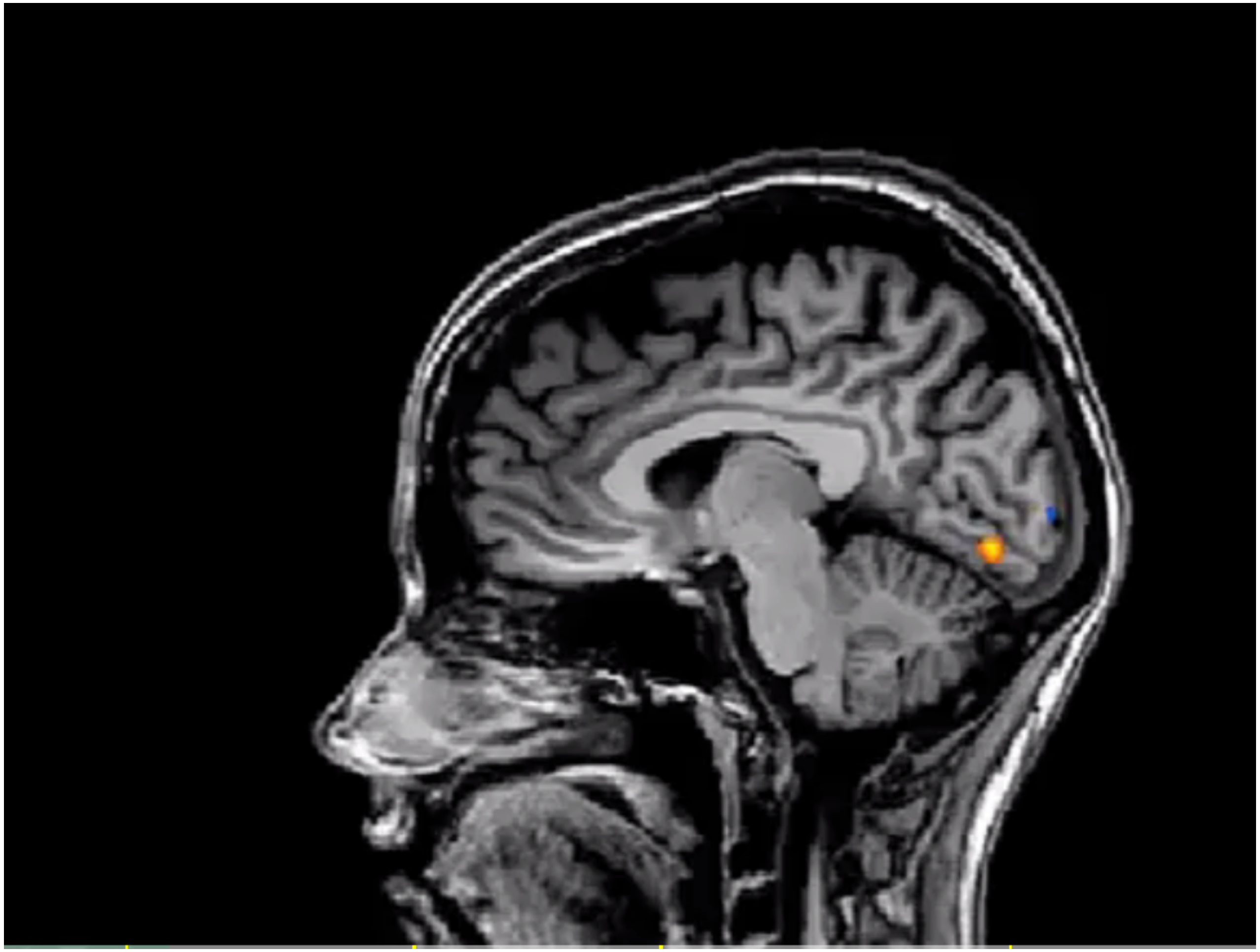
Encoding and decoding neural information

Course of Neuroinformatics

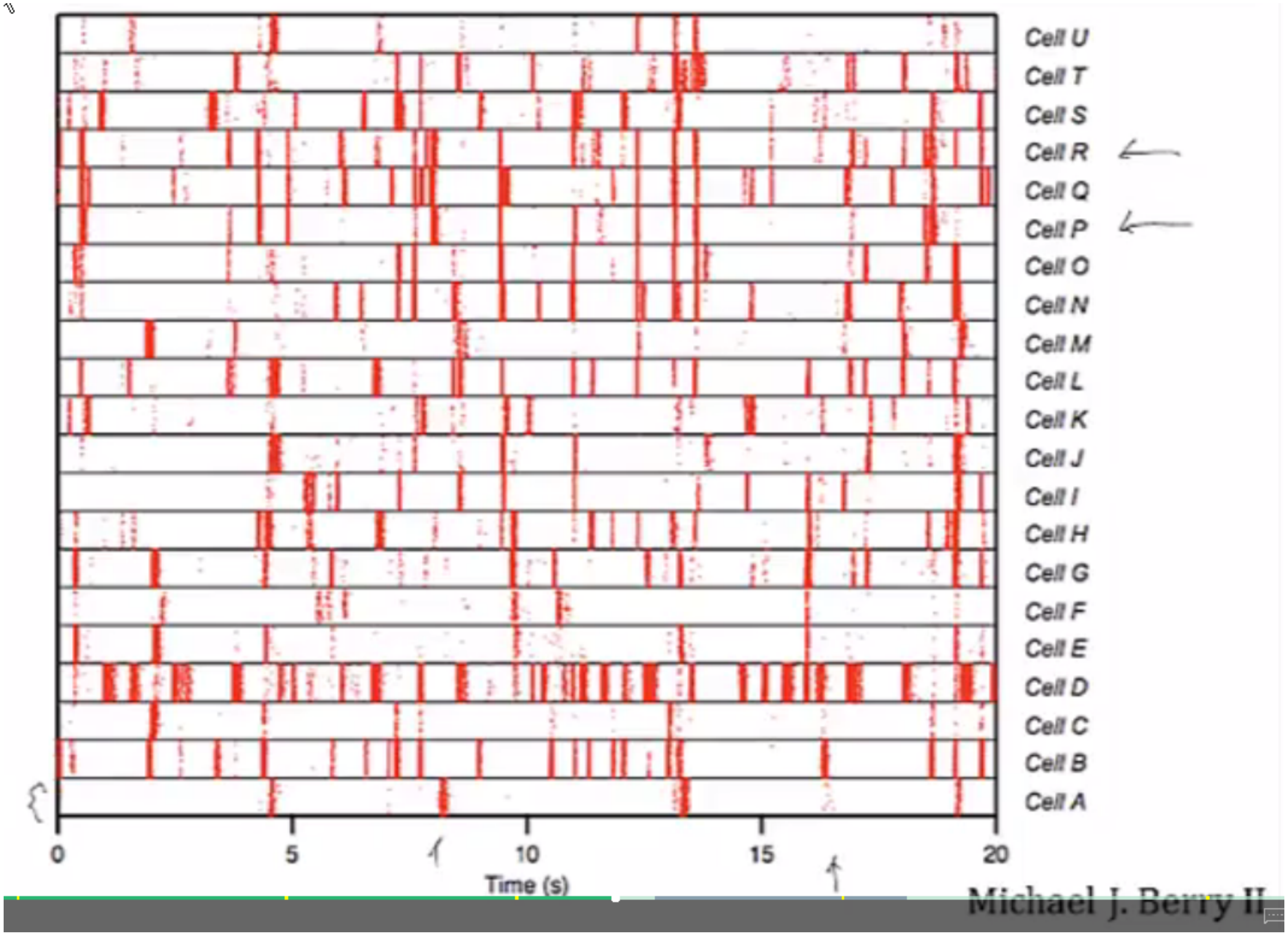
Karla Štěpánová

CIIRC, CTU in Prague

Introduction



Introduction



Introduction

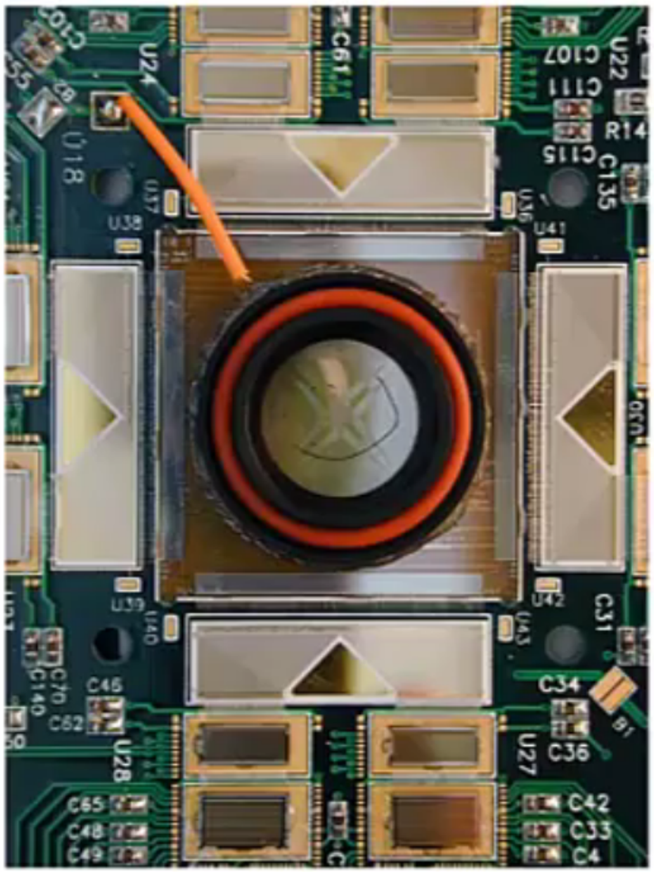
- Encoding
  - How the brain represents information (models to express it)
  - P(response | stimulus)
  - Each neuron different response – only frequency is not enough
- Decoding
  - What was the cause of the given spike pattern of the neuron?
  - Decide whether to behave or not
  - What the brain is doing based on its activity
  - P(noise) >> P(signal)
- Information theory
  - How much information can be encoded and transferred?
  - Information theory to quantify neural representation

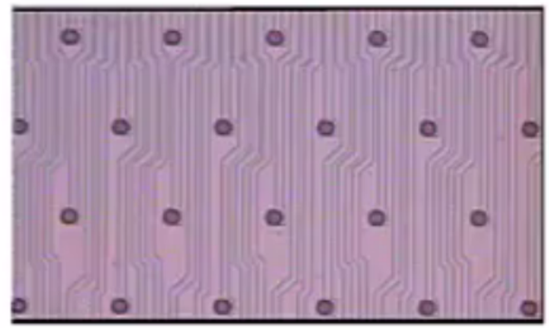
Reading neural code

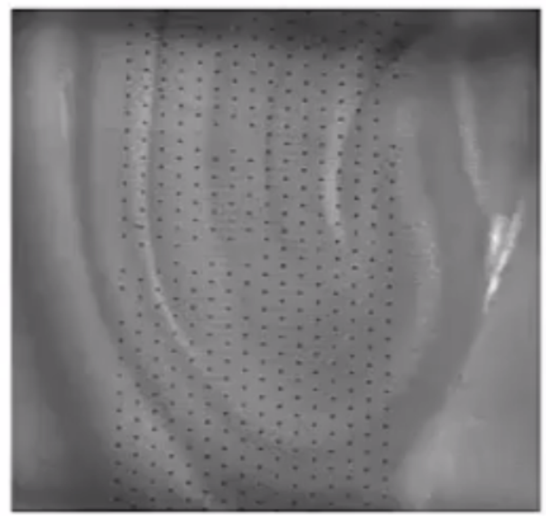
- Non-invasive – EEG, fMRI

Reading neural code

- EEG
- Multielectrode array (amplifying signals, electrode 10μm ~ neuron)

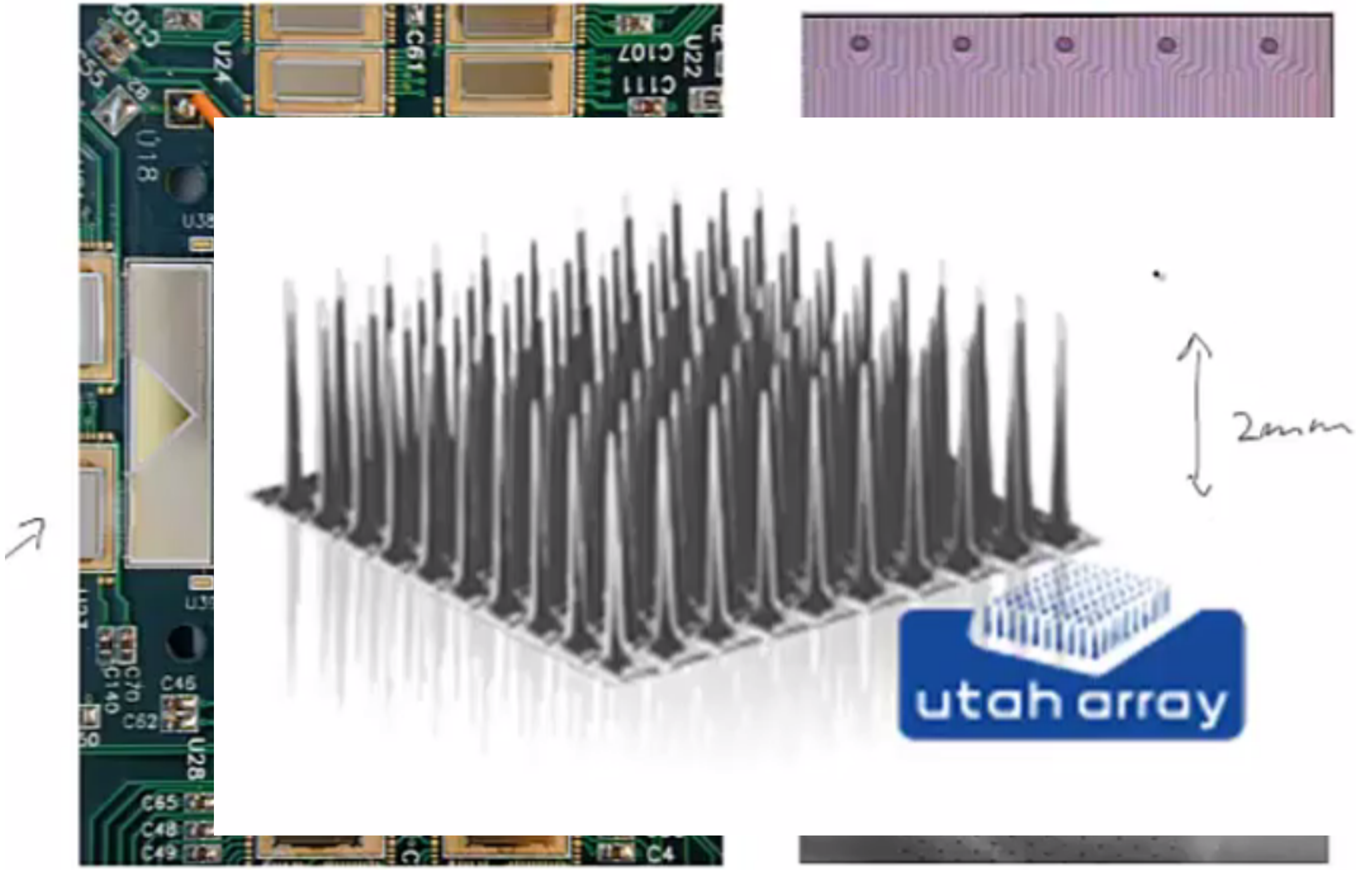






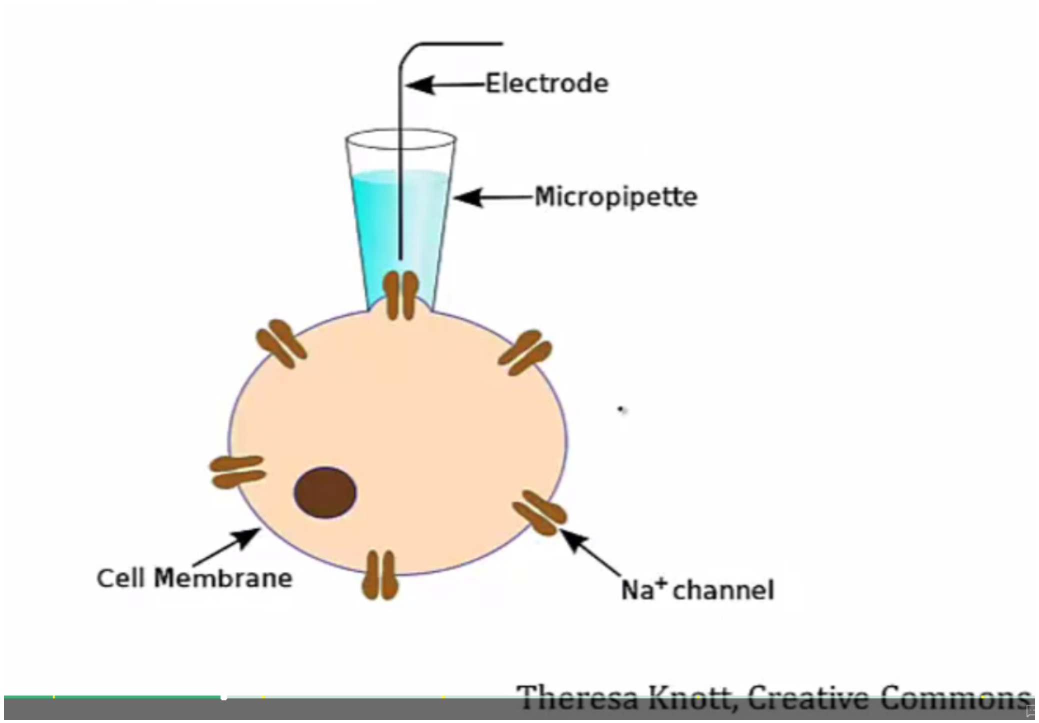
Reading neural code

- EEG
- Multielectrode array (amplifying signals, electrode 10μm ~ neuron)

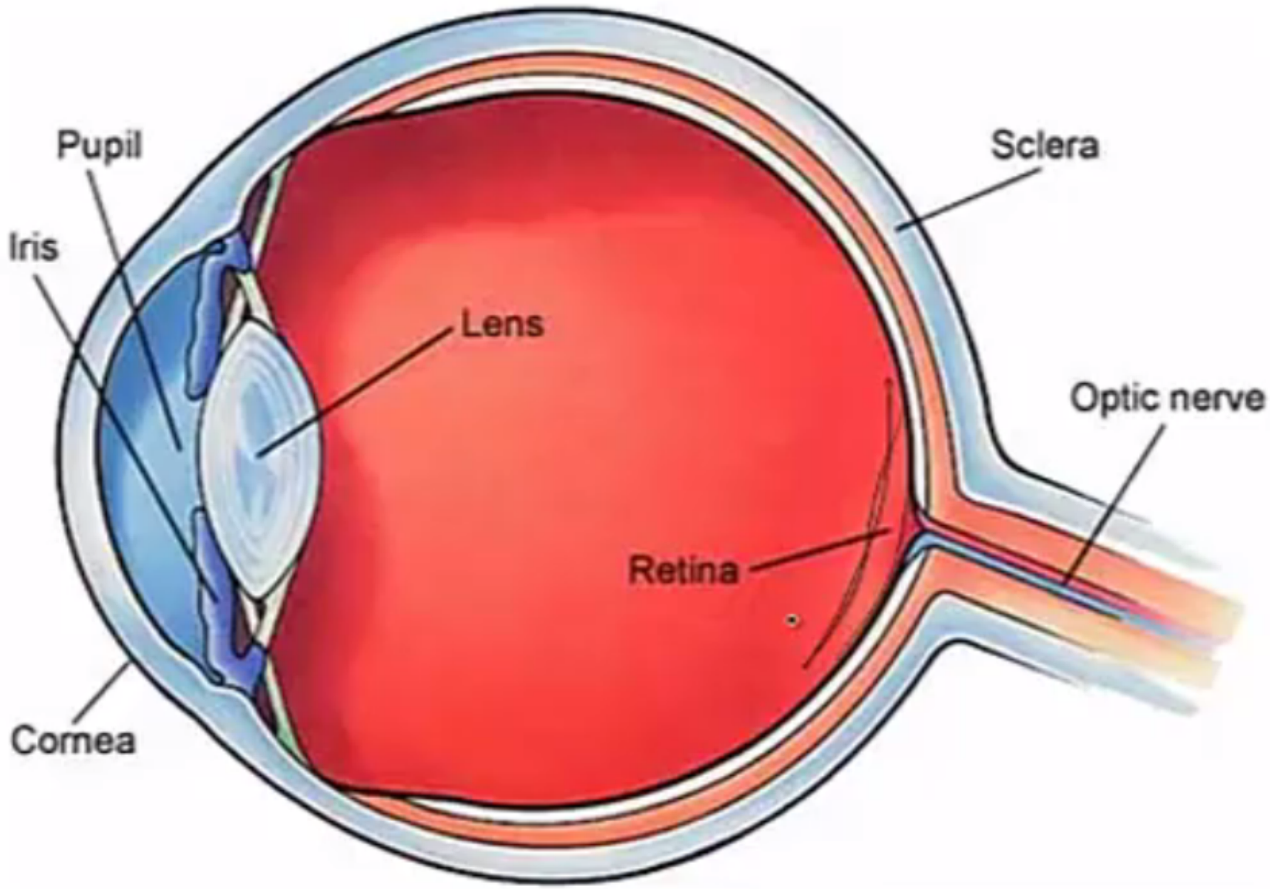


Reading neural code

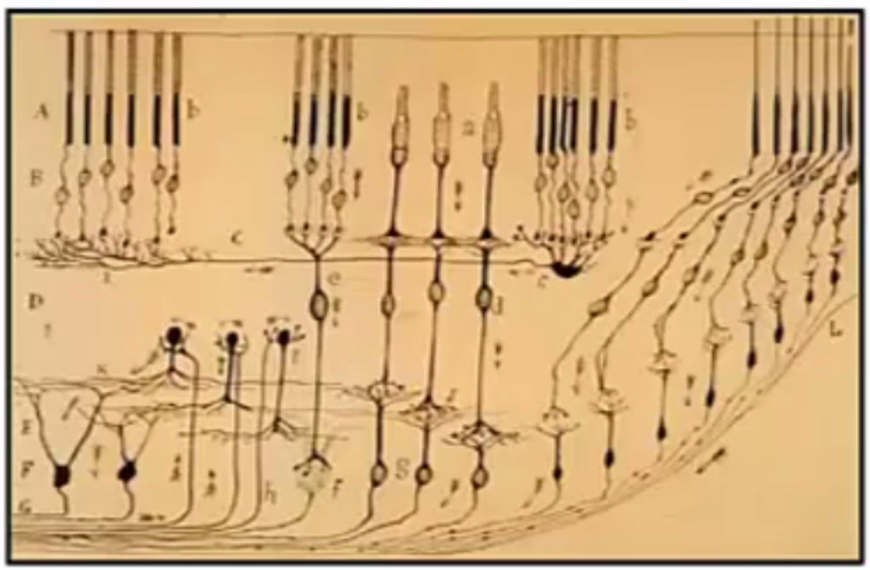
- EEG
- Multielectrode array (amplifying signals, electrode 10μm ~ neuron)
- Single neuron recordings



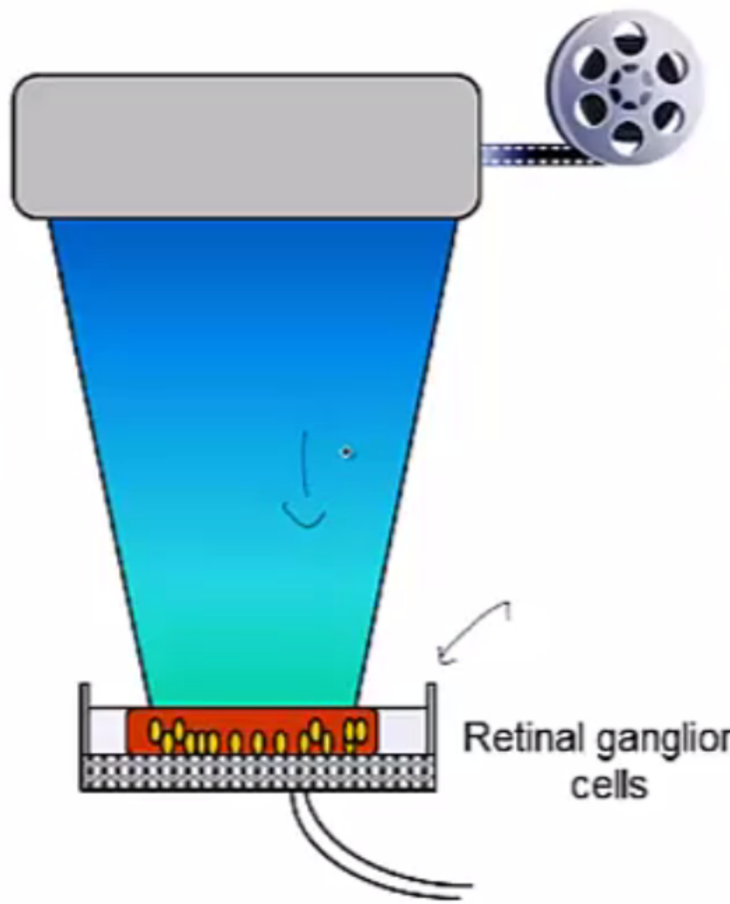
Encoding



Encoding



Ramon y Cajal, 1901

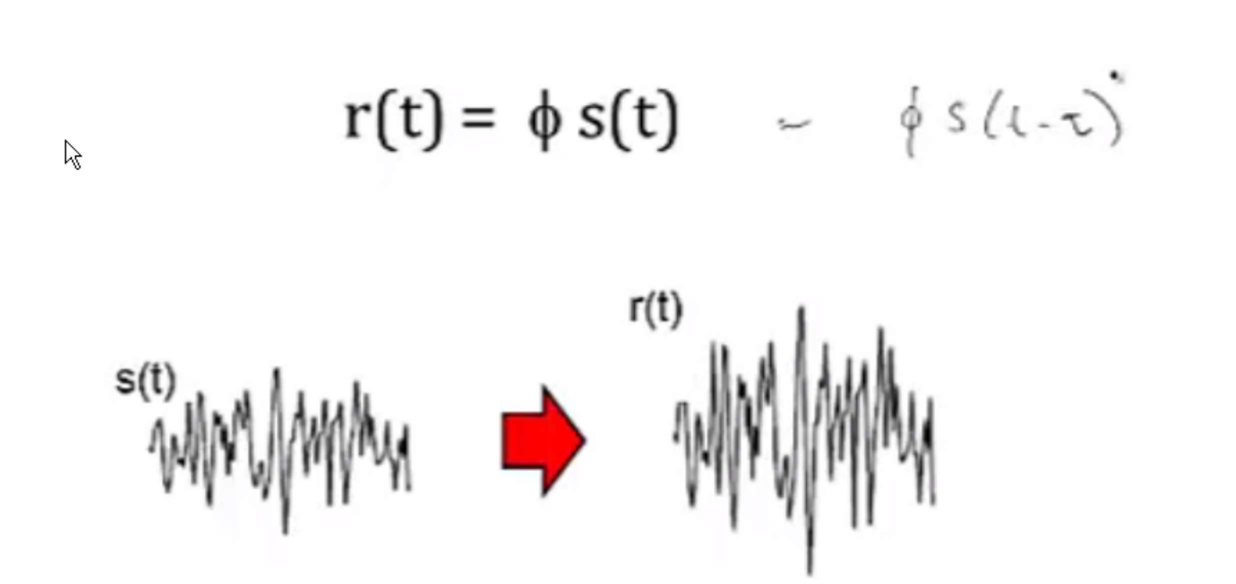


Encoding

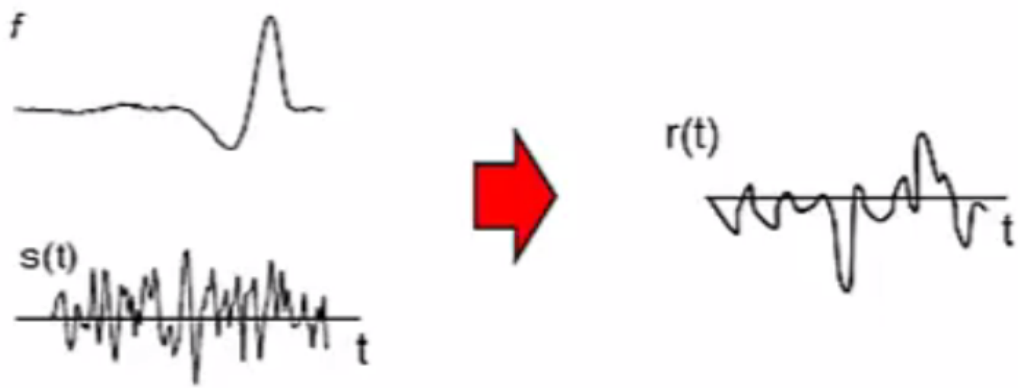
- P (response | stimulus) – r(t) given a stimulus

Encoding

- P (response | stimulus) – r(t) given a stimulus
- Linear response



Encoding – temporal filtering

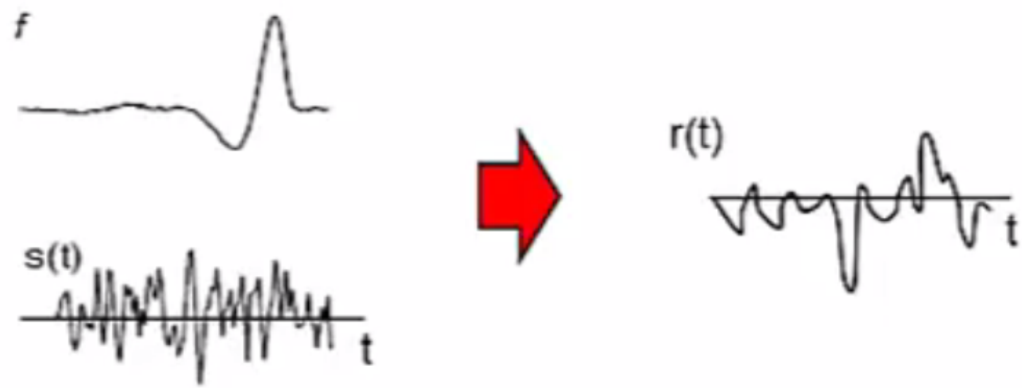


Linear filter:

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

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

Encoding – temporal filtering

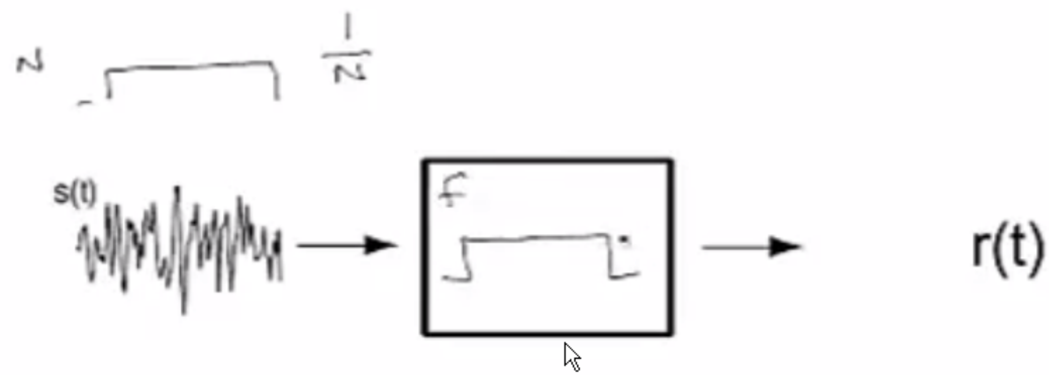


Linear filter: $$r(t) = \sum_{k=0}^{n} s_{t-k} f_k$$

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

Encoding – temporal filtering

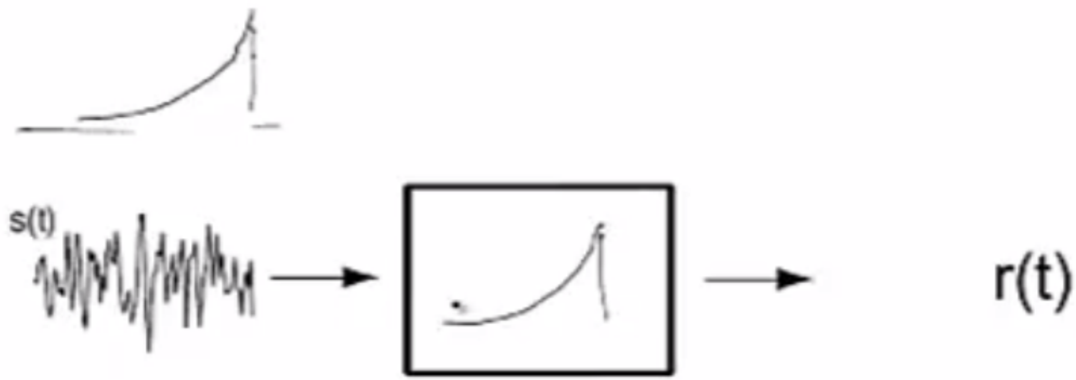
- Running average



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

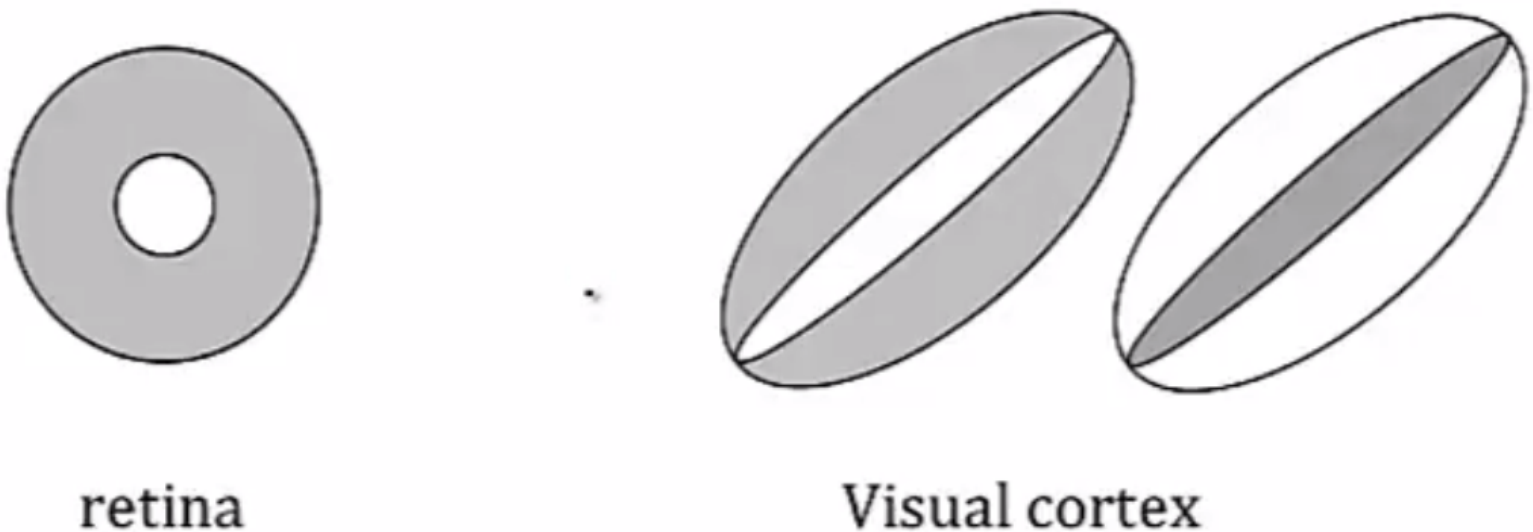
Encoding – temporal filtering

- Leaky average – differential filter (high output for quick increase in signal, not if held high)



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

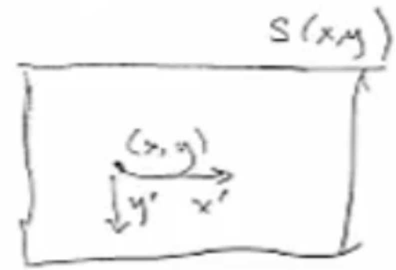
Encoding – spatial filtering



Encoding – spatial filtering

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

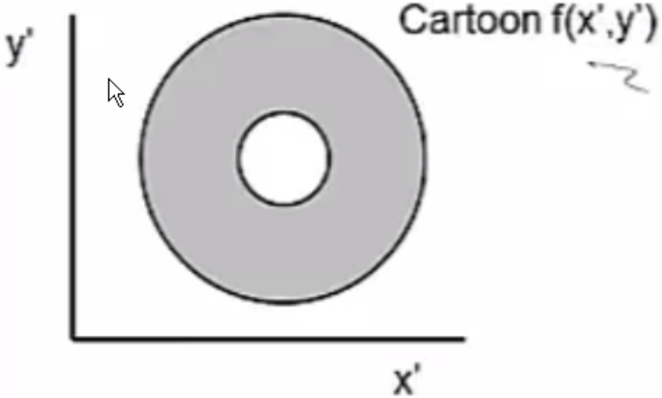
Temporal filter

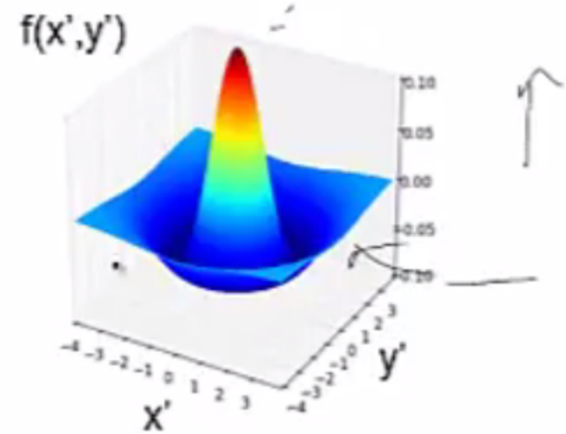


$$r(x,y) = \sum_{x'=-n,y'=-n}^{n} s_{x-x',y-y'} f_{x',y'}$$

Encoding – spatial filtering

- Retinal receptive fields

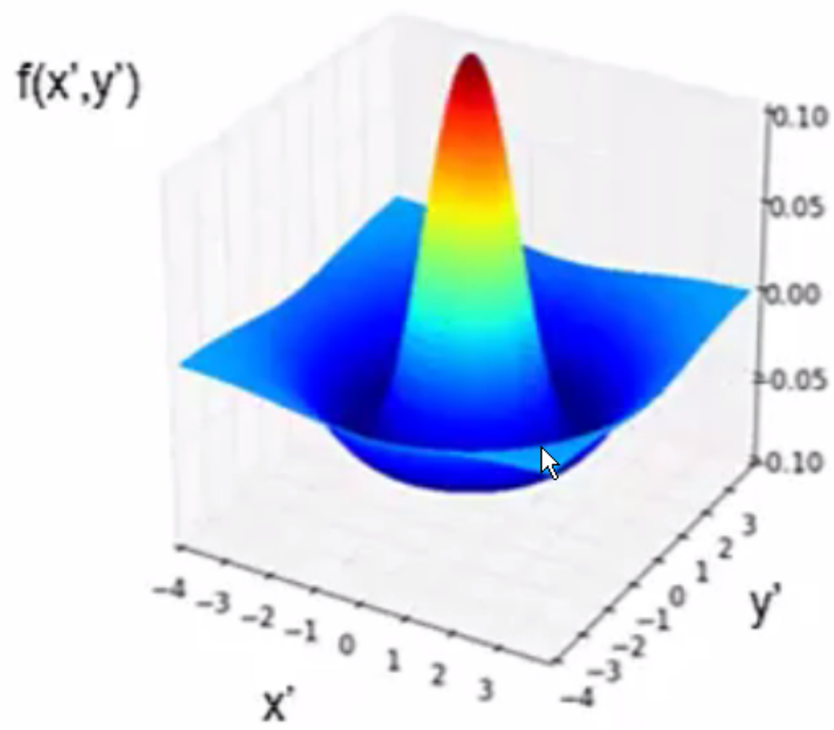




$$r(x,y) = \sum_{x'=-n,y'=-n}^{n} s_{x-x',y-y'} f_{x',y'}$$

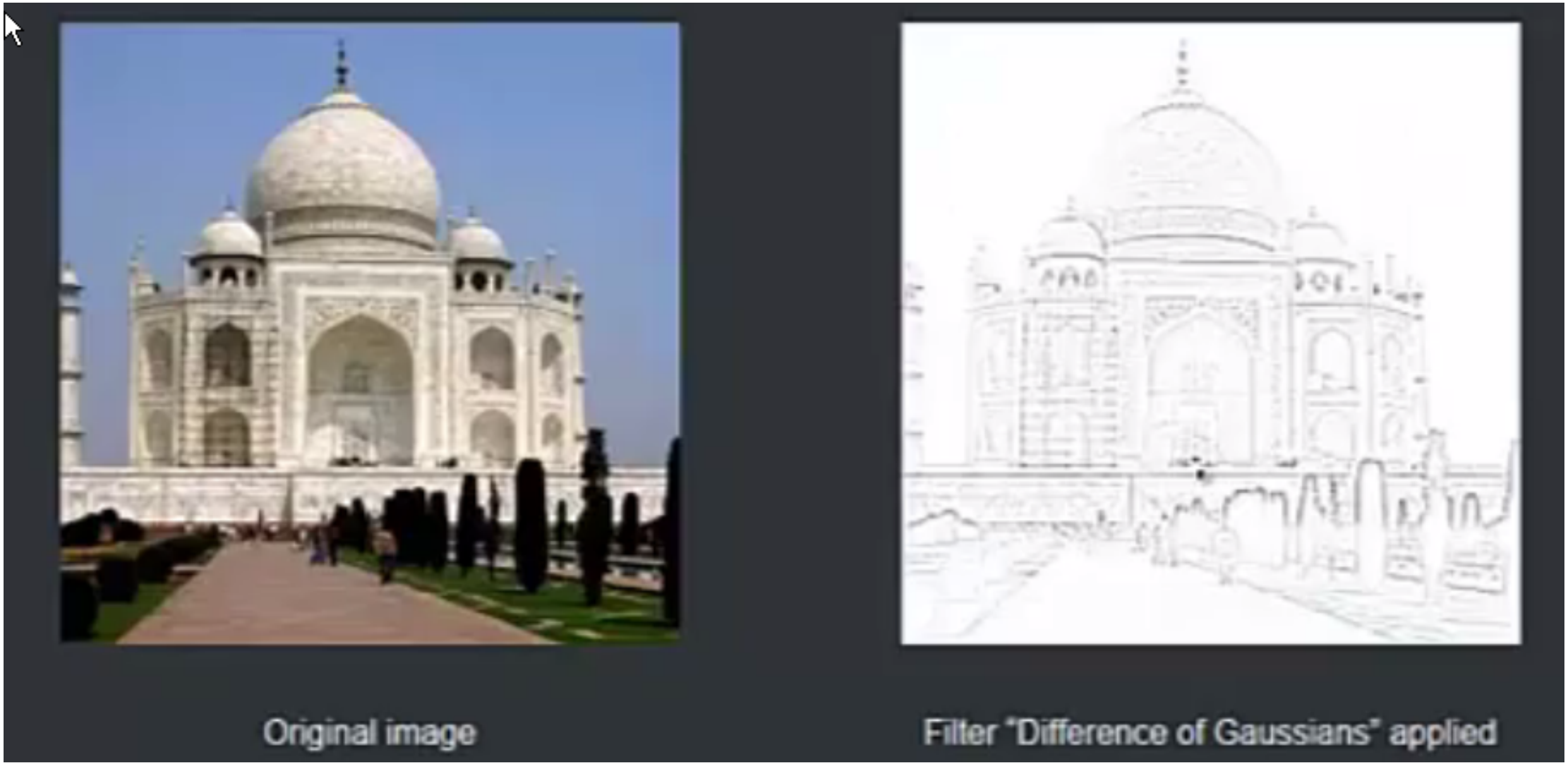
Encoding – spatial filtering

- Retinal receptive fields

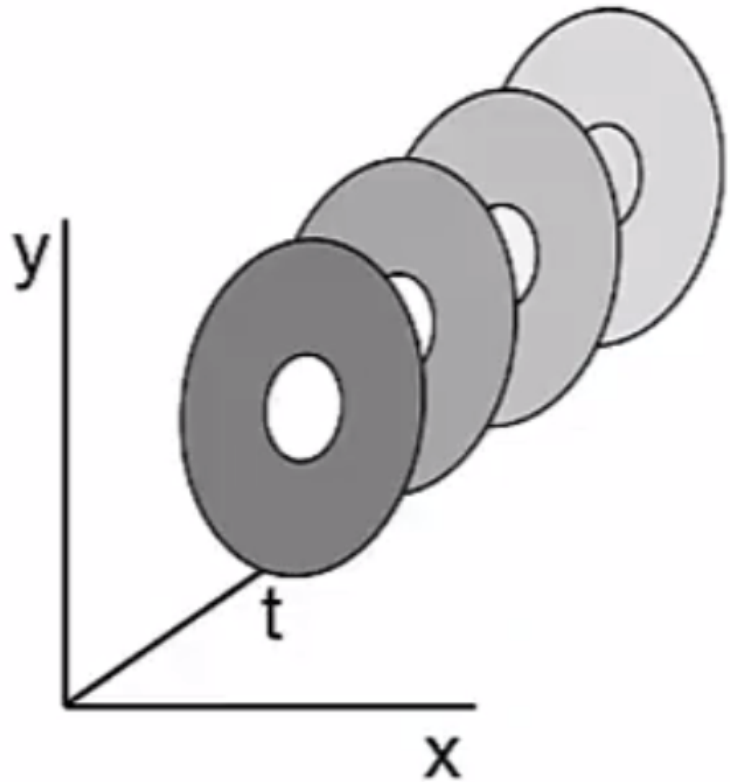


Encoding – spatial filtering

- Retinal receptive fields

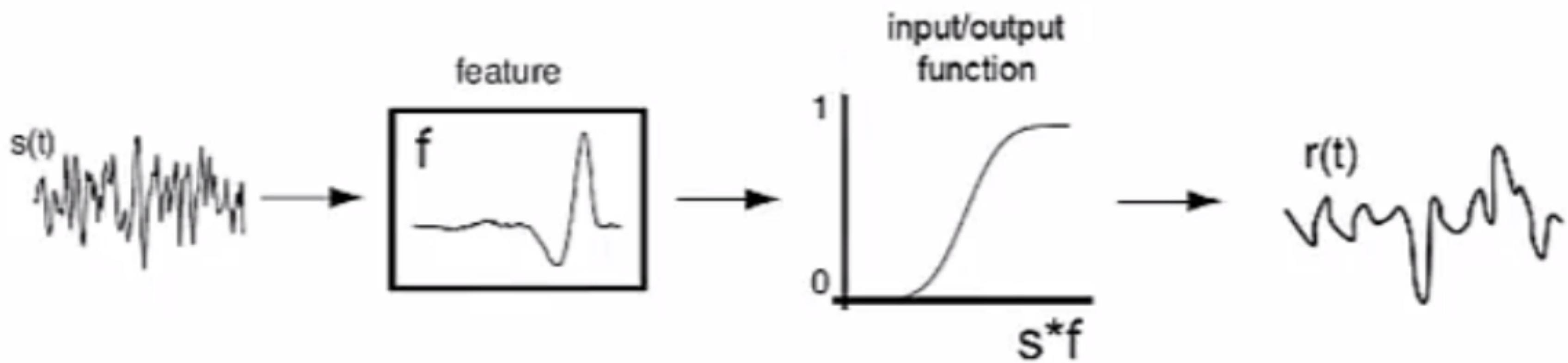


Encoding – spatiotemporal filtering



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

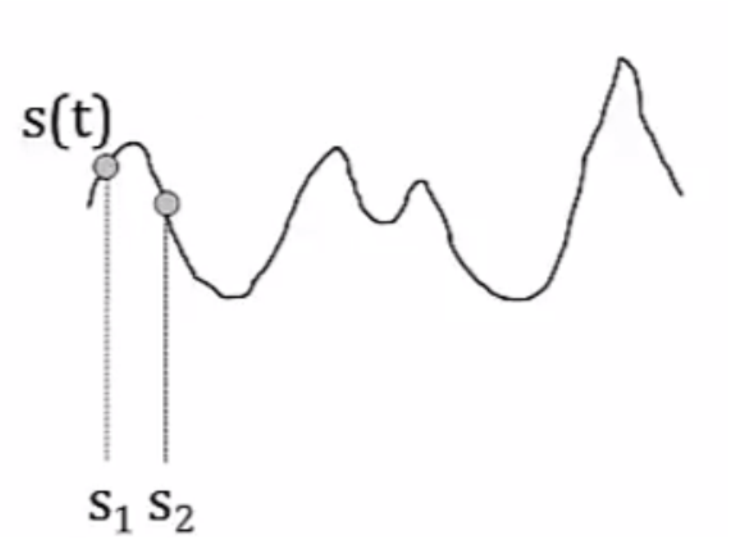
Encoding – temporal filtering



Linear filter & nonlinearity: $r(t) = g(\int s(t-\tau)\, f(\tau)\, d\tau)$

Encoding – feature selection

- One stimulus 300 000 values – impossible to handle
- Sample responses of the system to many stimuli to characterize P(response|stimulus) -> P(response|s1)
- Pick up small set of relevant dimensions
  - Discretize P(response|{s1,…,sn})

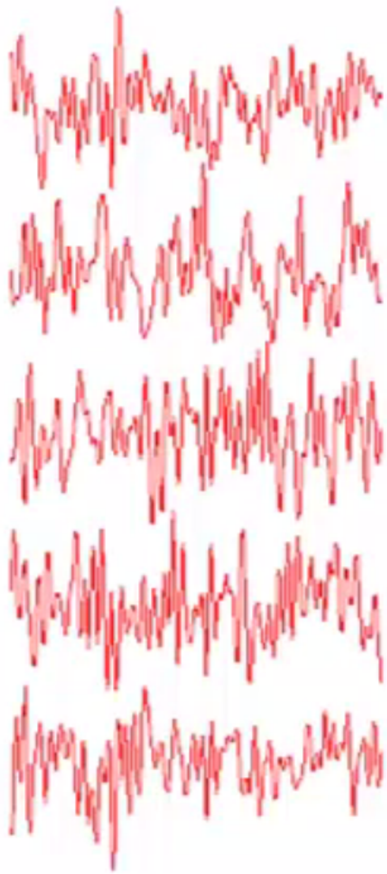


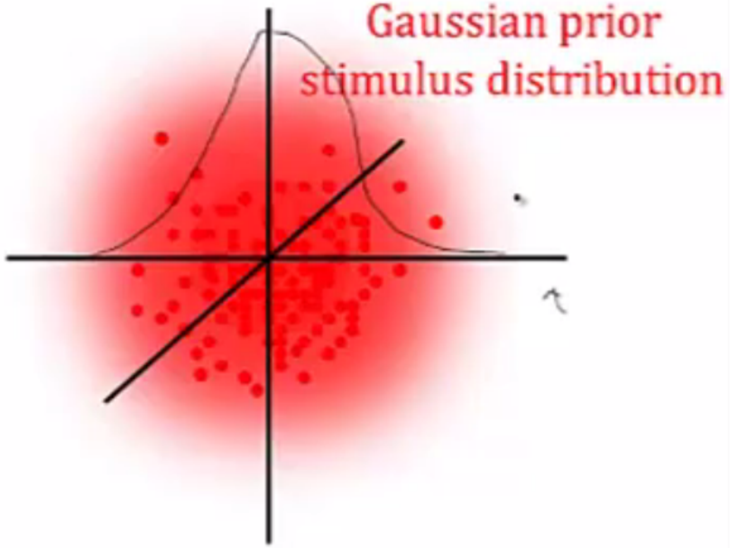
Encoding – feature selection

- Gaussian white noise as an input

Encoding – feature selection

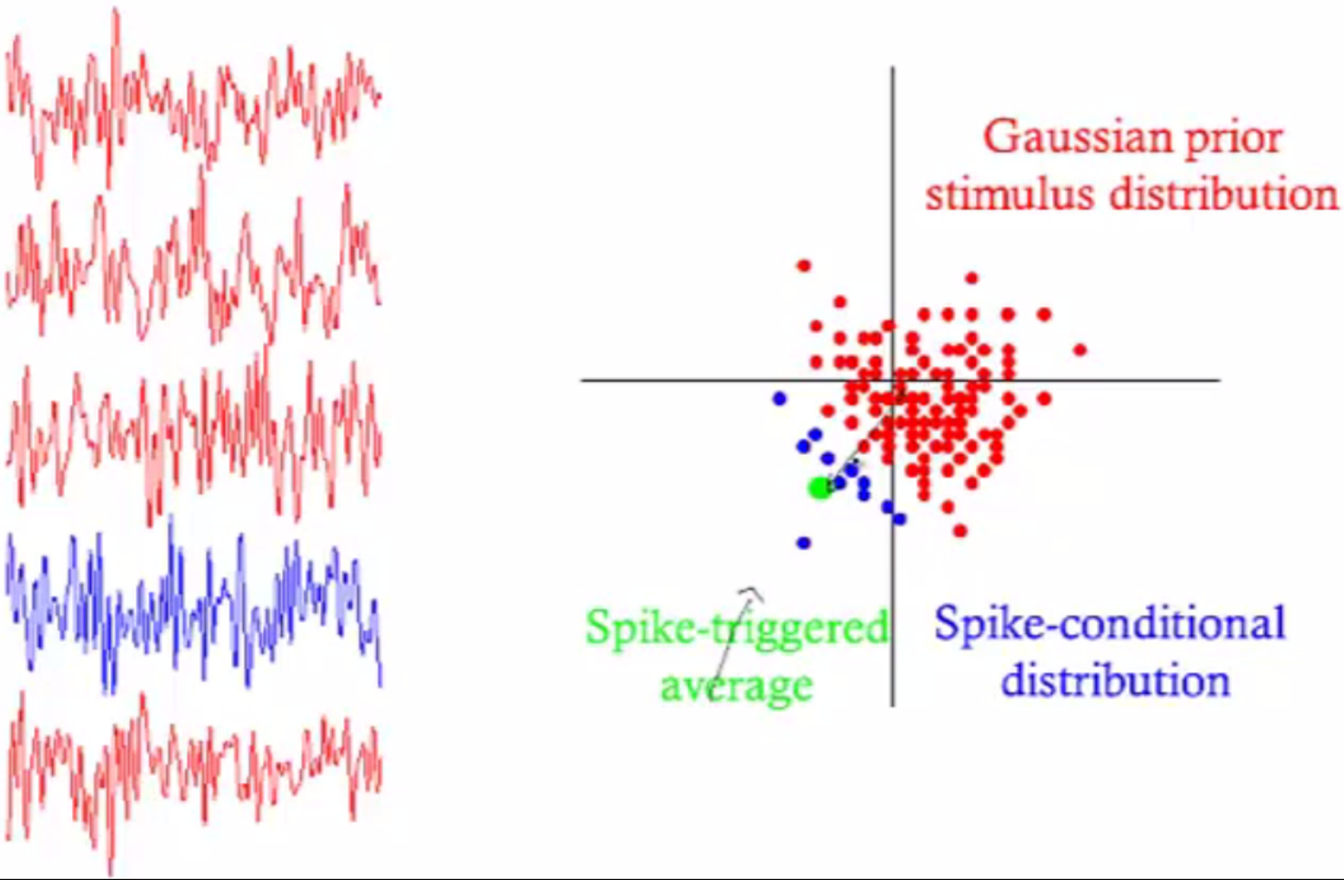
- Gaussian white noise as an input
  - 100 dimensions (time values) – Gaussian dist. along any axis





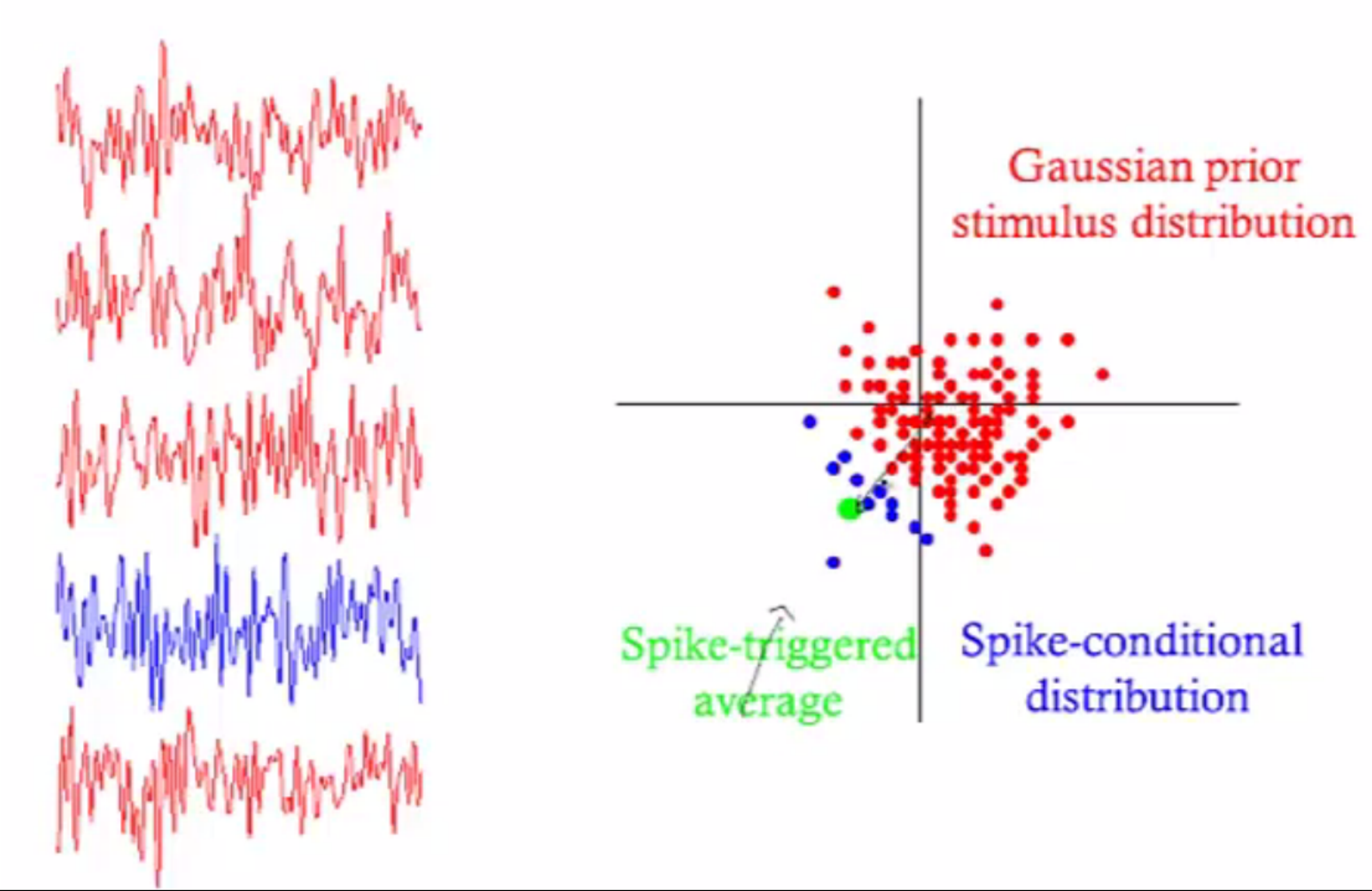
Encoding – feature selection

- Gaussian white noise as an input –
  - some stimuli will trigger spikes



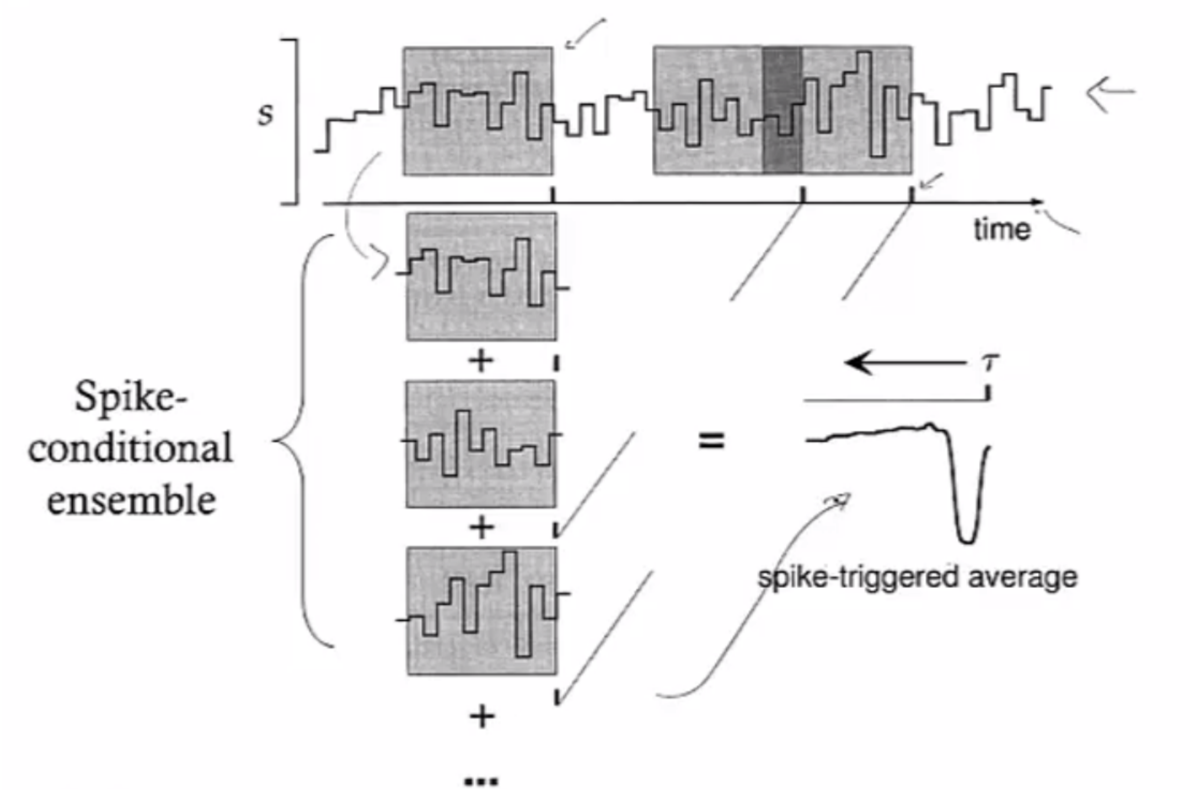
Encoding – feature selection

- Gaussian white noise as an input –
  - some stimuli will trigger spikes



Encoding – feature selection

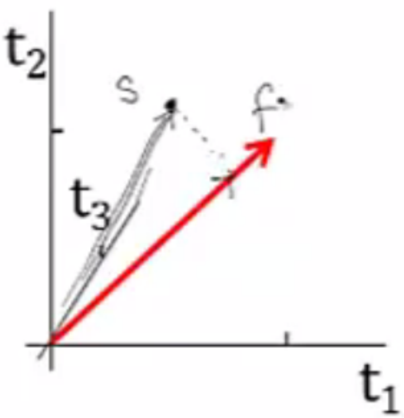
- Gaussian white noise as an input –
  - some stimuli will trigger spikes



Encoding – feature selection

- Linear filtering
  - takes stimulus and project it to lower dimensional space

Stimulus feature *f* is a vector in a high-dimensional stimulus space



Linear filtering = convolution = projection

Encoding – feature selection

- Input/Output function with respect to this extracted feature

The input/output function is:

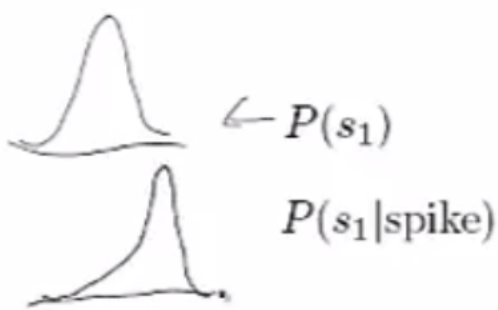
$$P(\text{spike}|\text{stimulus}) \;\Rightarrow\; P(\text{spike}|s_1)$$

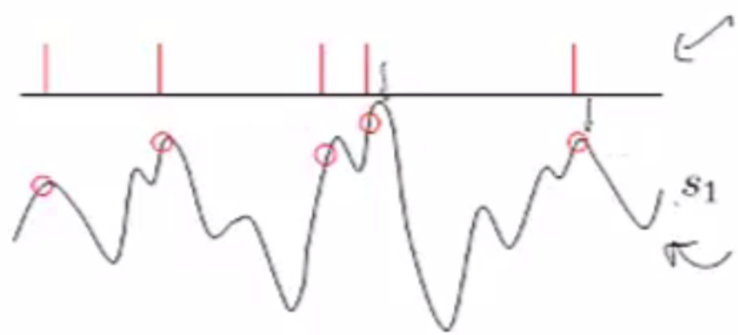
This can be found from data using Bayes' rule:

spike-conditional

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

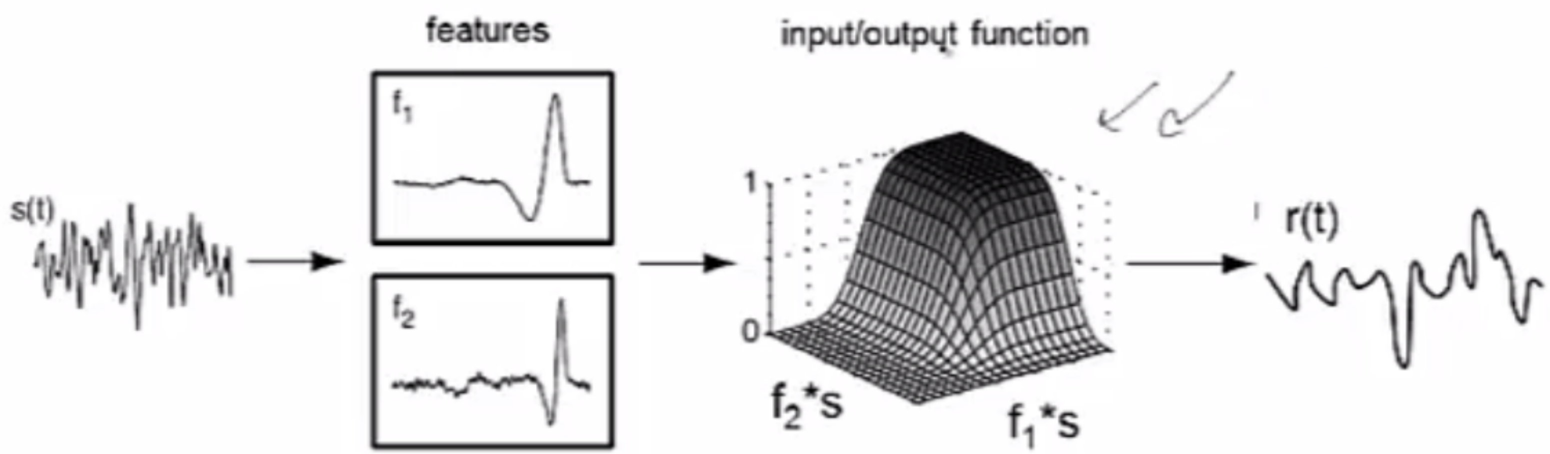
prior





Encoding – feature selection

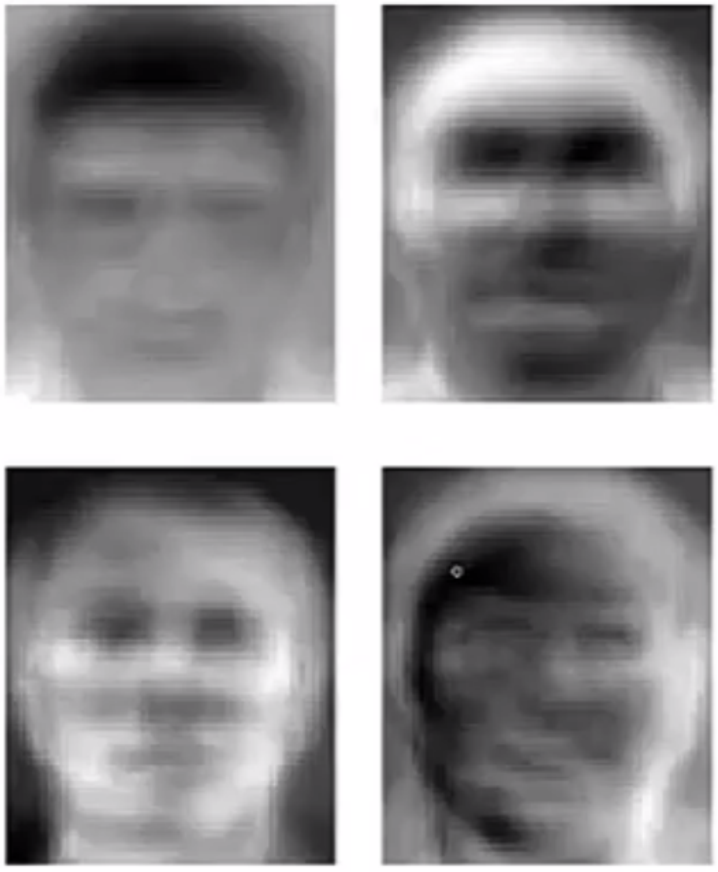
- Input/Output function with respect to this extracted feature



Linear filters & nonlinearity: $r(t) = g(f_1*s, f_2*s, ..., f_n*s)$

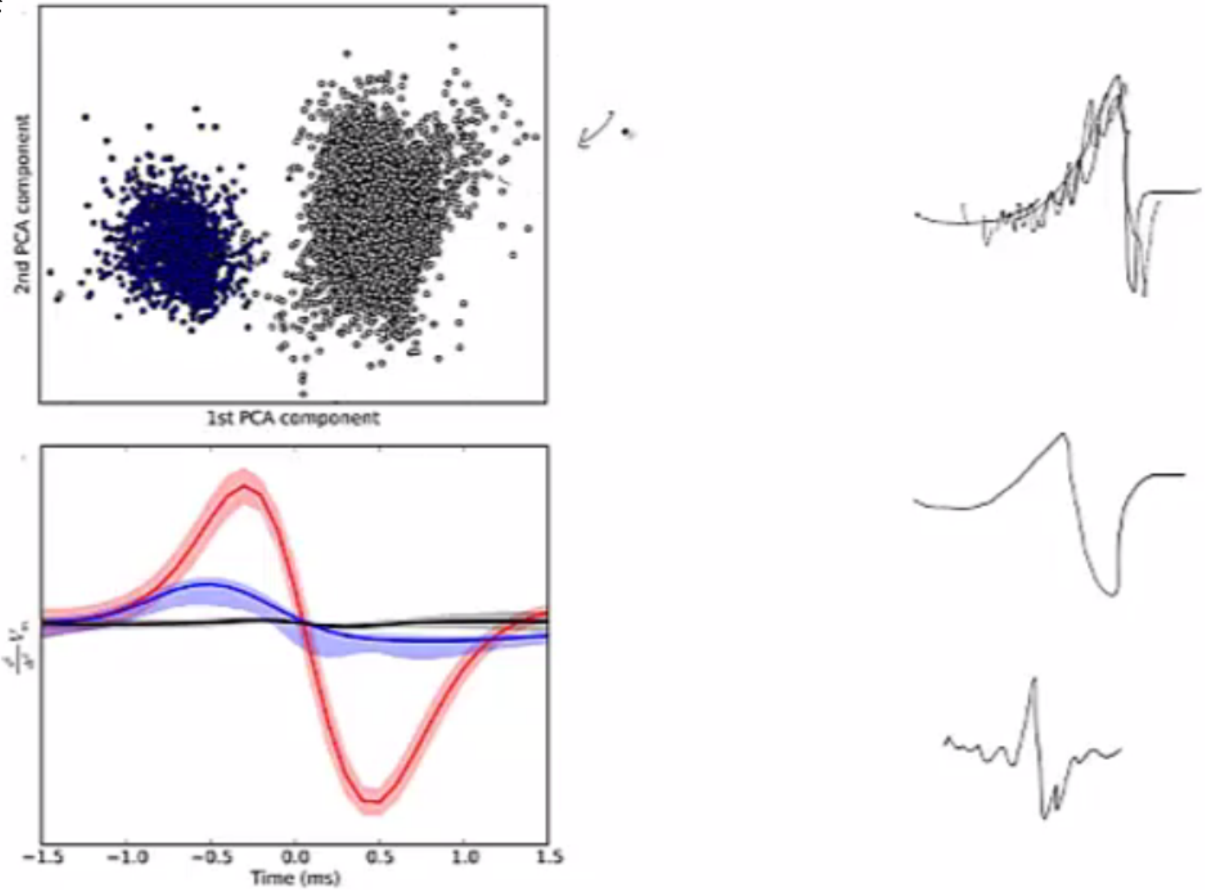
Encoding – feature selection

- PCA
  - eigenfaces from which we can create any face (lin.combination)



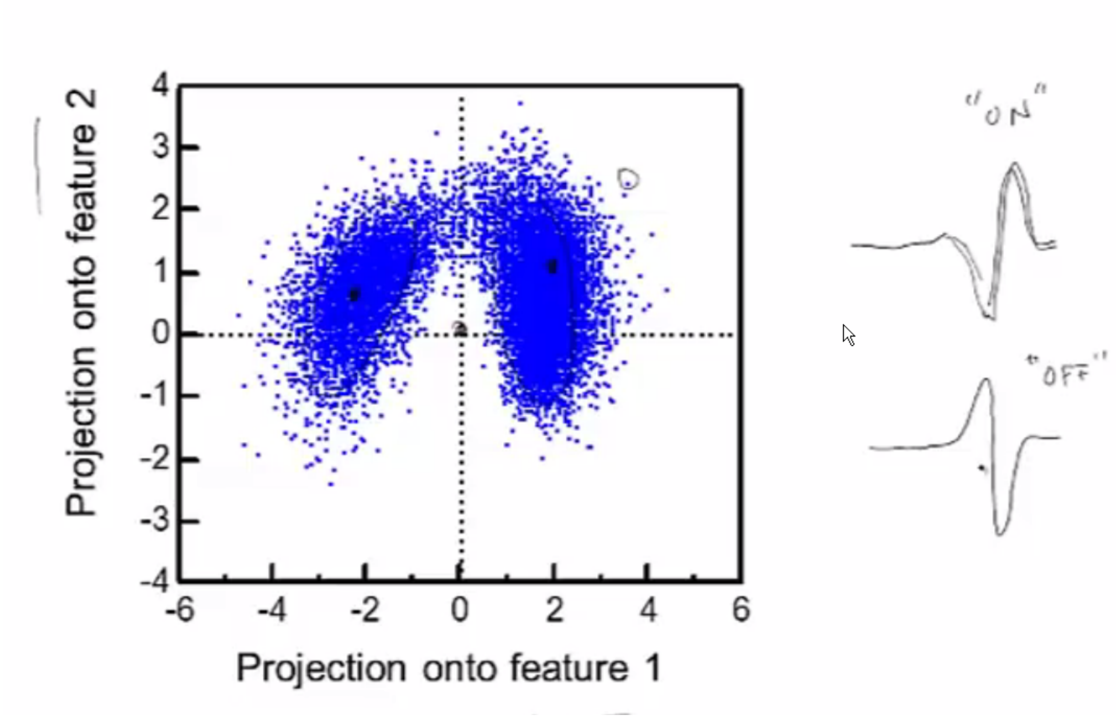
Encoding – feature selection

- PCA
  - Spike sorting (Koepsi et al., 2009)



Encoding – feature selection

- PCA
  - Spike sorting (Fairhall et al., 2007) – response to light changes of visual neurons

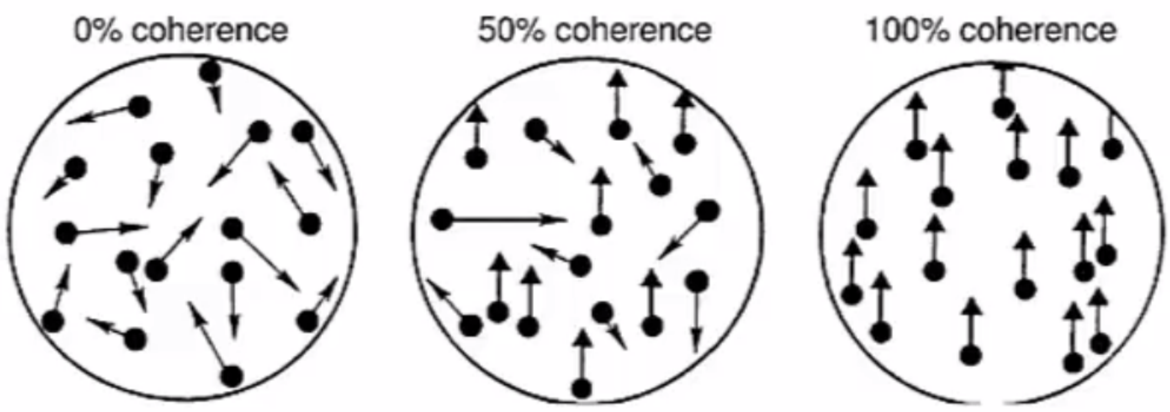


Decoding

- Neural response -> what was the stimulus
- Predictable from neural activity
- Signal detection theory

Decoding

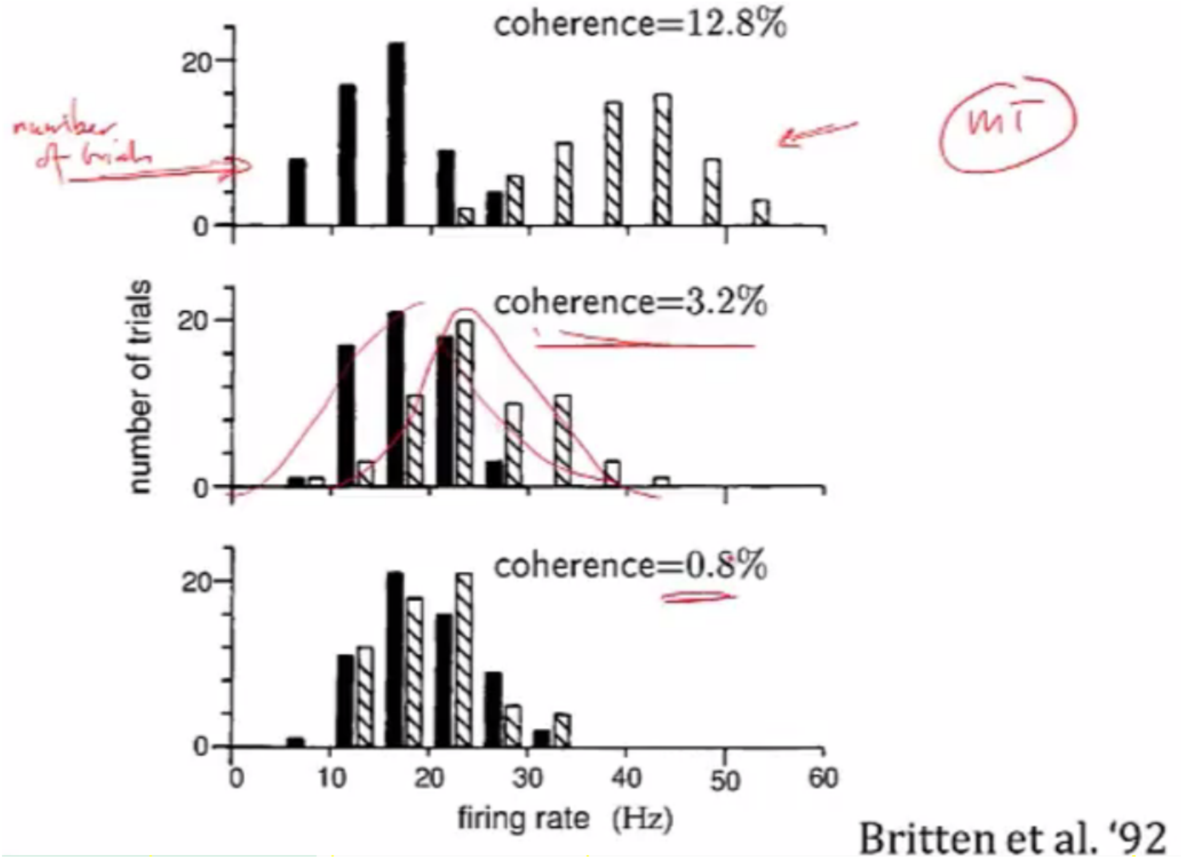
- Monkey – neurons sensitive to visual motion
  - Different coherence of input (nmb of dots moving together)



Britten et al. '92

Decoding

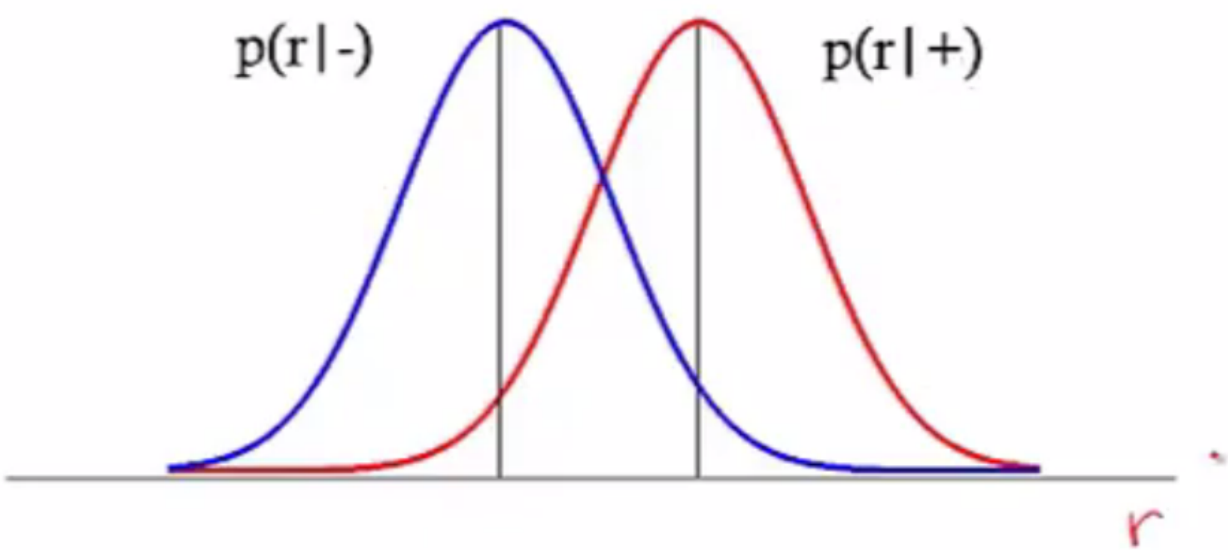
- Monkey – neurons sensitive to visual motion
  - Different coherence of input (nmb of dots moving together)



Britten et al. '92

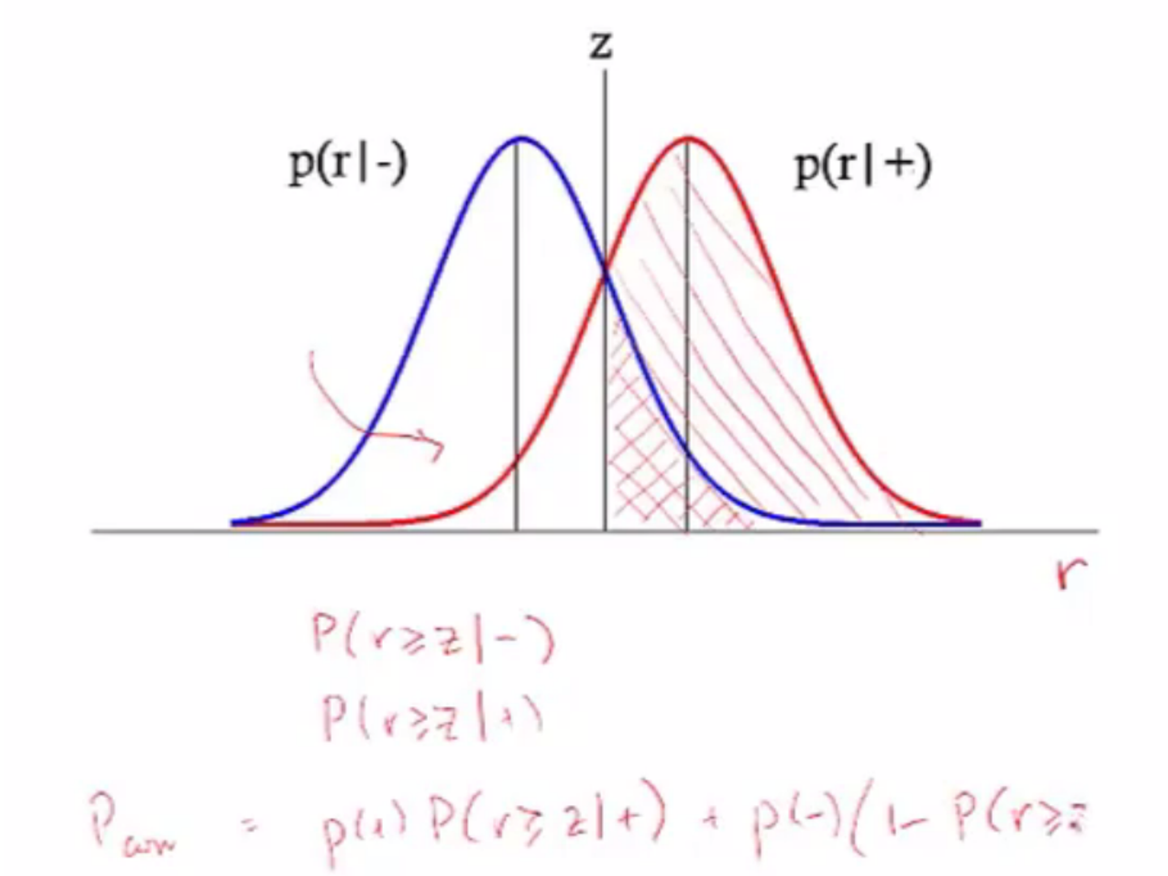
Decoding

- Monkey – neurons sensitive to visual motion
  - Different coherence of input (nmb of dots moving together)
  - Moving up vs. moving down



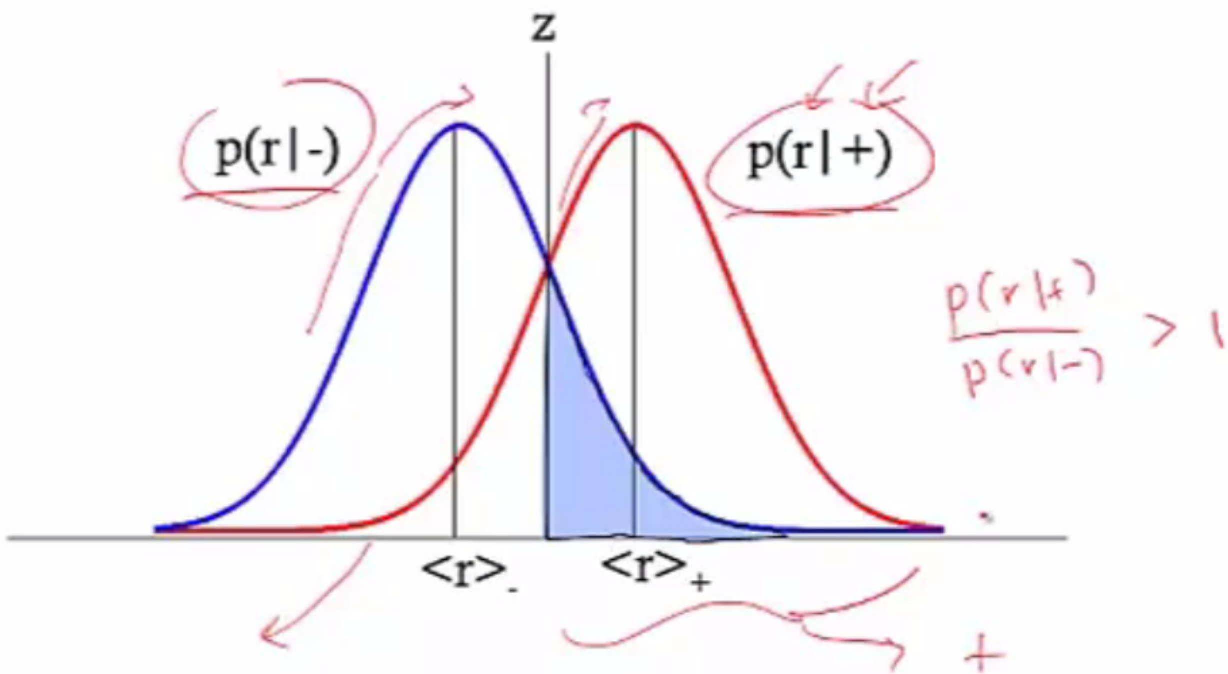
Decoding

- Monkey – neurons sensitive to visual motion
  - Threshold – maximum probability of correct answer



Decoding

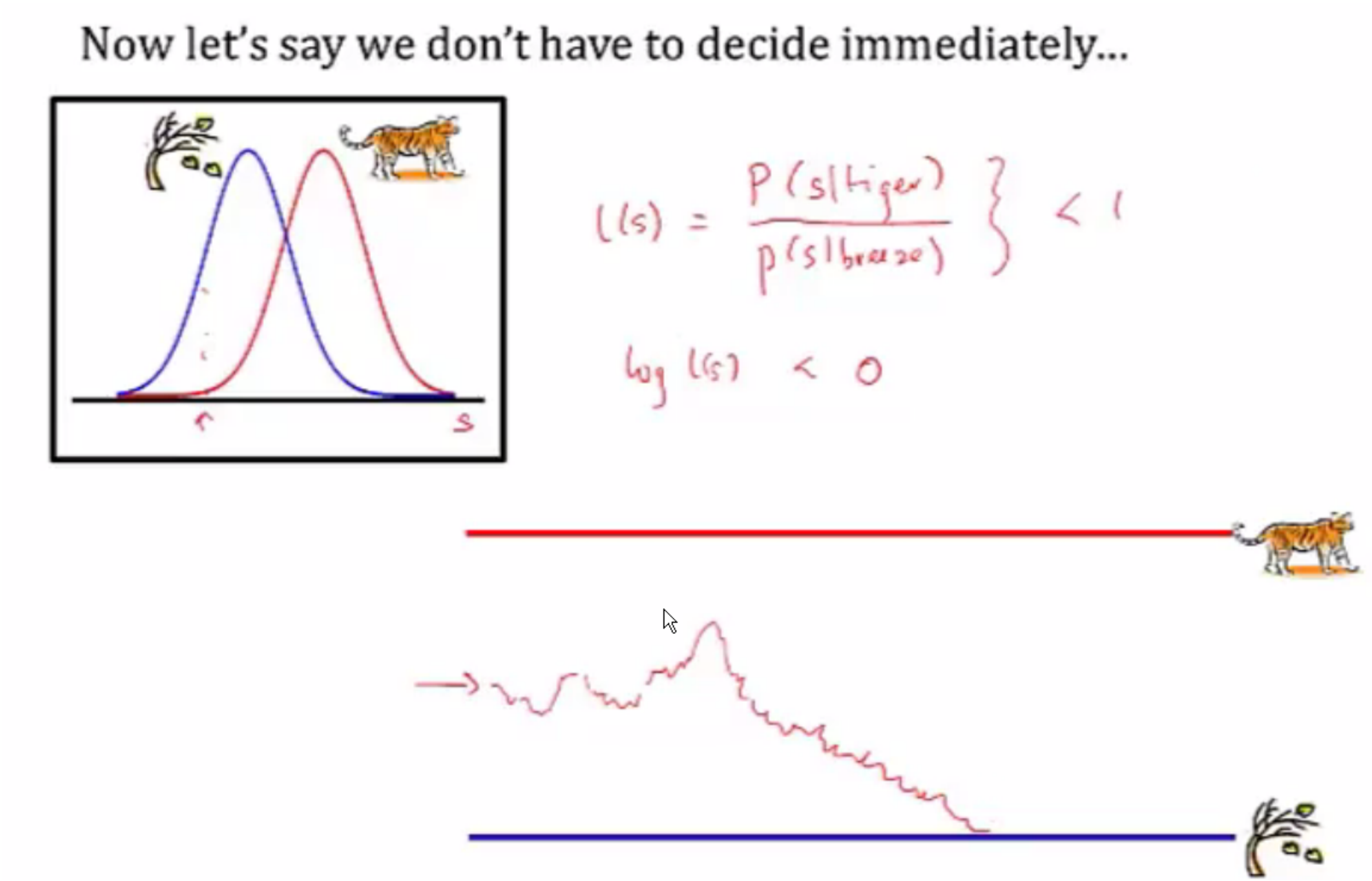
- Likelihood ratio – Neymann-Pearson lemma



The likelihood ratio test is the most efficient statistic, in that it has the most power for a given size

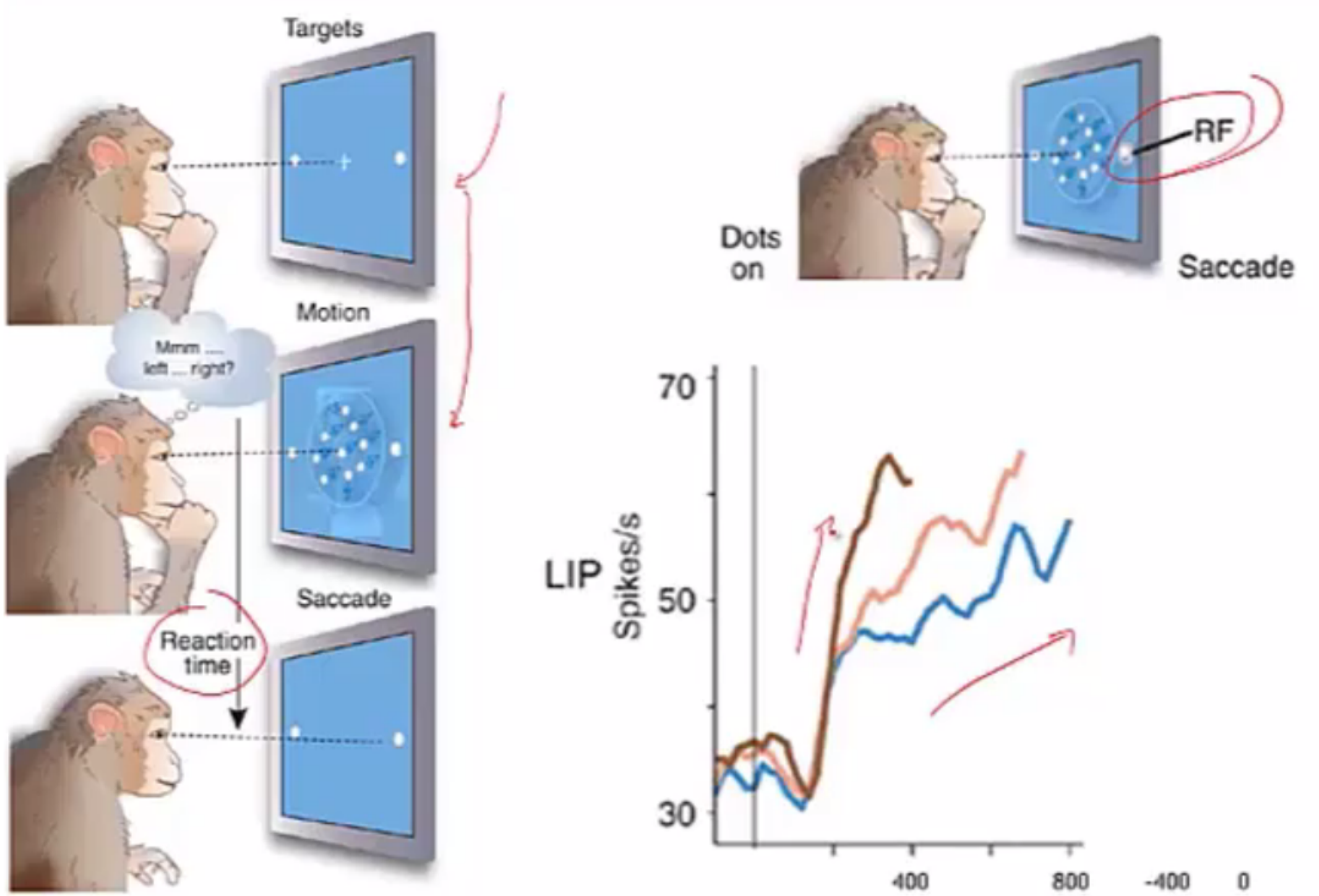
Decoding

- Sequence of observations – accumulated evidence



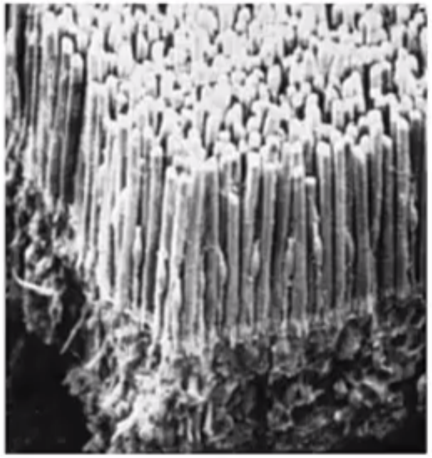
Decoding

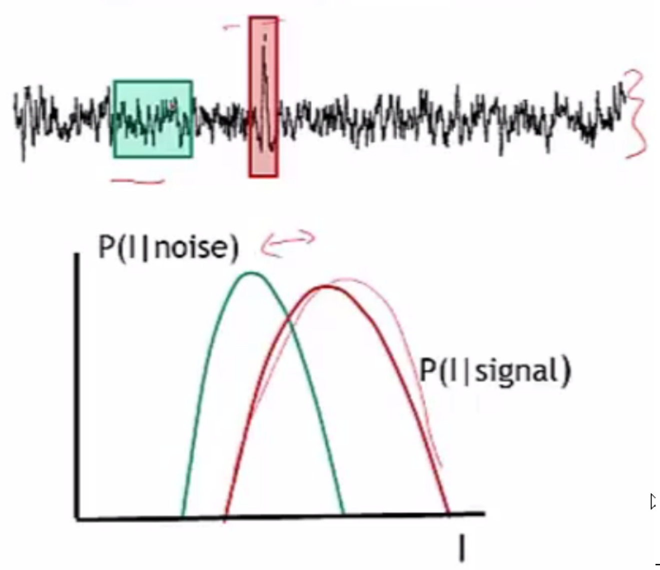
- Sequence of observations – accumulated evidence
  - Kiani et.al, 2006

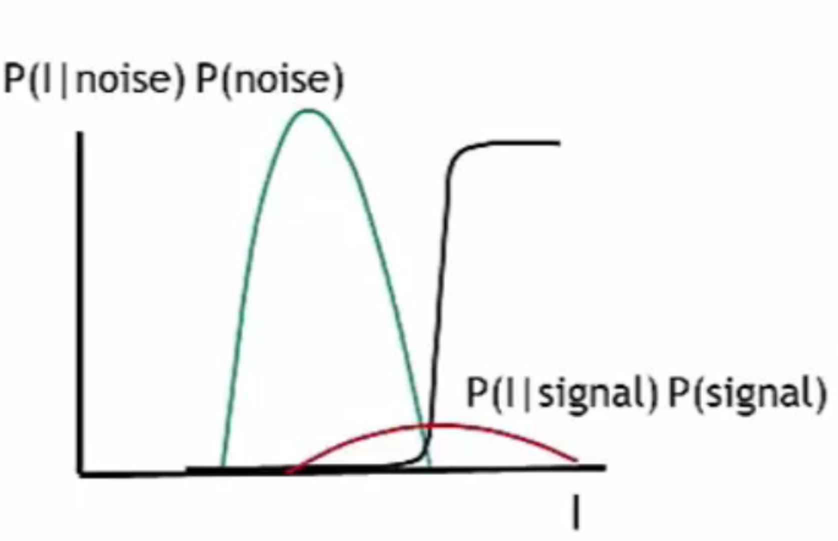


Decoding

- Priors -> changes decisions (retina: P(noise) = 99.9%)







Decoding

- Priors -> changes decisions (retina: P(noise) = 99.9%)
- Costs

Cut your losses: answer + when $Loss_+ < Loss_-$

i.e. $L_+ P[-|r] < L_- P[+|r]$.

Decoding – from populations

- Population vector average (paralel to arm movement)

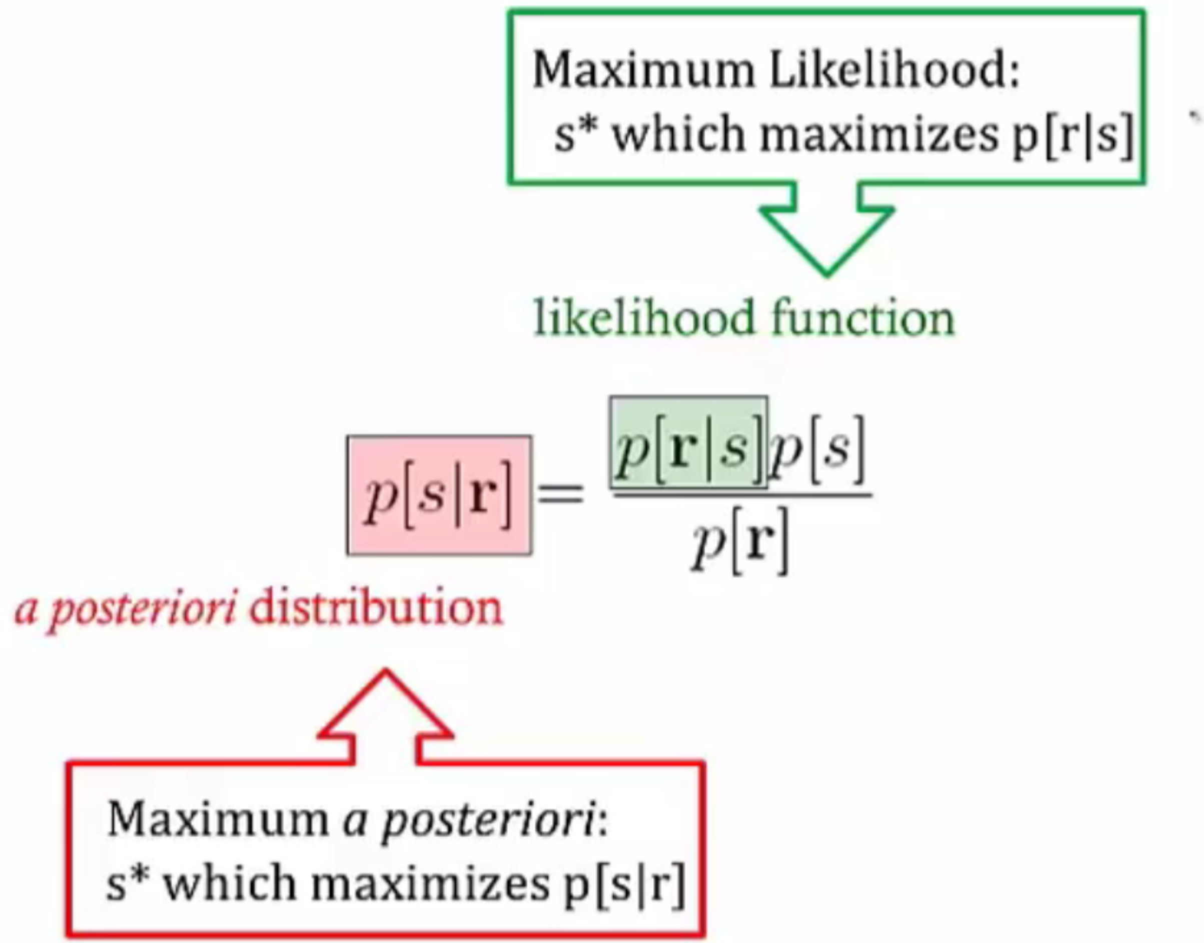
Pop. vector:

$$\vec{v}_{\text{pop}} = \sum_{a=1}^{N} \left(\frac{r - r_0}{r_{\max}} \right) \vec{c}_a$$

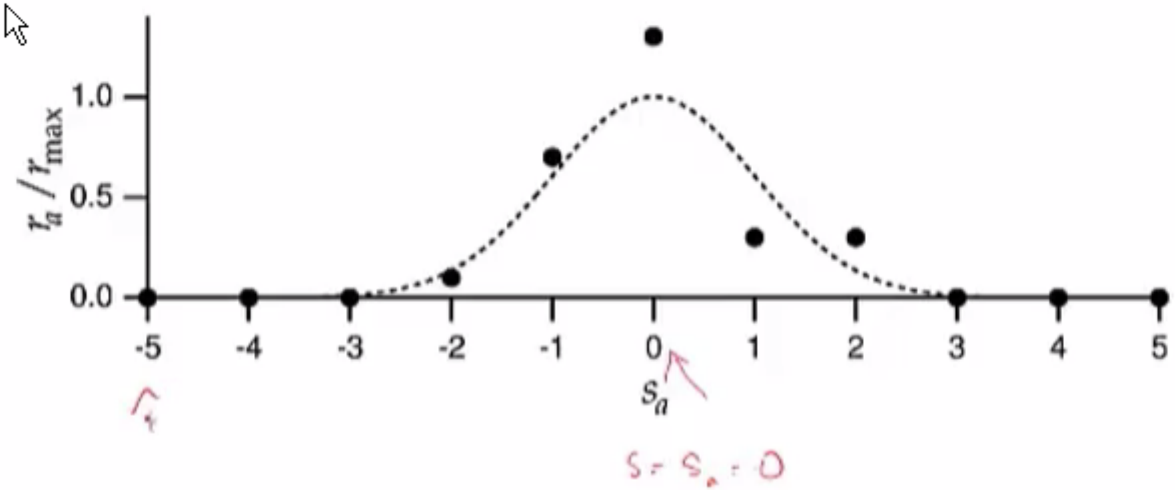
For sufficiently large N,

$$\langle \vec{v}_{\text{pop}} \rangle = \sum_{a=1}^{N} (\vec{v} \cdot \vec{c}_a) \vec{c}_a$$

Decoding – from populations



Decoding – from populations



Population response of 11 cells with Gaussian tuning curves

$$P[\mathbf{r}|s] = \prod_{a=1}^{N} \frac{(f_a(s)T)^{r_a T}}{(r_a T)!} \exp(-f_a(s)T)$$

Decoding – stimulus reconstruction

Want an estimator s_{Bayes} $s \leftarrow r$

Introduce an error function, $L(s, s_{\text{Bayes}})$; minimize error.

$$\frac{\partial}{\partial s_{\text{Bayes}}} \int ds\, L(s, s_{\text{bayes}})\, p[s|\mathbf{r}] = 0$$

For least squares cost, $L(s, s_{\text{Bayes}}) = (s - s_{\text{Bayes}})^2$;

$$\frac{\partial}{\partial s_B} \int ds\, (s - s_B)^2 p(s|r) = 2 \int ds\, (s - s_B)\, p(s|r) = 0$$

$$\int ds\, s\, p(s|r) = \int ds\, s_B$$

Solution: $$s_{\text{bayes}} = \int ds\, p[s|\mathbf{r}]\, s$$

Decoding – stimulus reconstruction

Want an estimator s_{Bayes} $s \leftarrow r$

Introduce an error function, $L(s, s_{Bayes})$; minimize error.

$$\rightarrow \quad \frac{\partial}{\partial s_{Bayes}} \int ds\, L(s, s_{\text{bayes}})\, p[s|\mathbf{r}] = 0$$

For least squares cost, $L(s, s_{Bayes}) = (s - s_{Bayes})^2$;

$$\frac{\partial}{\partial s_B} \int ds\, (s - s_B)^2 p(s|r) = 2 \int ds\, (s - s_B)\, p(s|r) = 0$$

$$\int ds\, s\, p(s|r) = \int ds\, s_B$$

Solution: $$s_{\text{bayes}} = \int ds\, p[s|\mathbf{r}]\, s \quad \leftarrow$$

If response is a single spike -> we just average all stimuli by the probability they appeared given this response

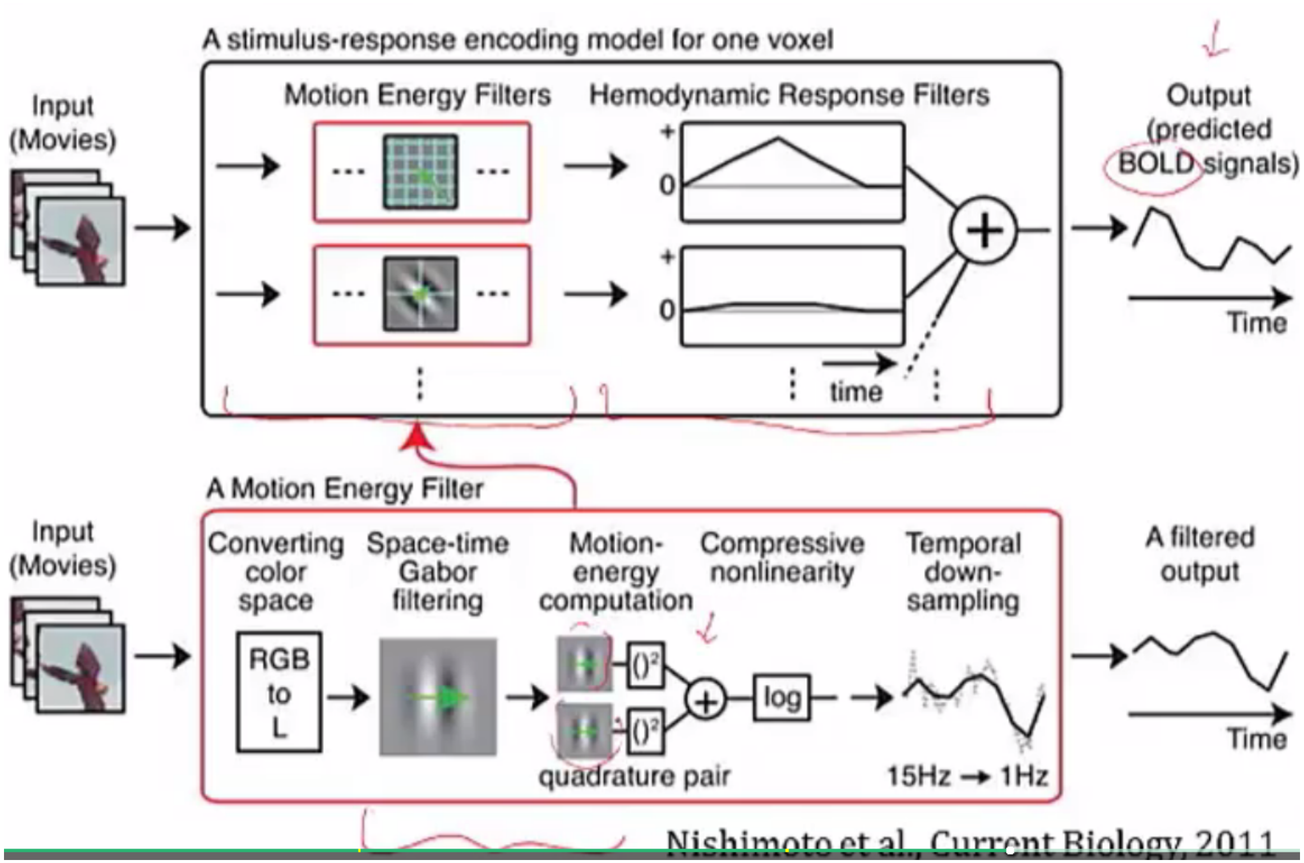
Decoding – stimulus reconstruction

- Yang Dan, Barkley – cat LGN neurons reconstruction

https://www.youtube.com/watch?v=tFdZ9eGTG5A

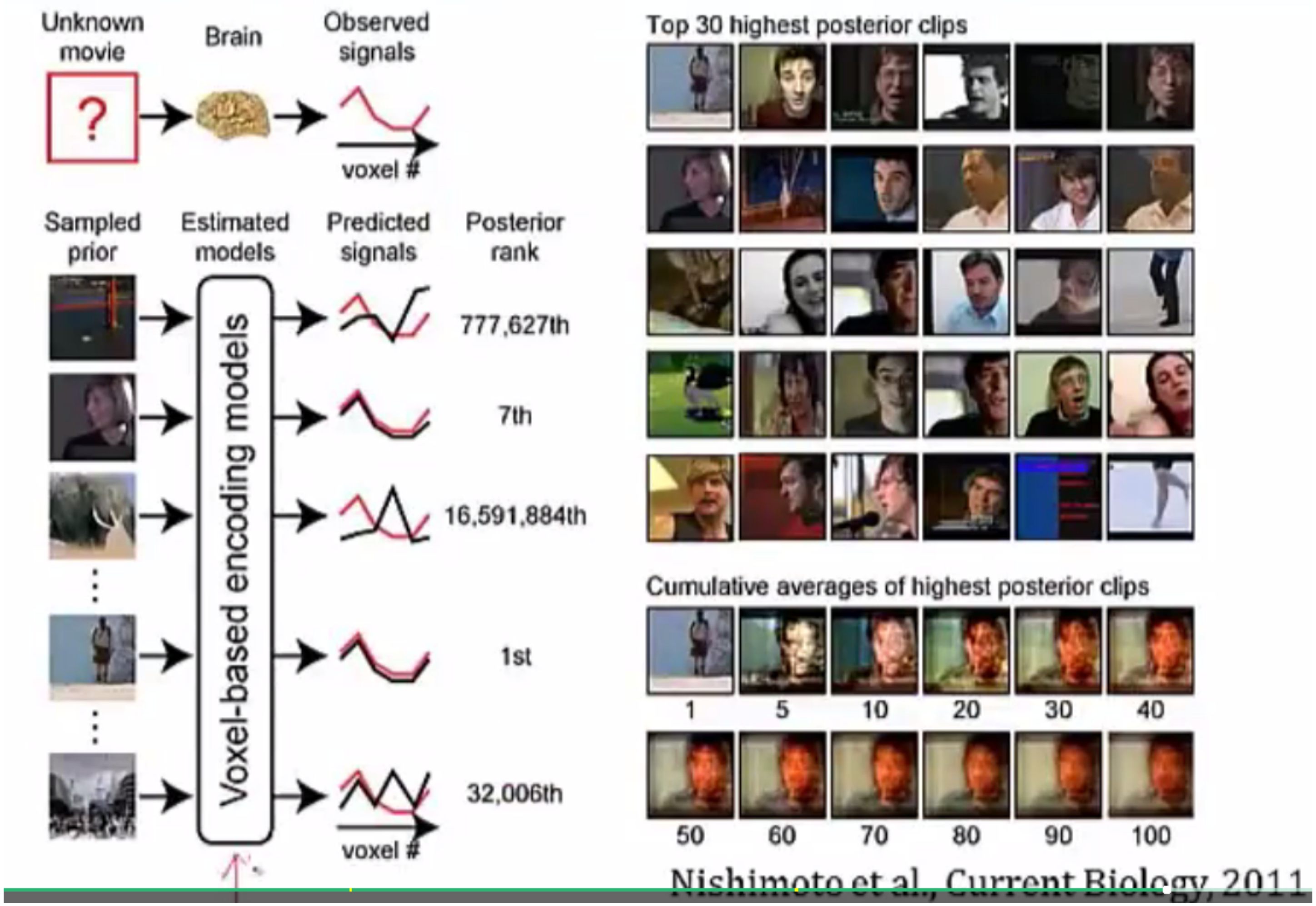
Decoding – stimulus reconstruction

- Nishimoto et al., 2011 (fMRI)



Decoding – stimulus reconstruction

- Nishimoto et al., 2011 (fMRI)



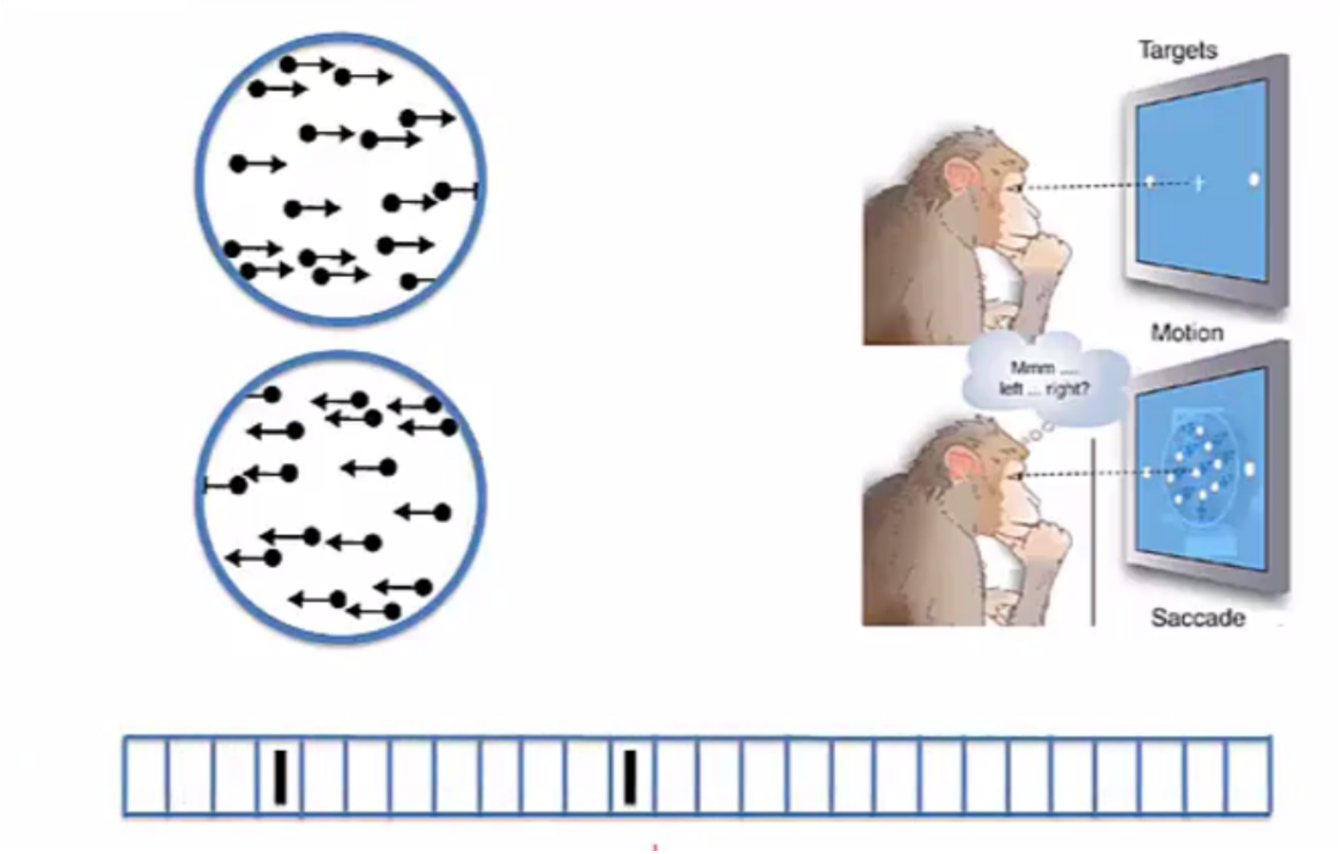
Nishimoto et al., Current Biology, 2011

Information theory

- Information transmission limits
  - Problem sparse representation
  - Entropy and information
  - How information tells us about coding

Information theory

- Information (represents this surprise for monkey)



Information theory

- Information that we get from spike/non-spike

$$P(1) = p$$

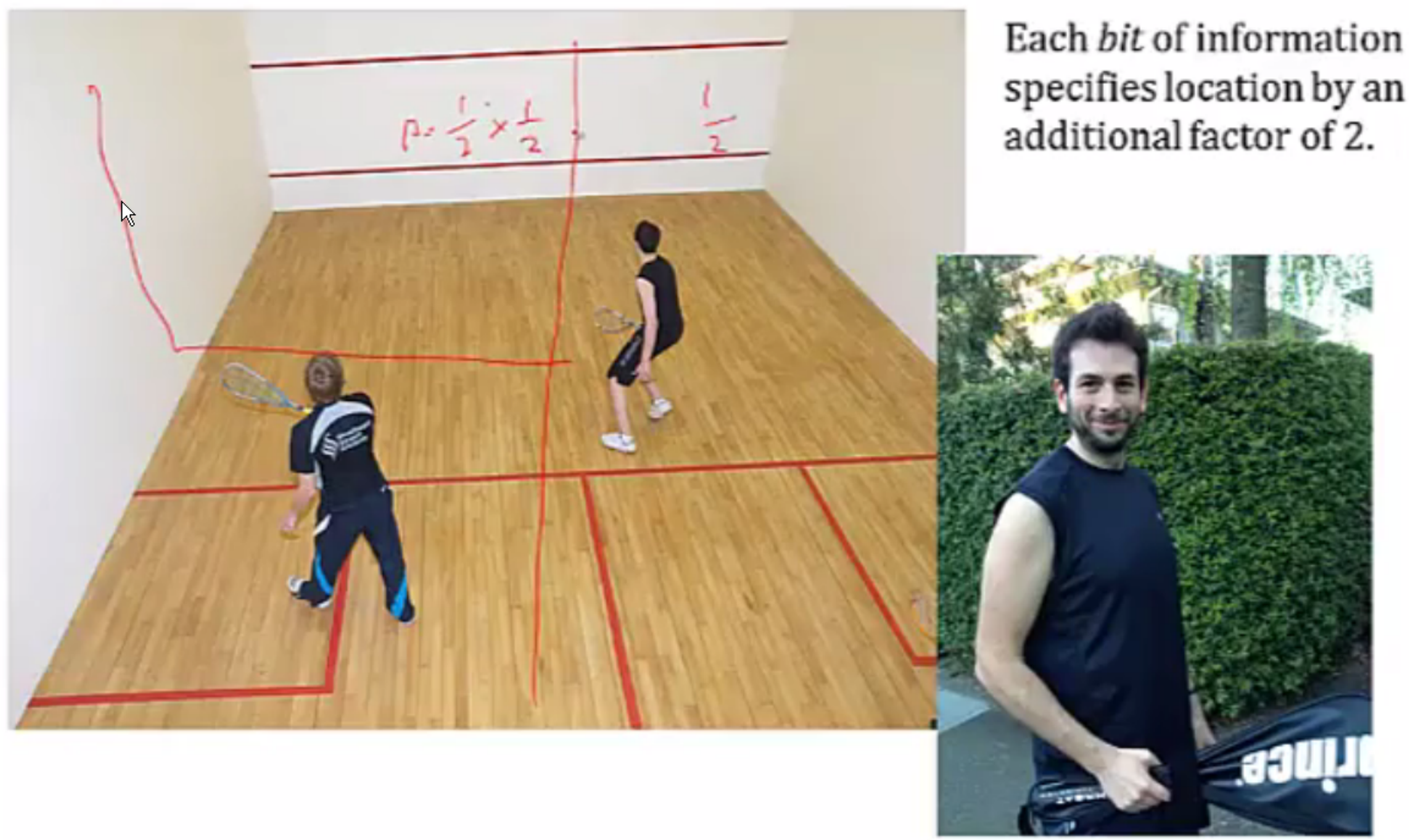
$$P(0) = 1 - p$$

$$\text{information}(1) = -\log_2 p$$

$$\text{information}(0) = -\log_2 (1-p)$$

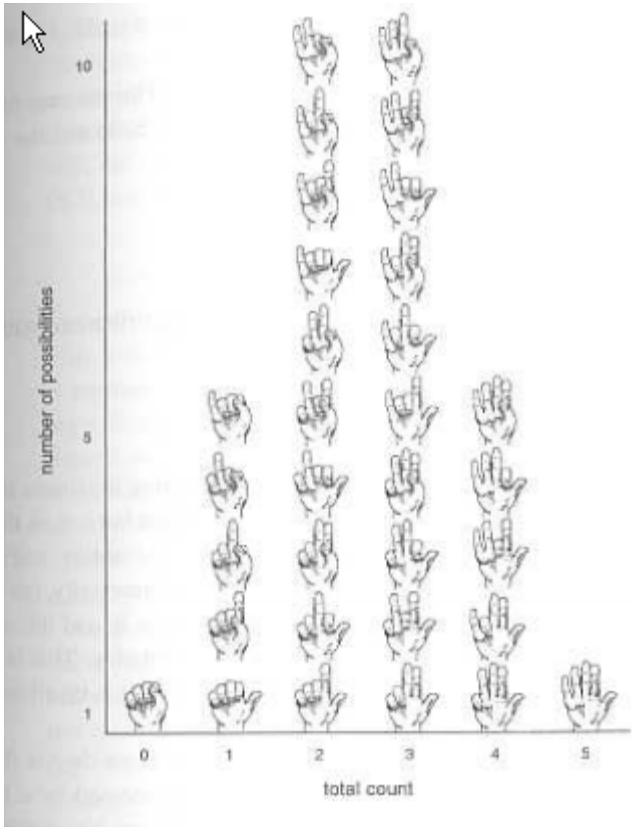
Information theory

- Each bit of information splits space to half



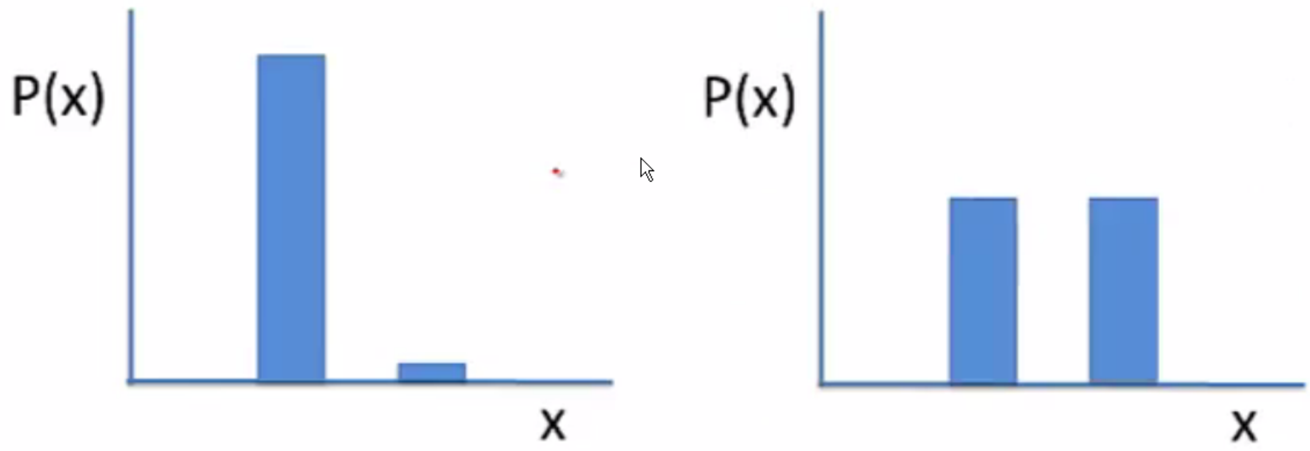
Information theory

- Different encodings



Information theory - Entropy

Entropy – is the expected value (average) of information contained in each message



Entropy = average information

$$= -\sum_i p_i \log_2 p_i$$

$$= -\int dx\, p(x) \log_2 p(x)$$

Units are *bits*

Information theory - Entropy

Entropy – how many questions are needed to find the position of my car?

Information theory - Entropy

Entropy – how many questions are needed to find the position of my car?

$$H = -\sum p_i \log p_i$$

$$p_i = \frac{1}{8}$$

$$H = -\sum_{i=1}^{8} \frac{1}{8} \log_2\left(\frac{1}{8}\right)$$

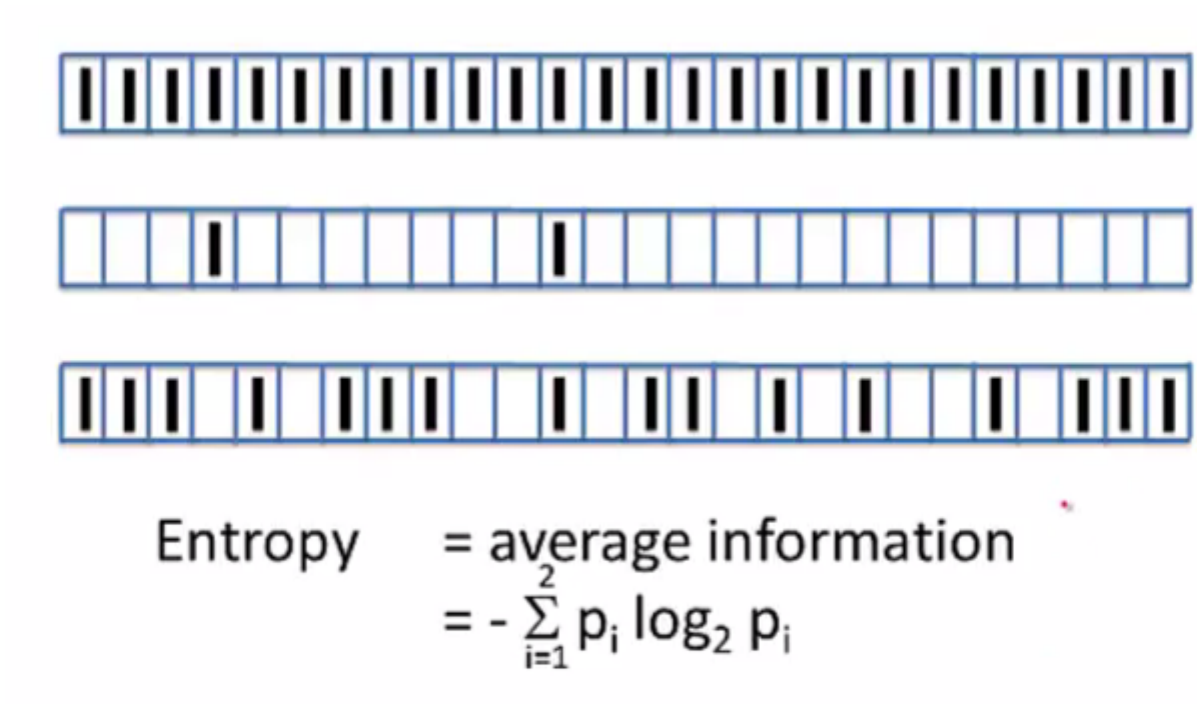
$$= \sum \frac{1}{8} \cdot (-3) \qquad 8 = 2^3$$

$$3$$

$$= 3$$

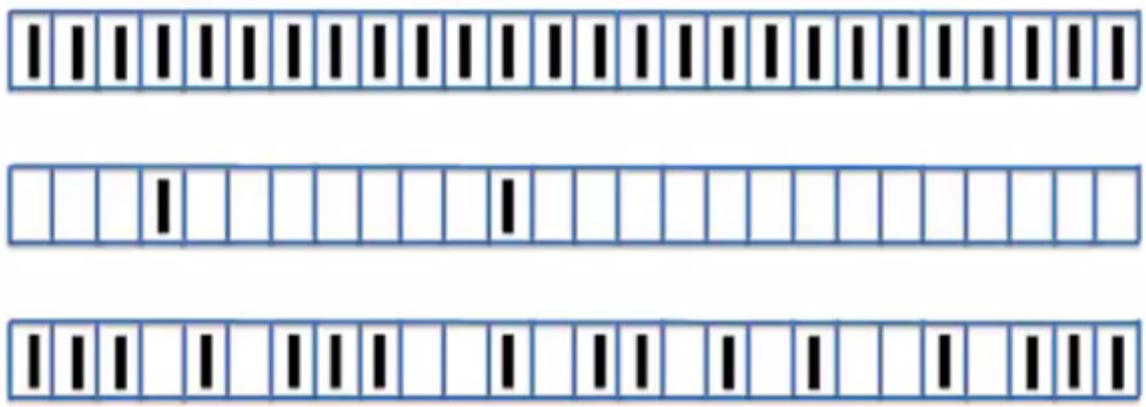
Information theory - Entropy

- Which of the messages/encoding can contain the most information?
- Not varying-> does not provide much information…



Information theory - Entropy

- Large entropy -> lot of posibility for representing inputs

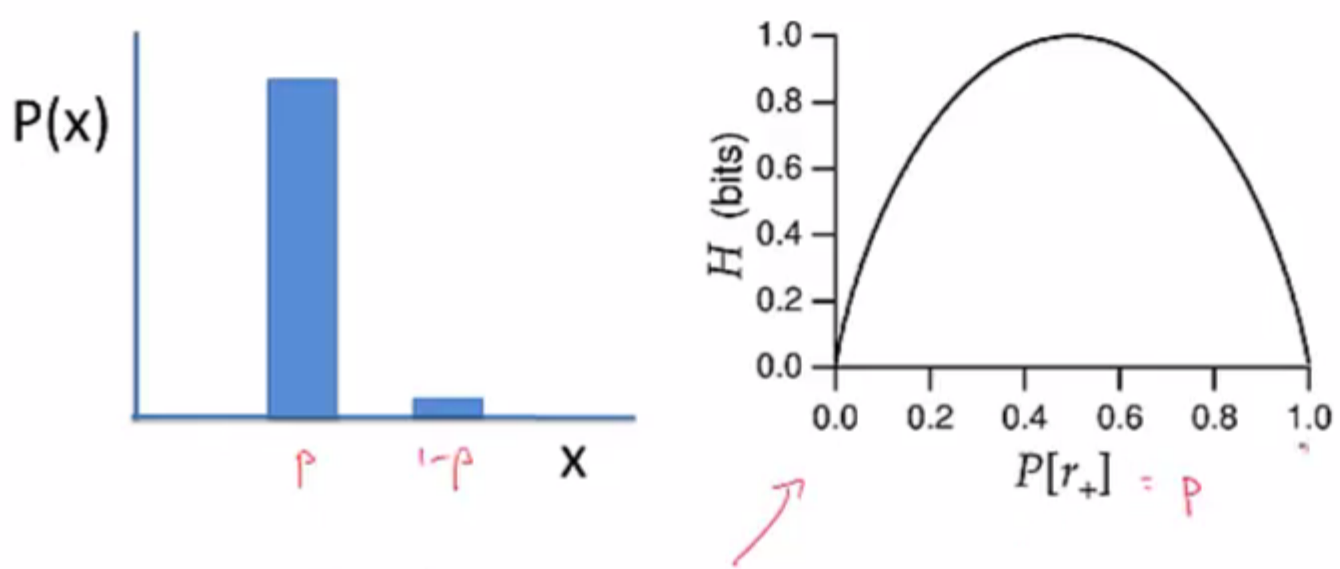


Entropy = average information

$$= -\sum_{i=1}^{2} p_i \log_2 p_i$$

Information theory - Entropy

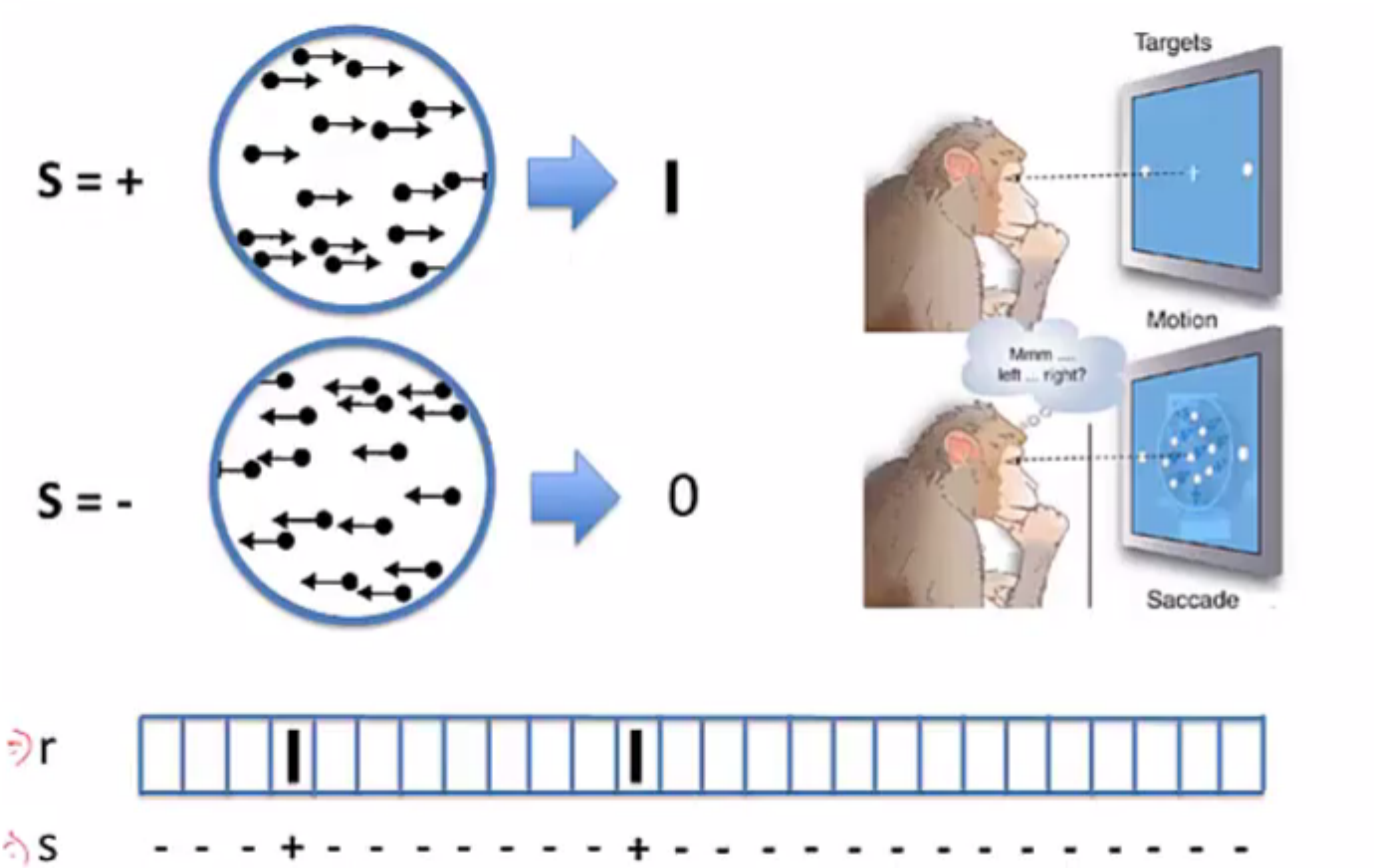
- Maximum entropy for p = ½ -> would be best if uniform distribution (if we could place ball in squash to any place then we need least bits to encode the information)



$$\text{Entropy} = -\sum p_i \log_2 p_i$$

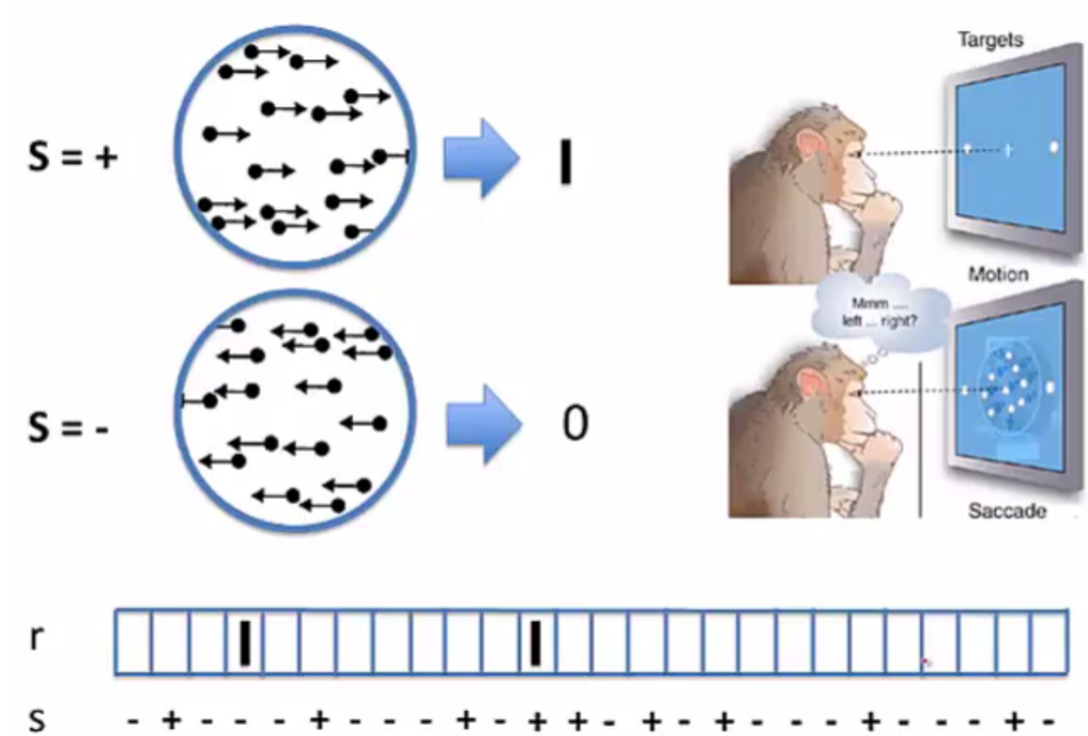
$$= -p\log p - (1-p)\log(1-p)$$

Information theory - Entropy



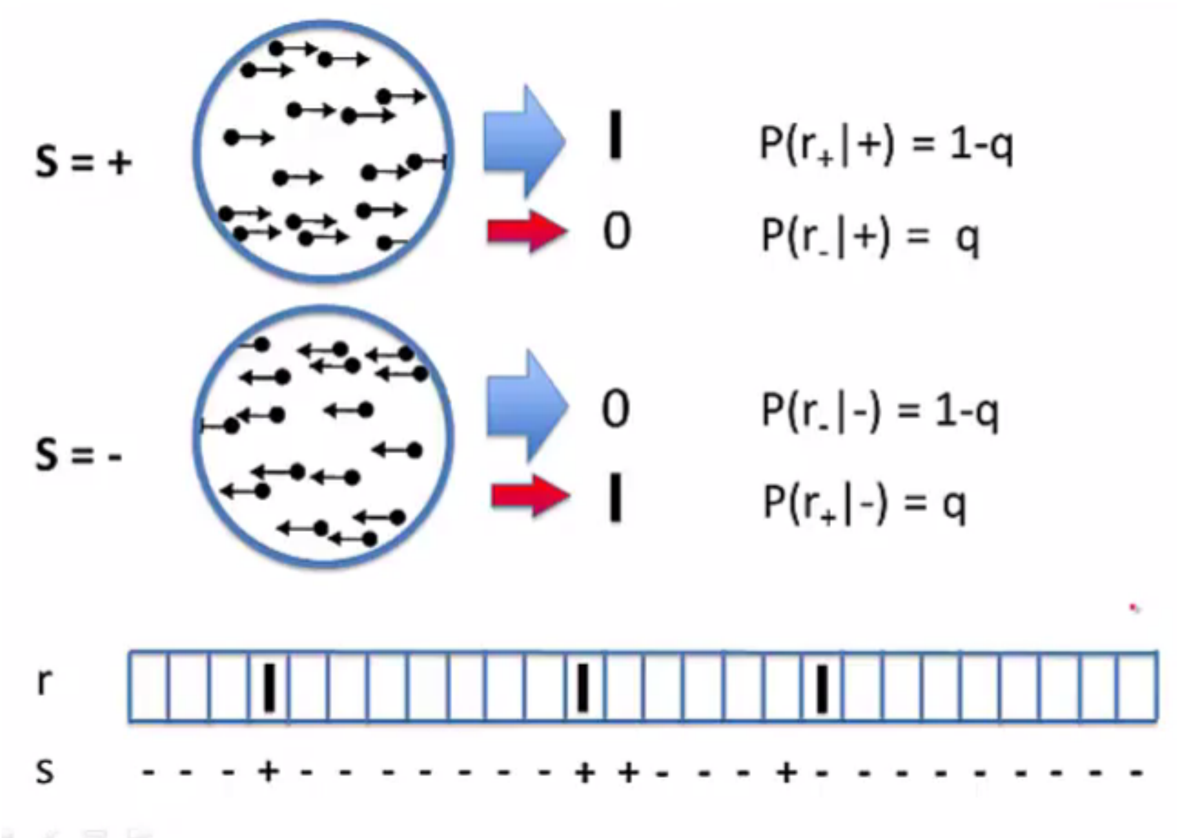
Information theory - Entropy

- Either:
  - Neuron not responding to stimulus
  - bad feature
  - didn't understand what code is doing



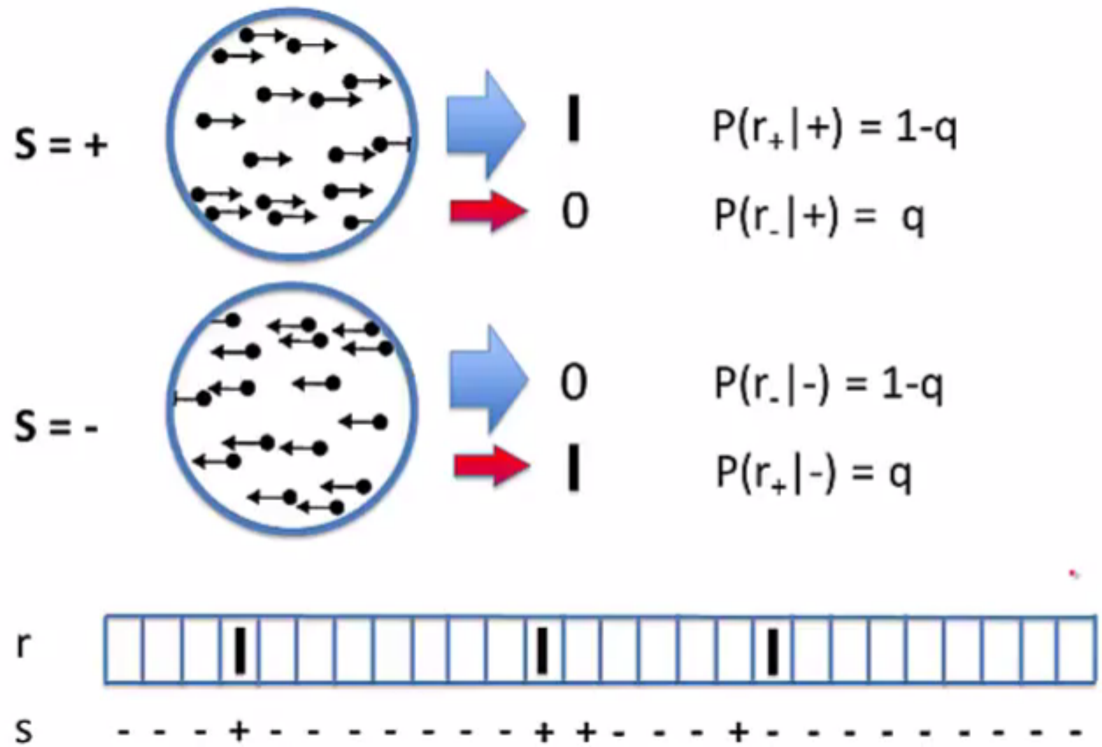
Information theory - Entropy

- How much variablity in r is encoding s
  - Posibility of error



Information theory - Entropy

- How much entropy accounts for these errors (noise)



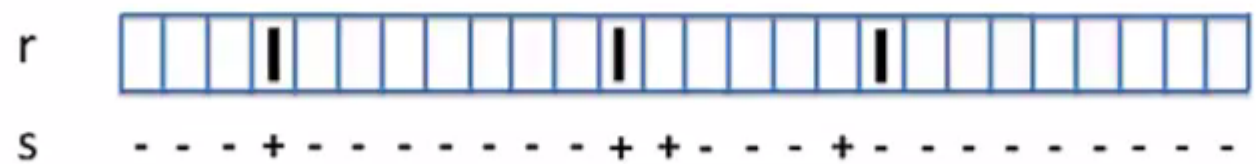
$$H = -\sum_{i=1}^{2} p_i \log_2 p_i$$

Total entropy: $H[R] = - P(r_+) \log P(r_+) - P(r_-) \log P(r_-)$

Noise entropy: $H[R|+] = - q \log q - (1-q) \log (1-q)$

Information theory – Mutual information

- The amount of information that the response carries about the stimulus



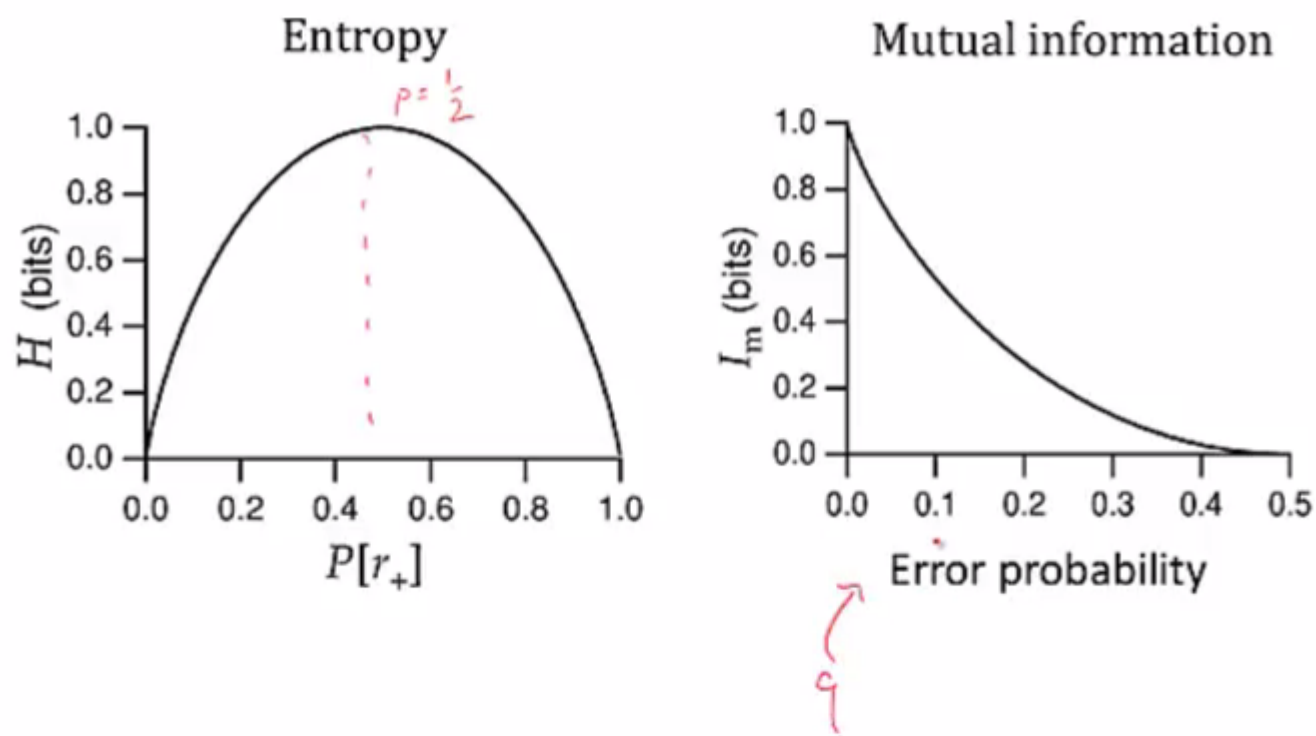
The amount of entropy that is used in coding the stimulus

MI(S,R) = Total entropy – average noise entropy

$$MI = -\sum_r p(r)\log_2 p(r) - \sum_s p(s)[-\Sigma_r\, p(r|s)\log_2 p(r|s)]$$

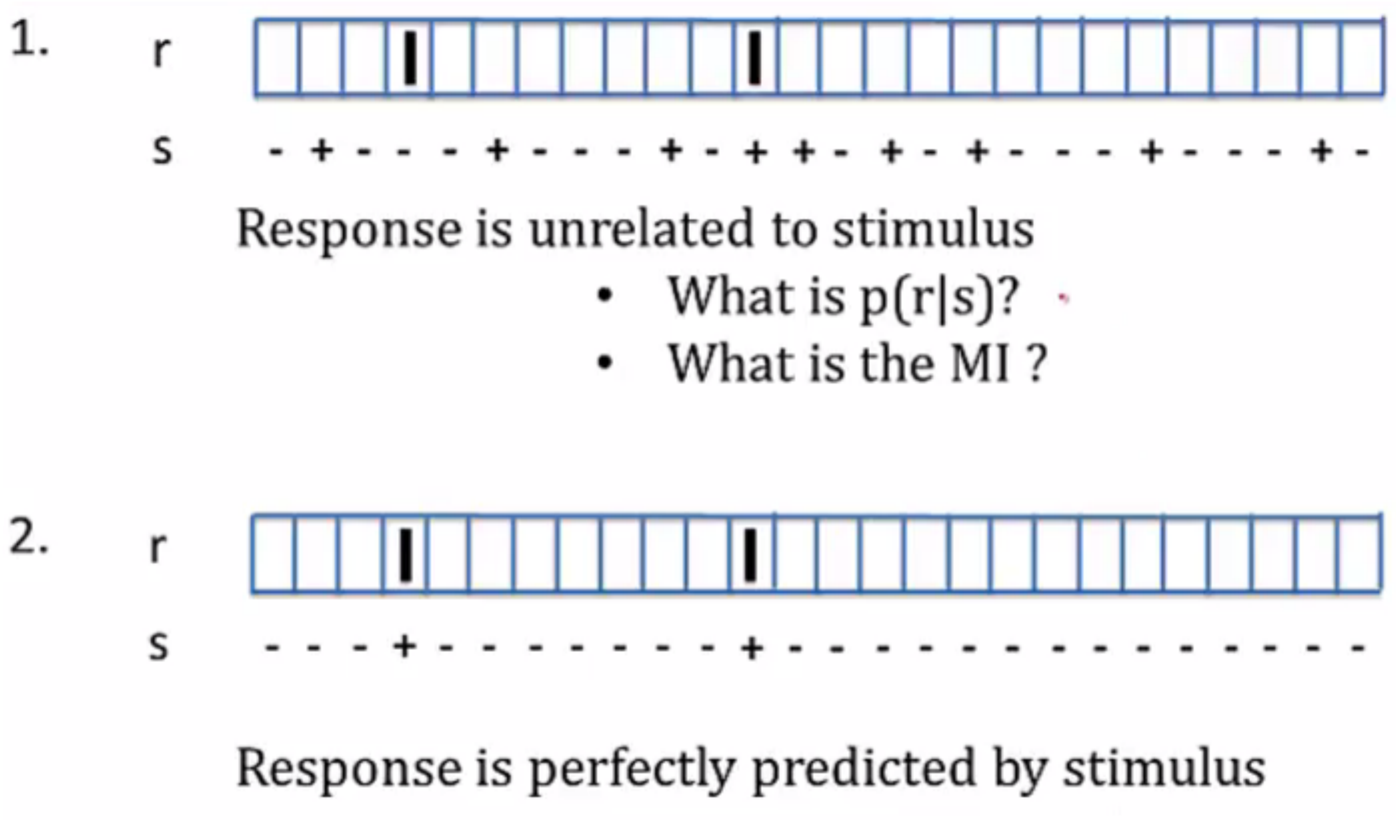
Information theory – Mutual information

- The amount of information that the response carries about the stimulus



Information theory – Mutual information

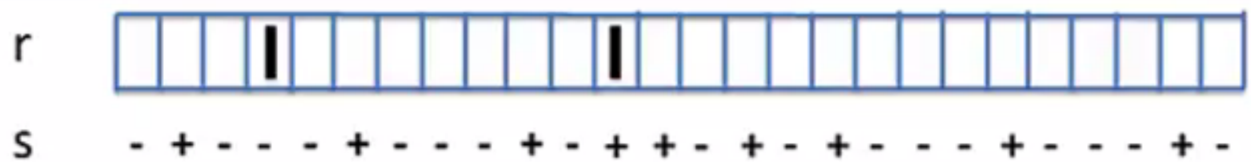
- The amount of information that the response carries about the stimulus



Information theory – Mutual information

- The amount of information that the response carries about the stimulus

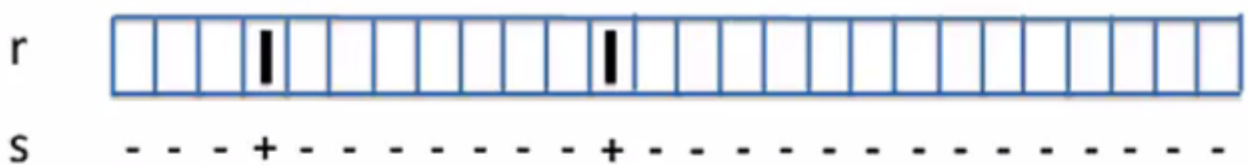
1.



Response is unrelated to stimulus

- What is p(r|s)? = p(r)
- What is the MI ? =0

2.



Response is perfectly predicted by stimulus

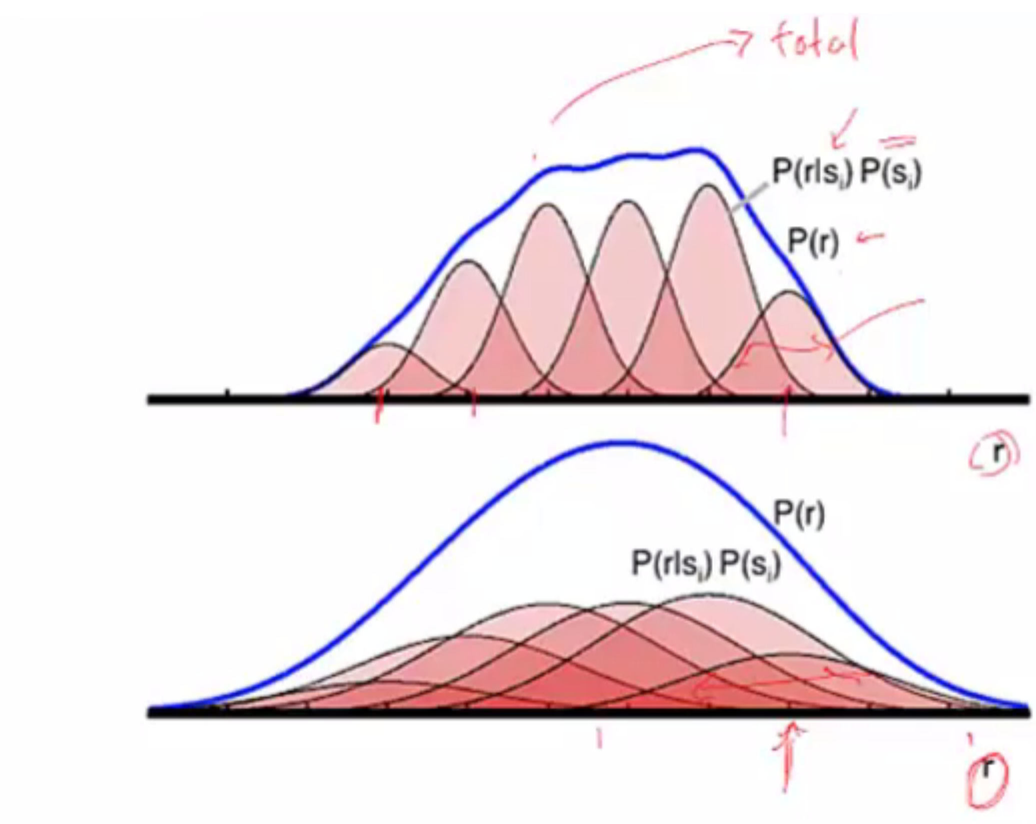
Noise entropy = 0 -> MI = total entropy of response

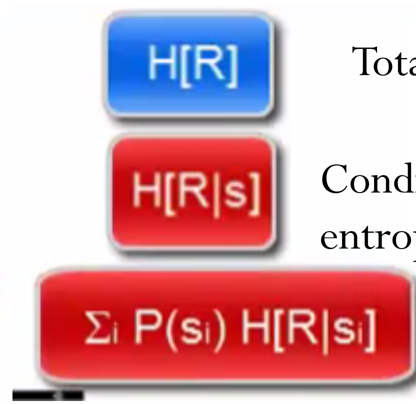
Information theory - Entropy

- Information summarizes how much independent two variables are
- Encoding
  - Most coherent form
  - Maximizing entropy

Information theory – Continuous variables

- Which distribution encodes more information about stimulus?





Total entropy

Conditional entropy

Average cond. entropies over stimulus they drove them

Information theory – Information in spikes

- Information in spike pattern/trains
- Information in single spikes

Information theory – Information in spikes

- Information in spike pattern/trains
- Information in single spikes

Mutual information is the difference between the total response entropy and the mean noise entropy:

$$I(S;R) = H[R] - \Sigma_s P(s)\, H[R|s)] \,.$$

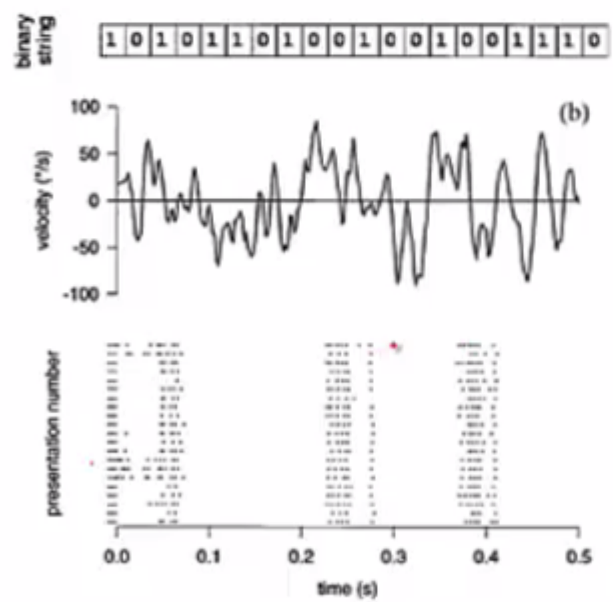
- Take one stimulus s and repeat many times -> p(R | s)
- Compute variablity due to noise: noiseentropy H[R | s]
- Repeat for all s and average: $\Sigma_s P(s)\, H[R|s)]$
- Compute $P(R) = \Sigma_s P(s)\, P(R|s)$ and the total entropy H[R]

Information theory – Information in spikes

So far only dealt with single spikes, or firing rates.

What information is carried by patterns of spikes?

Analyze patterns of the code: how informative are they?

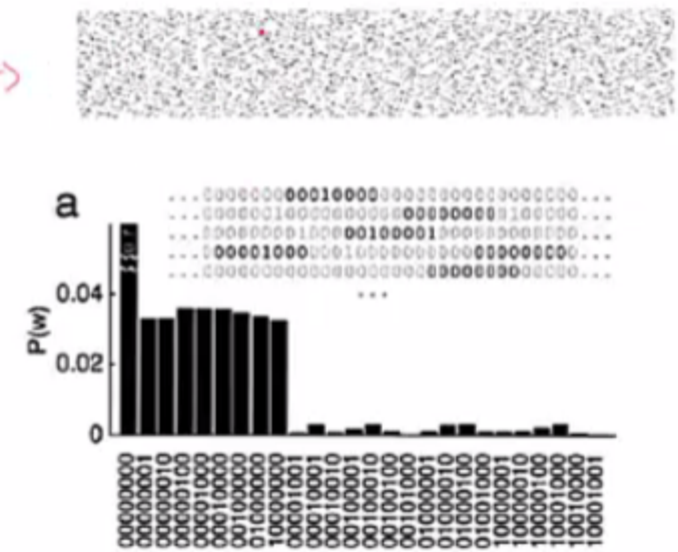


Strong et al., 1997

Information theory – Information in spikes

Entropy:

- Binary words w with letter size Δt, length T.
- Compute $p(w_i)$



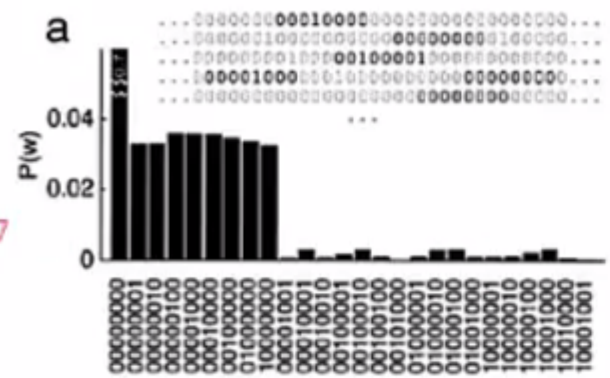
$$H[w] = -\sum p(w_i)\log_2 p(w_i)$$

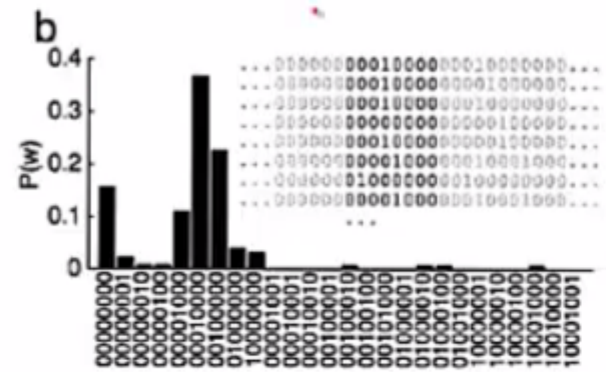
Entropy of that word distribution

Strong et al., 1997; Reinagel and Reid, 2000

Information theory – Information in spikes

Information :
difference between the total variability driven by stimuli and that due to noise, averaged over stimuli.





Information theory – Information in spikes

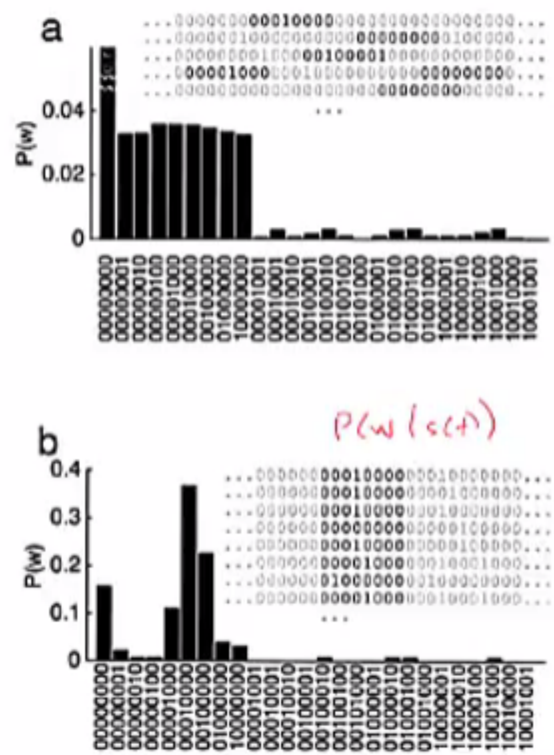
Take a stimulus sequence and repeat many times.

How to sample P(*s*)?
Average over *s* → average over time:

For each time in the repeated stimulus, get a set of words $P(w|s(t))$.

$$H_{noise} = \langle H[P(w|s_i)] \rangle_i .$$

Choose length of repeated sequence long enough to sample the noise entropy adequately.



Information theory – Information in spikes

Take a stimulus sequence and repeat many times.

How to sample P(*s*)?
Average over *s* → average over time:

For each time in the repeated stimulus, get a set of words P(*w*|*s*(t)).

$$H_{noise} = \langle H[P(w|s_i)] \rangle_i.$$

Choose length of repeated sequence long enough to sample the noise entropy adequately.