Reinforcement Learning 12 March 2007
Lecture 19

SOFMs: SELF-ORGANISING FEATURE MAPS (KOHONEN)

- Topographic maps
- Network architecture
- The SOFM algorithm (on-line)
- The SOFM algorithm (batch)
- Applications
- Properties of the SOFM
- Use in reinforcement learning

TOPOGRAPHIC MAPS

•Topographic = topology preserving – neighbouring places in the world are found in neighbouring places in the map

•Topographic maps are found in biological systems, e.g. the retinotopic map from the retina to visual cortex, the somatosensory map and tonotopic map

•In the visual cortex, adjacent neruons have adjacent visual receptive fields, and collectively they constitute a map of the retina

•SOFM (or SOM) developed by Kohonen since 1982

•Builds on ideas of Willshaw and von der Malsburg about how retinotopic maps are wired up

CLUSTERING

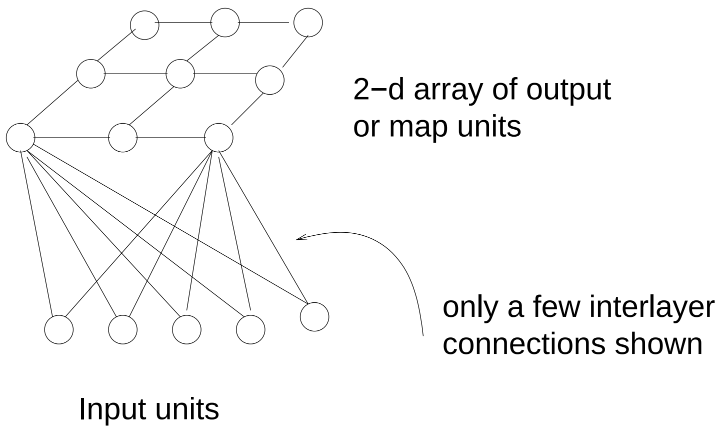
•Self -organising map similar to a clustering algorithm, except that there is an additional constraint

•Cluster centres are embedded in another space (the output space) and points that are nearby in the input space must map to points that are nearby in the output space

•So when we update a cluster centre, we also update its neighbours.

•Has the effect of keeping close-by units in the ouput space mapping to close-by regions of input space

KOHONEN ARCHITECTURE



•Input has dimension d, i.e. d units

•Array – usually 1D or 2D – of grid/map/output units on a rectangular or hexagonal grid

•Each input unit is connected to each grid unit

•Neighbourhood relations calculated on this grid

•An example input could be $(x, y, \dot{x}, \dot{y}, \theta)$ for position, velocity and orientation of a robot

•Each grid unit j has a vector $\mathbf{w}_j$ associated with it, of the same dimension as the input

THE SOM ALGORITHM

Initialise a grid of units to have weight vectors $\mathbf{w}_j$ set to random values

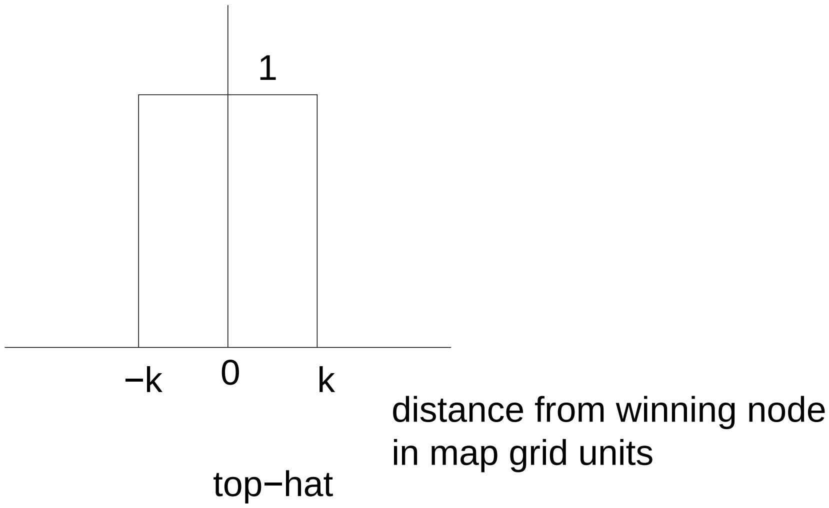
Loop until weights change by only tiny amounts

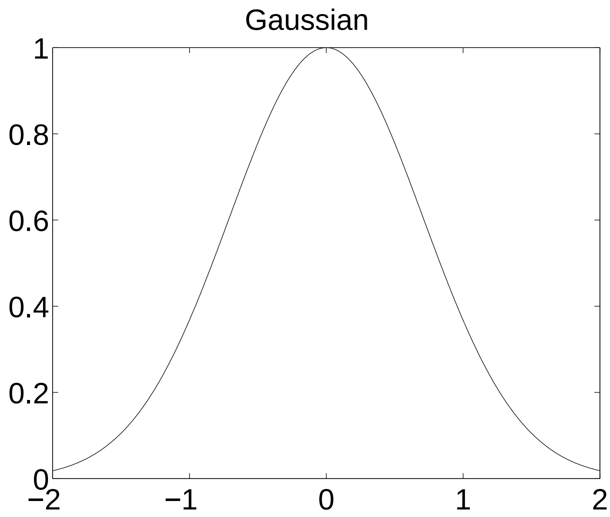
Take a sample input $\mathbf{x}$
Find the winning map node i^* that best matches the input:
$i^* = \arg\min_j \|\mathbf{x} - \mathbf{w}_j\|$

Update the winning weight vector and the weights of those nodes in its neighbourhood:
$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t) N_t(j, i^*)(\mathbf{x} - \mathbf{w}_j(t))$

The learning rate $\eta(t)$ needs to decrease during the learning, as does the width of the neighbourhood function $N_t(j, i^*)$. We start with N having a wide range and narrow it down, and η starts large and is successively reduced to zero.

NEIGHBOURHOOD FUNCTIONS





SOM BATCH ALGORITHM

Luttrell (1990), Kohonen (1993)

Initialise the grid of units to have weight vectors $\mathbf{w}_j$ set to random values

Loop until terminated

- for $k = 1$ to K (number of data vectors)

  find the best matching (winning) unit

  $$i^*(k) = \arg\min_j \|\mathbf{x}_k - \mathbf{w}_j\|$$

  end for

- Update the weight vectors using

  $$\mathbf{w}_j = \frac{\sum_{k=1}^{K} \mathbf{x}_k N(i^*(k),j)}{\sum_{k=1}^{K} N(i^*(k),j)}$$

End loop

PRACTICAL ISSUES

•Grid:
  –dimension?
  –size?
  –topology?

•Typical training regimes:
  –Sort out gross structure in early iterations
  –Fine structure later

•Preprocessing of input signals + scaling

PROPERTIES OF THE SOM

After convergence, the map will have the properties:

•Topological Ordering
(as far as possible given the topology of the output space)

•Density Matching
There will be more units in high-density regions of the input space

APPLICATIONS

- Many! Thousands of SOM papers
- Phonetic typewriter (early application by Kohonen)
  - Convert short (about 10ms) slices of sound to 15 frequency bands + volume
  - Train network on 16D vectors
  - Label network with phoneme names
  - Rule-based post-processing improves recognition accuracy
- Robot map-making