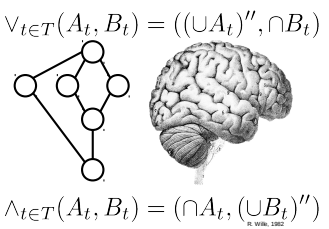
Bayesian Statistics and Machine Learning

for Neuroscience

Dominik Endres
AE Theoretical Neuroscience

October 14, 2015

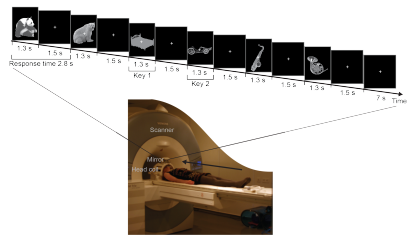
Philipps Universität Marburg



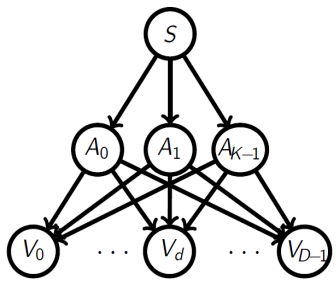
Outline

Why machine learning in Neuroscience?

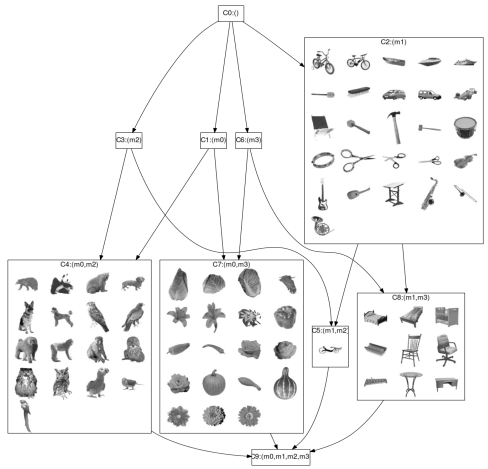
1. Application: data analysis



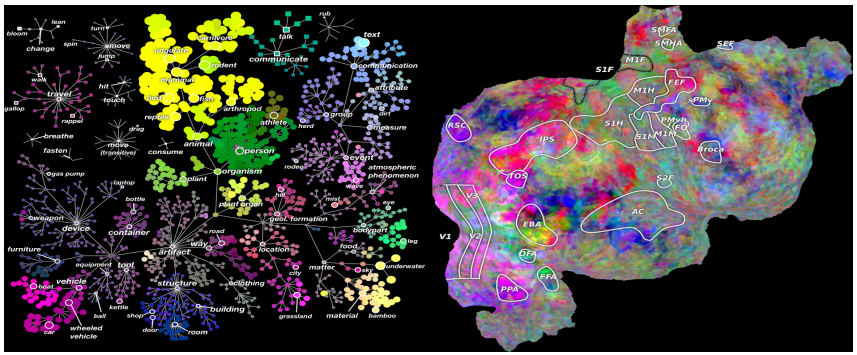
fMRI experiment



Bayesian machine learning model



Decoded semantic network



(c) A Huth, J Gallant

(c) M. Grosse-Wentrup

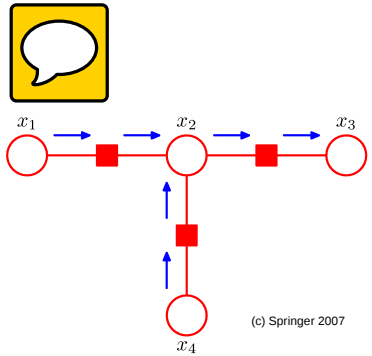
Why machine learning in Neuroscience?

2. Inspiration: new brain theories

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

$$P(A\vee\neg A)=1$$

Probabilistic reasoning principles



Bayesian message-passing model

Neural implementation (?)

Biol Cybern. Author manuscript; available in PMC Nov 7, 2012.
Published in final edited form as:
Biol Cybern. Feb 2011; 104(1-2): 137–160.
Published online Feb 17, 2011. doi: 10.1007/s00422-011-0424-z
PMCID: PMC3491875
EMSID: EMS49028

Action understanding and active inference

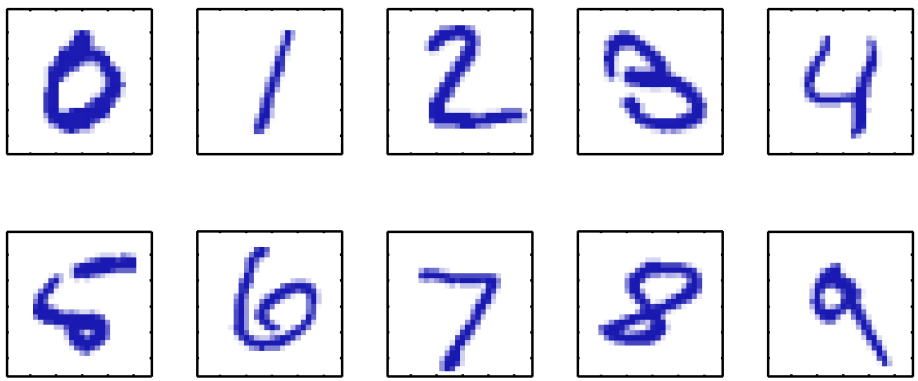
Karl Friston,* Jérémie Mattout, and James Kilner
Author information ► Copyright and License information ►

Abstract Go to:

We have suggested that the mirror-neuron system might be usefully understood as implementing Bayes-optimal perception of actions emitted by oneself or others. To substantiate this claim, we present neuronal simulations that show the same representations can prescribe motor behavior and encode motor intentions during action–observation. These simulations are based on the free-energy formulation of active inference, which is formally related to predictive coding. In this scheme, (generalised) states of the world are represented as trajectories. When

Example: handwritten digit recognition

The 'Drosophila' of Machine Learning

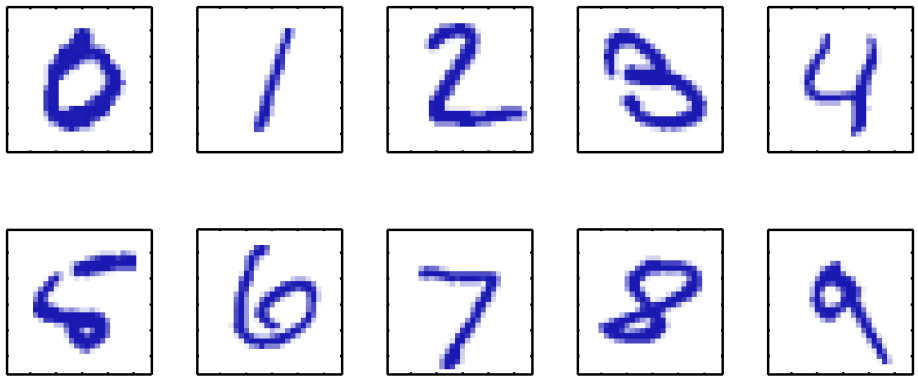


- 28x28 grayscale pixel images $\Rightarrow$ vector $\mathbf{x}$ comprising 784 integers in $\{0; \ldots; 255\}$.

- Build a machine that takes $\mathbf{x}$ as input and outputs digit $d(\mathbf{x})$.
- More generally: output digit-identity related information (e.g. probability).

Example: handwritten digit recognition

The 'Drosophila' of Machine Learning



- 28x28 grayscale pixel images $\Rightarrow$ vector $\mathbf{x}$ comprising 784 integers in $\{0; \ldots; 255\}$.
- Build a machine that takes $\mathbf{x}$ as input and outputs digit $d(\mathbf{x})$.
- More generally: output digit-identity related information (e.g. probability).

Example: handwritten digit recognition

Difficult problem:

http://www.cvl.isy.liu.se/ImageDB/images/external_images/MNIST_digits/tn/mnist_test7.jpg.index.html

- Hand-crafted heuristics for discrimination of letters?
- Variability between writers is high!
- $\Rightarrow$ proliferation of rules.
- $\Rightarrow$ poor results.

Example: handwritten digit recognition

Difficult problem:

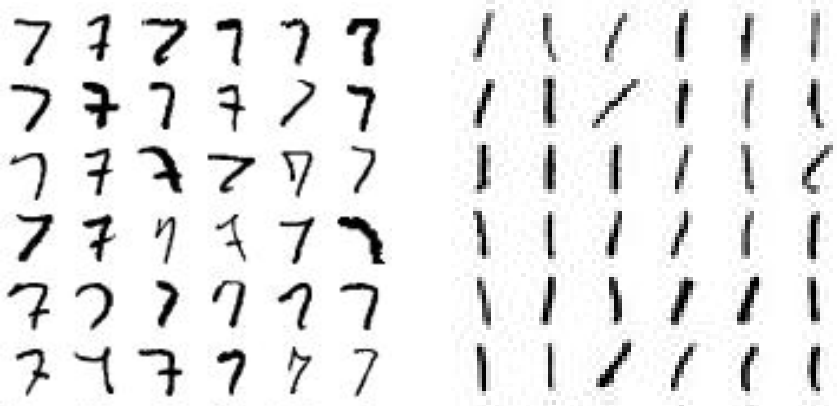
http://www.cvl.isy.liu.se/ImageDB/images/external_images/MNIST_digits/tn/mnist_test7.jpg.index.html

- Hand-crafted heuristics for discrimination of letters?
- Variability between writers is high!
- $\Rightarrow$ proliferation of rules.
- $\Rightarrow$ poor results.

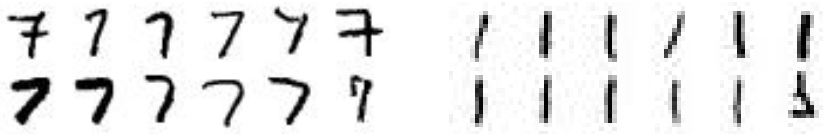
Example: handwritten digit recognition

Machine learning (ML) approach: split data into

Training set

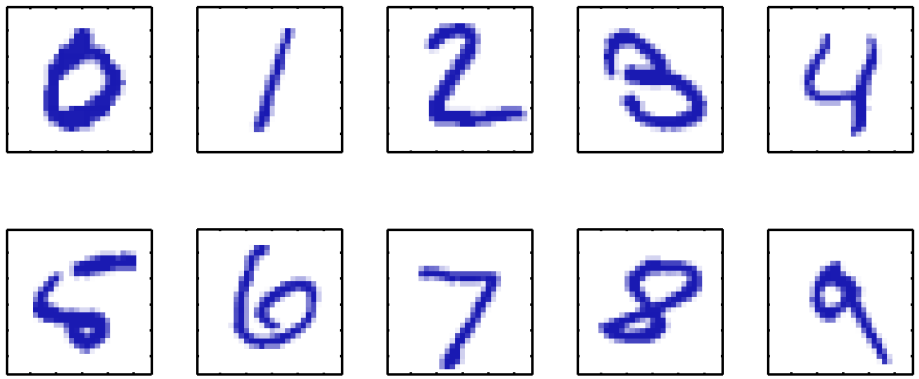


and test set



- Learn regularities in data from **training set** with an **adaptive system**.
- Test generalization ability of trained system on **test set**.
- This is an instance of **supervised learning**.

Supervised learning: classification

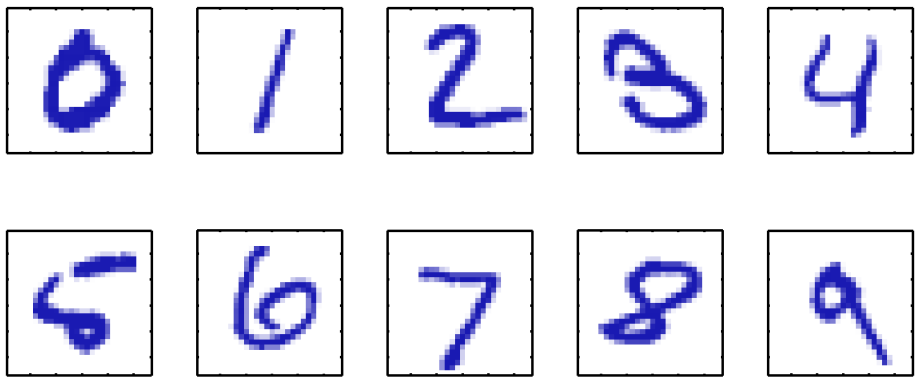


Learning handwritten digit recognition, i.e. learning the function

$$t = d(\mathbf{x}), \text{ where } t \in \{0; \ldots 9\}, \mathbf{x} \in \{0; \ldots; 255\}^{784}$$

which maps digit images $\mathbf{x}$ onto **target** digits t is an example of a **classification** problem, which is an instance of **supervised learning**.

Curse of dimensionality, uncertainty



Problem: the set of possible input vectors $\mathbf{x} \in \{0; \ldots; 255\}^{784}$ is huge:

$$|\{0; \ldots; 255\}^{784}| > 10^{1800}$$

The MNIST database contains 'only' $\approx 10^5$ training images.
$\Rightarrow$ one major type of **uncertainty** in ML is due to undersampling/curse of dimensionality.
$\Rightarrow$ uncertainty management is **REALLY** important in ML.

Random noise

Another type of uncertainty:

$$\eta \sim \mathcal{N}(\mu, \sigma)$$

(read: η is distributed according to a Normal distribution with mean μ and standard deviation σ).

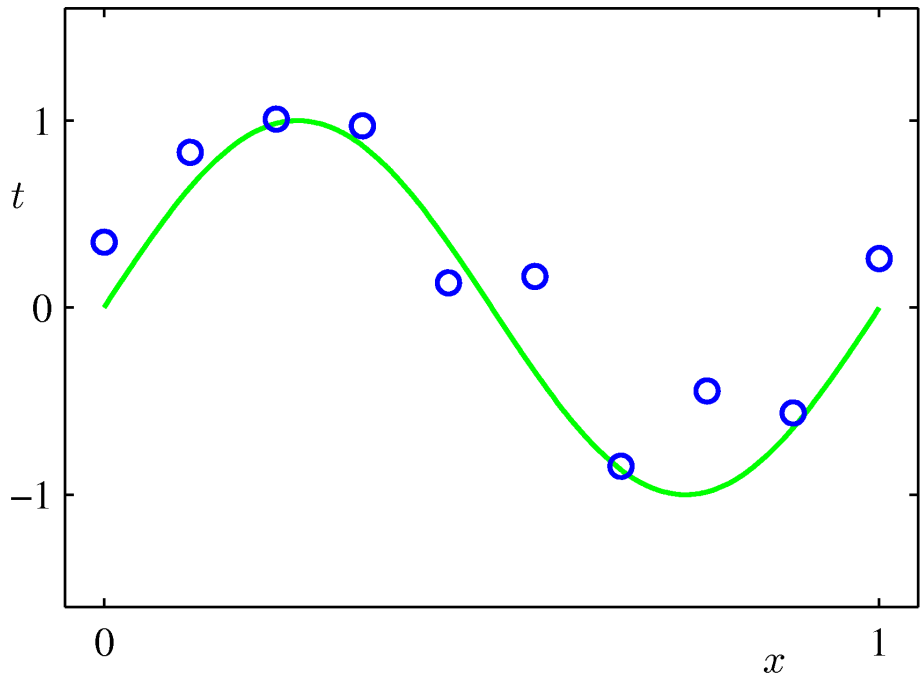
η is an instance of a **random** variable.

$\Rightarrow$ this important type of uncertainty is due to randomness:

- value of η really unknowable in advance, or
- we're too lazy/it's impractical to model η.

Example: transmit a signal from a sender to a receiver. In the transmission process, noise is added to the signal. We describe this noise by a random variable η.

Example: polynomial curve fitting

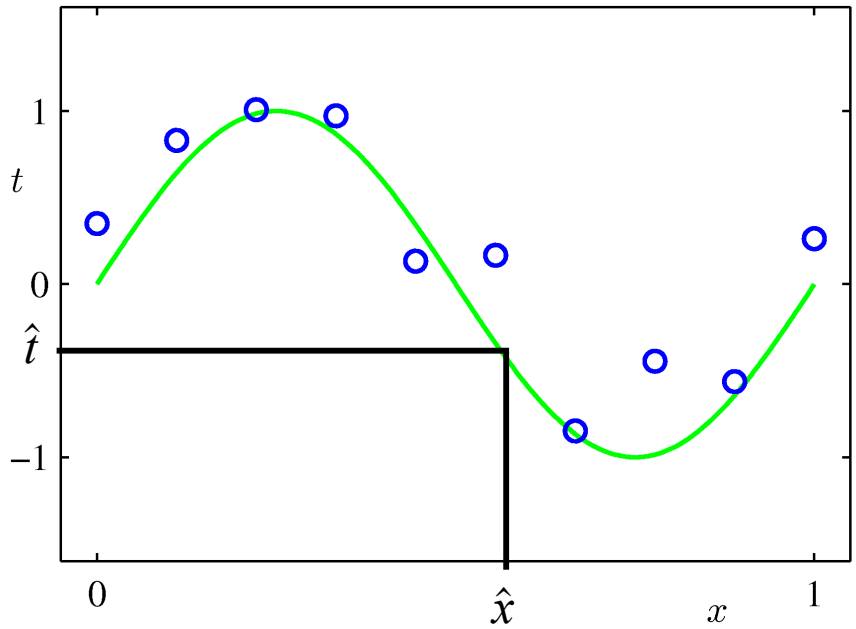


$N = 10$ datapoints, artificially created via

$$t_i = \sin(2\pi x_i) + \eta$$

with $\eta \sim \mathcal{N}(\mu, \sigma)$, x_i evenly spaced.
Artificial data are often instructive, because a **ground truth** is available.

Supervised learning: regression



Task: given $\mathbf{x} = (x_0, \ldots, x_{N-1})$ and $\mathbf{t} = (t_0, \ldots, t_{N-1})$, predict $\hat{t}$ at a (previously unseen) $\hat{x}$.

Since t is continuous, this is a **regression** problem.

Supervised learning: regression

Task: given $\mathbf{x} = (x_0, \ldots, x_{N-1})$ and $\mathbf{t} = (t_0, \ldots, t_{N-1})$, predict $\hat{t}$ at a (previously unseen) $\hat{x}$.
Model the function $t(x)$ by a polynomial of order M:

$$t(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots = \sum_{j=0}^{M} w_j x^j$$

Assume a polynomial, because it's easy to control the complexity of the model via M: the larger M, the more components has $\mathbf{w}$. $\mathbf{w}$ is the vector of model parameters used for explaining/predicting data.

Least-square regression

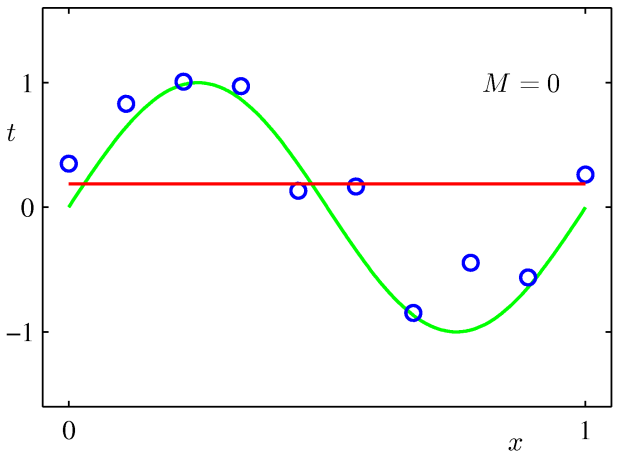
Task: given $\mathbf{x} = (x_0, \ldots, x_{N-1})$ and $\mathbf{t} = (t_0, \ldots, t_{N-1})$, predict $\hat{t}$ at a (previously unseen) $\hat{x}$.
Determine $\mathbf{w}$ by minimizing the squared deviation between predictions $t(x_i, \mathbf{w})$ and observed target values t_i:

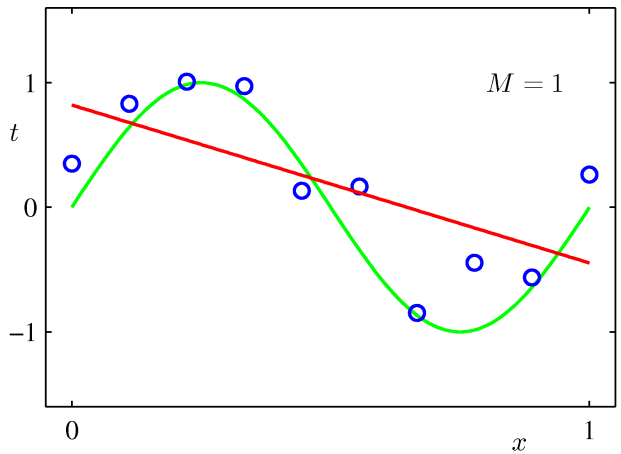
$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} E(\mathbf{w})$$

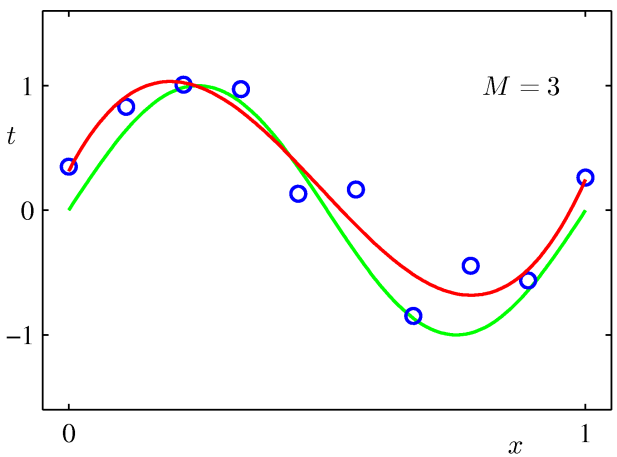
where

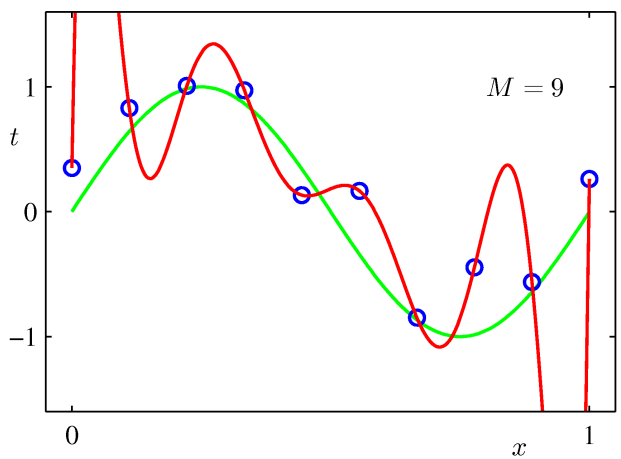
$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=0}^{N-1} (t(x_i, \mathbf{w}) - t_i)^2$$

Model selection







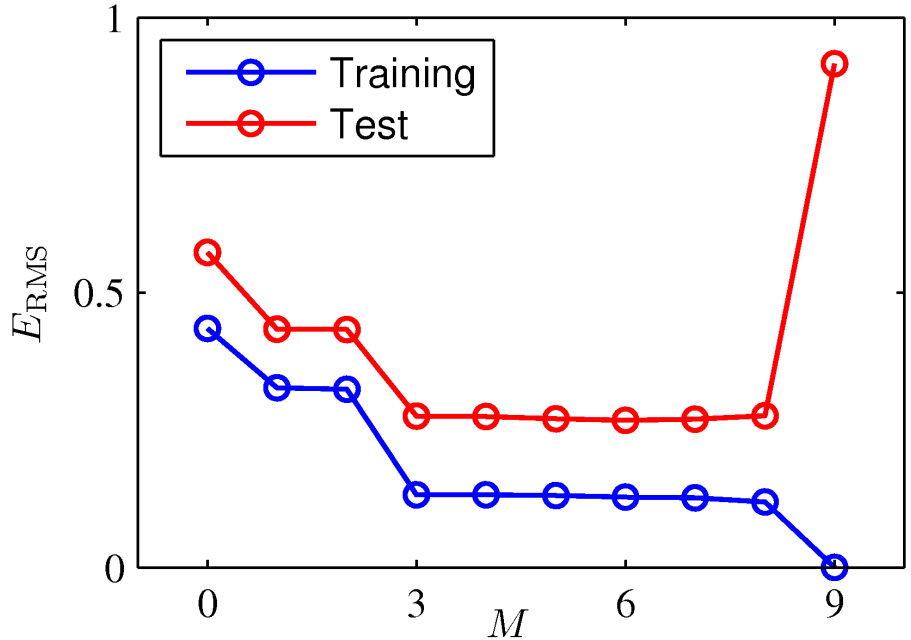


Question: which M is useful?

Model selection

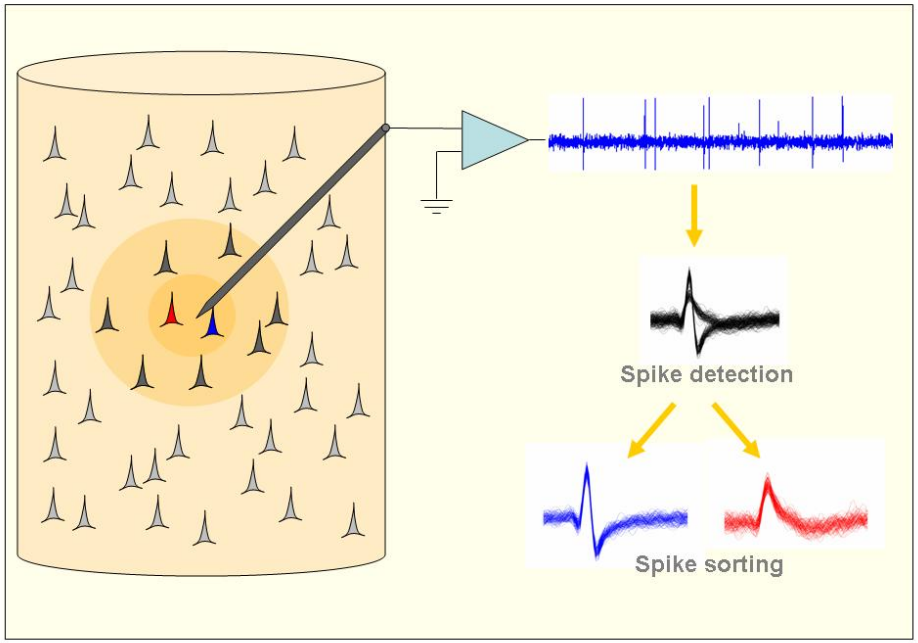
Question: which M is useful?

Use a **training data set** for determining the optimal $\mathbf{w}^*$, then evaluate the error root-mean square error $E_{RMS} = \sqrt{E(\mathbf{w}^*)/N}$ on a test set. Pick the M which minimizes the test error.



- $M = 0$: under-fitting
- $M = 9$: over-fitting, training data are 'memorized'.

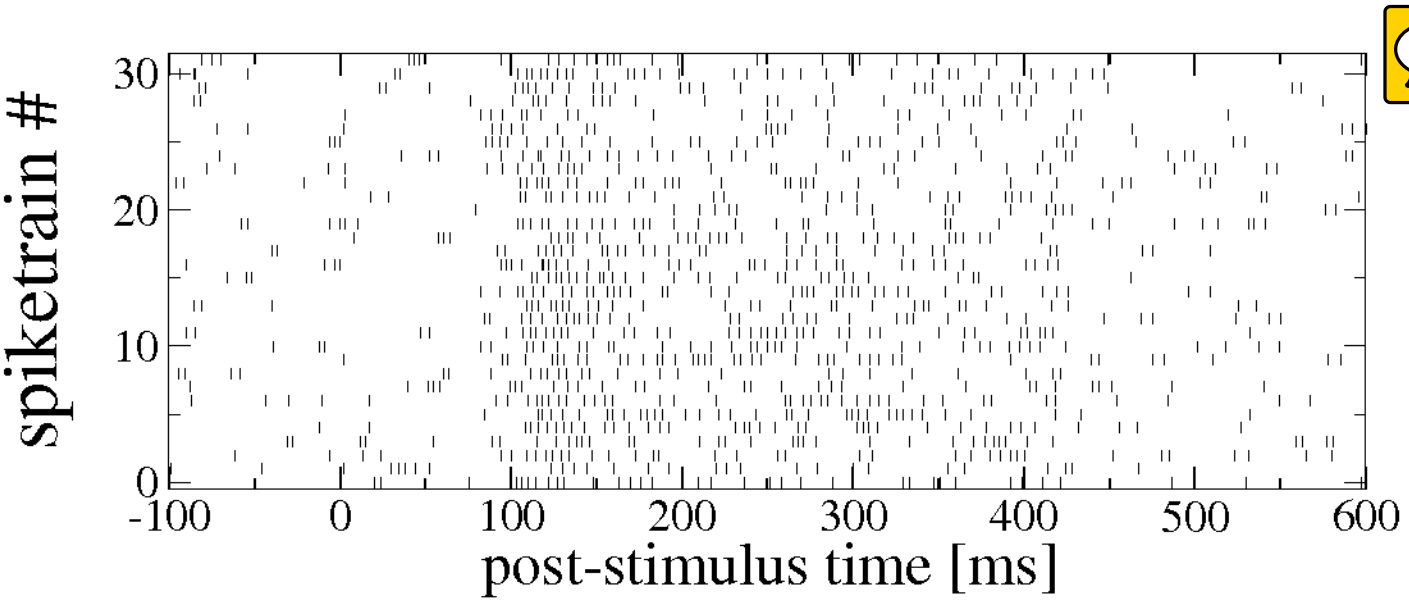
Modelling neural data



©2007 Q.Quiroga, Scholarpedia

- Insert electrode into live brain tissue next to a neuron.
- Record time course of extracelluar potential.
- Neurons produce steretypical **action potentials**, a.k.a **spikes**

Spike time rastergram



- Because spikes are stereotypical, only spike timing is considered.
- Every tick mark represents time of a spike.
- A temporal sequence of spikes is called a spike train.
- Every spike train contains data from one trial of the experiment.
- **Question**: how to describe/model the spike generating process within/across trials?

Peri-stimulus time histogram and SDF

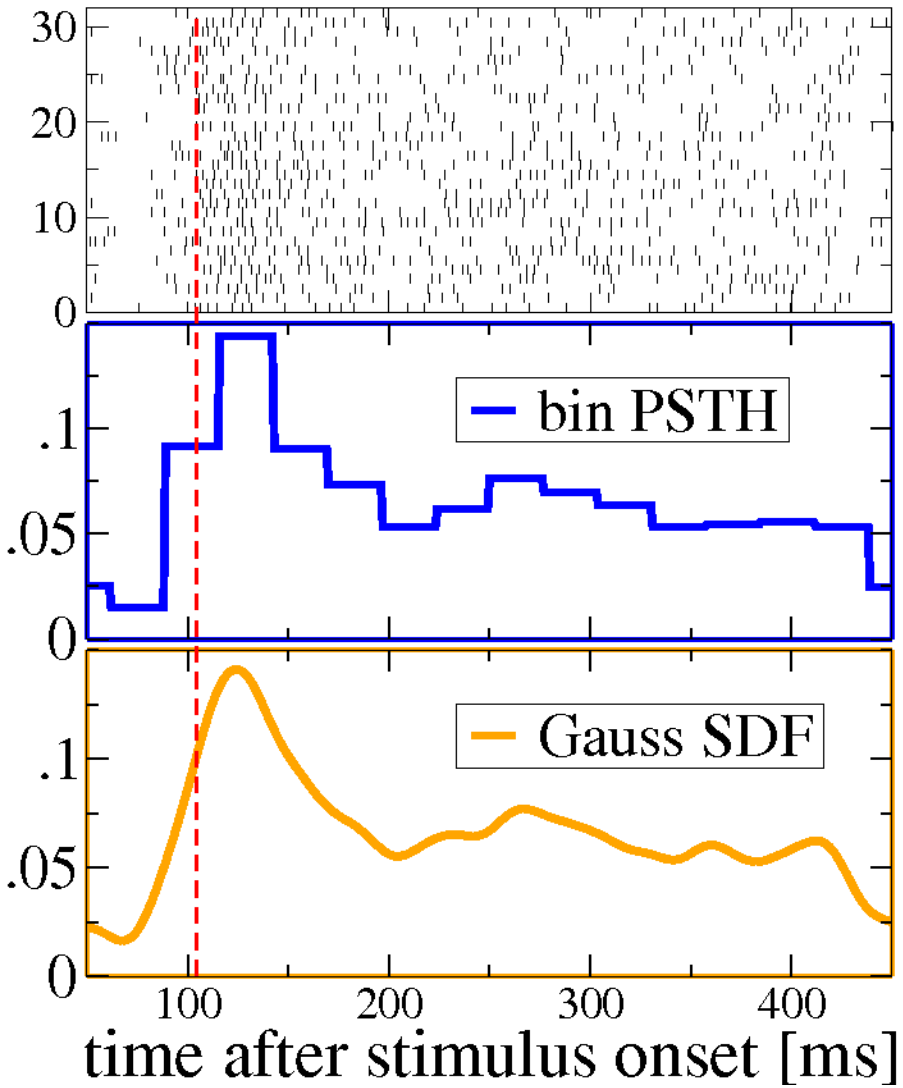
Traditional approach 1: model the frequency/rate $f(t)$ of observing a spike at time t assuming that

- $f(t)$ is constant across trials.
- $f(t)$ is constant within short time intervals (bins).
- Peri-stimulus time histograms, or **PSTH**.

Traditional approach 2: model the frequency/rate $f(t)$ of observing a spike at time t assuming that

- $f(t)$ is constant across trials.
- $f(t)$ varies smoothly across time.
- Spike density function, or **SDF**.

Peri-stimulus time histogram and SDF



Regularization

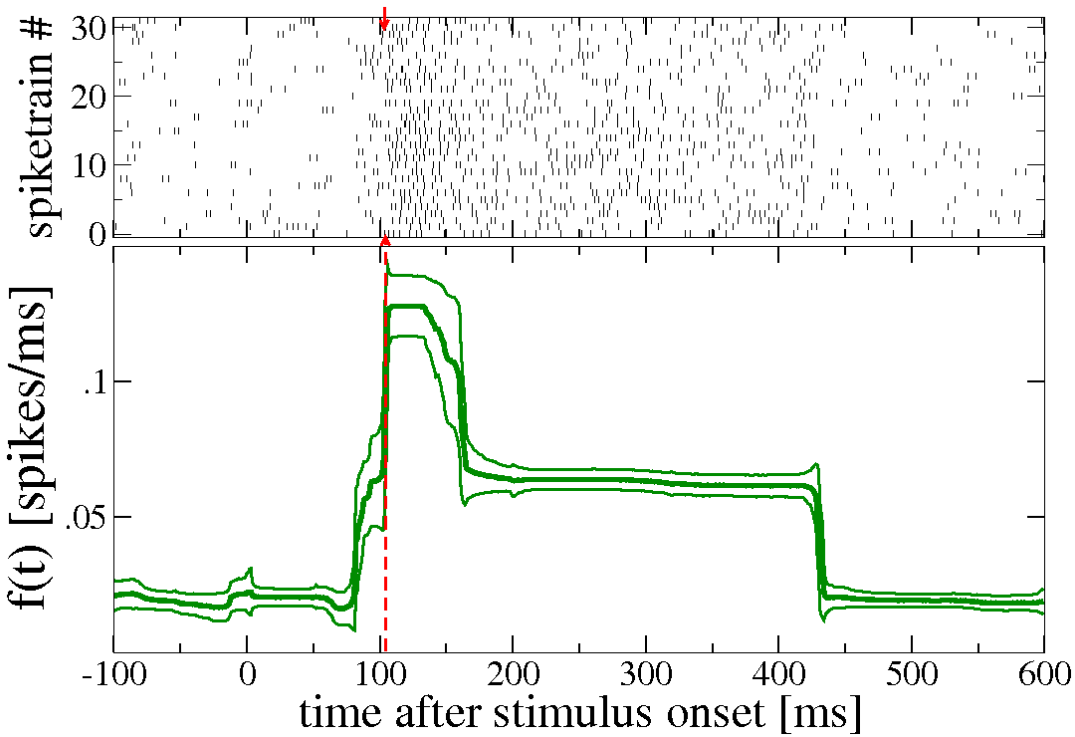
Regularization: dealing with the curse-of-dimensionality type of uncertainty by constraining the possible models/explanations we consider.

- PSTH: only consider models for $f(t)$ which are piecewise constant.
- SDF: only consider models where $f(t)$ is varies smoothly and slowly.

Question: which type of regularization to choose? How do we know if the regularization is too strong/too weak?
Problem: ground truth not known.

Regularization with Bayesian binning

Question: which type of regularization to choose? How do we know if the regularization is too strong/too weak?
Problem: ground truth not known $\Rightarrow$ try training set/test set method, a.k.a. **cross-validation**.



$\Rightarrow$ good regularization needs variable bin size: **Bayesian binning**.

Modelling face images



Another type of learning problem: no labels given. Instead, we wonder: what are the commonalities/differences between (all) face images?

Modelling face images



In other words: what are **good descriptors** of face images or parts thereof?

Finding those descriptors is an instance of **unsupervised learning** (no labels).

Humans: description by parts

Eyes: blue-grey
Nose: large
Hair: mostly absent
....

Humans would tend to describe a face by the (prominent) parts which comprise it. A *part* might be described as a mostly non-overlapping, additive component of the whole.

What is a good descriptor?

Question: what is a good description of an image?
(Partial) **answer**: description is good, if we can reconstruct the image from the description.
$\Rightarrow$ we'd like to keep the randomness-type uncertainty small.

But then we could just use the pixel values.
Additional requirement: we'd like to find a description of the image which is comprised of (mostly) non-overlapping, additive parts.
$\Rightarrow$ we regularize the learning problem, to keep the curse-of-dimensionality uncertainty small. $\Rightarrow$ Image descriptors will represent a trade-off between these two objectives.

What is a good descriptor?

Question: what is a good description of an image?
(Partial) **answer**: description is good, if we can reconstruct the image from the description.
$\Rightarrow$ we'd like to keep the randomness-type uncertainty small.
But then we could just use the pixel values.
Additional requirement: we'd like to find a description of the image which is comprised of (mostly) non-overlapping, additive parts.
$\Rightarrow$ we regularize the learning problem, to keep the curse-of-dimensionality uncertainty small. $\Rightarrow$ Image descriptors will represent a trade-off between these two objectives.

What is a good descriptor?

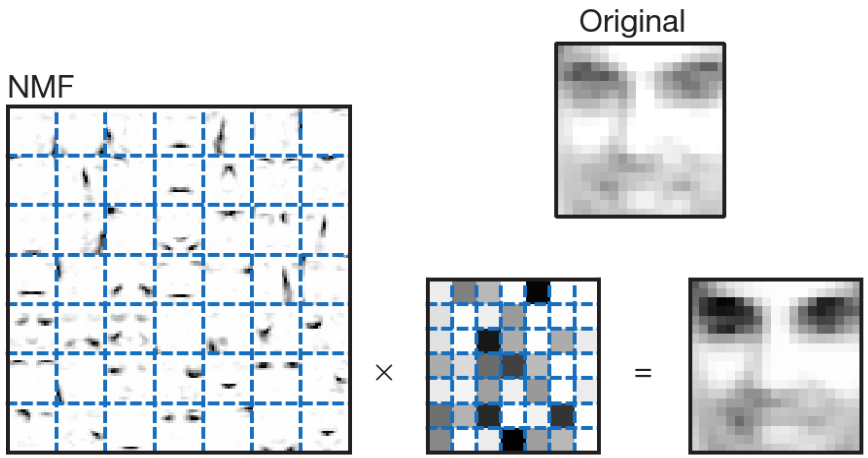
Question: what is a good description of an image?
(Partial) **answer**: description is good, if we can reconstruct the image from the description.
$\Rightarrow$ we'd like to keep the randomness-type uncertainty small.
But then we could just use the pixel values.
Additional requirement: we'd like to find a description of the image which is comprised of (mostly) non-overlapping, additive parts.

$\Rightarrow$ we regularize the learning problem, to keep the curse-of-dimensionality uncertainty small. $\Rightarrow$ Image descriptors will represent a trade-off between these two objectives.

What is a good descriptor?

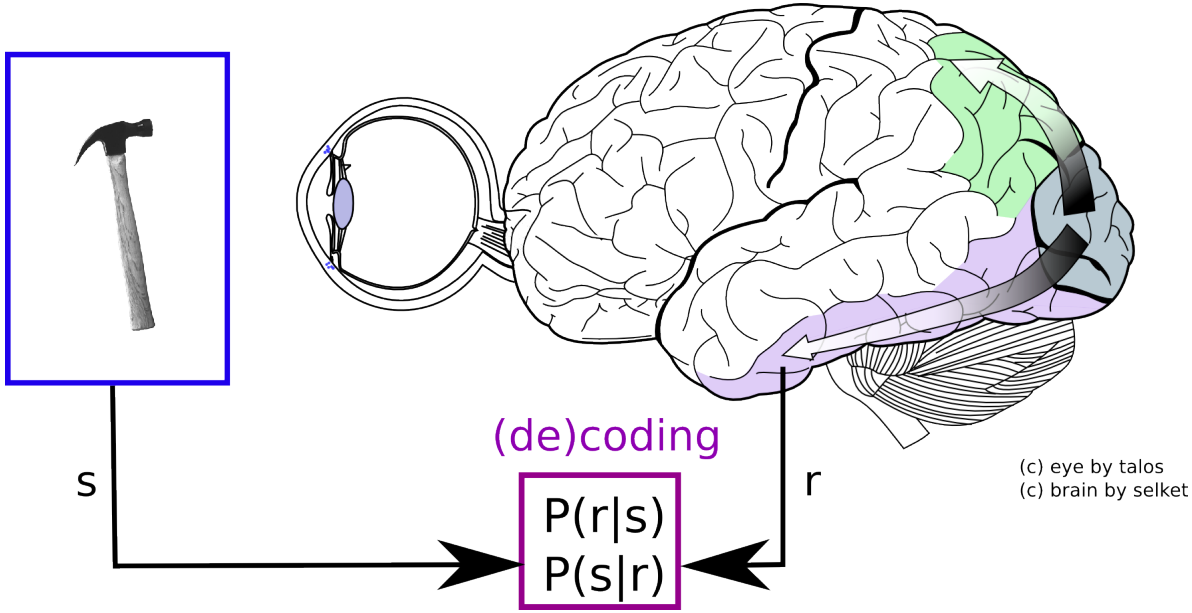
Question: what is a good description of an image?
(Partial) **answer**: description is good, if we can reconstruct the image from the description.
$\Rightarrow$ we'd like to keep the randomness-type uncertainty small.
But then we could just use the pixel values.
Additional requirement: we'd like to find a description of the image which is comprised of (mostly) non-overlapping, additive parts.
$\Rightarrow$ we regularize the learning problem, to keep the curse-of-dimensionality uncertainty small. $\Rightarrow$ Image descriptors will represent a trade-off between these two objectives.

Image description with non-negative matrix factorization



- Learning carried out with non-negative matrix factorization (NMF).
- Face images: 19x19 pixels.
- Many interpretable face parts.

Motivation: neural (de)coding



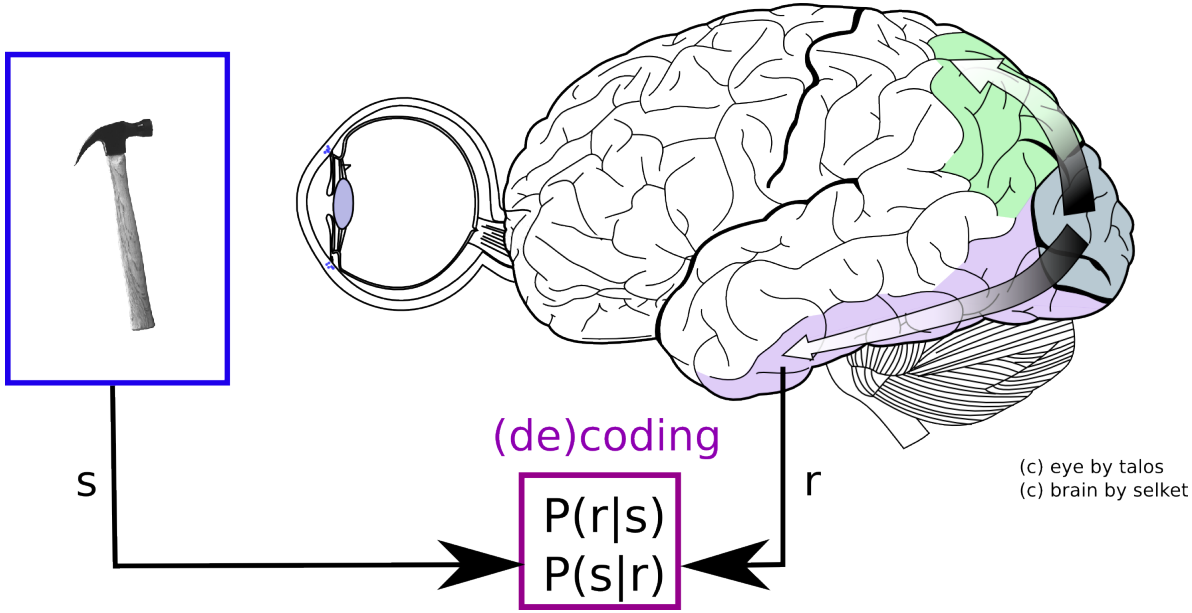
Neural code, $P(r|s)$:

- Activation pattern of a population of neurons (codewords).
- Represents sensory information items, e.g. presence/absence of stimuli.

Neural decoding, $P(s|r)$:

- Reconstruct information item from activation pattern.
- Often classification semantics: stimuli either *same* or *different*
- Quality measures: classification rates, mutual information etc.
- Well explored approach, e.g. [Barlow, 1972, Quiroga, 2007].

Motivation: neural (de)coding



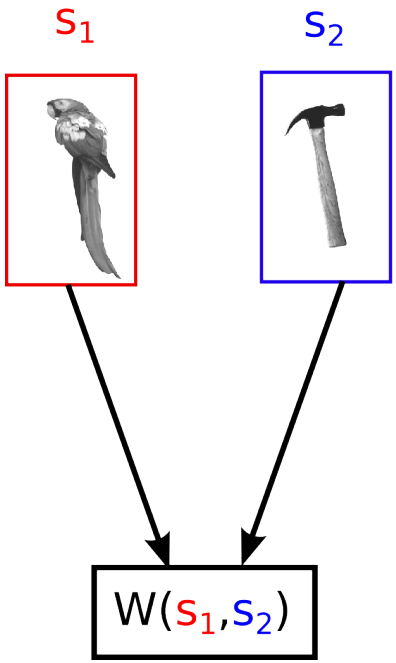
Neural code, $P(r|s)$:

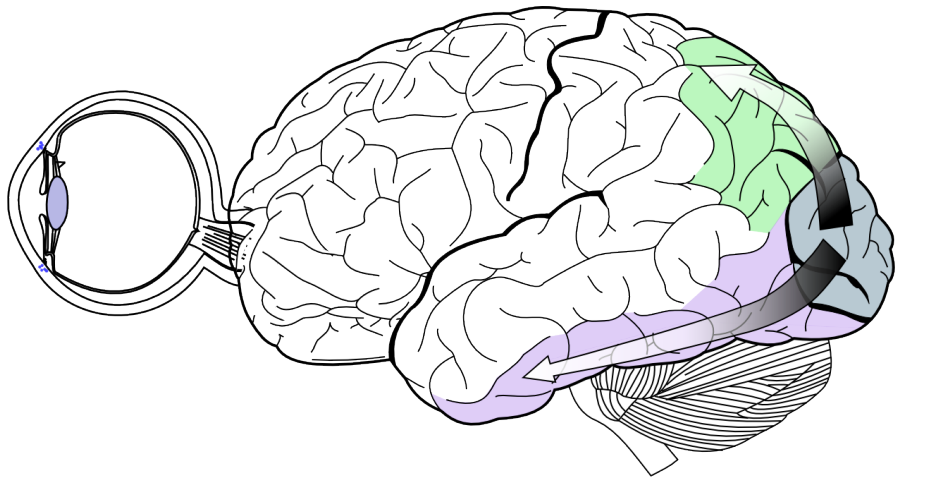
- Activation pattern of a population of neurons (codewords).
- Represents sensory information items, e.g. presence/absence of stimuli.

Neural decoding, $P(s|r)$:

- Reconstruct information item from activation pattern.
- Often classification semantics: stimuli either *same* or *different*
- Quality measures: classification rates, mutual information etc.
- Well explored approach, e.g. [Barlow, 1972, Quiroga, 2007].

Beyond classification: relational representations

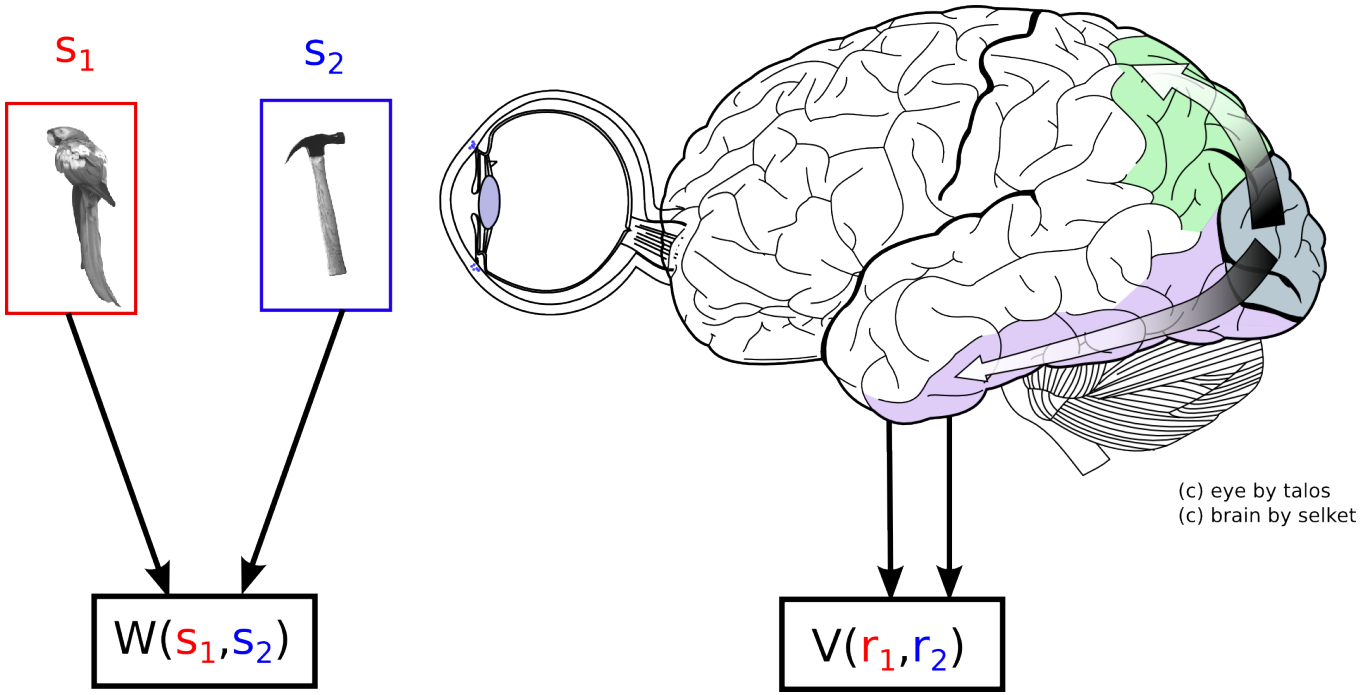




(c) eye by talos
(c) brain by selket

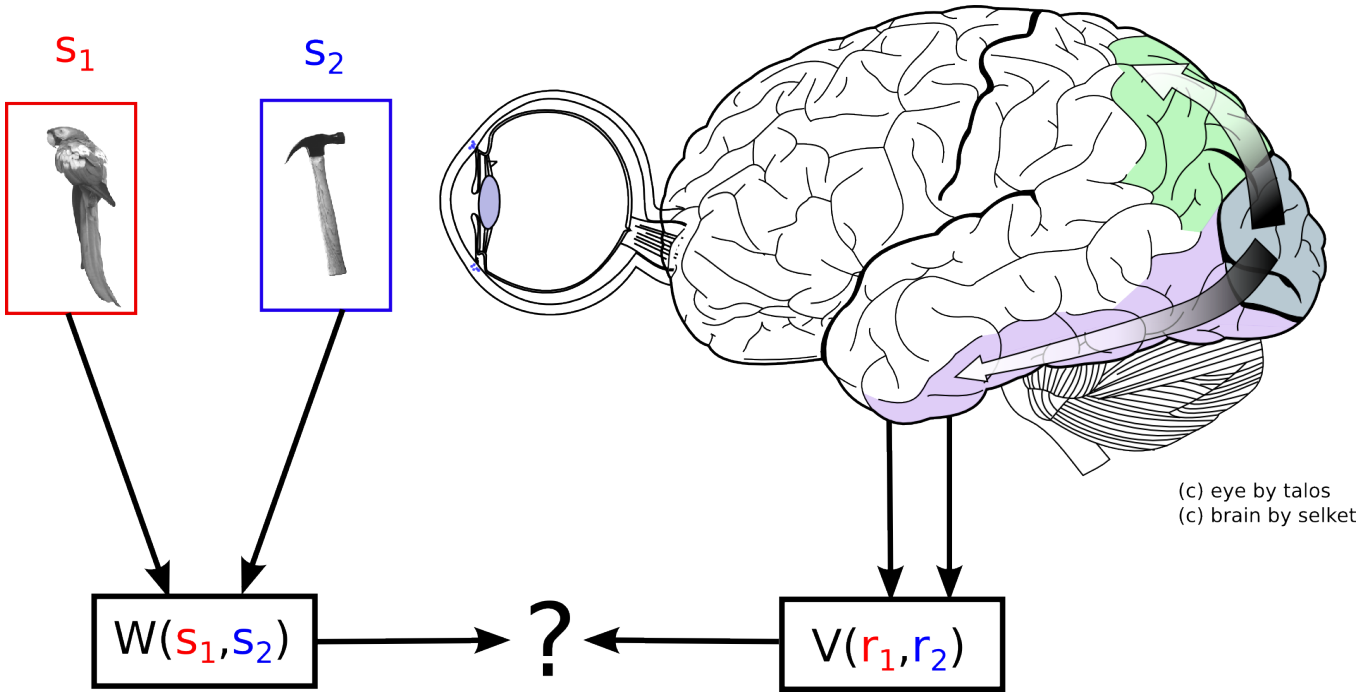
- Stimuli s_1, s_2
- $W(s_1, s_2)$: *perceived* relationship btw. s_1,s_2, e.g. similarity

Beyond classification: relational representations



- Stimuli s_1, s_2
- $W(s_1, s_2)$: *perceived* relationship btw. s_1,s_2, e.g. similarity
- $V(r_1, r_2)$: *measured* relationship btw. r_1,r_2, e.g. distance, overlap

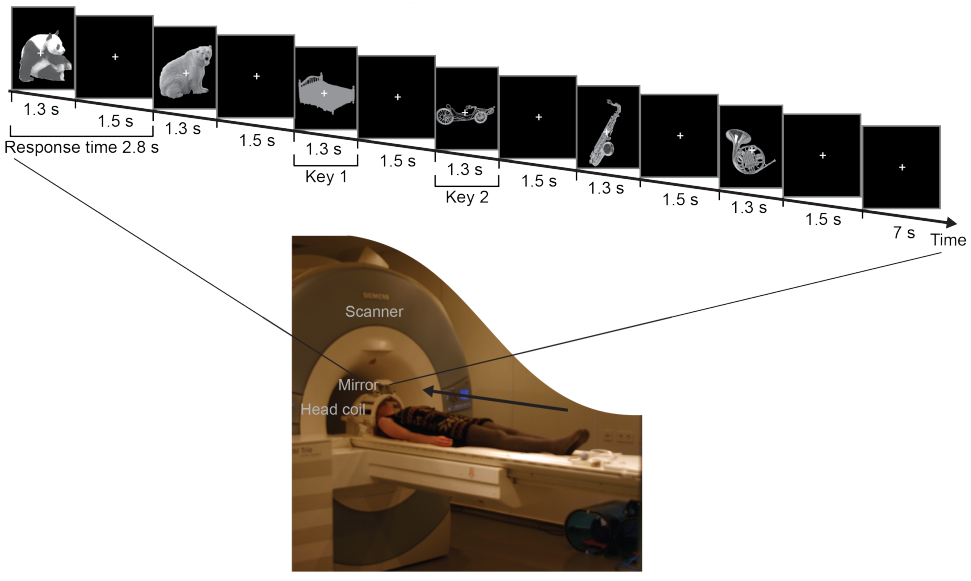
Beyond classification: relational representations



- Stimuli s_1, s_2
- $W(s_1, s_2)$: *perceived* relationship btw. s_1,s_2, e.g. similarity
- $V(r_1, r_2)$: *measured* relationship btw. r_1,r_2, e.g. distance, overlap

Question: How are *perceived* relationships between represented information items reflected in the neural code?

Experimental design



- Target detection task:
- silhouette or intact image
- indicated by keypress
- 48 sessions

©R. Adam and T. Rohe

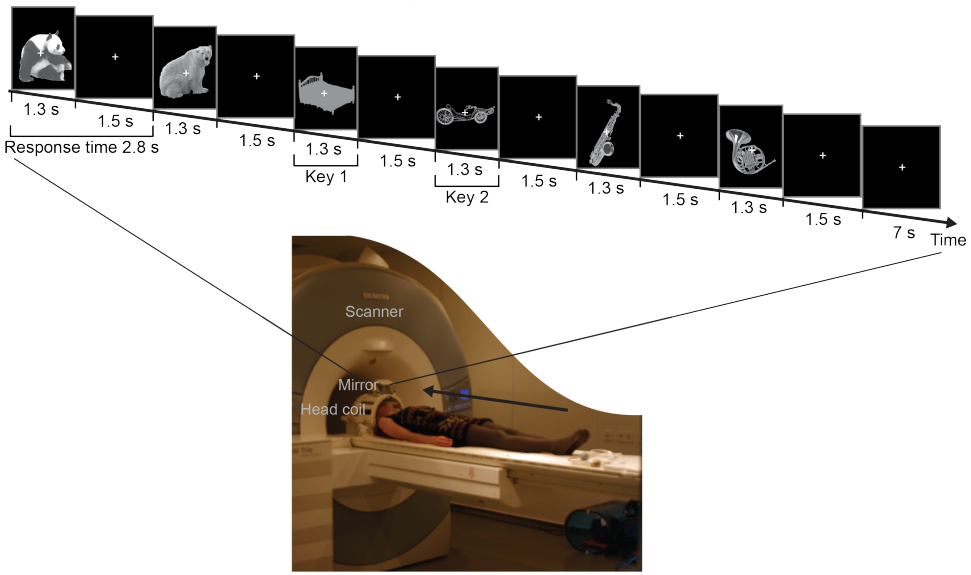
- 72 grayscale photographs as stimuli, animate and inanimate.
- **Animate**: mammals, birds, vegetables, flowers.
- **Inanimate**: furniture, vehicles, tools, musical instruments.
- Luminance equalized, size along main diagonal equalized.

Experiment yields labels and voxel activities. We'd like to know:

- does the brain organize the stimuli into 'concepts'?
- how are these concepts related neurally and in perception?

$\Rightarrow$ need to learn latent structure of the data!

Experimental design



- Target detection task:
- silhouette or intact image
- indicated by keypress
- 48 sessions

©R. Adam and T. Rohe

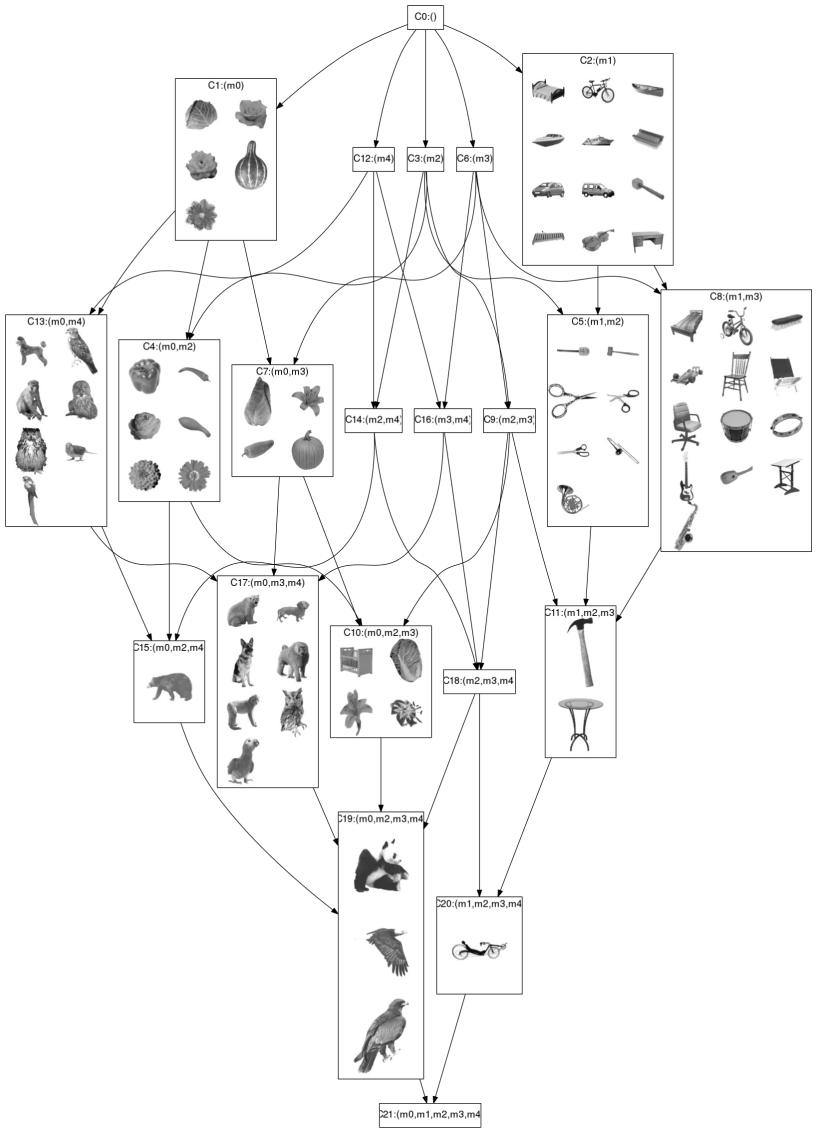
- 72 grayscale photographs as stimuli, animate and inanimate.
- **Animate**: mammals, birds, vegetables, flowers.
- **Inanimate**: furniture, vehicles, tools, musical instruments.
- Luminance equalized, size along main diagonal equalized.

Experiment yields labels and voxel activities. We'd like to know:

- does the brain organize the stimuli into 'concepts'?
- how are these concepts related neurally and in perception?

$\Rightarrow$ need to learn latent structure of the data!

Concept lattice IT



An example of *relational learning*.

- 5 attributes
- animals (m4): **C13** (7/7), **C15**, **C17** (7/7), **C19** (3/3)
- plants (m0): **C1** (5/5), **C4** (6/6), **C10** (3/4)
- tools, furniture, vehicles: **C2** (12/12), **C5** (13/13), **C5** (7/7), **C11** (2/2)

Summary

- Representation of **uncertainty** is a major concern for Machine Learning.
- For successful learning, types of uncertainty need to be traded off against each other:
  - randomness
  - uncertainty about the model parameters 
- **Supervised learning**: learn the mapping of some input $\mathbf{x}$ to an output t of which examples are available.
- **Unsupervised learning**: learn a representation of the (distribution of the) data.
- **Relational learning**: part of the data are labelled, learn latent structure (relations) in the data.

Summary

- Representation of **uncertainty** is a major concern for Machine Learning.
- For successful learning, types of uncertainty need to be traded off against each other:
  - randomness
  - uncertainty about the model parameters
- **Supervised learning**: learn the mapping of some input **x** to an output t of which examples are available.
- **Unsupervised learning**: learn a representation of the (distribution of the) data.
- **Relational learning**: part of the data are labelled, learn latent structure (relations) in the data.