Neural Networks

Lecture 13: Revision and Conclusion

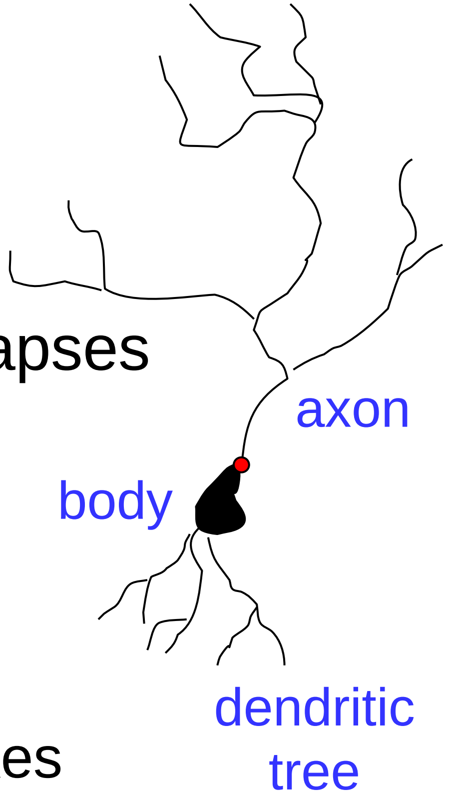
http://www.informatik.uni-hamburg.de/WTM/

Revision: goals of neural computation

- To understand how the brain actually works
- To understand a new style of computation
  - Inspired by neurons and their adaptive connections
  - Very different style from sequential computation
    - should be good for things that brains are good at (e.g. vision)
    - should be bad for things that brains are bad at (e.g. 23 x 71)
- To solve practical problems by developing novel learning algorithms
  - Learning algorithms can be very useful even if they have nothing to do with how the brain works

A typical cortical neuron

- Gross physical structure:
  - There is one axon that branches
  - There is a *dendritic* tree that collects input from other neurons
- Axons typically contact dendritic trees at synapses
  - A spike of activity in the axon causes charge to be injected into the post-synaptic neuron
- Spike generation:
  - There is an *axon* that generates outgoing spikes whenever enough charge has flowed in at synapses to depolarize the cell membrane



Three kinds of learning

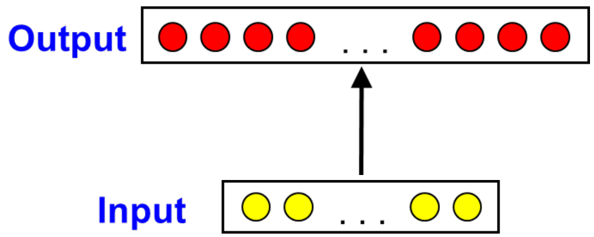
- Supervised Learning: *this models $p(y|x)$*
  - Learn to predict a real valued output or a class label from an input.
- Reinforcement learning: *tries to be successful at the end*
  - Choose actions that maximize payoff
- Unsupervised Learning: *this models $p(x)$*
  - Build a causal generative model that explains why some data vectors occur and not others

    or

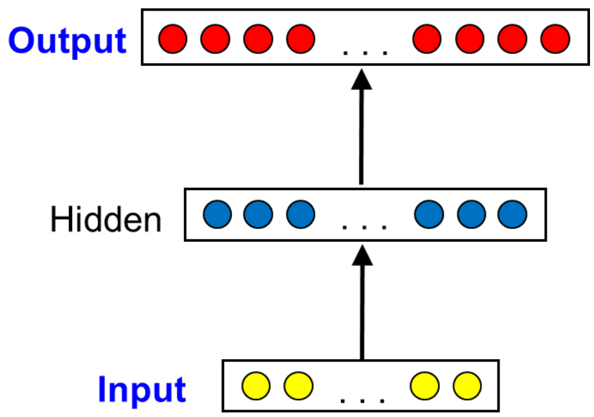
  - Discover interesting features; separate sources that have been mixed together; find temporal invariants etc.

Neural Networks

- Single-layer perceptron (SLP)
  - Linear classifier
  - Binary threshold (linear) activation



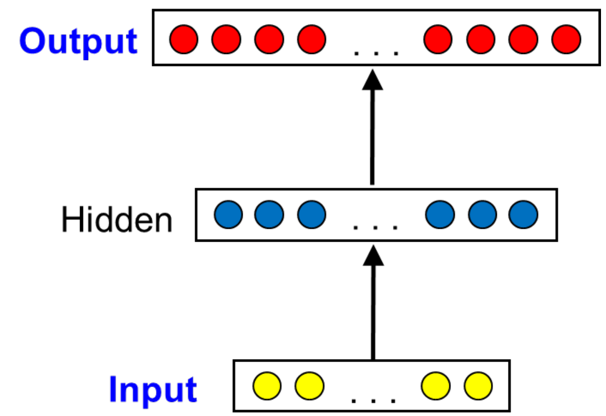
- Multi-layer perceptron (MLP)
  - Non-linear classifier
  - Sigmoid (non-linear) activation



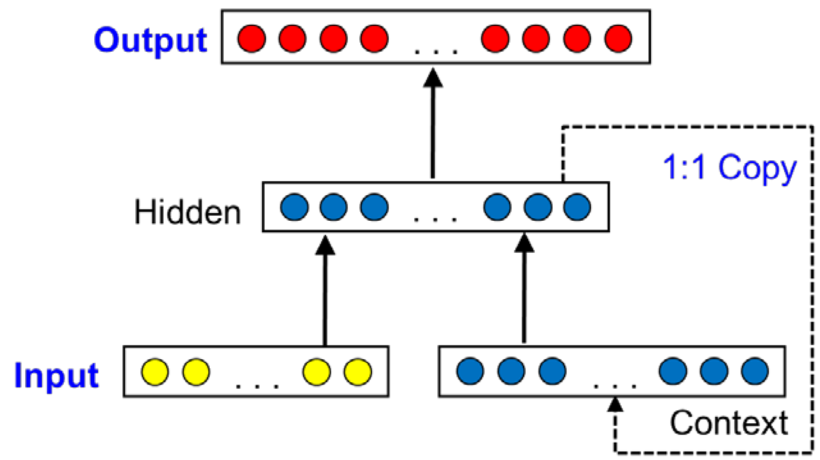
- Important activation functions:
  Threshold, logistic, tanh, ReLU, softmax

More sophisticated Networks: Types of connectivity

- Feedforward networks
  - These compute a series of transformations
  - Typically, the first layer is the input and the last layer is the output.

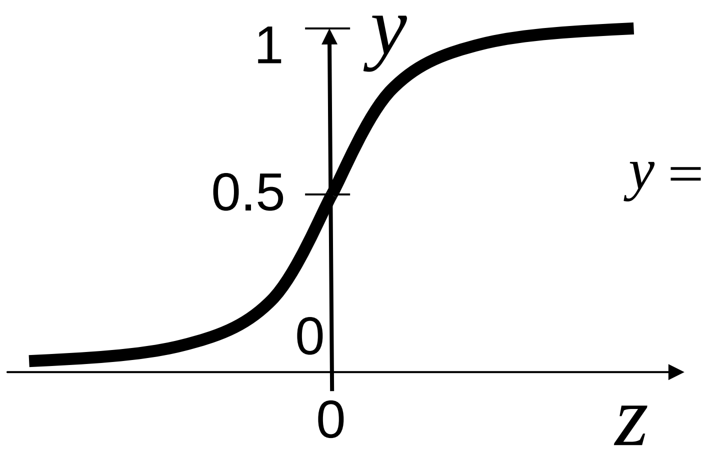


- Recurrent networks
  - These have directed cycles in their connection graph. They can have complicated temporal dynamics.
  - More biologically realistic.

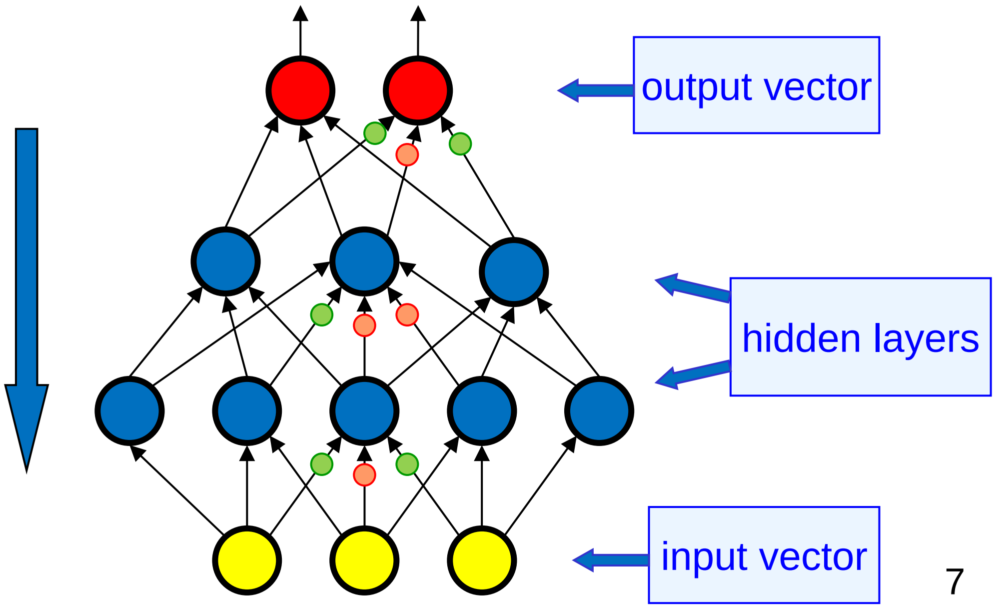


Learning by back-propagating error derivatives

- Sigmoid neurons:
  - Smooth and bounded real-valued output of their total input.
  - Typically logistic function or tanh
- Back-propagation:
  - Compare outputs with *correct answer* to get error signal
  - Back-propagate error signal to get *derivatives* for learning

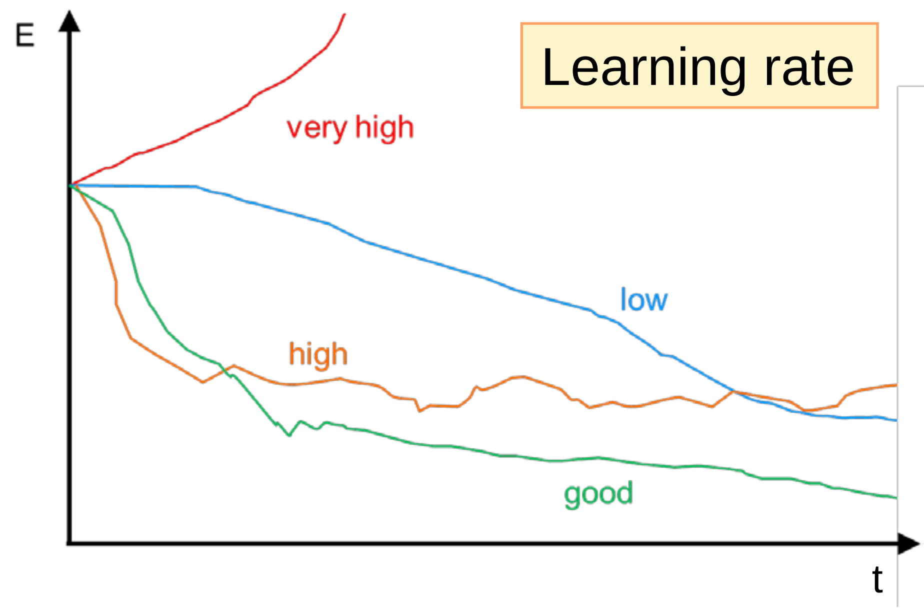


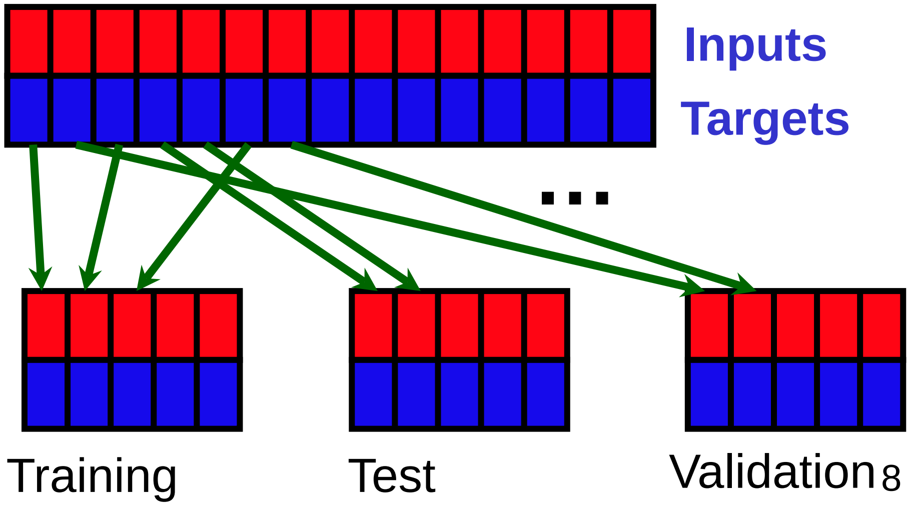
$$y = \frac{1}{1 + e^{-z}}$$



Training and Optimization

- Learning Rate:
  - too small:
    local minima trap
  - too large:
    overshooting narrow valleys
- Momentum:
  - Keep fraction of previous gradient
- Preventing Overfitting:
  - Cross Validation,
    Early Stopping
  - Regularization
- Hyperparameter Optimization

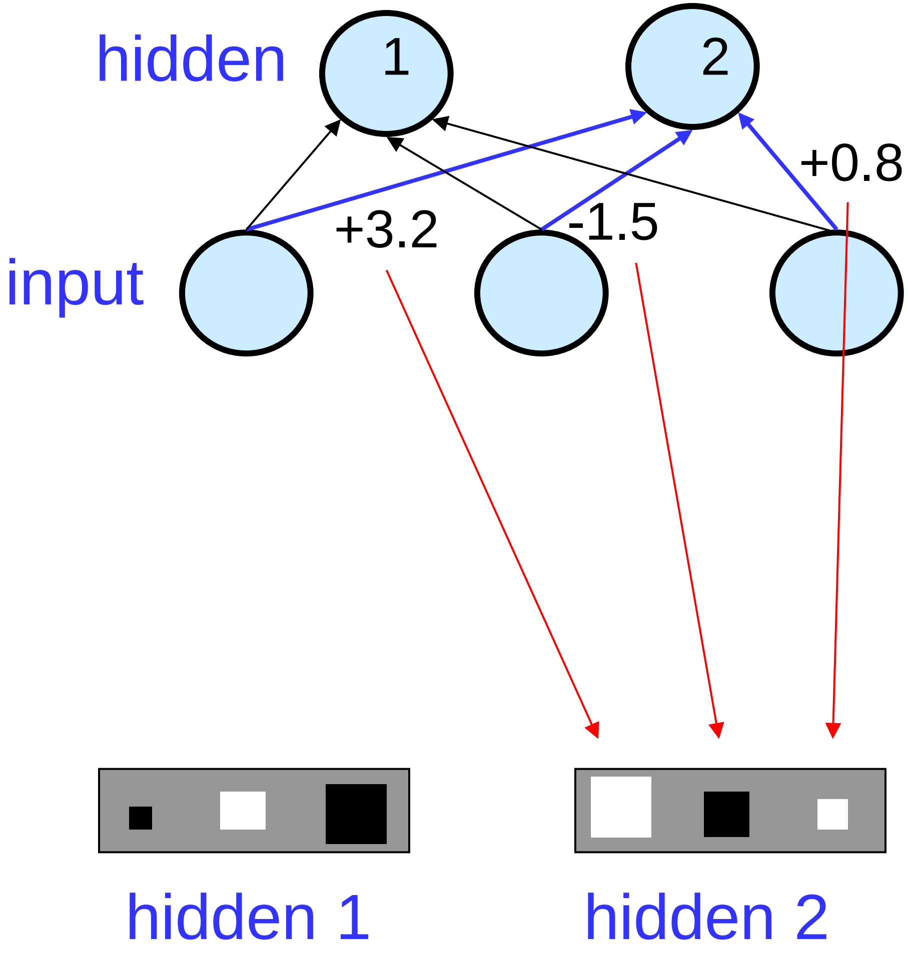




How to show the weights of hidden units

- The obvious method is to show numerical weights on the connections:
  - Try showing 25,000 weights this way!

- Better: Hinton Diagram
  Show weight matrix as black (negative) or white (positive valued) blobs
  - Better use of pixels
  - Easier to see patterns



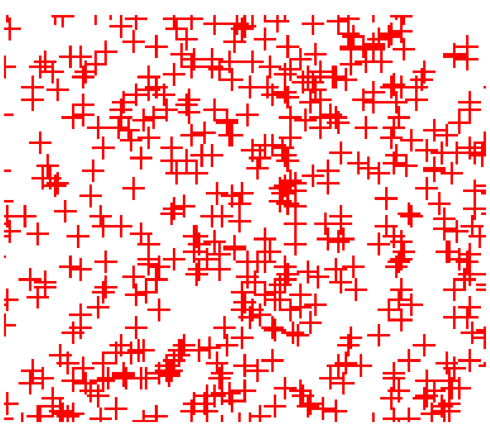
Localist and distributed representations

- Localist: simplest way to represent things with neural networks is to dedicate *one neuron to each thing*.
  - Easy to understand, code by hand, and learn,
  - Easy to associate with other representations or responses,
  - Inefficient whenever the data has componential structure.
  - E.g. one-hot: apple := (1,0,0,0), banana := (0,0,1,0)

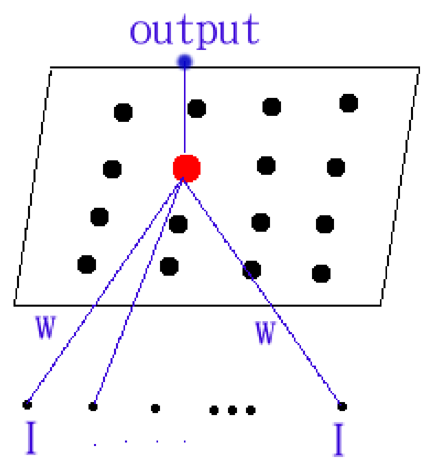
- Distributed: *many-to-many* relationship between two types of representation (such as concepts and neurons).
  - Each concept is represented by many neurons,
  - Each neuron participates in the representation of many concepts.
  - E.g. word embeddings (word2vec)

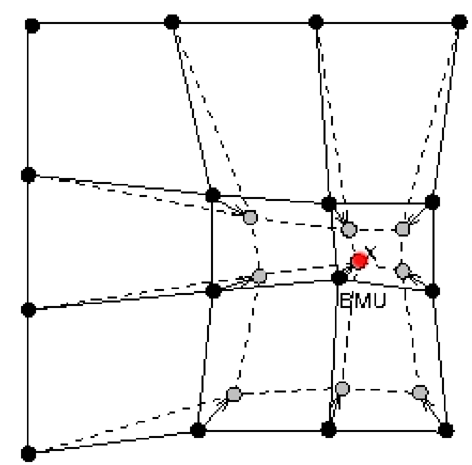
Self-organizing Maps

- One neural method of clustering was proposed by Kohonen
- His idea was inspired by the way that mappings are learnt in the topographic feature maps found in many brain areas
- These *feature maps* consist of one- and two-dimensional sheets of neurons with lateral connections



Raw Data

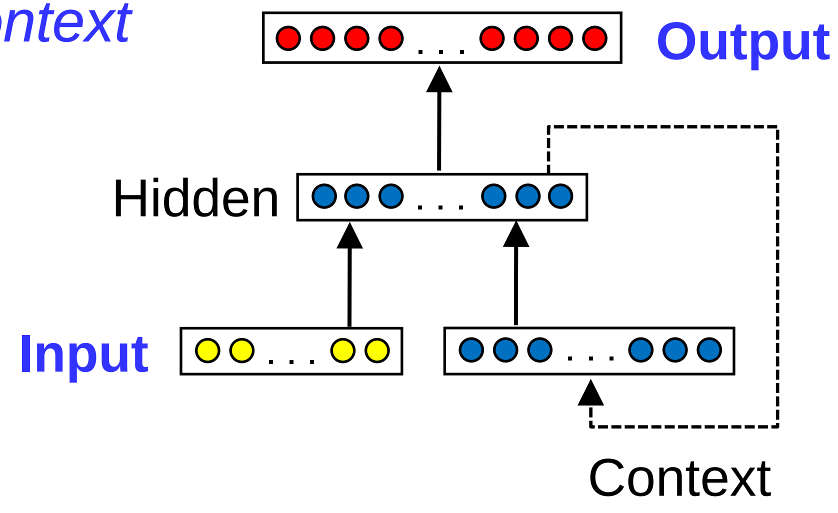




Recurrent Neural Networks

- Simple Recurrent Neural Network (SRN)
  - Previous activation added as *context* to the current activation

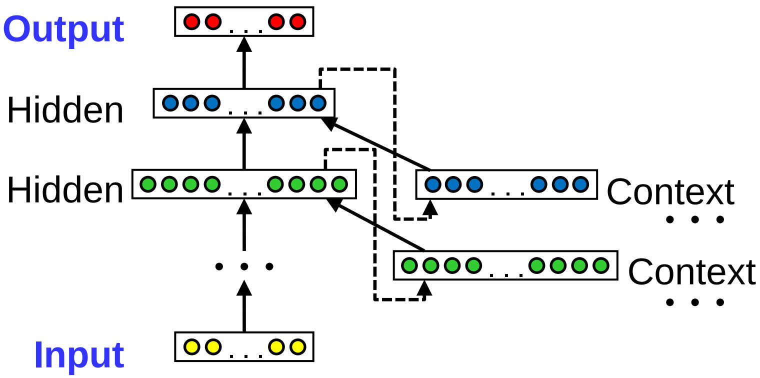
  Examples:
  - Elman Network
  - Jordan Network



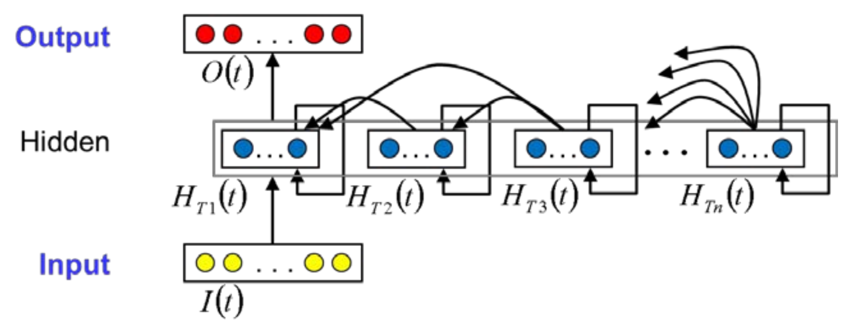
- Issues with the SRN:
  - Time leaks or disturbances in the sequences can be destructive
  - Vanishing/exploding gradient problem
  - *Training* methods are simple but can be *uninformed* or *slow*

Advanced recurrent neural architectures

- Constrain SRN towards easier training or easier capturing of task characteristics

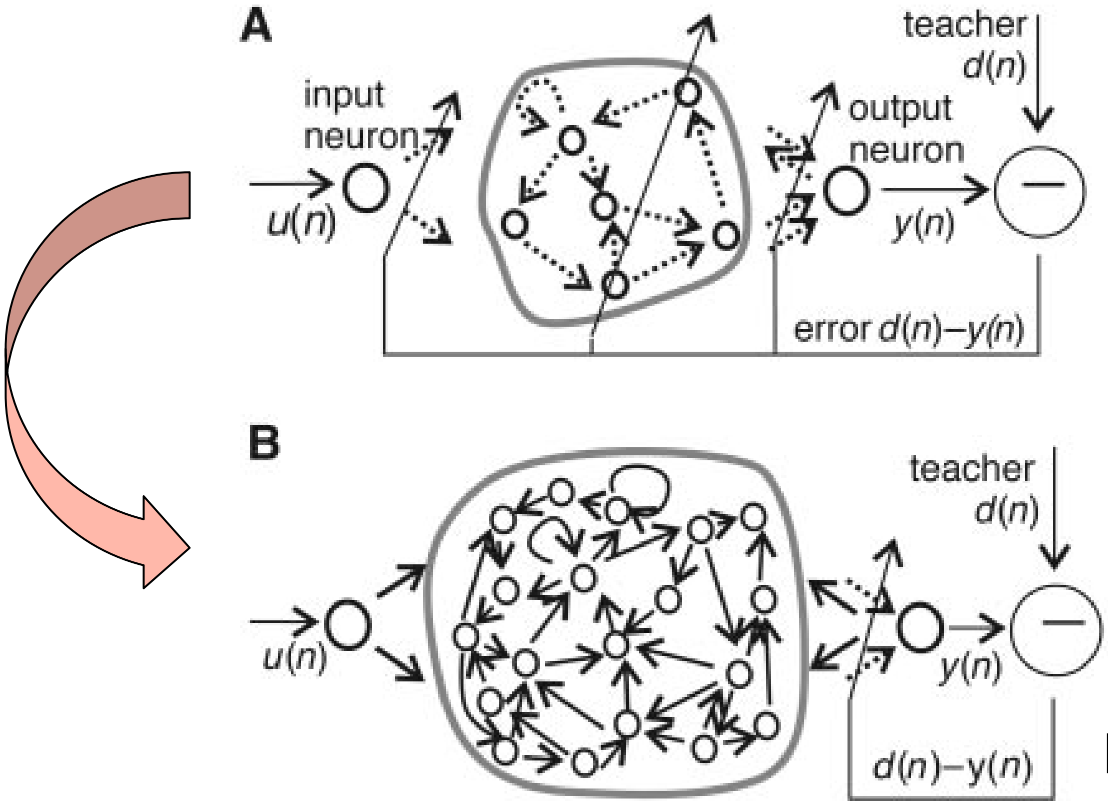


- Constrained approaches:
  1. Multiple context layers with *different timescales* based on hysteresis/leakage (RPN, MTRNN)
  2. Learn multiple timescales by constraining activation times (CWRNN)
  3. Gated unit activations (LSTM, GRU)



Echo State Networks

- Do not train all the network (e.g. *just the output layer*)
- *Randomize* untrained connections (input & hidden layers)
- Use *linear* methods for training (e.g. Linear regression)



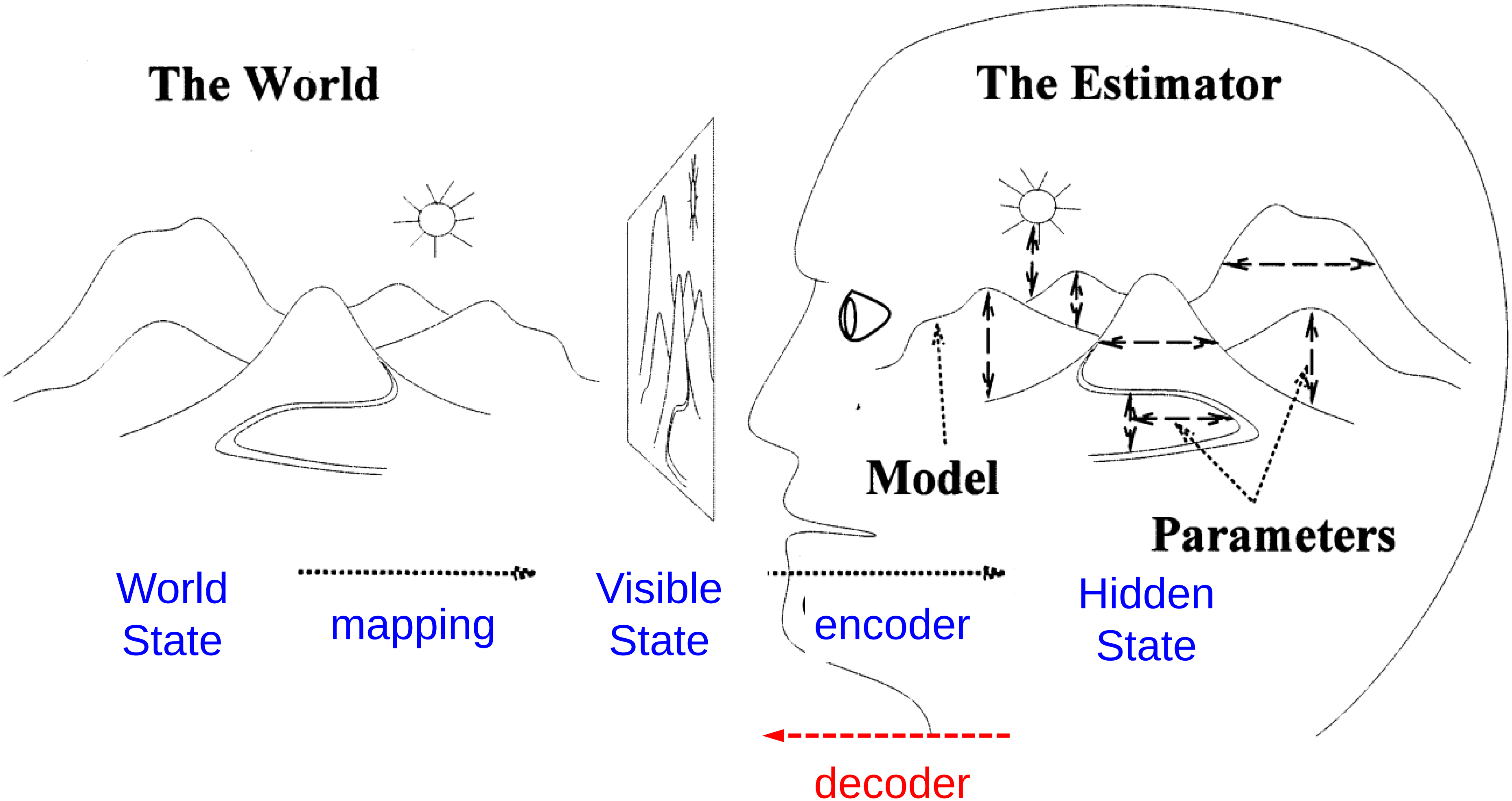
[Jaeger & Hass 2004]

Applications of recurrent neural networks

- They can remember things for a long time.
  - network has *internal state* and can ignore the input for a while
  - harder to train a recurrent net to keep information over long time gaps; better for *local context*

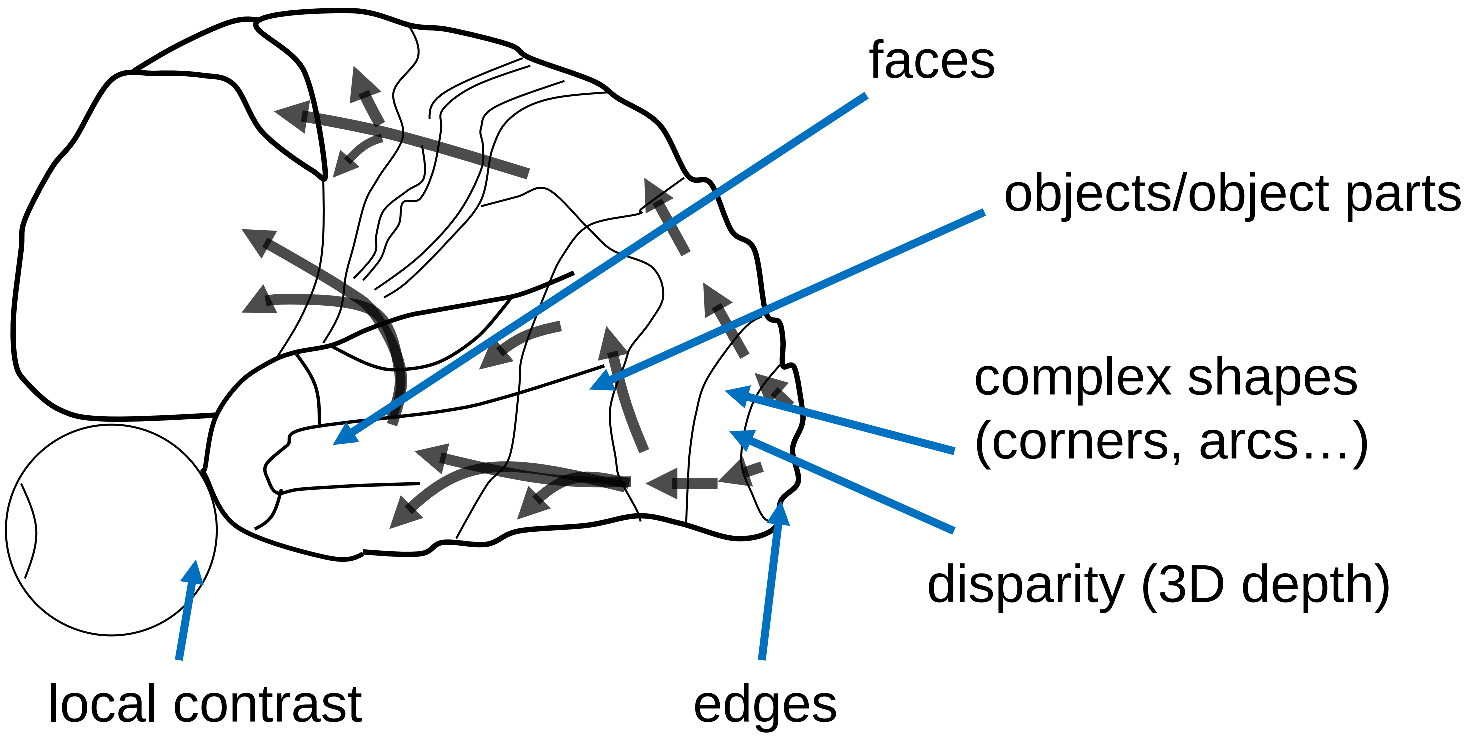
- They can *model sequential data* in a natural way than by using a fixed number of previous inputs to predict the next input
  - Tasks: Sequence prediction, classification, generation
  - Applications: Handwriting and speech recognition, attentive vision and keywords spotting, music composition ...

Generative Models



Hierarchical Representations in Vision

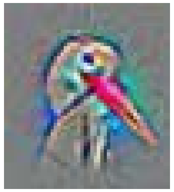
- Unsupervised learning with *constrained* autoencoders



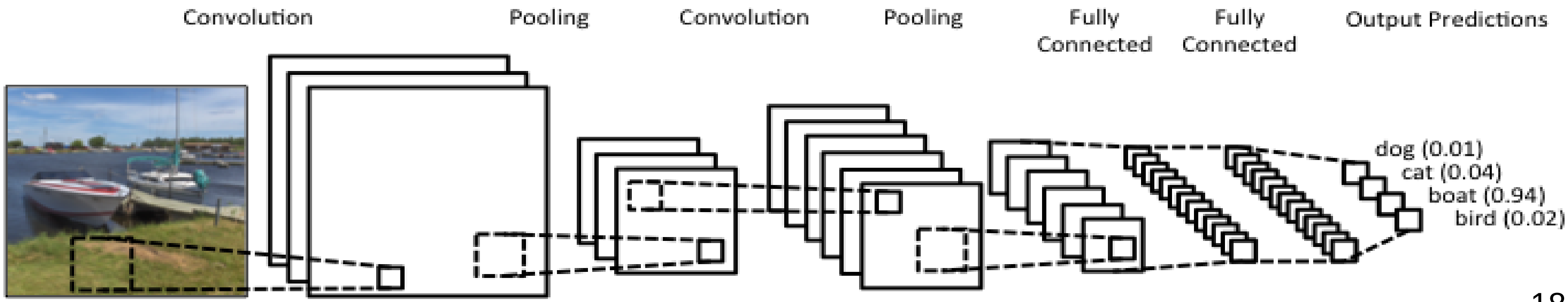
Deep Neural Networks

- Traditional computer vision: hand-crafted features
- Deep Learning: feature learning
  - Deep networks learn hierarchical abstractions:

    pixel → edges → shapes → objects

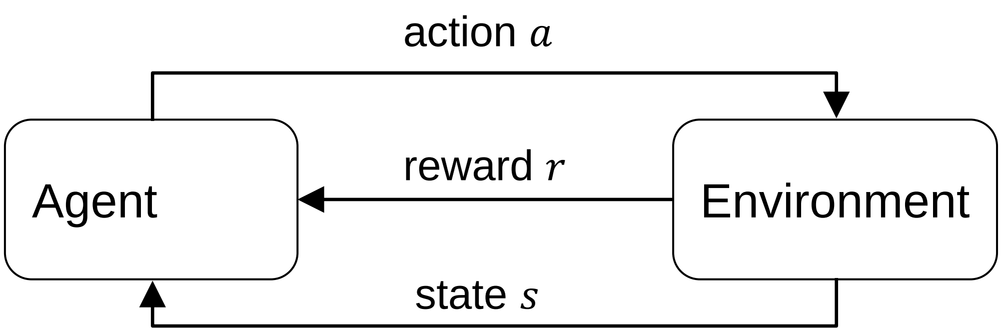


  - Convolutional Neural Networks:
    Repeated convolution (filtering) and pooling before feeding learned features to a MLP or RNN.
- Very effective, but needs lots of data and resources

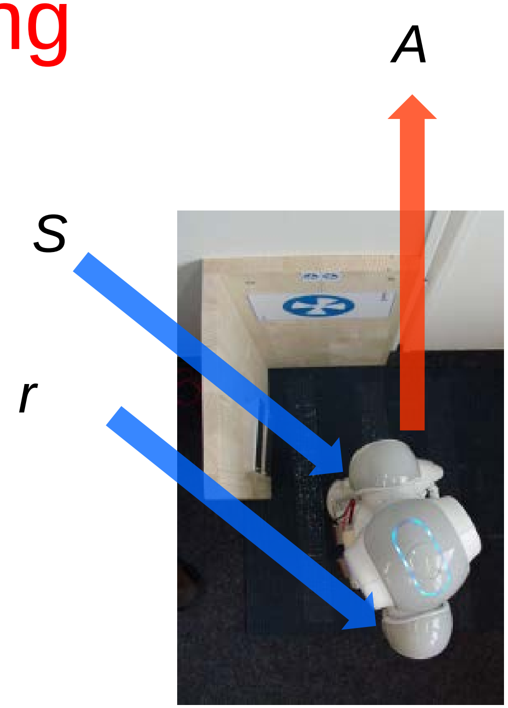


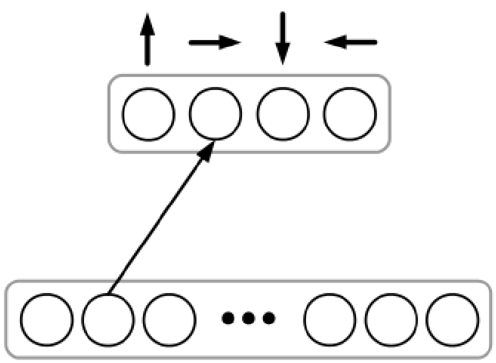
Reinforcement Learning

- Reinforcement Learning (RL):
  - Perceive state s
  - Select and perform an action a
  - Occasionally receive feedback r

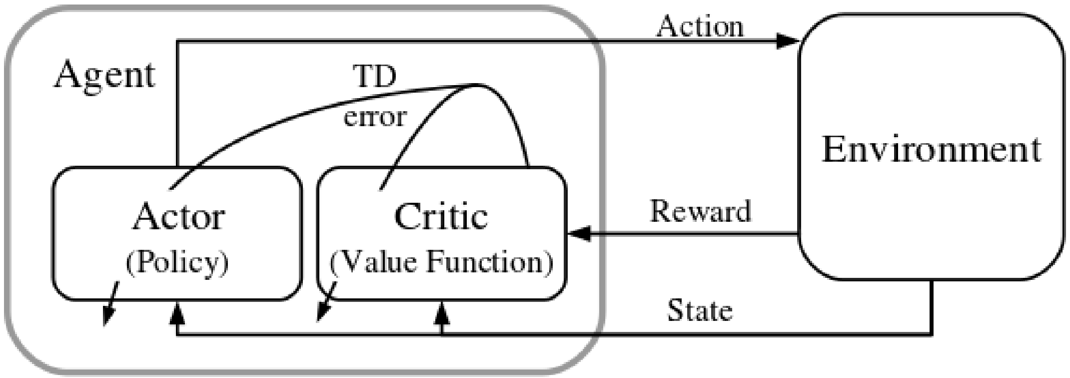


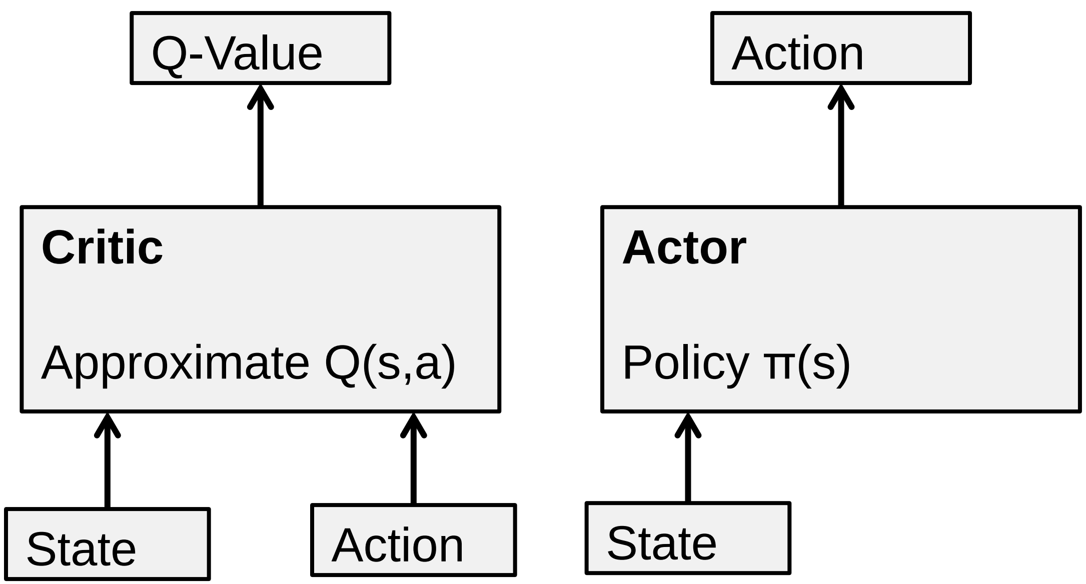
- Neural Networks can be used to learn both the policy and the value function.





Guided Actor-Critic Reinforcement Learning



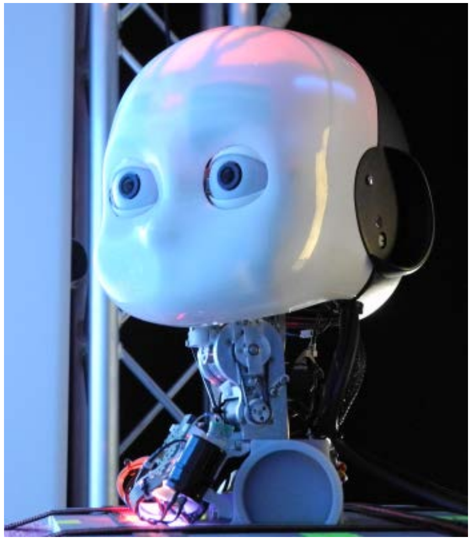


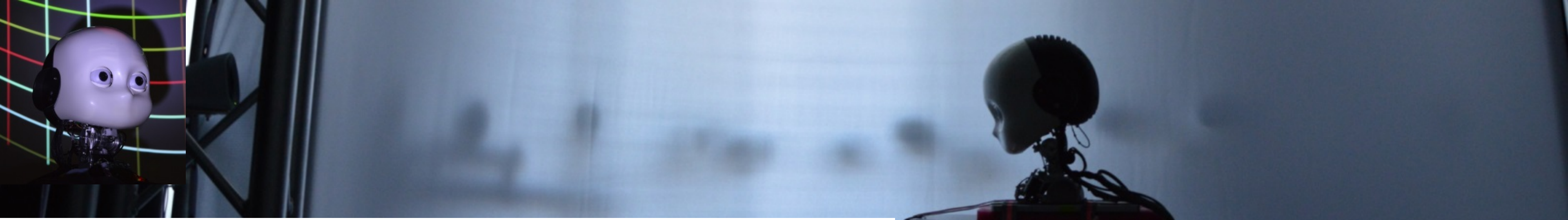
- If no reward was received from the critic:
  - If the agent moved closer to the goal state the Actor was rewarded
  - Otherwise the Actor was punished
- Example applications: Robot navigation, AlphaGo

Neural Networks:
Concluding remarks

- *Inspired* by the brain, we discovered *neural computing*
  - Neural network *architectures*
  - *Learning* with neural networks
  - *Hybrid* neural systems
  - Neural models for various *cognitive capabilities*

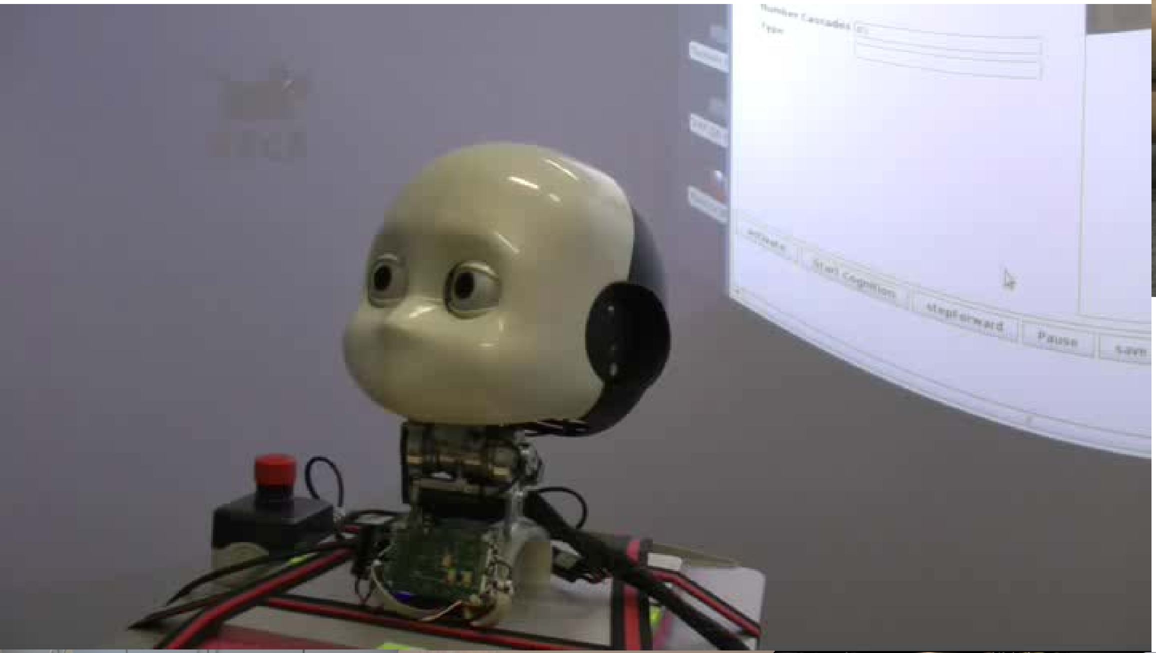
- ... and learned to apply these capabilities to artificial intelligent systems
  - Effective *classification* and *approximation*
  - *Adaptive* and robust learning robots



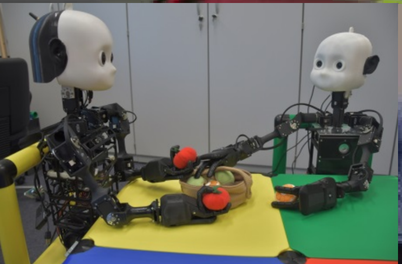


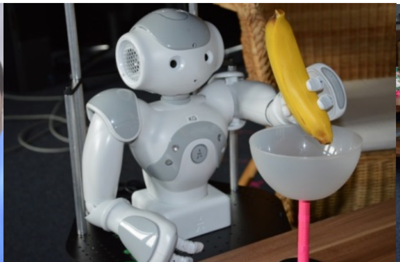
Neural computing in the lab

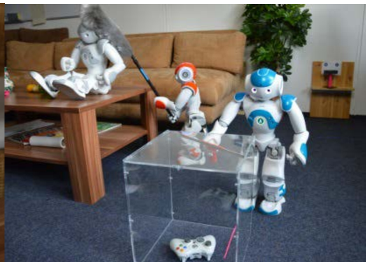
Face recognition and tracking with recurrent neural networks

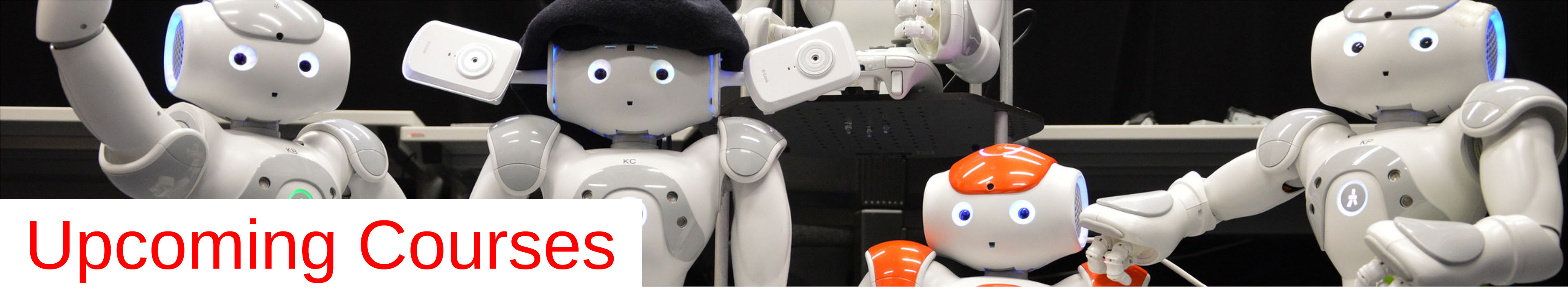


Predictive docking and grasping with neural reinforcement learning





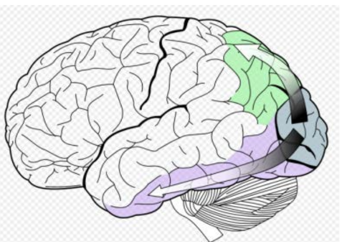


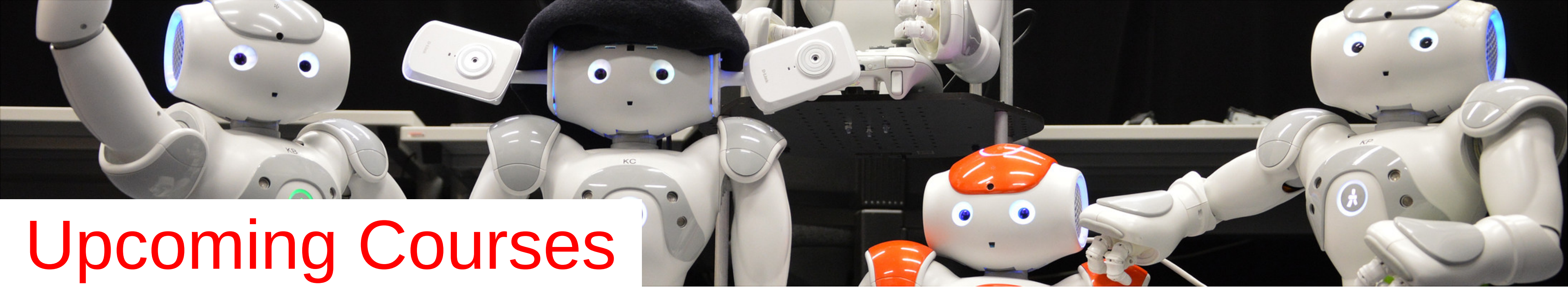


Upcoming Courses

L+S Bio-inspired Artificial Intelligence (WS)

- Adaptation, learning, development, evolution!
- Learn about the nature and human!
- Learn about brain and mind!
- Experience how to build intelligent systems and robots!

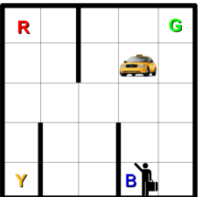


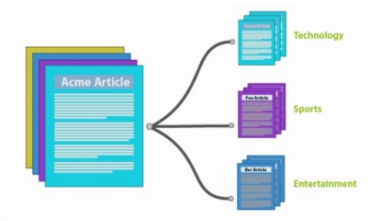
Upcoming Courses

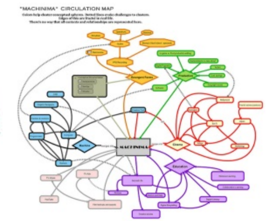
L+S Knowledge Processing for Intelligent Systems (WS)

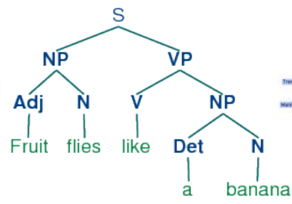
- Learn about *knowledge- and stat-based models* for e.g. natural language understanding, problem solving, knowledge retrieval and more.

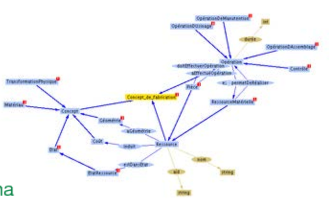
- Take part in the *practical seminar*:
  Pick an interesting challenge, implement and evaluate your approach freely over the course of the semester

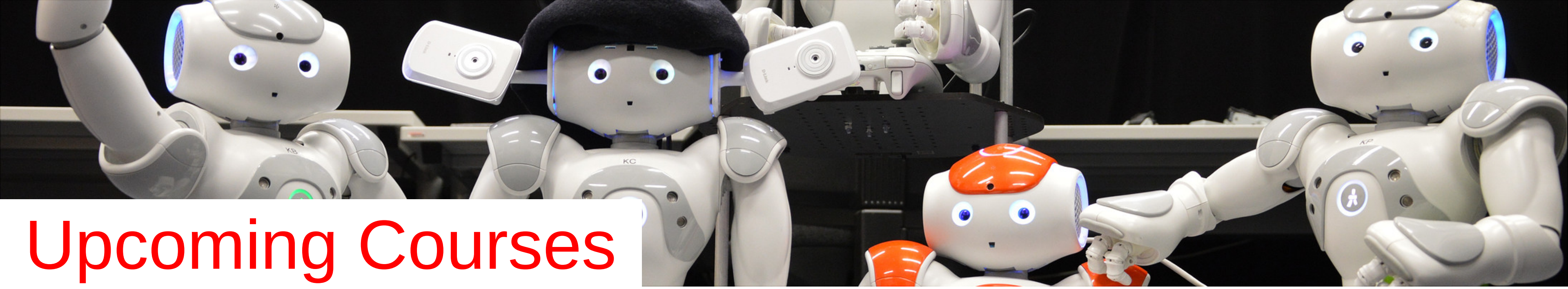








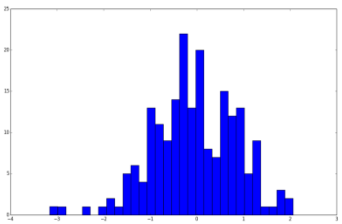


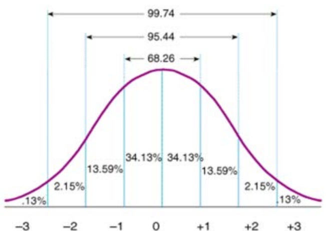


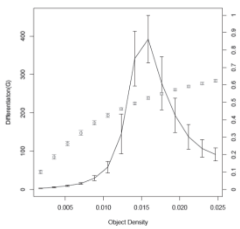
Upcoming Courses

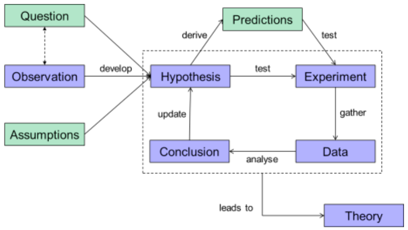
M+P: Research Methods (WS)

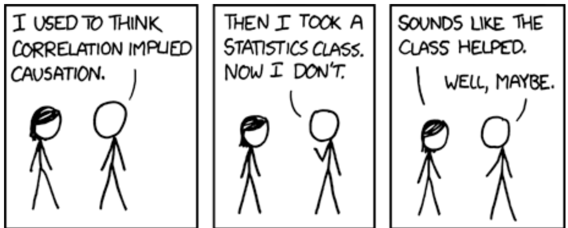
- Types and design of *empirical studies*
- Gain applicable knowledge on experiment design and execution
- *Statistical methods* for analysis of quantitative and qualitative data
- *Important aspects* surrounding studies like publication or legal regulations
- Interactive lecture, tightly coupled with a mixture of seminar and practical course

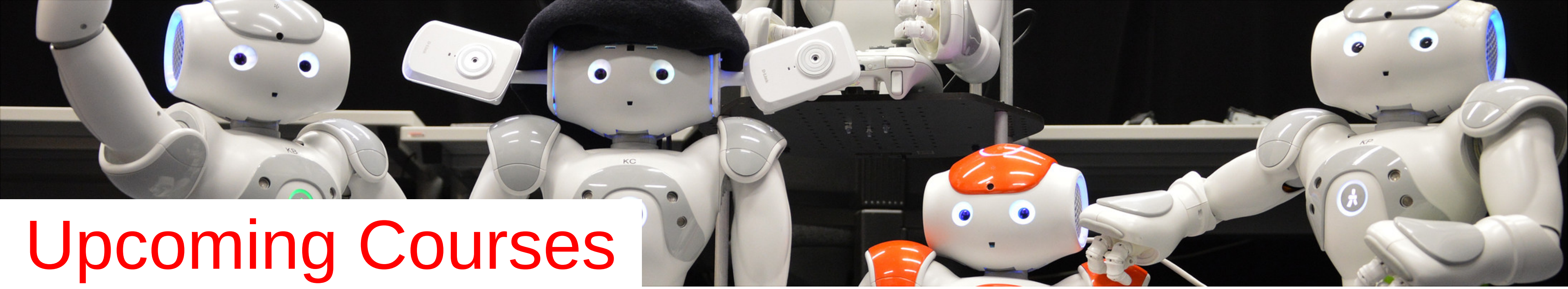








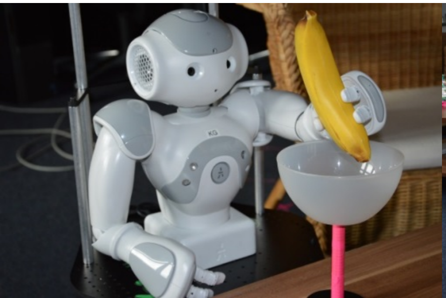


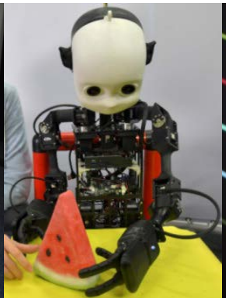
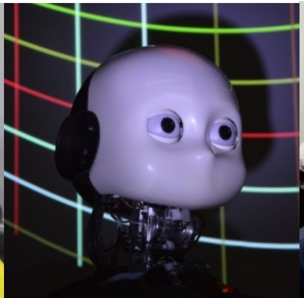
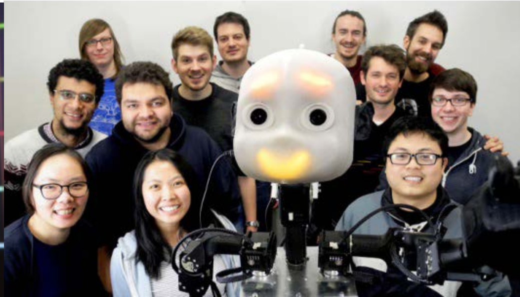


Upcoming Courses

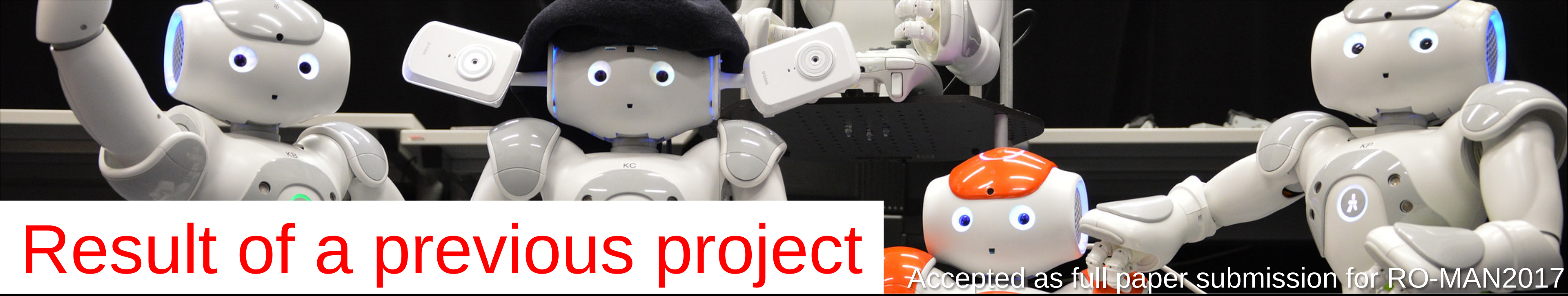
Master Project: Human-Robot Interaction (WS)

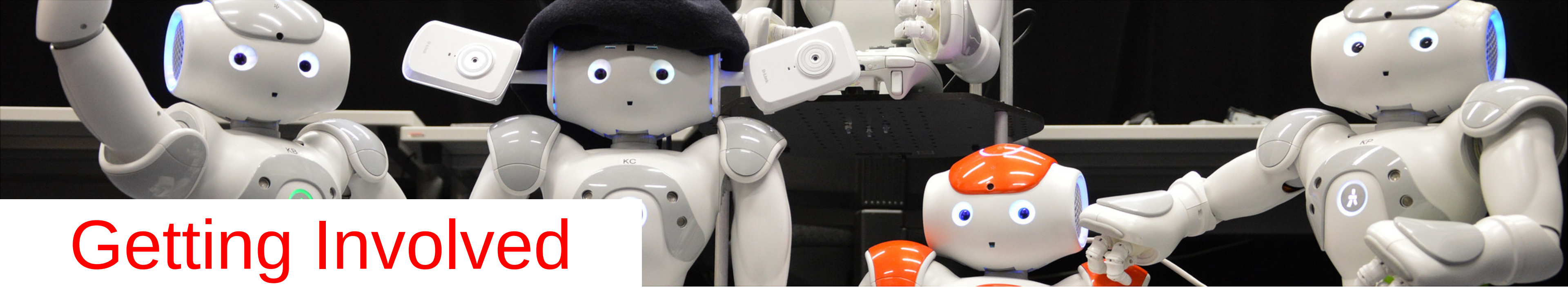
- Challenge: Robotic device capable *of interacting with people* as naturally as we interact with each other
- Approach: solve a *simple task* in a *complex environment*, e.g. “serve coffee!”
- Inspiration: RoboCup@home tasks
- Chance: Follow up on ideas & progress of recent groups



Result of a previous project





Getting Involved

- Oberseminar Knowledge Technology

  https://www.inf.uni-hamburg.de/en/inst/ab/wtm/teaching.html/

- BSc or MSc thesis

  https://www.inf.uni-hamburg.de/en/inst/ab/wtm/teaching/thesis.html

  Feel free to discuss your own ideas with us!

  {alpay, eppe, griffiths, heinrich, jirak, magg, twiefel, weber}@informatik.uni-hamburg.de

https://www.knowledge-technology.info

Upcoming Dates

Oral Exams:

- Wed 08. Aug 2018; 10–13h & 14–17h; F-230 / F-233
- Thu 09. Aug 2018; 10–13h & 14–17h; F-230 / F-233
- Fri 10. Aug 2019; 10–13h & 14–17h; F-230 / F-233

- Wed 26. Sep 2018; 10–13h & 14–17h; F-230 / F-233
- Thu 27. Sep 2018; 10–13h & 14–17h; F-230 / F-233

- Registration for exams:
  - 26th Jun – 4th Jul; 8th Aug – 15th Aug, 09–15h; at study office.

  ⇨ register now!

- Reminder: Seminar blocks
  - Thu – Fri 19th – 20th Jul & Mon – Tue 23rd – 24th Jul

Info on the seminar

- Be crisp and stay in time limit
- Quickly get into the main research issue
- Less than 2 minutes for slide
- Most important ‘medium’ for the talk: the speaker!
  - Slides are just a tool to support the talk
  - Focus on the audience
- Use examples, videos, illustrations, metaphors, and gestures to get your points across
- Talk freely, be confident, and have fun!

Reminder: Paper submission deadline today