Learning

Optimizing the behavior!

in order to reach this:
Revealing inherent structures in the inputs: unsupervised learning: dimension reduction

Learning what leads to the reward: reinforcement learning

Learning what the teacher says: supervised learning
Learning at psychological level

- Classical conditioning

Before conditioning

<table>
<thead>
<tr>
<th>FOOD (UCS)</th>
<th>SALIVATION (UCR)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bell

<table>
<thead>
<tr>
<th>BELL</th>
<th>NO RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

During conditioning

<table>
<thead>
<tr>
<th>BELL + FOOD (UCS)</th>
<th>SALIVATION (UCR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ding ding</td>
<td></td>
</tr>
</tbody>
</table>

After conditioning

<table>
<thead>
<tr>
<th>BELL (CS)</th>
<th>SALIVATION (CR)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WATCH WHAT I CAN MAKE PAVLOV DO. AS SOON AS I DROOL, HE'LL SMILE AND WRITE IN HIS LITTLE BOOK.
Hebb's rule

When an axon of cell A is near enough to excite cell B, and repeatedly or consistently takes part in firing it, some growth process or metabolic change takes part in one or both cells such that A's efficiency, as one of cell firing B, is increased"  
(Hebb, The Organization of Behavior, 1949)

\[ \tau \frac{dw_i}{dt} = \nu \cdot u_i \]
Hebb's rule in an experiment at population level

LTP – long term potentiation
LTD – long term depression
Rate-based and spike-time dependent learning rules
What could be represented by a simple, one layered, feed forward network called perceptron?

It is able to learn many functions, but there are some exceptions such as XOR.

Problem:
The linearly inseparable functions are more numerous as the dimension of the problem increases.
In two dimensions the problem can be transformed: this requires a two layered network

With this two layered network, all the two dimensional Boolean-functions can be learned.

But in higher dimensions?

The weights and the thresholds appropriate to the XOR solution:
A possible solution: increasing the embedding dimension

In three dimension the XOR problem is linearly separable. As the embedding dimension increases, the fraction of linearly inseparable logical functions vanishes.
The two layered perceptron

The aim $z = F(x)$ to learn the $F$ function. By increasing the number of neurons in the hidden layer, the two layered perceptron is able to represent any $F$ function. It is an **universal approximator**.

But how to set the weights?

It is possible to train only with error backpropagation algorithm.

The number of neurons in the hidden layer corresponds to the embedding dimension.
Supervised learning: the backpropagation algorithm.

\[ y_j = g(u) = g(a_{0j} + \sum_{j=1}^{N_{in}} a_{ij} x_i) \]

\[ z_k = g(b_{0k} + \sum_{j=1}^{N_{hid}} b_{jk} y_j) \]

\[ g(u) = \frac{1}{1 + e^{-u}} \]

\[
E = \frac{1}{2NK} \sum_{k=1}^{N} \sum_{k'=1}^{K} (z_{k'} - t_{k'})^2
\]

\[ \Delta w(t) = -\eta \frac{\partial E}{\partial w} + \mu \Delta w(t - 1) \]

The required output should be presented for every input sample. The difference (error) is used to set the weights backward. For example: labeled data required in case of a classification problem.

The implementation is not too biologically realistic. But often used in artificial neural networks (ANN).
Perceptron for time series prediction
COMPUTATION WITH ATTRACTORS

NETWORK STRUCTURE
DYNAMIC SYSTEM

parameters
initial values
external inputs

geography of the attractor–basin portrait ("categories")
qualities to be classified

"windows" to the external world

"E"([a])

- How dynamic systems can process continuously varying inputs
- How to preserve (if at all) the power of computation with attractors
Hopfield-network: a recurrent NN
Hopfield-network: a recurrent NN

- Associative memory
- MCP neurons or rate models
- Patterns to be learned: binary vectors
- Symmetric weight matrix
- Dale's law: a cell can't be both excitatory and inhibitory is violated
- An example of the recurrent networks in the brain: hippocampus CA3 region, ...
Hopfield-network: a recurrent NN

Energy for neuron pairs:

$$e_{ij} = -w_{ij}x_ix_j$$

Connections are set as:

$$w_{ij} = \sum_{s=1}^{n} x_i^s x_j^s$$

Total energy:

$$E = \sum_{i,j=1}^{N} e_{ij} = -\frac{1}{2} \sum_{i,j=1}^{N} w_{ij}x_ix_j$$

![Hopfield Network State Space](image)

![Energy function](image)
Hopfield-network: a recurrent NN

Capacity estimation of the network:

\[ M \approx \frac{N}{2 \log_2 N} \]

In case of a CA3 region of the hippocampus: 
cca. 200000 neurons, cca. 6000 patterns could be stored
MODIFICATIONS OF THE CLASSICAL SCENARIO

Fixed points vs. strange attractors

- Not only fixed points
- but also limit cycle
- or strange attractors might be involved

- neurophysiological experiments,
  theoretical studies
- structural conditions for the possible mechanism
  for the generation of rhythms and chaos can be given
  based on the notions of qualitative stability and instability

W. J. Freeman
The somatosensory map
Kohonen's self-organizing map

\[ D_{jk} = \sqrt{\sum_{k=1}^{N} (x_{kn} - w_{nk}^j)^2} \]

Winner take all

\[ \overline{W}^j(t+1) = \overline{W}^j(t) + \eta(t) N(c, r)[X_k - \overline{W}^j(t)] \]
Kohonen's self-organizing map

Generates feature maps

Captures the structure in the inputs
Reinforcement learning: an actor-critique architecture

FIGURE 2. The actor-critic system. a: An input layer of place cells projects to the critic cell, C, whose output is used to evaluate behavior. Place cells also project to eight action cells, which the actor uses to select between eight possible directions of movement from any given location. b: An example of a Gaussian place field (x and y axes represent location, z axis represents firing rate).
Parallel learning of policies and the values
The temporal difference rule:

The critic depends on position p:

\[ C(p) = \sum_i w_i f_i(p) \]

The value function (V) must satisfy:

\[ V(p_i) = \langle R_i + \gamma R_{i+1} + \gamma^2 R_{i+2} + \cdots \rangle \]

Where \( \gamma \) is a constant discount factor for predicted (not actual) reward. From this the consistency of the value:

\[ V(p_i) = \langle R_i \rangle + \gamma V(p_{i+1}) \]

A well trained critic should satisfy the same consistency assumption:

\[ C(p_r) = \langle R_r \rangle + \gamma C(p_{r+1}) \]

The actual difference between the two sides governs the learning, this is called the temporal difference learning rule:

\[ \delta_i = R_i + \gamma C(p_{i+1}) - C(p_i) \]

The weight changes are proportional to the difference:

\[ \Delta w_i \propto \delta_i f_i(p_r) \]
The temporal difference rule:

Two sources of value:

\[ \delta_t = R_t + \gamma C(p_{t+1}) - C(p_t) \]

The actually achieved reward

The difference between the expected and the achieved increase.

Together expresses the difference between the expected and actual reward.
The effect of reward in dopaminergic cell of basal ganglia

An interpretation:

Dopamine cells signal the difference between the expected and received reward.
Principal component analysis

\[
\overline{x}_n = \frac{1}{K} \sum_{k=1}^{K} x_{kn} \quad \sigma_n = \sqrt{\frac{1}{K-1} \sum_{k=1}^{K} (x_{kn} - \overline{x}_n)^2}
\]

\[
S = \left( \text{covar}[y_i, y_j]_{i,j=1}^{N} \right)^N \left( \frac{1}{K-1} \sum_{k=1}^{K} y_{ki} y_{kj} \right)^N_i, j=1 = \frac{1}{K-1} y^T \cdot Y
\]

\[S \cdot X' = \lambda \cdot X'
\]

\[Z = X \cdot A
\]
Principal component network, derivation of Oja's rule:

\[ w_i(n + 1) = w_i(n) + \eta y(x(n)) x_i(n) \]

\[ w_i(n + 1) = \frac{w_i(n) + \eta y(x(n)) x_i(n)}{(\sum_{j=1}^{m} [w_j(n) + \eta y(x(n)) x_j(n)]^p)^{1/p}} \]

\[ w_i(n+1) = \frac{w_i(n)}{(\sum_j w_j^p)^{1/p}} + \eta \left( \frac{y(n)x_i(n)}{(\sum_j w_j^p)^{1/p}} - \frac{w_i(n) \sum_j y(n)x_j(n)w_j(n)}{(\sum_j w_j^p)^{1+1/p}} \right) + O(\eta^2) \]

\[ y(x(n)) = \sum_{j=1}^{m} x_j(n)w_j(n) \quad \|w\| = (\sum_{j=1}^{m} w_j^p)^{1/p} = 1 \]

\[ w_i(n + 1) = w_i(n) + \eta y(n)(x_i(n) - w_i(n)y(n)) \]
Independent component analysis (ICA)
Independent component analysis (ICA)