Self organizing maps

A visualization technique with data dimension reduction

Juan López González

University of Oviedo

Inverted CERN School of Computing, 24-25 February 2014

General overview

- **Lecture 1**
  - Machine learning
    - Introduction
    - Definition
    - Problems
    - Techniques
- **Lecture 2**
  - ANN introduction
  - SOM
  - Simulation
  - SOM based models

LECTURE 1

Self organizing maps.

A visualization technique with data dimension reduction.

1. Artificial neural networks (ANN)

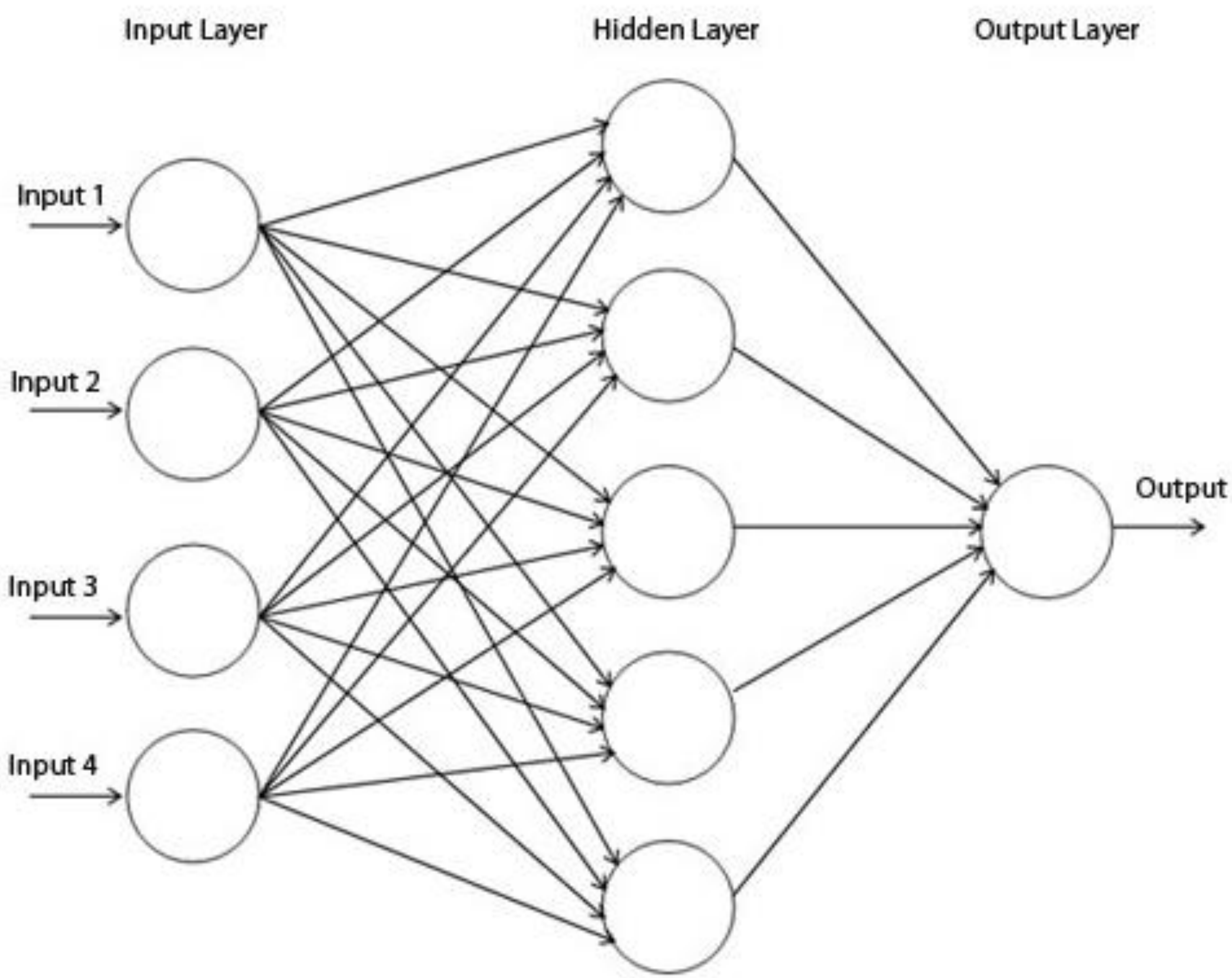
1.1. Introduction

1.2. Types

1.2.1. Feedforward NN
1.2.2. Recurrent NN
1.2.3. Self organizing NN
1.2.4. Others

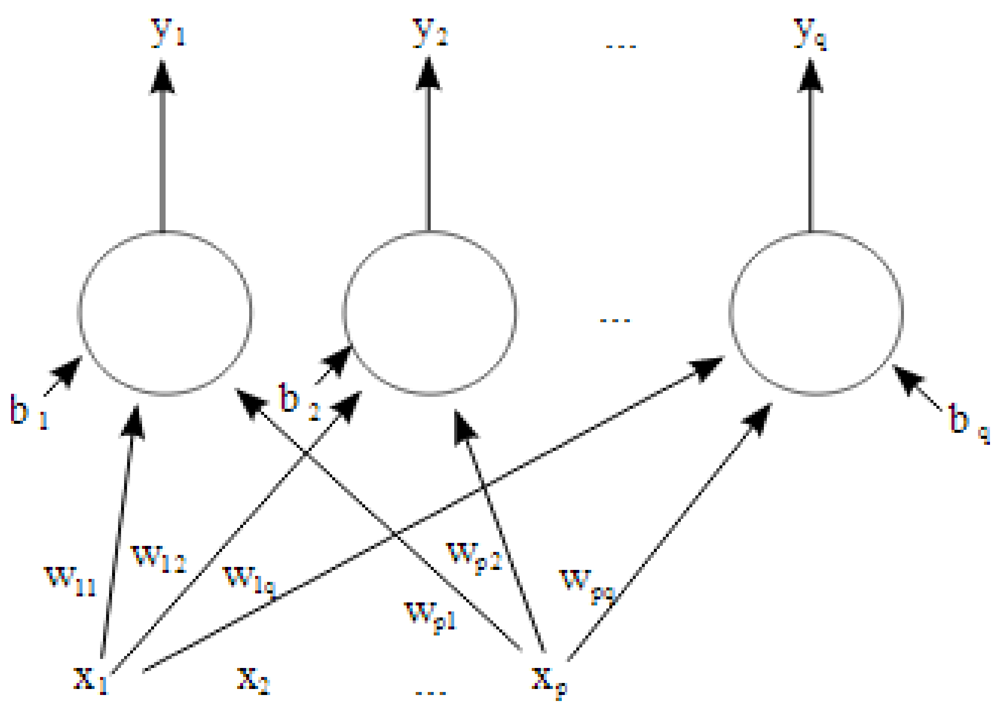
CERN School of Computing

1.2.1. Feedforward NN



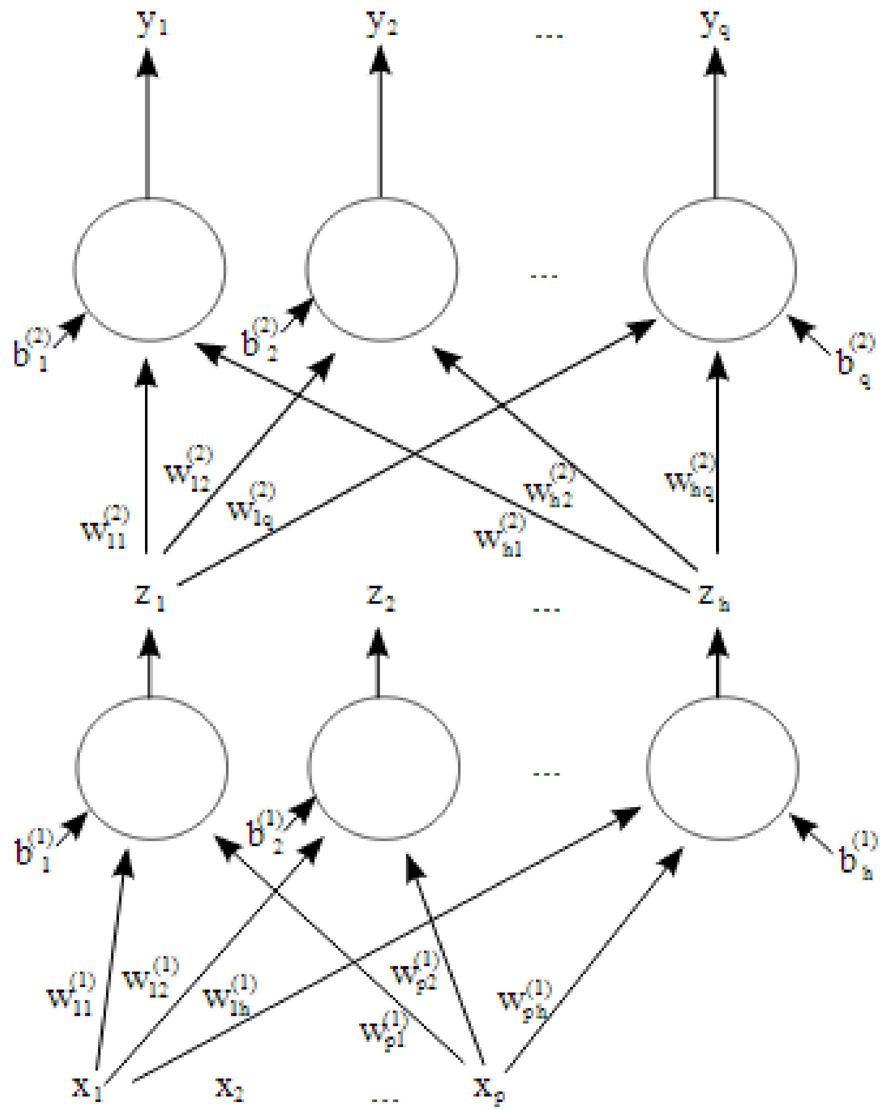
1.2.1. Feedforward NN

- **Single layer feedforward**

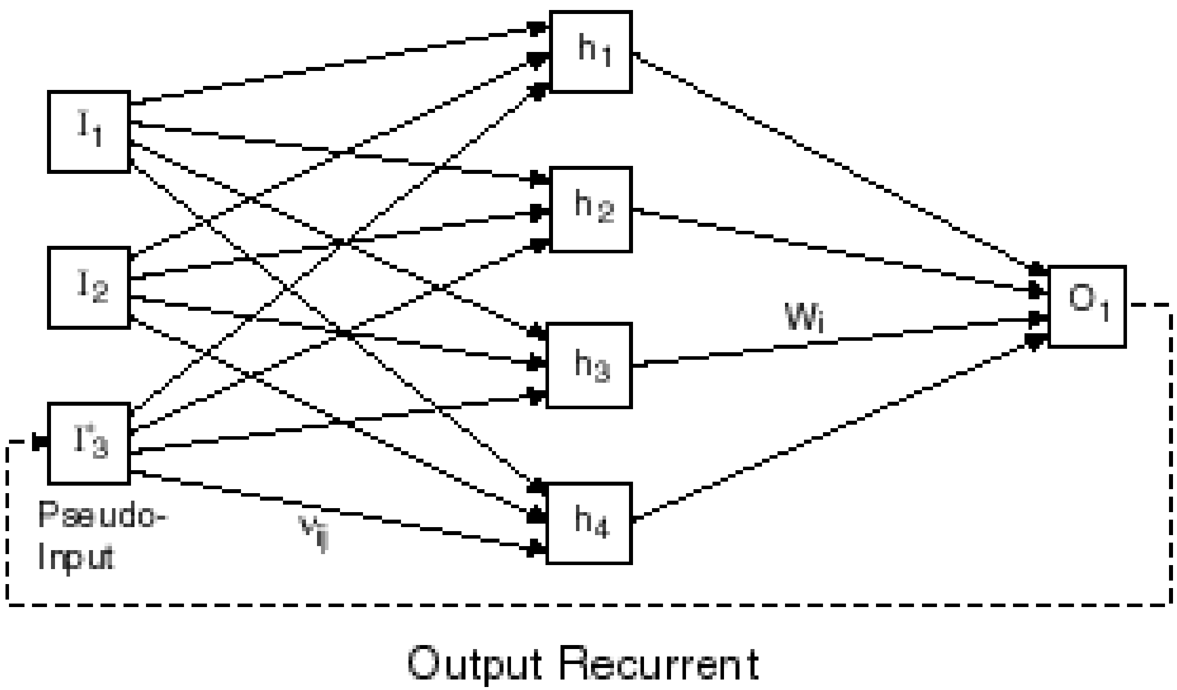


1.2.1. Feedforward NN

- **Multi-layer feedforward**
  - Supervised learning
  - Backpropagation learning algorithm

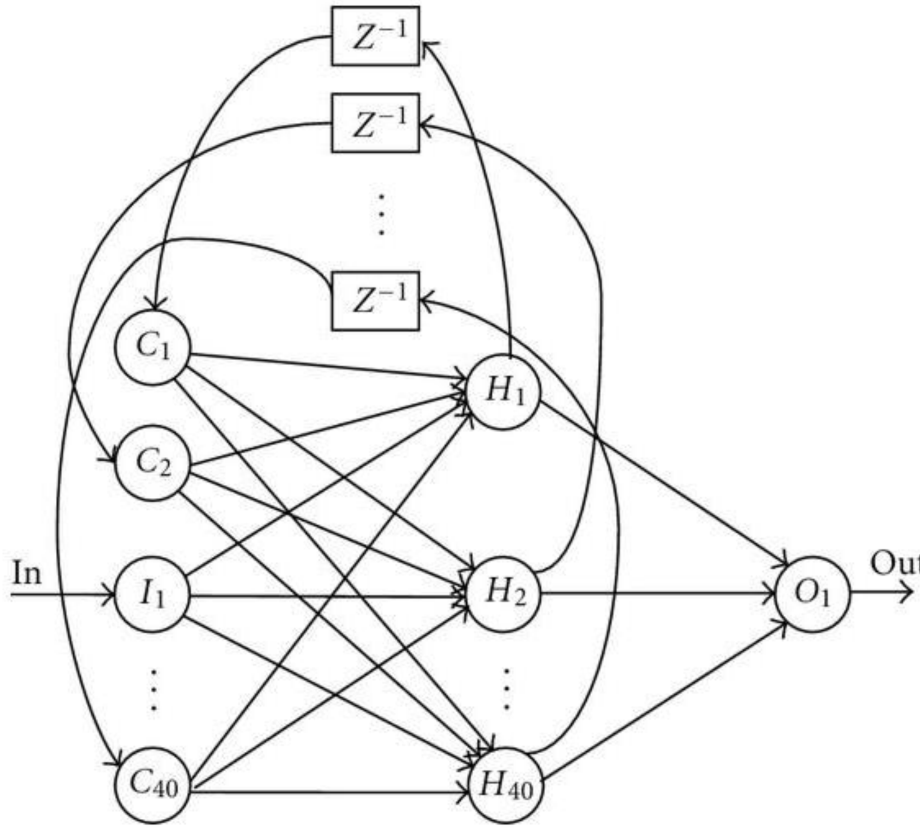


1.2.2. Recurrent neural networks



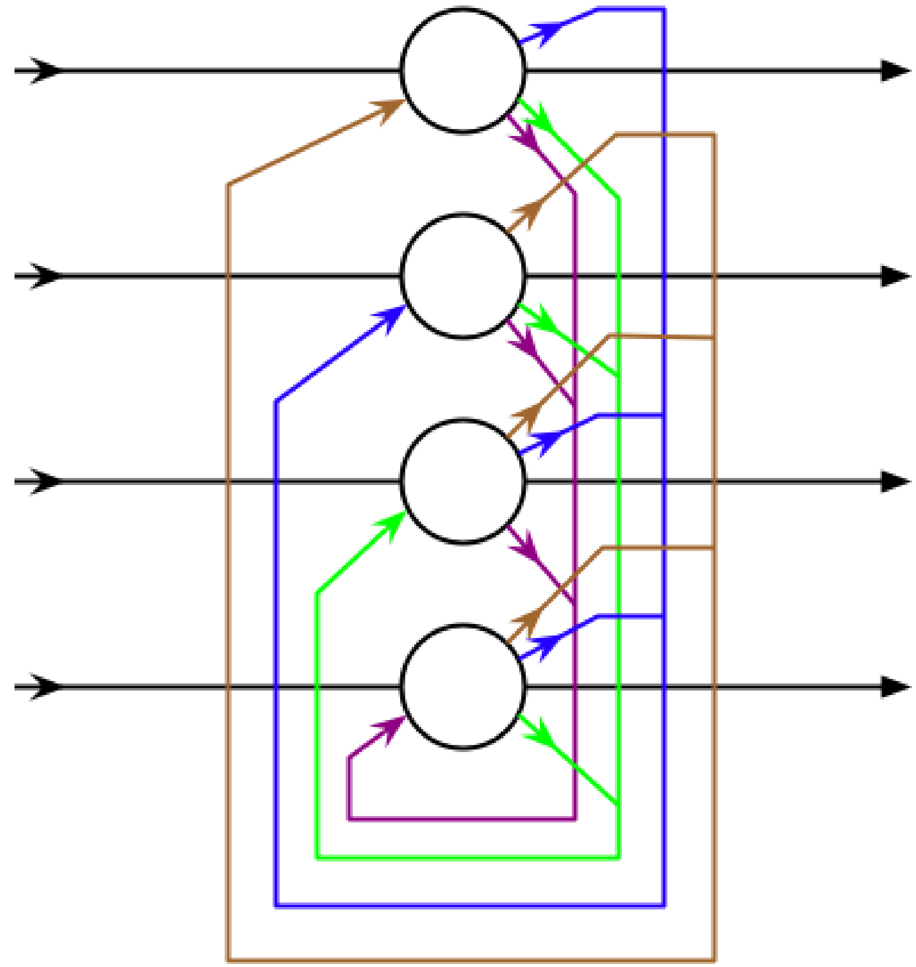
1.2.2. Recurrent neural networks

- **Elman networks**
  - ‘Context units’
  - Maintain state



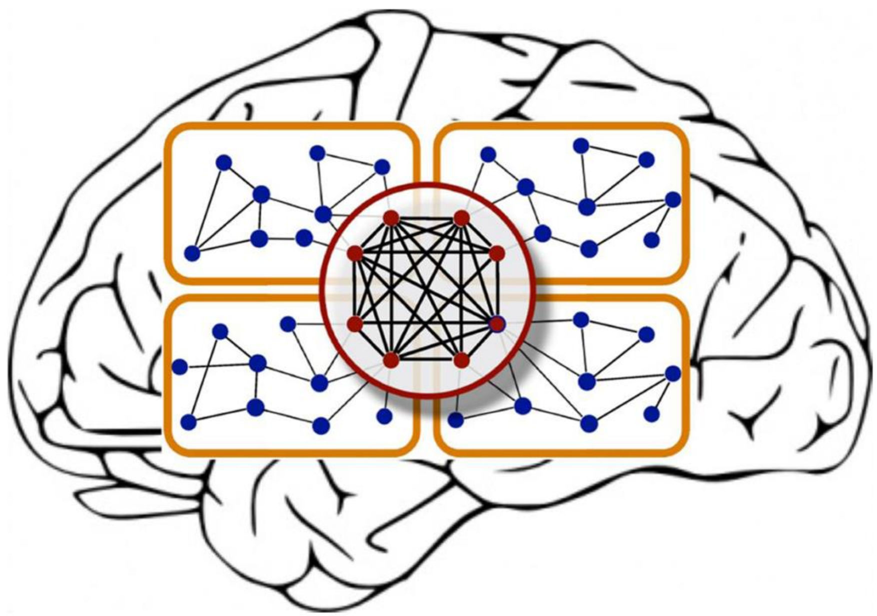
1.2.2. Recurrent neural networks

- **Hopefield network**
  - Symmetric connections
  - Associative memory



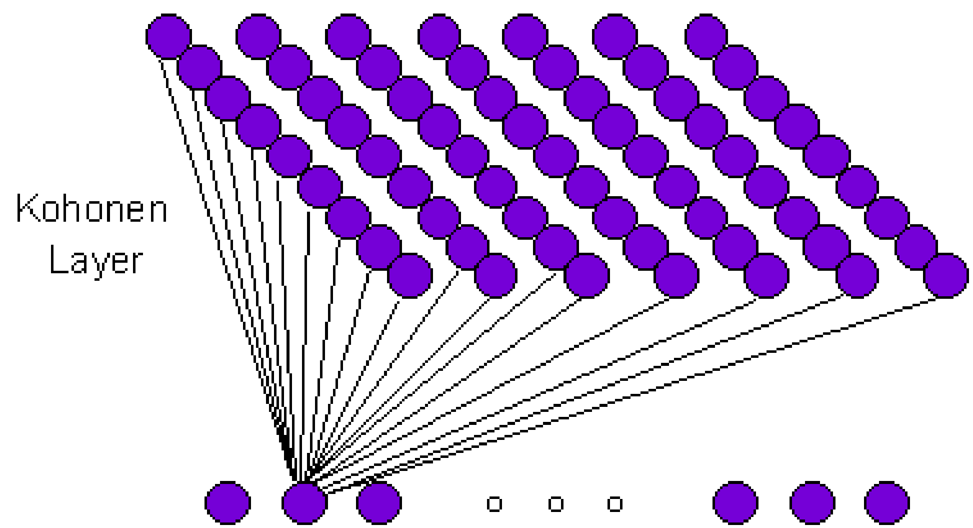
1.2.2. Recurrent neural networks

- **Modular neural networks**
  - The human brain is not a massive network but a collection of small networks



1.2.3. Self-organizing networks

- **Self-organizing networks**
  - A set of neurons learn to map points in an input space to coordinates in an output space



CERN School of Computing

1.2.4. Others

- Holographic associative memory
- Instantaneously trained networks
- Learning vector quantization
- Neuro-fuzzy networks
- ...

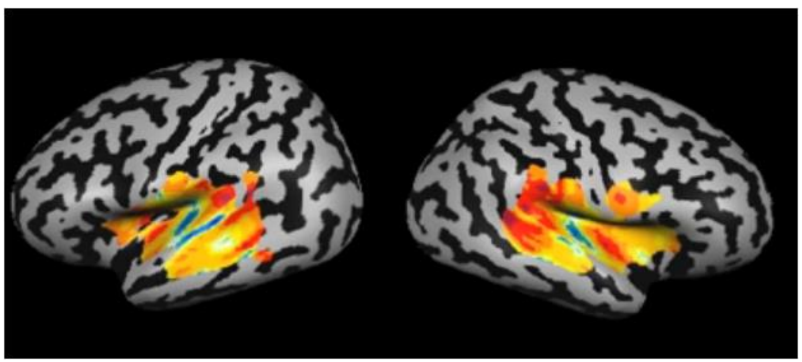
2. Self-organizing maps

2.1. Motivation

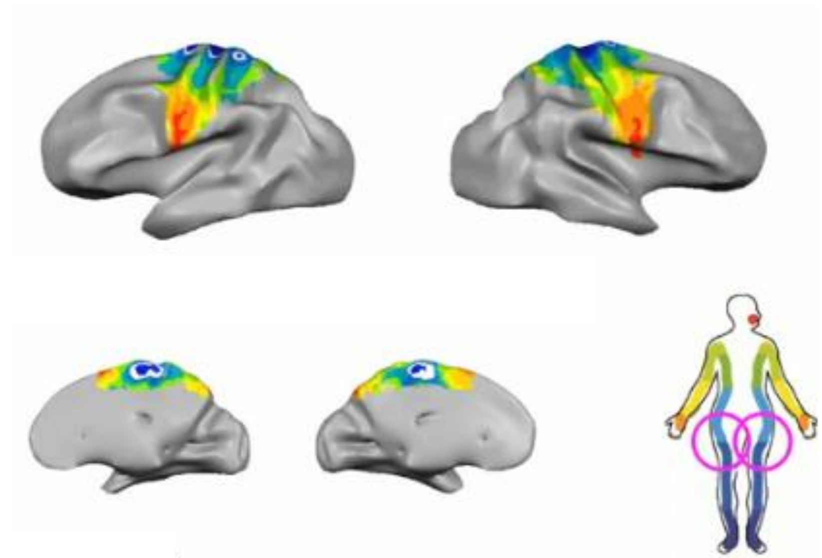
- **Topographic maps**
  - Different sensory inputs (motor, visual, auditory…) are mapped in areas of the cerebral cortex in an orderly fashion

2.1. Motivation

- “The spatial location of an output neuron in a topographic map corresponds to a particular domain or feature drawn from the input space”



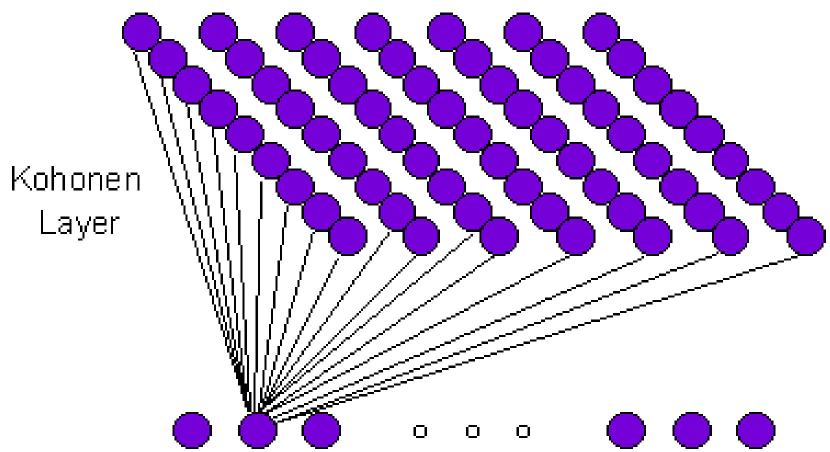
Auditory cortical fields

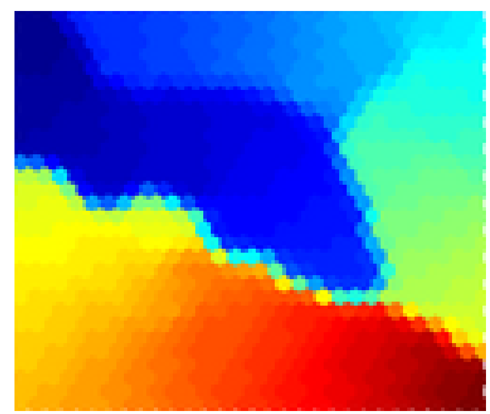


Motor-somatotopic maps

2.2. Goal

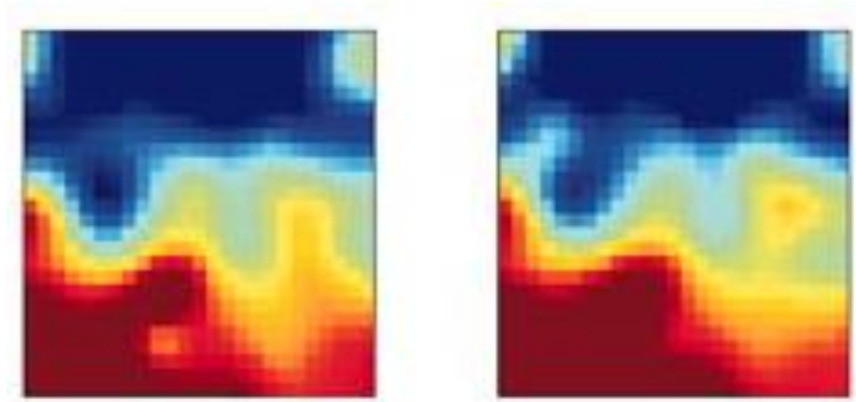
- Transform incoming signal of arbitrary dimension into a 1-2-3 dimensional discrete map in a topologically ordered fashion





2.3. Main properties

- Transform continuos input space to discrete output space
  - **Dimension reduction**
    - winner-takes-all neuron
- Ordered feature map



Input with similar characteristics produce similar ouput

2.3.1 Dimension reduction

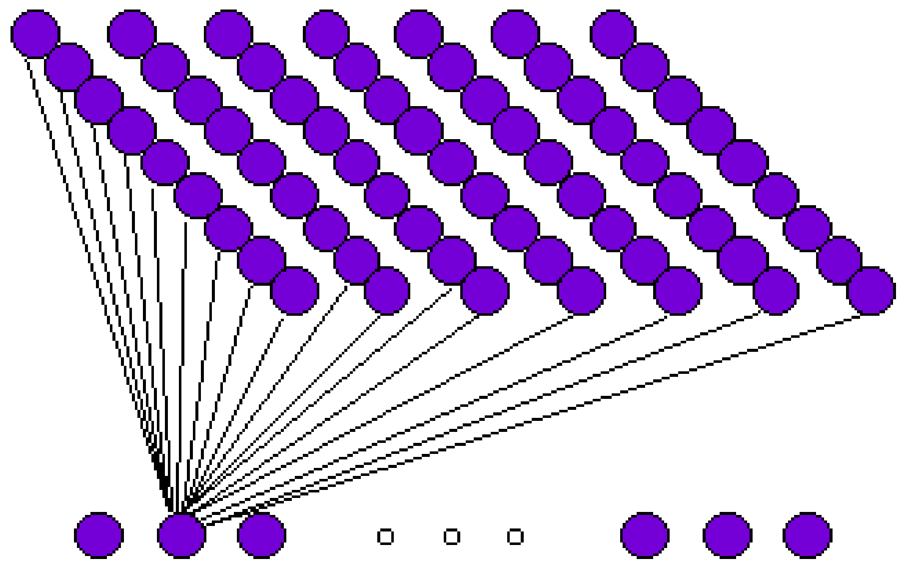
- **Curse of dimensionality** (Richard E. Bellman)
  - The amount of data needed grows exponentially with the dimensionality
- Types
  - **Feature extraction**
    - Reduce input data (features vector)
  - **Feature selection**
    - Select subset (remove redundant and irrelevant data)

2.4. Elements

- **…of machine learning**
  - A pattern exists
  - We don't know how to solve it mathematically
  - A lot of data
    - $(a_1,b_1,..,n_1), (a_2,b_2,..,n_2) \ldots (a_N,b_N,..,n_N)$

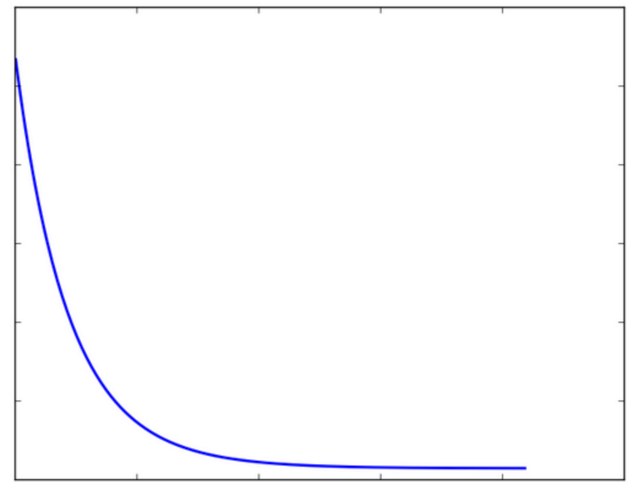
2.4. Elements

- **Lattice of neurons**
  - Size?
  - Weights

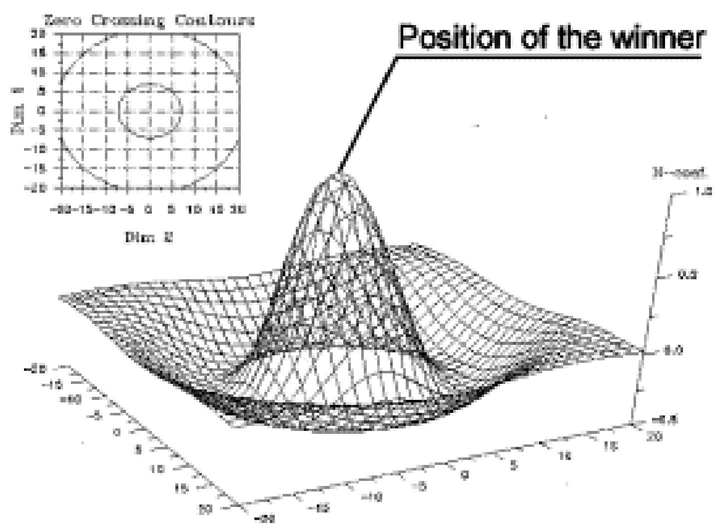


2.4. Elements

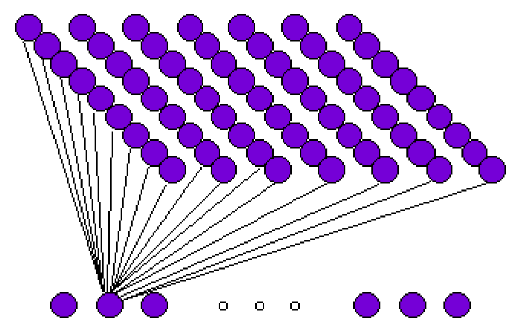
- Learning rate
- Neighborhood function



Learning rate function

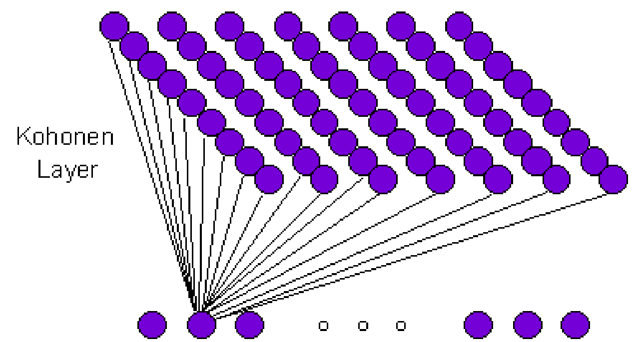


Neighbourhood function or Mexican Hat function



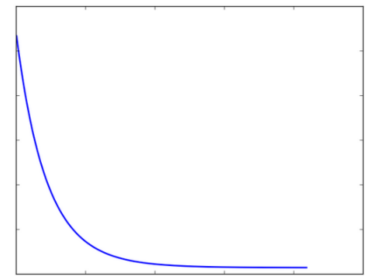
2.5. Algorithm

- Initialization
  - Input data preprocessing
    - Normalizing
    - Discrete-continuous variables?
  - Weight initialization
    - Random weights

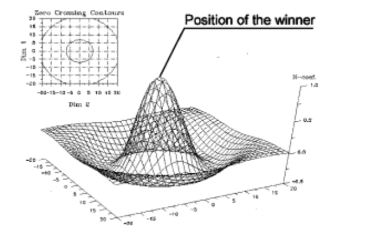


2.5. Algorithm (2)

- Sampling
  - Take sample from input space
- Matching
  - Find BMU: i.e. min of $\sum_{i=1}^{D}(x_i - w_{ji})^2$
- Update weights
  - i.e. $\Delta w_{ji} = \eta(t)\, T_{j,I(\mathbf{x})}(t)\, (x_i - w_{ji})$

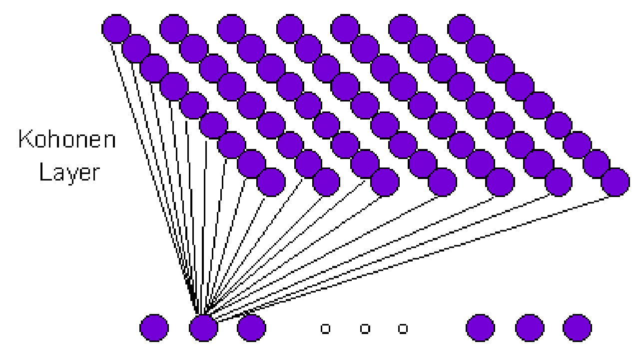


$\eta(t)$



Neighbourhood function or Mexican Hat function

$T_{j,I(\mathbf{x})}(t)$

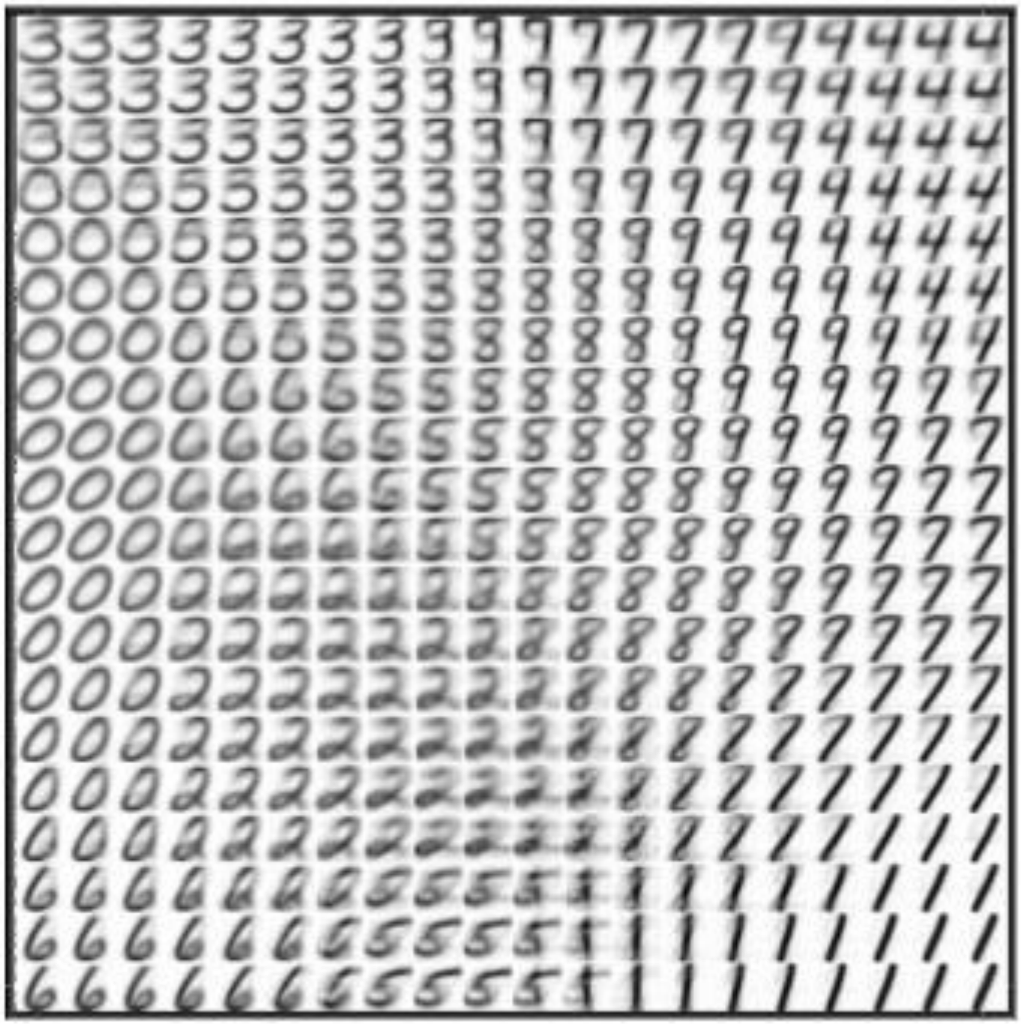


3. Practical exercise

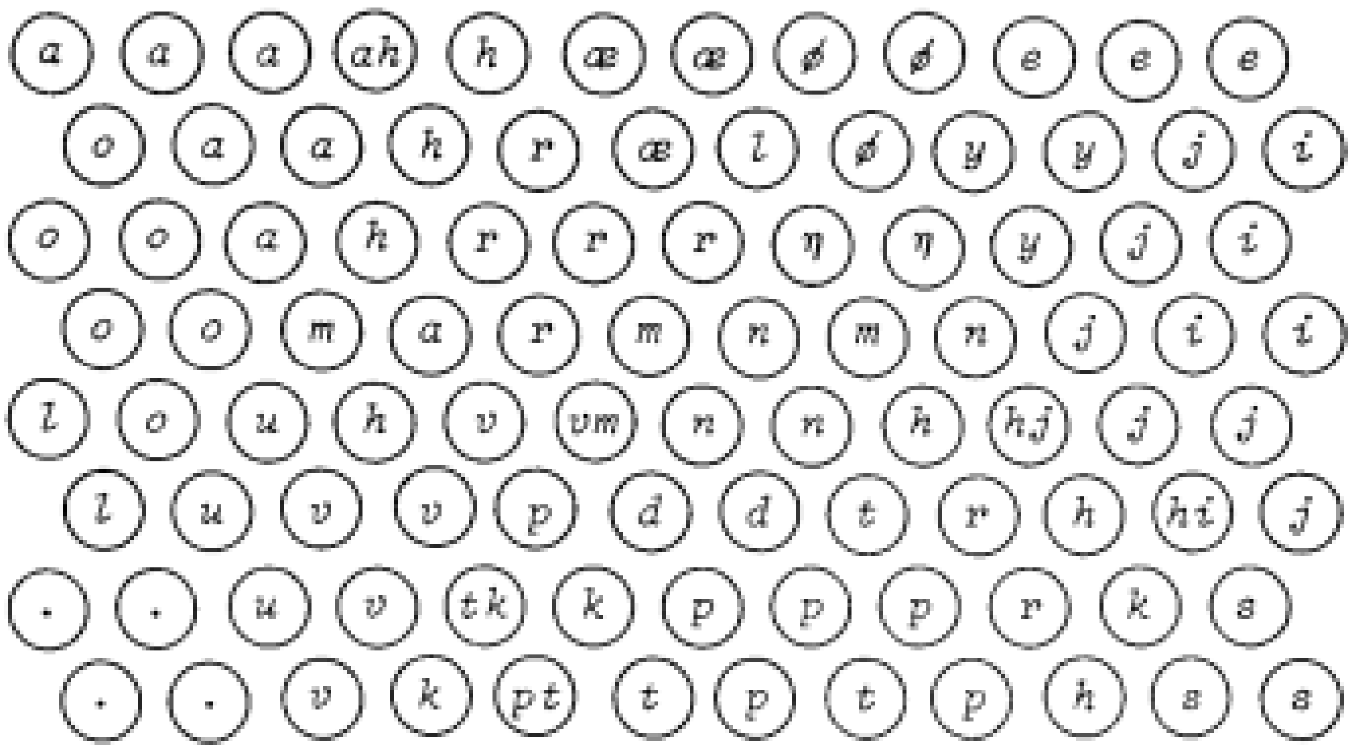
4. Map examples

4.1. Digit recognition
4.2. Finish phonetics
4.3. Semantic map of word context

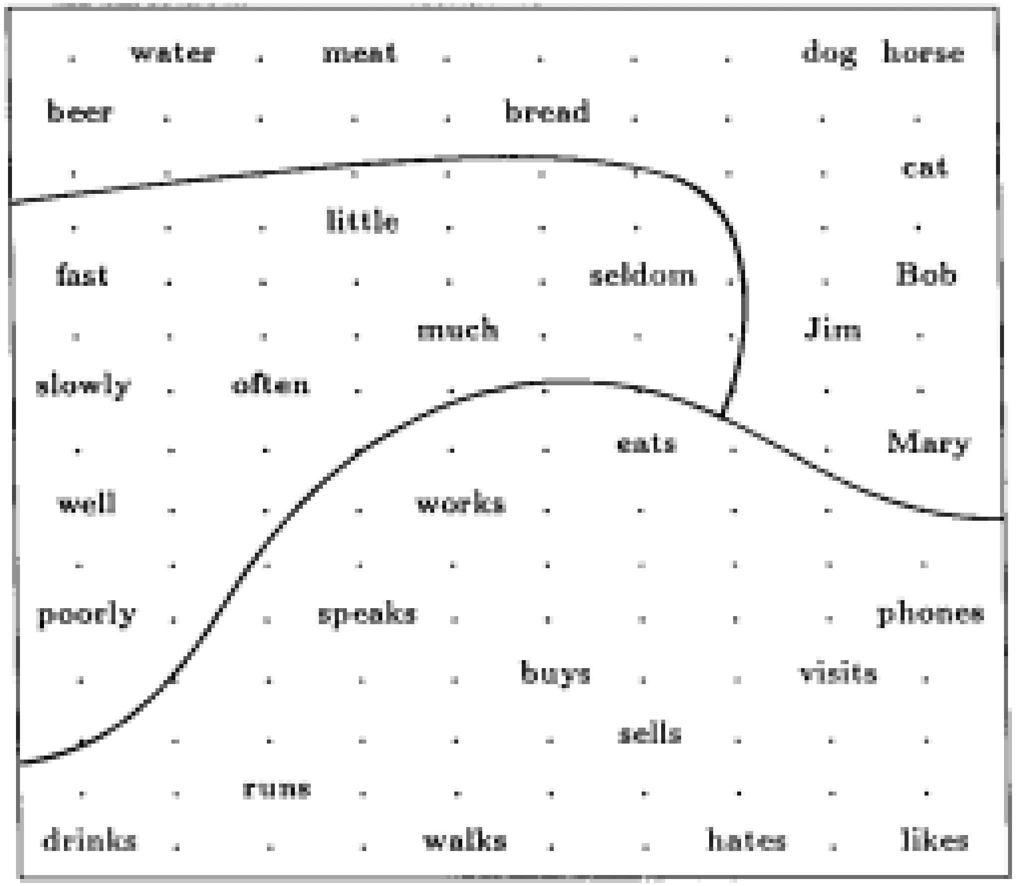
4.1. Digit recognition



4.2. Finnish phonetics



4.3. Semantic map of word context



5. Other SOM based models

5.1. TASOM
5.2. GSOM
5.3. MuSOM

5.1. TASOM

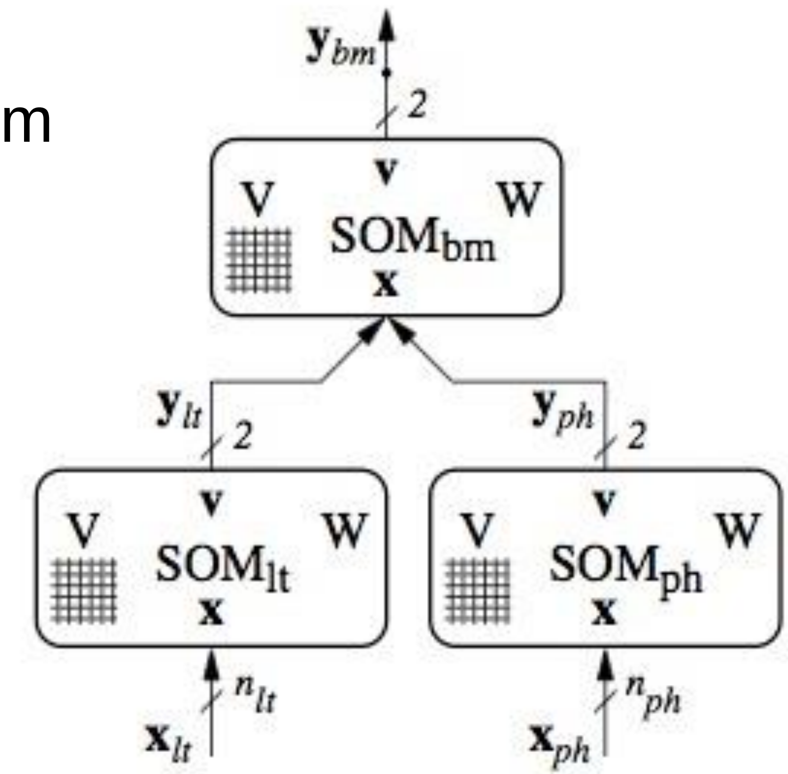
- **Time adaptative self-organizing maps**
  - Deals with non-stationary input distributions
  - Adaptative learning rates: $n(w,x)$
  - Adaptative neighborhood rates: $T(w,x)$

5.2. GSOM

- **Growing self-organizing maps**
  - Deals with identifying sizes for SOMs
    - Spread factor
    - New nodes in boundaries
  - Good when unknown clusters

5.3. MuSOM

- **Multimodal SOM**
  - High level classification from sensory integration



Q & A