Hebbian learning and Hopfield networks

Pietro Berkes, Brandeis University
“Classical” models of learning

• Characterized by deterministic update and learning rules
• The models have a number of parameters that are adjusted such that the model performs a certain function
• Examples: neural networks, support vector machines, PCA, ...
• Issues: cannot cope with uncertainty
Hebb’s rule

• Most of learning in the brain is unsupervised
• Brilliant idea by Hebb (1949):
  *cells that fire together, wire together*
Hebb’s rule

• Most of learning in the brain is unsupervised
• Brilliant idea by Hebb (1949): *cells that fire together, wire together*
Hebb’s rule

• Most of learning in the brain is unsupervised
• Brilliant idea by Hebb (1949):
  *cells that fire together, wire together*

```
  “fruity-smell” neuron  “yellow” neuron
```

Hebb’s rule

• Most of learning in the brain is unsupervised
• Brilliant idea by Hebb (1949):
  *cells that fire together, wire together*
Hebb’s rule

• Most of learning in the brain is unsupervised
• Brilliant idea by Hebb (1949): *cells that fire together, wire together*

“fruity-smell” neuron

“yellow” neuron
Hebb’s rule

• Most of learning in the brain is unsupervised
• Brilliant idea by Hebb (1949): *cells that fire together, wire together*
Hopfield network

• Hebb’s ideas where formalized much later: Hopfield network (1982)
• Most direct implementation of Hebb’s ideas:
  \[ \Delta w_{ij} = x_i x_j \]
• Note: local learning rule, no teacher
Architecture

• $x_i = +1$ or $-1$  
  (binary neural activity)

• $w_{ii} = 0$  
  (no self-connections)

• $w_{ij} = w_{ji}$  
  (symmetric, bidirectional connections)
Learning rule

• Set neurons $x_i$ to desired pattern
• Update weights as $\Delta w_{ij} = x_i x_j$

• i.e., if we have $N$ patterns $x_i^{(n)}$ the final weights will be

$$w_{ij} = \sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}$$
Activity rule

- For each neuron $i$:

1) compute activation $a_i = \sum_j w_{ij}x_j$

2) update state of neuron as $x_i = \begin{cases} +1 & a \geq 0 \\ -1 & a < 0 \end{cases}$

- Updates can be synchronous or asynchronous
Applications: denoising

noisy pattern

complete pattern
Applications: pattern completion

incomplete pattern  complete pattern
Applications: pattern recognition

novel pattern

stored pattern
Issues

• Capacity of the network: how many memories can be stored?
  Not many: $\alpha = \frac{M}{N} = 0.144$
  There is a whole literature about memory capacity and information storage in the hippocampus

• Robustness: how much can we modify a stored pattern such that we recover is perfectly/with a small error?
Hands-on part

• Download exercises from
  http://people.brandeis.edu/~berkes/data/cognitive_models/exercises2