Machine Learning, Lecture 8: Competitive Learning

S. Nõmm

[1]Department of Computer Science, Tallinn University of Technology

March 24, 2016

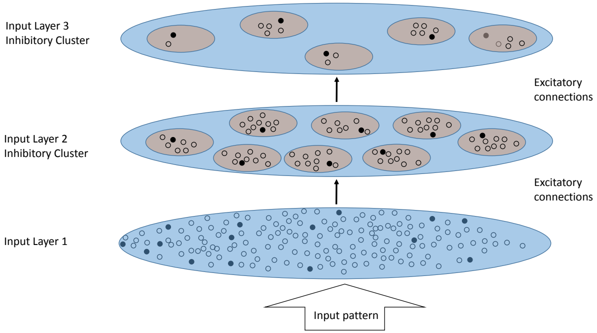
What is *Competitive Learning?*

- Usually considered and implemented with the NN framework.
- Unsupervised learning
- Corrections to the network weights are **not** performed by an external agent.
- The network itself decides what output is best for a given input.
- During the training process, output of the unit or neuron which provides the highest activation to a given input pattern is declared a winner.
- Only winning neuron (some times closest associates) learn.
  - Hard learning (The winner takes it all) - weight of only the winning neuron is updated.
  - Soft learning - weight of the winning neuron and its close associated is updated.
- There is a number of different approaches to the problem: von der Malsburg(1973), Fukushima(1975), Grossberg(1976).

Grossberg(1976) version has the following propoerties:

- Neurons in each layer are split into several non-overlapping clusters.
- Each neuron within the cluster inhibits other neurons of the cluster.
- Within the cluster, the neuron receiving the largest input achieves its maximum value (1) and all the others set to minimum (0)
- Every neuron in each cluster connected (receives signals) from all the input neurons (from all the inputs).
- Neuron learns if an only if it wins the competition within its cluster.
- A stimulus pattern is binary.
- Each neuron has fixed amount of weigh which is distributed among the inputs $\sum \omega_{i,j} = 1$.
- A neuron learns by shifting weight from inactive to active inputs.

The learning rule



The learning rule in such case is as follows:

$$\Delta\omega_{i,j} = \begin{cases} 0 & \text{if } i \text{ loses on stimulus } k \\ \epsilon\dfrac{\delta_{j,k}}{\sigma_k} - \epsilon\omega_{i,j} & \end{cases}$$

where $\delta_{j,k}$ takes the value 1 if stimulus S_k neuron j is active and 0 otherwise, σ_k number of active neurons in stimulus S_k.

Hamming Net + MaxNet

Another approach is to consider competitive network to be composed of two neural networks:

- Hamming net: measures how much the input vector resembles the weight vector of each perceptron.
- Max net: Finds the perception with the maximum value.
- Competitive learning rule:

$$\omega_i(n+1) = \omega_i(n) + \epsilon(x(n) - \omega_i(n))$$

where i - neuron index, n-instance index, $\omega - i$ x - input, ϵ -learning rate.

Self - organizing maps (Kohonen - maps)

Self organizing map (SOM)

- Proposed by Kohonen in 1982. Key feature is topology preservation.
- Competitive learning model with a neighbourhood constrain on the output neurons.
- Usually 1D or 2D grid to restrict dimensionality and provide better visualization.
- Have a strong neurobiological basis.
- Kohonen SOM is the result from three processes Competition, cooperation, adaptation.

Competition

- Each neuron in a SOM is assigned a weight vector with the same dimensionality N as the input space.
- Any given input pattern is compared to the weight vector of each neuron and the closest neuron is declared the winner.
- The Euclidean norm is commonly used as the distance measure.

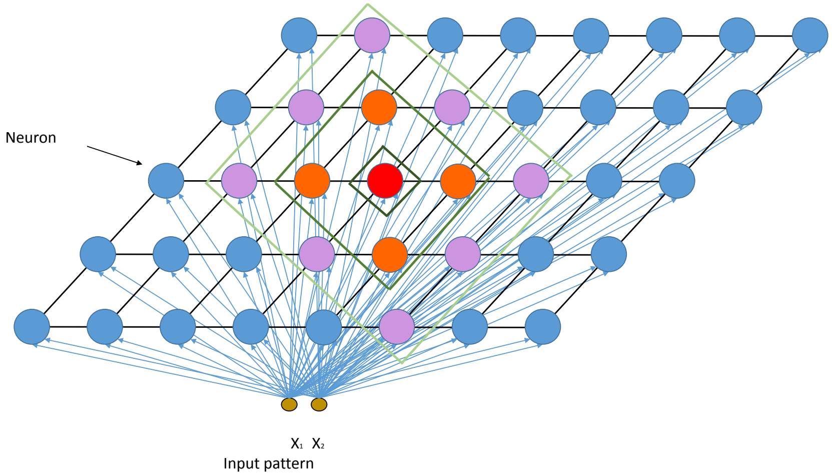
Cooperation

- In order to allow topologically close neurons to become sensitive to similar patterns the activation of the winning neuron is spread to neurons in its immediate neighborhood.
- Neighbourhood of the wining neuron is determined on the basis of topology with city-block (Manhattan or taxi driver) distance.
- The size of the neighbourhood is changes from large to small. Large neighbourhood in the beginning allows to preserve the topology. Smaller neighbourhood on the later stages allows to specialization of the neurons.

ANN

- During the training, winning neuron and its topological neighbors are adapted to make their weight vectors more similar to the input pattern that caused the activation.
- Neurons that are closer to the winner will adapt more than neurons that are further away.
- The magnitude of the adaptation is controlled with a learning rate, which decays over time to ensure convergence of the SOM.

Kohonen SOM



Kohonen SOM algorithm

- A learning rate decay rule

$$\epsilon(t) = \epsilon - 0e^{-t/\tau_1}$$

- A neighborhood kernel function

$$h_{ik}(t) = e^{-\frac{d_{ik}^2}{2\sigma^2(t)}}$$

  where d_{ik} distance in the grid between ω_i and ω_k.

- A neighborhood size decay rule

$$\sigma(t) = \sigma_0 e^{\frac{-t}{\tau_2}}$$

Kohonen SOM algorithm

- Initialize weights
- Repeat until converge
    - Select following pattern from the input sequence
    - Find ω_i that best matches the input pattern ξ^n

$$i = \mathrm{argmin}_j \|\xi^n - \omega_j\|$$

    - Update the weights of the winner ω_k

$$\omega_k = \omega_k + \epsilon(t) h_{ik}(t)(\xi^n - \omega k)$$

    - decrease learning rate $\epsilon(t)$
    - Decrease neighbourhood size $\sigma(t)$

Example

To be continued in the computer class