EE 583
PATTERN RECOGNITION

NeuroPR : Non-feedforward solutions

Hopfield Approach

Unsupervised Learning in NeuroPR

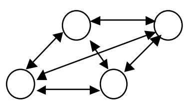
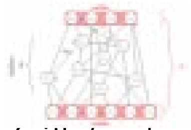
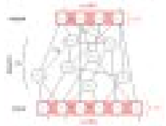
Self-Organizing Feature Maps

ART-based approaches

Introduction

- Feedforward networks are not the only solution to supervised NeuroPR problems
- There are other network topologies which have feedbacks in their connections in contrast to feedforward-only structures
  - Hopfield nets
- Some NeuroPR problems are lack of labeled (supervised) training data
- For those cases, the neural networks are used to determine some natural clusters or features from unlabeled samples.
  - Kohonen's Self Organizing Feature Maps
  - Adaptive Resonance Theory (ART)-based solutions

Hopfield Approach to NeuroPR (1/5)

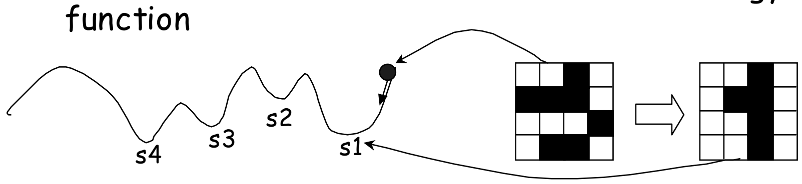




- Assume a network (with *d* number of cells) in which every cell (neuron) has connections with all other cells
  - o_i : output (state) of the i^{th} neuron
  - α_i : activation threshold of the i^{th} neuron (usually α_i is equal to 0)
  - w_{ij} : interconnection weight from neuron j to neuron i
- There are d(d-1)/2 possibly non-zero and unique weights
- All connections are two-way with equal weights; i.e. $w_{ij} = w_{ji}$
- There are no self-connections; i.e. $w_{ii}=0$
- A simple form for Hopfield neuron firing characteristic :

$$o_i = \begin{cases} +1 & \text{if } \sum_{j, j \neq i} w_{ij} o_j > \alpha_i \\ -1 & \text{otherwise} \end{cases}$$

Hopfield Approach to NeuroPR (2/5)

- Due to their interconnected structure, beginning from an initial state (output) vector, Hopfield networks may oscillate indefinitely according to their weights
- For Hopfield network, “learning” is simply constructing some stable states that do not oscillate
- This problem (finding stable states) is equivalent to defining an energy function to be minimized while stable states as the local minima of this energy function



Hopfield Approach to NeuroPR (3/5)

- Consider a simple net : (i) ←w→ (j) $(assume\,\alpha_i = 0)$
- for w>0: i=+1, j=+1 ➔ i : +1→+1→+1 … j : +1→+1→+1 … (stable)
  i=-1, j=+1 ➔ i : -1→+1→-1… j : +1→-1→+1 … (unstable)
  i=-1, j=-1 ➔ i : -1→-1→-1… j : -1→-1→-1 … (stable)
- for w<0: i=+1, j=+1 ➔ i : +1→-1→+1 … j : +1→-1→+1 … (unstable)
  i=-1, j=+1 ➔ i : -1→-1→-1 … j : +1→+1→+1 … (stable)
  i=-1, j=-1 ➔ i : -1→+1→-1… j : -1→+1→-1 … (unstable)
- A possible energy function : $e_{ij} = -w_{ij}o_j o_i$
  - For w>0, stable states (+1,+1)&(1,-1) gives $\min\{e_{ij}\}$
  - For w<0, stable states (+1,-1)&(-1,+1) gives $\min\{e_{ij}\}$
- For multiple neurons :

$$E = -\sum_{pairs} w_{ij} o_i o_j = -\left(\frac{1}{2}\right) \sum_i \sum_{j, j \neq i} w_{ij} o_i o_j$$

- A possible weight determination strategy : $w_{ij} = o_i o_j \quad i \neq j$

o_i	o_j	e_{ij}
-1	-1	$-W_{ij}$
-1	+1	$+W_{ij}$
+1	-1	$+W_{ij}$
+1	+1	$-W_{ij}$

Hopfield Approach to NeuroPR (4/5)

- For n training data, weight determining strategy :

$$w_{ij} = \sum_{s=1}^{n} o^s{}_i o^s{}_j \qquad i \neq j$$

- This weight finding strategy is quite similar to that of correlation based formulation
- Due to nonlinearity of multiple connections, it is possible to have local minimums other than the prescribed ones
- One important design aspect is maximizing the distance between different local minimums
- Simulations show that a network consisting of n neurons allows approximately about 0.15n stable states

Hopfield Approach to NeuroPR (5/5)

- Lets analyze the proposed energy function in terms of stability concerns :

$$E = -\frac{1}{2}\sum_i \sum_{j,i\neq j} w_{ij} o_i o_j = -\frac{1}{2}\underline{o}^T W \underline{o} \qquad \underline{o} \equiv [o_i], W \equiv [w_{ij}], w_{ii} = 0$$

$$\Rightarrow \frac{\partial E(\underline{o})}{\partial \underline{o}} = -W\underline{o} \quad \Rightarrow \frac{\Delta E(\underline{o})}{\Delta o_i} = -\sum_{j,i\neq j} w_{ij} o_j$$

$$\text{If} \quad -\sum_{j,i\neq j} w_{ij} o_j < 0 \quad \text{then} \quad \sum_{j,i\neq j} w_{ij} o_j > 0$$

$$\Rightarrow \Delta o_i = 0 \ \text{ or } \ \Delta o_i > 0 \quad \Rightarrow \quad \Delta E \ \textit{not} +$$

- As iterations goes to infinity, since outputs & E are bounded, output should converge to a value which represents a local minimum in E

Example : Hopfield Net

- Problem : Store state (+1,-1,+1,-1) to a Hopfield net
- For 4 states (output), there are 4*3/2=6 weights :

$$w_{ij} = o_j^s o_i^s \qquad \underline{o}^s = \begin{pmatrix} o_1^s \\ o_2^s \\ o_3^s \\ o_4^s \end{pmatrix} = \begin{pmatrix} +1 \\ -1 \\ +1 \\ -1 \end{pmatrix} \Rightarrow W = \begin{bmatrix} 0 & -1 & +1 & -1 \\ -1 & 0 & -1 & +1 \\ +1 & -1 & 0 & -1 \\ -1 & +1 & -1 & 0 \end{bmatrix}$$

$$W\begin{pmatrix} +1 \\ -1 \\ +1 \\ -1 \end{pmatrix} = \begin{pmatrix} +3 \\ -3 \\ +3 \\ -3 \end{pmatrix} \overset{*}{\Rightarrow} \begin{pmatrix} +1 \\ -1 \\ +1 \\ -1 \end{pmatrix}$$

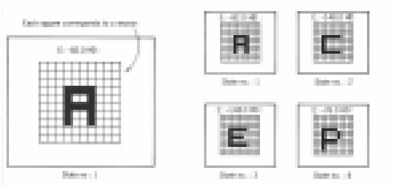
Stable (E=-6)

$$W\begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} +1 \\ +1 \\ +1 \\ +1 \end{pmatrix} \overset{*}{\Rightarrow} \begin{pmatrix} +1 \\ +1 \\ +1 \\ +1 \end{pmatrix} \mapsto W\begin{pmatrix} +1 \\ +1 \\ +1 \\ +1 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \overset{*}{\Rightarrow} \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

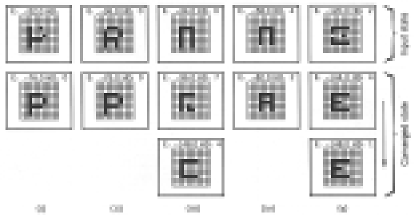
Oscillating between two states (E_1=+2,E_{-1}=-2)

Example : Character Recognition

- Hopfield nets do not have classification outputs
  ➔ mostly used for pattern completion problems



- Character Recognition :
  - 10x10 array mapped to a network of 100 neurons
  - matrix W : 100x100
  - 4 states are stored



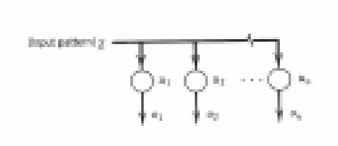
- Convergence properties :
  - For distorted patterns, the system reaches to one of the stable states

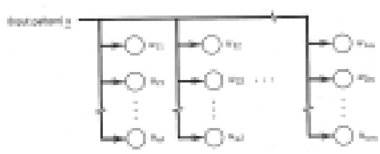
Unsupervised Learning in NeuroPR

- Multilayer feedforward and Hopfield neural nets are both examples for supervised learning
- There are also some networks with "cluster discovery" capability ➔ self-organizing
- Pattern similarity is crucial to measure the learning process in unsupervised learning
- Two main directions in unsupervised learning for NeuroPR :
  - Self-Organizing Feature Maps
  - Adaptive Resonance Theory (ART)-based architectures

Self-Organizing Feature Maps (1/4)

- One of the ways of unsupervised NeuroPR is based on dimensionality reduction
  - conversion of higher dimensional feature space to lower dimensional (topologically related) clustering diagrams (feature maps)
- The selection of feature map dimension requires a good judgment
  - 1-D or 2-D topologies have been successful

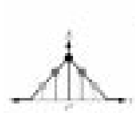


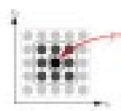


- The main idea is to classify "similar" features to "similar" locations

Self-Organizing Feature Maps (2/4)

Algorithm :





1. Define an equivalent dimension neighborhood, Λ_i, around each neuron, y_i (each neuron has a weight vector $\mathbf{w}_i$)
2. Before iterations, initialize network with random weights (iteration index : k)
3. Find neuron, y^*, which has "minimum distance" $d(\mathbf{x},\mathbf{w}_i)$ between $\mathbf{w}_i(k)$ & input vector $\mathbf{x}(k)$

   $d(\mathbf{x},\mathbf{w}_i)$: inner product or Euclidian distance
4. Let the winner neuron index be c, then all the weights of the units inside Λ_c are updated according to a window function $\Lambda(d)$ which assigns smaller values for larger d

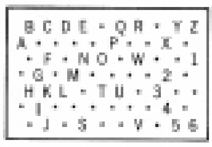
   $$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta(k)\ \Lambda(|\,y^* - y_i\,|)\ \mathbf{x}(k) \quad i \in \Lambda_c,$$

   (i.e. weights become similar to $\mathbf{x}$)
5. Shrink Λ_c & normalize $|\mathbf{w}_i|=1$ & decrease $\eta(k)$ during iterations

Self-Organizing Feature Maps (3/4)

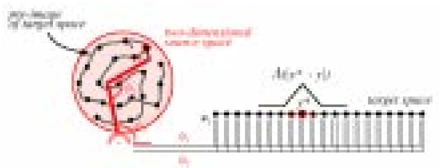
- <u>Example</u>: A set of 5-D feature vectors are organized in 2-D
  - 2-D map is chosen as hexagonal 7x10 units
  - Input vectors are randomly chosen
  - Radius of Λ_i is decreased from 6 to 1 (while k<1000)
  - Total iterations = 10,000
  - $\eta(k)$ is decreased linearly from 0.5 to 0.04

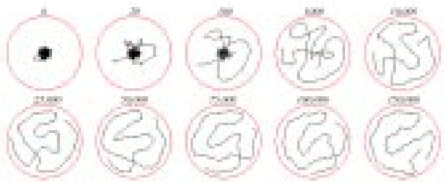
Attribute \ Item	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
a1	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
a2	0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
a3	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	3	3	3	3	6	6	6	6	6	6	6	6	6	6
a4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	1	2	3	4	2	2	2	2	2	2
a5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6



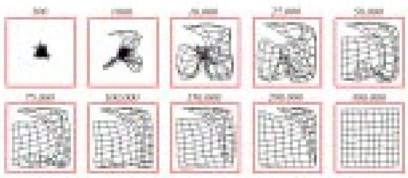
Self-Organizing Feature Maps (4/4)

- <u>Example</u>: 2-D (source space) to 1-D (target space) mapping
(Note that for every point in 2-D, there is a single neuron, y, with a maximum activation value





- <u>Example</u>: 2-D (square source) to 2D (square grid space) mapping



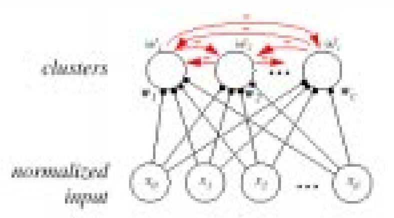
Some initial random weights and a particular sequence of patterns may not yield correct mapping

Adaptive Resonance Architectures (1/4)

- In order to understand ART architectures, "on-line clustering" should be understood
  - On-line clustering : clustering without storing data
- Input is checked whether there is a match with clusters weights
- Only the weight for the "most active cluster" is modified

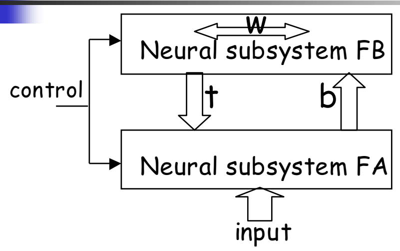
  $\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta(k)\,\mathbf{x}(k)$
- "Most active cluster" is determined by a "competition" between clusters

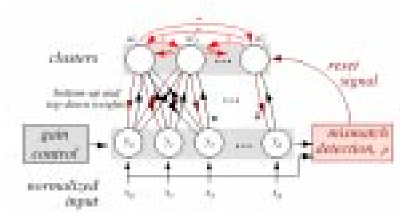
  ➔ Competitive learning



- If the number of clusters, c, is not constant and a data is quite different from all clusters, then a new cluster will be formed

Adaptive Resonance Architectures (2/4)

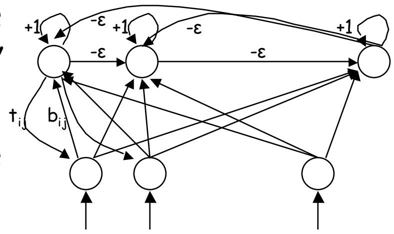




- Typical ART system consists of 2 neural subsystems :
  - FA layer : a bottom layer holding both input and weighted (t_{ij}) outputs of FB layer
  - FB layer : fed with weighted (b_{ij}) outputs of FA layer, FB is a fully interconnected top layer with cells each representing a pattern class
- After a competition at the output, the winner class feedbacks FA to receive a "clean-up" version of the input
- A "resonance" between two layers occur when the encoded and input patterns are matched

Adaptive Resonance Architectures (3/4)

- In ART network two phases exist which are governed by control signals :
  - Attentionalphase : input data
  - Recognition phase : find class (only one pattern class wins)



- The recognition phase is a cyclic process for
  - modifying weights b from FA to FB for selecting a pattern class in FB (FB outputs have neighbor inhibiting structure)
  - mapping this result back to FA until a consistent result at FA is achieved (learned expectations)
  - Consistent result back at FA→ resonance :

Adaptive Resonance Architectures (4/4)

<u>Algorithm</u>:

- Initialize interlayer connections t and b
- Present input, **x**=(x_1,…,x_d), to FA layer
- Using b_{ij}, determine inputs for FB layers
- Find winner output at FB layer (iterations within FB until one winner is left)
- The winner output is feedback to each FA unit; the outputs of these units are compared back to **x**
- If comparison shows similarity, <u>update</u> b and t. If they are not sufficiently similar, the corresponding output is left out and a new winner is obtained to perform the same steps
  - Update process only take into account winner output

Final Words on NeuroPR

- Mapping of a PR problem into the neural domain is not straightforward
- Selection of network topology and parameters is one major problem
- Some other problems that require further research
  - Encoding of relational information via neural networks
  - Problems with hierarchical structures
  - Feature extraction and recognition with incomplete features
  - Convergence analysis for some networks
  - Lack of formal tools for the design of neural nets
  - Implementation of neural nets in hardware using massively parallel analog circuits