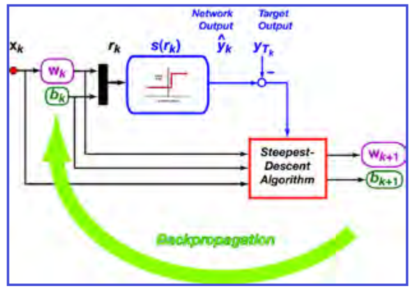
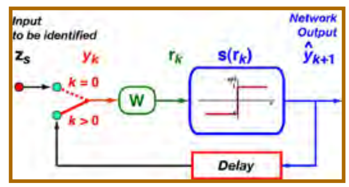
Neural Networks

Robert Stengel
Robotics and Intelligent Systems, MAE 345,
Princeton University, 2015

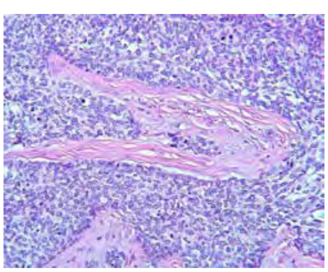
Learning Objectives

- Associative/recurrent networks
  - Hopfield network
  - Adaptive resonance theory network
- Unsupervised training
  - k-means clustering
  - Self-organizing map
- Deep learning
  - Restricted Boltzmann machine
  - Semi-supervised learning



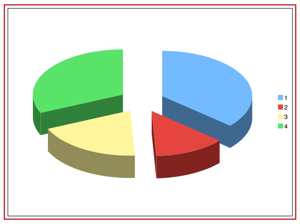


Small, Round Blue-Cell Tumor Classification Example



Desmoplastic small, round blue-cell tumors

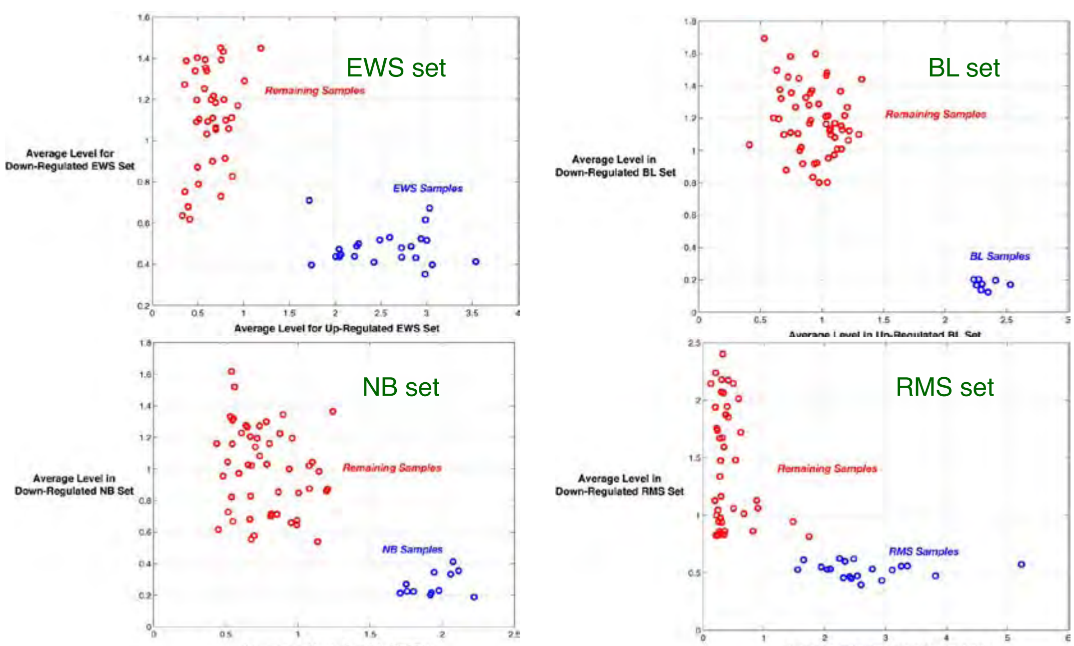
- Childhood cancers, including
  - Ewing's sarcoma (EWS)
  - Burkitt's Lymphoma (BL)
  - Neuroblastoma (NB)
  - Rhabdomyosarcoma (RMS)
- cDNA microarray analysis presented by **J. Khan, *et al.*, *Nature Medicine*, 2001, 673-679.**
  - 96 transcripts chosen from 2,308 probes for training
  - 63 EWS, BL, NB, and RMS training samples
- **Source of data for my analysis**



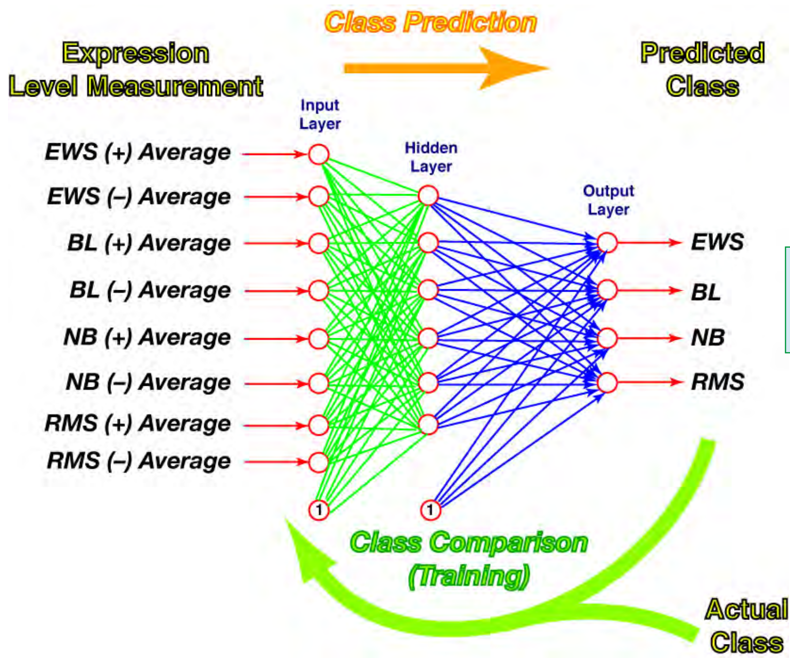
Overview of Present SRBCT Analysis

- **Transcript selection by *t* test**
  - **96 transcripts, 12 highest and lowest *t* values for each class**
  - **Overlap with Khan set: 32 transcripts**
- **Ensemble averaging** of genes with highest and lowest *t* values in each class
  - **Cross-plot** of ensemble averages
- **Classification by sigmoidal neural network**
  - **Validation** of neural network
    - **Novel set simulation**
    - **Leave-one-out simulation**

Clustering of SRBCT Ensemble Averages



SRBCT Neural Network

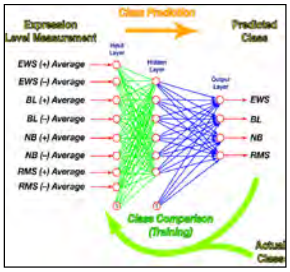


Binary vector output (0,1) after rounding

Neural Network Training Set

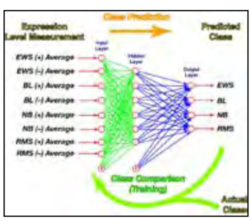
Each input row is an ensemble average for a transcript set, normalized in (–1,+1)

Identifier	*Sample* 1	*Sample* 2	*Sample* 3	...	*Sample* 62	*Sample* 63
Target Output	*EWS*	*EWS*	*EWS*	...	*RMS*	*RMS*
Transcript Training Set	*EWS*(+)*Average*	*EWS*(+)*Average*	*EWS*(+)*Average*	...	*EWS*(+)*Average*	*EWS*(+)*Average*
	EWS(−)*Average*	*EWS*(−)*Average*	*EWS*(−)*Average*	...	*EWS*(−)*Average*	*EWS*(−)*Average*
	BL(+)*Average*	*BL*(+)*Average*	*BL*(+)*Average*	...	*BL*(+)*Average*	*BL*(+)*Average*
	BL(−)*Average*	*BL*(−)*Average*	*BL*(−)*Average*	...	*BL*(−)*Average*	*BL*(−)*Average*
	NB(+)*Average*	*NB*(+)*Average*	*NB*(+)*Average*	...	*NB*(+)*Average*	*NB*(+)*Average*
	NB(−)*Average*	*NB*(−)*Average*	*NB*(−)*Average*	...	*NB*(−)*Average*	*NB*(−)*Average*
	RMS(+)*Average*	*RMS*(+)*Average*	*RMS*(+)*Average*	...	*RMS*(+)*Average*	*RMS*(+)*Average*
	RMS(−)*Average*	*RMS*(−)*Average*	*RMS*(−)*Average*	...	*RMS*(−)*Average*	*RMS*(−)*Average*



SRBCT Neural Network Training

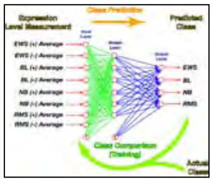
- Neural network
  - 8 ensemble-average inputs
  - various # of sigmoidal neurons
  - 4 linear output neurons
  - 4 outputs
- Training accuracy
  - Train on all 63 samples
  - Test on all 63 samples
- 100% accuracy



Leave-One-Out Validation of SRBCT Neural Network

- Remove a single sample
- Train on remaining samples (125 times)
- Evaluate class of the removed sample
- Repeat for each of 63 samples

- 6 sigmoids: 99.96% accuracy (3 errors in 7,875 trials)
- 12 sigmoids: 99.99% accuracy (1 error in 7,875 trials)



Novel-Set Validation of SRBCT Neural Network

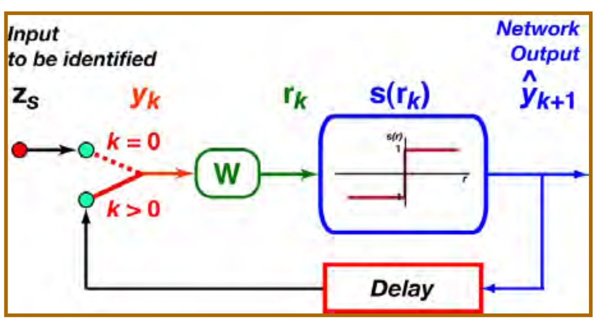
- Network always chooses one of four classes (i.e., “unknown” is not an option)
- Test on 25 novel samples (400 times each)
  - 5 EWS
  - 5 BL
  - 5 NB
  - 5 RMS
  - 5 samples of unknown class
- 99.96% accuracy on first 20 novel samples
  (3 errors in 8,000 trials)
- 0% accuracy on unknown classes

Observations of SRBCT Classification using Ensemble Averages

- *t* test identified strong features for classification in this data set
- Neural networks easily classified the four data types
- Caveat: Small, round blue-cell tumors occur in different tissue types
  - Ewing’s sarcoma: Bone tissue
  - Burkitt’s Lymphoma: Lymph nodes
  - Neuroblastoma: Nerve tissue
  - Rhabdomyosarcoma: Soft tissue

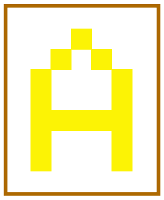
Gene expression (i.e., mRNA) level is linked to tissue difference as well as tumor genetics

Recurrent Networks

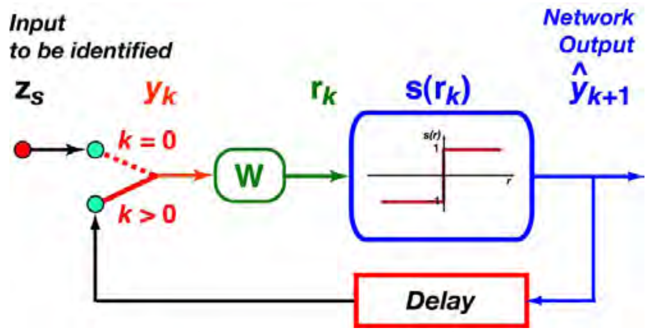


Recurrent Networks

- Recursion to identify an unknown object
  - Network is given a single, fixed input, and it iterates to a solution
- Convergence and stability of the network are critical issues
- Single network may have many stable states
  - Classified outputs of the map
  - Pattern recognition with noisy data



Hopfield Network



$$\mathbf{z}_s = \mathbf{y}_s + \mathbf{n}_s$$

- Bipolar (–1,1) inputs and outputs
  - dim(y) = n x 1
- Supervised training with perfect exemplar outputs
- Noisy measurement of an exemplar as input to be identified
- Network operation

$$\hat{\mathbf{y}}_0 = \mathbf{z}_s$$

$$\hat{\mathbf{y}}_{k+1} = \mathbf{s}(\mathbf{r}_k) = \mathbf{s}(\mathbf{W}\hat{\mathbf{y}}_k)$$

$$\hat{y}_{i_{k+1}} = \begin{cases} 1, & r_{i_k} > 0 \\ Unchanged, & r_{i_k} = 0 \\ -1, & r_{i_k} < 0 \end{cases} \quad, i = 1 \; to \; n$$

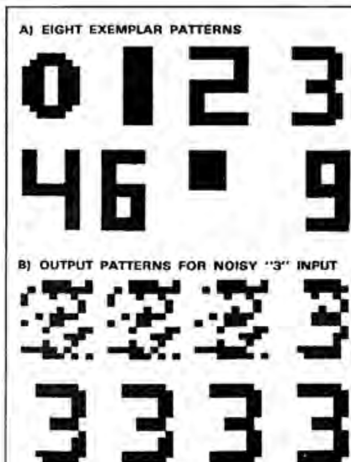
- Iterate to convergence

Training a Hopfield Network

- Network training
  - Given M exemplars, $\mathbf{y}_s$ (n x 1)
  - Each exemplar is a character represented by n pixels
  - Batch calculation of weighting matrix

$$\mathbf{W} = \sum_{s=1}^{M} \left(\mathbf{y}_s \mathbf{y}_s^T - \mathbf{I}_n \right)$$
$$= \begin{bmatrix} y_1^2 - 1 & y_1 y_2 & \ldots \\ y_1 y_2 & y_2^2 - 1 & \ldots \\ \ldots & \ldots & \ldots \end{bmatrix}$$

- No iterations to define weights
- Large number of weights
- Limited number of exemplars (< 0.15 n)
- Similar exemplars pose a problem

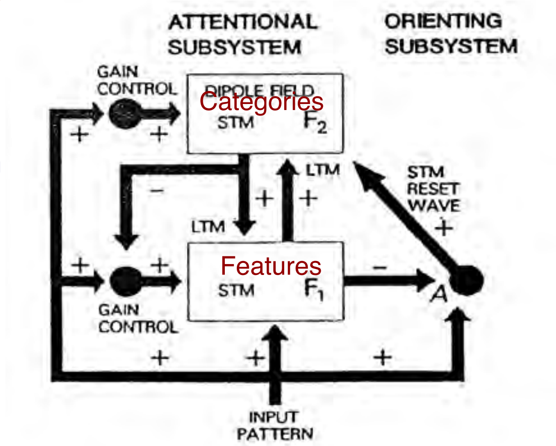


An example of the behavior of a Hopfield net when used as a content-addressable memory. A 120 node net was trained using the eight exemplars shown in (A). The pattern for the digit "3" was corrupted by randomly reversing each bit with a probability of .25, and then applied to the net at time zero. Outputs at time zero and after the first seven iterations are shown in (B).

$$n = 120; \quad M = 8$$
$$\#\, weights = n^2 = 14{,}400$$

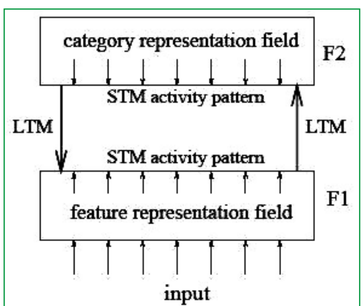
Adaptive Resonance Theory Network
(Grossberg, Carpenter, 1976)

- Self-organizing and self-stabilizing network for binary pattern recognition (ART-1)
- Subjective evolution through “Fuzzy ART”
- Learns new patterns when it discerns sufficient mismatch from old patterns
  - Long-Term Memory
  - Short-Term Memory
  - Stability and plasticity
  - Unsupervised and supervised learning
  - “Bottom-up” input
  - “Top-down” priming
  - Pre-cursor to “deep learning”



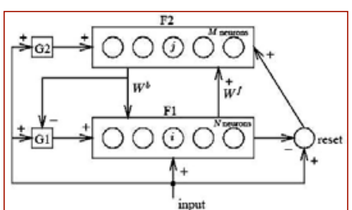
Adaptive Resonance Theory Network

Architecture



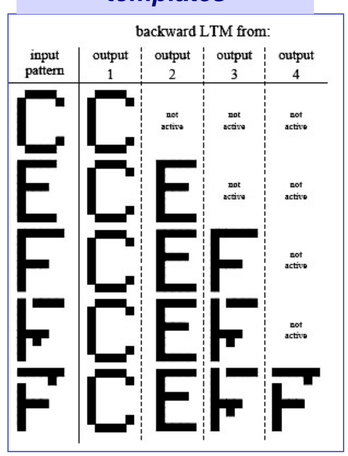
The ART architecture.

Binary Neurons represent Pattern Pixels



The ART1 neural network.

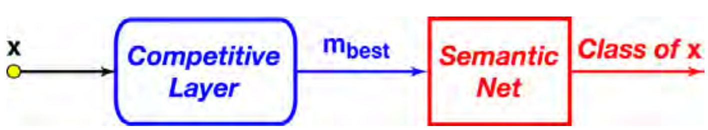
Recursive Training Example: adding new templates



Neural Networks with Unsupervised Learning

Self-Organizing Map

(Kohonen, 1981)



- Competitive, unsupervised learning
- Premise: input signal patterns that are close produce outputs that are close
- Ordered inputs produce a spacial distribution, i.e., a map
- Cells of the map are likened to the cell structure of the cerebral cortex

- **x**: (n x 1) input vector characterizes features (attributes) of a signal
- **m**: (n x 1) weight vector of a cell that represents an output class

Competition in the Self-Organizing Map

- Competition is based on minimizing distance from x to m

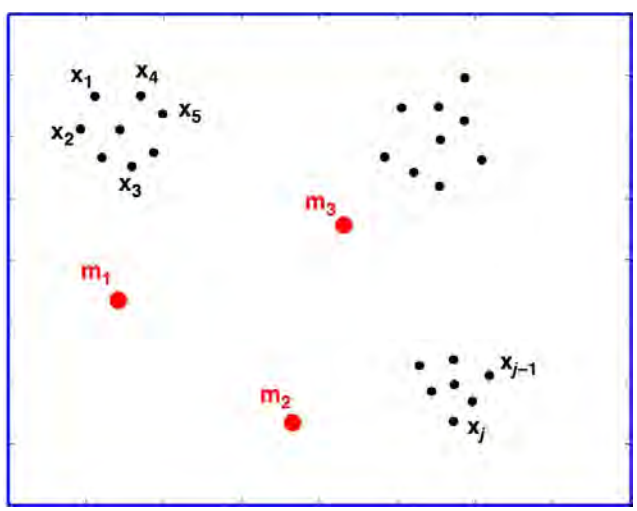
$$\boxed{\begin{aligned} &Cost = distance = \|\mathbf{x} - \mathbf{m}_i\| \\ &\min Cost = \min_{\mathbf{m}_i} \|\mathbf{x} - \mathbf{m}\| \end{aligned}}$$

- m encodes the output classes
- Semantic net decodes the output to identify classes

$$\boxed{\mathbf{m}_1 = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix} \rightarrow Class\ A; \quad \mathbf{m}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \rightarrow Class\ B}$$

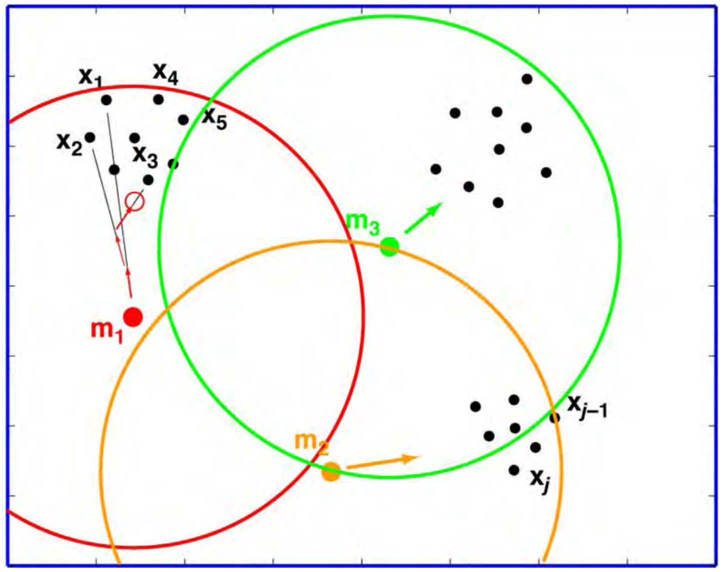
Goal of the Self-Organizing Map

- **Given:**
  - *I* output classes
  - Input training set, x_j, $j = 1$ to J
- **Find:** Cell weights, m_i, $i = 1$ to I that best cluster the data (*i.e., with minimum norm*)
- **Initialize** the cell weights, m_i, randomly in the space of x



Training the Self-Organizing Map

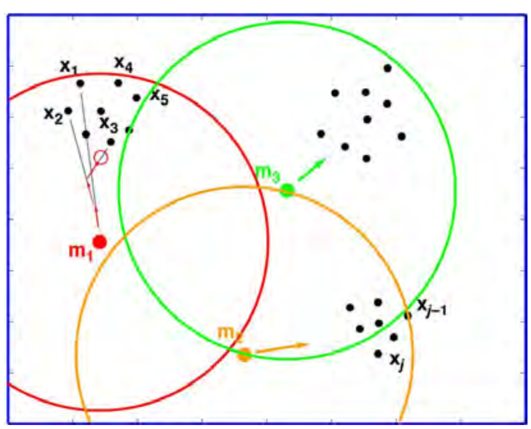
- Define a neighborhood set within a radius of N_c around each cell, m_i
  - Choose N_c to overlap with neighboring cells
- Find the best cell-weight match, m_{best}, (i.e., the closest m_i) to the 1[st] training sample, x_1



Cell Weight Updates

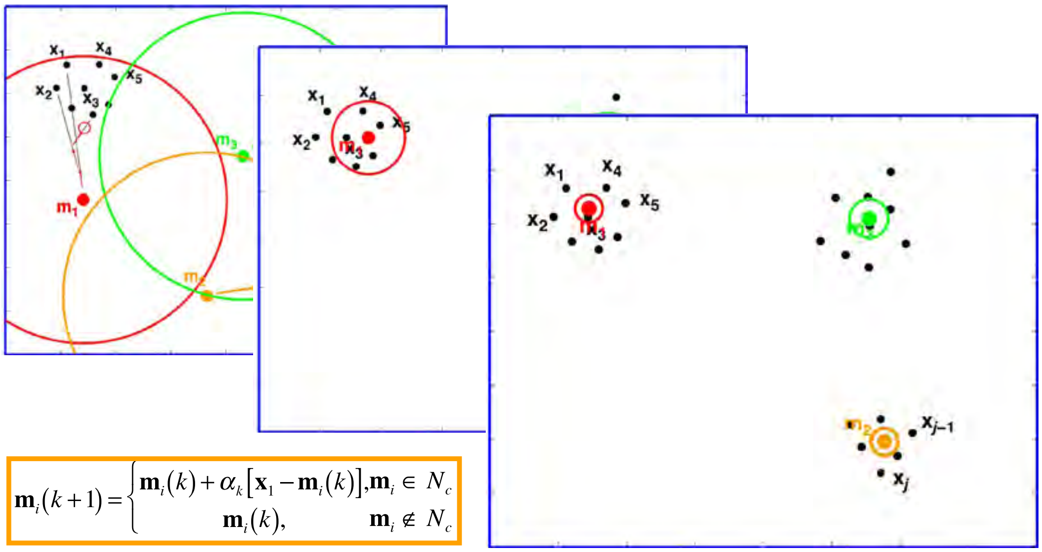
$$\mathbf{m}_i(k+1) = \begin{cases} \mathbf{m}_i(k) + \alpha_k\left[\mathbf{x}_1 - \mathbf{m}_i(k)\right], & \mathbf{m}_i \in N_c \\ \mathbf{m}_i(k), & \mathbf{m}_i \notin N_c \end{cases}$$

- Update cell weights for all cells in the neighborhood set, N_c, of m_{best}
  - α_k = adaptation gain or learning rate
- Repeat for
  - x_2 to x_J
  - m_1 to m_I



Convergence of Cell Weights

Repeat entire process with decreasing N_c radius until convergence occurs

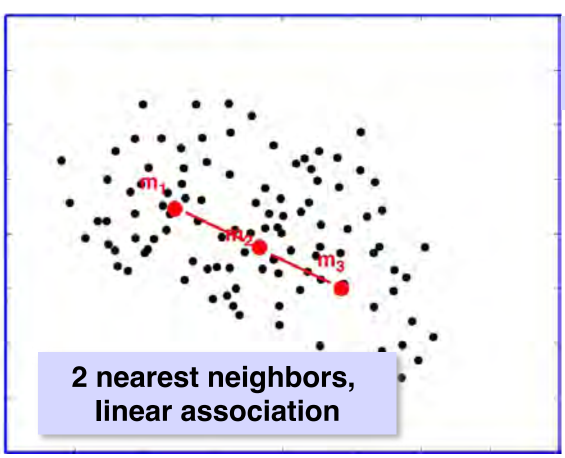


$$\mathbf{m}_i(k+1) = \begin{cases} \mathbf{m}_i(k) + \alpha_k\left[\mathbf{x}_1 - \mathbf{m}_i(k)\right], & \mathbf{m}_i \in N_c \\ \mathbf{m}_i(k), & \mathbf{m}_i \notin N_c \end{cases}$$

Semantic Map

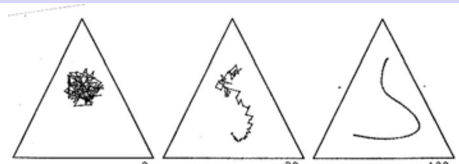
- **Representation of abstract or categorical information**
- **Contextual information used to generate map of symbols**
- **Dimensionality and # of nearest neighbors affects final map**

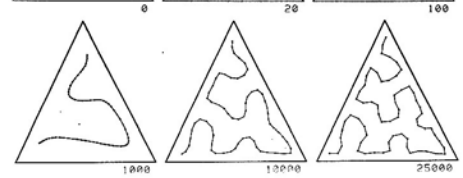
***Example*: linear association of cell weights**
Points for cell-weight update chosen randomly



2 nearest neighbors, linear association

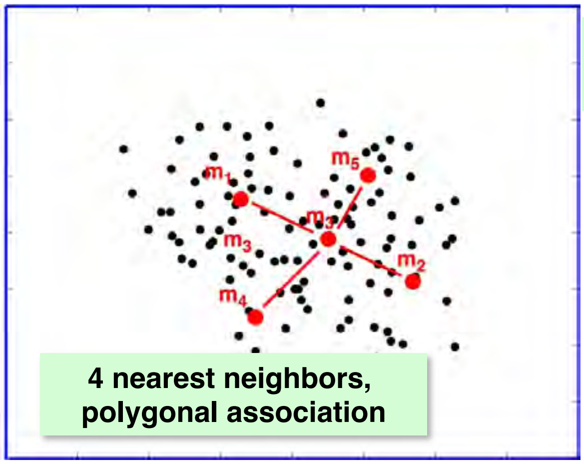
Evolution of points on a line that identifies locations of m_i
(Uniform random field of data points not shown)



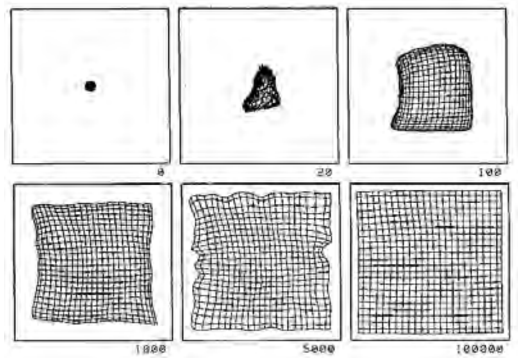


Choice of Neighborhood Architecture

***Example*: Map is assumed to represent a grid of associated points**
Number of cell weights specified
Random starting locations for training



Evolution of grid points that identify locations of m_i
(Uniform random field of data points not shown)



Minimum Spanning Tree

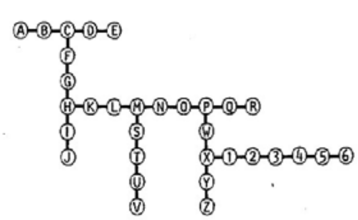
Example of hexagonal map association identification
32 points with 5 attributes that may take six values (0, 1, 2, 3, 4, 5)

Input Data Matrix

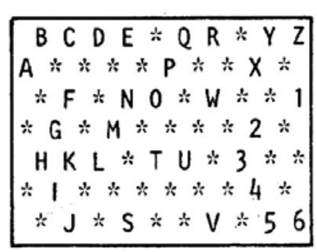
Attribute \ Item	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
a_1	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
a_2	0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
a_3	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	3	3	3	3	6	6	6	6	6	6	6	6	6	6
a_4	0	0	0	0	0	0	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
a_5	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

Minimum spanning tree: smallest total edge length

Hexagonal lattice of grid points that identify locations of m_i



Minimal spanning tree corresponding to Table



Self-organized map of the data matrix of Table

Semantic Identification

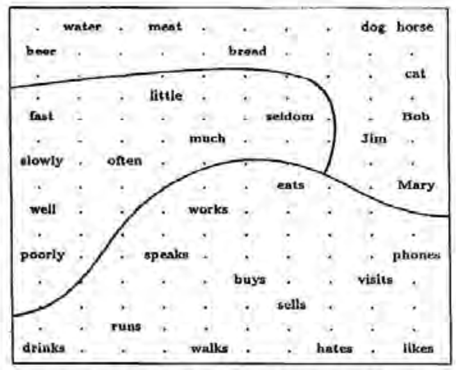
Example of semantic identification
Each item for training has symbolic expression and context
Categories: noun, verb, adverb

(a)	
Bob/Jim/Mary	1
horse/dog/cat	2
beer/water	3
meat/bread	4
runs/walks	5
works/speaks	6
visits/phones	7
buys/sells	8
likes/hates	9
drinks/eats	10/11
much/little	12
fast/slowly	13
often/seldom	14
well/poorly	15

Sentence Patterns: (b)		
1-5-12	1-9-2	2-5-14
1-5-13	1-9-3	2-9-1
1-5-14	1-9-4	2-9-2
1-6-12	1-10-3	2-9-3
1-6-13	1-11-4	2-9-4
1-6-14	1-10-12	2-10-3
1-6-15	1-10-13	2-10-12
1-7-14	1-10-14	2-10-13
1-8-12	1-11-12	2-10-14
1-8-2	1-11-13	2-11-4
1-8-3	1-11-14	2-11-12
1-8-4	2-5-12	2-11-13
1-9-1	2-5-13	2-11-14

(c)
Mary likes meat
Jim speaks well
Mary likes Jim
Jim eats often
Mary buys meat
dog drinks fast
horse hates meat
Jim eats seldom
Bob buys meat
cat walks slowly
Jim eats bread
cat hates Jim
Bob sells beer
(etc.)

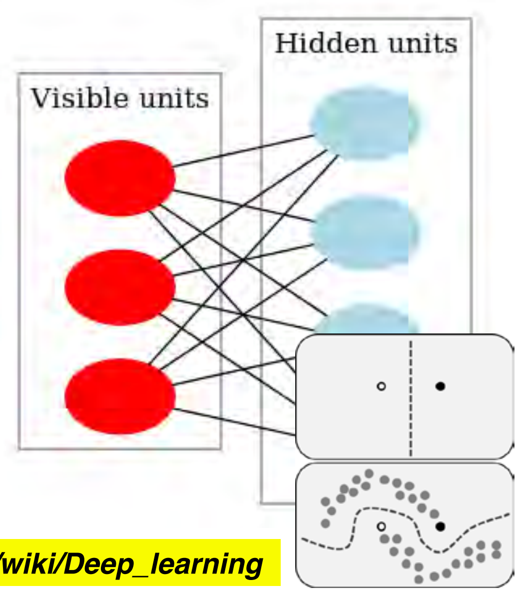
Outline of vocabulary used in this experiment. (a) List of used words (nouns, verbs, and adverbs), (b) sentence patterns, and (c) some examples of generated three-word-sentences.



"Semantic map" obtained on a network of 10 × 15 cells after 2000 presentations of word-context-pairs derived from 10 000 random sentences of the kind shown in Fig. 10(c). Nouns, verbs, and adverbs are segregated into different domains. Within each domain a further grouping according to aspects of meaning is discernible.

Deep Learning

- Multi-layered network
- Single sigmoid layer ~ Restricted Boltzmann machine (RBM)
- Pre-train each layer separately and contextually (unsupervised)
- Fine-tune with backpropagation
- Semi-supervised learning
  - Initial clustering
  - Smoothness
  - Manifold
- Overcoming "vanishing gradient" problem in multi-layer backpropagation
- See Ch. 2, MATLAB User's Guide (Unsupervised autoencoder layers followed by supervised layer)



http://en.wikipedia.org/wiki/Deep_learning

Hinton et al, 2006

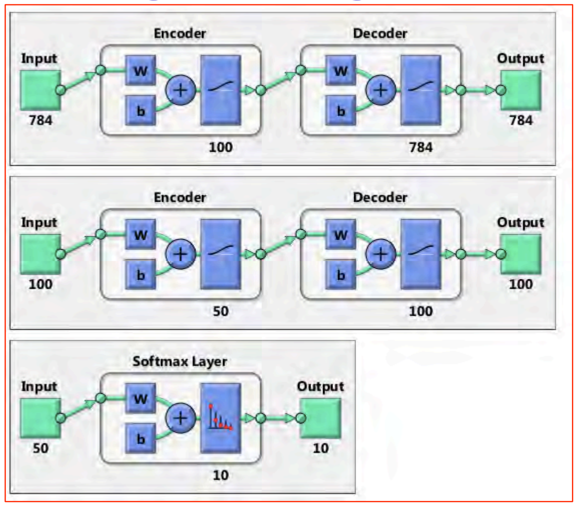
http://deeplearning.net

MATLAB Example
Deep Learning for Digit Classification

1st Encoder Unsupervised

2nd Encoder Unsupervised

Preliminary NN training Supervised

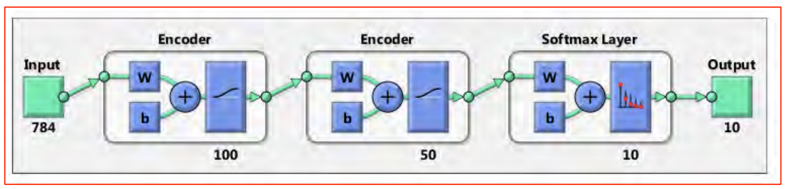


Output matched to input (1)

Output matched to input (2)

Output (2) trained for classification

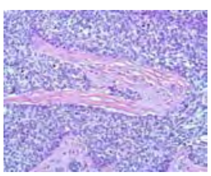
DeepNet trained to classes



29

Next Time:
Communication, Information, and Machine Learning

Supplemental Material



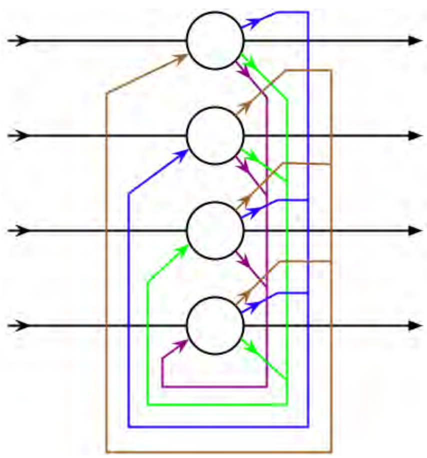
Comparison of Present SRBCT Set with Khan Top 10

Image ID	Gene Description	EWS Student t Value	BL Student t Value	NB Student t Value	RMS Student t Value	Most Significant t Value	Khan Gene Class
296448	insulin-like growth factor 2 (somatomedin A)	-4.789	-5.226	-1.185	5.998	RMS	RMS
207274	Human DNA for insulin-like growth factor II (IGF-2); exon 7 and additional ORF	-4.377	-5.424	-1.639	5.708	RMS	RMS
841641	cyclin D1 (PRAD1: parathyroid adenomatosis 1)	6.841	-9.932	0.565	-4.300	BL (-)	EWS/NB
365826	growth arrest-specific 1	3.551	-8.438	-6.995	1.583	BL (-)	EWS/RMS
486787	calponin 3, acidic	-4.335	-6.354	2.446	2.605	BL (-)	RMS/NB
770394	Fc fragment of IgG, receptor, transporter, alpha	12.037	-6.673	-6.173	-4.792	EWS	EWS
244618	ESTs	-4.174	-4.822	-3.484	5.986	RMS	RMS
233721	insulin-like growth factor binding protein 2 (36kD)	0.058	-7.487	-1.599	2.184	BL (-)	Not BL
43733	glycogenin 2	4.715	-4.576	-3.834	-3.524	EWS	EWS
295985	ESTs	-9.260	-0.133	3.237	2.948	EWS (-)	Not EWS

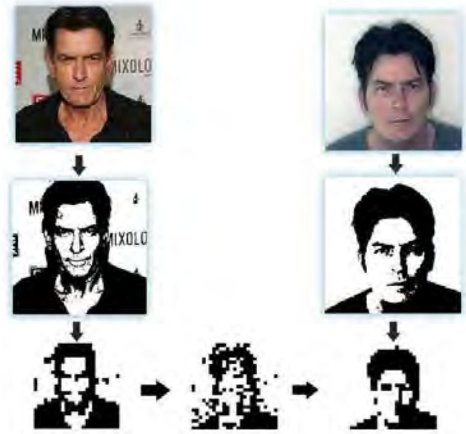
- **Red:** both sets
- Black: Khan set only

Hopfield Network

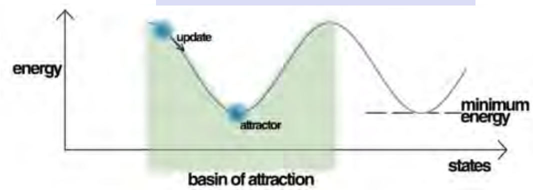
Alternative plot of 4-node network

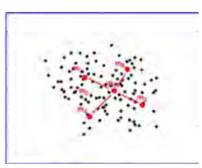


Novel Image **Exemplar**



"Energy Landscape"





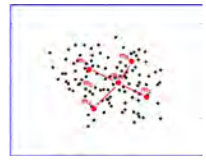
Linear Vector Quantization

- **Incorporation of supervised learning**
- **Classification of groups of outputs**

- **Type 1**
  - **Addition of codebook vectors, m_c, with known meaning**

$$\mathbf{m}_c(k+1)=\begin{cases}\mathbf{m}_c(k)+\alpha_k\left[\mathbf{x}_k-\mathbf{m}_c(k)\right], & \textit{if classified correctly}\\ \mathbf{m}_c(k)-\alpha_k\left[\mathbf{x}_k-\mathbf{m}_c(k)\right], & \textit{if classified incorrectly}\end{cases}$$

Linear Vector Quantization



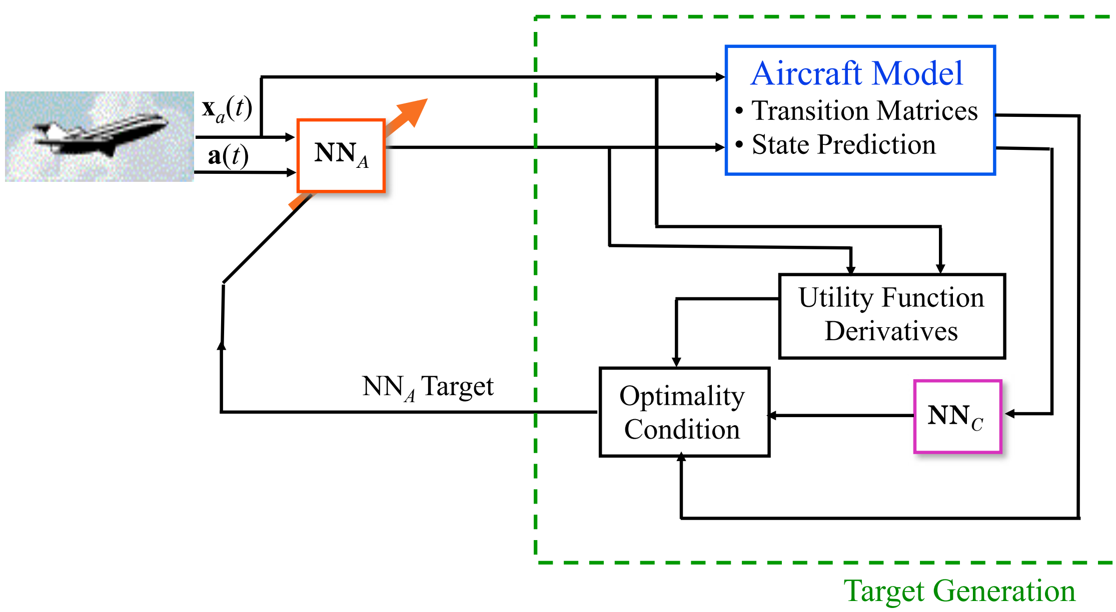
- Type 2
  - Inhibition of nearest neighbor whose class is known to be different, e.g.,
    - x belongs to class of $\mathbf{m}_j$ but is closer to $\mathbf{m}_i$

$$\mathbf{m}_i(k+1) = \mathbf{m}_i(k) - \alpha_k\left[\mathbf{x}_k - \mathbf{m}_i(k)\right]$$

$$\mathbf{m}_j(k+1) = \mathbf{m}_j(k) + \alpha_k\left[\mathbf{x}_k - \mathbf{m}_j(k)\right]$$

Adaptive Critic Proportional-Integral Neural Network Controller

Adaptation of Control Network



Adaptive Critic Proportional-Integral Neural Network Controller

Adaptation of Critic Network