339.623
Computational Neuroscience and Neuroinformatics

Prof. Marcus Kaiser

March 29, 2010

Week 5: Information theory and neural coding (Appendix D)

Computational Neuroscience

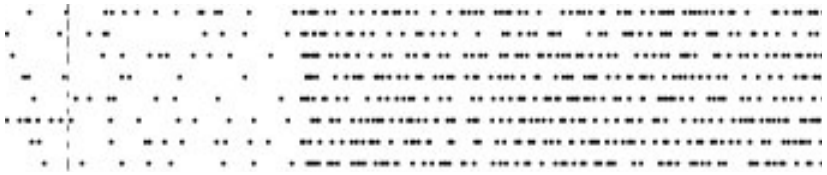
- Simulation e.g. simulating single neurons or sets of neurons (population model, IF neurons)

- Modeling understanding the function of neural circuits.

- Analysis of brain connectivity using computational tools to analyze the structure of neural networks. This analysis can give insights about function and dynamics!

- Analysis of brain dynamics using computational tools to analyze experimental or *in silico* brain activity. This analysis can lead to improved models and simulations!

Neural coding



- Rate coding features of an object are encoded through neural firing rate (e.g. orientation tuning).

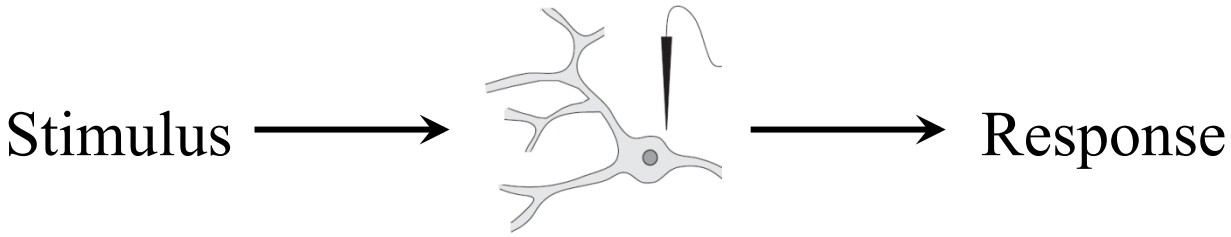
- Population coding simultaneous firing of a population of neurons encodes for an object or feature of an object.

- Sequence coding the firing pattern of a neuron contains information about an object.

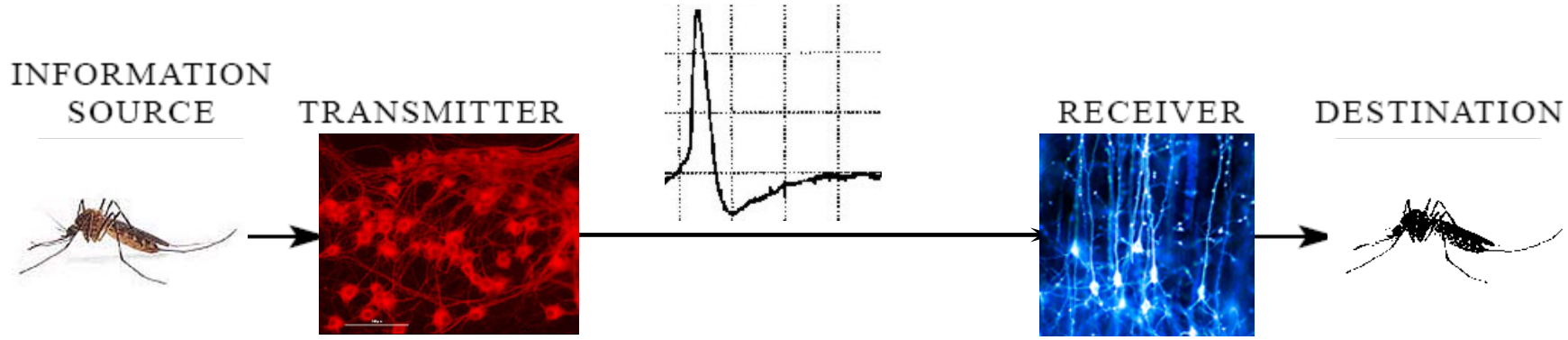
Modeling brain processes

Psychophysics

Neurophysiology



Processing steps



Stimulus

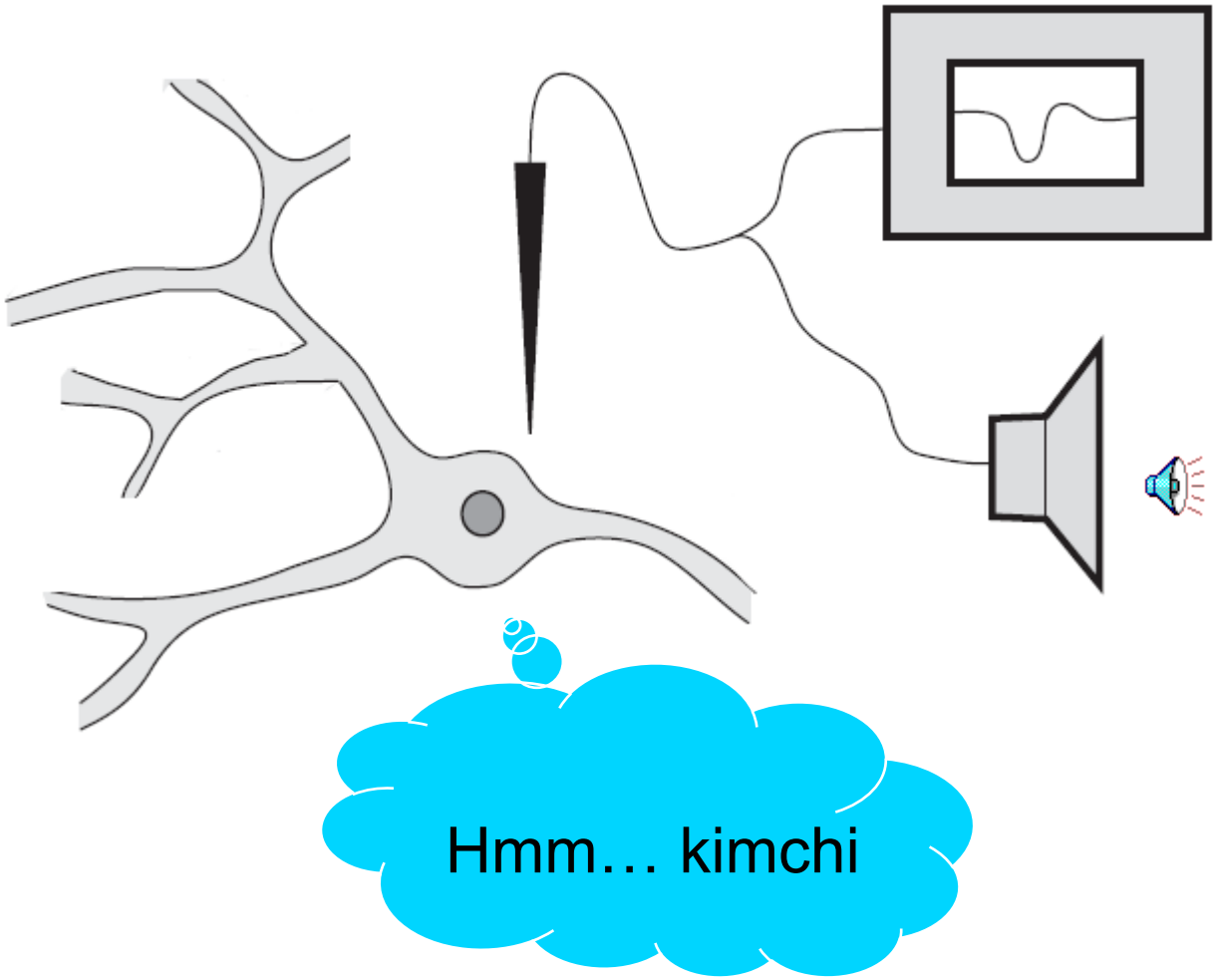
Sensory encoding

Signal

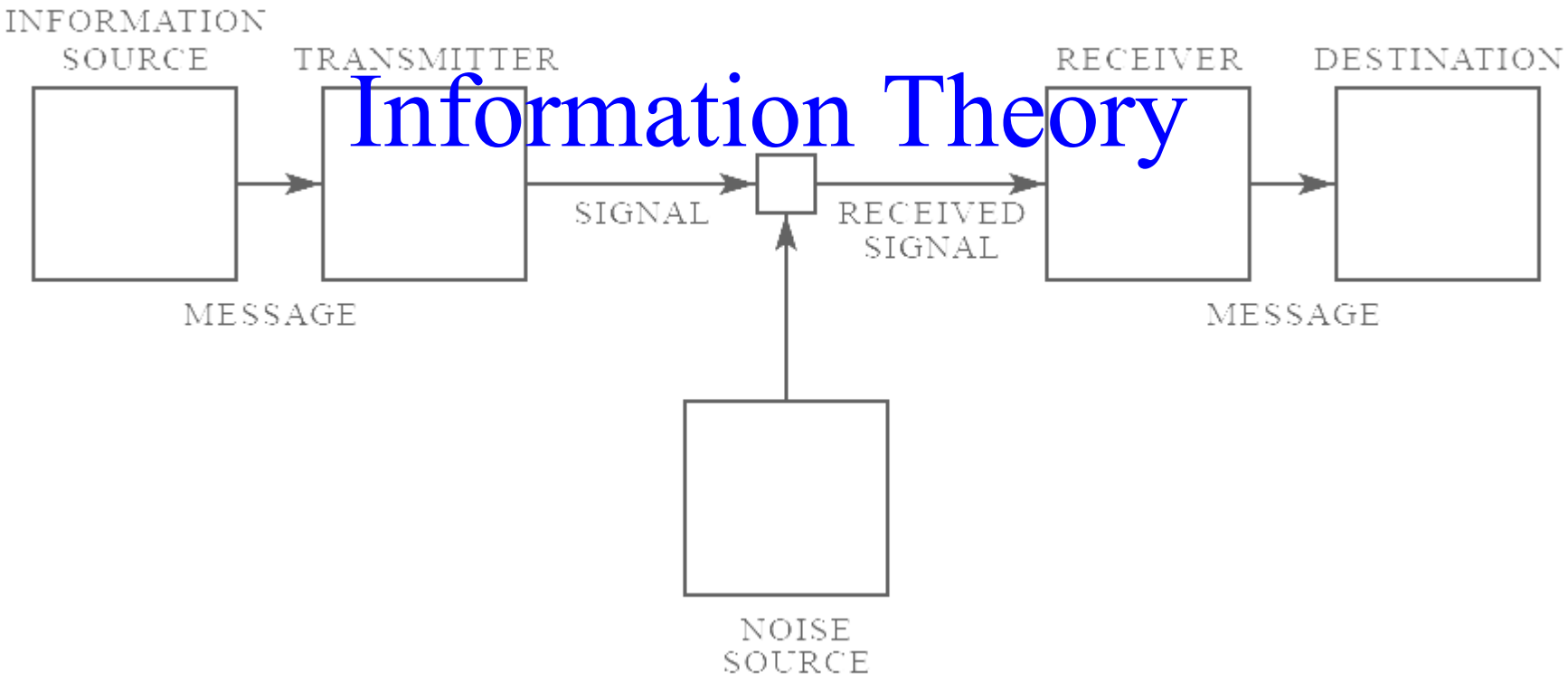
Decoding

Perception

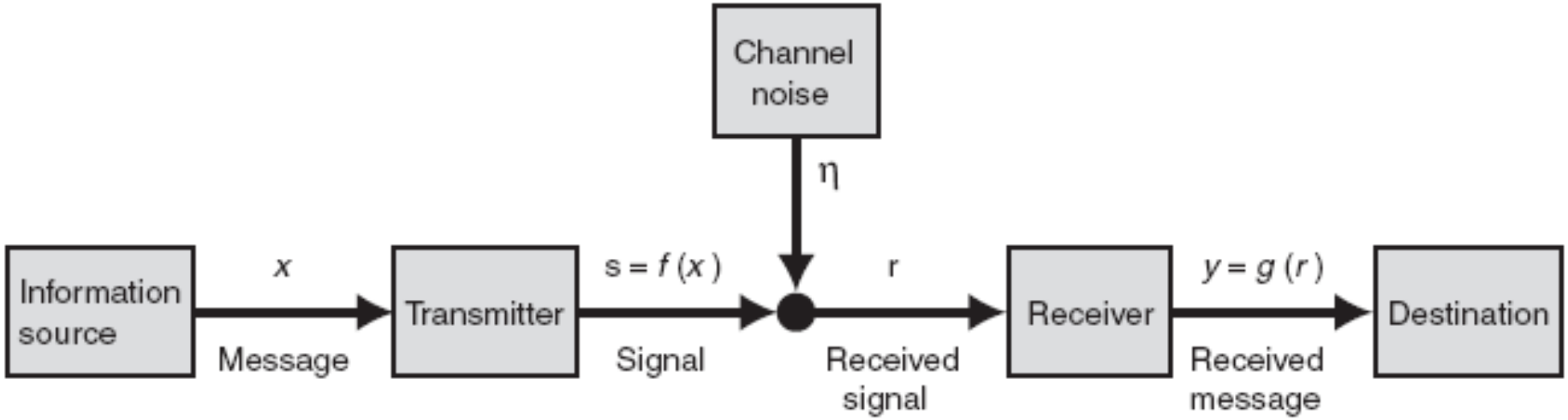
What the brain ‘sees’



Information Theory

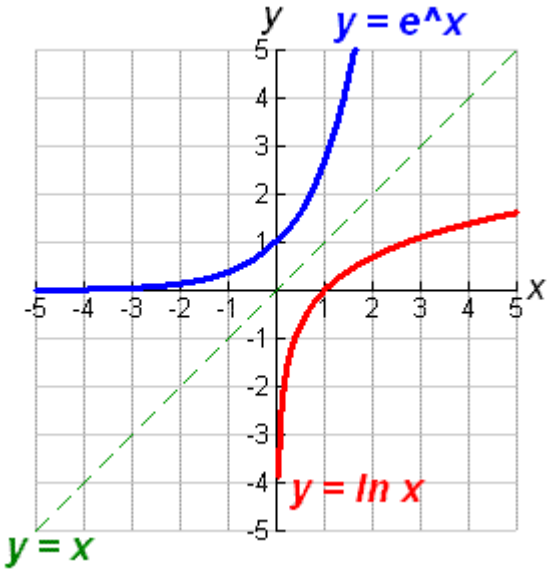


Information Theory



Claude E. Shannon (1948) A Mathematical Theory of Communication, Bell Sys. Tech. J. 27:379-423

Information gain *I*



Information unit [bit(s)]

Information gain after receiving message x

$$I(x) = -\log_2 P(x)$$

P(x): probability to receive message x (why minus?)

Note: instead of $\log_2$ the notation ld (logarithmus digitalis) can also be used

Examples: equal message probabilities

Two messages with equal probability

P(0) = 0.5 P(1) = 0.5

I(0) = I(1) = - $\log_2$ 0.5 = 1 bit

Three messages with equal probability

P(0) = 1/3 P(1) = 1/3 P(2) = 1/3

I(0) = I(1) = I(2) = - $\log_2$ 0.333 = 1.585 bits

Information gain (general case)

What if a message is uncertain?

Example: previous knowledge for coin tossing:
p(heads) = 0.5 (prior knowledge)

message that heads occurs with 90% probability
p(heads) = 0.9 (posterior knowledge after message receipt)

$$I(x) = -P^{posterior}(x) \log_2 \frac{P^{prior}(x)}{P^{posterior}(x)}$$

$P^{posterior}$: probability of message x after the message is sent

P^{prior}: probability of message x before the message is sent (also called *a priori* probability)

Average information gain: Entropy

- Entropy is defined in terms of probabilistic behavior of a source of information
- In information theory the source output are discrete random variables that have a certain fixed finite alphabet with certain probabilities
  - **Entropy is the *average* information content for the given source symbol**

Entropy *S* *(or H)*

average information gain =
sum of information gains weighted
by their probability

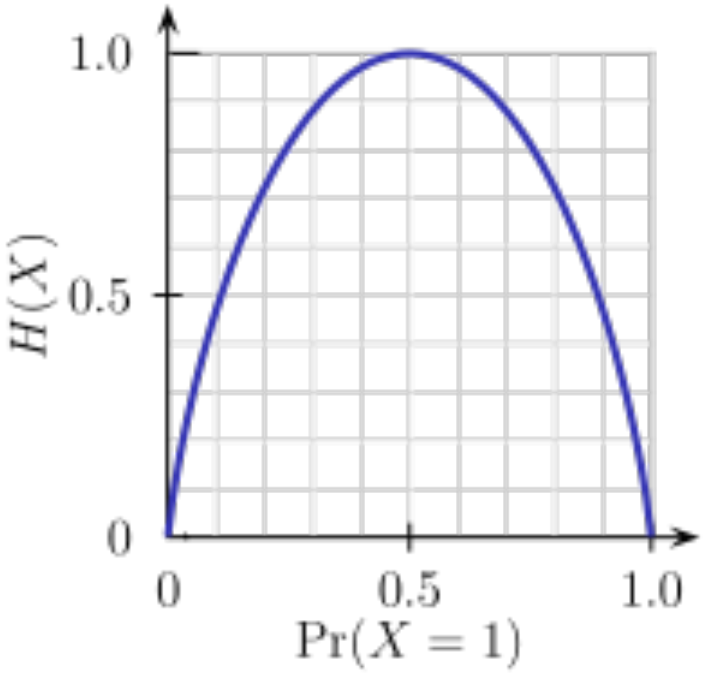
$$S(x) = -\sum_i p_i \log_2(p_i)$$

Entropy with N equally likely messages:

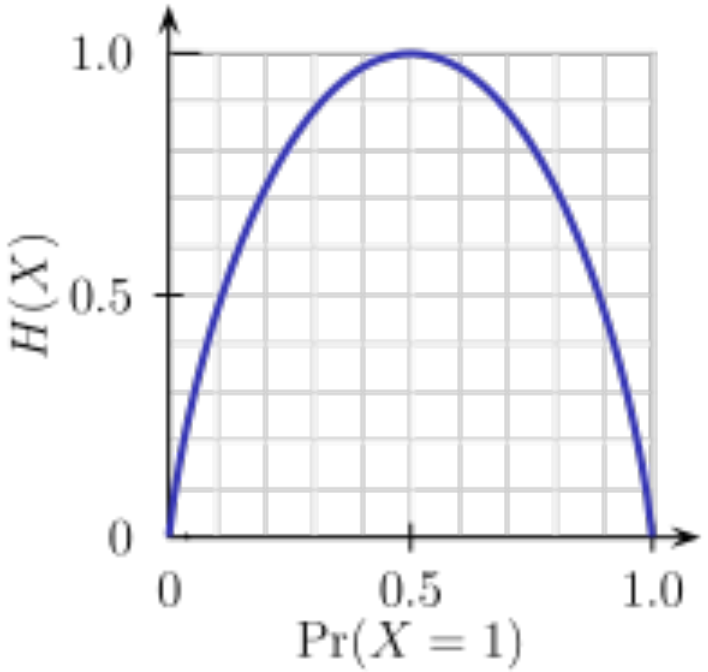
$$S(x) = -\sum_{i=1}^{N} \frac{1}{N} \log_2(\frac{1}{N}) = \log_2(N)$$

Note: Rare events have a huge impact on the entropy! That means, a large data set is needed for accurate values. For normal experiments with a small sample size, the entropy is underestimated as rare events are missing.

Example

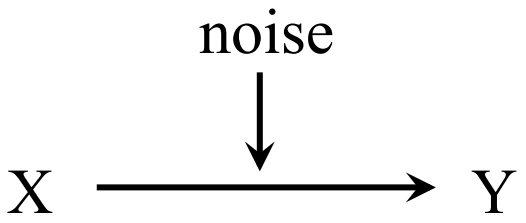


- Entropy of a Coin toss relative to the probability of showing heads.
- Consider tossing a coin with known, not necessarily fair, probabilities of coming up heads or tails.
- The entropy of the unknown result of the next toss of the coin is maximised if the coin is fair (that is, if heads and tails both have equal probability 1/2). This is the situation of maximum uncertainty as it is most difficult to predict the outcome of the next toss; the result of each toss of the coin delivers a full 1 bit of information.



- However, if we know the coin is not fair, but comes up heads or tails with probabilities p and q, then there is less uncertainty. Every time, one side is more likely to come up than the other. The reduced uncertainty is quantified in a lower entropy: on average each toss of the coin delivers less than a full 1 bit of information.
- The extreme case is that of a double-headed coin which never comes up tails. Then there is no uncertainty. The entropy is zero: each toss of the coin delivers no information.

Mutual Information



- The mutual information of two random variables is a quantity that measures the mutual dependence of the two variables. The most common unit of measurement of mutual information is the bit, when logarithms to the base 2 are used.

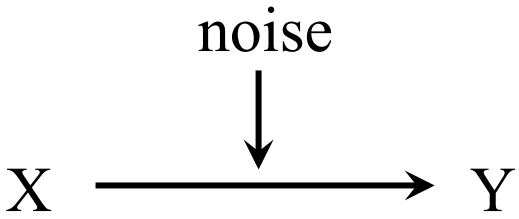
- Mutual information is a useful concept to measure the amount of information shared between input and output of noisy channels.

Mutual Information

- Mutual information measures the information that X and Y share: it measures how much knowing one of these variables reduces our uncertainty about the other. For example, if X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero.

- At the other extreme, if X and Y are identical then all information conveyed by X is shared with Y: knowing X determines the value of Y and vice versa. As a result, the mutual information is the same as the uncertainty contained in Y (or X) alone, namely the entropy of Y (or X: clearly if X and Y are identical they have equal entropy).

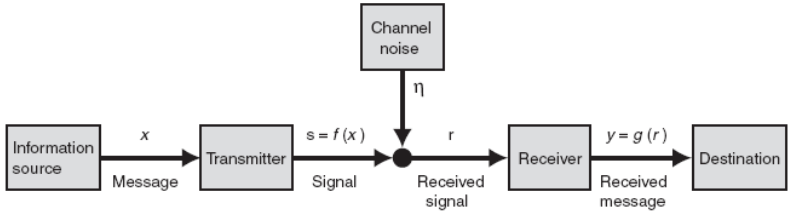
Mutual information



Mutual information ~~(or cross-entropy)~~ = information gain by receiving message Y when a signal X was sent

$$I^{mutual}(X, Y) = S(X) + S(Y) - S(X, Y)$$

Channel Capacity



How much information can be transmitted?

Depends on

- number of possible signal states
- amount of noise

$$I \leq \frac{1}{2}\log_2(1+\frac{\langle x^2\rangle}{\langle \eta^2\rangle})$$ Channel capacity as upper limit

Signal to noise ratio (SNR) =

variance of signal / variance of noise = $\frac{\langle x^2\rangle}{\langle \eta^2\rangle}$

Side note: Efficient coding

Use shorter messages for transmitting messages that are frequent or important.

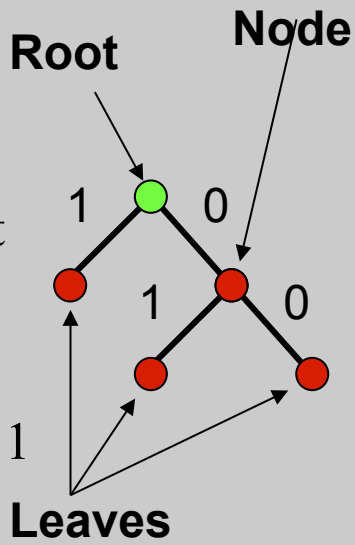
Example: time needed for articulating words (~word length)

frequent: yes, no, a, the, and, or

important: help, fire

Huffman Coding Algorithm

- **Encoding algorithm**
  - Order the symbols by decreasing probabilities
  - Starting from the bottom, assign **0** to the least probable symbol and **1** to the next least probable symbol
  - Combine the two least probable symbols into one composite symbol
  - Reorder the list with the composite symbol
  - Repeat Step 2 until only two symbols remain in the list
- **Huffman tree**
  - Nodes: symbols or composite symbols
  - Branches: from each node, 0 defines one branch while 1 defines the other
- **Decoding algorithm**
  - Start at the root, follow the branches based on the bits received
  - When a leaf is reached, a symbol is decoded



Huffman Coding Example

Symbols	Prob.	
A	0.35	
B	0.17	
C	0.17	
D	0.16	1
E	0.15	0

Symbols	Prob.	
A	0.35	
DE	0.31	
B	0.17	1
C	0.17	0

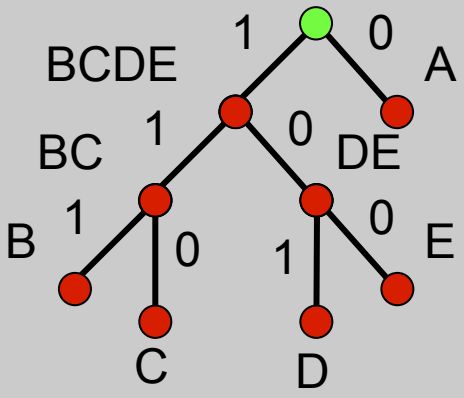
Symbols	Prob.	
A	0.35	
BC	0.34	1
DE	0.31	0

Symbols	Prob.	
BCDE	0.65	1
A	0.35	0

Huffman Codes

A	0
B	111
C	110
D	101
E	100

Huffman Tree



Average code-word length = 0.35 x 1 + 0.65 x 3 = 2.30 bits per symbol

Examples for fast coding in neuroscience

Anatomy: Fibres that transmit pain information have faster conduction velocities (better myelination or larger thickness for non-myelinated fibres)

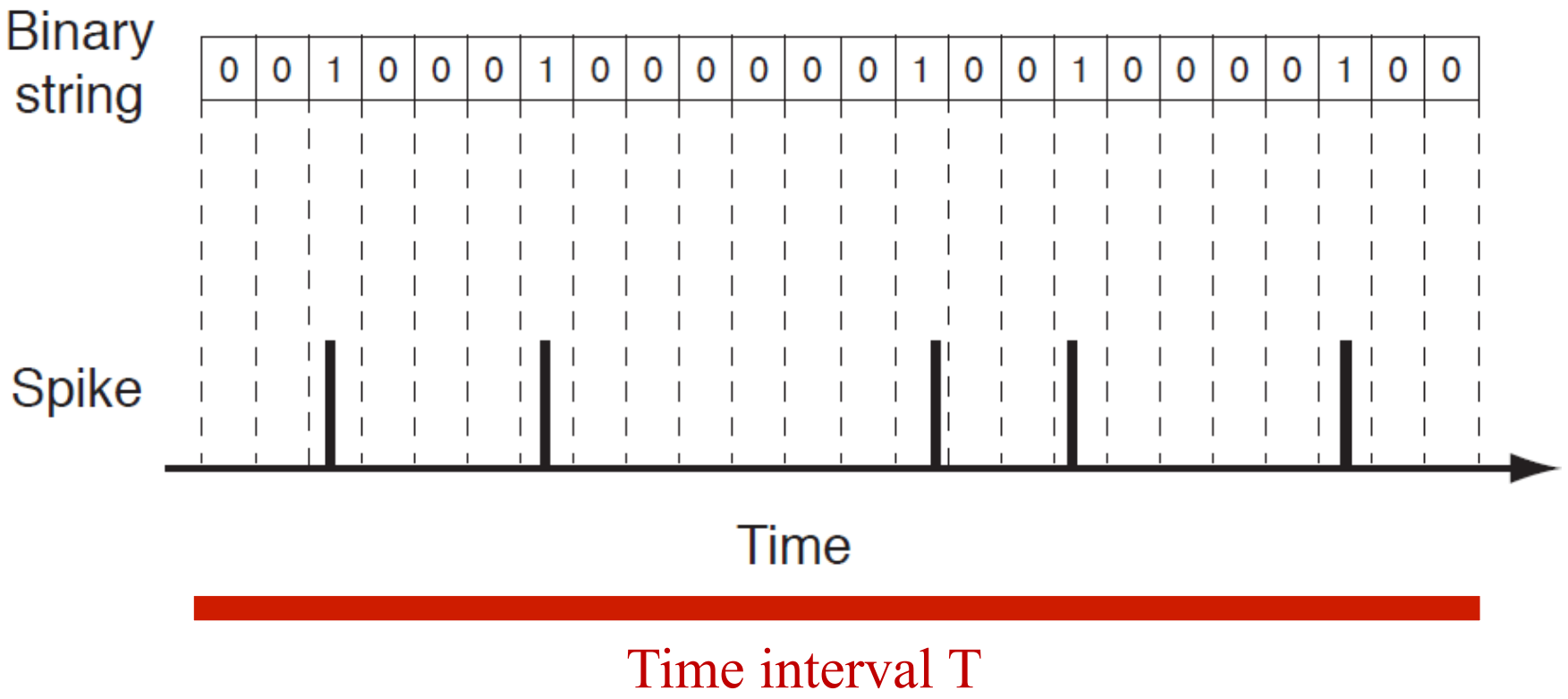
EEG: responses which are important are processed faster (snake vs. table). Attention can also help to speed up processing.

Spike train analysis

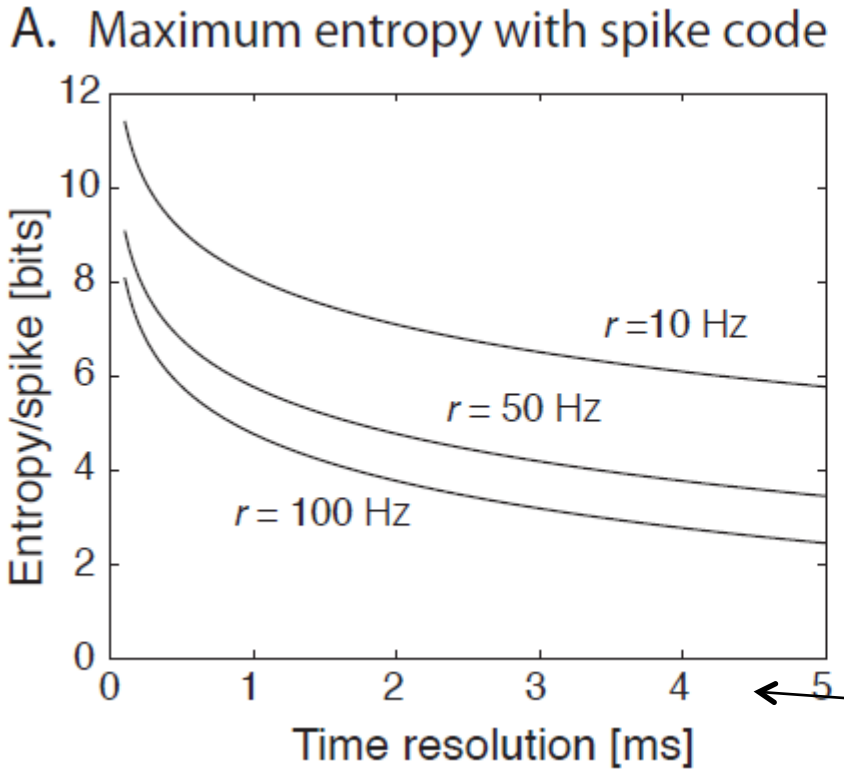
Spike train -> message

Translation: Spike(s) or no spike for each time window

Time resolution: width of time window Δt [ms]



How much information does one spike on average transmit?



Low firing rate -> spikes are a rare event -> each spike carries a high entropy

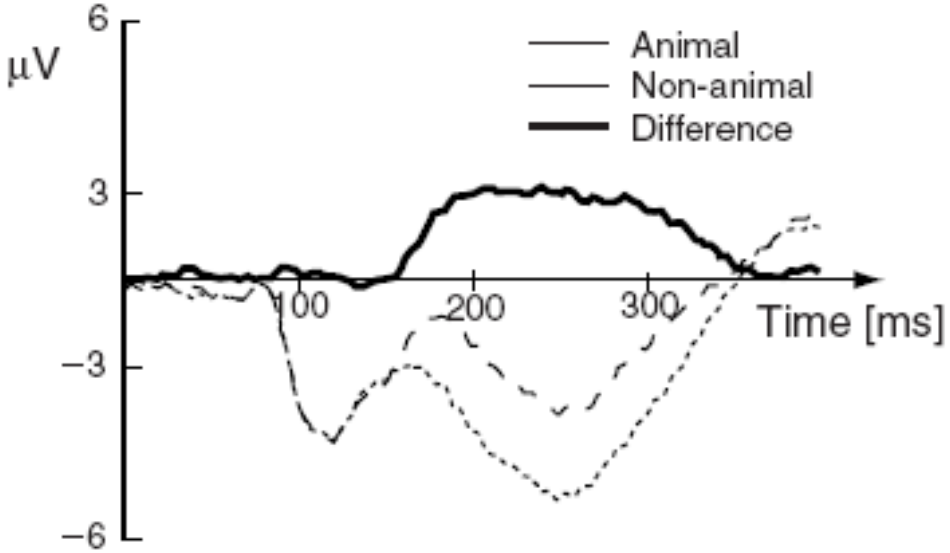
High firing rate -> spikes are more expected -> each spike carries a lower entropy

Low time resolution -> fewer patterns can be encoded -> less information/entropy

-> low firing rate saves energy and increases entropy/spike

Data transmission speed

How many spikes are needed to encode information?



Task: Detection of animals in a visual scene presented for 20 ms.

EEG recordings with 15 human subjects. Recognition after only 150 ms!

(Thorpe et al., Nature, 1996)

From spikes to perception (decoding stimulus features)

Some statistics

The probability that an event *A* happens under the condition that an event *B* happened before is denoted as the **conditional probability** *P(A|B)*.

Example: *P(lung-cancer|smoker)=0.7*
P(lung-cancer|non-smoker)=0.1

If two events do not influence each other (unconditional), the following equation holds:

$$P(A, B) = P(A) * P(B)$$

otherwise $$P(A, B) = P(A|B) * P(B)$$

Encoding

The probability of response *r* of a population of neurons given a stimulus *s*

$$P(\mathbf{r} \mid s) = P(r_1^s, r_2^s, r_3^s, r_4^s, \ldots \mid s)$$

For a specific stimulus *s*, *P(r|s)* is maximum for the typical stimulus-dependent response *r*.

Note: The response *r* may also occur in the absence of the stimulus

Decoding

The ‘reverse of encoding’

Probability that stimulus *s* was presented if response *r* occurred:

$$P(s \mid \mathbf{r}) = P(s \mid r_1^s, r_2^s, r_3^s, r_4^s, \ldots)$$

The most likely stimulus for response *r*

$$s = arg\ max_s P(s \mid \mathbf{r})$$

Can be calculated using the recording of the cells, *P(r|s),* with *Bayes’s theorem*:

$$P(s \mid \mathbf{r}) = \frac{P(\mathbf{r} \mid s)P(s)}{P(\mathbf{r})}$$

Decoding; maximum likelihood

Maximizing P(s|r) is equivalent to maximizing P(r|s)

$$P(s \mid \mathbf{r}) = \frac{P(\mathbf{r} \mid s)P(s)}{P(\mathbf{r})}$$

Therefore the stimulus estimated by

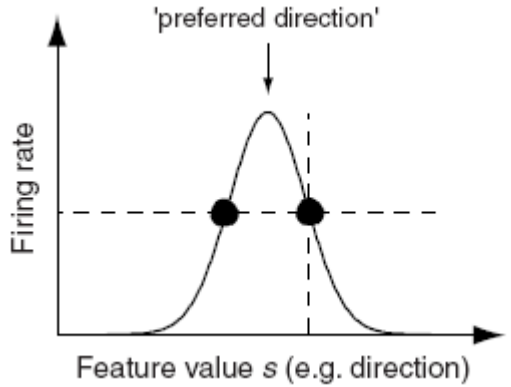
$$s = arg\ max_s P(s \mid \mathbf{r})$$

Is equal to the one estimated by *maximum likelihood*

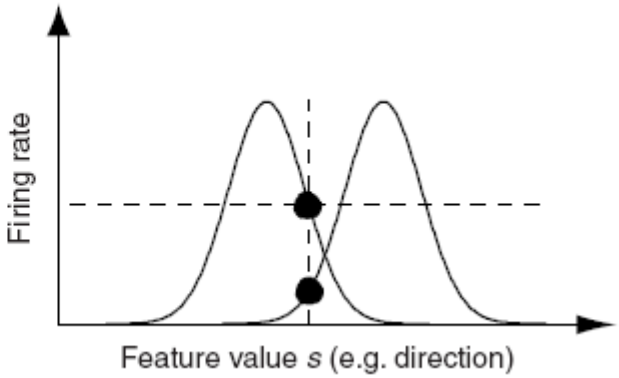
$$s_{ML} = arg\ max_s P(\mathbf{r} \mid s)$$

Example: Response tuning curves

Guessing the feature from the response of one neuron (with tuning curve) is ambiguous



But the feature can be extracted by looking at the response of two neurons



Example: Population vector

How to overcome wrong responses of few neurons?

The use of populations…

Individual neuron *i*:

tuning curve with cosine fit;

maximum activity r_i at preferred direction s_i

Population level:

Population vector shows preferred direction:

$$s_{dir} = \sum_i r_i s_i^{pref}$$

Representation

Local - one stimulus is only represented by one cell (grandmother cell) $\sim N$

Fully distributed – a stimulus is encoded by the combination of the activities of all neurons $\sim x^N$

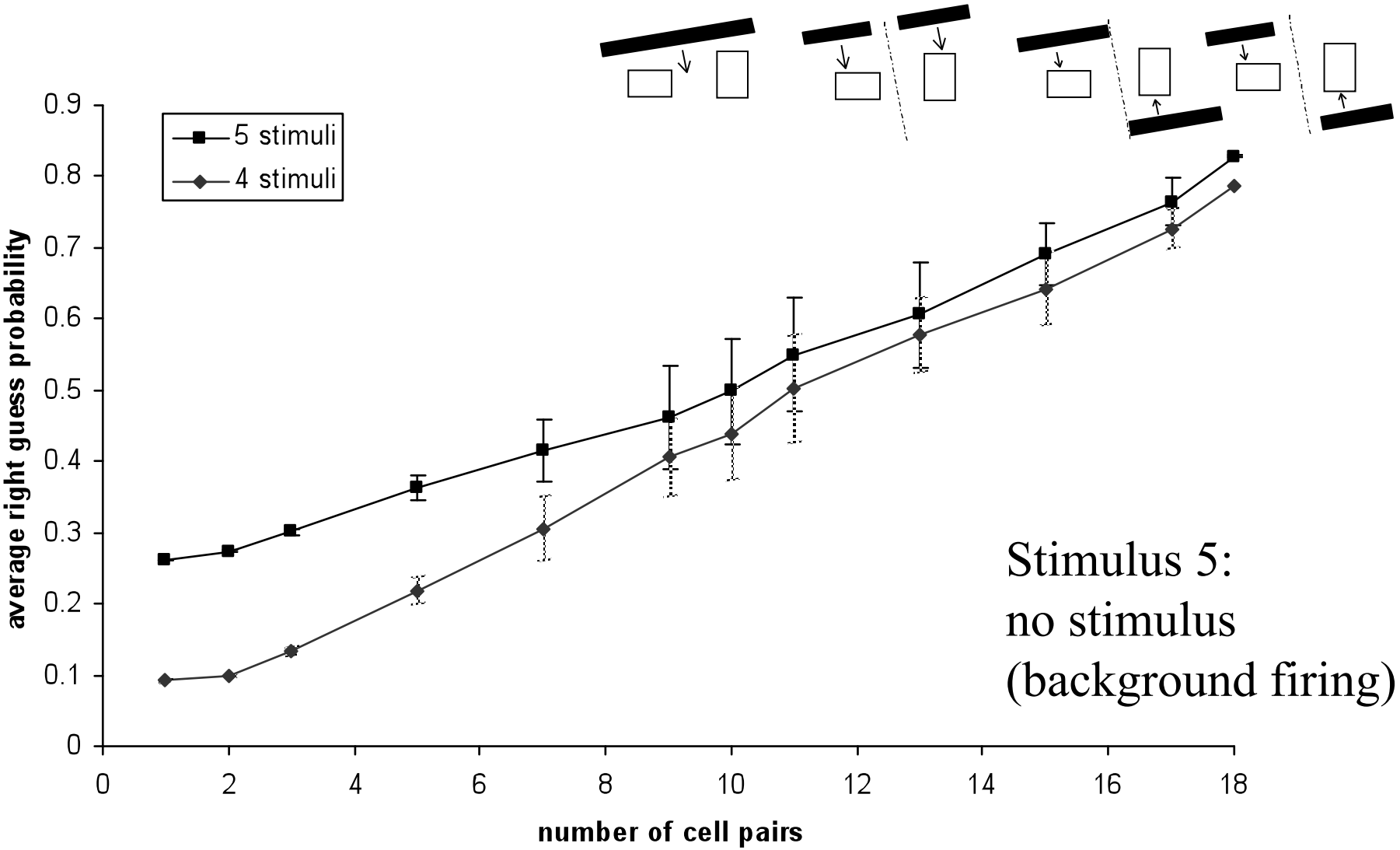
Sparse – only part of the neurons take part in identifying a stimulus (less wiring and firing than in fully distributed network)

What is a feature?

Complex stimuli comprise several features (e.g., color, direction, shape, figure-background contrast,…)

The population response to the stimulus is then the combination of the responses to the different features. The population representation of the stimulus is a *feature decomposition*.

How many neurons are needed to detect a stimulus?



Summary

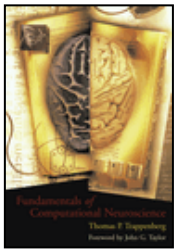
The brain uses a short time to evaluate spike trains:

- timing is important
- population vectors reduce the effect of noise on single signals
- distributed (sparse) encoding has several advantages (many stimuli can be encoded, robust against loss of neurons).

Further readings

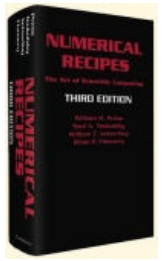
Trappenberg, 2010

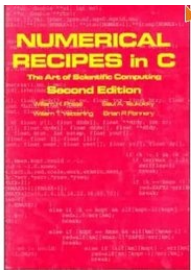
Appendix D (or chapter 5 in the first edition)

Numerical Recipes (3rd edition, 2007)

http://nr.com/





Additional material:

Dayan & Abbott, 2001

Rieke, Bialek, Warland, 1999

Shannon & Weaver, 1949