# Neural Networks
# - Homework 9 -

Petr Lukin, Evgeniya Ovchinnikova

Lecture date: 21 November 2016

## 1 Mind map

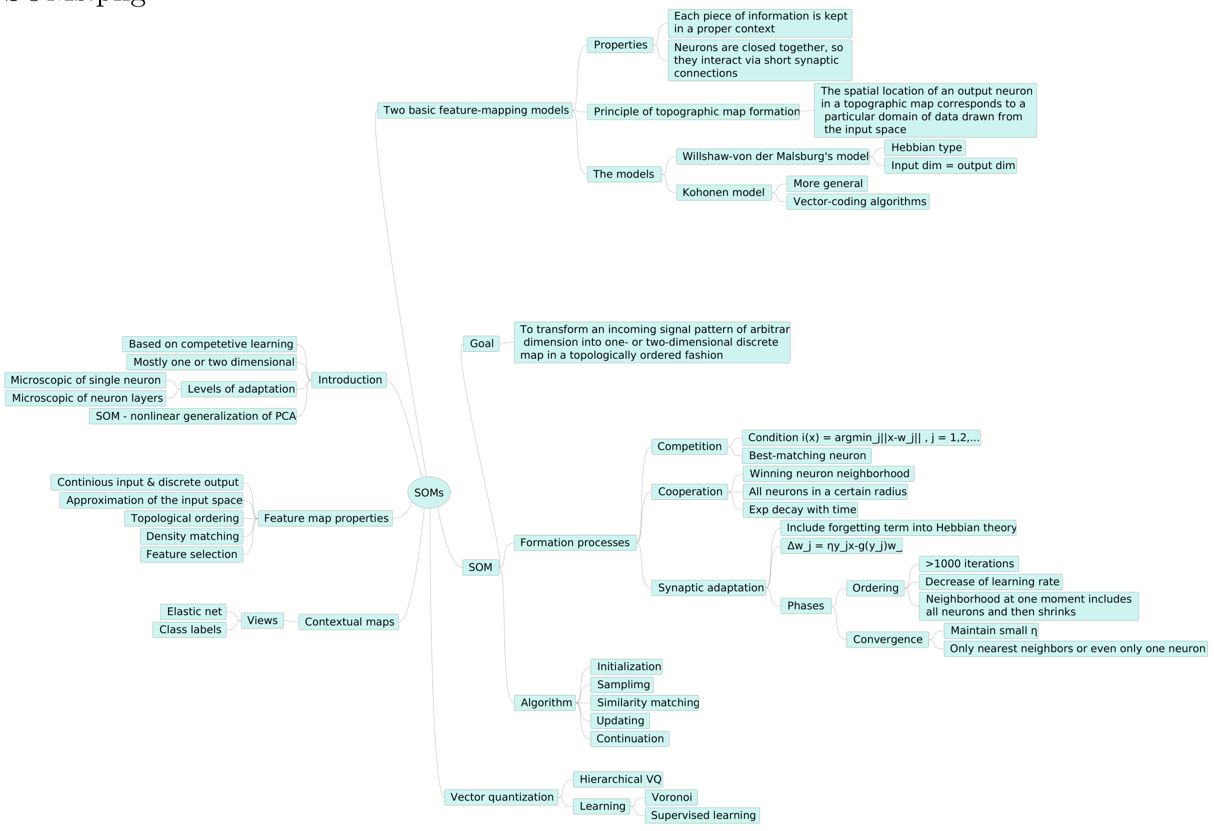
SOMs
Two basic feature-mapping models
Properties
Each piece of information is kept in a proper context
Neurons are closed together, so they interact via short synaptic connections
Principle of topographic map formation
The spatial location of an output neuron in a topographic map corresponds to a particular domain of data drawn from the input space
The models
Willshaw-von der Malsburg's model
Hebbian type
Input dim = output dim
Kohonen model
More general
Vector-coding algorithms
Introduction
Based on competetive learning
Mostly one or two dimensional
Levels of adaptation
Microscopic of single neuron
Microscopic of neuron layers
SOM - nonlinear generalization of PCA
Feature map properties
Continious input & discrete output
Approximation of the input space
Topological ordering
Density matching
Feature selection
Contextual maps
Views
Elastic net
Class labels
Goal
To transform an incoming signal pattern of arbitrar dimension into one- or two-dimensional discrete map in a topologically ordered fashion
SOM
Formation processes
Competition
Condition i(x) = argmin_j||x-w_j|| , j = 1,2,...
Best-matching neuron
Cooperation
Winning neuron neighborhood
All neurons in a certain radius
Exp decay with time
Synaptic adaptation
Include forgetting term into Hebbian theory
Δw_j = ηy_jx-g(y_j)w_
Phases
Ordering
>1000 iterations
Decrease of learning rate
Neighborhood at one moment includes all neurons and then shrinks
Convergence
Maintain small η
Only nearest neighbors or even only one neuron
Algorithm
Initialization
Samplimg
Similarity matching
Updating
Continuation
Vector quantization
Hierarchical VQ
Learning
Voronoi
Supervised learning

Figure 1: Mind map. Chapter 9 from Haykin's book. A zoomed version is attached as SOMs.png

## 2 Exercises

### 2.1 Exercise 2

Show that in the SOM algorithm the winner neuron for an input $x$ is the neuron $k$ whose weight vector $w_k$ maximizes the inner product $< wk, x >$ of x and w k , with $x$ and $w_k$ normalized.

## 2.2 Exercise 3

Consider the one dimensional input space S=0.1, 0.2, 0.4, 0.5. Cluster S using a one dimensional SOM network with:

- 2 nodes.
- learning rate equal to 0.1.
- Neighborhood function which is equal to 1 for the winner neuron and is 0 otherwise.
- Weight initialization:
  - $w_1$=0.15, $w_2$=0.45
  - $w_1$=0.3, $w_2$=0.9
- Stopping criterion:

$$\sum_{i=1}^{2} |w_i^{old} - w_i^{new}| < 0.01$$

Comment on the two clusterings you obtained using the two different weight initializations.

## 2.3 Exercise 4

Program a SOM to solve the traveling salesman problem (TSP):

- Input layer contains just two neurons called "Xcoord" and "Ycoord"
- The Kohonen lattice is thought of as a circle containing at least as many neurons as we have cities (usage of more neurons is possible), the neurons are numbered in round about fashion.
- Each neuron n(i) (numbered by i) on the circle is connected to "Xcoord" and "Ycoord" and the weights on these edges are the initial x and y coordinates of the neuron in the plane (see picture)
- The input patterns are the (x,y) coordinates of all target cities.
- When applying one concrete city pattern the winner is of course the neuron with lies closest to the given city. Only the weights of the winning neuron will be adapted according to the learning rule, no other neighboring weights are changed. This "moves" the neuron closer to "its" city.
- After some cycles there is just ONE neuron closest to each city. These build pairs (n(i),City(closest to n(i)).
- If we sort this pairs according to the number of the neurons "i" this will give a journey for the respective cities, which solves the TSP approximately.
- Use tools like ICONNECT, TSPLIB or alike to do the programming, compare runtime, memory and achieved length of path.