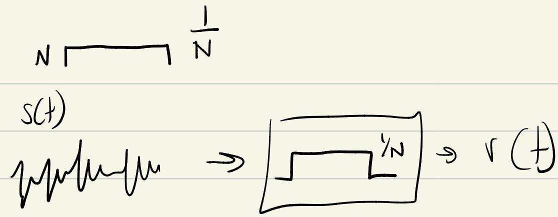
$$r(t) = \sum_{k=0}^{n} \underline{\underline{S_{t-k}}} \, \underline{f_k}$$

boradan hauf looking · weigth by factor dep on time

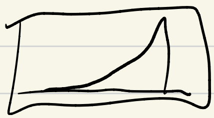
$$r(t) = \int_{-\infty}^{t} d\tau \, s(t-\tau) \, f(\tau)$$

$\underline{* \text{ Convolution}}$!

Ex 1: Running Avg:

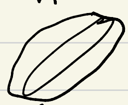


Linear filter: $r(t) = \sum_{k=0}^{n} S_{t-k} f_k$

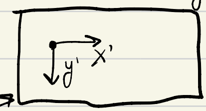
Leaky Avg:  exp. decay into past

Spatial Filtering

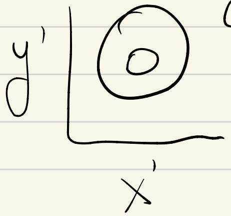

retina

V_1

visual Cortex

$r(t) = \sum_{k=0}^{N} \underline{S_{t-k}} f_k$ · Temporal filtering

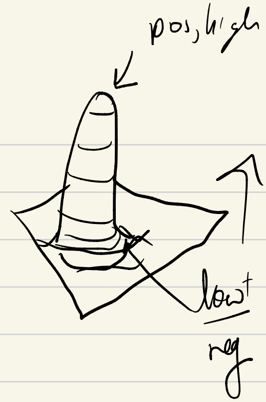
S(x,y)


$r(x,y) = \sum_{\substack{x'=-n \\ y'=-n}}^{n} S_{x-x', y-y'} \underline{f_{x',y'}}$ filtering



cartoon $f(x',y')$

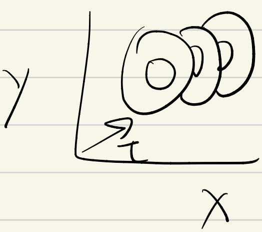
$f(x',y')$

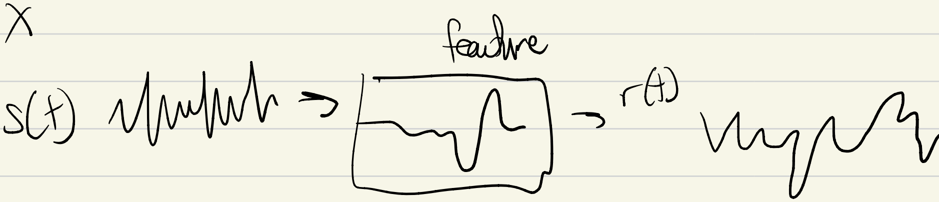


approx as diff by gaussians

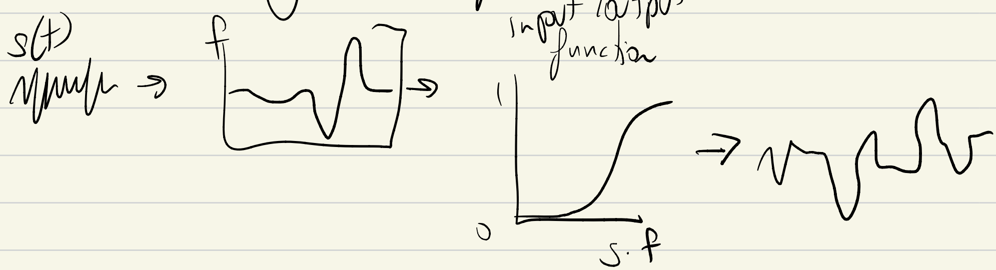
SpatioTemporal filtering

$$r_{x,y}(t) = \iiint dx'\,dy'\,d\tau\, f(x',y',\tau)\, s(x-x', y-y', t-\tau)$$





Finding Components



P(response | stimulus)

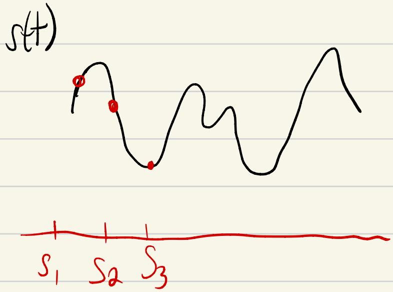
- Our problem is dimensionality!

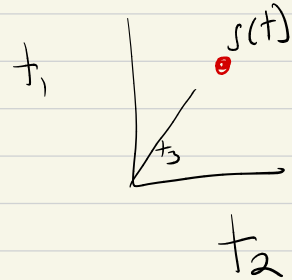
300,000 per stimulus, comp. too heavy.
- data unmanageable.

- We want to sample the responses of the system to many stimuli so that we can characterize what it is about the input that triggers responses

P(response | stimulus) → P(response | s_1)

- Start w/ high-dimensional description (image/ time-varying waveform) + pick out small set of relevant dims.





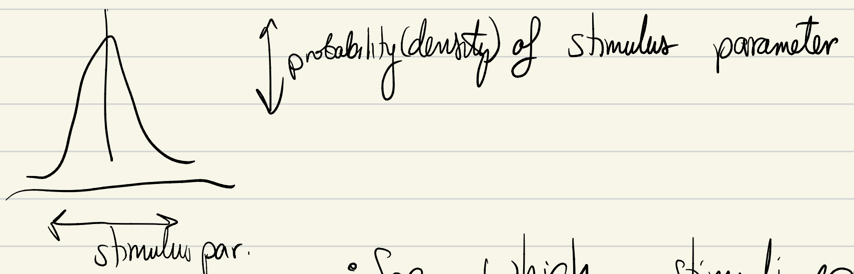
dimension becomes # of points in discretization

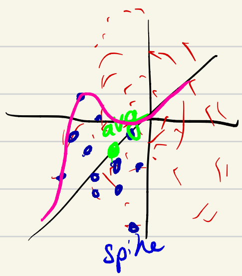
· first axis to stimulus @ first point in time, ~~second to~~ second, ...

$P(\text{response} \mid \text{stimuli}) \rightarrow P(\text{response} \mid s_1, s_2, s_3, \ldots, s_n)$

Useful method: Gaussian white noise

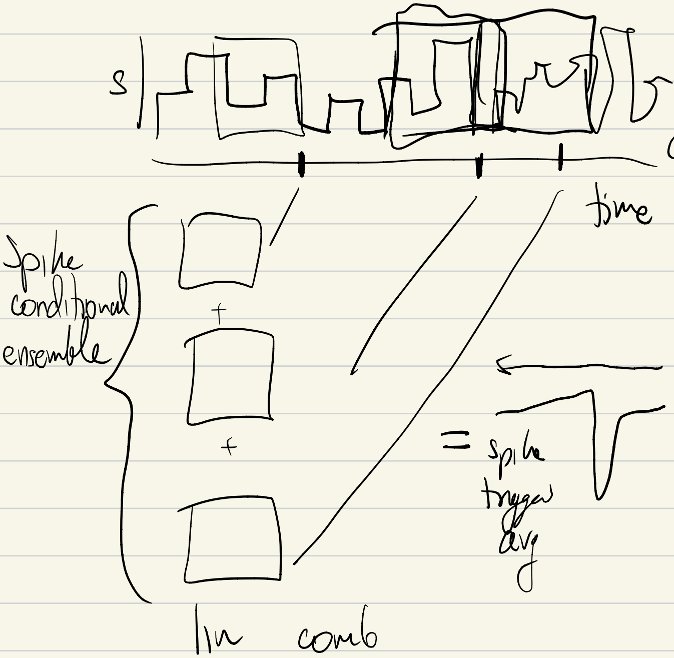
- as we continue to stimulate w/ various white noise, start to fill up distribution.
  - Gaussian prior stimulus distribution
  - * even along any axis, distribution gaussian
    - ↳ also along any lin. comb.





- See which stimuli contributed spikes to occur
- Get spike-conditional distributions
- Calculate spike-triggered avg, and the vector to ~~the~~ point is good vector orientation for axis

Reverse correlation: spike-triggered avg

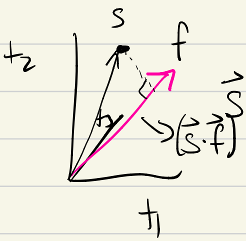


· look back @ time preceding spike + grab it

(tends to be normal looking)

* Finding what's common to illicit a spike

Linear filtering = convolution = projection



$\vec{s}$ projected on $\vec{f}$

$(\vec{s}\cdot\vec{f})$

f_1 feature

s(t) ⟶

?

input/output fnct

· Determining nonlinear input/output function

input/output function is:

$$P(\text{spike} \mid \text{stimulus}) \rightarrow P(\text{spike} \mid s_1)$$

- Found from data using Bayes's Rule

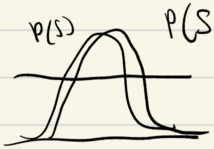
spike conditional dist

$$P(\text{spike} \mid s_1) = \frac{P(s_1 \mid \text{spike}) \cdot P(\text{spike})}{P(s_1)}$$

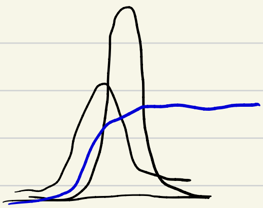
← prior

ex : Say neuron fires @ random,

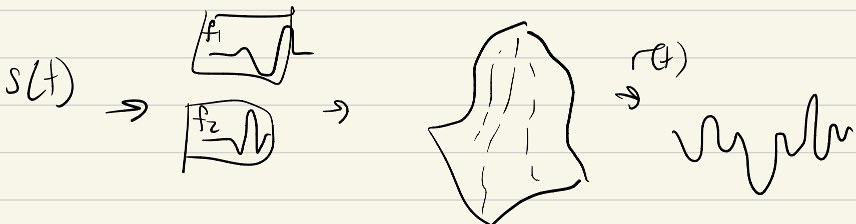
$$P(\text{spike} \mid s_1) = \frac{P(s_1 \mid \text{spike})\, P(\text{spike})}{P(s_1)}$$



want to see:

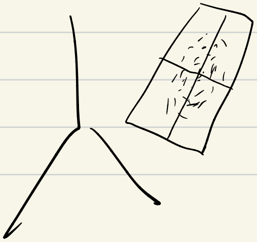


* Consider several filters



$$r(t) = g(f_1 \cdot s,\ f_2 \cdot s, \ldots,\ f_n \cdot s)$$

Covariance + PCA : discover low-dim structure

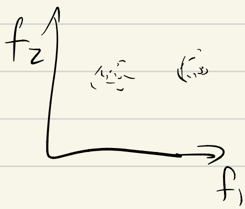


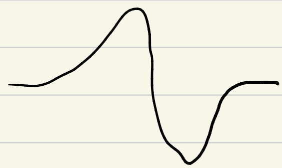
Two principle components : 2 dimensions
perp.

* gives new, sig smaller,
basis set

ex: most faces can be reconstructed from like 8 'eigenfaces'

- spike sorting





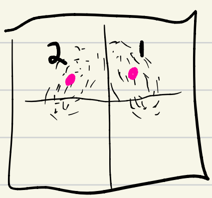
noise gets put into set w/ two features

PCA: ① gives representation of data w/ lower dim

② finds vectors w/ maximal variation of data

1 cell

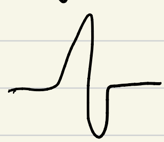
proj f_2



avg 0, but not our story

proj. featr 1

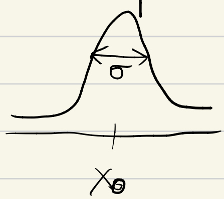
1:  "on"

2:  "off"

neuron likes when light off & light on.

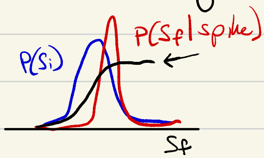
Variability

Magical Gaussian $P(x) = A\exp[-(x-x_0)^2/2\sigma^2]$



$\langle x \rangle$
$\bar{x} = \text{mean}(x) = x_0$
$\text{var}(x) = \langle (x-x_0)^2 \rangle = \sigma^2$

*You know you found a good feature if input/output curve over your variable looks interesting



* Want two curves as different as possible
⇒ Kullback-Leibler divergence

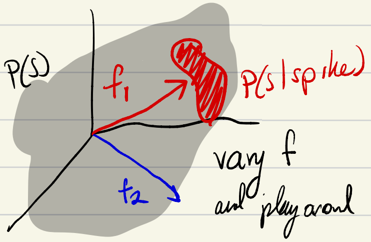
$$D_{KL}[P(s), Q(s)] = \int ds\, P(s) \log_2 P(s)/Q(s)$$

Goodness measure $D_{KL}[P(s_f|\text{spike}), P(s_f)]$ maximal

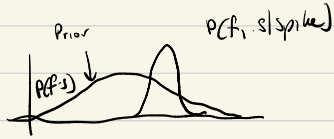
*Maximally Informative dimensions

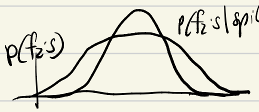
Max. Inform. Dim cont.

Choose filter in order to maximize D_{KL} b/w spike conditional & prior distributions



vary f and play around





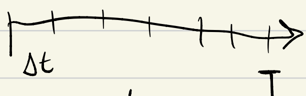
(more similar, worse filter)

*Does not depend on white noise
* Can be used for deriving from natural stimuli

Advantages: • Gives a way of seeking filters that maximize discriminability of spike-conditional dist. & prior
• does not req. a specific structure for distributions such as Gaussian

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

• Bernoulli Trial
- take biased coin w/ small prob. of heads (say a spike occurs whenever lands heads)



$n = \frac{T}{\Delta t}$

p (prob firing)

t

$1-p$ (prob ! firing)

Distribution $P_n[k] = ?$ → $\binom{n}{k} p^k \cdot (1-p)^{n-k}$

Mean $\langle k \rangle = ?\ n \cdot p$

Var $Var(k) = ?\ np(1-p)$

$r = \frac{\rho}{\Delta t}$

Poisson spikingDist: $P_T(k) = (rT)^k \exp(-rT)/k!$

Mean: $\langle k \rangle = rT$

Var: rT

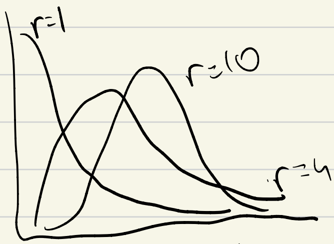
Fano Factor: $F = 1$

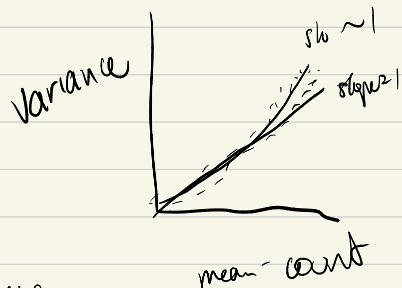
Interval dist: $P(T) = r\exp(-rT)$

$$P_T(0) = \frac{(4T)^0 \exp(-4 \cdot 1)}{0!}$$

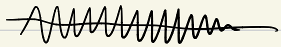
$P_T(k)$ = probability of k spikes in T seconds

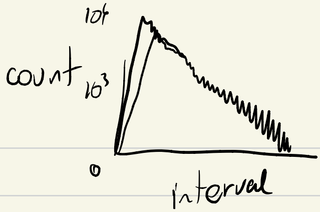
r = # spikes / second





*these as r changes w/ Time

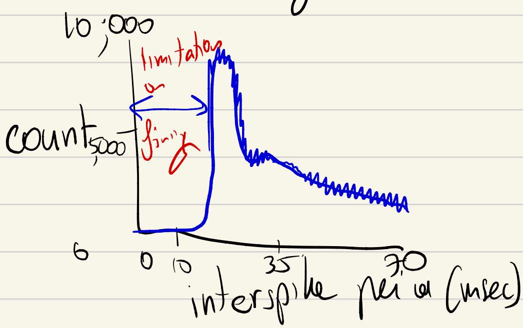




- When is a Poisson model poor:

When a neuron fires many hundreds of times per second

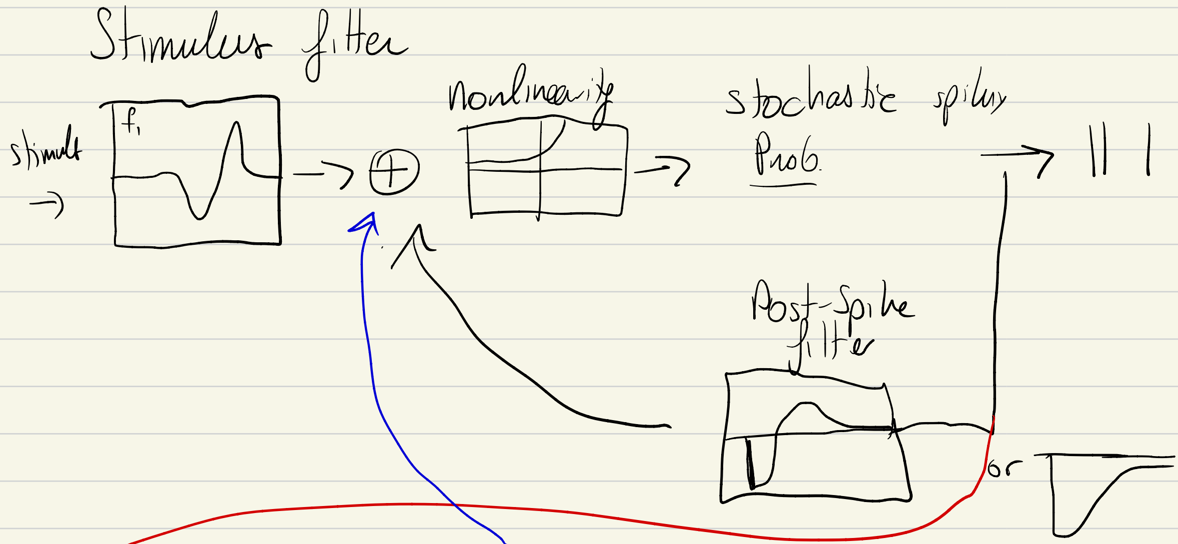
- Poison spiking assumes that each spike time is independent of all others. However, in real neurons, $\exists$ "cooldown" time (refractory period) where a cell cannot fire immediately after first firing (~1ms), so if a cell is firing hundreds of times, this effect is magnified



What to add/include into our spike-generation model

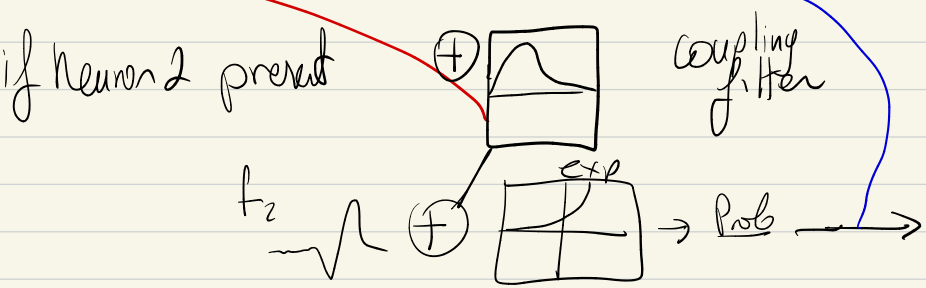
- taking into account history of cell's own firing
- taking into account cell's interaction w/ other cells

Generalized linear model

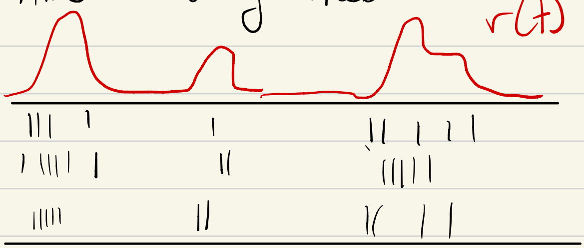


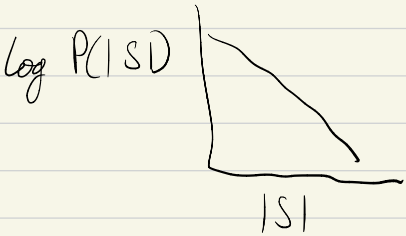
GLM: $P(\text{spike at } t) \sim \exp(f_i \cdot s + h_i \cdot r)$

*Sacrifice some generality for model that's complete, which in optimization is globally convergent.



Time-rescaling Theorem





exp. dist. of scaled ISI's

$(t_{i+1} - t_i)\, r(t)$