Neuromorphic Artificial Intelligence: From Mathematical Foundations of Deep Learning to "Cortex-on-a-Chip"

John S. Baras

Institute for Systems Research and Dept. of Electrical and Computer Engin.

University of Maryland College Park, USA,

and ACCESS Centre Royal Instit. of Technology (KTH), Stockholm, Sweden,

and Instit. for Advanced Study Technical Univ. of Munich (TUM), Germany

June 4, 2018

NTUA, Athens, Greece

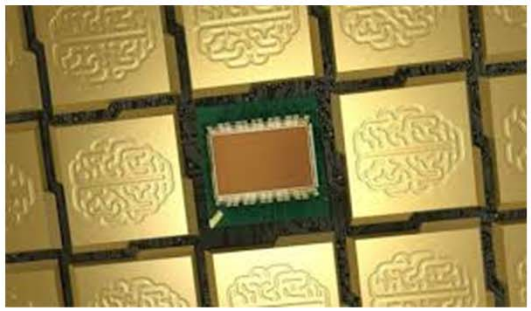
Cognitive Computing and AI

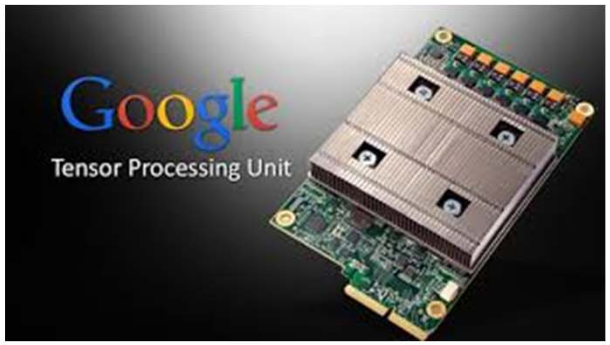
- **Increasing interest in Cognitive Computing and AI**
- **Market spending $12B in 2017; 60% increase from 2016**
- **By 2021 to reach $57.6B; annual growth rate of 50%**
  - Half of expenditures in software and cognitive platforms
  - Rest in hardware and services
  - Hardware expenditures to increase at 40% per year
- **Applications:**
  - Retail, banking, manufacturing, healthcare
  - Automated service agents, Diagnostic and treatment systems, Expert shopping advisors
  - Manufacturing, Intelligent processing automation, Product recommendations
  - Public safety and emergency response

Brain-Like Computers

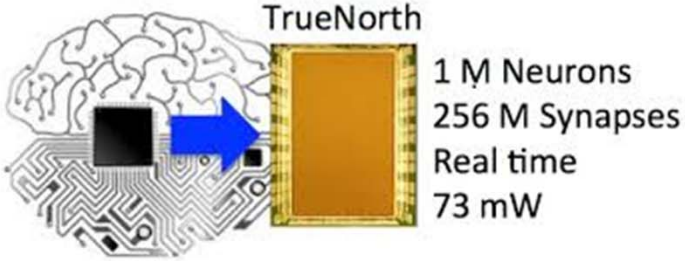
Race to design and manufacture "brain-like" computers is on

IBM





NEUROMOPRPHIC?



INTEL LOIHI

1000x more energy efficient
Spike based info processing
Storing info on synapses
130K neurons, 130M synapses

Feb 2018 INTEL establishes INTEL Neuromorphic Research Community (INRC) -- academic-industry-government group/consortium

Brain-Like Computers

- **Most promising: neuromorphic systems – modeled after the human brain**
  - Potential of being more generalizable than existing cognitive systems

- **Limitations of current cognitive computing systems**
  - Require a very large body of information to be ingested before having a knowledge base large enough to develop a model to which new data can be compared and classified -- For many applications such amounts of data may simply not be available
  - Training time is often extensive -- Human brain is much more efficient: after just a few examples a toddler can recognize a dog – in contrast with current image recognition systems
  - Cannot match human brain in power efficiency

Neural Networks vs Synaptic Networks

- **Can a computer ever simulate effects of Ms of neurons and Bs of synaptic connections (in the brain)?**
- **Key challenge: create synapses that emulate the brain**
- **Deep Learning Neural Networks are used for Optimization and Pattern Recognition**
- **Need technologies designed for decisions: system takes in data of various types, makes inferences, provides output as scored list of answers for a context or use case -- Machine, or human, then decides based on these answers**
- **Traditional NN:**
  - An input is provided, processed through multiple layers of nodes
  - An output layer provides answer (house, human, car, dog)
  - Hidden layers improve NN's interpretation of image
  - Training: processing many examples to adjust node weights

Neural Networks vs Synaptic Networks

- Synaptic networks: nodes indicate objects or features
- Connections between synaptic nodes indicate similarity or not of objects or features at the nodes
- When an unknown case is presented, certain nodes will be activated and others will be inhibited (as in brain)
- Other similar nodes can be activated to form inferences
- Learning rules provided by humans set the weights on the connections
- Synaptic networks via their wiring diagrams describe how objects, features relate to each other
- They do not need to be trained on large amounts of data
- Path from data to decisions is transparent: the "why" of a decision is provided through the activated and inhibited nodes – explainable decisions

Neural Networks vs Synaptic Networks

- **Human brains provide several options – rarely one answer**
- **Synaptic networks return different options based on data that are built over time**
- **Traditional NN must retrain when new data is presented**
- **Synaptic networks continues to add data as the data become available, constantly adjusting weights between nodes and building new inferences**
- **In traditional NN when a new type of data becomes available, the model has to be rebuilt from the ground up vs adding knowledge as it is learned**

Hierarchical Temporal Memory

- **The Hierarchical Temporal Memory (HTM) framework is modeled after the neocortex of the human brain**
  - Applied to anomaly detection and mitigation in the cloud (e.g. Grok)
  - Applied to Natural Language Processing (NLP)
- **Inspiration comes from understanding how key features of the brain support memory and learning**
- **Neocortex is very homogeneous: each part carries the same type of operation even though different parts of the neocortex process different types of information (vision, language, etc.)**
- **Thus if a common algorithm can be developed to replicate the functioning of the neurons, it could generalize to other tasks in an intelligent computer system in a way that current Machine Learning (ML) cannot**

Hierarchical Temporal Memory

- **The neocortex receives two types of inputs: One is sensory and the other is information about actions that are being carried out by the person (e.g. speaking, walking, eye movements)**
  - Both inputs provide streams of data in time-based patterns
  - The neocortex stores this information , the memory of which allows it to make predictions based on experience
  - The information then directs the person's future actions
- **Key: an intelligent computer needs to have a way to discover things about the world around it either through sensors or through directed actions for knowledge -- Current computer systems do not incorporate this aspect of computer intelligence**
- **Newborn humans and animals have learning structures that have evolved through many millennia – architecture ?**

Impact of AI ?

- **Automating Cognitive Tasks**
  - Alexa, Home, digital twins, …
- **Industrial Economy: primary challenge – automate the task**
- **New Intelligence Economy: understand and define the cognitive tasks our machine intelligence must carry out**
  - Multiple kinds of intelligence
  - Human - machine collaboration
  - Safety and safe learning
- **Impact on every aspect of life, work, society, etc. at both the individual and group level**
- **Impact on jobs**
- **Can we handle the technology?**

Key Problems/ Challenges

Advancing AI further needs: best of neuroscience, computer science and mathematics

- **Link Machine Learning with Knowledge Representation and Reasoning**
- **Progressive Learning**
- **Knowledge Compacting**
- **Rigorous analytics needed to create design tools to meet specifications**
- **Hardware architectures: hybrid (digital and analog)**
- **Non von-Neumann computing – do not separate CPU form Memory – in-memory processing**

Outline

- **Deep Learning**

DL Breakthrough Results in Understanding: speech, images, gestures, texts, …

Hinton et al, 2012 : http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/38131.pdf

modeling technique	#params [10^6]	WER Hub5'00-SWB	WER RT03S-FSH
GMM, 40 mix DT 309h SI	29.4	23.6	27.4
NN 1 hidden-layer×4634 units	43.6	26.0	29.4
+ 2×5 neighboring frames	45.1	22.4	25.7
DBN-DNN 7 hidden layers×2048 units	45.1	17.1	19.6
+ updated state alignment	45.1	16.4	18.6
+ sparsification	15.2 nz	16.1	18.5
GMM 72 mix DT 2000h SA	102.4	17.1	18.6

task	hours of training data	DNN-HMM	GMM-HMM with same data	GMM-HMM with more data
Switchboard (test set 1)	309	18.5	27.4	18.6 (2000 hrs)
Switchboard (test set 2)	309	16.1	23.6	17.1 (2000 hrs)
English Broadcast News	50	17.5	18.8	
Bing Voice Search (Sentence error rates)	24	30.4	36.2	
Google Voice Input	5,870	12.3		16.0 (>>5,870hrs)
Youtube	1,400	47.6	52.3	

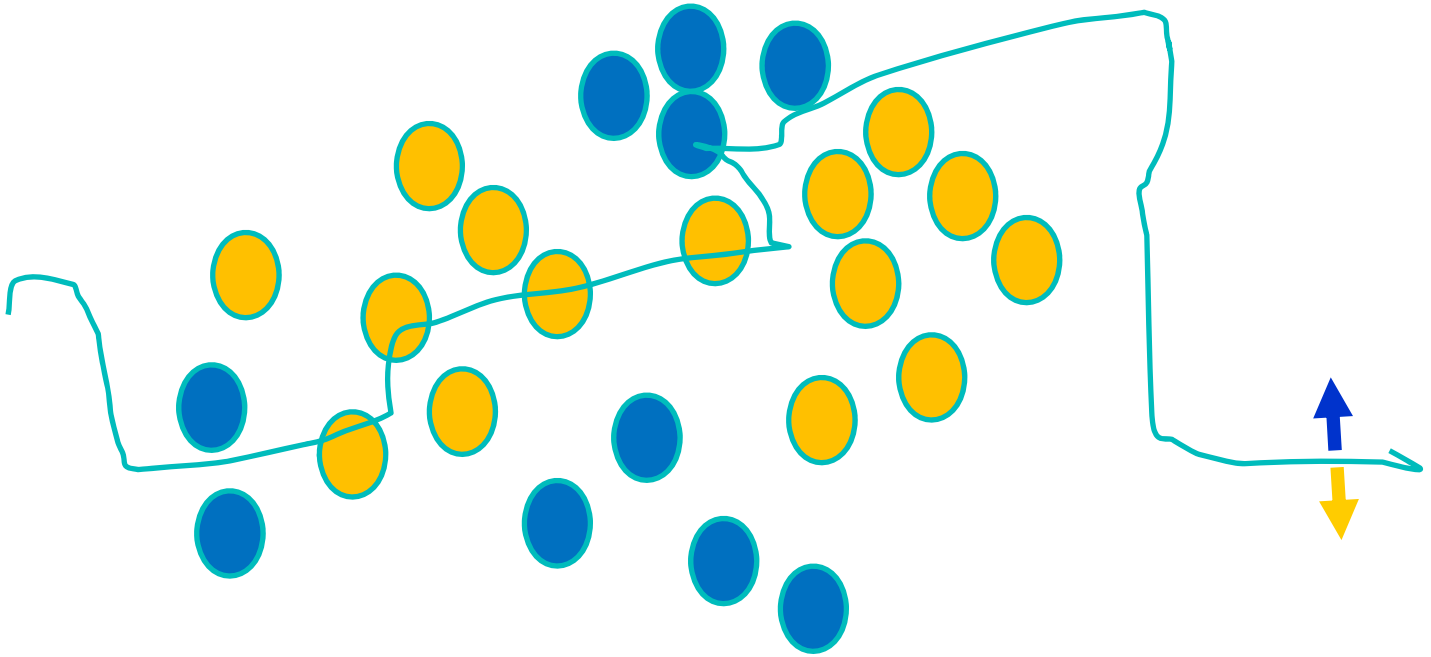
http://yann.lecun.com/exdb/mnist/

http://people.idsia.ch/~juergen/cvpr2012.pdf

Dataset	Best result of others [%]	MCDNN [%]	Relative improv. [%]
MNIST	0.39	0.23	41
NIST SD 19	see Table 4	see Table 4	30-80
HWDB1.0 on.	7.61	5.61	26
HWDB1.0 off.	10.01	6.5	35
CIFAR10	18.50	11.21	39
traffic signs	1.69	0.54	72
NORB	5.00	2.70	46

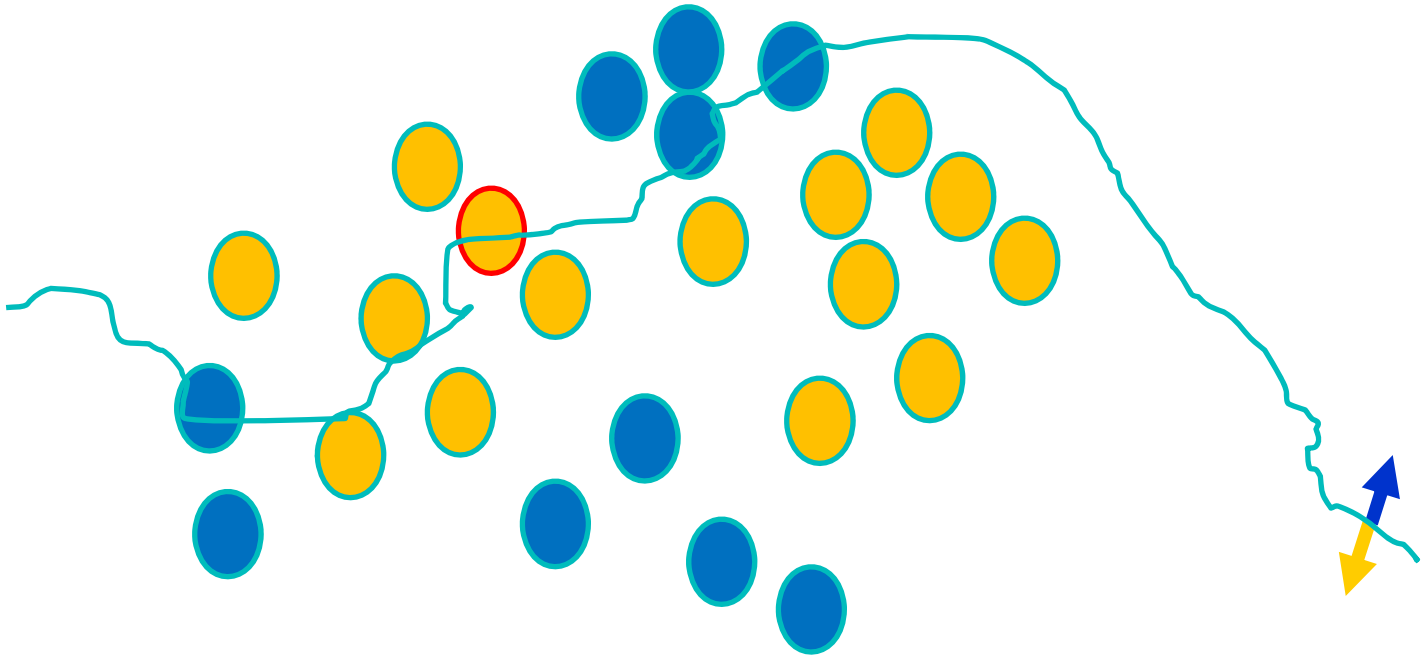
Decision Boundary

Initial random weights



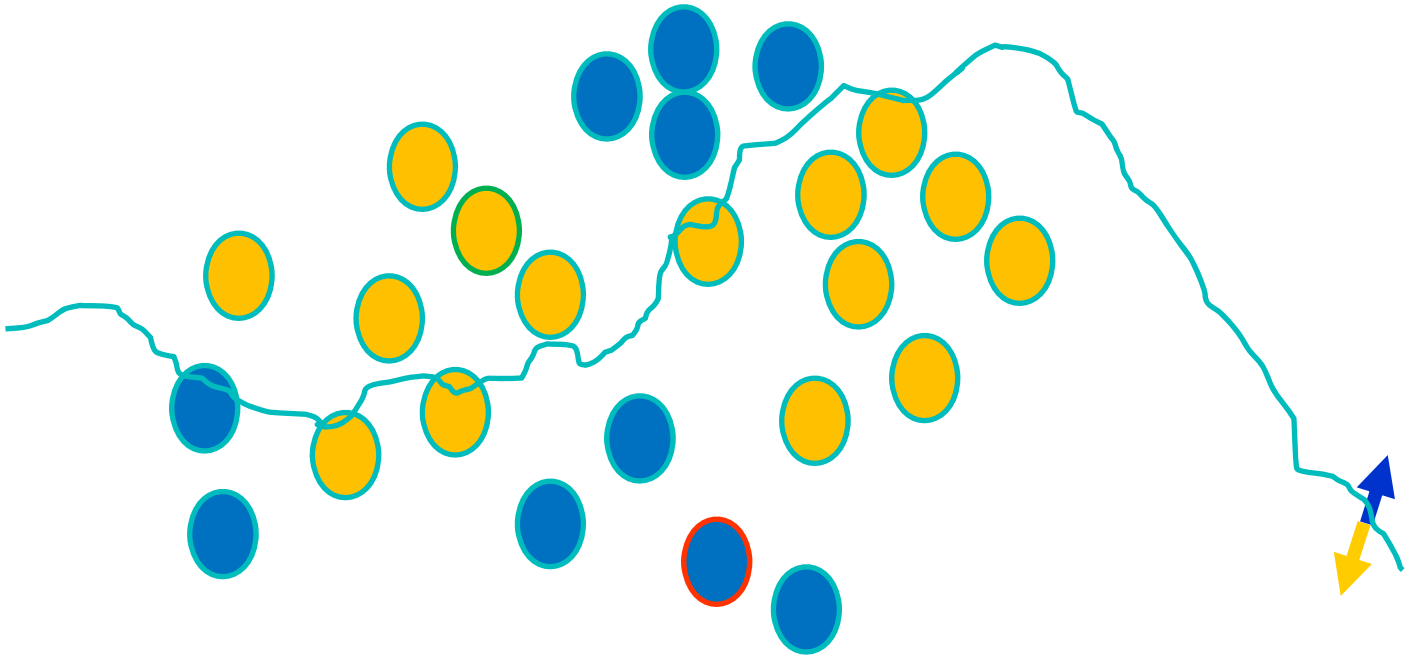
Decision Boundary (cont.)

Present a training instance / adjust the weights



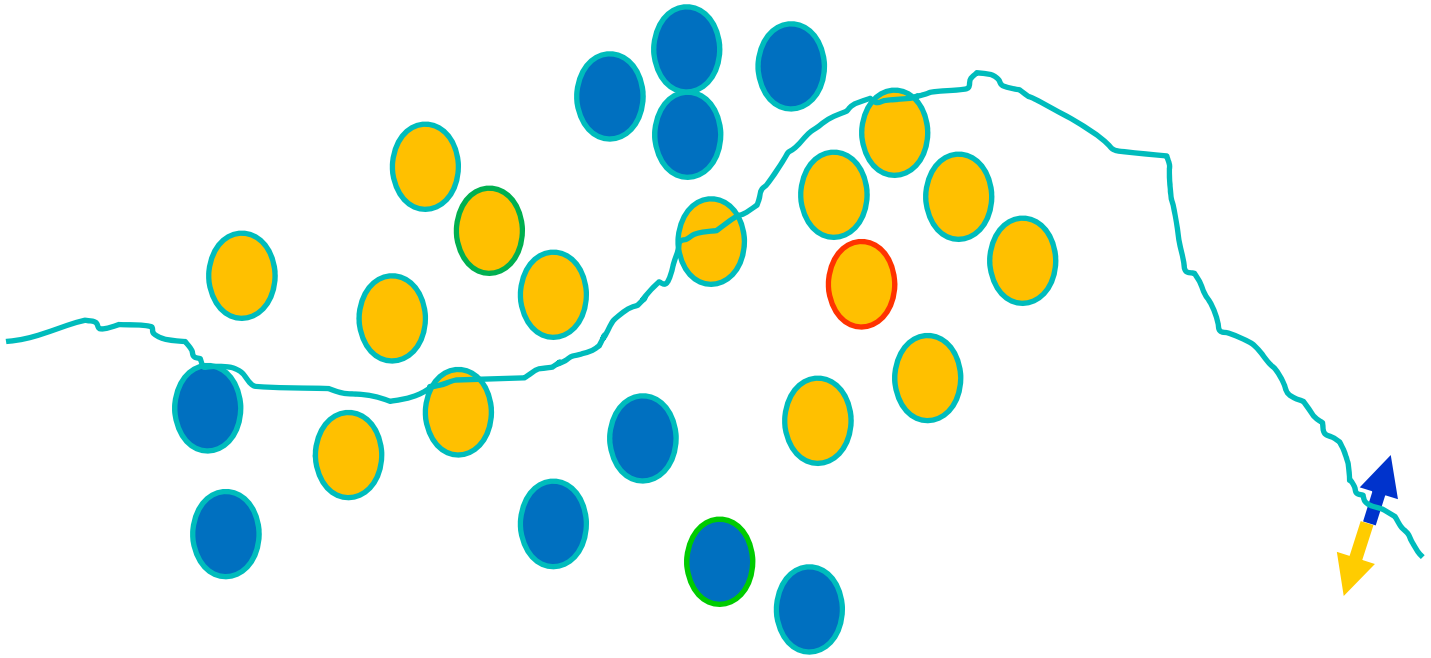
Decision Boundary (cont.)

Present a training instance / adjust the weights



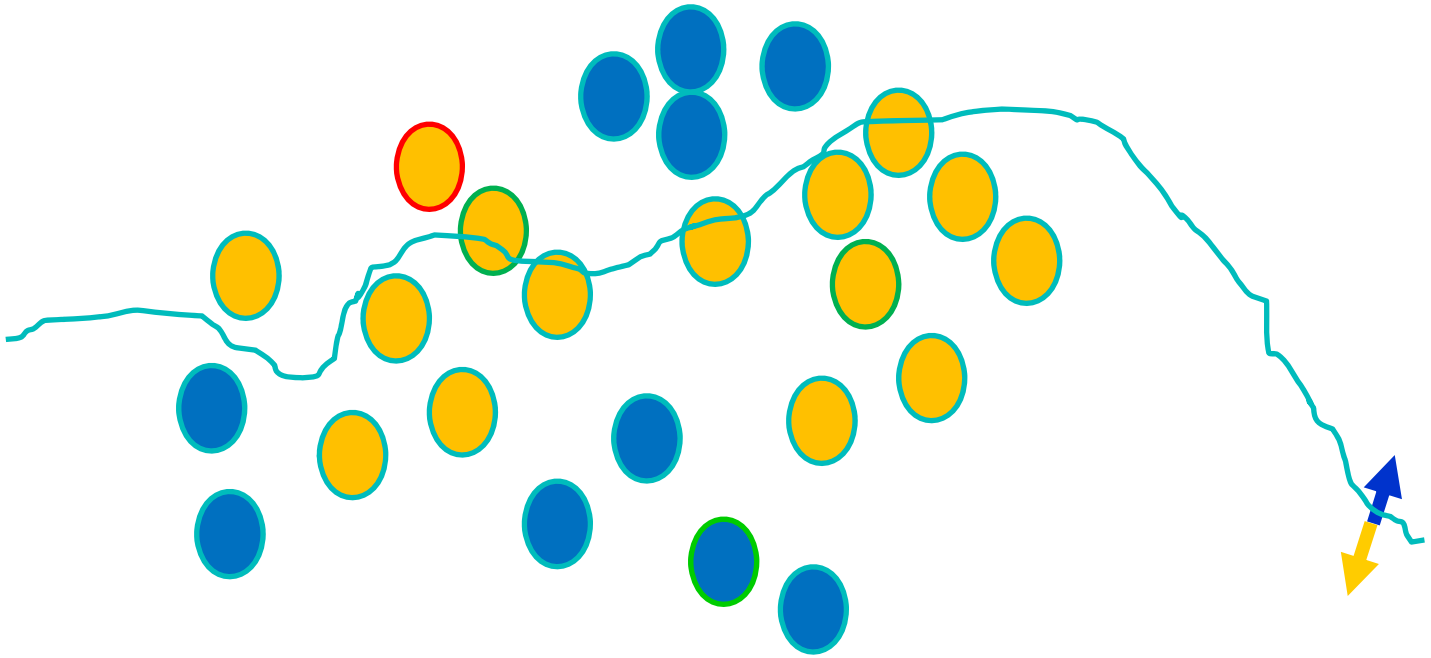
Decision Boundary (cont.)

Present a training instance / adjust the weights



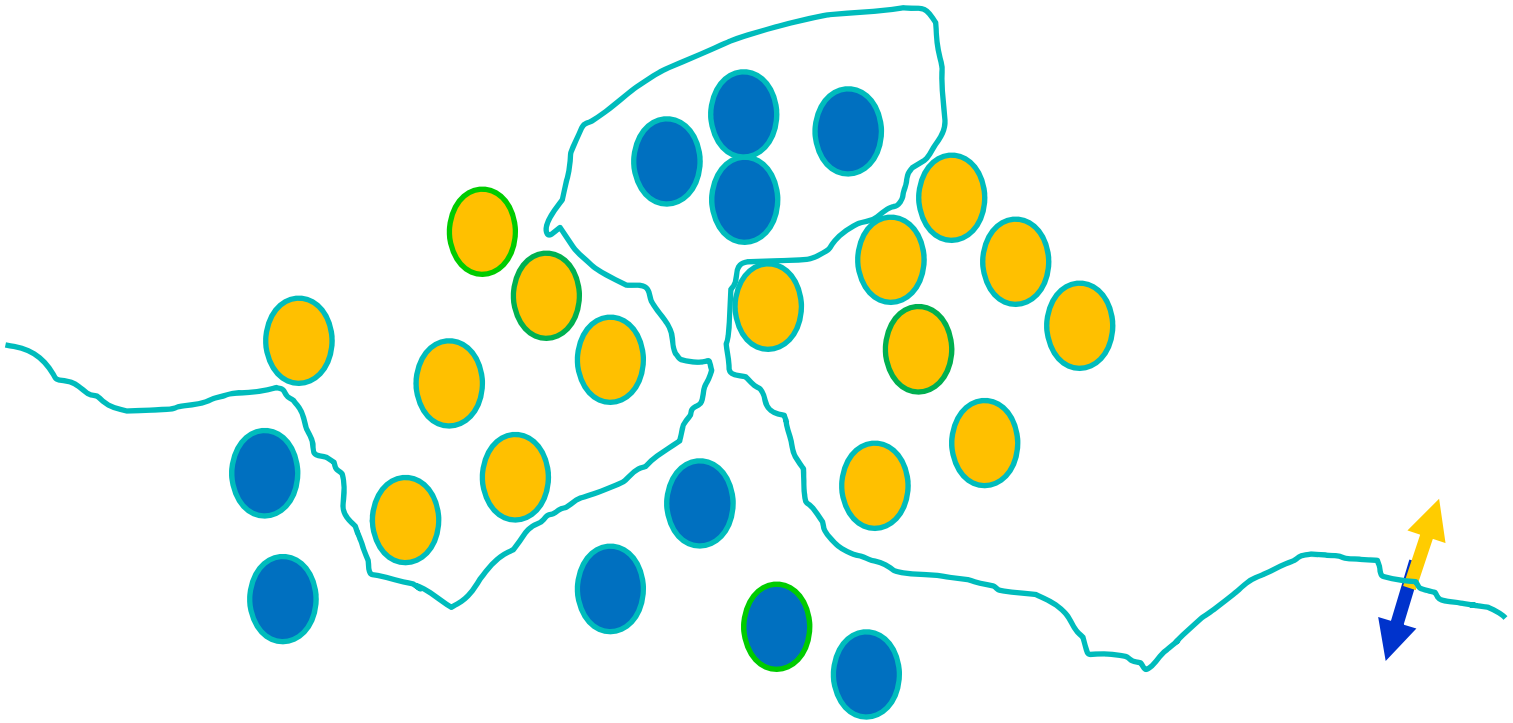
Decision Boundary (cont.)

Present a training instance / adjust the weights

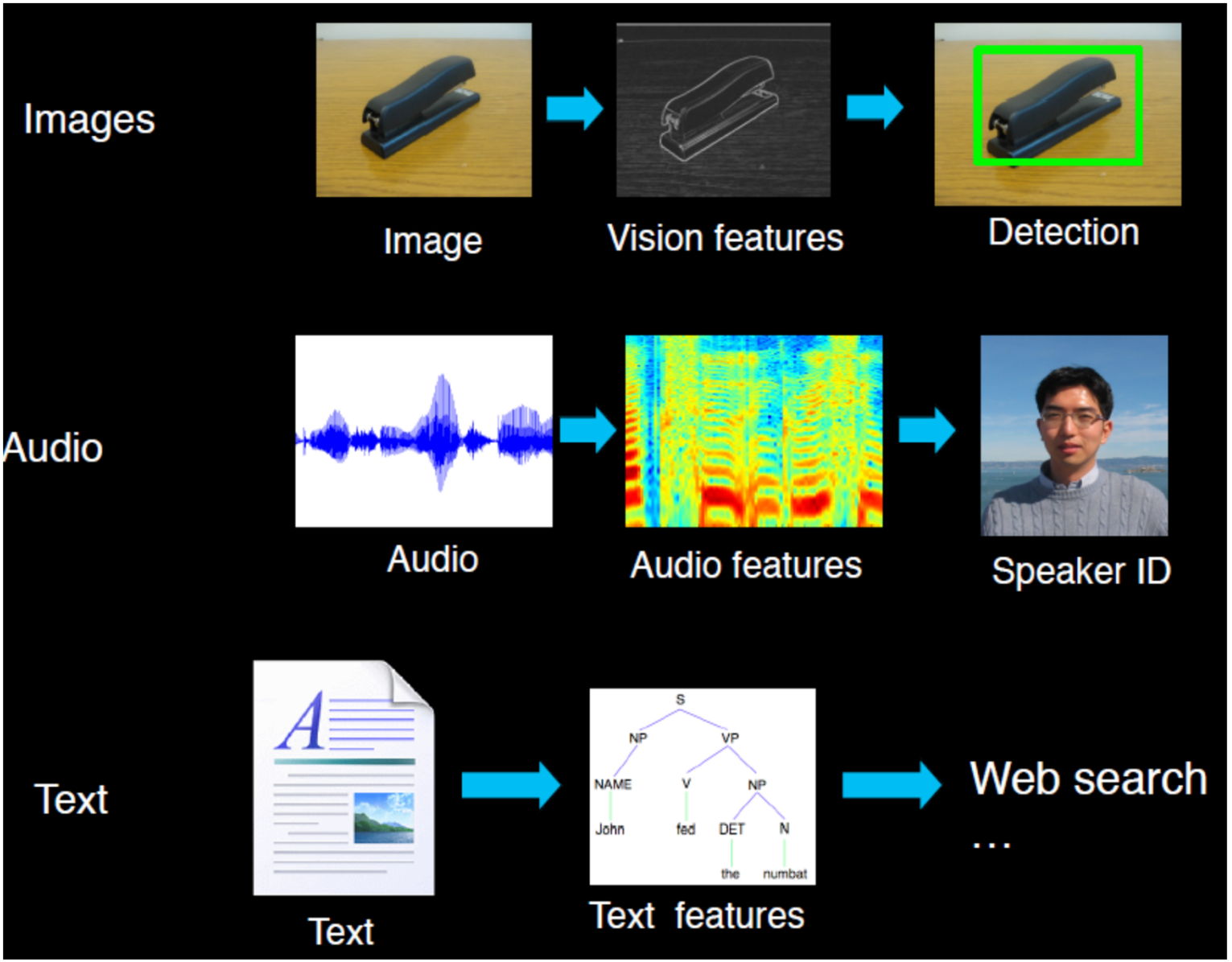


Decision Boundary (cont.)

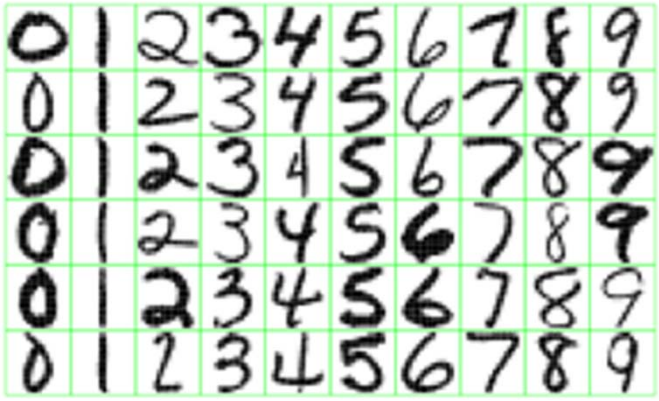
Eventually ….



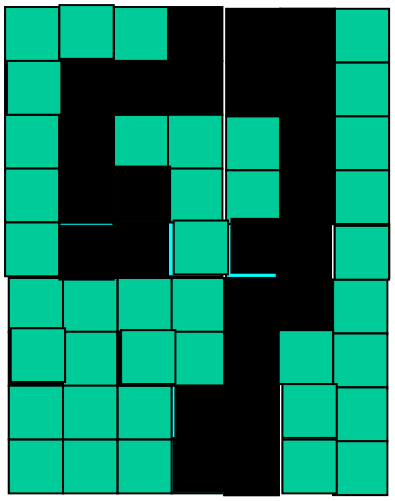
Features -- Learning

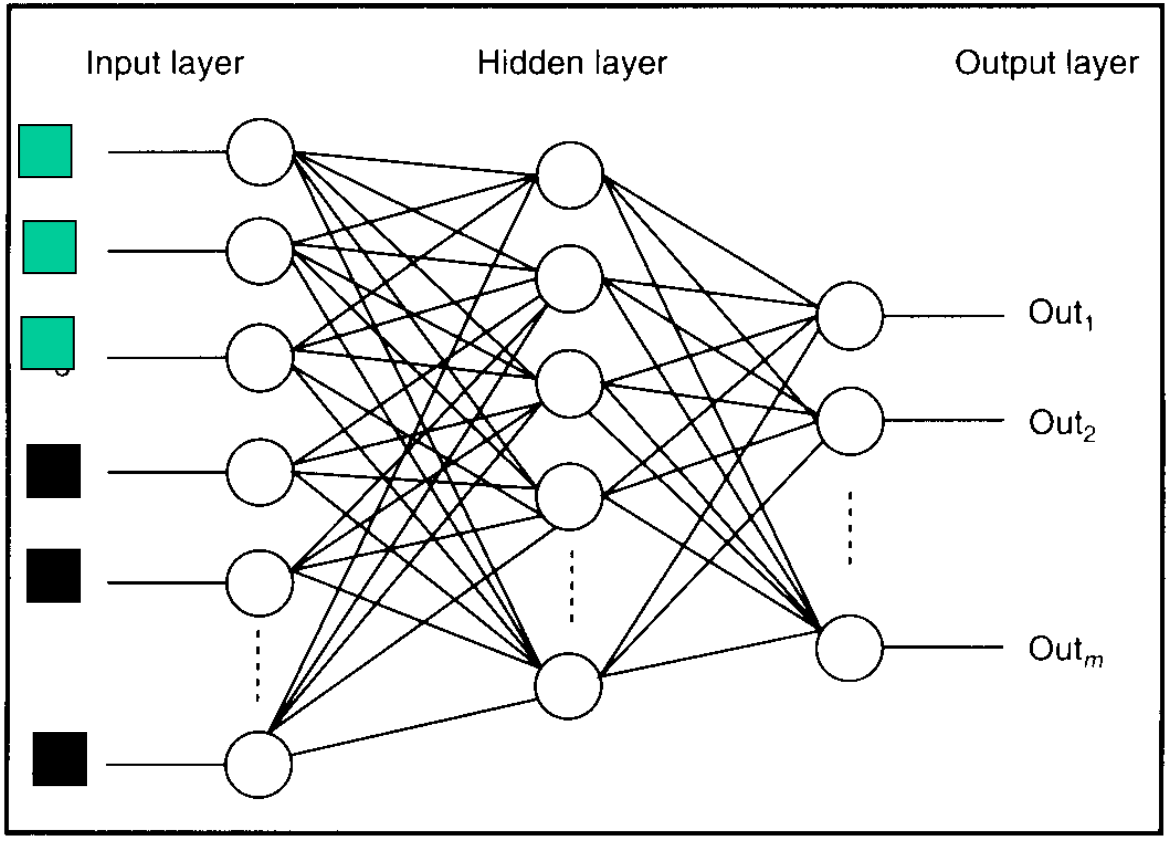


Feature Detectors

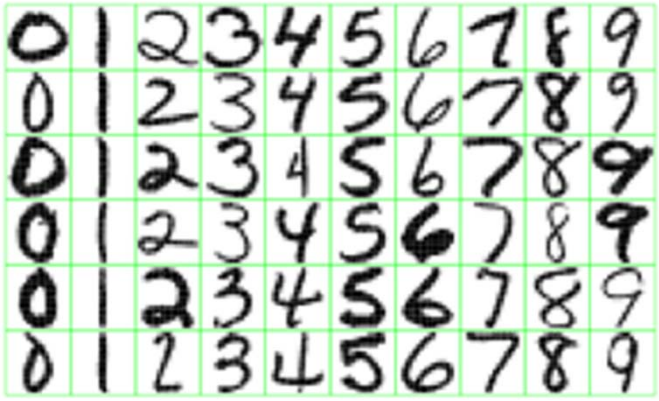


Examples of Handwritten digits from US postal envelopes

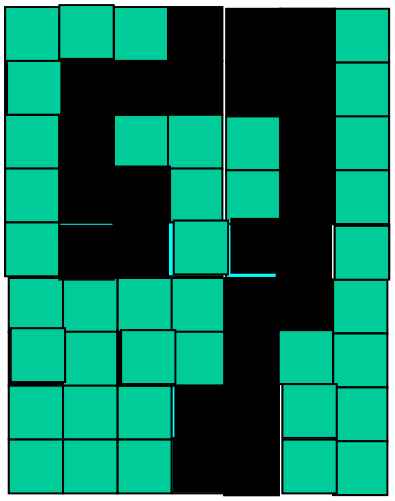


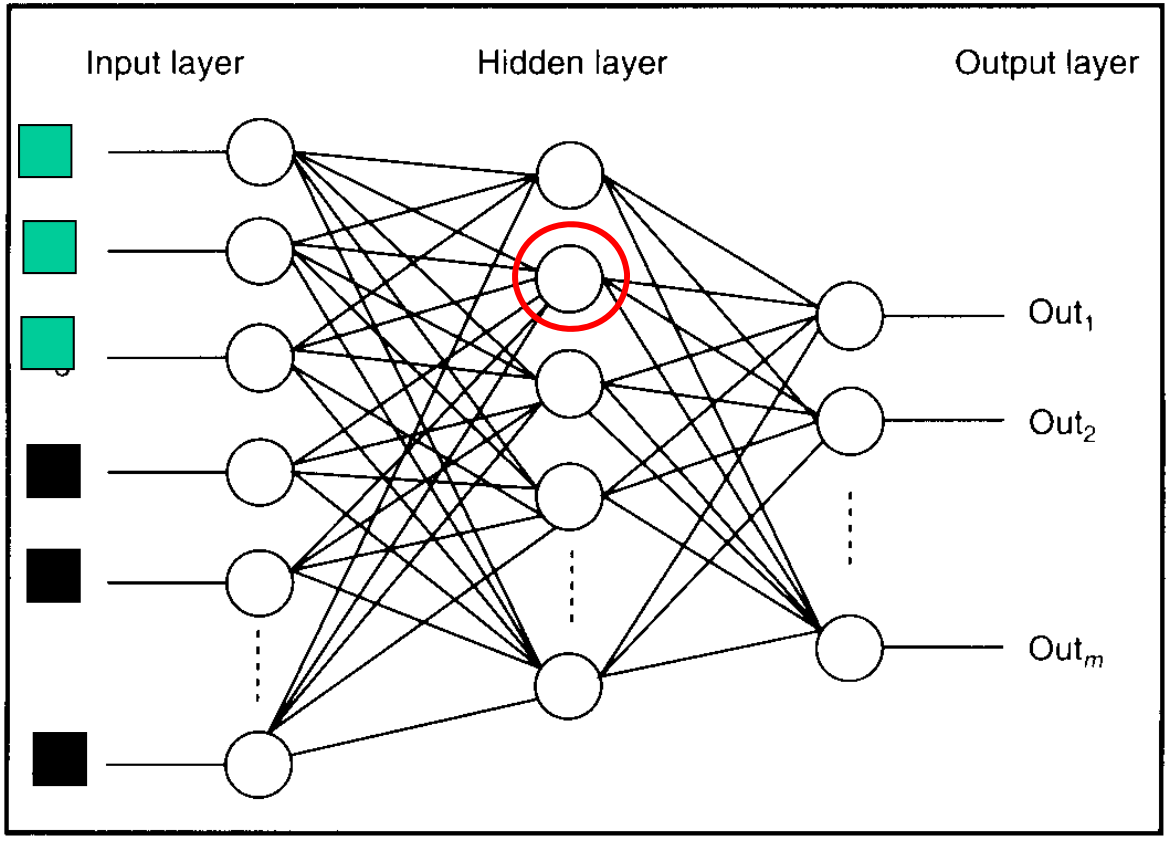


Feature Detectors – What is this unit doing?

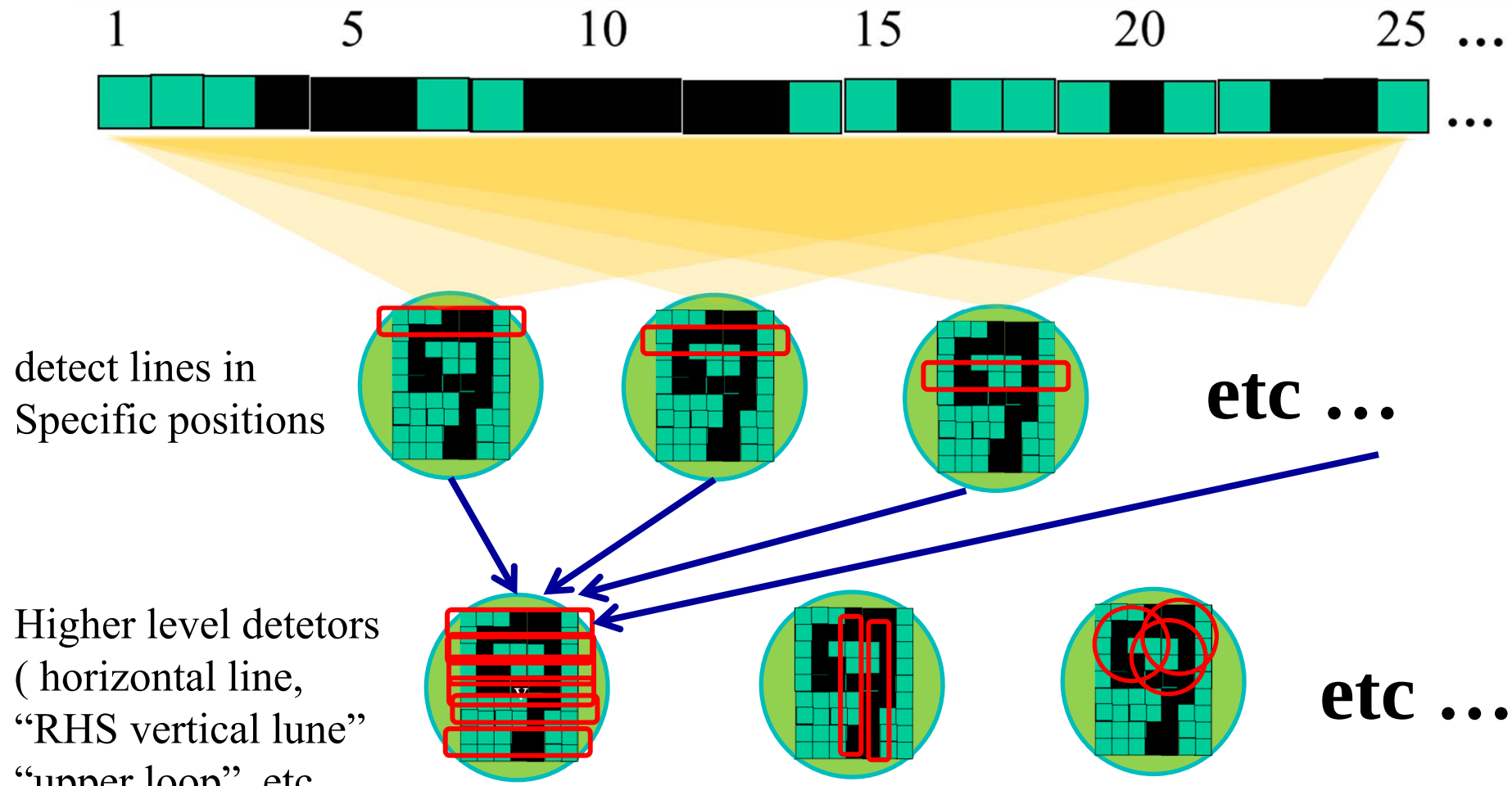


Examples of Handwritten digits from US postal envelopes

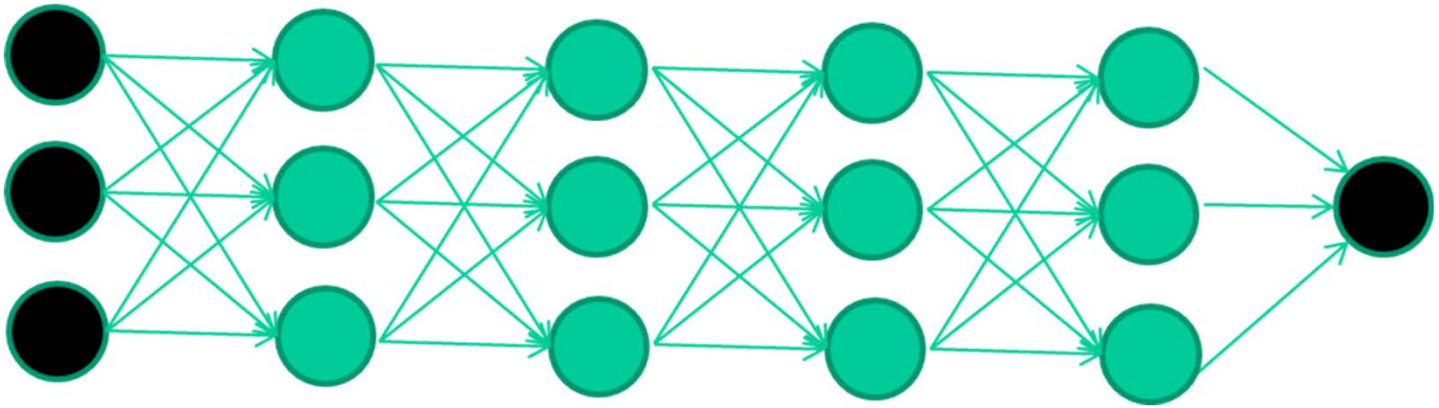




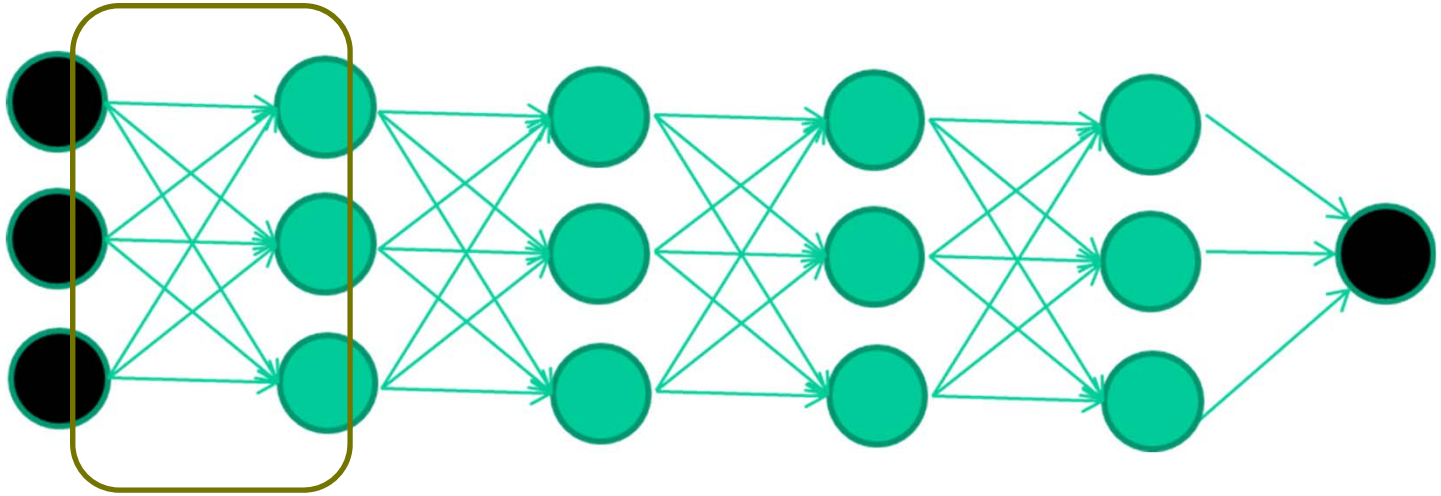
Successive Layers Can Learn Higher-level Features



Deep Learning: New Way to Train Multi-layer NNs

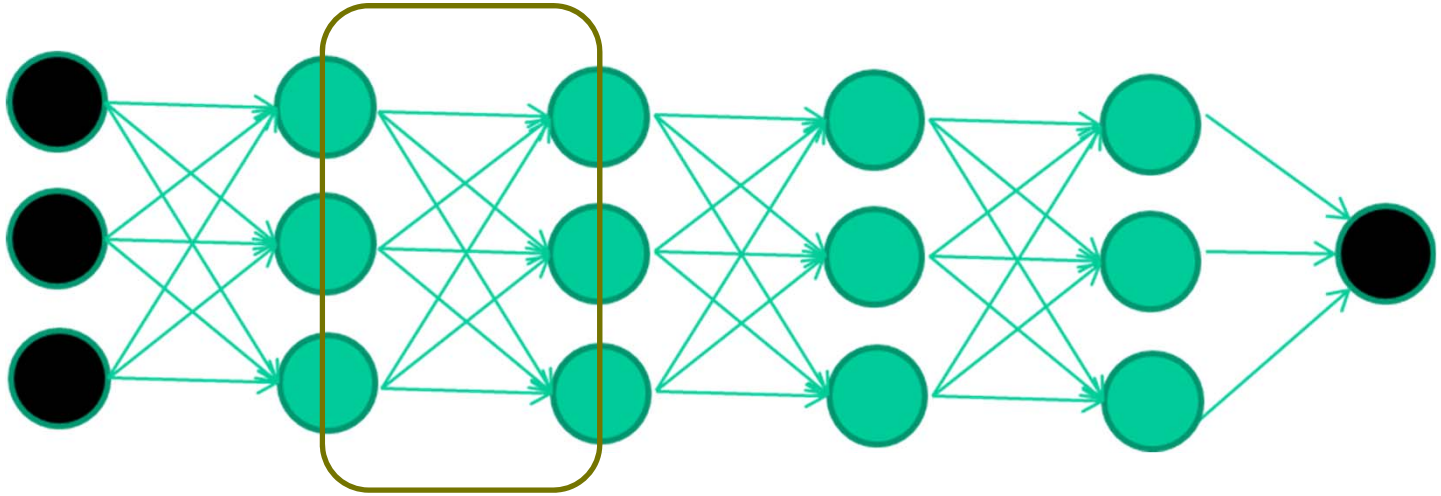


Deep Learning: New Way to Train Multi-layer NNs (cont.)



Train **this** layer first

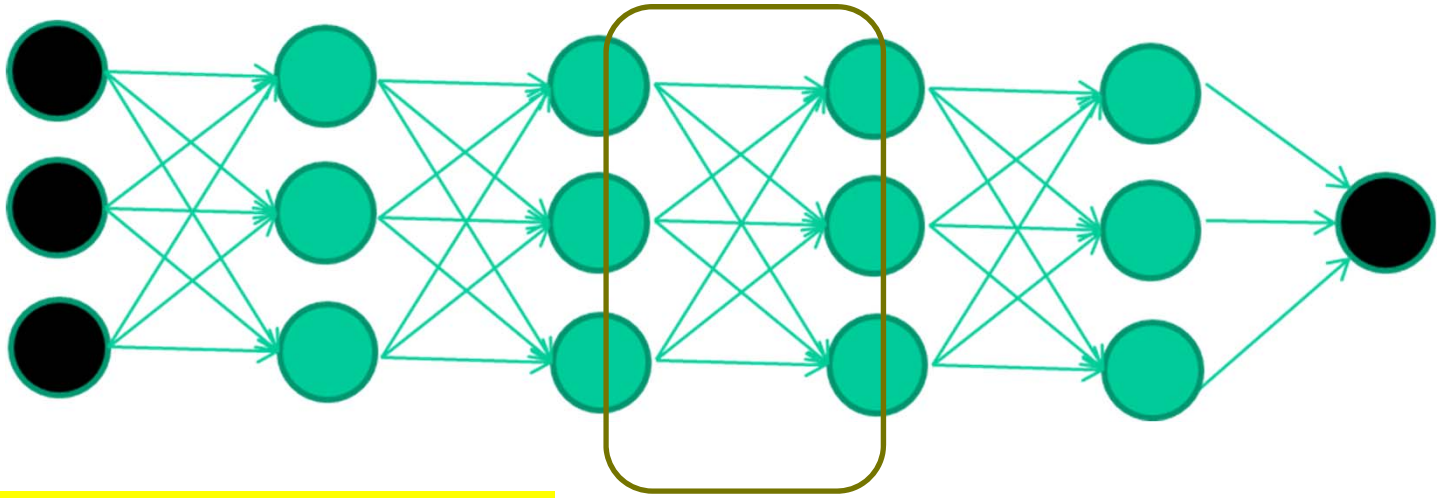
Deep Learning: New Way to Train Multi-layer NNs (cont.)



Train **this** layer first

Then **this** layer

Deep Learning: New Way to Train Multi-layer NNs (cont.)

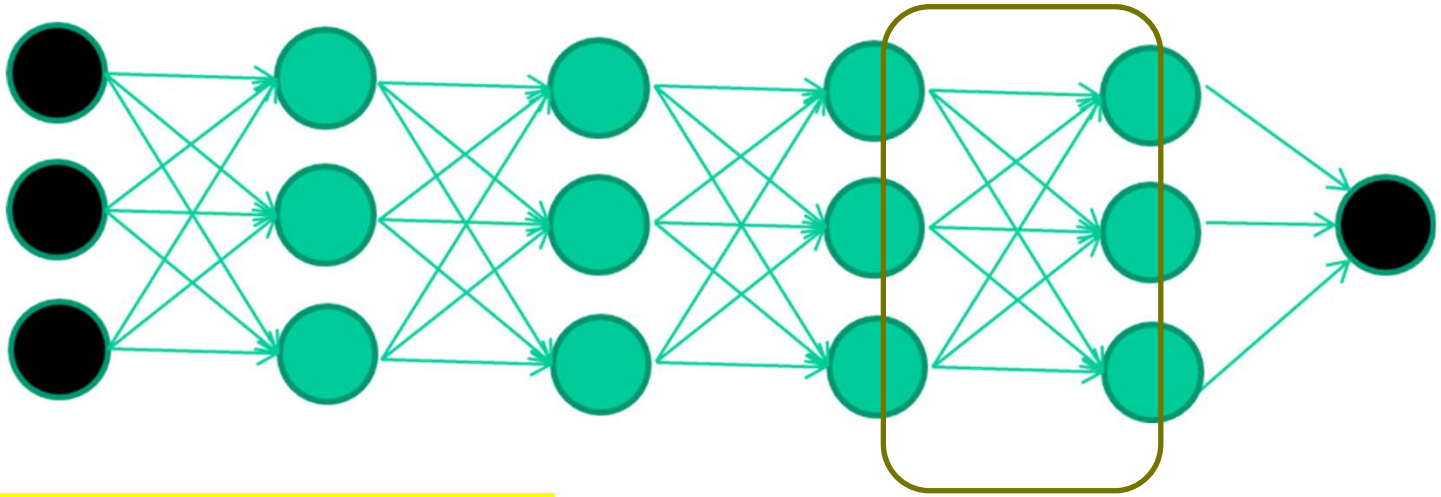


Train **this** layer first

Then **this** layer

Then **this** layer

Deep Learning: New Way to Train Multi-layer NNs (cont.)



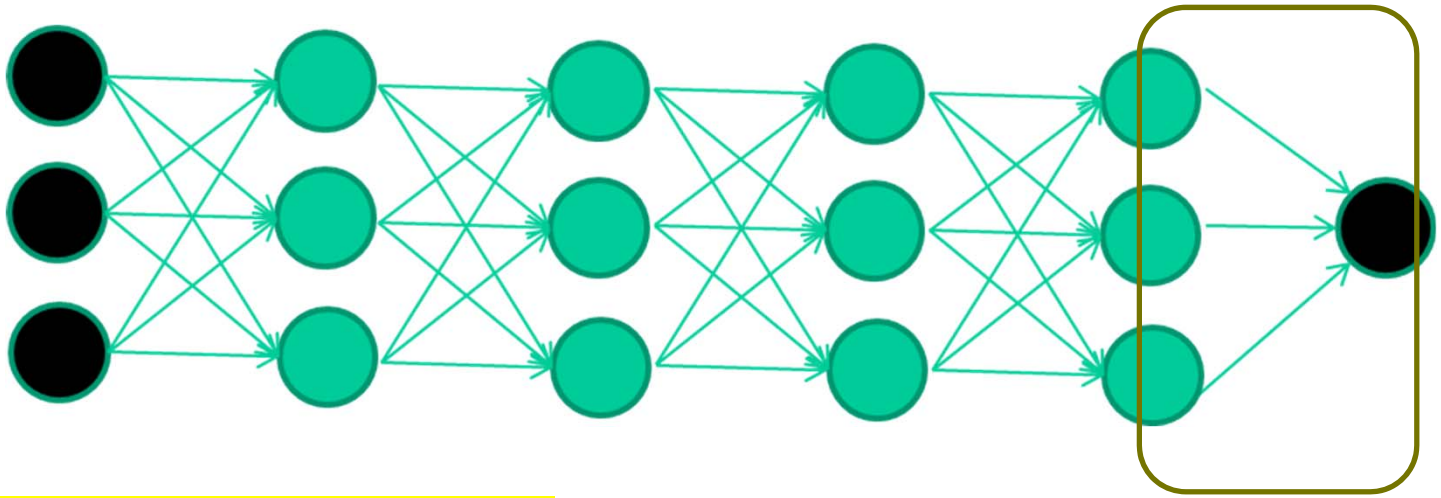
Train **this** layer first

Then **this** layer

Then **this** layer

Then **this** layer

Deep Learning: New Way to Train Multi-layer NNs (cont.)



Train **this** layer first

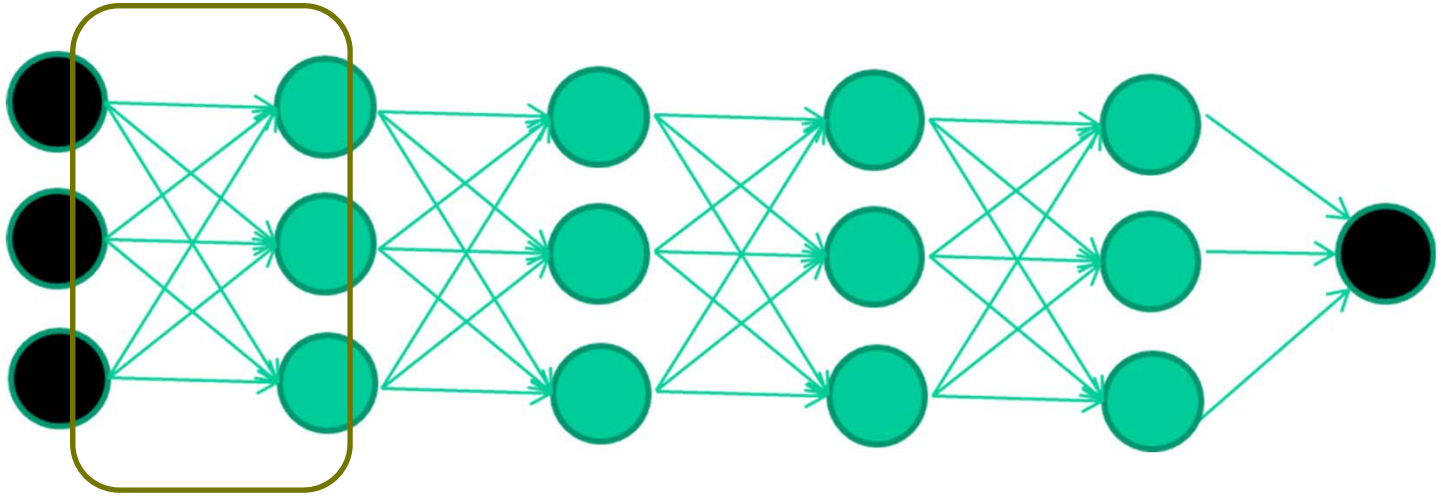
Then **this** layer

Then **this** layer

Then **this** layer

Finally **this** layer

Deep Learning: New Way to Train Multi-layer NNs (cont.)



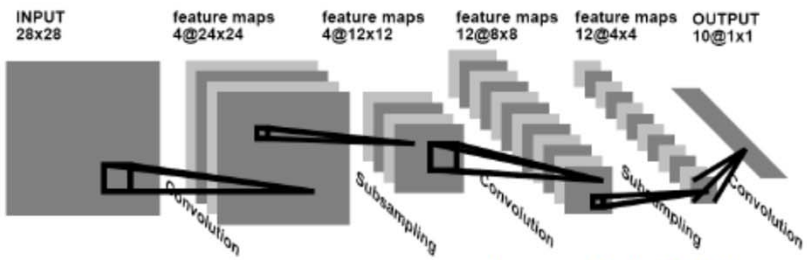
EACH of the (non-output) layers is trained to be an

auto-encoder

Basically, it is forced to learn good features that describe what comes from the previous layer

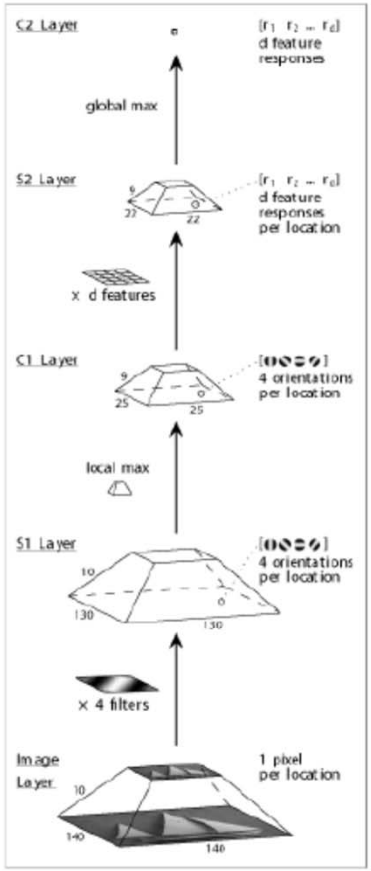
Some Current Deep Architectures

Convolutional Networks (Lecun)

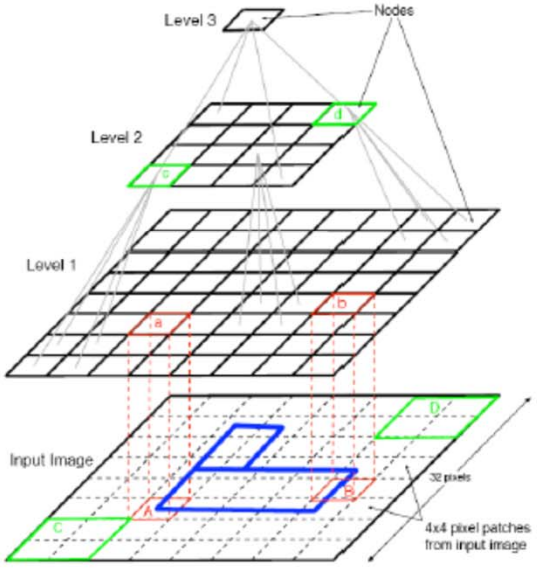


Lecun et al. (1995)

HMAX (Poggio)

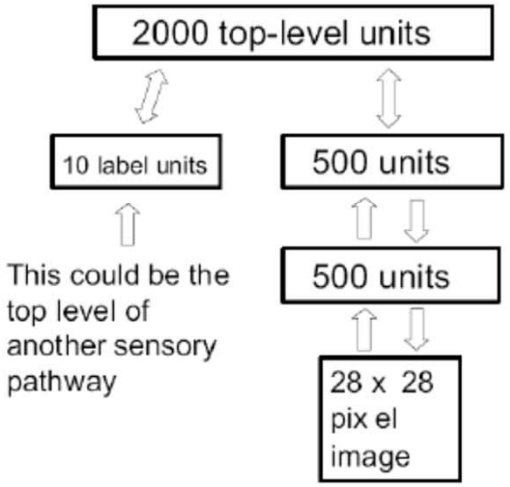


HTM (Hawkins)



Dileep George (2008)

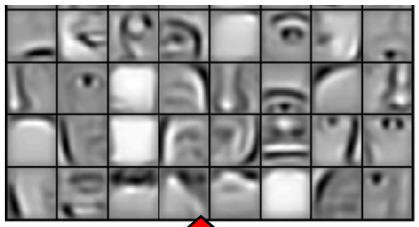
DBNs (Hinton)



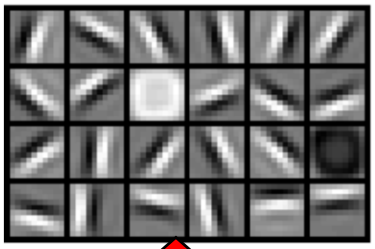
Hinton et al. (2006)

Feature Hierarchies

object models

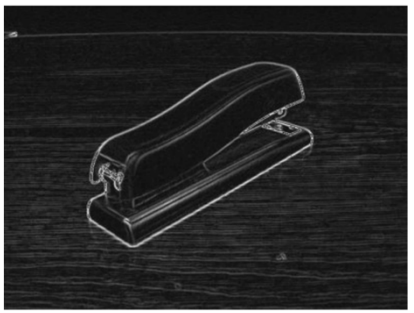


object parts
(combination
of edges)



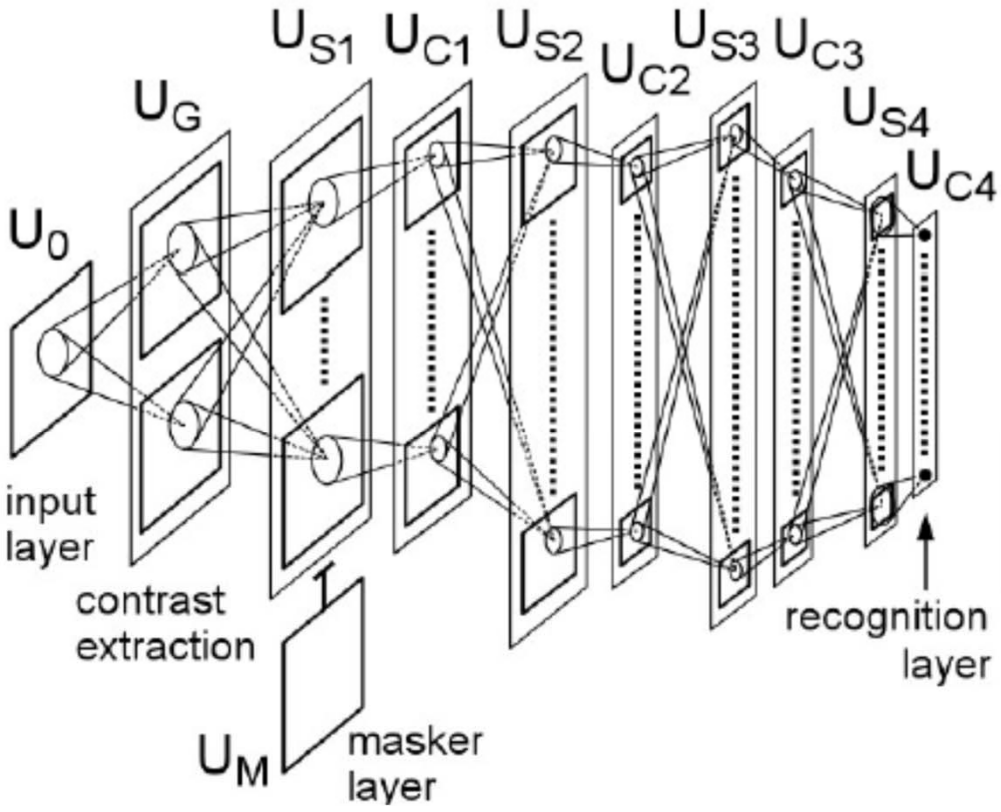
edges

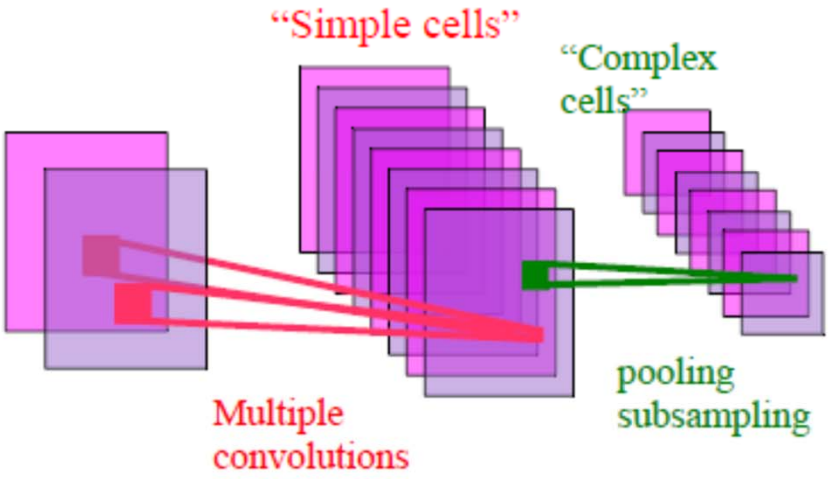
pixels



Early Hierarchical Feature Models for Vision

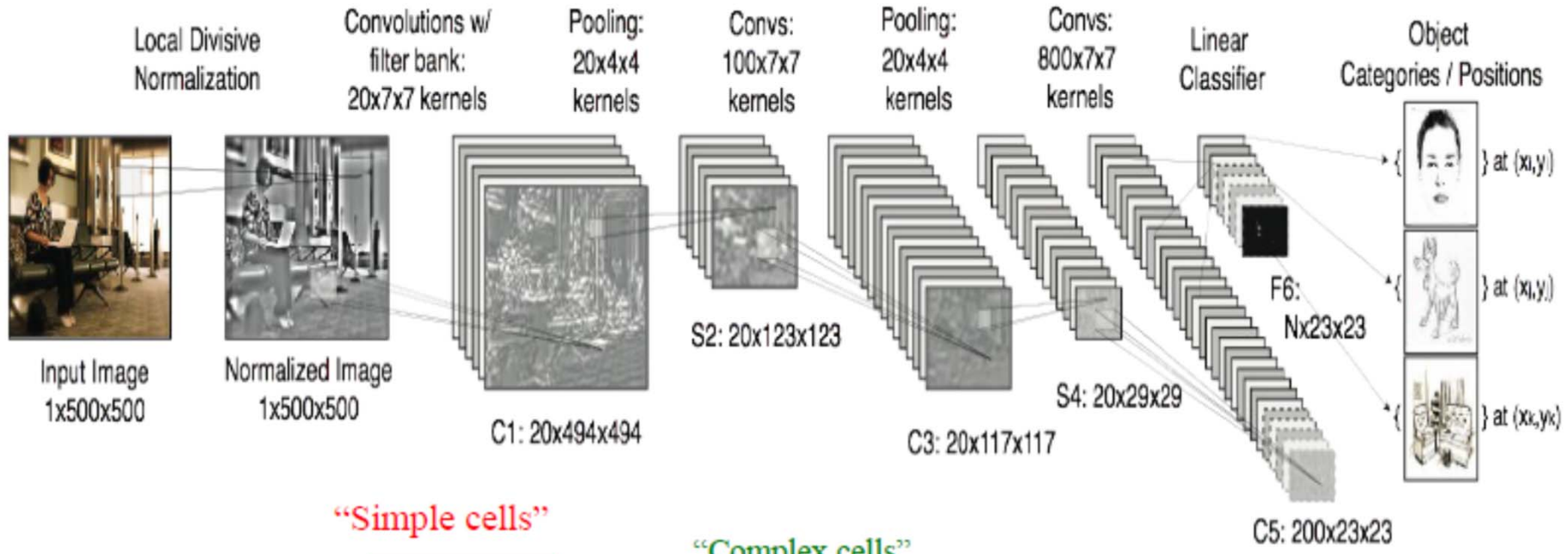
- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

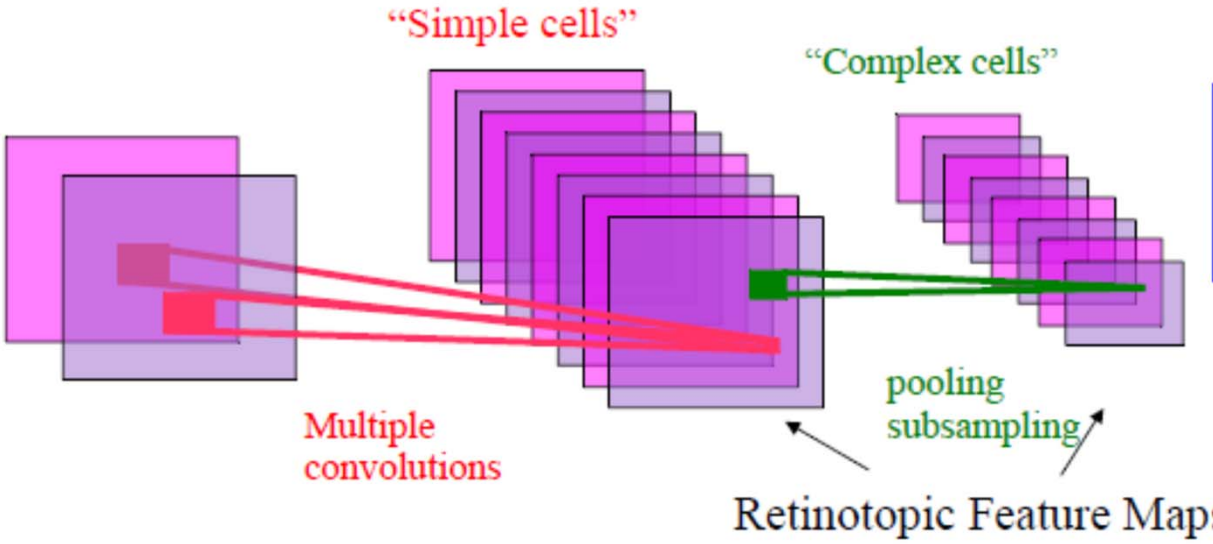




The Convolutional Net Model

Multistage Hubel-Wiesel System





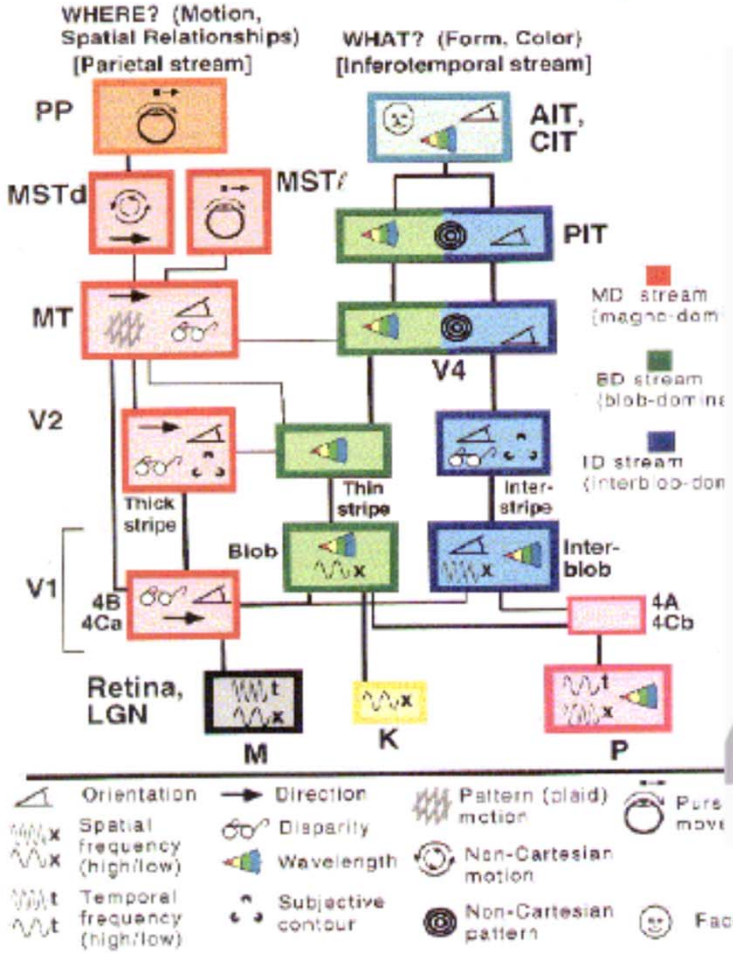
- Training is supervised
- With stochastic gradient descent

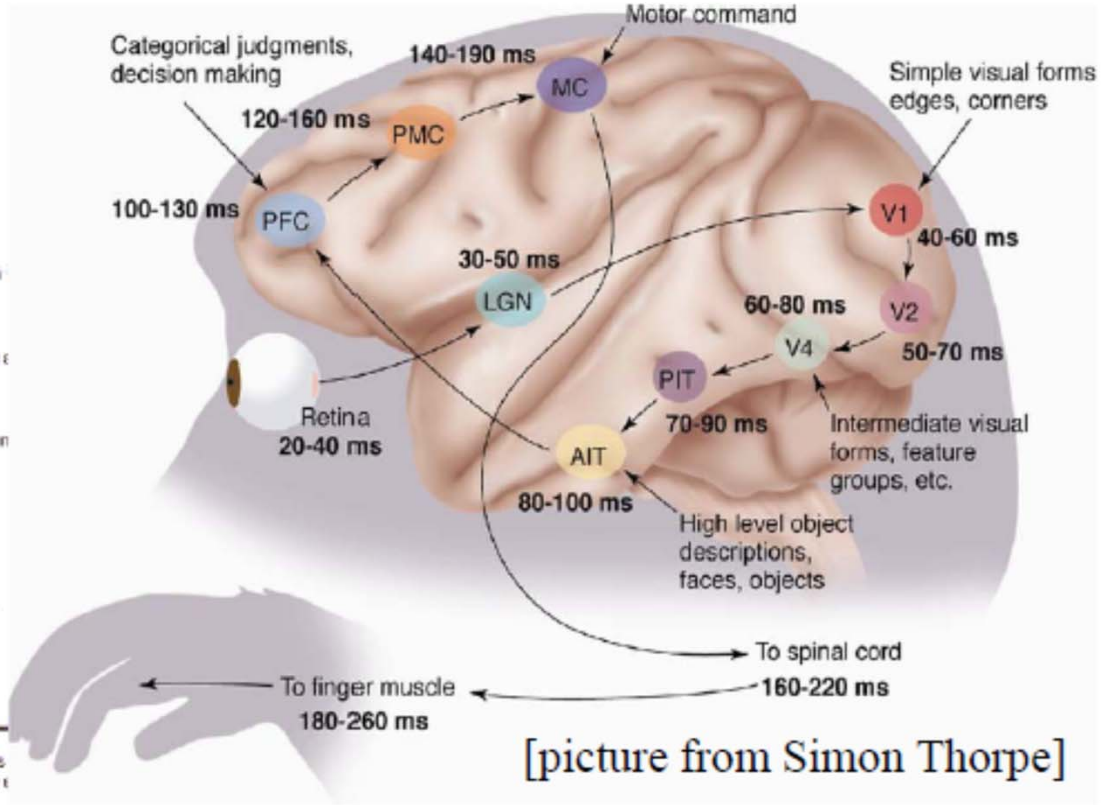
[LeCun et al. 89]

[LeCun et al. 98]

Mammalian Visual Cortex is Hierarchical

- The ventral (recognition) pathway in the visual cortex has multiple stages
- Retina - LGN - V1 - V2 - V4 - PIT - AIT ....
- Lots of intermediate representations

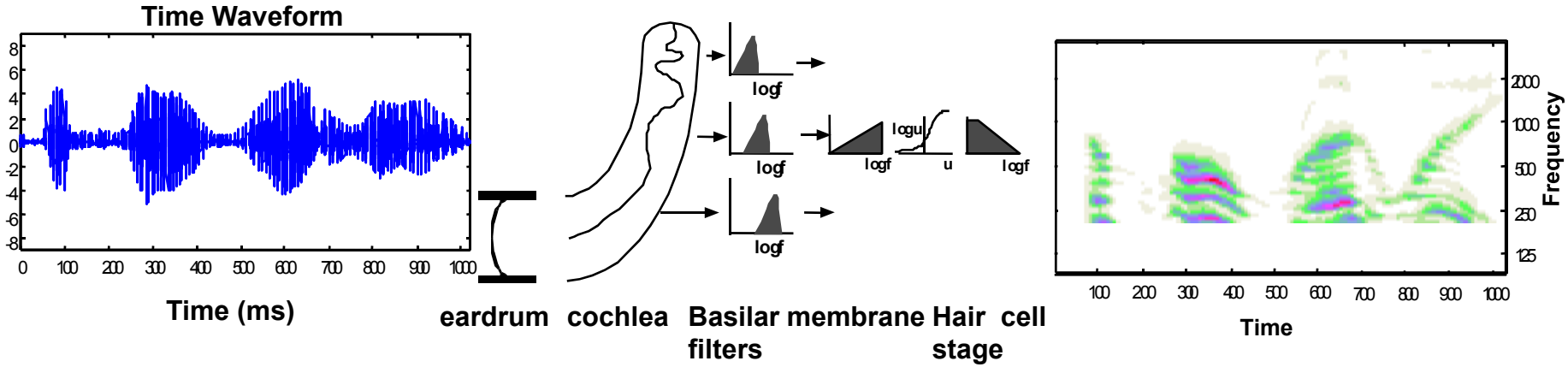




[Gallant & Van Essen]

Multiresolution Preprocessor: Auditory Filtering (Shamma et al)

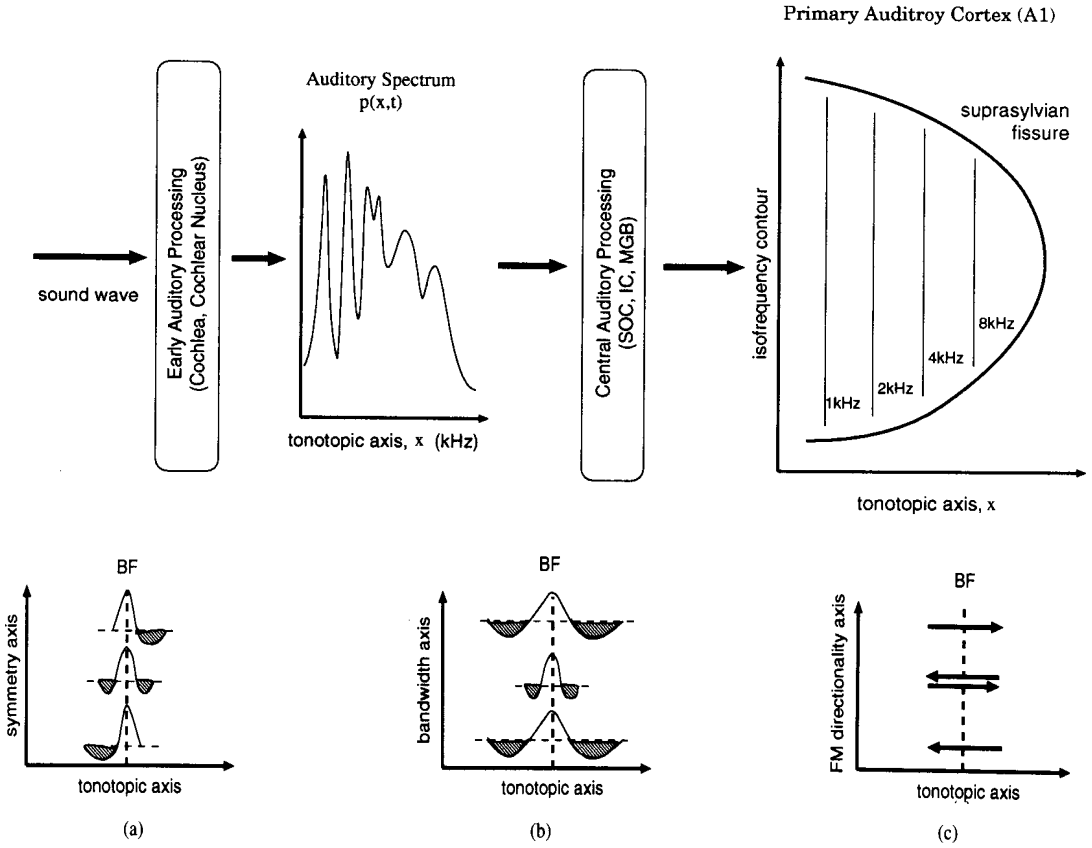
Two auditory filters, motivated and designed according to acoustic physiology and acoustic cortex models, were used to compute the timbre spectrogram of one particular subframe in each frame



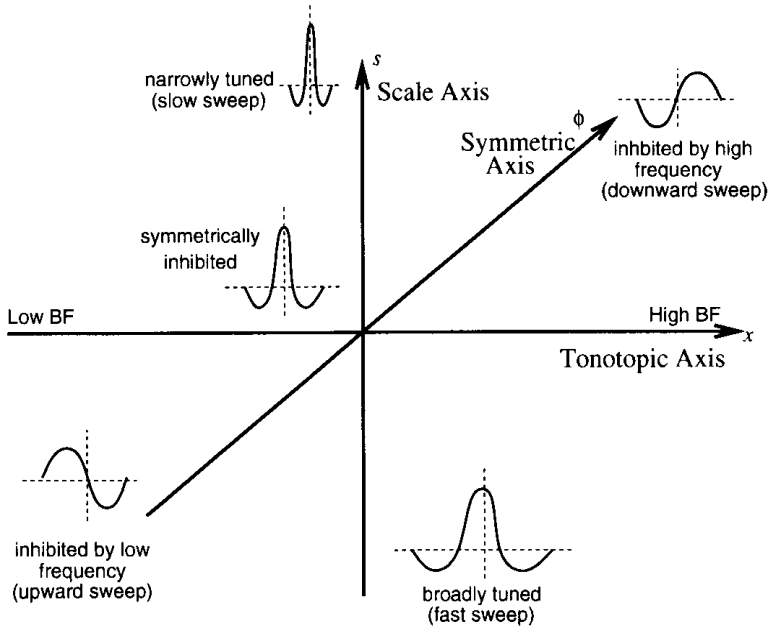
- **The first filter mimics the action of the inner ear**
- **Computes the spectrogram of the sound sample, and performs various nonlinear operations, which models the nonlinear fluid-cilia couplings and ionic channels of conduction**

(Wavelet Transform)

Spectro-Temporal Processing of Sound (Shamma et al)



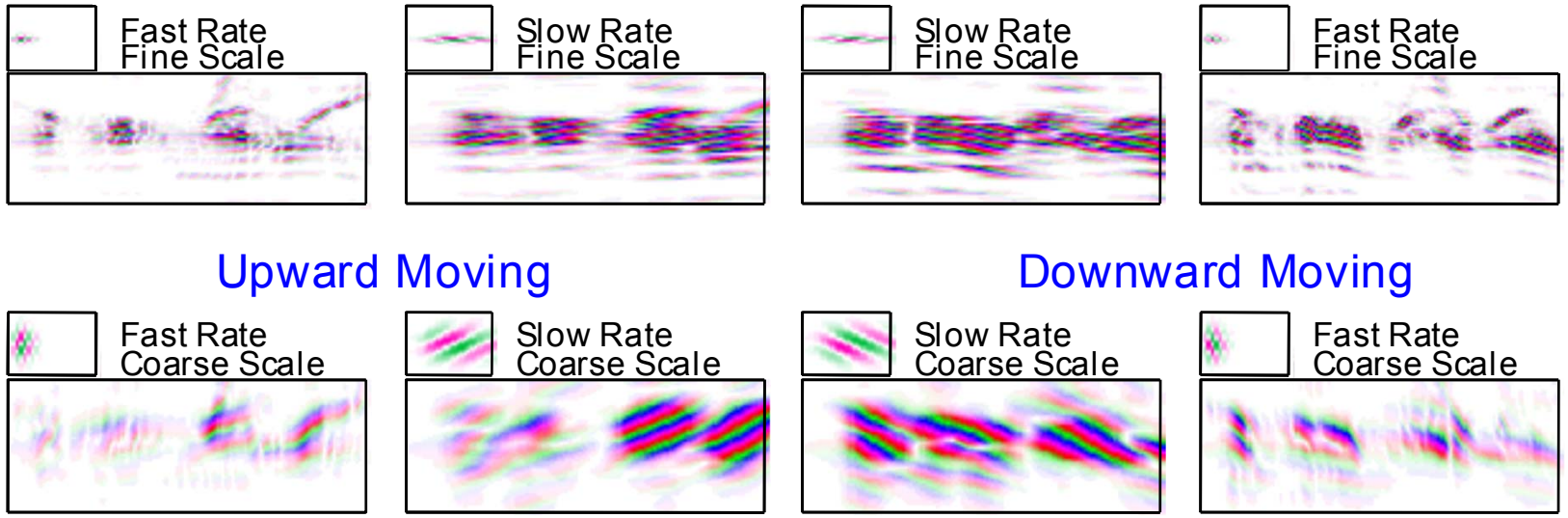
[3-D virtual cortical model: Tonotopic axis (BF), scale axis (bandwidth), phase axis (asymmetry)]



- **Tonotopic axis: best frequencies (BF) of filter bank along the cochlea (log scale)**
- **Auditory spectrum: short-time, self-normalized power spectrum (x is the BF)**
- **1-D pattern virtually mapped to at least 3-D pattern in cortex**
- **Cell tuning along isofrequency contours varries: symmetry, bandwidth and FM selectivity**

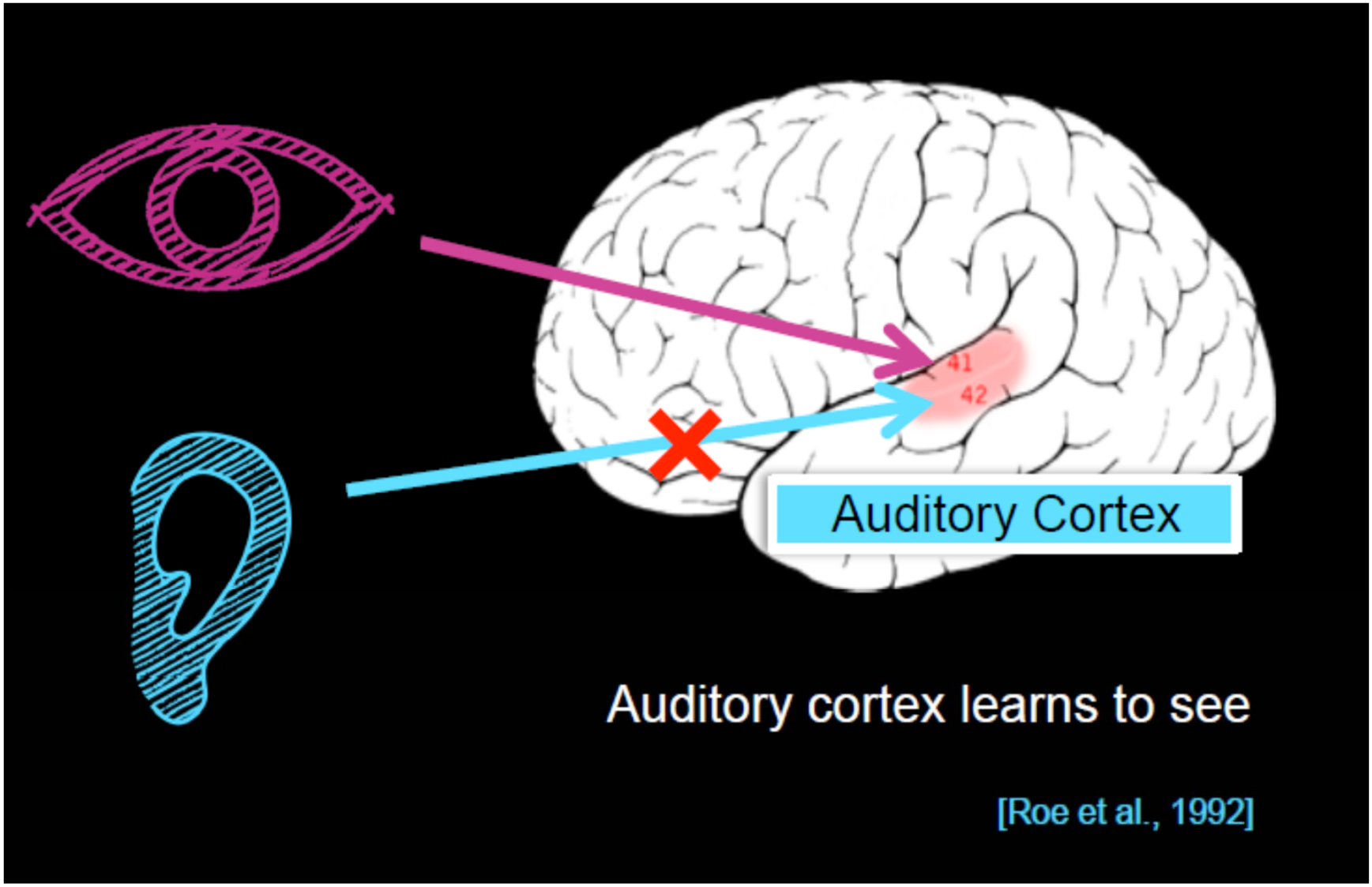
Multiresolution Preprocessor: Auditory Filtering

Multiresolution cortical filter outputs

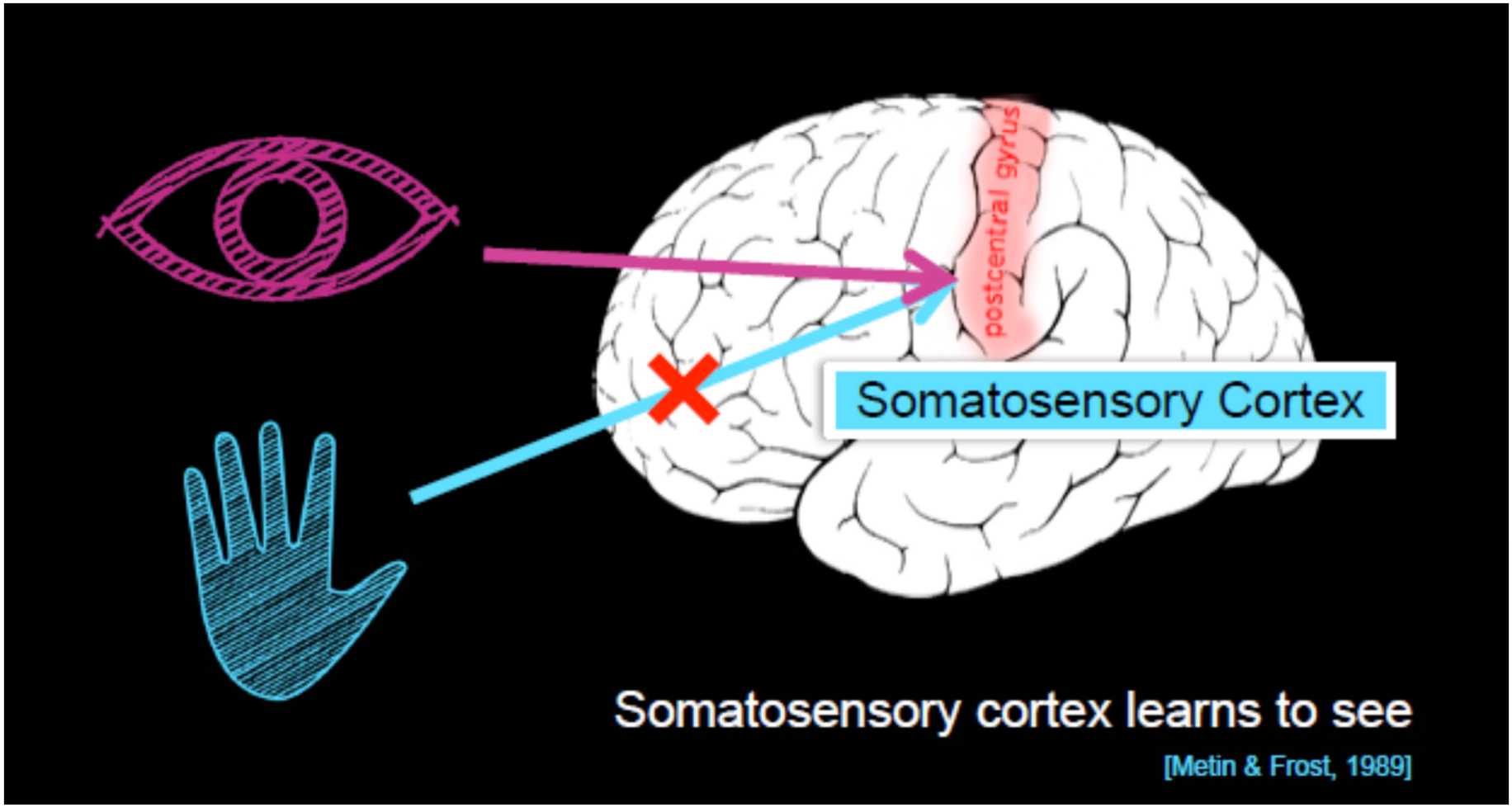


- **The second filter models the multiscale processing of the signal that happens in the auditory cortex**
- **A Ripple Analysis Model, using a ripple filter bank, acts on the output of the inner ear to give multiscale spectra of the sound timbre (Wavelet Transform)**

"One Learning Algorithm" Hypothesis



"One Learning Algorithm" Hypothesis (cont.)



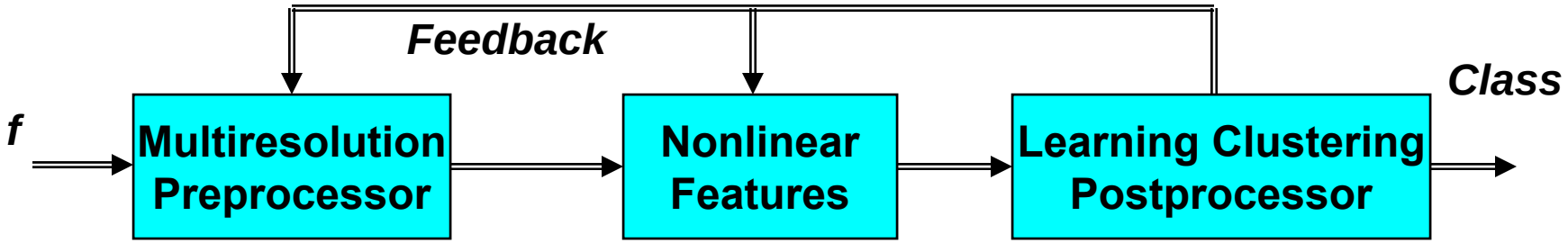
Progressive Classification

- Small amounts of information in the form of a coarse approximation of the signal, are used first to provide partial classification
- Progressively finer details are added until satisfactory performance is obtained
- Approach results in a scheme where:
  - Small amounts of computation are used initially (at coarse level)
  - Additional computations (more detailed) are performed as needed
- Approach leads to:
  - Faster classification algorithms (faster search)
  - Algorithms that preserve high fidelity in the search (the challenge)
  - Easily parallelizable algorithms

Motivation and Applications

- Classification of image data and the recognition of objects in images
  - Formidable difficulties due to:
    - size of image databases
    - lack of systematic procedures for data compaction for recognition
    - multitude of situations where images appear in practice
- ATR based on sensor data from pulsed radar (PR), Doppler radar (DR), synthetic aperture radar (SAR), inverse synthetic aperture radar (ISAR) millimeter--wave (MM--wave) radar, LADAR, and FLIR
  - Key step for high performance ATR is the construction of efficient target models which result in significant search speed-up and memory reduction
- Face recognition
- Text-independent robust speaker identification
- Speaker-independent speech recognition
- Acoustic signal recognition (e.g. fault identification in gearboxes and bearings, ground vehicle identification from array microphones)
- Image understanding and object recognition in vision systems

Multiresolution and Learning Clustering



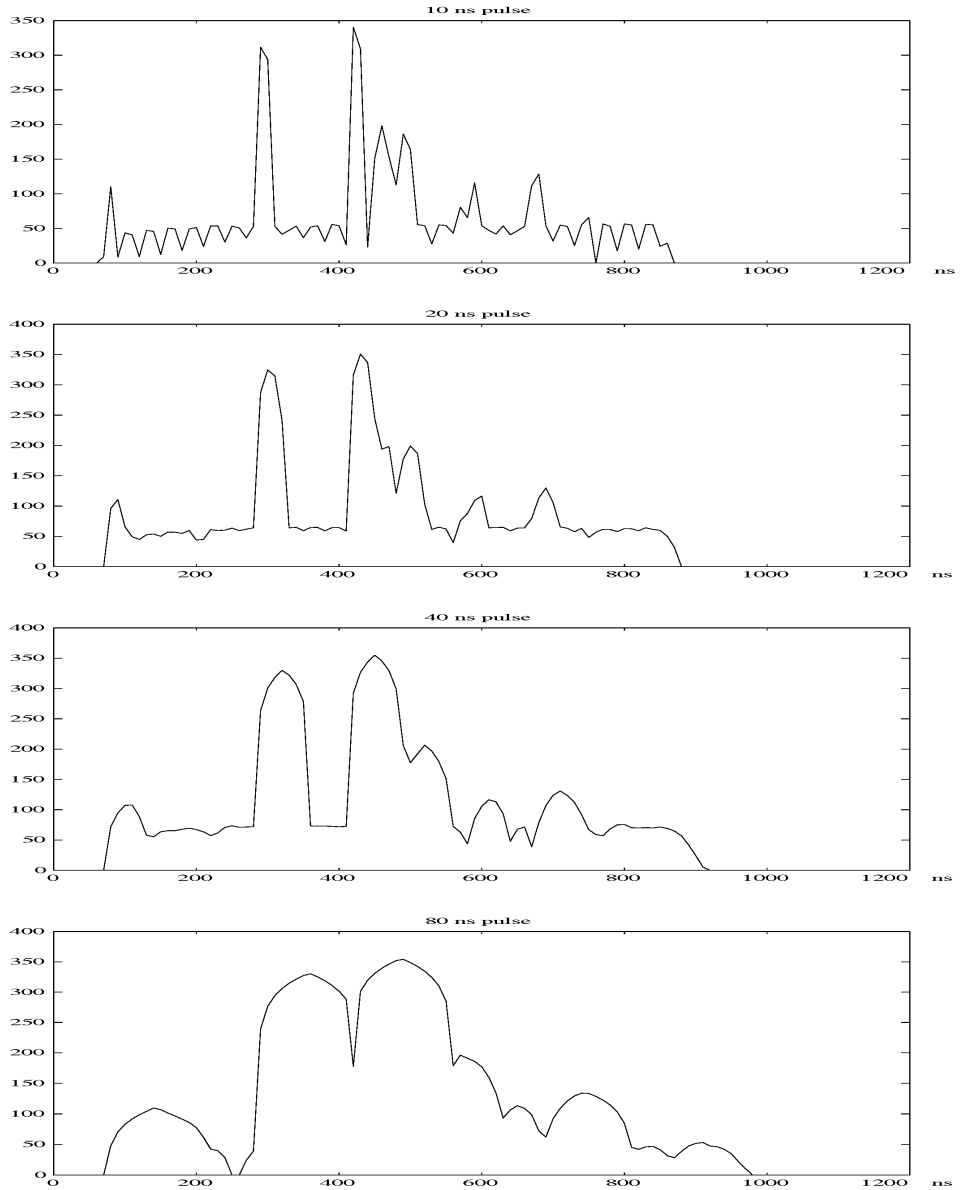
- Address both the hierarchical organization of signal databases and progressive classification:
  - Combine a multiresolution preprocessor with a learning clustering postprocessor
  - Feedback is also an option
- Resulting algorithms proved to have some "universal" qualities
- Found analogs of such algorithms in animals and humans:
  - Hearing and sound classification
  - Vision and identification of objects by humans
- Most promising mathematical formulation of the problem:

  combined compression and classification for general signals

An Example: ATR Based on Radar Returns

- High resolution radar returns contain substantial information for the identification of complex targets (targets with many scatterers)
- Can store many returns indexed by: aspect, elevation, pulsewidth etc.
- Creates need for huge memory to store data and slows search for matching
- Critical need: Develop extremely efficient representaions of high resolution radar data which result in fast and high performance identification
- Our key idea:
  - Organize signal database hierarchically based on resolution (from coarse to fine)
  - Develop progressive identification/classification algorithms

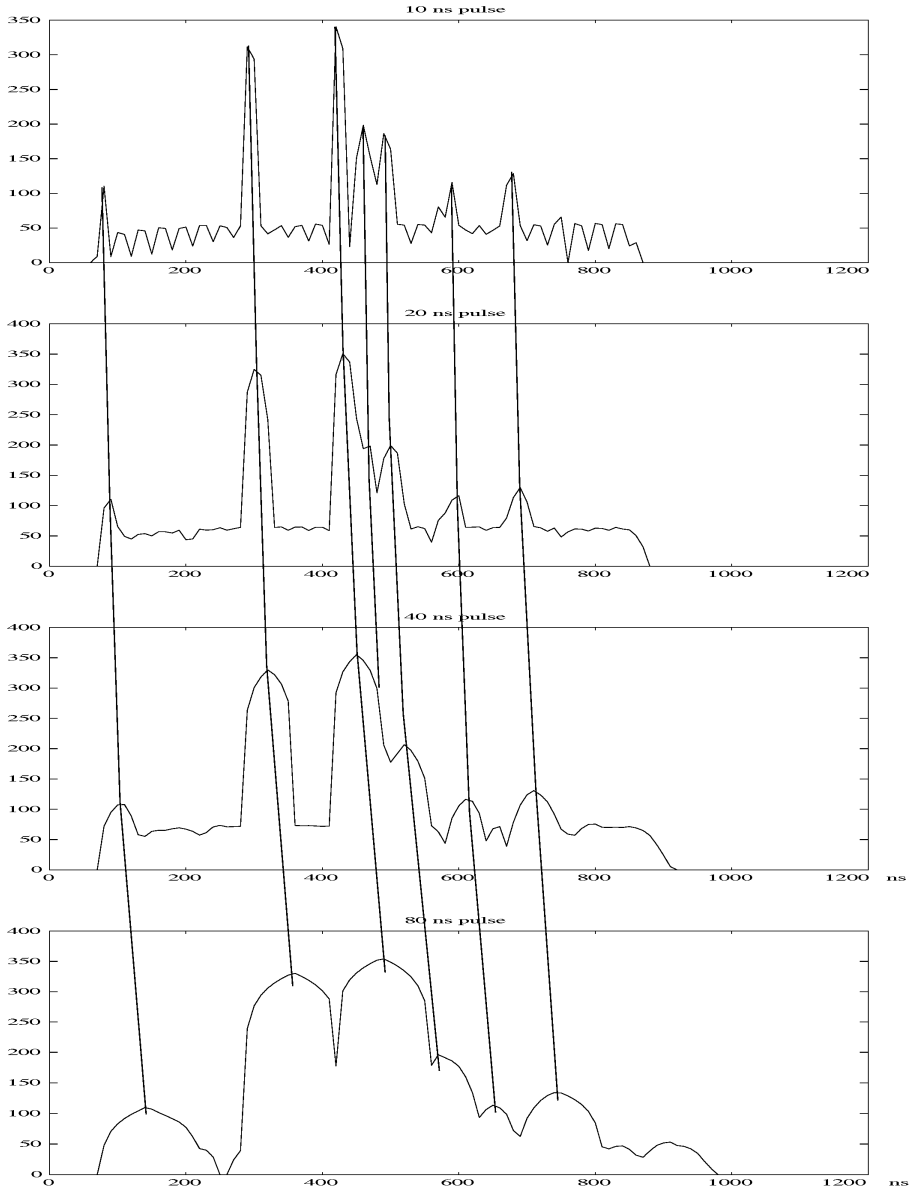
Radar Pulse Returns vs Pulsewidth

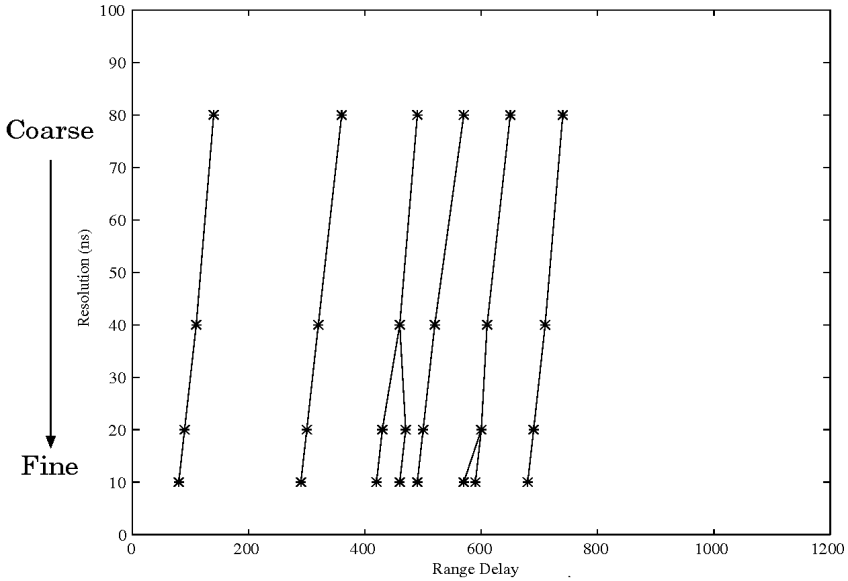


- High curvature points: dominant scatterers
- Coarse resolution (large δ): large structures of ship
- Fine resolution (small δ): finer structures of ship

- Local *max* or *min* are (trivial) edges
- Multi-scale edge representation of signals; complete representation
- How do they coalesce from fine to coarse scale? Scale-space diagrams

Scale-Space Diagrams of Radar Returns





- **Uniform Localization**
- **A different "fingerprint" of the ship**

Hierarchical Organization of Radar Pulses

- **Problem:** How to develop a hierarchical, tree-structured organization of radar returns, which utilizes the multiresolution representations provided by wavelets and leads to fast classification
- Vector Quantization (VQ) is a widely used data compression method
  - Vector Quantizer: a map $Q: R^n \Rightarrow C = \{c_1, c_2, \ldots, c_M\}$; $x \rightarrow Q(x)$ in C
  - C the *codebook* of Q; set of *centroids*; set of *Voronoi vectors*
  - Q induces *partition of* R^n: $A_i = \{x: Q(x)=c_i\}$, $i= 1, 2, \ldots, M$; *(Voronoi) cells*
  - $\rho : \mathbf{R}^n \times \mathbf{R}^n \Rightarrow [0,\infty)$, $\rho(u, v)$ distortion when u is represented by v
- Typically applied to a large sample (size N) of random vectors x_i
  - γ, δ: encoder, resp. decoder of the VQ; if $Q(x_i) = c_m$, then $\gamma(x_i) = m$, and $\delta(m) = c_m$, $Q(x_i) = \delta(\gamma(x_i))$; Since $M<<N$, VQ compresses data
  - **Average Distortion** $D = E[\rho(x, Q(x))] = E[\rho(x, \delta(\gamma(x))] = \int \rho(x, Q(x))\, dP(x)$

    Empirical Distortion: $\bar{D} = \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} \rho(x_i, Q(x_i))$
  - **Rate:** $R= \log_2 M /n$; # of bits per vector component
  - **bit-rate:** $R_b= nR$; **vector rate:** f_V; **bit-rate:** $R_s= nRf_v$;
  - Blocked scalar process *(block size n);* **sampling rate:** f_s, **bit-rate:** $R_s= Rf_s$

Hierarchical Organization of Data and VQ

- Progressive compression: both encoder and decoder use the structure
- Designing the tree structure: application of the LBG algorithm to successive stages using a training set
- We used a variant of this method which is of the "greedy" variation. Our algorithm splits the cell which contributes the largest portion of the current overall distortion (Makhoul, Roucos, Gish, 1985)
- Codewords have variable lengths; variable rate coding: devote more bits to important vectors, and fewer to unimportant vectors
- Better strategy for TSVQ: split node that results in the biggest decrease in average distortion after the split (Gray *et al*, Nobel, …)
- T a binary tree; T_L the leaf nodes; *depth(v)* of a node *v* is the length of the path from the root to *v* ; Corresponding TSVQ described by T with nodes labeled with distinct vectors in R^n;
- $Q_T(x)$ determined by binary comparisons that trace a path in *T*
  - *x* moves to the "child" of *v* that is closest to *x*
  - Codebook of Q_T = vectors labeling nodes in T_L

Hierarchical Organization of Data and VQ

- **Performance of TSVQ applied to random vector *x* with distribution *P* :**

  Distortion: $D(Q, P) = E[\rho(x, Q(x))]$

  Rate (Expected Depth of T): $R(T,P) = \Sigma\, depth(v)\, P(v)\ |v \text{ in } T_L$

- ***V* a region in R^n; a centroid for *V* with respect to *P* is a vector *c* that minimizes** $\int_V \rho(x, c)\, dP(x)$ **$C_1(V, P)$ all centroids for *V***

- **Split *V* into two regions; assign each *x* to *a* or *b***

$$V_{ab} = \{x \in V : \rho(x,a) \le \rho(x,b) \qquad V_{ba} = \{x \in V : \rho(x,b) \le \rho(x,a)$$

***(a,b)* is a centroid pair for *V* with respect to *P* if it minimizes**

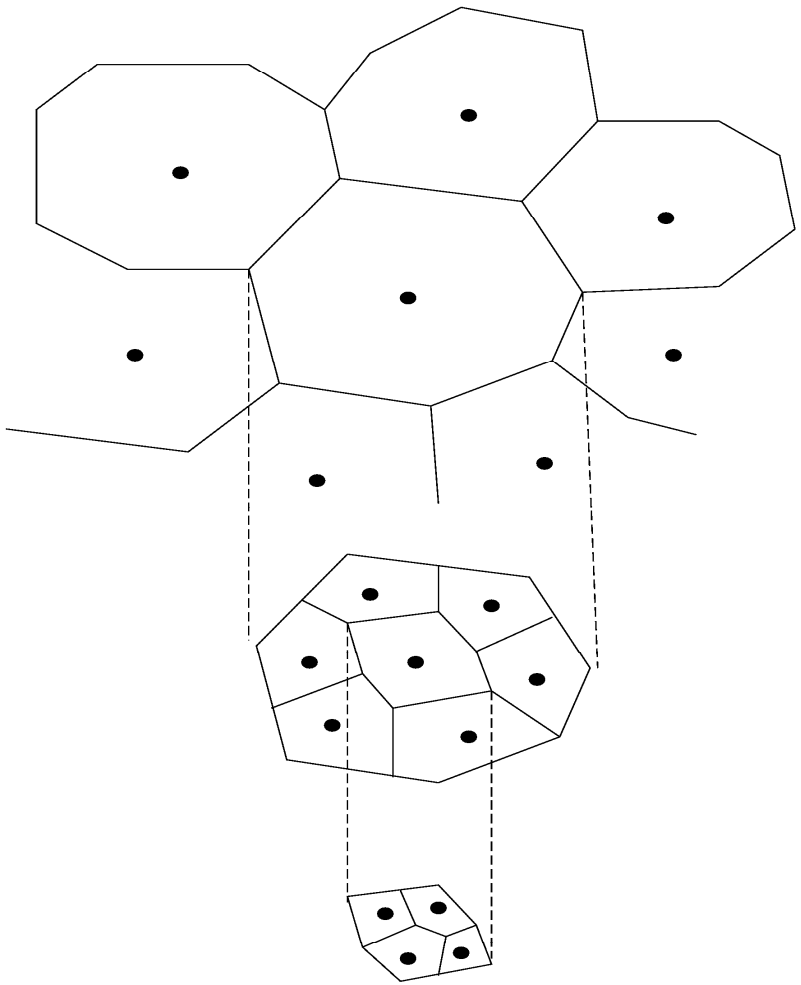
$$\int_V \min(\rho(x,a), \rho(x,b))dP(x)$$ **$C_2(V, P)$ centroid pairs for *V***

- **Splitting *V* yields a decrease in distortion**

$$\Delta D^*(V) = \int_V \rho(x,c)dP(x) - \int_V \min(\rho(x,a), \rho(x,b))dP(x)$$

- **Greedy design rule for TSVQ: split so as to maximize $\Delta D^*(V)/P(V)$**
- **Trees converge, distortion converges based on empirical estimates**

Wavelet Tree-Structured Vector Quantization



First perform a multiresolution wavelet representation of the signals

Consider each signal f at different resolutions

$S^0 f,\ S^1 f,\ \ldots,\ S^{J^*} f$

Proceed by partitioning the signal space at various resolutions in progressively finer cells

Layer in tree $l = J^* - m$,

m the scale

(top layer 0: coarsest)

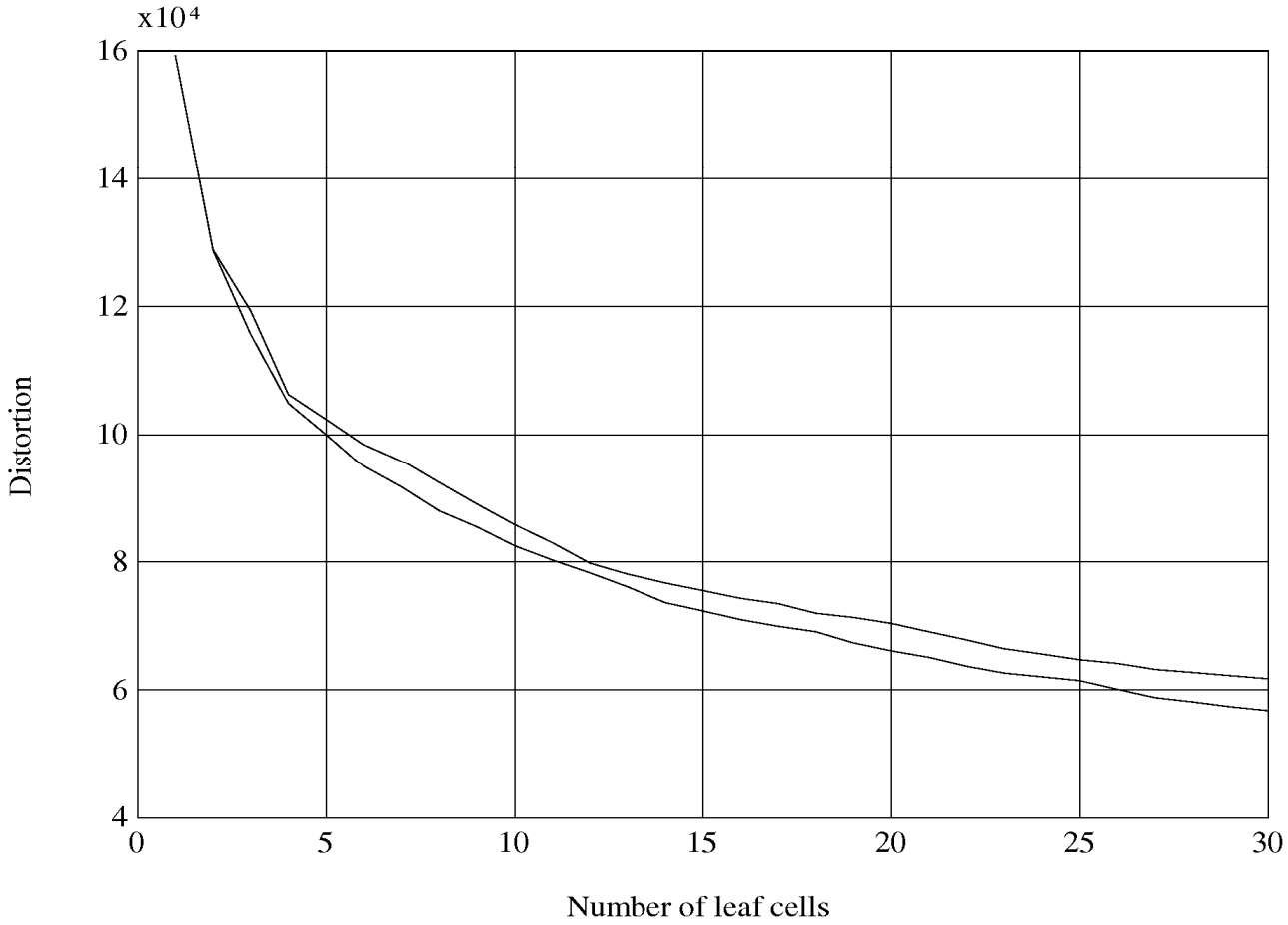
Cell labels: (layer, index)

or (scale, index)

WTSVQ

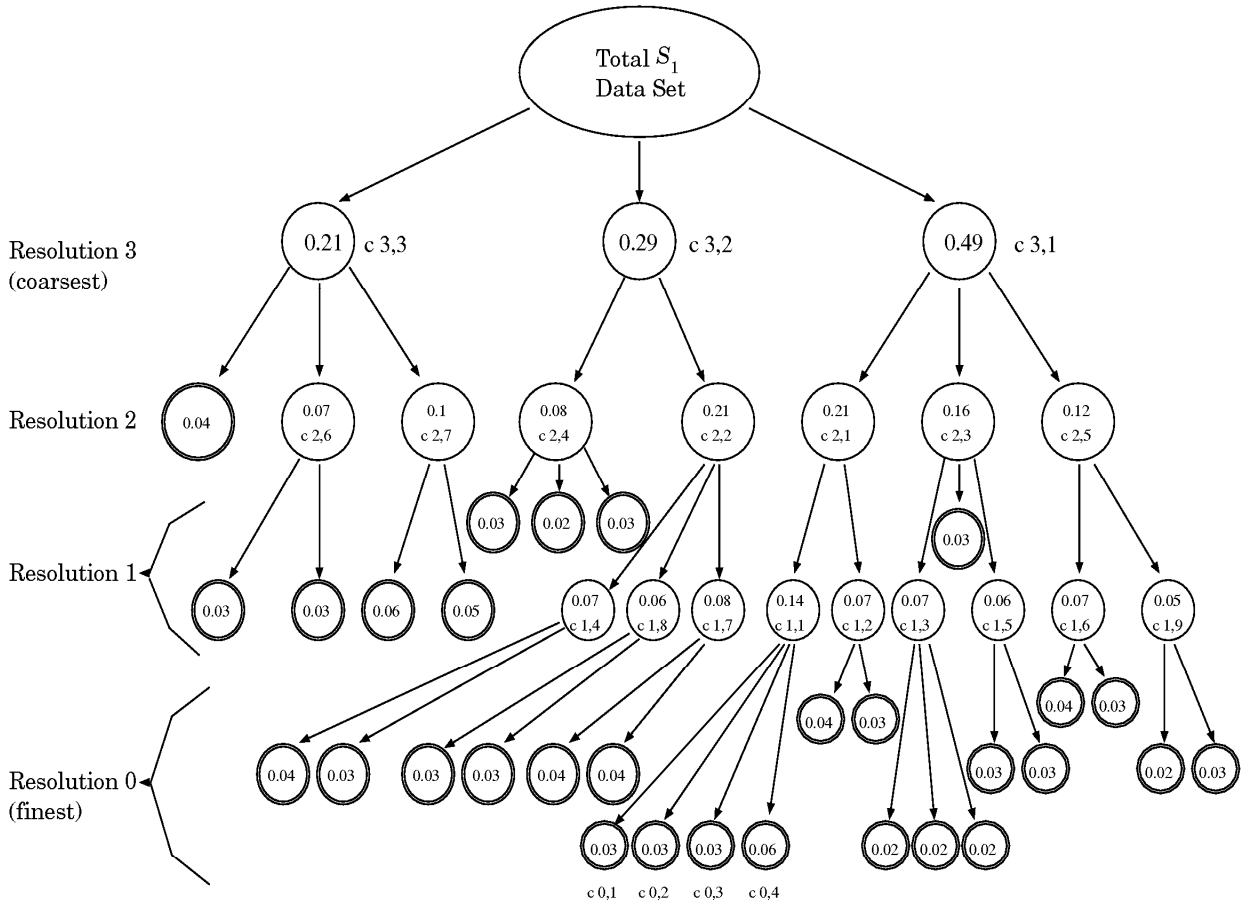
- The data vector space (signal space) is partitioned into cells by the repeated application of the Linde-Buzo-Gray (LBG) algorithm
- LBG first applied to coarsest representation of data vectors $\{ S^{J^*}f,\ f \in S \}$
- Resultant distortion determined based on a mean squared distance metric; computed using the finest resolution representation of the data vectors
  - Coarsest representation, corresponding length data vectors the shortest (16)
  - Clustering is faster than a clustering performed on the much longer fine resolution representations of the data vectors.
- Split the cell (coarse resolution) which is the greatest contributor to total average distortion for the entire partition ; in the next application of LBG
  - A new Voronoi vector is found near the Voronoi vector for the cell to be split and is added to the Voronoi vectors previously used for LBG
  - LBG is then applied to the entire population of data vectors, again using the coarsest representation of each vector
- These steps are repeated until the percentage reduction in distortion for the entire population falls below a predetermined threshold

WTSVQ: Performance on Radar Data



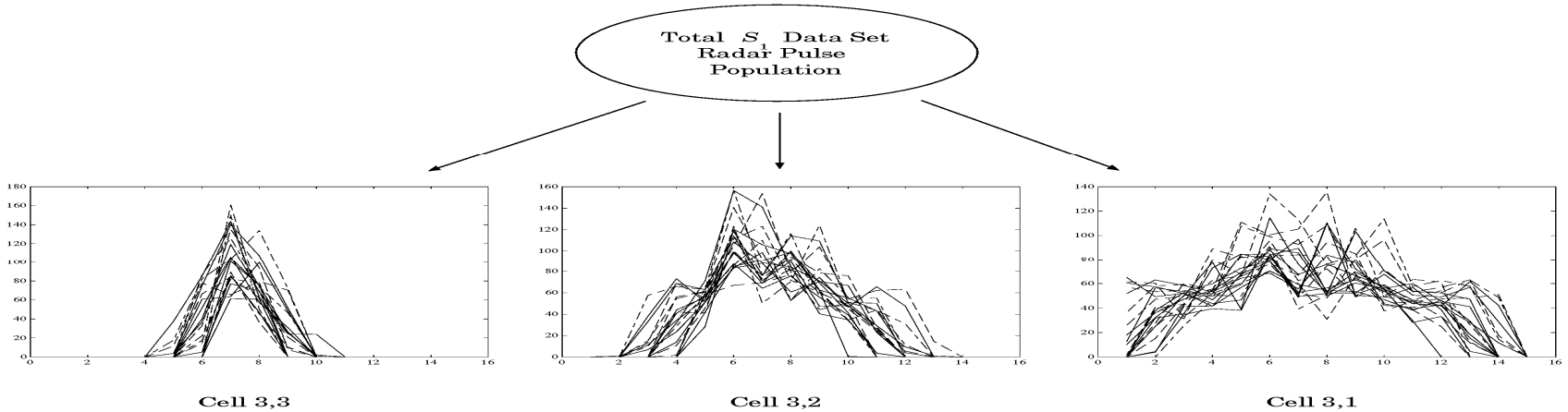
- "Greedy" but faithfull
- Comparison between WTSVQ and full search VQ on fine resolution data
- Excellent performance in many experiments
- Pulses were properly "centered"

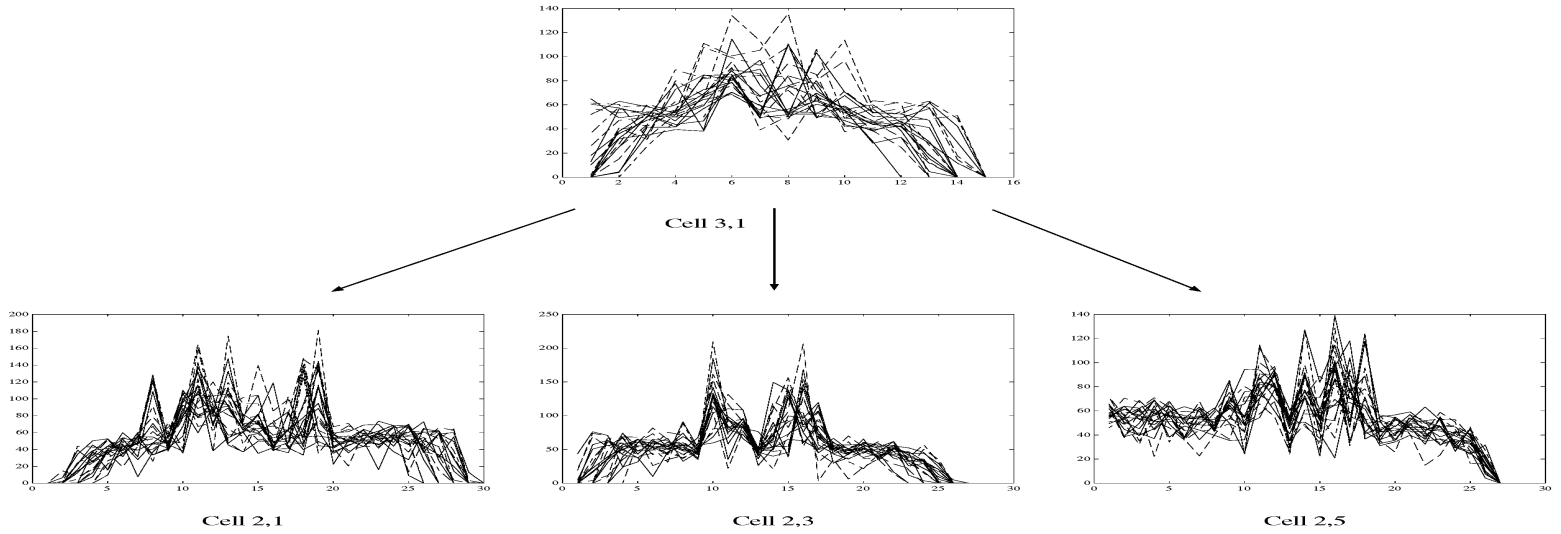
WTSVQ: Aspect Graph Interpretation



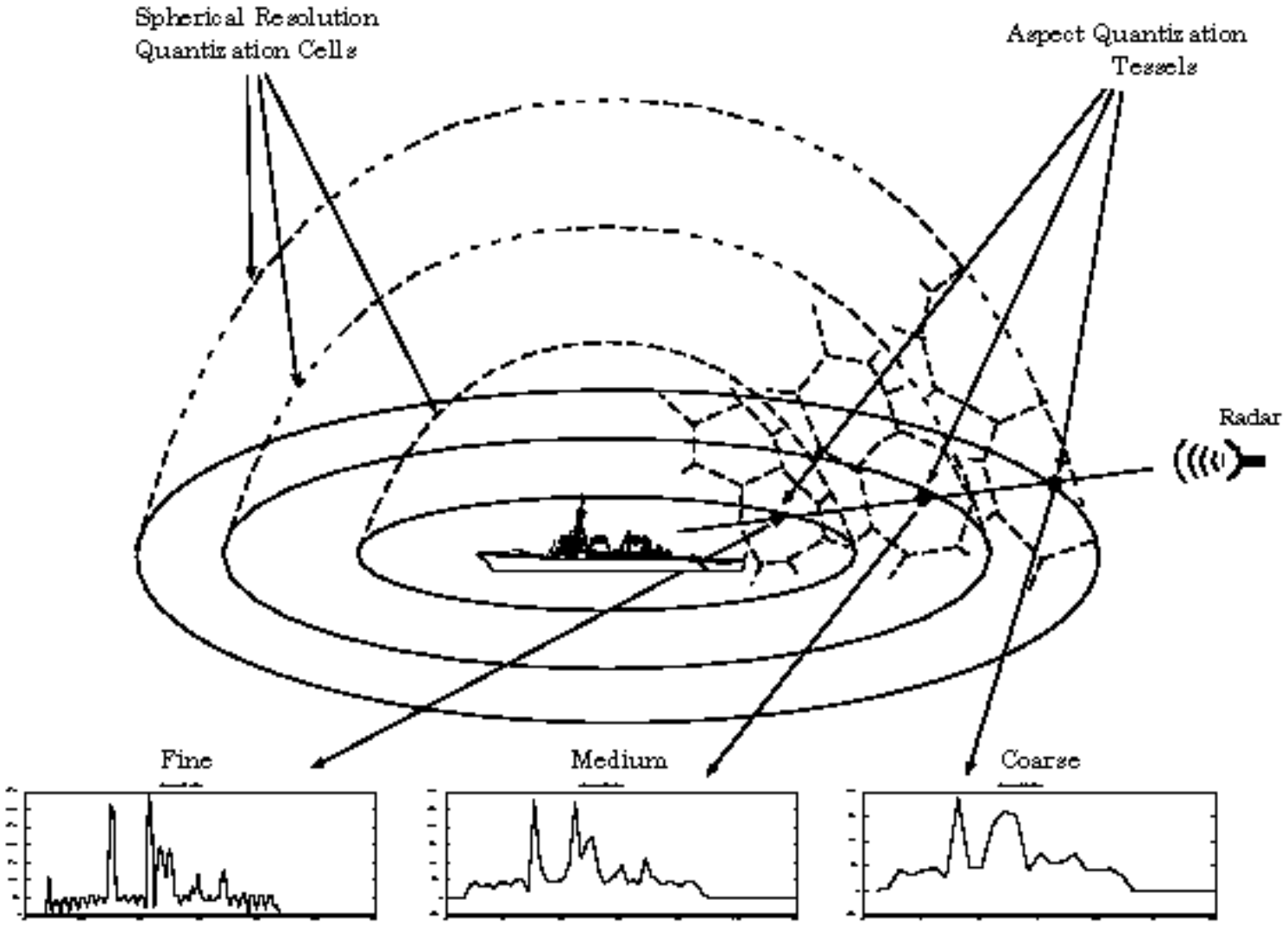
- At coarse level WTSVQ clusters pulses according to aspect
- Moving to finer resolutions clusters pulses according to local maxima
- Extremely efficient indexing scheme akin to aspect graph: A multiresolution aspect graph

WTSVQ: Aspect Graph Interpretation

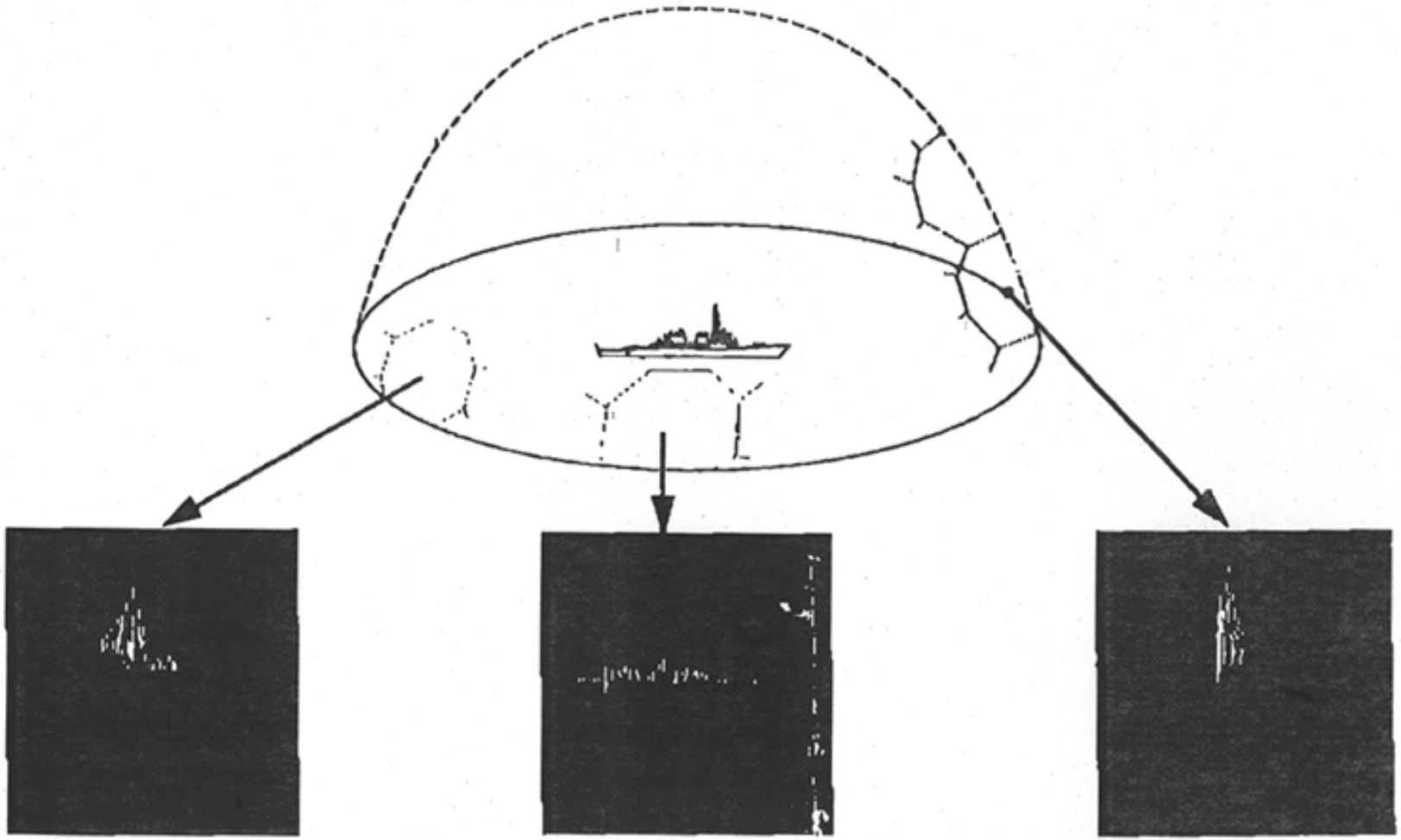




Multiresolution Aspect Graph: Radar Data



Multiresolution Aspect Graph: ISAR Data



WTSVQ: Aspect Graph Interpretation

- The nodes (cells in these figures) correspond to aspect-elevation neighborhoods from which the pulse returns are indistinguishable :

  A tesselation of aspect-elevation space
- Transitions from one node to the other indicate a change in aspect-elevation, or in resolution, causing a detectable change in the pulse
- In the radar case these changes are due to grouping (or ungrouping) of scatterers, or scatterer visibility (or non visibility) from the cell
- We have developed an algorithmic construction of

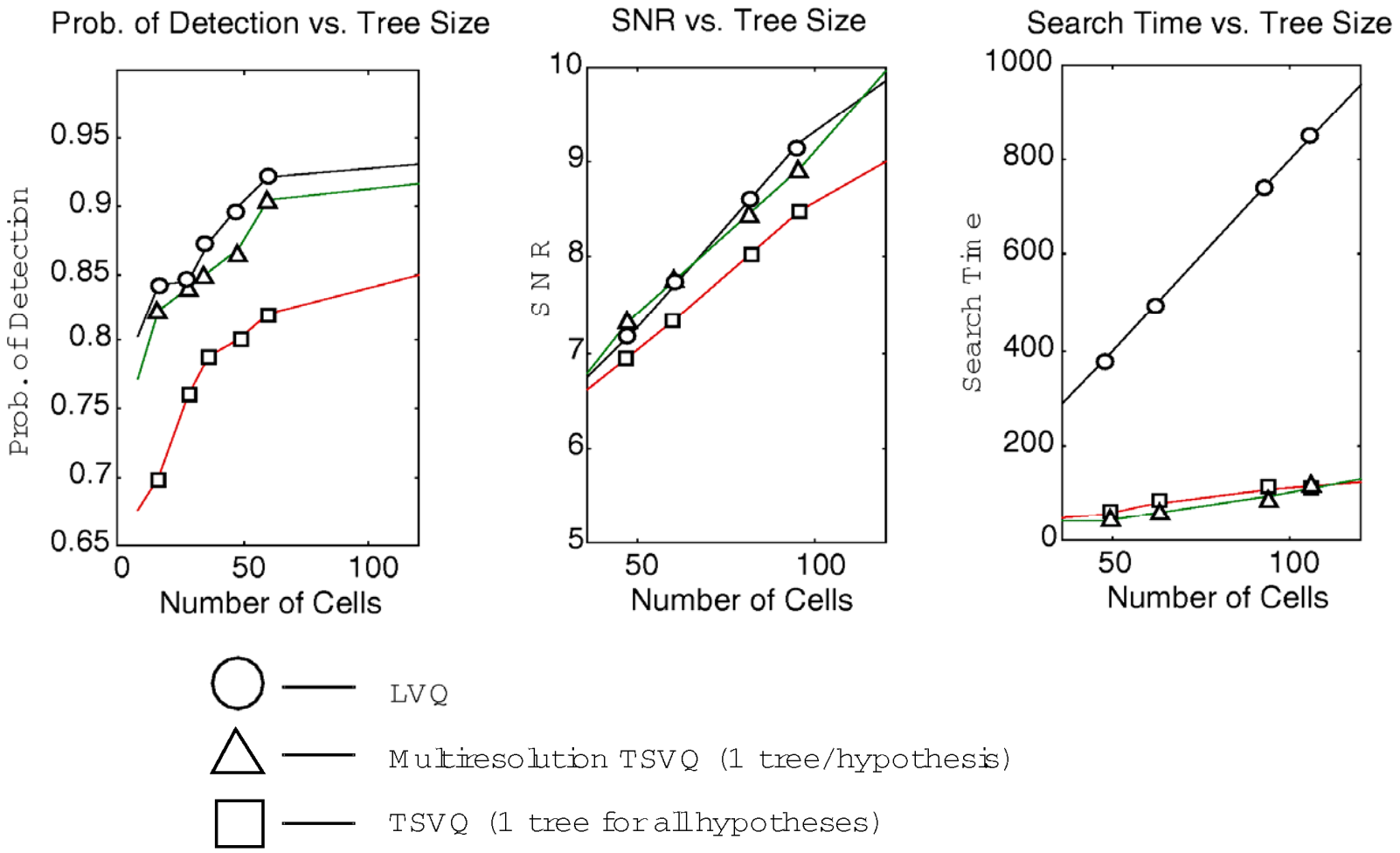
  scale space aspect graphs (or multi-resolution aspect graphs)
  - Compare with conventional *ad hoc* methods of indexing radar pulses

    based on small aspect-elevation cells
- Method reflects the accuracy limitations of the sensor
  - Does not attempt to separate the pulses more than the sensor noise will permit
- Algorithm constructs the minimum number of "views" needed

ATR with Global or Parallel Aspect Graph

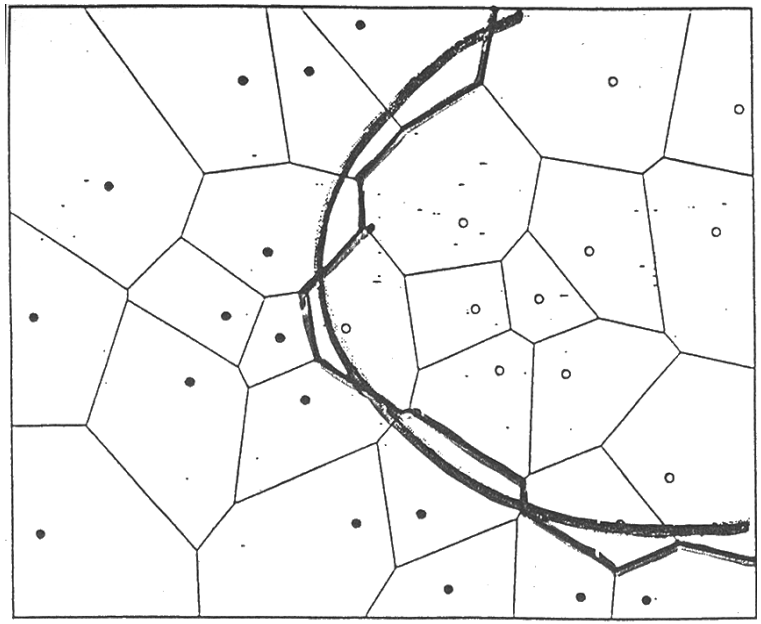
- Represent the entire radar database on targets by
  a single aspect graph : *the Global Aspect Graph* (GAG)
- Represent each data set from a single target by
  an aspect graph: *the Parallel Aspect Graph* (PAG)
- ATR using the first approach we find the leaf node of the Global Aspect Graph that is closest to the data and this gives us the decision
- ATR using the second approach we pass the real-time data in parallel from each aspect graph, find the aspect graph whose leaf provides the best proximity to the data and identify the target with the label of this aspect graph
- Second algorithm is much faster due to its parallel implementation
- Many experiments have shown that it performs much better than the first consistently, as measured by confusion matrix and ROC curves
- It also performs close to the optimal provided by LVQ (Bayes optimal)
- Second algorithm provides the best solution for inserting new targets in the database because it does not require a recomputation of the aspect graphs

Performance of WTSVQ: PAG vs GAG



Learning Vector Quantization

- Data driven; uses past data directly in the classification scheme
- Does not assume any models for underlying data



- Estimates the decision regions directly
- Training phase and classification phase
- Training phase:

Z = training data = $\{(y_n, d_{y_n})\}_{n=1}^{N}$

Voronoi vectors = $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$

decisions = $\{d_{\theta_1}, d_{\theta_2}, \Lambda, d_{\theta_k}\}$

- Pick $z_j=(y_j, d_{y_j})$ from Z and find ρ - closest vector θ_c
- Modify θ_c as follows $\theta_c(n+1) = \theta_c(n) - \alpha_n \nabla_\theta \rho(\theta_c(n), y_j)$ if $d_{y_j} = d_{\theta_c}$

$\theta_c(n+1) = \theta_c(n) + \alpha_n \nabla_\theta \rho(\theta_c(n), y_j)$ if $d_{y_j} \neq d_{\theta_c}$

- Continue until convergence

LVQ

- **Classification phase: for new observation *x* declare**

$$d_x = d_{\theta_j} \quad \text{if} \quad x \in V_{\theta_j}$$

- **LVQ adjustment has the general form**

$$\theta_i(n+1) = \theta_i(n) + \alpha_n \gamma(d_{y_n}, d_{\theta_i}(n), x_n, \Theta_n) \nabla_\theta \rho(\theta_i(n), y_n)$$

$$\gamma(d_{y_n}, d_{\theta_i}(n), y_n, \Theta_n) = -1_{\{y_n \in V_{\theta i}\}}(1_{\{d_{y_n} = d_{\theta_i}\}} - 1_{\{d_{y_n} \neq d_{\theta_i}\}})$$

- $\Theta_{n+1} = \Theta_n + \alpha_n H(\Theta_n, z_n)$ **; stochastic approximation**

$$z_n = (y_n, d_{y_n})$$

- **For appropriate conditions on α_n, H, z_n, Θ_n approaches the solution of the ODE**

$$\frac{d}{dt}\overline{\Theta}(t) = h(\overline{\Theta}(t))$$

for appropriate *h*(Θ)

Combined Compression and Classification

- Same as partitioning the feature space into decision regions corresponding to each class
  - Boundaries of the cells can approximate the Bayes decision surfaces
- Need to develop efficient adjustment methods to weight the relative importance of the two aspects of the algorithm
  - Efficient compression can reduce significantly the complexity of classification by bringing forward essential local features of the signal
  - Excessive compression may throw away valuable information and thus reduce the accuracy of classification
  - The design of this tradeoff is the key problem
- Rate and Distortion characterize compression performance of VQ
- Bayes Risk characterizes classifier performance
- Competing performance measures: *multi-objective* design problem:

  Combine measures or optimize one while satisfying constraints on the other

Analytical Framework

- Given a encoder-decoder pair γ, δ we associate the average distortion

$$D(\gamma,\delta)=E[\rho(x,\delta(\gamma(x)))]$$

- Associate the rate R(γ,δ) to a encoder-decoder pair γ,δ
- Given a classification rule *d*, the classification performance of the overall scheme can be measured by the Bayes risk

$$J_B(\gamma,d)=\sum_{i=1}^{L}\sum_{j=1}^{L}P(d(\gamma(x))=H_j \,|\, x\in H_i)P(H_i)C_{ij}$$

- where C_{ij} is the relative cost assigned to the decision that $d(\gamma(x)) = H_j$, while the vector x comes from class H_i (typically $C_{ij} = 0$)
- Encoder δ does not affect the Bayes risk J_B
- Incorporate Bayes risk into the average distortion measure minimized by the design algorithm
- Resulting algorithm has complexity equivalent to that of an ordinary VQ algorithm

Analytical Framework

- Overall approach is non-parametric:

  probability distributions for the data are not needed
- Approach can be interpreted as using the training set to learn the empirical distributions of the vectors and use them as if they were true (like in LVQ)
- Combine the three criteria in one for some choice of the weights λ_R and λ_B

$$J_\lambda(\gamma,\delta,d)=D(\gamma,\delta)+\lambda_R\,R(\gamma,\delta)+\lambda_B\,J_B(\gamma,d),$$

- Three step iterative optimization:
  - Step 1 Choose $d^{(t+1)}$ to minimize J_λ ($\gamma^{(t)}$, $\delta^{(t)}$, $d^{(t+1)}$)
  - Step 2 Choose $\delta^{(t+1)}$ to minimize J_λ ($\gamma^{(t)}$, $\delta^{(t+1)}$, $d^{(t+1)}$)
  - Step 3 Choose $\gamma^{(t+1)}$ to minimize J_λ ($\gamma^{(t+1)}$, $\delta^{(t+1)}$, $d^{(t+1)}$)
  - The iterations continue until the desired stoping level for J_λ is met

A Variation on LVQ

- **Initialize with $\Theta(0)$ from VQ on training set**

- **Assign training vectors to their cells:**
  Find $i_l = \arg\min_m \rho(\theta_m(n), x_l)$ **l=1, 2, …, N, then** $x_l \in V_{\theta_{i_l}(n)}$
- **Cell decisions** $g_i(\Theta(n); N)$ **i=1, 2, …, K**

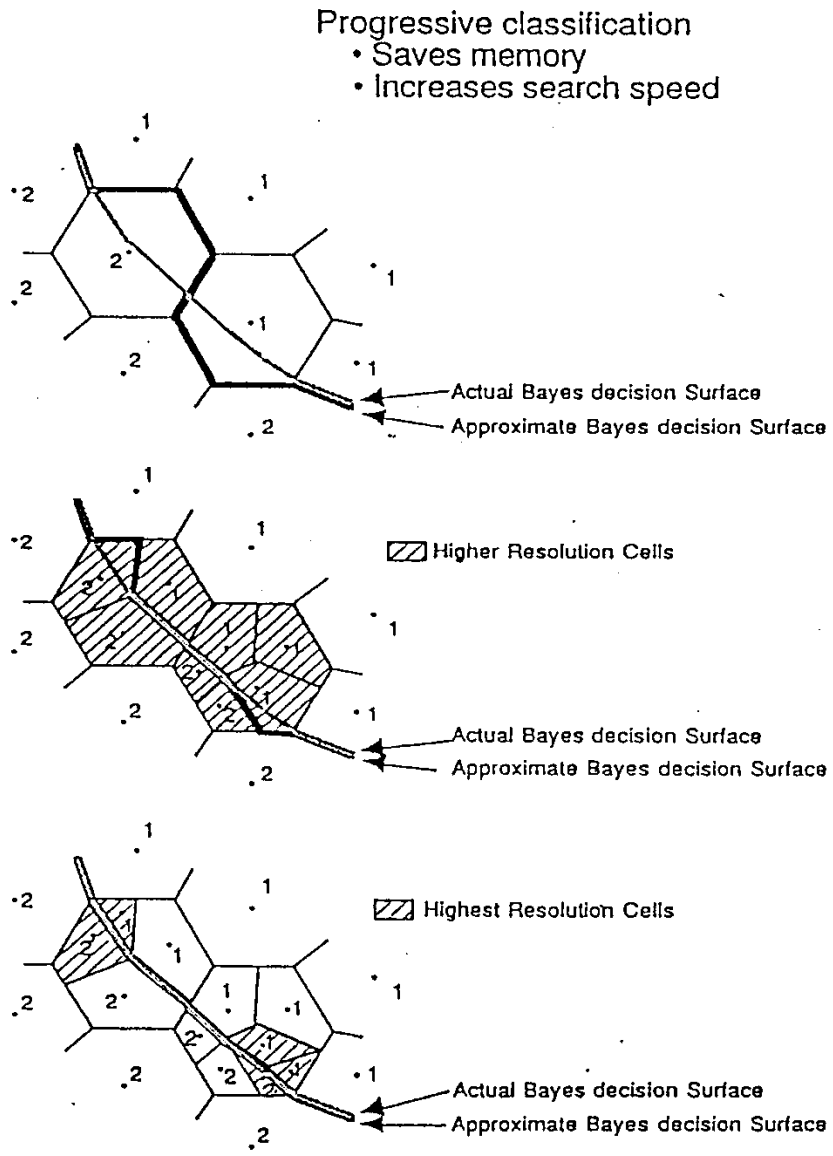
- **Update the Voronoi vectors**
- **If** $x_{n+1} \in V_{\theta_t(n)}$ **then**

$$\theta_t(n+1) = \theta(n) + \varepsilon_{n+1}(-\lambda - 1)\nabla_\theta \rho(\theta, x_{n+1})\big|_{\theta=\theta_t(n)} \quad if \quad dx_{n+1} = g_i(\Theta(n); N)$$

$$= \theta(n) + \varepsilon_{n+1}(-\lambda + 1)\nabla_\theta \rho(\theta, x_{n+1})\big|_{\theta=\theta_t(n)} \quad if \quad dx_{n+1} \neq g_i(\Theta(n); N)$$

- **For** $j \neq i,\ \theta_j(n+1) = \theta_j(n)$

Extension to LTSVQ and Interpretation



- Extension of the LVQ approach to Learning TSVQ
- This step is needed for the full analysis of WTSVQ and its application in progressive classification within the framework of combined compression and classification
- LTSVQ approximates directly the optimal Bayes surface with successive approximations and variable (along the surface) resolution
  - Split cells where approximation is not very good using finer resolution information
  - Akin to a multigrid numerical computation of the Bayes surface

Face Recognition

- Identification of a person based on standard ID picture and a database where he would have previously been filed
- Data we used for experiments: 349 ID pictures of 95 different people. These Photographs are 128 × 128 pixel, 8 bpp gray level images
- Every person is represented by several pictures (2 at least, 4 at most)

  254 photographs form the database; Remaining 95 are the unknowns
- Global identification scheme includes 5 steps
  - 4 occur during the design of the Tree-Structured Data-Base
  - Normalization of the DB, Wavelet Transform, Vector Quantization, Tree design, Tree Search
  - Organization results in fast real-time recognition
  - First of all, the images are normalized thanks to an eye-detection based algorithm
- We have obtained the best results (in terms of high fidelity ID in very short times) with a novel algorithm that combines global tree search with local full search. We showed that this is equivalent with allowing overlapping clusters at each resolution of our progressive classification scheme

Normalization of the Picture Database

- A key problem: selection of a good distortion measure
  - Euclidean distance, PSNR, do not fit well the visual identification process
  - Euclidean distance is very sensitive to slight shifts or rotations
  - Proper normalization of pictures is necessary to make sure that the Euclidean distance is not meaningless
  - Normalization of a picture is achieved by first enhancing it (i.e. increasing the contrast, remove granular noise), then by detecting the eyes, and finally by straightening the picture

Contrast Enhancement

Tree Search and Tree Design

- By organizing the database of pictures into a tree we can obtain fast search, which is logarithmic in the number of pictures instead of linear. The TSVQ part of our algorithm achieves this task. We used Euclidean distance as before.
- Node splitting: selected split that maximizes decrease in distortion

  This produced the best results in accordance with greedy algorithms

  We selected the centroid of each cell as the representative for ID purposes
- We performed tests comparing various search methods
  - Full Search of the DB
  - Tree-Search following our WTSVQ algorithm
  - Best results obtained by a combination method that used initially a Tree-Search followed by a Full-Search in the vicinity of the target image
- Obtained further improvements on performance with Multipath Search algorithm: more paths are followed down the tree, until the resolution (which improves with the layers) is such that a reliable choice can be made. If a choice cannot be made the matches are kept and a decision is made at the end with a multi-resolution Full Search.

Evidence From Biology and Neuroscience

SPEECH RECOGNITION AND ACOUSTICS AND VIRTUAL REPRESENTATION OF SOUNDS IN THE AUDITORY CORTEX

- Focus on functional model of auditory cortex (A1)
- Combine mathematical model with neurophysiology experiments and measurements
- Key questions: What are the features of sound?
  How are they mapped into function neurons in the cortex?
- Go beyond the traditional pitch, loudness and location features
- Use "timbre": interpret the ripples in auditory spectrum
- Use WTSVQ on a multiscale cortical representation of the auditory spectrum in order to identify candidate features identifying vowels, phonemes, instruments
- Perform experiments to validate and verify features used in classification by humans or animals
- Approach very successful ! Results have implications on neural architecture of auditory cortex

Evidence From Biology and Neuroscience

VISION AND VIRTUAL REPRESENTATION OF OBJECTS IN THE VISUAL CORTEX

- **Our work emphasizes storage of objects as a collection of a minimal set of views for storage and classification efficiency**
- **How do animals, humans perform these tasks? Experiments by Poggio and Logothetis , Poggio and Anselmi (2017) with animals (monkeys) and actual brain measurements support our thesis**
- **Monkeys appear to store two-dimensional views of 3-D objects, for various viewpoints**
- **They store more views for new or unknown objects; less views later as they learn to recognize the object**
- **The reduction of views is accompanied by some higher order interpolating function**
- **These observations and theories imply and support certain structure in the neural architecture of the visual cortex**

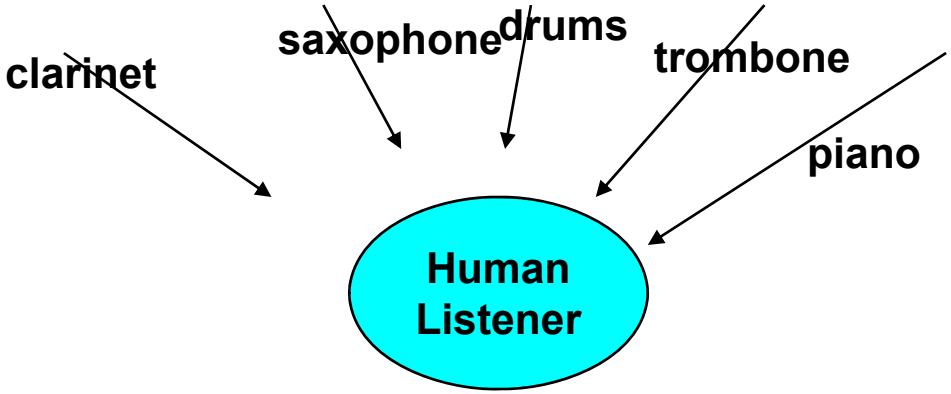
Applications

- **Algorithms motivated by similar processing in animals and humans:**
  - **Hearing and sound classification**
  - **Vision and identification of objects**
- **Text-independent robust speaker identification**
  - **Identifying the speaker from the "music" of his voice**
- **Speaker-independent speech recognition**
  - Identifying phonemes, vowels, words from their inherent sounds
- **Identification of musical instruments ("timbre")**

Applications to acoustic signal recognition
- **Fault identification in tools and wear prediction**
- **Ground vehicle identification from array microphones**

Joint DOA, Instrument ID, Note ID

Can we mimic and understand the ability of humans to do partial recognition of musical instruments and DOA in a combined and mutually enhancing fashion?



- Combine the Stereausis model and its derivatives , with the Auditory filtering multiscale VQ algorithms
- Using the cochlea, cortical, or combined spectra, perform DOA on a “per frequency band basis”
- Combine portions of spectra according to DOA
- Use the multiscale classifier to ID portions of spectra tagged by angle, as compared to stored vehicle spectra
- Repeat the cycle as the scenario evolves

Acoustic Detection and Identification of Vehicles

- Used and extended the two dimensional stereausis neural network (Shamma et al). Uses as input processing the two cochlear filter banks
- Measures binaural differences by detecting the spatial disparities between the instantaneous outputs of two filter banks
- The inhibition neural network computes effectively these differences and articulates the harmonic peaks
- Vehicle acoustics dominated by few low frequency harmonics
- Our experiments show that identification based only on the main diagonal spectrum gives only 52% correct classification. Thus additional features are needed
- Introduced a measure of the degree of association between the various vehicle harmonics and modified the stereausis network to compute this association by:
  - **Adding an envelope extractor immediately behind the filter banks**
  - **Computed correlation with longer lags (than one)**
  - **Allowed no-local (over frequency bands)correlations to be computed between left-right inputs**

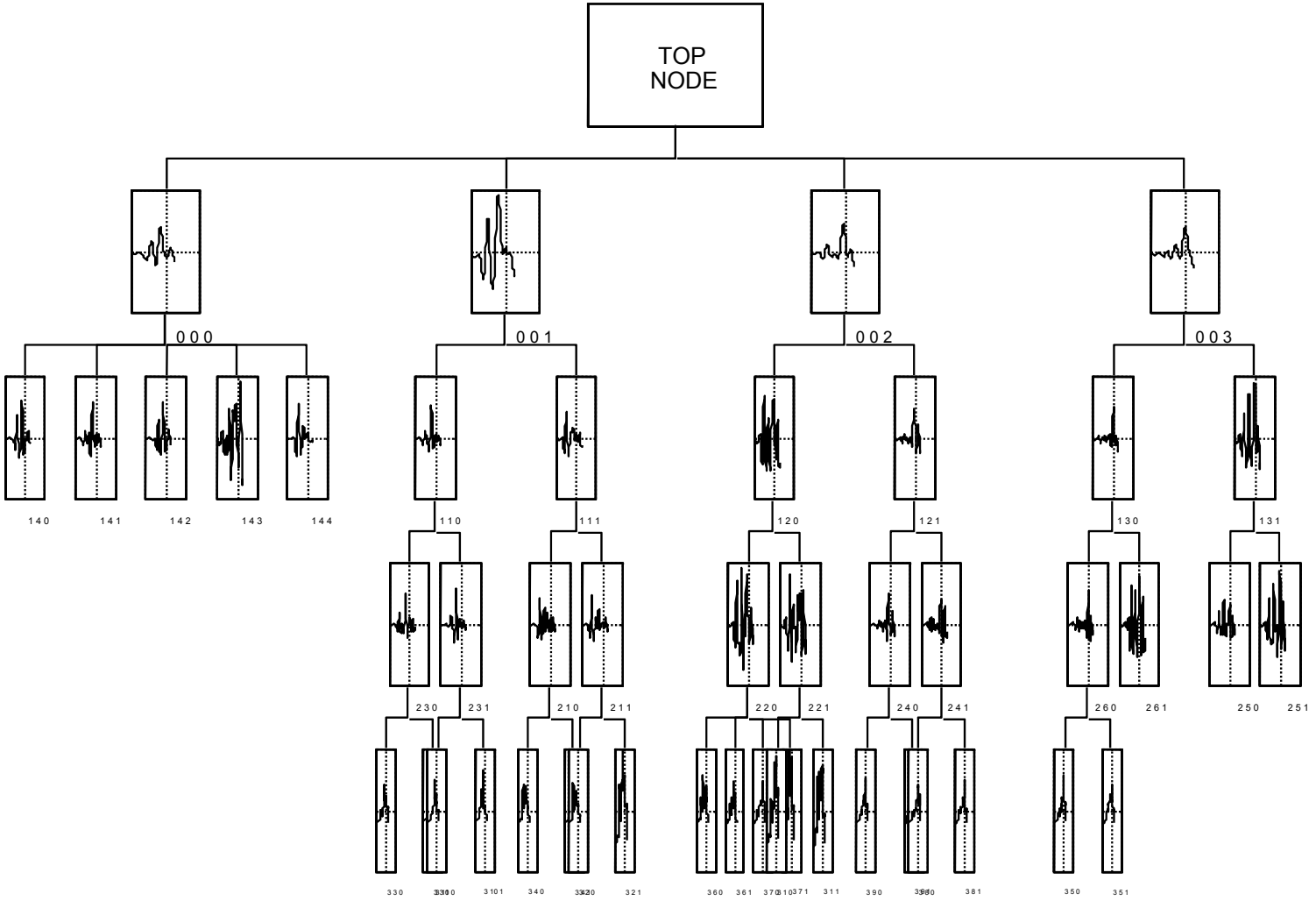
Options in Applying WTSVQ to Acoustic Vehicle Classification

- **GTSVQ: A global tree-structured multi-resolution clustering mechanism that mimics the aggressive and topological hearing capabilities of biological systems. Here a global tree is built on training data from all vehicles. New vehicle insertion problem.**
- **LVQ: A supervised learning neural network, LVQ achieves optimal classification in the Bayes sense. It has the disadvantages of a long search time and sensitivity to initial conditions.**
- **Parallel TSVQ (PTSVQ): build one (or more) trees for each vehicle. It achieves a trade-off between GTSVQ and LVQ on classification performance and search time. Easy new vehicle insertion.**
- **The following node allocation schemes are examined for PTSVQ:**
  - **PTSVQ(1): Allocation based on sample a priori probability**
  - **PTSVQ(2): Allocation based on equal distortion**
  - **PTSVQ(3): Allocation according to vehicle speed**

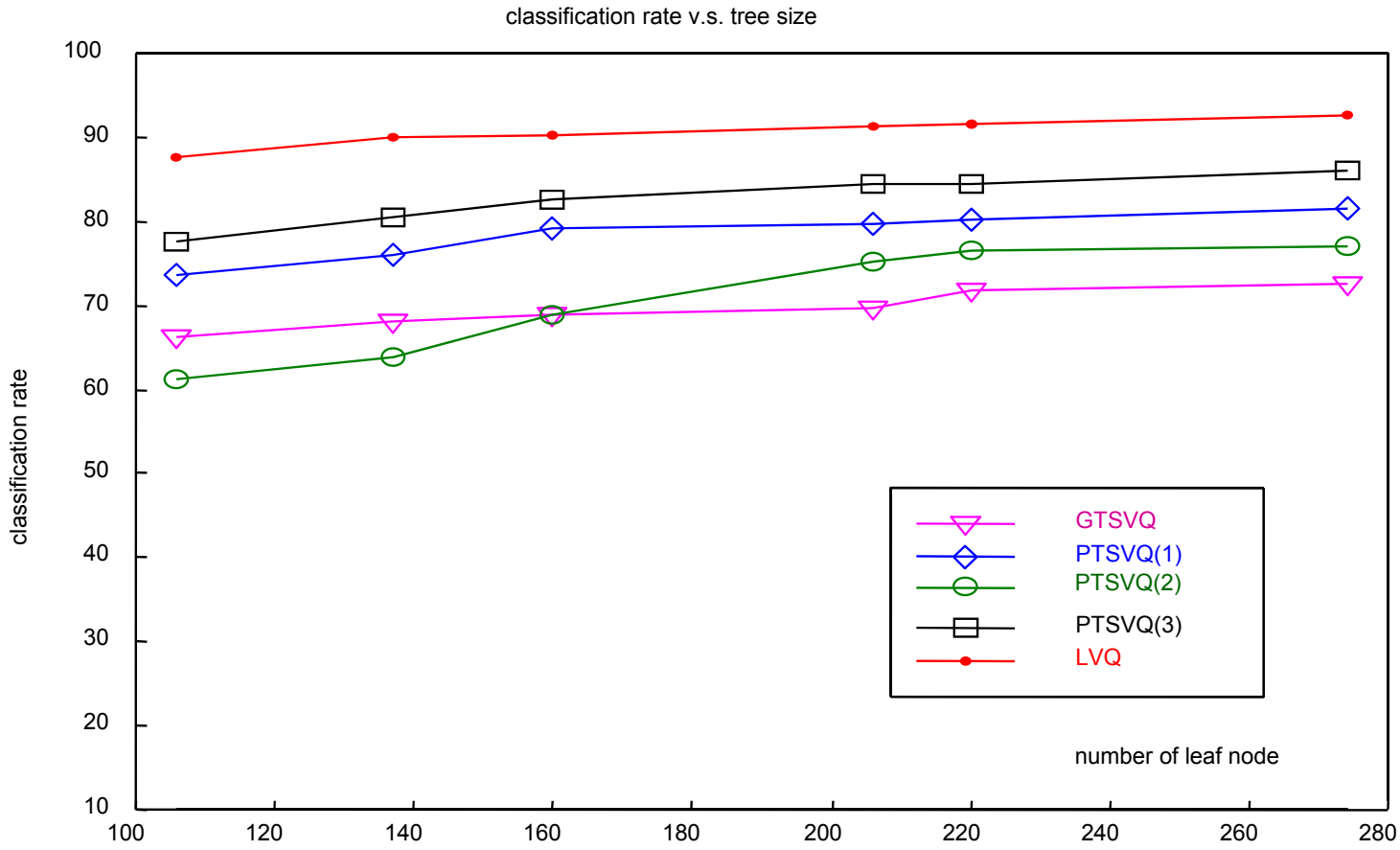
Leaf Node Entropies for PTSVQ Tree of Vehicle Type 8

cell entropy

1 4 0	1.3570
1 4 1	0.9503
1 4 2	1.1779
1 4 3	1.0735
1 4 4	1.3022
2 5 0	0.6365
2 6 1	0
3 1 0	0.5765
3 1 1	0.2993
3 2 0	0.7516
3 2 1	0.4765
3 3 0	0.7633
3 3 1	0.5670
3 4 0	0.4540
3 4 1	0.4384
3 5 0	0.2728
3 5 1	0.4975
3 6 0	0.5313
3 6 1	0.3061
3 7 0	0.6054
3 7 1	0.6383
3 8 0	0.4824
3 8 1	0.5377
3 9 0	0.5044
3 9 1	1.2556
3 10 0	1.0144
3 10 1	1.1967

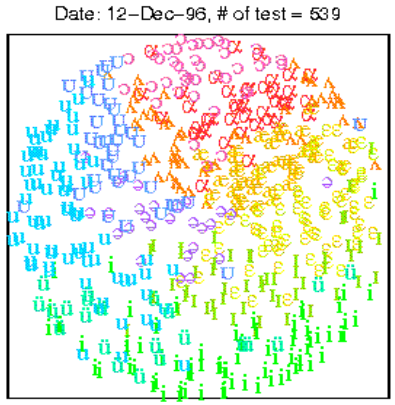


Performance Comparisons among Options

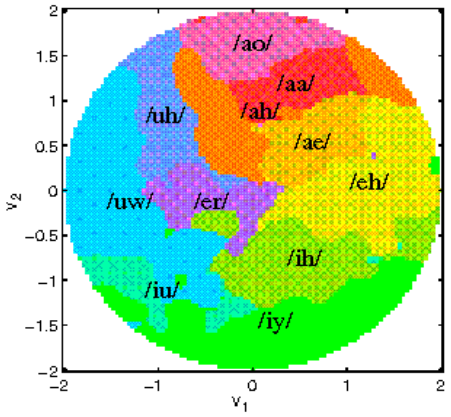


Classification Performance: 70% samples for training, 30% for testing (same microphone)

WTSVQ in Multiscale Cortical Model



Perceptual Result

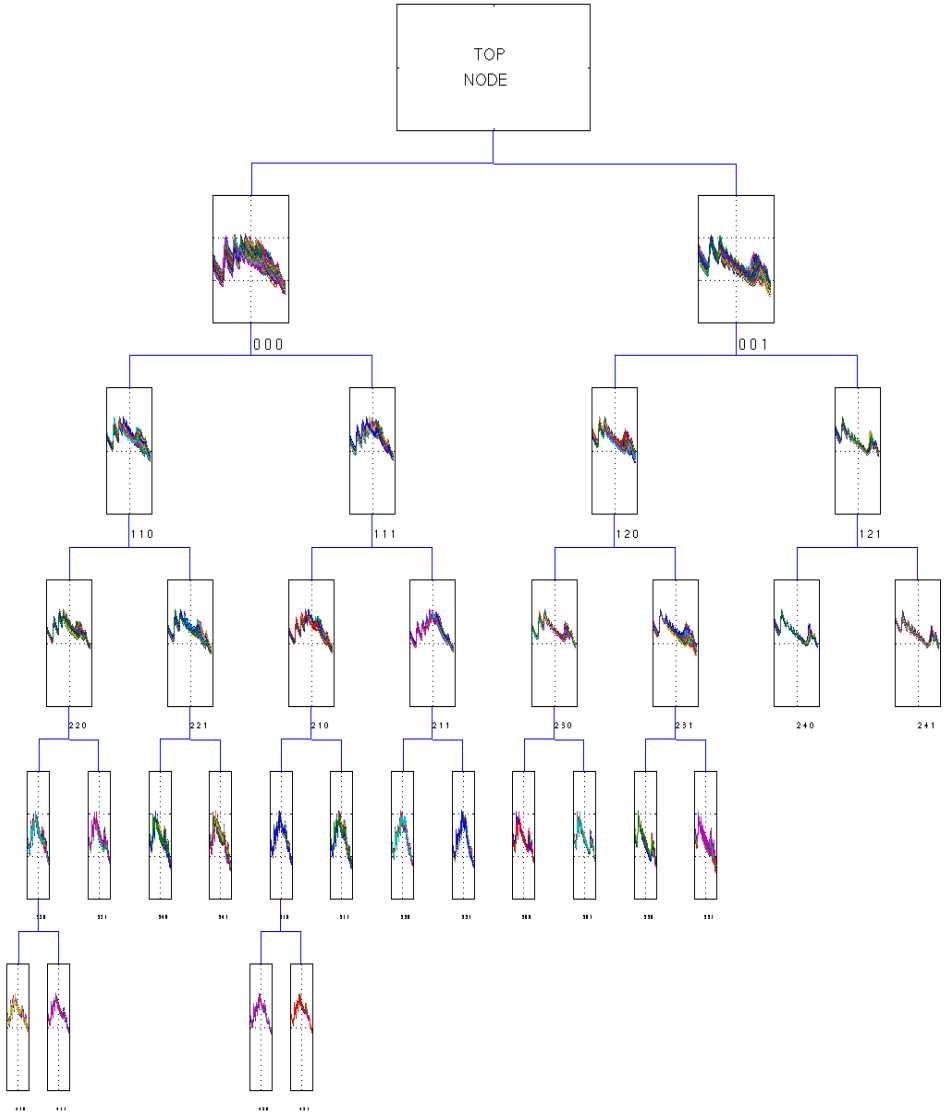


Processed Result

- Experiments with synthetically created vowels using a sinusoidal shape of the vocal tract
- Shape parameters: amplitude (ρ) and initial phase (θ) of the sinusoid
- v_1, v_2 : cartesian coordinates for pair (ρ,θ)

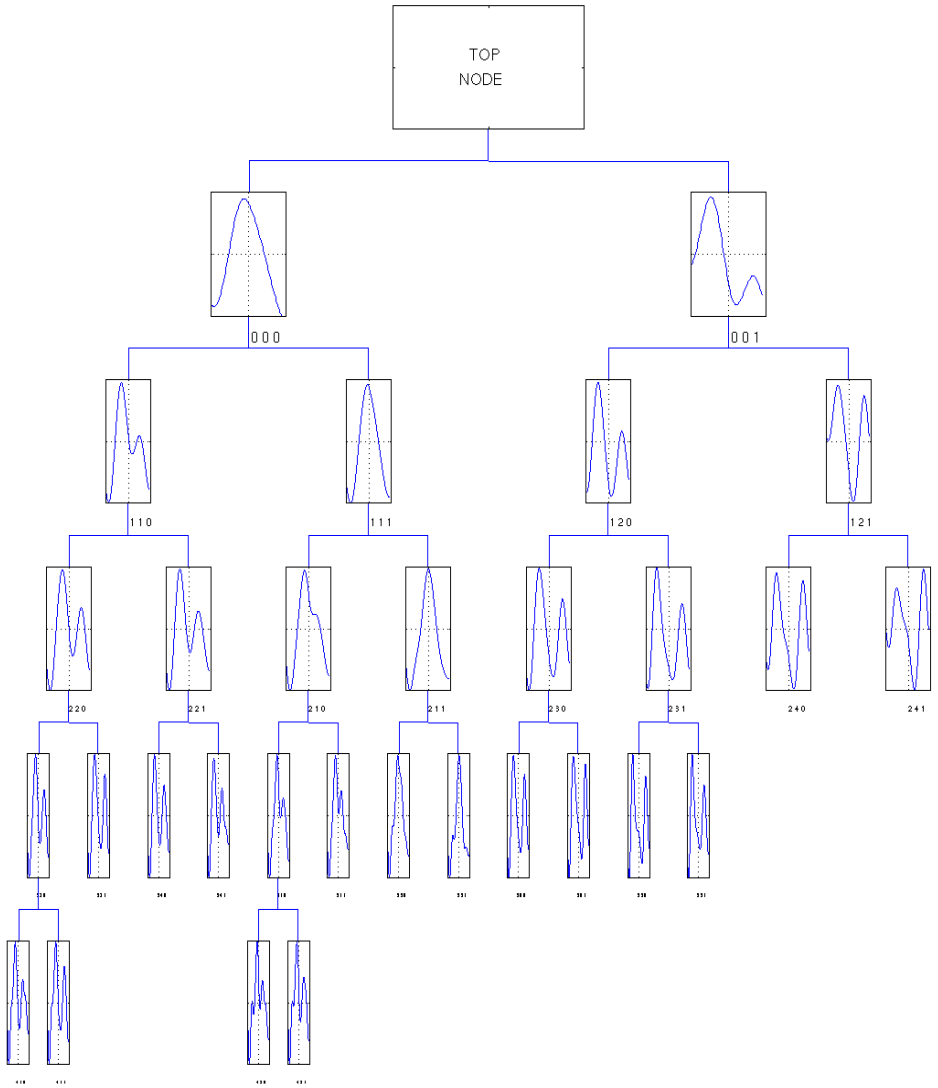
Combine the wavelet representation of the "double transform" cortical model with TSVQ to identify features revealed by resolutions

WTSVQ in Multiscale Cortical Model



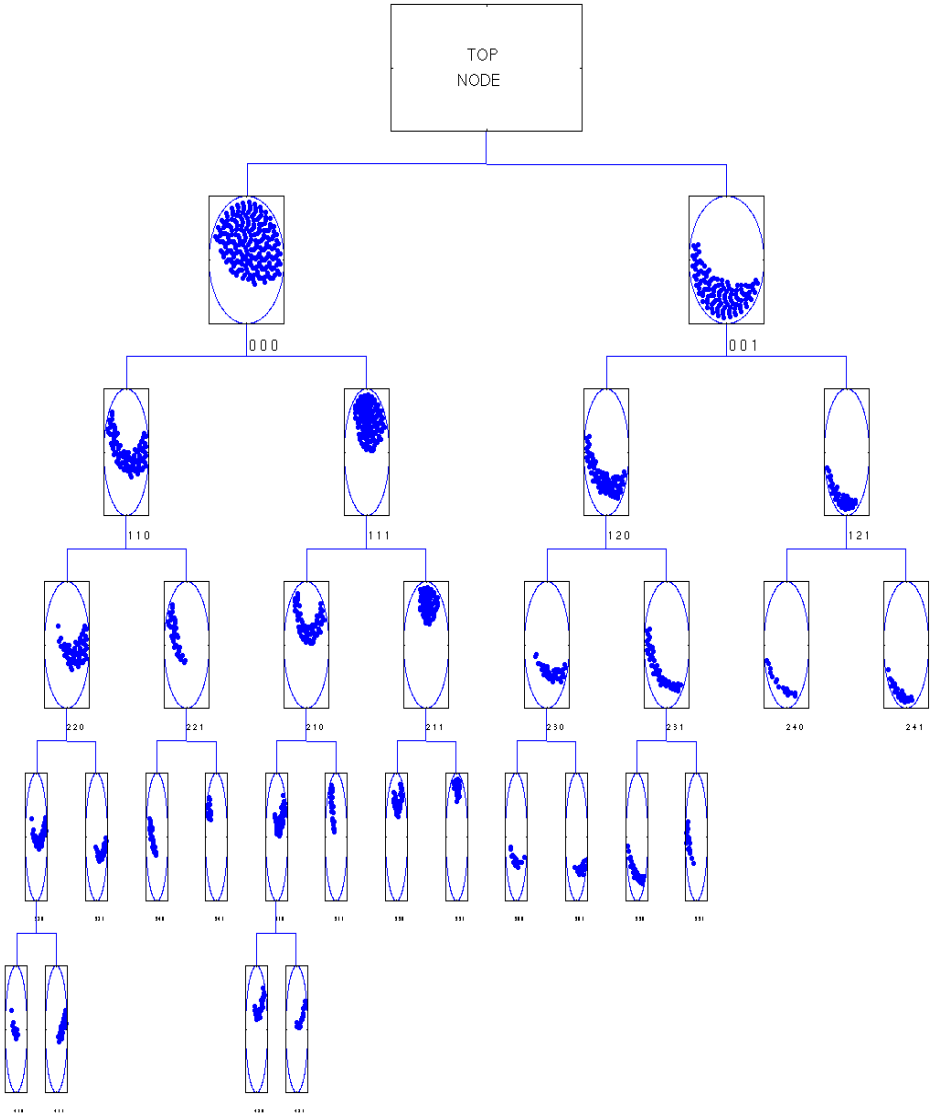
Auditory spectra
pitch: 120 Hz

WTSVQ in Multiscale Cortical Model



Centroids of spectra at different scales pitch: 120Hz

WTSVQ in Multiscale Cortical Model

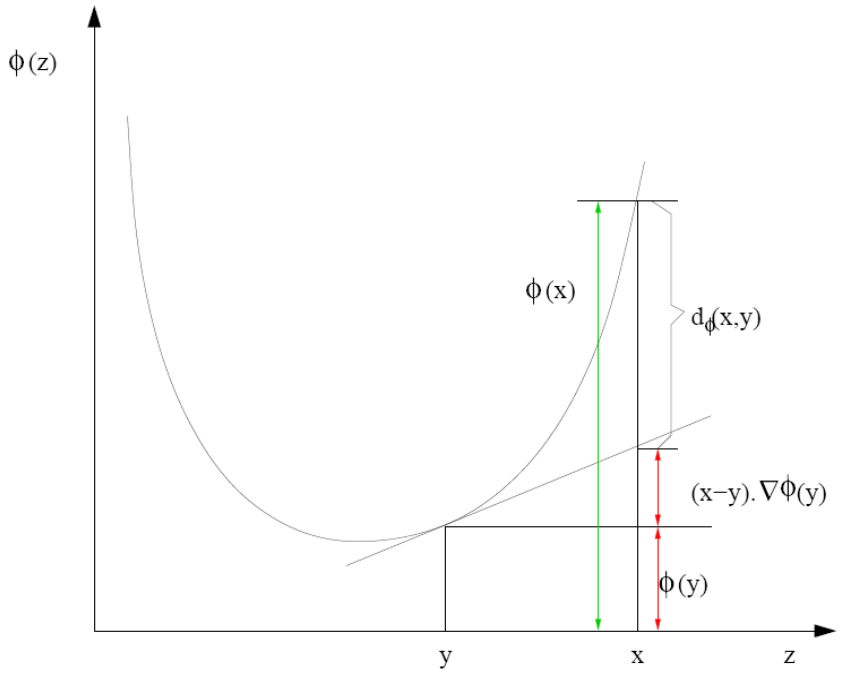


Identification of front, back and subsets of vowels from synthetic database pitch: 120 Hz

Dissimilarity Measures: Bregman Divergence

- **Euclidean distance – most commonly used**
  - Nearest neighbor, k-means clustering, least squares regression, PCA, distance metric learning, etc
- **But…is it always appropriate? No!**
  - Nominal attributes (e.g. binary)
  - Distances between distributions
  - Need to handle numerical, logical, and even rule variables
- **Probabilistic interpretation:**
  - Euclidean distance ⇔ Gaussian data
  - Beyond Gaussian? Exponential family distributions ⇔Bregman divergences

Dissimilarity Measures: Bregman Divergence



ϕ is strictly convex, differentiable

$$d_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla\phi(\mathbf{y}) \rangle$$

Dissimilarity Measures: Bregman Divergence

Properties of Bregman Divergences

- Not a metric (symmetry, triangle inequality do not hold)
- $D_\phi(\mathbf{x}, \mathbf{y}) \geq 0$, and equals 0 iff $\mathbf{x} = \mathbf{y}$
- Strictly convex in the first argument, but not convex (in general) in the second argument
- Three-point property generalizes the "Law of cosines":

$$D_\phi(\mathbf{x}, \mathbf{y}) = D_\phi(\mathbf{z}, \mathbf{y}) + D_\phi(\mathbf{x}, \mathbf{z}) - (\mathbf{x} - \mathbf{z})^T(\nabla\phi(\mathbf{y}) - \nabla\phi(\mathbf{z}))$$

- Generalized Pythagoras Theorem:

$$D_\phi(\mathbf{x}, \mathbf{y}) \geq D_\phi(\mathbf{z}, \mathbf{y}) + D_\phi(\mathbf{x}, \mathbf{z})$$

where $\mathbf{z}$ is the "Bregman" projection onto the convex set Ω. When Ω is an affine set, then it holds with equality

Dissimilarity Measures: Bregman Divergence

- $\phi(\mathbf{x}) = \|\mathbf{x}\|^2$ is strictly convex and differentiable on $\mathbb{R}^m$
  - $d_\phi(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$ [squared Euclidean distance]

- $\phi(\mathbf{p}) = \sum_{j=1}^m p_j \log p_j$ (negative entropy) is strictly convex and differentiable on the m-simplex
  - $d_\phi(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^m p_j \log\left(\frac{p_j}{q_j}\right)$ [KL-divergence]

- $\phi(\mathbf{x}) = -\sum_{j=1}^m \log x_j$ is strictly convex and differentiable on $\mathbb{R}^m_{++}$
  - $d_\phi(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^m \left(\frac{x_j}{y_j} - \log\left(\frac{x_j}{y_j}\right) - 1\right)$ [Itakura-Saito distance]

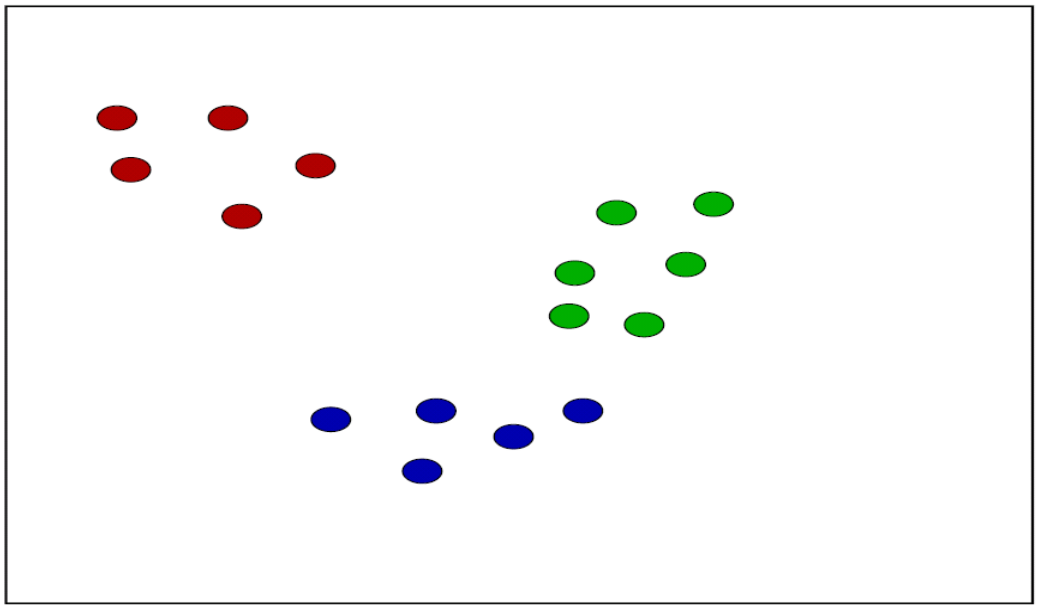
Dissimilarity Measures: Bregman Divergence

Bregman Divergences

Function Name	$\phi(x)$	$\mathrm{dom}\,\phi$	$D_\phi(x;y)$
Squared norm	$\frac{1}{2}x^2$	$(-\infty,+\infty)$	$\frac{1}{2}(x-y)^2$
Shannon entropy	$x\log x-x$	$[0,+\infty)$	$x\log\frac{x}{y}-x+y$
Bit entropy	$x\log x+(1-x)\log(1-x)$	$[0,1]$	$x\log\frac{x}{y}+(1-x)\log\frac{1-x}{1-y}$
Burg entropy	$-\log x$	$(0,+\infty)$	$\frac{x}{y}-\log\frac{x}{y}-1$
Hellinger	$-\sqrt{1-x^2}$	$[-1,1]$	$(1-xy)(1-y^2)^{-1/2}-(1-x^2)^{1/2}$
ℓ_p quasi-norm	$-x^p \quad (0<p<1)$	$[0,+\infty)$	$-x^p+p\,xy^{p-1}-(p-1)\,y^p$
ℓ_p norm	$\lvert x\rvert^p \quad (1<p<\infty)$	$(-\infty,+\infty)$	$\lvert x\rvert^p-p\,x\,\mathrm{sgn}\,y\lvert y\rvert^{p-1}+(p-1)\lvert y\rvert^p$
Exponential	$\exp x$	$(-\infty,+\infty)$	$\exp x-(x-y+1)\exp y$
Inverse	$1/x$	$(0,+\infty)$	$1/x+x/y^2-2/y$

Clustering

- Given: A set of objects
- Goal: Partition the set into clusters of similar objects



ISR

Clustering

K-Means Clustering

- Given: A finite set $\mathcal{X}$ with weights $\pi_{\mathbf{x}}$, $\forall \mathbf{x} \in \mathcal{X}$, desired number of clusters k
- Goal: Find clusters $\{\mathcal{X}_1, \cdots, \mathcal{X}_k\}$ and cluster representatives $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$ that minimize

$$\sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}} \|\mathbf{x} - \boldsymbol{\mu}_h\|^2$$

- Properties
  - Applicable to finite dimensional real vectors
  - NP-complete problem for $k > 2$
  - Practical locally optimal solution: k-means algorithm

Clustering

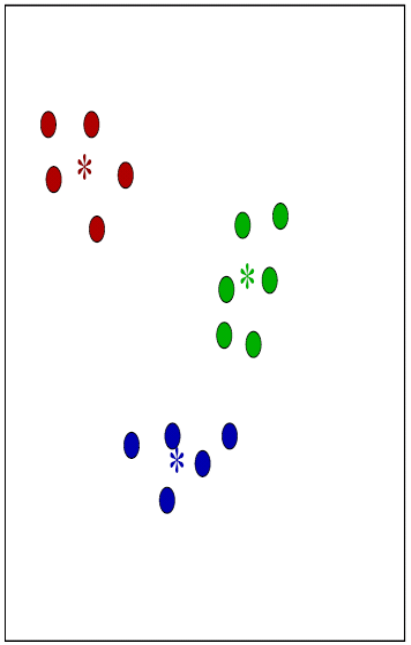
K-Means Algorithm

- Initialize $\{\boldsymbol{\mu}_h\}_{h=1}^{k}$
- Repeat until *convergence*
  - { Assignment Step }
    Assign $\mathbf{x}$ to $\mathcal{X}_h$ if

$$h = \underset{h'}{\operatorname{argmin}} \|\mathbf{x} - \boldsymbol{\mu}_{h'}\|^2$$

  - { Re-estimation step }
    For all h

$$\boldsymbol{\mu}_h = \frac{\sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}} \, \mathbf{x}}{\sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}}}$$



Mean is the best cluster representative !

Clustering

Clustering as Compression

- Original random variable $X \sim \pi_{\mathbf{x}}, \forall \mathbf{x} \in \mathcal{X}$
- Clustering determines
  - clusters $\{\mathcal{X}_1, \cdots, \mathcal{X}_k\}$
  - cluster representatives $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$
  - cluster priors $\{\pi_1, \cdots, \pi_k\}$
- Compressed random variable $\hat{X} \sim \pi_h, \forall \boldsymbol{\mu}_h \in \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$

ISR

Clustering

Information-Theoretic Clustering (ITC)

- Given: Joint probability distribution $p(X, Y)$ where X takes values in a finite set $\mathcal{X}$, desired number of clusters k
- Goal: Find clusters $\{\mathcal{X}_1, \cdots, \mathcal{X}_k\}$ and compressed representation $\hat{X}$ that minimize

$$I(X;Y) - I(\hat{X};Y)$$

- Properties
  - Applicable to probability distributions
  - NP-complete problem
  - Locally optimal solution exists

Clustering

Information Theoretic Clustering

- Objective function

$$I(X,Y) - I(\hat{X},Y) = \sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} p(x) KL(p(Y|x) \| p(Y|h))$$

$$= \sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} \pi_x KL(\mathbf{z}(x) \| \boldsymbol{\mu}_h)$$

- Alternate Formulation
  - Represent each $x \in \mathcal{X}$ by the discrete distribution $\mathbf{z}(x) = p(Y|x)$
  - Cluster the set $\mathcal{Z} = \{\mathbf{z}(x) : x \in \mathcal{X}\}$ based on KL-divergence
- ITC Algorithm
  - Iterative relocation similar to k-means
  - Optimal assignment: closest in terms of KL-divergence
  - Optimal cluster representative: mean distribution !

Clustering

A Closer Look

- K-means objective function:

$$\sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}} \|\mathbf{x} - \boldsymbol{\mu}_h\|^2 = \sum_{\mathbf{x} \in \mathcal{X}} \pi_{\mathbf{x}} \|\mathbf{x} - \boldsymbol{\mu}\|^2 - \sum_{h=1}^{k} \pi_h \|\boldsymbol{\mu}_h - \boldsymbol{\mu}\|^2$$
$$= var(X) - var(\hat{X})$$

where $\boldsymbol{\mu}$ is the global mean

- ITC objective function:

$$\sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} \pi_x KL(\mathbf{z}(x) \| \boldsymbol{\mu}_h) = \sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} p(x) KL(p(Y|x) \| p(Y|h))$$
$$= I(X, Y) - I(\hat{X}, Y)$$

Clustering

General Formulation ?

- Objective function for any distance measure:
  Expected "distance" to cluster representatives = Loss in "information"

$$\sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} \pi_x d(x, \boldsymbol{\mu}_h) = I(X) - I(\hat{X})$$

- Main questions
  - How is information $I(\cdot)$ related to distance $d(\cdot, \cdot)$?
  - Will a simple iterative relocation algorithm work?
  - Is mean still the best cluster representative ?

Clustering

Bregman Information

- Theorem: For all Bregman divergences, mean is the best constant predictor of a random variable and the best single representative for a set of values

$$\boldsymbol{\mu} = \operatorname*{argmin}_{\mathbf{c}} E_{\pi}[d_{\phi}(X, \mathbf{c})] = \operatorname*{argmin}_{\mathbf{c}} \sum_{\mathbf{x} \in \mathcal{X}} \pi_{\mathbf{x}} d_{\phi}(\mathbf{x}, \mathbf{c})$$

where $\boldsymbol{\mu}$ is the global mean

- Definition: The minimum loss is the Bregman information of X

$$I_{\phi}(X) = E_{\pi}[d_{\phi}(X, \boldsymbol{\mu})]$$

Clustering

Bregman Hard Clustering

- Given: $X \sim \pi_{\mathbf{x}},\ \mathbf{x} \in \mathcal{X}$, desired number of clusters k
- Theorem: Expected Bregman divergence to cluster means equals loss in Bregman information

$$\sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}} d_\phi(\mathbf{x}, \boldsymbol{\mu}_h) = I_\phi(X) - I_\phi(\hat{X})$$

- Goal: Find clusters $\{\mathcal{X}_1, \cdots, \mathcal{X}_k\}$, cluster representatives $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$ and compressed representation $\hat{X}$ that optimize

$$\sum_{h=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}} d_\phi(\mathbf{x}, \boldsymbol{\mu}_h) = I_\phi(X) - I_\phi(\hat{X})$$

Clustering

Bregman Hard Clustering Algorithm

- Initialize $\{\boldsymbol{\mu}_h\}_{h=1}^{k}$
- Repeat until *convergence*
  - { Assignment step }
    Assign $\mathbf{x}$ to $\mathcal{X}_h$ if

$$h = \underset{h'}{\operatorname{argmin}}\ d_\phi(\mathbf{x}, \boldsymbol{\mu}_{h'})$$

  - { Re-estimation step }
    For all h

$$\boldsymbol{\mu}_h = \frac{\sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}}\ \mathbf{x}}{\sum_{\mathbf{x} \in \mathcal{X}_h} \pi_{\mathbf{x}}}$$

Clustering

Properties

- Guarantee: Monotonically decreases objective function till convergence
- Scalability: Every iteration is linear in the size of the input
- Exhaustiveness: If such an algorithm exists for a loss function $L(\mathbf{x}, \boldsymbol{\mu})$, then L has to be a Bregman divergence
- Linear Separators: Clusters are separated by hyperplanes
- Mixed Data types: Allows appropriate Bregman divergence for subsets of features

ISR

Clustering

Exponential Families

Definition: A multivariate parametric family of distributions is called an exponential family if the probability density function is of the form

$$p_{\psi,\theta}(\mathbf{x}) = \exp\{\langle \mathbf{x}, \theta \rangle - \psi(\theta) - \lambda(\mathbf{x})\}$$

- ψ is the *cumulant* or *log-partition* function that uniquely determines a family, e.g., Gaussian, multinomial, Poisson, etc.
- θ fixes a particular distribution in the family
- ψ is a strictly convex function

Exponential families include most of the popular generative models

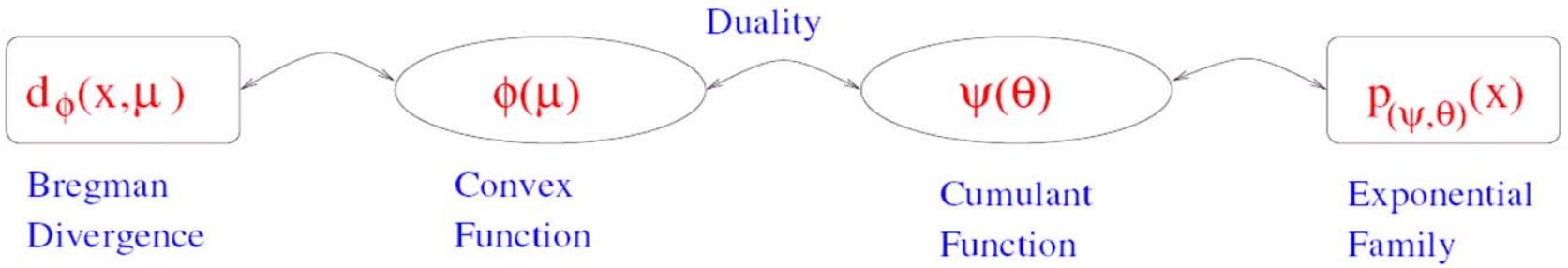
Clustering

Bijection Theorem

Theorem: There is a bijection between exponential densities $p_{(\psi,\theta)}(\cdot)$ and Bregman divergences $d_\phi(\cdot, \boldsymbol{\mu})$

$$p_{(\psi,\theta)}(\mathbf{x}) = \exp(-d_\phi(\mathbf{x}, \boldsymbol{\mu}))f_\phi(\mathbf{x}),$$

- where ϕ and ψ are Legendre duals and $\boldsymbol{\mu} = \nabla\psi(\theta)$, $\theta = \nabla\phi(\boldsymbol{\mu})$
- f_ϕ is a uniquely determined function



Clustering

Bijection between BD and Exponential Family

Regular exponential families $\leftrightarrow$ Regular Bregman divergences

Gaussian	$\leftrightarrow$	Squared Loss
Multinomial	$\leftrightarrow$	KL-divergence
Geometric	$\leftrightarrow$	Itakura-Saito distance
Poisson	$\leftrightarrow$	I-divergence

Clustering

Bregman Soft Clustering

- Learning a *mixture of exponential densities*
  - Maximum likelihood under incomplete information
  - Solvable by Expectation Maximization (EM)

- Bijection theorem

$p_{\psi,\theta}(x)$	$\leftrightarrow$	$\exp(-d_\phi(\mathbf{x}, \boldsymbol{\mu}))f_\phi(\mathbf{x})$
log-likelihood	$\leftrightarrow$	negative Bregman divergence
maximum likelihood	$\leftrightarrow$	minimum Bregman divergence
EM for mixture of exponential densities	$\leftrightarrow$	soft clustering for Bregman divergences

Clustering

Bregman Soft Clustering

- Given: $X \sim \pi_{\mathbf{x}},\ \mathbf{x} \in \mathcal{X}$ and desired number of clusters k
- Goal: Find parameters $\Theta = \{\pi_h, \boldsymbol{\mu}_h\}_{h=1}^{k}$ that maximize

$$\mathcal{L}(\Theta|\mathcal{X}) = \sum_{\mathbf{x}\in\mathcal{X}} \pi_{\mathbf{x}} \log \left(\sum_{h=1}^{k} \pi_h \exp(-d_\phi(\mathbf{x}, \boldsymbol{\mu}_h)) \right)$$

- Efficient Expectation Maximization is possible

Clustering

Bregman Soft Clustering Algorithm

- Initialize $\{\pi_h, \boldsymbol{\mu}_h\}_{h=1}^{k}$
- Repeat until *convergence*
  - { Expectation step }
    For all $\mathbf{x}, h$,
    $$p(h|\mathbf{x}) = \pi_h \exp(-d_\phi(\mathbf{x}, \boldsymbol{\mu}_h))/Z(\mathbf{x}),$$
    where $Z(\mathbf{x})$ is the log-partition function
  - { Maximization step }
    For all h,
    $$\pi_h = \sum_{\mathbf{x}\in\mathcal{X}} \pi_\mathbf{x} p(h|\mathbf{x})$$
    $$\boldsymbol{\mu}_h = \frac{\sum_{\mathbf{x}\in\mathcal{X}} \pi_\mathbf{x} p(h|\mathbf{x})\ \mathbf{x}}{\sum_\mathbf{x} \pi_\mathbf{x} p(h|\mathbf{x})}$$

Clustering

Main Results: Clustering

- Hard Clustering
  - KMeans-type algorithm possible for any Bregman divergence
- Hard Clustering $\leftrightarrow$ Soft Clustering
  - Bregman divergences $\leftrightarrow$ Exponential family distributions
- Soft Clustering
  - Efficient learning of mixtures of exponential family distributions

Clustering and Compression with BD

Compression Vs Loss in Bregman Information

- Theorem: Expected distortion = Loss in Bregman Information

$$E[d_\phi(X, \hat{X})] = I_\phi(X) - I_\phi(\hat{X})$$

- Rate distortion problem as a trade-off

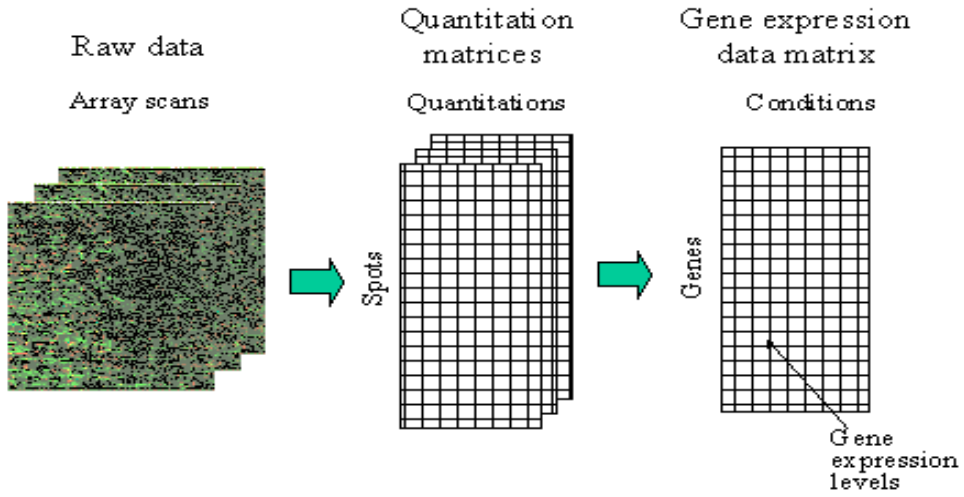
$$\min_{p(\hat{X}|X)} [I(X;\hat{X}) + \beta E[d_\phi(X, \hat{X})]] \equiv \min_{p(\hat{X}|X)} [I(X;\hat{X}) - \beta I_\phi(\hat{X})]$$

- Information Bottleneck is a special case

$$\min_{p(\hat{X}|X)} [I(X;\hat{X}) - \beta I(\hat{X};Y)]$$

LVQ with BD: Gene Expression Microarrays

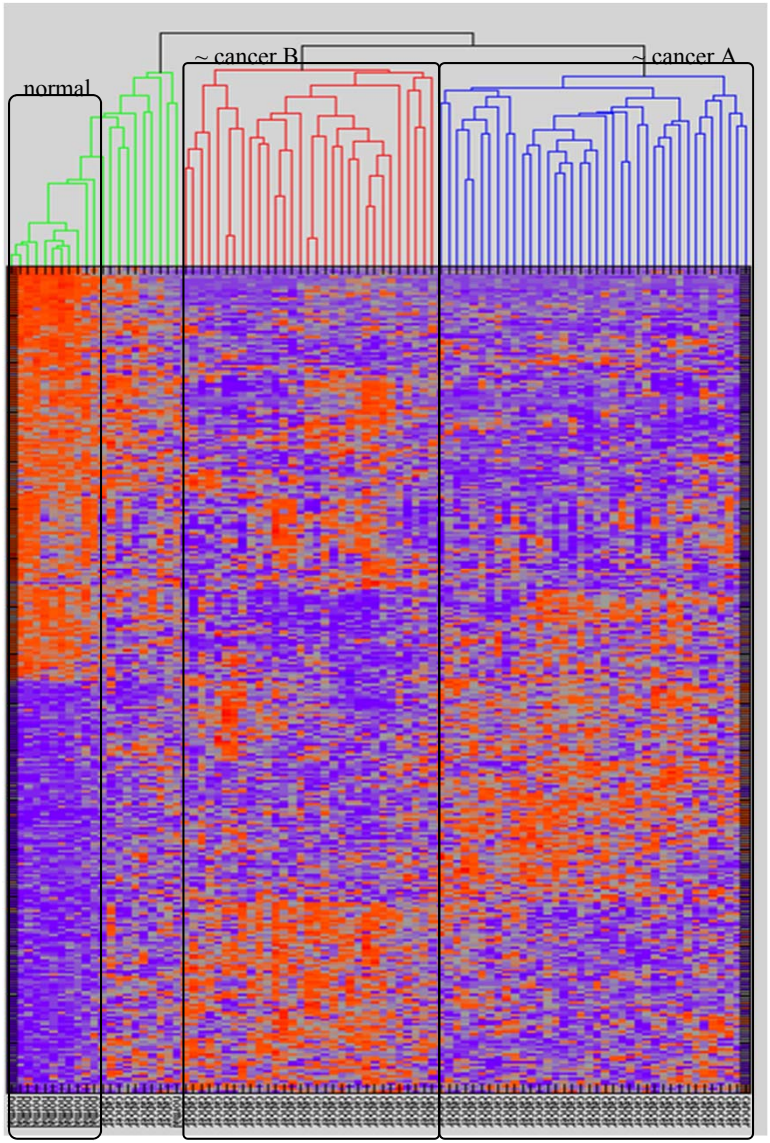
- **High-throughput experimental techniques that detect the expression of thousands of genes in a tissue simultaneously.**
- **Presence of DNA is detected by fluorescence following laser excitation..**



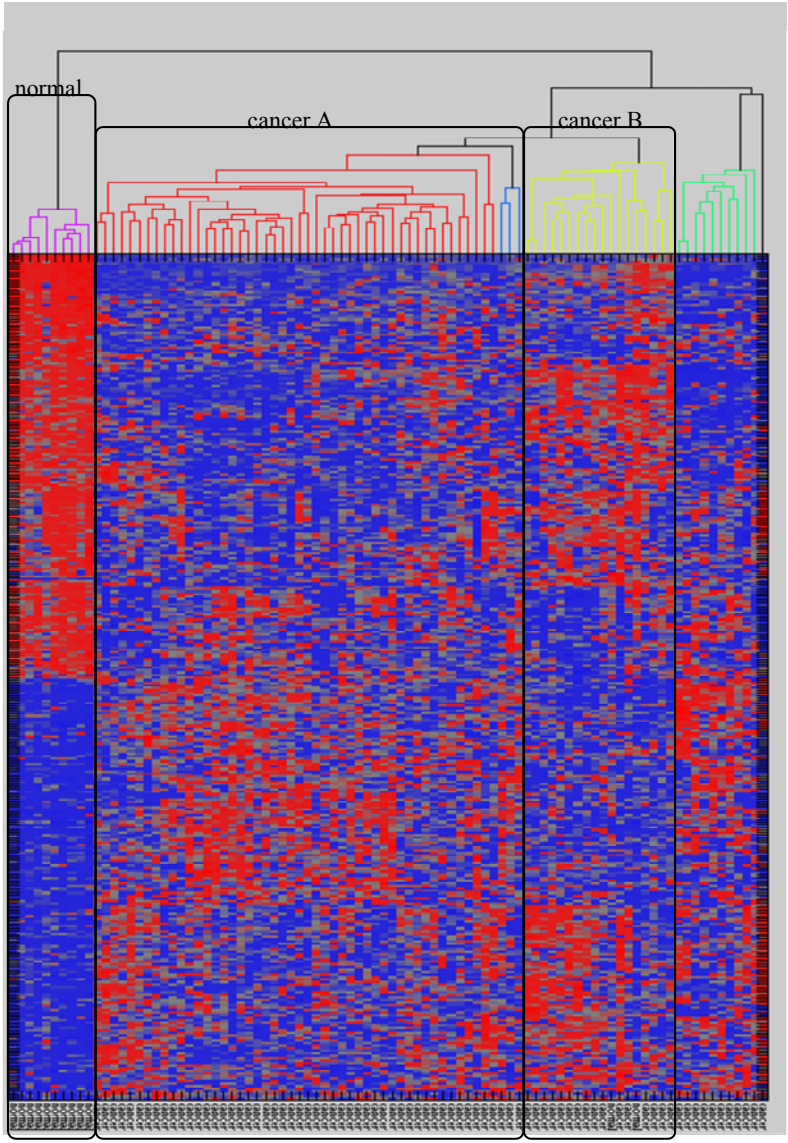
Applications:

- Identification of functions for newly discovered sequences.
- Drug discovery and toxicology.
- Mutation or single nucleotide polymorphism (SNP) detection

Comparison of Clustering Solutions



Correlation Coefficient



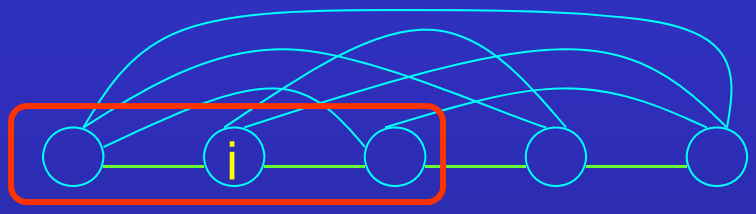
Un-normalized* K-L

Self-Organizing Map (SOM – Kohonen)

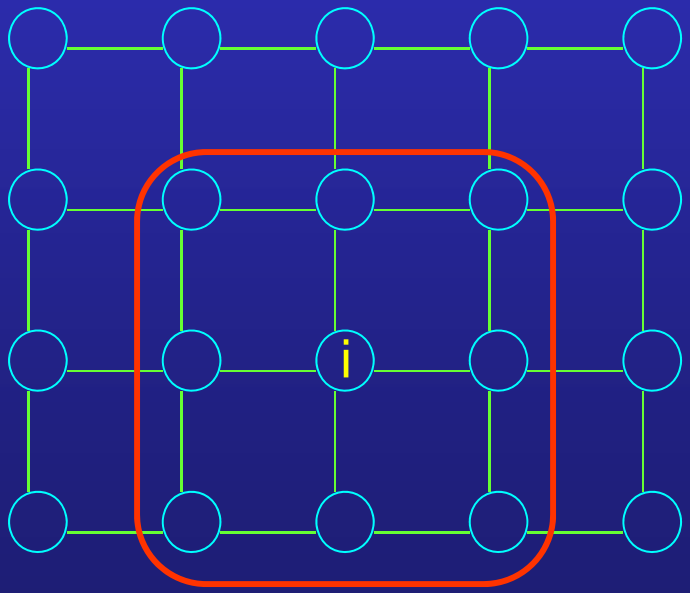
- In the human cortex, multi-dimensional sensory input spaces (e.g., visual input, tactile input) are represented by two-dimensional maps.
- The projection from sensory inputs onto such maps is topology conserving.
- This means that neighboring areas in these maps represent neighboring areas in the sensory input space.
- For example, neighboring areas in the sensory cortex are responsible for the arm and hand regions.

Self-Organizing Map (SOM – Kohonen)

Common output-layer structures:



One-dimensional
(completely interconnected for determining "winner" unit)

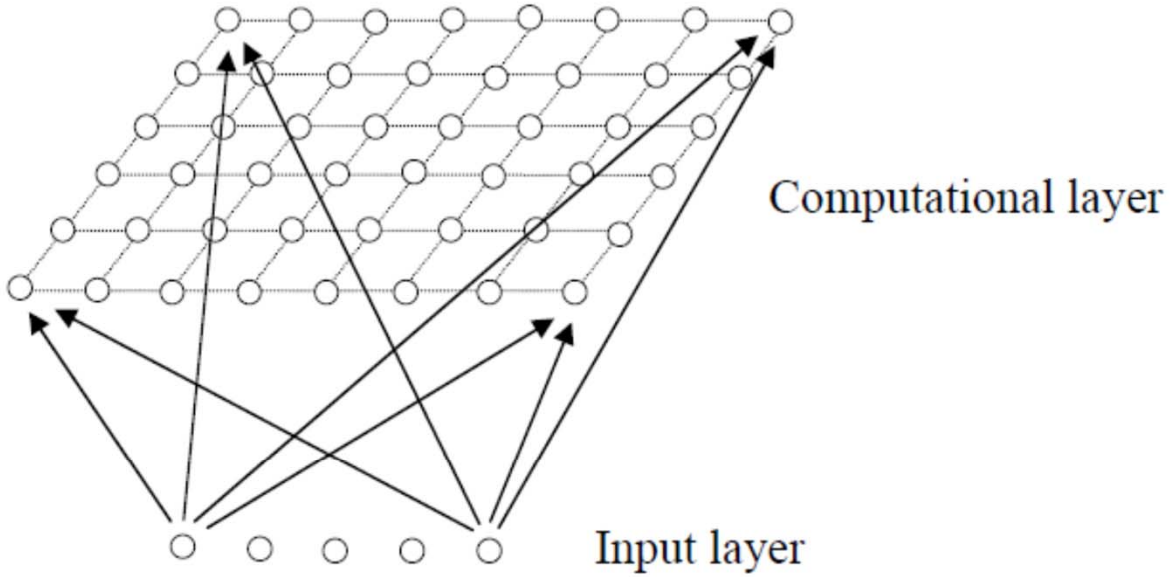


Two-dimensional
(connections omitted, only neighborhood relations shown [green])

Neighborhood of neuron i

Kohonen Networks

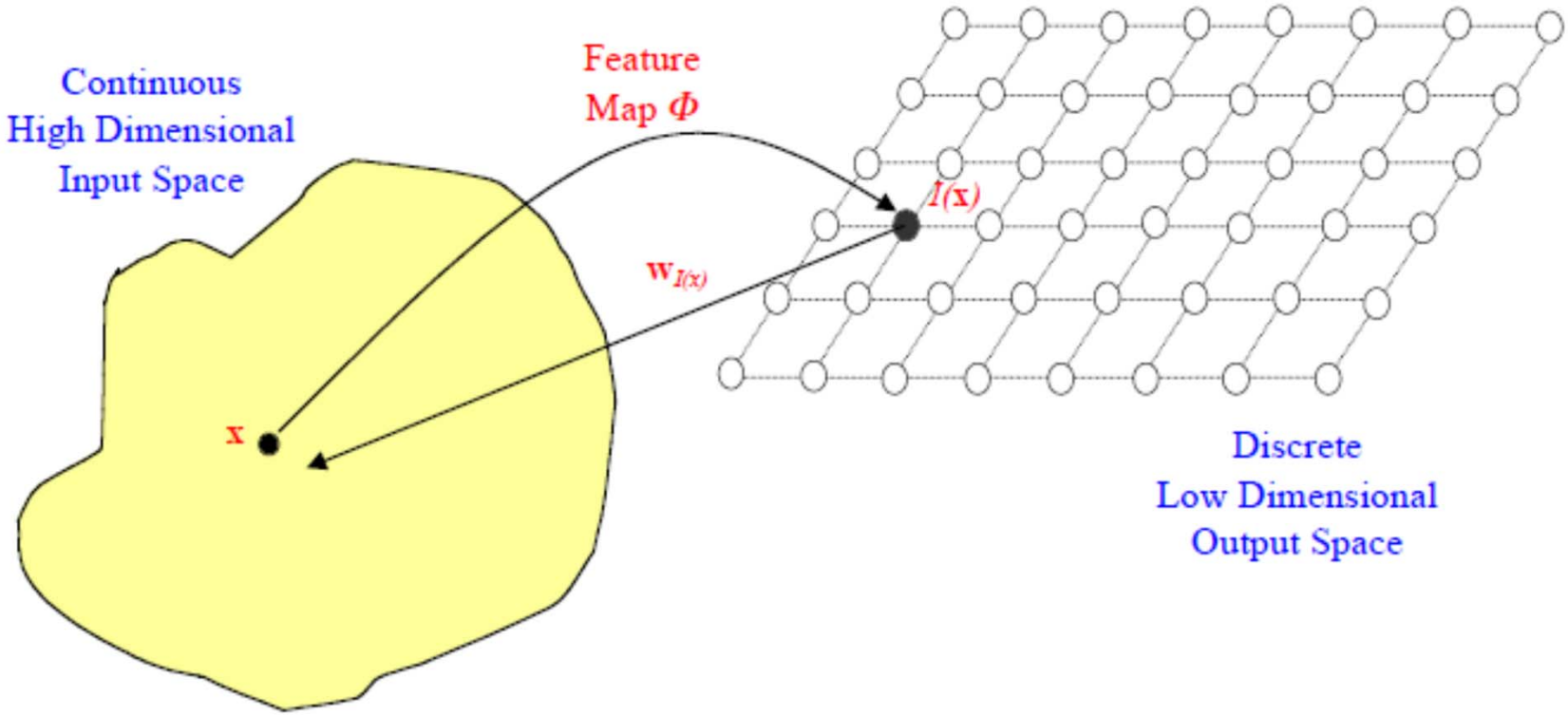
SOM has a feed-forward structure with a single computational layer arranged in rows and columns. Each neuron is fully connected to all the source nodes in the input layer:



Clearly, a one dimensional map will just have a single row (or a single column) in the computational layer.

Kohonen Mapping

We have points $\mathbf{x}$ in the input space mapping to points $I(\mathbf{x})$ in the output space:



Each point I in the output space will map to a corresponding point $\mathbf{w}(I)$ in the input space.

Unsupervised Learning: SOM Algorithm

- **The operation of the algorithm is summarized as follows:**
  1. *Initialization:* Choose random values for the initial weight vectors $\mathbf{w}_j(0)$. The weight vectors must be different for all neurons. Usually keep the magnitude of the weights small.
  2. *Sampling:* Draw a sample $\mathbf{x}$ from the input space with a certain probability; the vector $\mathbf{x}$ represents the activation pattern that is applied to the lattice. The dimension of $\mathbf{x}$ is equal to m.
  3. *Similarity Matching:* Find the best-matching (winning) neuron $i(\mathbf{x})$ at time step n by using the minimum Euclidean distance criterion:
     - $\mathbf{i(x) = \arg\min_j \|x - w_j\|, j = 1,2,\ldots,l}$

Summary of SOM Algorithm

4. *Updating:* Adjust the synaptic weight vectors of all neurons by using the update formula:

- $$\mathbf{w_j(n+1) = w_j(n) + \eta(n)\, h_{ji(x)}(n)\, (x(n) - w_j(n))}$$

Where $\eta(n)$ is the learning rate and $h_{ji(x)}(n)$ is the neighborhood function around the winner neuron $i(\mathbf{x})$; both $\eta(n)$ and $h_{ji(x)}(n)$ are varied dynamically for best results.

5. *Continuation:* Continue with step 2 until no noticeable changes in the feature map are observed.

Overview of SOM Algorithm

We have a spatially ***continuous input space***, in which our input vectors live. The aim is to map from this to a low dimensional spatially ***discrete output space***, the topology of which is formed by arranging a set of neurons in a grid. Our SOM provides such a non-linear transformation called a ***feature map***.

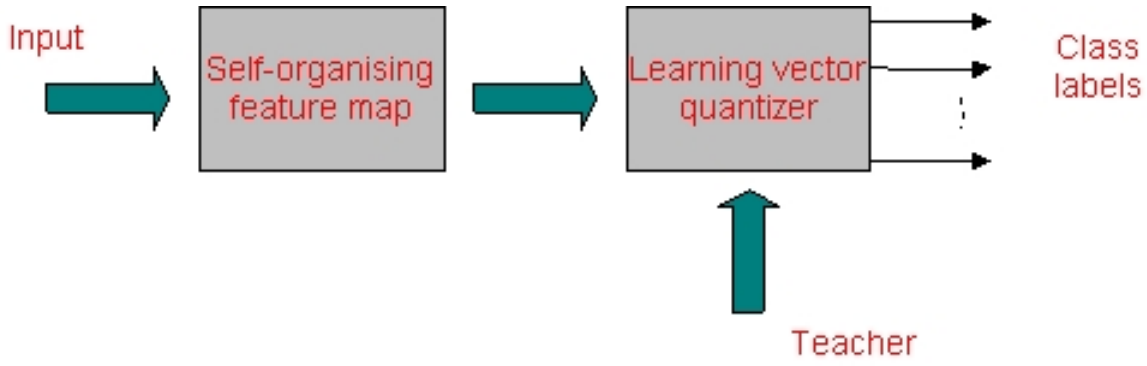
The stages of the SOM algorithm can be summarised as follows:

1. ***Initialization*** – Choose random values for the initial weight vectors $\mathbf{w}_j$.
2. ***Sampling*** – Draw a sample training input vector $\mathbf{x}$ from the input space.
3. ***Matching*** – Find the winning neuron $I(\mathbf{x})$ with weight vector closest to input vector.
4. ***Updating*** – Apply the weight update equation $\Delta w_{ji} = \eta(t)\, T_{j,I(\mathbf{x})}(t)\, (x_i - w_{ji})$.
5. ***Continuation*** – keep returning to step 2 until the feature map stops changing.

$$T_{j,I(\mathbf{x})} = \exp(-S_{j,I(\mathbf{x})}^2 / 2\sigma^2) \qquad \eta(t) = \eta_0 \exp(-t/\tau_\eta)$$

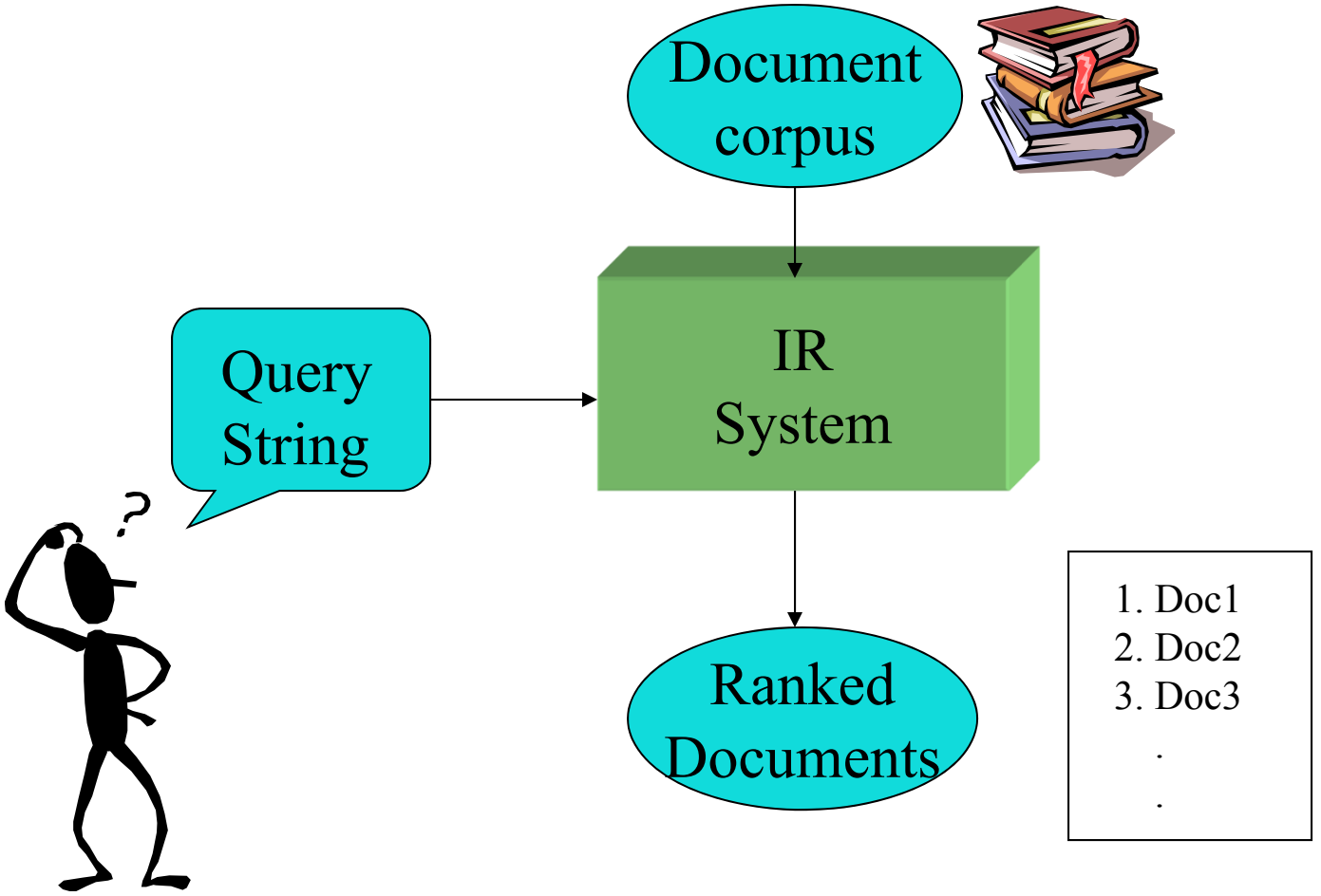
SOM followed by LVQ

Computation of the feature map can be viewed as the first of two stages for adaptively solving a pattern classification problem as shown below. The second stage is provided by the learning vector quantization, which provides a method for fine tuning of a feature map. This is useful and typical for DNN



Semantic Vector Spaces

Information Retrieval System



Semantic Vector Spaces

The Vector-Space Model

- Assume t distinct terms remain after preprocessing; call them index terms or the vocabulary.
- These "orthogonal" terms form a vector space.

  Dimension = t = |vocabulary|
- Each term, i, in a document or query, j, is given a real-valued weight, w_{ij}.
- Both documents and queries are expressed as t-dimensional vectors:

  $d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$

Semantic Vector Spaces

Term Weights: Term Frequency

- More frequent terms in a document are more important, i.e. more indicative of the topic.

$$f_{ij} = \text{frequency of term } i \text{ in document } j$$

- May want to normalize *term frequency* (*tf*) by dividing by the frequency of the most common term in the document:

$$tf_{ij} = f_{ij} / max_i\{f_{ij}\}$$

Semantic Vector Spaces

Term Weights: Inverse Document Frequency

- Terms that appear in many *different* documents are *less* indicative of overall topic.

  df_i = document frequency of term i
  = number of documents containing term i

  idf_i = inverse document frequency of term i,
  = $\log_2 (N/ df_i)$
  (N: total number of documents)

- An indication of a term's *discrimination* power.
- Log used to dampen the effect relative to *tf*.

Semantic Vector Spaces

TF-IDF Weighting

- A typical combined term importance indicator is *tf-idf weighting*:

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N/ df_i)$$

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight.
- Many other ways of determining term weights have been proposed.
- Experimentally, *tf-idf* has been found to work well.

Semantic Vector Spaces

Similarity Measure

- A similarity measure is a function that computes the *degree of similarity* between two vectors.

- Using a similarity measure between the query and each document:
  - It is possible to rank the retrieved documents in the order of presumed relevance.
  - It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

Semantic Vector Spaces

Vector-Space (Distributional) Lexical Semantics

- Represent word meanings as points (vectors) in a (high-dimensional) Euclidian space.
- Dimensions encode aspects of the context in which the word appears (e.g. how often it co-occurs with another specific word).
  - "You will know a word by the company that it keeps." (J.R. Firth, 1957)
- Semantic similarity defined as distance between points in this semantic space.

Semantic Vector Spaces

Other Feature Weights

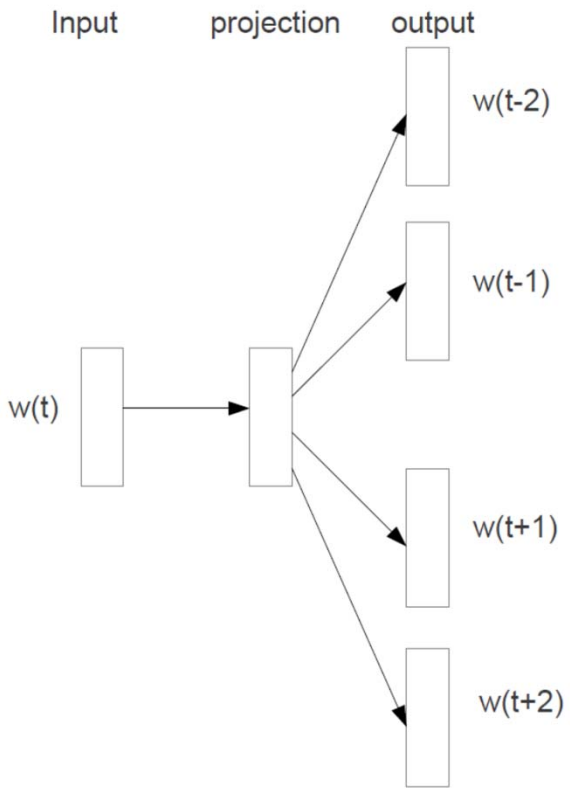
- Replace TF-IDF with other feature weights.
- *Pointwise mutual information* (PMI) between target word, *w*, and the given feature, *f*:

$$PMI(f,w) = log \frac{P(f,w)}{P(f)p(w)}$$

Semantic Vector Spaces

Neural Word2Vec (Mikolov et al., 2013)

- Learn an "embedding" of words that supports effective prediction of surrounding "skip gram" of words.



Semantic Vector Spaces

Skip-Gram Word2Vec Network Architecture

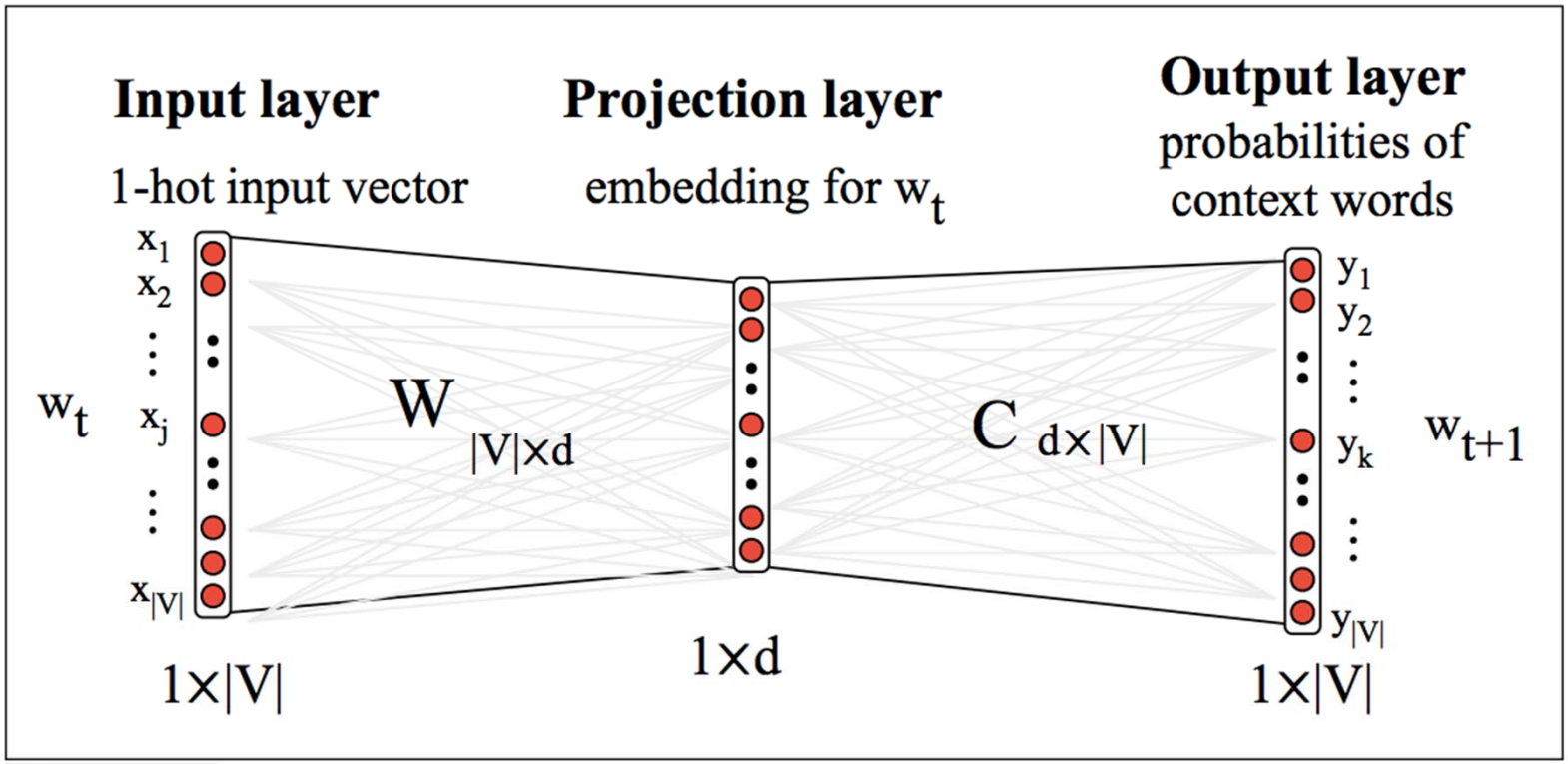


Figure 16.5 The skip-gram model viewed as a network (Mikolov et al. 2013, Mikolov et al. 2013a).

Semantic Vector Spaces

Word2Vec Math

- Softmax classifier predicts surrounding words from a word embedding.

$$\log p(o|c) = \log \frac{\exp\left(u_o^T v_c\right)}{\sum_{w=1}^{W} \exp\left(u_w^T v_c\right)}$$

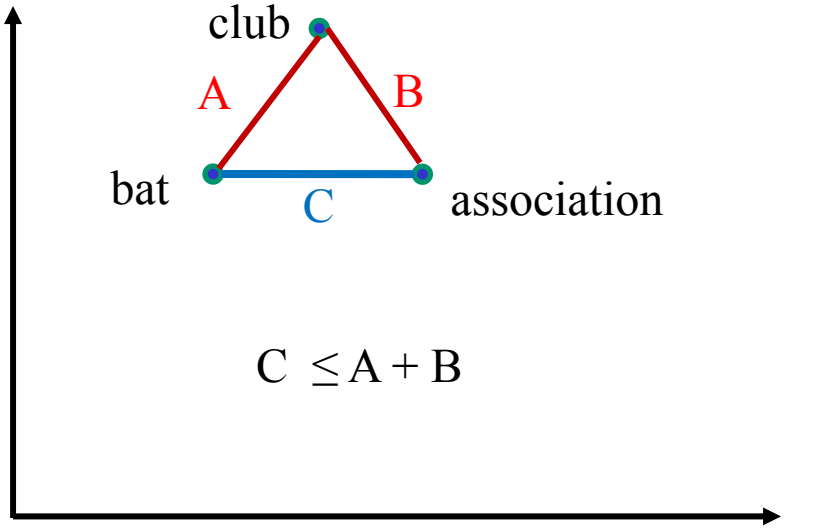
- Train to maximize the probability of skip-gram predictions.

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j}|w_t)$$

Semantic Vector Spaces

Word Sense and Vector Semantics

- Having one vector per word ignores the impact of homonymous senses.
- Similarity of ambiguous words violates the triangle inequality.



Semantic Vector Spaces

Vector-Space Word Meaning in Context

- Compute a semantic vector for an individual occurrence of a word based on its context.
- Combine a standard vector for a word with vectors representing the immediate context.

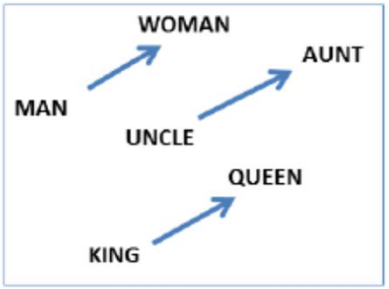
Compositional Vector Semantics

- Compute vector meanings of phrases and sentences by combining (composing) the vector meanings of its words.
- Simplest approach is to use vector addition or component-wise multiplication to combine word vectors.
- Evaluate on human judgements of sentence-level semantic similarity (*semantic textual similarity*, STS, SemEval competition).

Semantic Vector Spaces

Other Vector Semantics Computations

- Compute meanings of words by mathematically combining meanings of other words (Mikolov, et al., 2013)
  - $\overrightarrow{king} = \overrightarrow{queen} - \overrightarrow{female} + \overrightarrow{male}$

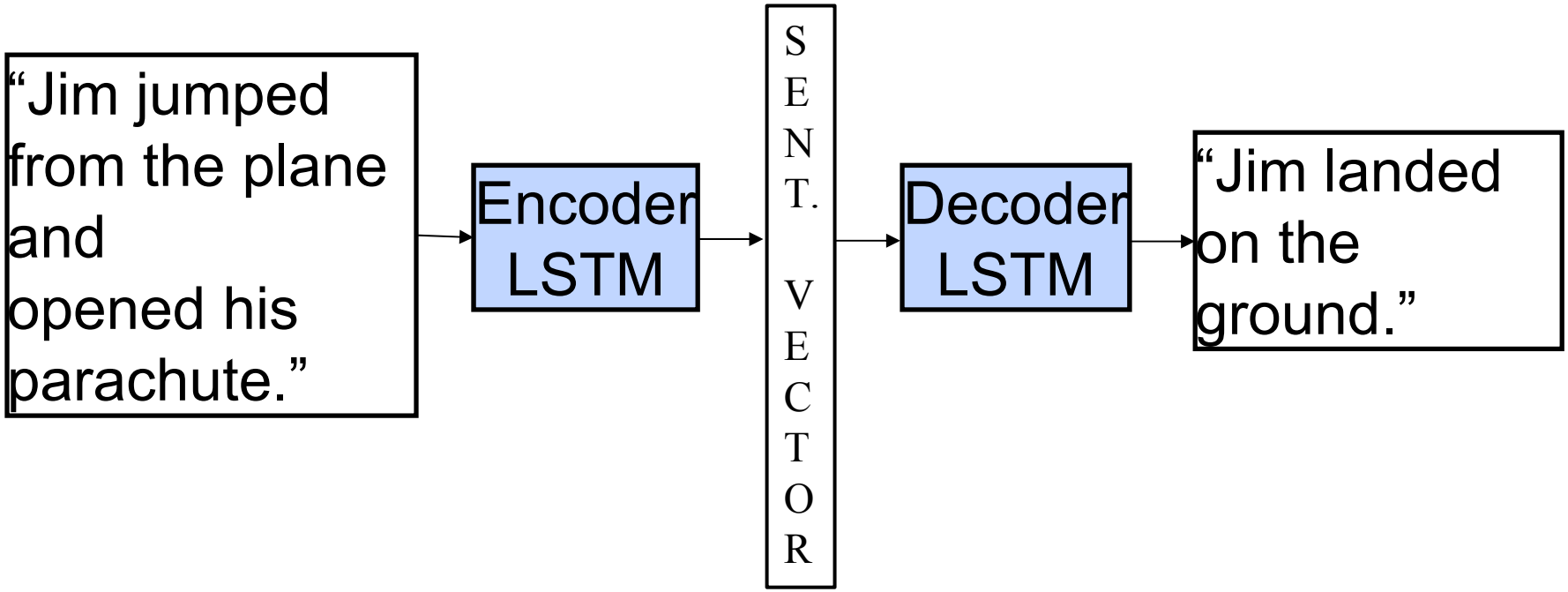


- Evaluate on solving word analogies
  - King is to queen as uncle is to _____?

Semantic Vector Spaces

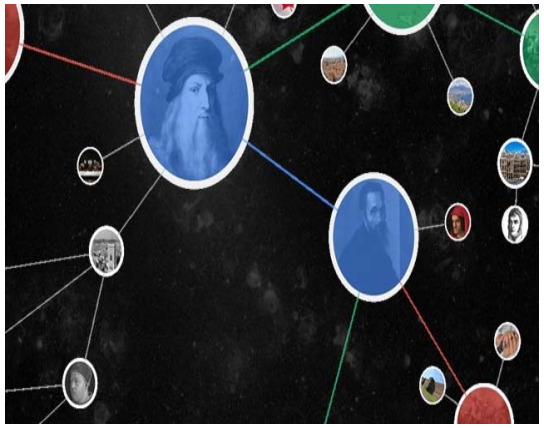
Sentence-Level Neural Language Models

- "Skip-Thought Vectors" (Kiros et al., NIPS 2015)
  - Use LSTMs to encode whole sentences into lower-dimensional vectors.
  - Vectors trained to predict previous and next sentences.

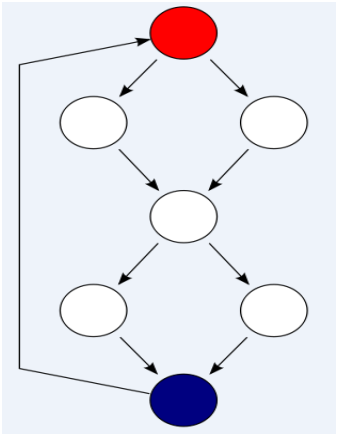


Knowledge Graphs

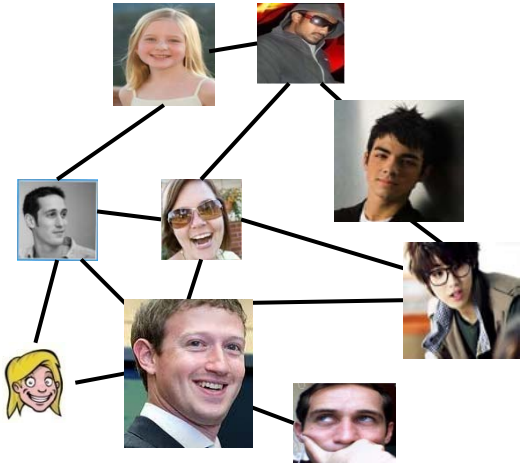
Data as big graphs



Knowledge Graph



Program Flow/ Callgraphs



Social Network

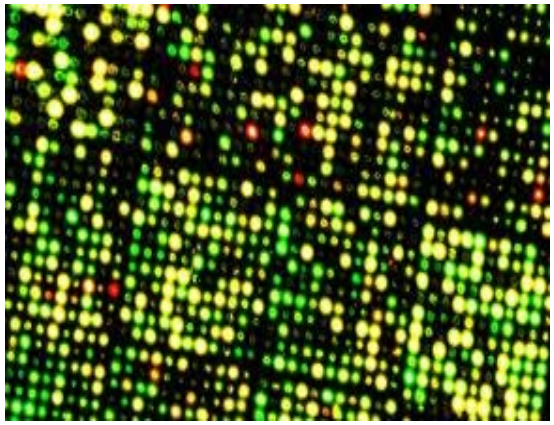
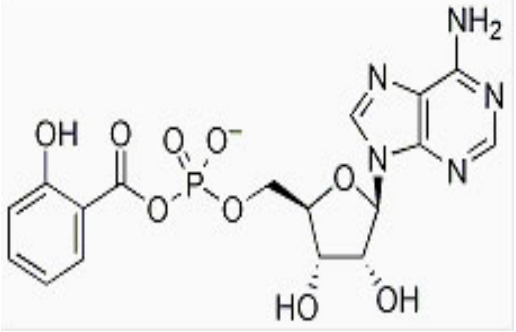


Image Data



Chemical Structure

Biological Network

Knowledge Graphs

Semantic Approaches

Propositional:

- "dog bites man" ➔ bites(dog, man)
- bites(*,*) is a binary relation. man, dog are objects.
- Probabilities can be attached.

Vector representation:

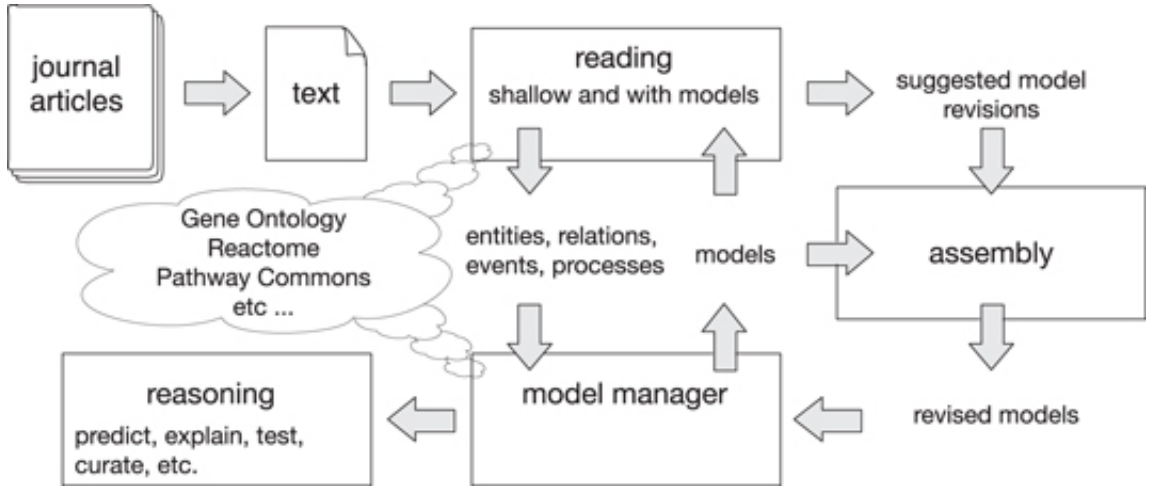
- vec("dog bites man") = (0.2, -0.3, 1.5,…) $\in \Re^n$
- Sentences similar in meaning should be close to this embedding (e.g. use human judgments)

Knowledge Graphs

Propositional Semantics

- Allow logical inferences "Socrates is a man," + "all men are mortal" → "Socrates is mortal"
- Important for inference in well-defined domains, e.g. inferring ***gene regulation*** from medical journals.

"Big Mechanism" project



From "DARPA's Big Mechanism program" Paul R Cohen, Phys. Biol. 12 (2015)

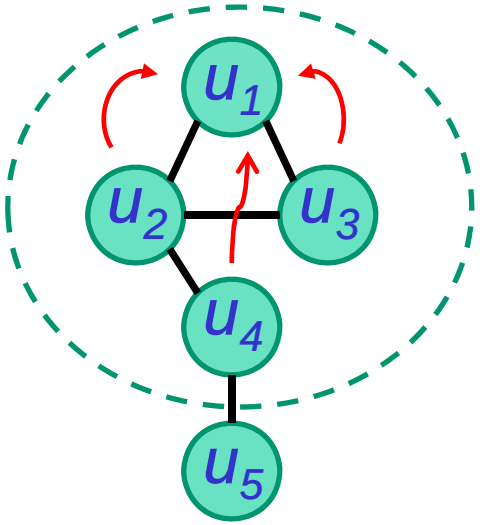
Knowledge Graphs

Need to Integrate Semantic Vector Space Models and Knowledge Graphs (Hypergraphs) Models

- Convert the h-hop neighborhood of each node u in G into a multi-dimensional vector $R_G(u)=\{\langle u', w_u(u')\rangle\}$, based on the distance of neighbor nodes u' from u.

$$w_u(u') = \begin{cases} \alpha^{d(u,u')}, d(u,u') \le h; \\ 0, \quad \text{otherwise.} \end{cases}$$

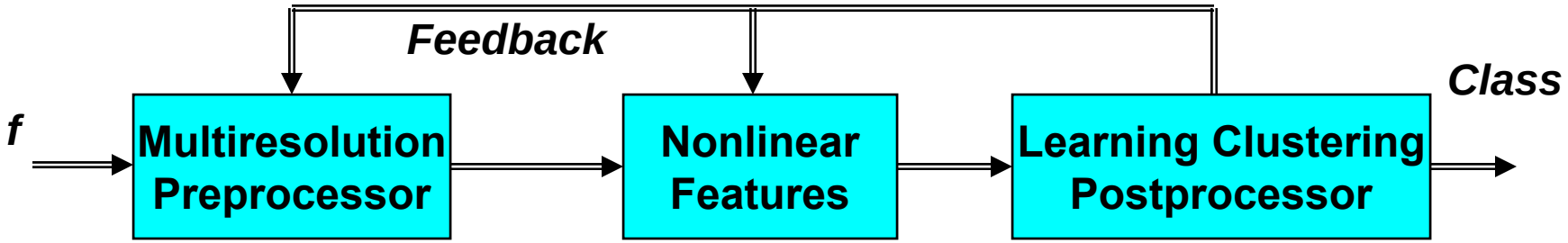
Distance between u and u'



Information Propagation

Previous Applications of Information Propagation:
Semi-supervised Learning [AI' 08], Concept Propagation [CIKM '06]

Summary: Multiresolution and Progressive Learning Clustering

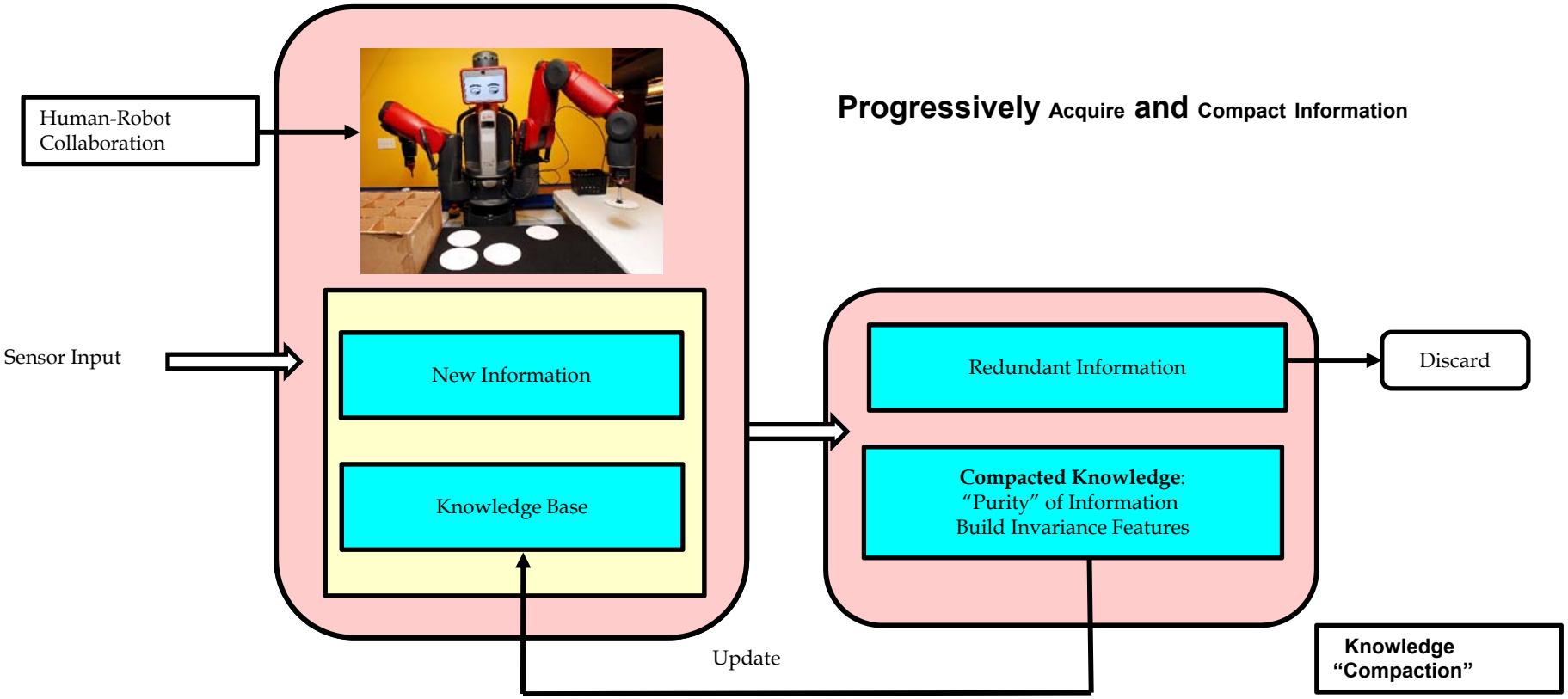


- Address both the hierarchical organization of signal databases and progressive classification:
  - Combine a multiresolution preprocessor with a learning clustering postprocessor
  - Feedback is also an option
- Resulting algorithms proved to have some "universal" qualities
- Found analogs of such algorithms in animals and humans:
  - Hearing and sound classification
  - Vision and identification of objects by humans
- Most promising mathematical formulation of the problem:

  combined compression and classification for general signals

Summary, Current Work

Progressive Machine Learning, Knowledge "Compaction", Human-Robot Collaboration



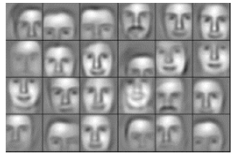
Summary, Current Work

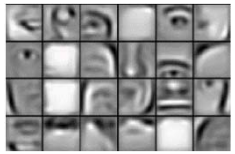
How do humans learn?

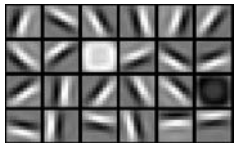
Progressive attention

Visual cortex processes images in **multi-scale fashion**

Understanding Symmetries and Invariant Features

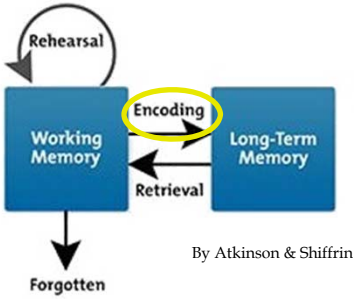






Can recognize (e.g. faces) using only **few samples**

Knowledge compaction

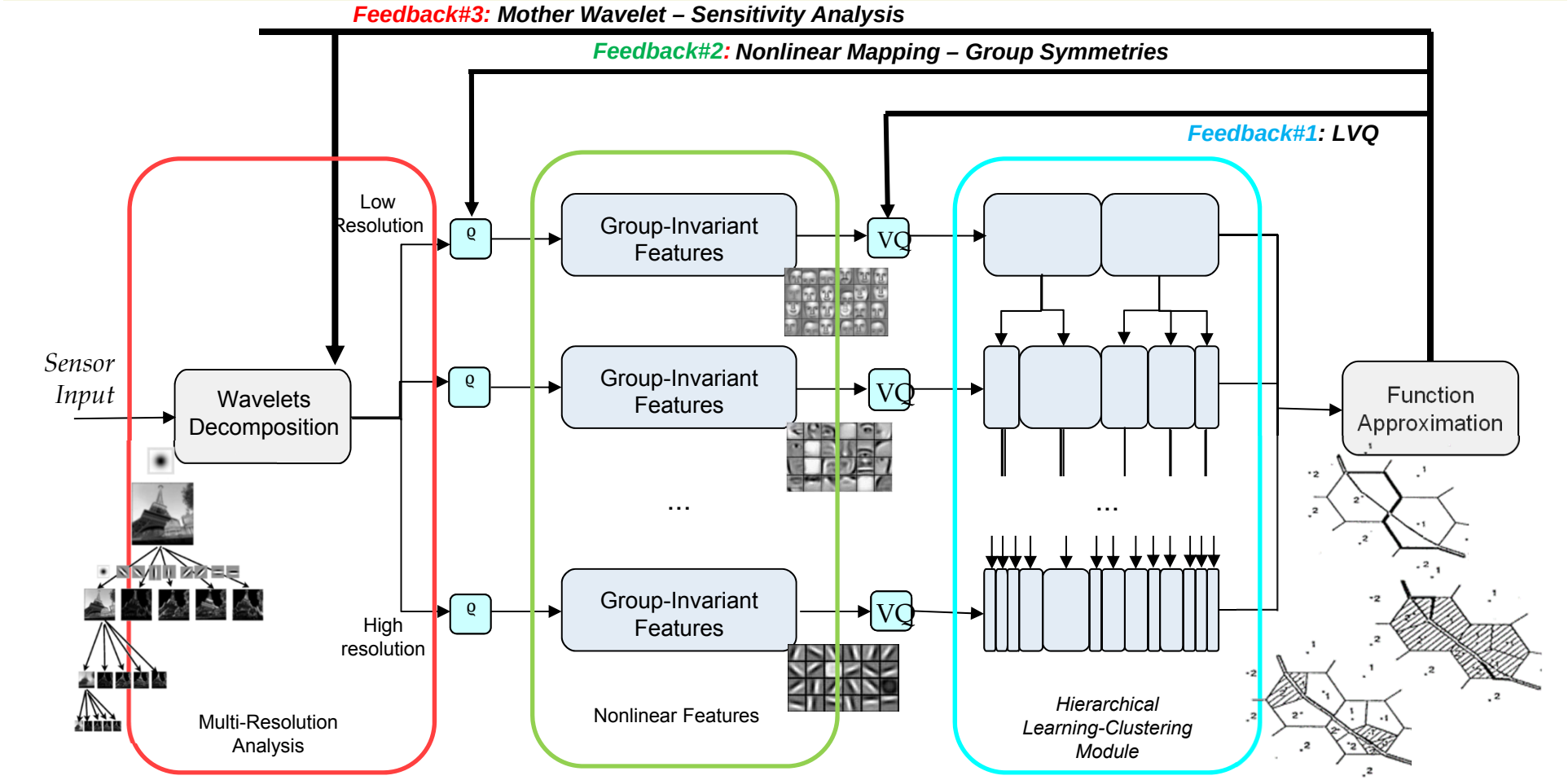


By Atkinson & Shiffrin

Do not memorize every new sample

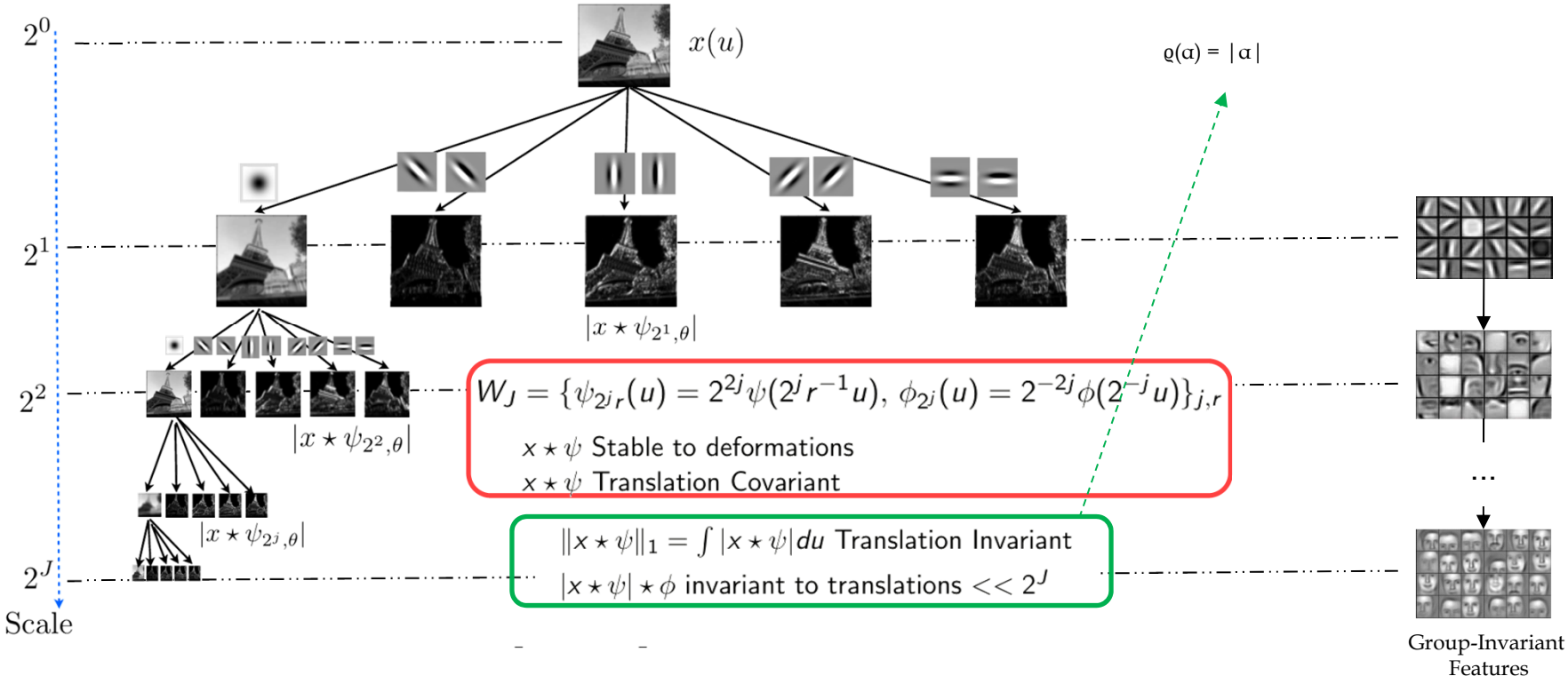
Summary, Current Work

Proposed Universal Machine Learning Architecture



Summary, Current Work

Multi-Scale Directional Wavelets & *L1* Norm
in Image Classification

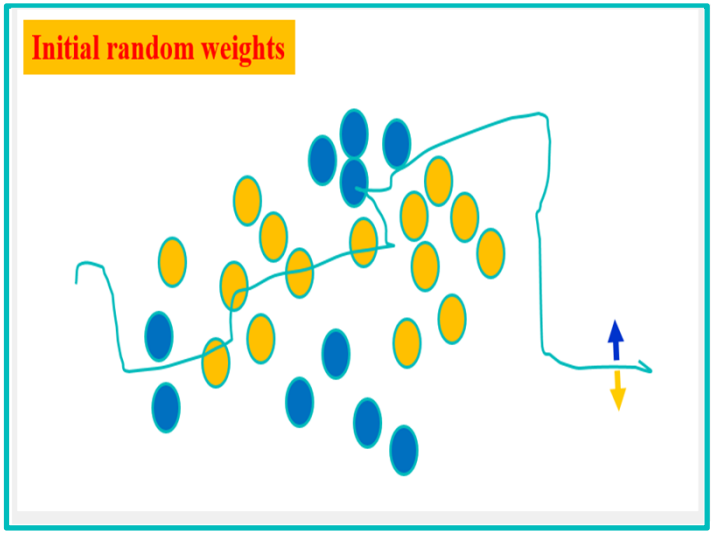


Mallat et. al., "Scattering Networks", …

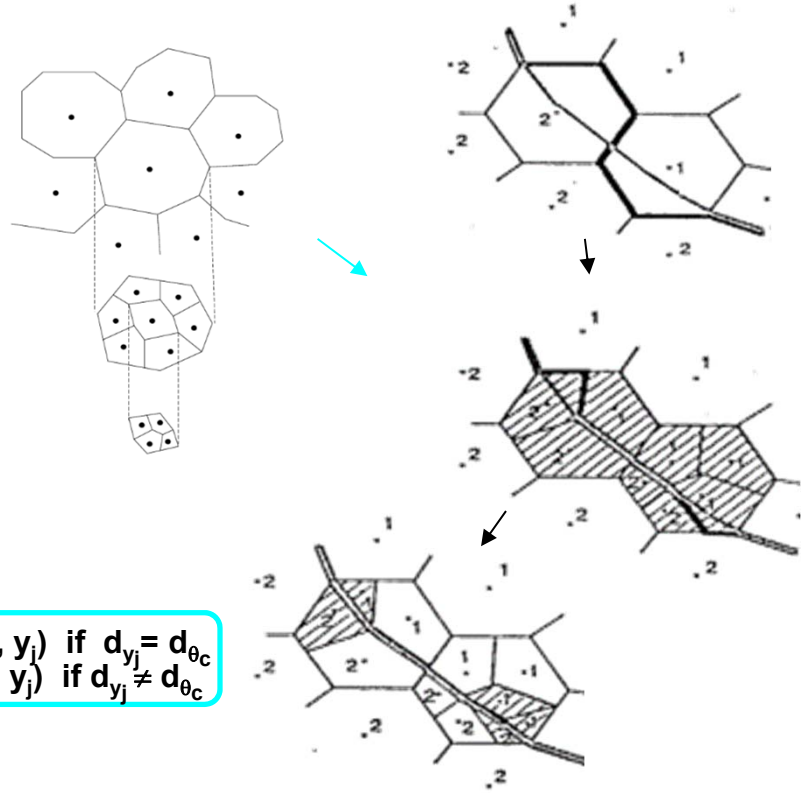
Summary, Current Work

Learning Vector Quantization

- **Non-Parametric → No Model Errors**



Hierarchical Structure



Z = training data = $\{(y_n, d_{y_n})\}_{n=1}^{N}$

Voronoi vectors $= \Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$

decisions = $\{d_{\theta_1}, d_{\theta_2}, \Lambda, d_{\theta_k}\}$

$$\theta_c(n+1) = \theta_c(n) - \alpha_n \nabla_\theta \rho(\theta_c(n), y_j) \quad \text{if } d_{y_j} = d_{\theta_c}$$
$$\theta_c(n+1) = \theta_c(n) + \alpha_n \nabla_\theta \rho(\theta_c(n), y_j) \quad \text{if } d_{y_j} \neq d_{\theta_c}$$

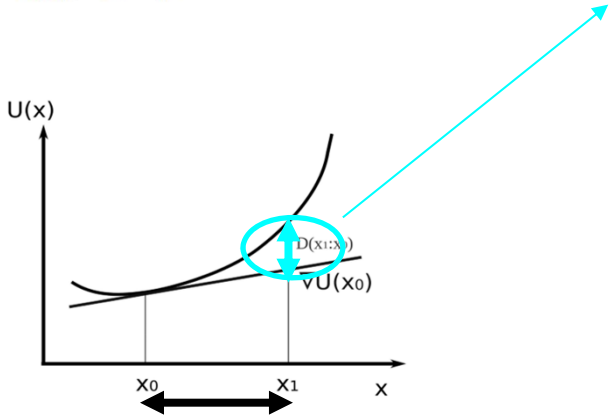
Baras and LaVigna, **Convergence of LVQ**, 1990

Summary, Current Work

Dissimilarity Measures: Bregman Divergences

Bounding the Error Probabilities using Hueffding's Inequality

$$P_{FA} \leq e^{-2nD^2(p_1||p_0)/c^2}$$



Can handle vectors of:
i) **Numerical values**
ii) **Logical values**
iii) **Rule values**

$$D_\varphi(\mathbf{x},\mathbf{y}) = \varphi(\mathbf{x}) - \varphi(\mathbf{y}) - (\mathbf{x}-\mathbf{y})^T \nabla\varphi(\mathbf{y})$$

φ real-valued, strictly convex

- $\phi(x) = \|x\|^2$ is strictly convex and differentiable on $\mathbb{R}^m$, $D_{\phi(x,y)} = \|x-y\|^2$ [squared Euclidean distance]
- $\phi(p) = \sum_{j=1}^m p_j \log p_j$ is strictly convex and differentiable on the m-simplex, $D_{\phi(p,q)} = \sum_{j=1}^m p_j \log \frac{p_j}{q_j}$ [Unnormalized Kullback-Leibler Divergence]
- If $\sum_{j=1}^m p_j = 1$, then $\phi(p)$ becomes the negative entropy and $D_{\phi(p,q)}$ is the KL-divergence

Unnormalized Kullback-Leibler Divergence between K-vectors

$$\delta(\mathbf{a},\mathbf{b}) = D(\mathbf{a}\,||\,\mathbf{b}) - \sum_{k=1}^K b(k) + \sum_{k=1}^K a(k)$$

$$= \sum_{k=1}^K b(k) \log \frac{b(k)}{a(k)} - \sum_{k=1}^K b(k) + \sum_{k=1}^K a(k)$$

Summary, Current Work

Applications for sound sensing and understanding

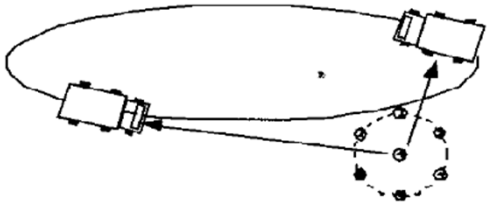
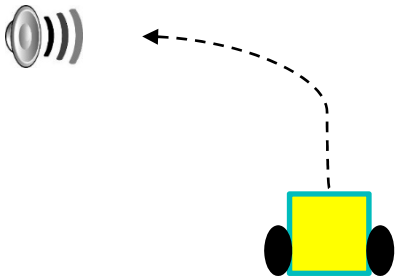


Figure 1: Schematic of the situation: two heavy vehicles go around a track while their acoustic signature is recorded by an array of microphones.

Vehicle Classification Using Acoustic Data

Musical Instrument Classification



Robot Navigation using Sound Signals

[1] Baras, et.al. , Vehicle Classification Using Acoustic Data Based on Biology Hearing Models r and Multiscale Vector Quantization
[2] Shammai, Encoding Sound Timbre in the Auditory System

Summary, Current Work

Applications for vision sensing and understanding

Face Recognition

Task-Driven Scene Understanding (find the cup)

Go for finer details

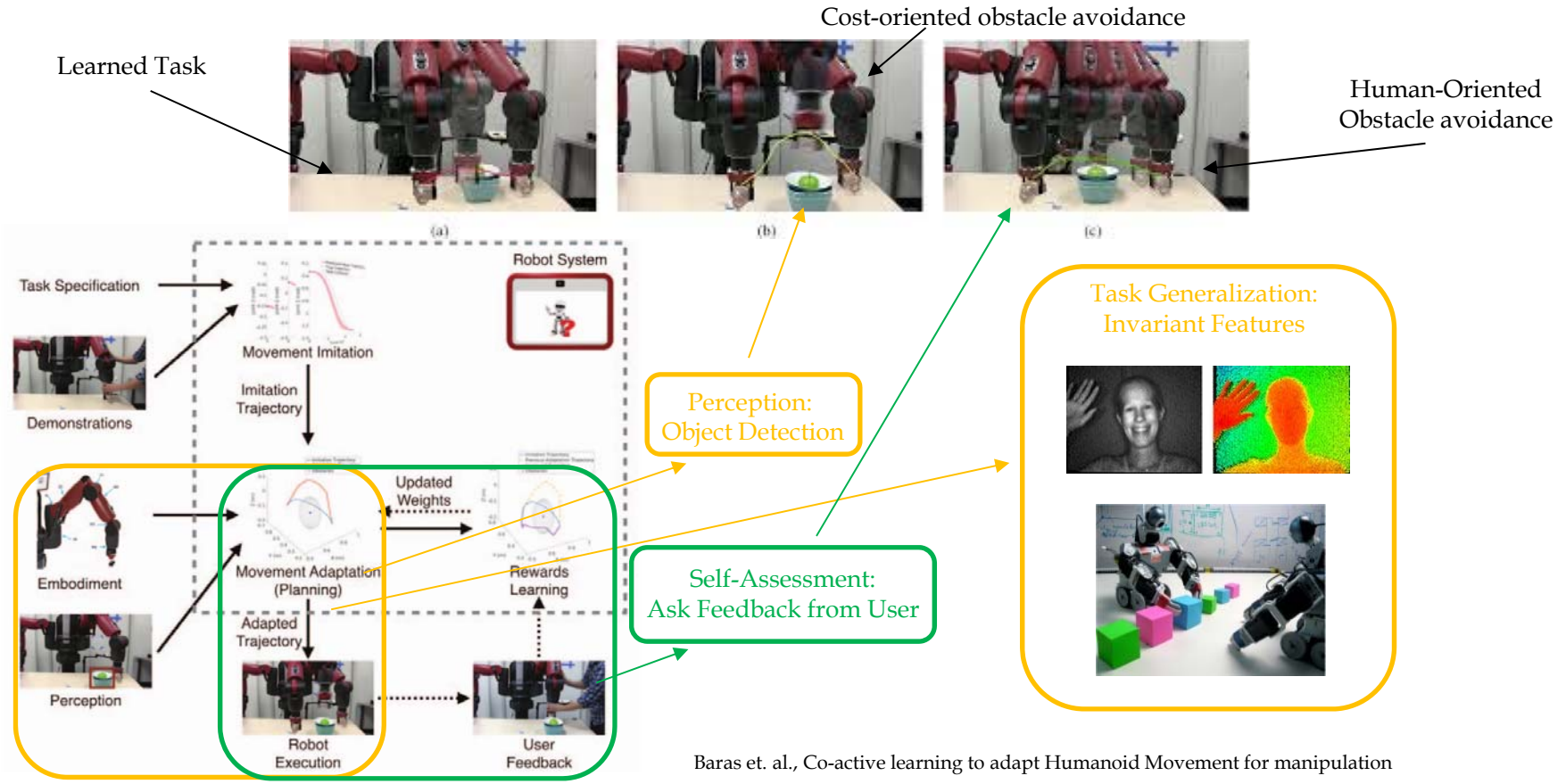
Detected!

No need for finer details

Attention Mechanism

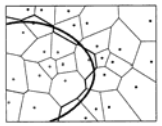
Summary, Current Work

Applications in Human-Robot Collaboration
Learning by demonstration with safety constraints



Baras et. al., Co-active learning to adapt Humanoid Movement for manipulation

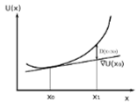
Summary, Current Work



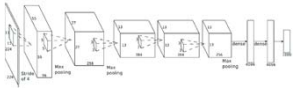
Universality of LVQ & SOM

Sound Sensing and Understanding:
Comparison with the State of the Art

Advantages of Bregman Divergences



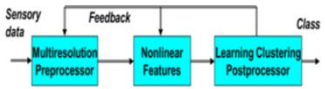
Vision sensing and understanding:
Comparison with the State of the Art



Mapping to existing Deep Architectures:
Universality of the proposed architecture

Attention mechanisms in ground UAV collaborative navigation

Foundation of Information ('knowledge')
Compaction and Progressive Learning



Human-Robot Collaboration for
safe learning in manipulation tasks

Future Directions and Significance

- Implement architecture in hybrid (analog and digital) chips -- Cortex on a chip
- Current competition for DNN chips – We are looking for the bigger challenge:
- Same architecture combines computing and storage as in the brains of higher level animals and humans – brain like computers
- True departure form von-Neumann architectures that separate memory and CPU hardware
- "von Neumann bottleneck" -- severely limits the applications of artificial intelligence systems
- In contrast, signals are processed in neurobiological networks via trillions of synapses with integrated logic and memory functions in a massively parallel mode. And the synapses are constantly modified in a parallel learning process to automatically optimize and simultaneously develop new functions in the networks.

Future Directions and Significance

- Plan to introduce recent innovation of synaptic resistors based on ion-doped polymer composites loaded with carbon nanotube -- opens a new opportunity to emulate neurobiological networks with parallel signal processing and learning capabilities
- Integrate Semantic Vector Space Models and Knowledge Graph Models and use them for progressive knowledge compaction and hardware implementations
- Our architecture enhanced with such synapses can lead to a scaled-up neuromorphic network that can circumvent the curse of dimensionality to process and learn from huge data sets with speed, power efficiency, and memory capacity exponentially superior to those of Si-based computing circuits.
- Looking to implement a cortex on a chip for sound processing and demonstrate it in 3 yrs from now

References http://dev-baras.pantheonsite.io/

J. S. Baras and A. LaVigna, "Convergence of Kohonen's Learning Vector Quantization" *Proceedings of the 1990 IJCNN Conference on Neural Networks*, pp. 476-482, San Diego, CA, June 17-21, 1990

J.S. Baras and A. LaVigna, "Convergence of a Neural Network Classifier", *Proc. 29th IEEE CDC*, pp. 1735-40, 1990.

A. LaVigna, *Nonparametric Classification Using Learning Vector Quantization*, Ph.D. Thesis, EE, UMD, Fall 1989.

J.S. Baras and S.I. Wolk, "Hierarchical Wavelet Representations of Ship Radar Returns" *to appear in IEEE Transactions on Signal Processing*. Expanded version also published as NRL Report NRL/FR/5755-93-9593.

J.S. Baras and S.I. Wolk, "Efficient Organization of Large Ship Radar Databases Using Wavelets and Structured Vector Quanitzation" *Proc. 27th Annual Asilomar Conference,* Vol. 1, pp. 491-498, 1993, Pacific Grove, CA.

J.S. Baras and S.I. Wolk, "Model Based Automatic Target Recognition from High Range Resolution Radar Returns", *Proc. SPIE Intern. Symp. on Intelligent Information Systems,* Vol. 2234, pp. 57-66, Orlando, FL, April 5-8, 1994

J.S. Baras and S.I. Wolk, "Wavelet Based Progressive Classification of High Range Resolution Radar Returns", *Proc. SPIE Intern. Symp. on Intelligent Information Systems,* Vol. 2242, pp. 967-977, Orlando, FL, April 5-8, 1994

J.S. Baras and S.I. Wolk, "Wavelet Based Progressive Classification with Learning: Applications to Radar Signals" *Proc. SPIE 1995 Intern. Symp. Aerospace/ Defense*, Vol. 2491, Part 1 of 2, pp. 339-350, Orlando, Florida, 1995

J.S. Baras and S.I. Wolk, "Wavelet-Based Hierarchical Organization of Image Data: Applications to ISAR and Face Recognition", *Proc. Conf. Information Sciences and Systems*, Johns Hopkins Univ., Baltimore, MD, 1997

J.S. Baras and S.I. Wolk, "Wavelet-Based Hierarchical Organization of Large Image Databases: ISAR and Face Recognition" *Proc. SPIE 12th Intern. Symp. Aerospace, Defense*, Orlando, Florida, April 13-17, 1998.

J.S. Baras and S. Dey, "Combined Compression and Classification with Learning Vector Quantization", *IEEE Transactions on Information Theory*, Vol. 45, No. 6, pp. 1911-1920, September 1999.

J.S. Baras and V.S. Borkar, "A Learning Algorithm for Markov Decision Processes with Adaptive State Aggregation," Proceedings 39th IEEE CDC, Sydney, December 2000.

J.S. Baras and A.S. Baras, "A Novel Nonparametric Discrimination Measure for Analysis of Gene Expression Data," SIAM Conference on the Life Sciences, Portland, July 2004.

ISR

Thank you!

baras@isr.umd.edu

301-405-6606

http://dev-baras.pantheonsite.io/

Questions?