Introduction to cognitive science
Session 5: Connectionist paradigm

Martin Takáč
Centre for cognitive science
DAI FMFI Comenius University in Bratislava

Príprava štúdia matematiky a informatiky na FMFI UK v anglickom jazyku
ITMS: 26140230008
Last week – Symbolic paradigm

- Universal Turing machine
  - Can compute anything that is computable.

- Symbol systems
  - Manipulate symbols by syntactical rules
  - How do symbols acquire meaning?

- Problems
  - Frame problem
  - Symbol grounding problem
In this session:

- Symbolic versus subsymbolic representation
- Distributed representation
- Gradedness
- Graceful degradation
- Robustness
- Feedback
- Neural architecture & knowledge
Connectionist (sub-symbolic) paradigm

- Inspired by biological brains
- Network
  - Many simple processors
    - Neurons <-> Units
  - Connectivity
    - Axons/synapses <-> Weighted connections
  - Parallel processing
  - Distributed representation
  - Representation - **not static**
Brain is comprised of networks of neurons connected and communicating via synapses.

$10^{11}$ neurons

$10^4$ synapses in and out


Learning and brain plasticity

- Traditionally: learning is an acquisition of memories.

- Memory is an organism's ability to store, retain, and subsequently recall information.

- Neural correlate of learning is brain plasticity.

- Brain plasticity (neuroplasticity) is a lifelong ability of the brain to reorganize connectivity of neural circuits based on new experience.

- Brain plasticity is based on synaptic plasticity.
Hebb’s rule of synaptic plasticity (1949):

When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.”
Brain plasticity = refinement of the connectivity of neural networks based on **synaptic plasticity**.

Synaptic plasticity is the ability of the synapse to change its strength (pre- and postsynaptic mechanisms).
Firing threshold and spikes

Excitation: EPSP
Inhibition: IPSP

EPSP - IPSP > \( \theta \)

Rate or frequency of spike train (Hz)
Spiking versus rate neuron models

Spiking model: output depends on timing of input spikes

Rate model: output depends on the sum of input rates
Perceptron – neuron (rate) model

Dendritic tree  
axon  
soma  
spines  
terminals  
output

input

\[ x_1, x_2, \ldots, x_{n+1} = \theta \]

\[ w_1, w_j \in \mathbb{R} \]

\[ w_j > 0 \text{ excitation} \]
\[ w_j < 0 \text{ inhibition} \]

\[ o \in \mathbb{R} \]
Neuron rate model

\[ x_j \in \mathbb{R} \]

Input vector = mean spike rates

\[ w_j \in \mathbb{R} \]

\[ w_j > 0 \text{ excitation} \]

\[ w_j < 0 \text{ inhibition} \]

\[ w_{n+1} = \theta \]

\[ x_{n+1} = -1 \]

\[ f = o = \text{activation function} \]

\[ o \in \mathbb{R} \]

Output = mean spike rate
Rosenblatt's binary perceptron (hyperplane)

\[ o = f(\text{net}) = f(w \cdot x) = f\left(\sum_{j=1}^{n+1} w_j x_j \right) = f\left(\sum_{j=1}^{n} w_j x_j - \theta \right) \]

\[ f(\text{net}) = \text{sign}(\text{net}) = \begin{cases} 
+1 & \text{if net} \geq 0 \Leftrightarrow \sum_{j=1}^{n} w_j x_j \geq \theta \\
-1 & \text{if net} < 0 \Leftrightarrow \sum_{j=1}^{n} w_j x_j < \theta 
\end{cases} \]
General learning rule

The weight vector changes as a function of the product of the input vector $x$ and the learning signal $s(t)$

$$w_j(t + 1) = w_j(t) + \Delta w_j(t) = w_j(t) + \alpha \ s(t) \ x_j(t)$$

- $0 < \alpha \leq 1$ is the learning speed

- Based on the type of the learning signal we distinguish:
  - Supervised learning
  - Reinforcement learning
  - Unsupervised learning
**Types of learning**

**Supervised learning:** weights are adjusted according to the *desired* output (perceptron, MLP, RBF, RNN)

\[ s = s(w, x, d) \]

**Reinforcement learning:** weights are adjusted according to the *reward* (MLP, RBF) – extension of the supervised learning

\[ s = s(w, x, r) \]

**Unsupervised learning:** weights are adjusted according to the *statistics* of the input:

\[ s = s(w, x) \]
History of connectionism

- **1950’** – perceptron (Rosenblatt, 1958)
  - Limitations
    - Linearly separable classification problems
- **1970’** - SOM
- **1980’** – multiple layers
  - Hidden layer
  - Back propagation algorithm
- **1990’** – recurrent networks
- **2000** – deep networks
MLP (Multilayer Perceptron)

MLP is a nonlinear differentiable function (hyperbolic tan, sigmoid, Gaussian, etc.)
Somatosensory and motor systems in animals

topological mapping from body to cortex
Self-Organizing Map (SOM): architecture

$n$ linear neurons in the output layer:

\[ O_i = \sum_{j=1}^{m} w_{ij} x_j = w_i \cdot x \]

\[ w_i = (w_{i1}, \ldots, w_{ij}, \ldots, w_{im}) \]

weight vector of neuron $i$

\[ x = (x_1, \ldots, x_j, \ldots, x_m) \]

Input pattern = vector of real values. They feed each neuron.
SOM training: competition phase

For each input vector in the training set, we find a winner neuron:

- According to maximal dot product: \( i^* = \text{argmax}(w_i, x) \)
- Or according to minimal Euclidean distance: \( i^* = \text{argmin}_i d_E(w_i, x) \)

All weights are initialised as small random numbers.

Training set consists of input vectors only, which are presented in random order:

\[ A_{\text{train}} = \{x^1, x^2, \ldots, x^p, \ldots, x^P\} \]
SOM training: weight update & cooperation

The weights of the winner and its neighbours in $N_{i*,i}$ are updated:

$$w_i(t+1) = w_i(t) + \alpha(t) \cdot N_{i*,i}(t) \cdot [x(t) - w_i(t)]$$

The weight vectors of $i \in N_{i*,i}$ move closer to the current input.

Training set consists of input vectors only, which are presented in random order:

$$A_{train} = \{x^1, x^2, ..., x^p, ..., x^P\}$$
Recurrent neural networks

- Able to process dynamic temporal patterns
- Additional **feedback loop**
  - recycles some aspects of the networks activity at time $t_1$ together with input in $t_2$.

(Clark, 2001)
Deep networks

## Symbolic vs. connectionist

<table>
<thead>
<tr>
<th>Symbolic</th>
<th>Connectionist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>Parallel</td>
</tr>
<tr>
<td>Logic &amp; Deduction</td>
<td>Patterns &amp; Induction</td>
</tr>
<tr>
<td>Algorithm must be known</td>
<td>Learning from examples</td>
</tr>
<tr>
<td>Noise intolerant</td>
<td>Noise tolerant</td>
</tr>
<tr>
<td>Semantically interpretable</td>
<td>Semantic interpretation often not feasible</td>
</tr>
<tr>
<td>Not robust</td>
<td>Robust (graceful degradation)</td>
</tr>
</tbody>
</table>
Symbolic vs. connectionist

- **DECTalk**
  - Hand coded rules and exceptions

- **NETtalk (Sejnowski & Rosenberg, 1987)**
  - 7 groups of input units
    - Each 29 units
    - letter specification
  - 4th unit – target
  - Others – context
  - Hidden layer – 80
  - Output – 26
    - 56 phonemes
  - Weights – 18,629

(Clark, 2001)
Distributed representation

- As opposed to a body of declarative statements
- Knowledge in the set of connection weights and structure of the network
  - Expressed by the *simultaneous activity* of a number of units
  - Semantically related items are represented by syntactically related (partially overlapping) patterns of activation.
If new input resembles an old one in some aspects, it will yield a response in partial overlap.
Generalization in deep networks

Graceful degradation

- In case of damage to the network, it can still produce sensible responses
Lack of understanding

- How to understand the knowledge and strategies that the network is actually using to drive its behavior?

  - Posttraining analysis
    - Statistical analysis
      - Cluster analysis
      - Very different networks with the same training data yield similar statistical properties
    - Systematic interference – damage to units/connections
      - “lesions”
Problems of (current) connectionism

- **Simplification of task**
  - Usually not dealing with real-world problems
  - Discrete, well-defined problems
  - Not general

- **Scaling**
  - Models use usually small numbers of units
  - Solutions that work well for small networks with narrow focus fail to deal with large input spaces and multiple tasks

- **Level of detail**
  - Blue brain??
What should be clear by now:

- Symbolic versus subsymbolic representation
- Distributed representation
- Gradedness
- Graceful degradation
- Robustness
- Feedback
- Neural architecture & knowledge
Questions?