Complexity in social dynamics : from the micro to the macro
Lecture 2 – Neural Networks

Franco Bagnoli

Namur 10/4/2008

1 Neural Networks

Microscopic level

1. Brain functions. Neurons. Neuron models. Neural Networks.
2. Dynamical Ising model. Phase transitions. Free Energy
3. Correlations and scaling.
4. Networks: regular lattices, random graphs, fully connected, small world and scale-free networks.
5. Mean field approximation.
6. Perceptron. Classification.
7. Attracting and feed-forward neural networks. Modeling association, generalization and memory.
8. Hopfield model. Capacity and robustness.

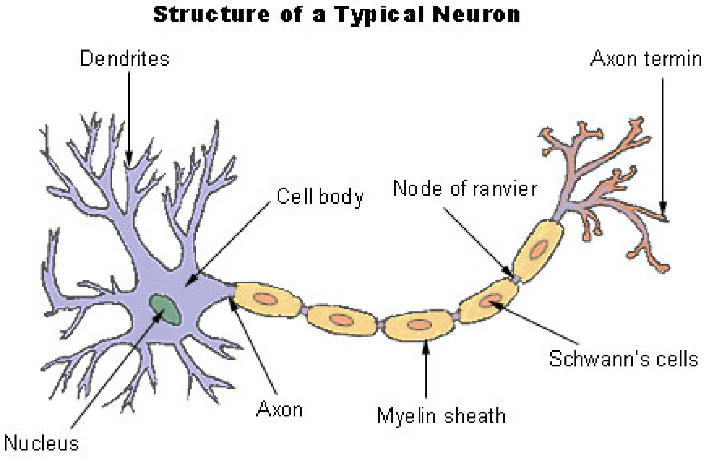
Brain functions

Some of the most relevant brain functions are

- Input filtering. For instance, vision consists first in recognizing shapes. Our Array of photoreceptors (the retina), communicates lightning informations to internal layerr. Informations are decomposed into elements (e.g., lines), and after some stage there are neurons that recognize lines of a given length and orientation.
- Classification. Similar inputs should excite same neurons. This implies throwing away informations.
- Memory (short and long-term).
- Recognition.
- Robustness (neurons die..).

Neurons

- A neuron is formed by a body (soma), a long tail (axon) and short “hairs” (dendrites).
- At the end of the axon there are connections (synapses) that attach to dendrites or to soma of other neurons. A neuron recieves some 10^4 connections.
- The neuron activity is formed by accumulation of electric potentials through ion pumps in the membrane, and discharge. An avalanche discharge may propagate along the axon.



Synapses

- Frequently enough discharges make synapses to release neurotransmitters and trigger other neuron's discharge.
- Synapses release neurotransmitters when they receive impulses ("spikes") at a sifficiently high rate, but each synapses has its own threshold.
- Hebb rule: connections that are "used" become more active (synapses enlarge or the neuron develops more synapses), while inactive synapses become less susceptible.

Neural models

- The neuron is a threshold machine: if the frequency of discharges is not large enough, no signal propagates.
- A typical model is the binary neuron. 1 for "firing" and 0 (or -1) for not firing.
- Synapses may be excitatory or inhibitory. We can model them through a matrix J_{ij}. $J_{ij} > 0$ for excitatory connections, and $J_{ij} < 0$ for inhibitory ones. $J_{ij} = 0$ for non-connected neurons.
- Each neuron may moreover have a different threshold activity $-\theta_i$.
- The dynamics is generally stochastic and asynchronous.

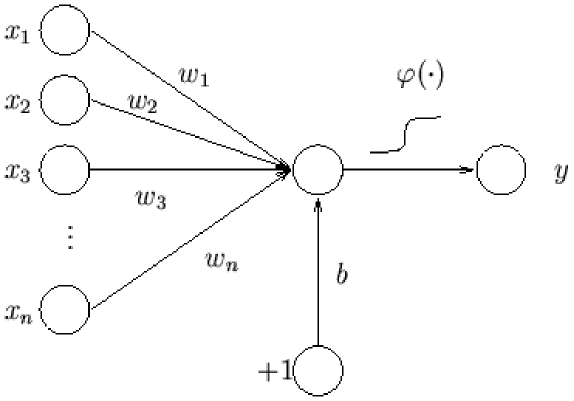
Perceptrons

- A single neuron can be represented as a perceptron, the inputs x_i are weighted by weigth w_i (corresponding to J_{ij}), the sum is compared to a threshold b and the output y is some sigmoid function of it

$$y = \phi\left(\sum_i x_i w_i - b\right)$$

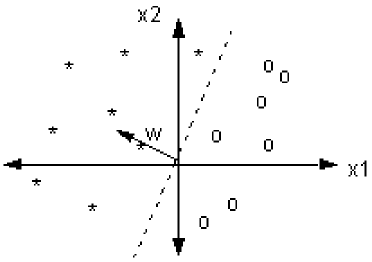
where

$$\phi(x) = \tanh(\beta x).$$

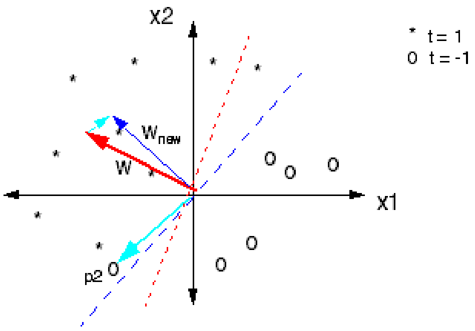


Perceptron classification

- A single perceptron can classify (for $\beta = 0$) only linearly separable inputs

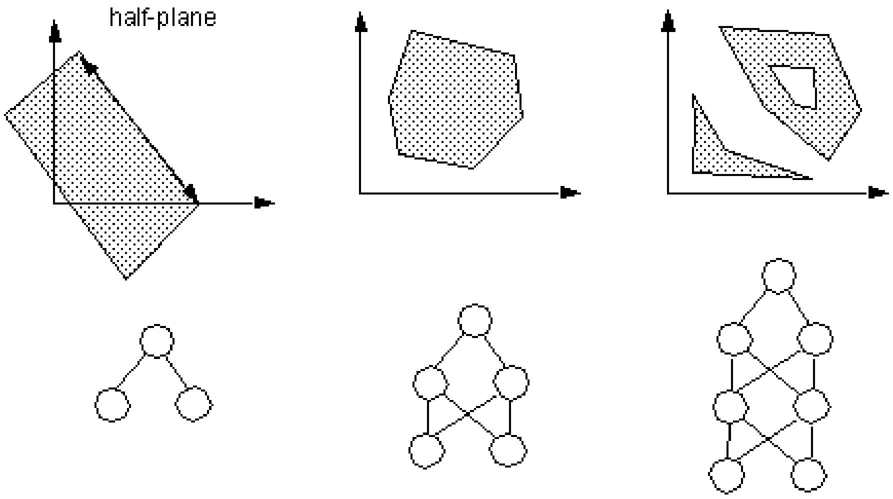


- Training (supervised learning) is performed on a set of inputs, by changing weights untill all patterns are correctly classified (can be done in a direct way).



Multi-layer perceptron

- The single-layer perceptron can only classify half-planes.
- For more complex boundaries, one can add further layers.



- In this case there is no direct rule for modifying the weights. We shall come back with simulated annealing.

Back-propagation in multipayer perceptrons

- Storing patterns in multilayer perceptrons is a minimization task.
- Present a training sample to the neural network.
- For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
- Adjust the weights of each neuron to lower the local error.
- Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
- Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error..

Unsupervised learning

- The process tries to mimicks the filtering stage of visual and auditory cortex.
- Kohonen maps have in general a regular grid topology, with local connections and a neigborhood of some radius.
- The input is feed to the network, and the network is left to relax.
- The "winning" neuron (that with highest output) changes the weight with neighboring neurons in order to enhance its response (competitive learning).
- The neighborhood is reduced with time.
- It is an example of a self-organizing process.

2 Attractor neural netwwork

Hopfield model

- Let us represent the state of a neuron with $\sigma_i \in \{-1, 1\}$ (it is called "a spin". Boolean variables s_i are related as $\sigma_i = 2s_i - 1$).
- A given neuron computes the "local field"

  $$h_i = \sum_j J_{ij}\sigma_j + \theta_i.$$

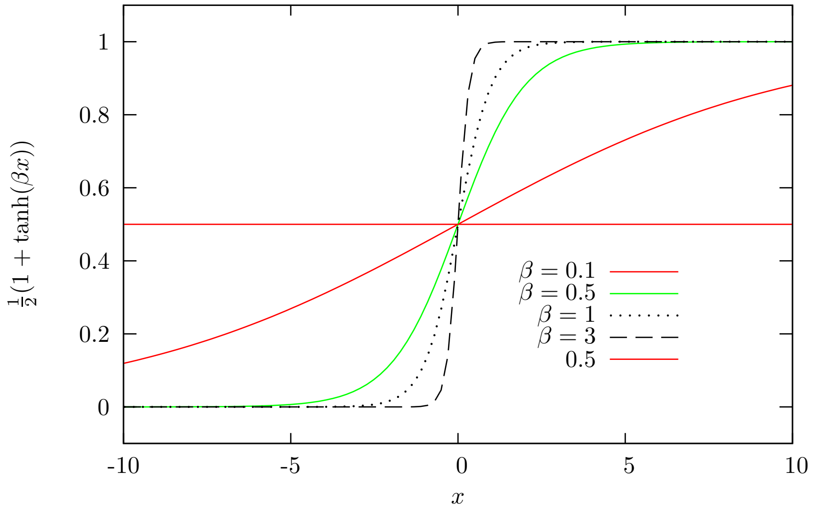
- The probability $p(\sigma_i'|h_i)$ that a neuron takes value σ_i' is

  $$p(\sigma_i'|h_i) = \frac{1}{2}\left(1 + \tanh(\beta h_i \sigma_i')\right).$$

  where $\beta = 1/T$ is a parameter (inverse of a "temperature") that modulates stochasticity.

Neural dynamics

- Asynchronous dynamics implies choosing a neuron at random, and trying to update its state. A "time step" is an essay for each neuron (i.e. N individual tentative updating – a "Monte-Carlo" step)
- The "response function" $f(x;\beta) = \frac{1}{2}\left(1 + tanh(\beta x)\right)$ is



Deterministic dynamics

- For $\beta \to \infty$ the dynamics is deterministic (but the random choice of updating).
- For symmetric J_{ij} the attractors are only fixed points (attractor neural networks). For asymmetric J_{ij} one can have cycles and chaotic states.
- The idea is that of relating the attractor to *memorized* patterns. The basins corresponds to retrieval regardless of "errors". The transients art the retrieval times.
- It can be easily realized that in this case there is a Lyapunov functional, the energy

$$E = -\sum_i \sigma_i h_j = -\sum_{ij} J_{ij}\sigma_i\sigma_j - \sum_i \theta_i \sigma_i,$$

  which is minimized by dynamics.

Landscape paradigm

- One can arrange all configurations $\boldsymbol{\sigma}$ in a 2^N-dimensional hypercube, links corresponding to single-spin flips.
- To each configuration one can associate its energy $E(\boldsymbol{\sigma})$. It gives a "landscape".
- The evolution for $\beta \to \infty$ corresponds to a steepest-descent search for a minima.
- The fixed points are local minima of the energy.
- There can exist "spurious states" which are high-energy minima with small basins.

Noisy dynamics

- For $\beta = 0$ all jumps are possible. The system is *ergodic* (the averages over replicas – ensemble averages – coincide with averages over time) and *self-averaging*(the averages over large systems concide with ensemble averages).
- For intermediate values of β, it is possible to exit local minima – the escape time depends on the width of the valley and the height of surrounding walls.
- It is possible to define a Lyapunov functional which is minimized: the *free energy*.

Asymptotic state

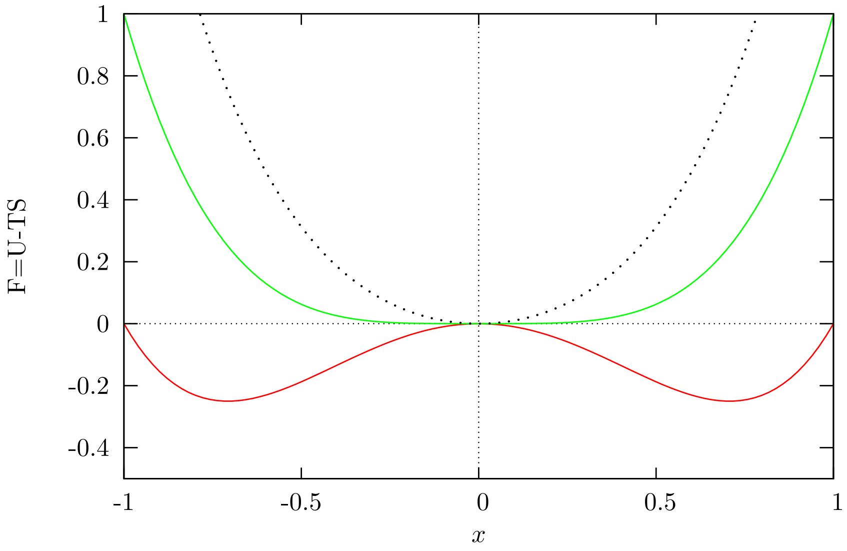
- When (as in this case) the dynamics obeys the *detailed balance* (or reversibility condition of Markov chains)
$$\frac{W(\boldsymbol{\sigma}_1|\boldsymbol{\sigma}_2)}{W(\boldsymbol{\sigma}_2|\boldsymbol{\sigma}_1)} = \frac{P(\boldsymbol{\sigma}_1)}{P(\boldsymbol{\sigma}_2)}$$
then $P(\boldsymbol{\sigma})$ is the asymptotic probability distribution.
- In this case it is given by
$$P(\boldsymbol{\sigma}) = \frac{\exp\left(-\beta E(\boldsymbol{\sigma})\right)}{Z},$$
where
$$Z = \sum_{\boldsymbol{\sigma}} \exp\left(-\beta E(\boldsymbol{\sigma})\right)$$
is called the *partition function.*

Free energy

- The entropy of a probability distribution $P(\boldsymbol{\sigma})$ is
$$S(P) = -\sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma}) \log\left(P(\boldsymbol{\sigma})\right).$$
It is maximal for a flat distribution, and minimal for a delta (maximum information).
- The average energy is $U = \sum_{\boldsymbol{\sigma}} E(\boldsymbol{\sigma})P(\boldsymbol{\sigma})$.
- It can be shown that $P(\boldsymbol{\sigma})$ is the distribution that maximizes $S(P)$ with the constraint of fixed U. The parameter β is the Lagrange multiplier.
- The free energy $U - TS$ ($T = 1/\beta$) is the Lagrange function to be freely minimized. It is also given by $F = -T\log(Z)$.

Free energy landscapes

- The evolution lead to minima of free energy. There is a competition between order (Energy) and disorder (entropy).
- A phase transition occurs when there is a bifurcation of minima, for instance by changing the temperature.



Simulated annealing

- One can visualize the energy landscape "blurred" by the temperature, so that evolution can "jump out" local minima.
- This suggest a powerful technique for minimization of functions of many variables: identify the function to be minimized with energy, eventually adding "soft" constraints multiplied by large weight. The dynamics of the system is given by *Monte Carlo* flips (or changes) at a given temperature.
- Slowly lower the temperature, trying to "concentrate" the probability distribution around the global minimum.
- Simulated annealing may be used for storing patterns in any network. In this case the cost function (energy) is the difference between the final state and the desired one.

Hebb rule for memorizing patterns

- Full connectivity
- $\{\boldsymbol{\xi}^\mu\}$ is set of memory pattern. We try to make them correspond to energy minima.
- Hebb rule:
$$J_{ij} = \frac{1}{N}\sum_{\mu=1}^{M} \xi_i^\mu \xi_j^\mu \qquad J_{ii} = 0.$$
- For **small** M, the ξ patterns are absolute minima, together with their opposite.
- For **uncorrelated** patterns, each of them has a large attraction basin.

Phase transitions

- For low temperature T, the final state is near to one of the memorized pattern (or the opposite). The order parameter is the *overlap* with respect to stored patterns
$$\omega(\mu) = \frac{1}{N}\sum_i \sigma_i \xi_i^\mu.$$
- By increasing the temperature the average overlap lowers, at a critical temperature T_c it becomes zero with respect to all patterns.

Capacity

- By increasing the number of stored patterns, spurious minima start to appear. They are given by *mixtures*of stored patterns. Their basins are larger the more correlated two patterns are.
- The energy level of spurious minima is higher than that of "true" minima.
- By increasing the temperature (below T_c), one can "remove" spurious minima.
- Simulated annealing may recover stored patterns.
- However, the difference in energy among true and spurious minima lowers with increasing number of stored patterns. The maximum number of uncorrelated patterns is about $0.14N$.

Increasing capacity

- Decorrelation among patterns increases capacity.
- Kohonen proposes the rule

  $$J_{ij} = \frac{1}{N} \sum_{\mu\nu} \xi_i^\mu \xi_j^\nu A_{\mu\nu}^{-1},$$

  where

  $$A_{\mu\nu} = \frac{1}{N} \sum_i \xi_i^\mu \xi_i^\nu$$

  is the correlation among patterns.
- The number of memorizable patterns increases to N.

Networks

- We have only used regular networks. However, in biology and social sciences we find many non-regular networks.
- A network is defined by an *adjacency matrix*, that says what node is connected to what node
- The sum along rows of the adjacency matrix gives the *in-degree* of a node (number of incoming links)
- The sum along columns of the adjacency matrix gives the *out-degree* of a node (number of outcoming links)
- Regular lattices correspond to *circulant* matrices, whose eigenvectors are periodic functions (this is why Fourier analysis works well for lattices).
- Mean field corresponds to *annealing* (always changing) networks with no structure (random graphs).
- By adding a small number of long-range connections, regular lattices behave often as mean-field ones (small-world effect).

Scale-free networks

- Random graphs have a Poissonian distribution of degrees (regular ones have fixed degree).
- Social nets often exhibit a power-law distribution of degrees.
- It is rather easy to show that such a structure arises from a *dynamical growth process* of the network (e.g, preferential attachment).
- Another characteristic is assortativity: the probability that a high-degree node (a hub) is attached to another hub.
- The mean-field approximation for scale-free networks is obtained by considering separately the probability distributions of nodes with different degree.
- Dynamic and stochastic processes often depend crucially on the structure of the network, and may also change it.