BENG 207 Special Topics in Bioengineering

Neuromorphic Integrated Bioelectronics

Week 8: Energy Conservation

Gert Cauwenberghs

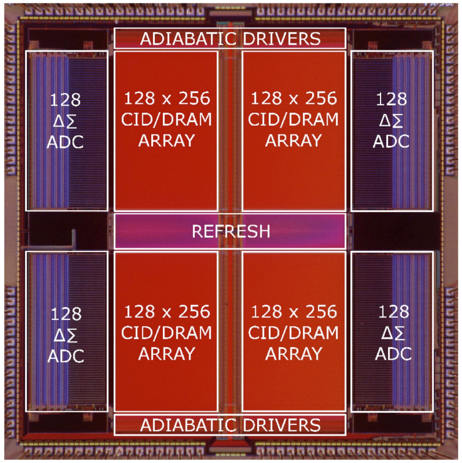
Department of Bioengineering
UC San Diego

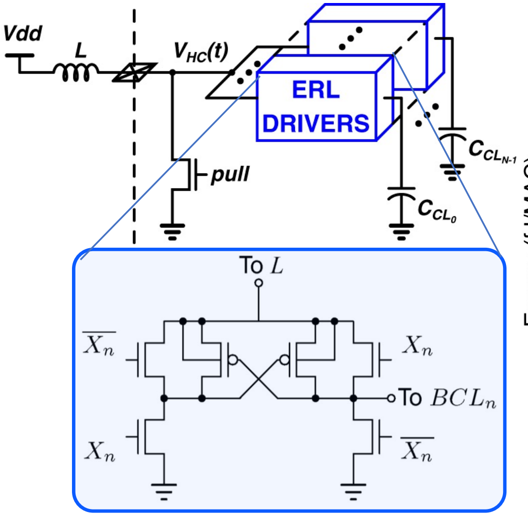
http://isn.ucsd.edu/courses/beng207

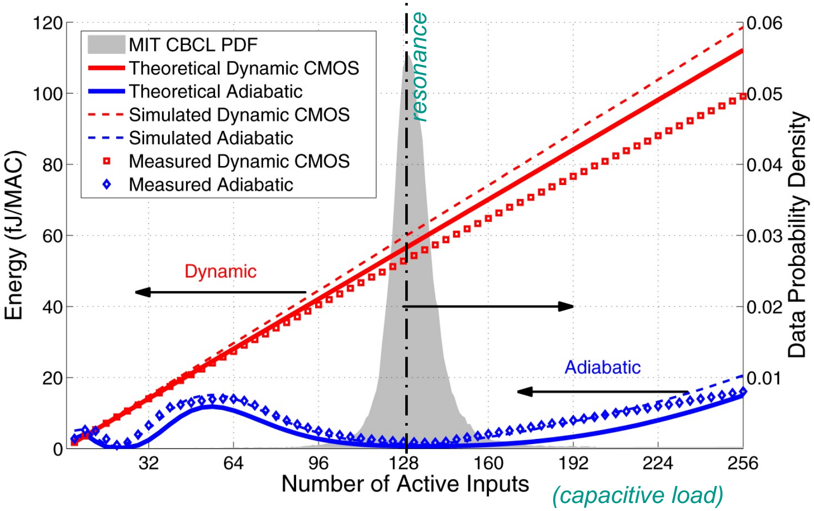
BENG 207 Neuromorphic Integrated Bioelectronics

Date	Topic
9/27, 9/29	Biophysical foundations of natural intelligence in neural systems. Subthreshold MOS silicon models of membrane excitability. Silicon neurons. Hodgkin-Huxley and integrate-and-fire models of spiking neuronal dynamics. Action potentials as address events.
10/4, 10/6	Silicon retina. Low-noise, high-dynamic range photoreceptors. Focal-plane array signal processing. Spatial and temporal contrast sensitivity and adaptation. Dynamic vision sensors.
10/11, 10/13	Silicon cochlea. Low-noise acoustic sensing and automatic gain control. Continuous wavelet filter banks. Interaural time difference and level difference auditory localization. Blind source separation and independent component analysis.
10/18, 10/20	Silicon cortex. Neural and synaptic compute-in-memory arrays. Address-event decoders and arbiters, and integrate-and-fire array transceivers. Hierarchical address-event routing for locally dense, globally sparse long-range connectivity across vast spatial scales.
10/28, 11/1	Review. Modular and scalable design for neuromorphic and bioelectronic integrated circuits and systems. Design for full testability and controllability.
11/1, 11/3	Midterm due 11/2. Low-noise, low-power design. Fundamental limits of noise-energy efficiency, and metrics of performance. Biopotential and electrochemical recording and stimulation, lab-on-a-chip electrophysiology, and neural interface systems-on-chip.
11/8, 11/10	Learning and adaptation to compensate for external and internal variability over extended time scales. Background blind calibration of device mismatch. Correlated double sampling and chopping for offset drift and low-frequency noise cancellation.
11/15, 11/17	Energy conservation. Resonant inductive power delivery and data telemetry. Ultra-high efficiency neuromorphic computing. Resonant adiabatic energy-recovery charge-conserving synapse arrays.
11/22, 11/24	Guest lectures
11/29, 12/1	Project final presentations. All are welcome!

Resonant Adiabatic Energy Recovery

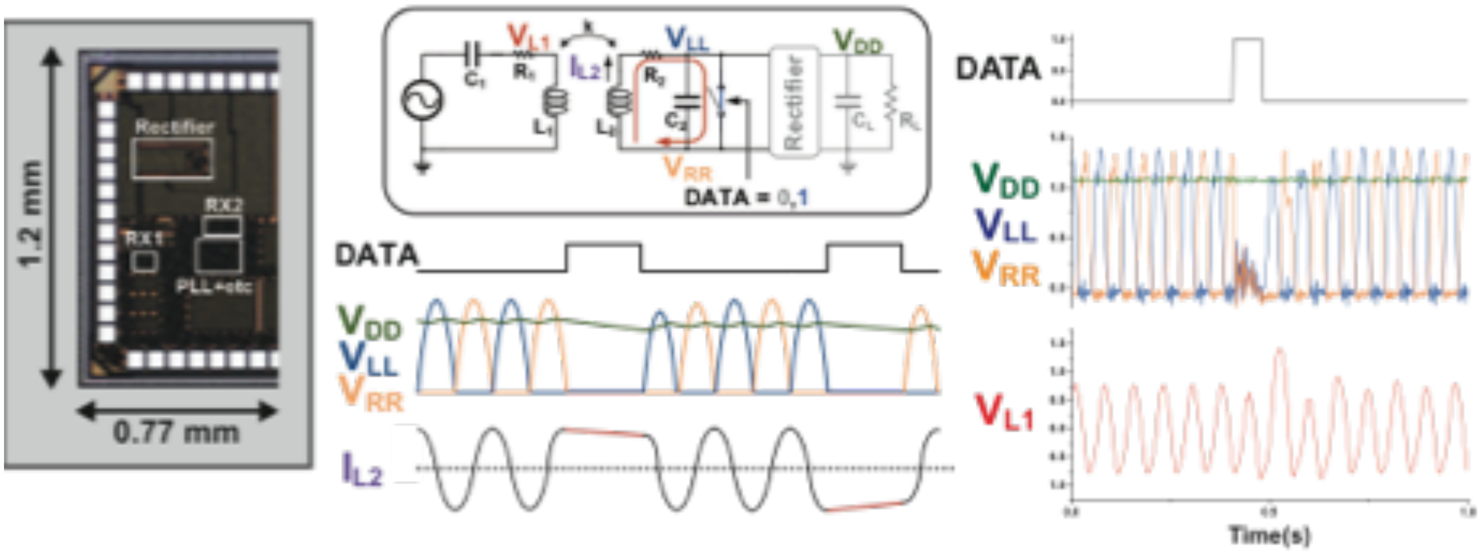






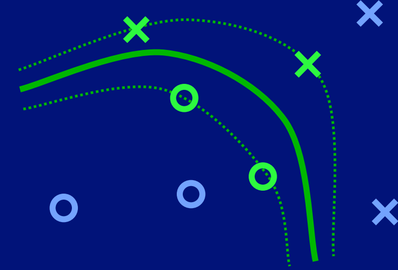
Resonant adiabatic energy recovery in computing: *Kerneltron* support vector machine for visual pattern recognition with resonant hot clock adiabatic energy recovery in charge-domain processing-in-memory computing at 1 fJ of energy per multiply-accumulate [Karakiewicz et al, 2017, 2012]. Energy recovery logic (ERL) CMOS adiabatic line drivers recover 98% of the CV^2 electrostatic energy in the charge-mode array.

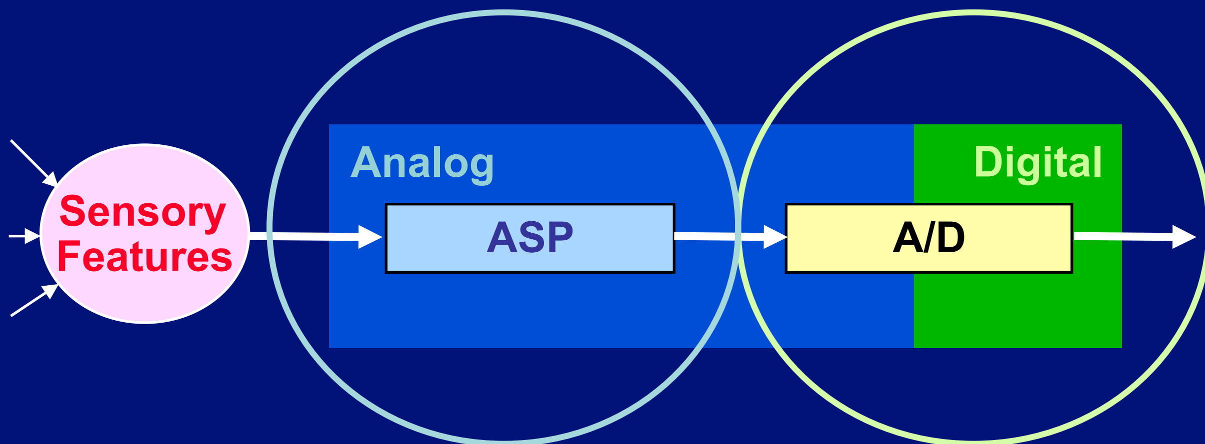
Resonant Adiabatic Energy Recovery



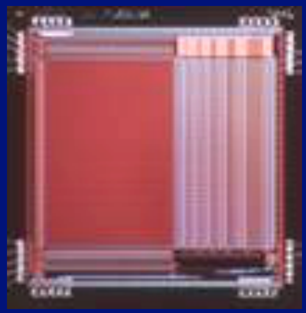
Resonant adiabatic energy recovery in communication: *Cyclic On-Off Keying* (*COOK*) modulation for wireless power and telemetry offers record bandwidth efficiency, allowing to transmit one bit of data every carrier cycle while simultaneously receiving RF power over the same high-Q inductive link [Ha et al, 2016].

SVM Pattern Recognition





Large-Margin Kernel Regression

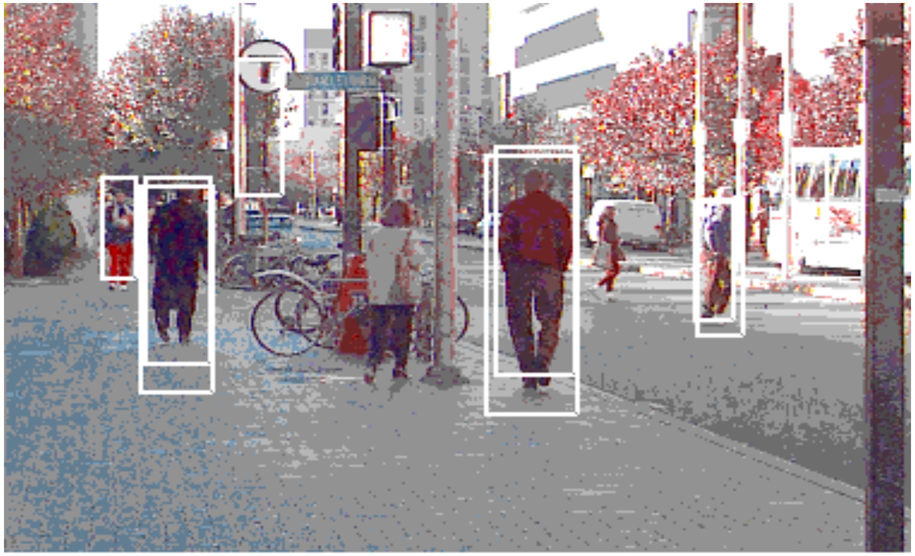


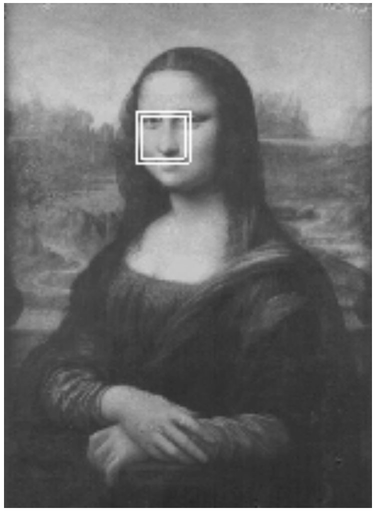
Class Identification

Kerneltron:
massively parallel
support vector
“machine” in silicon
(*ESSCIRC’ 2002*)

Trainable Modular Vision Systems: The SVM Approach

Papageorgiou, Oren, Osuna and Poggio, 1998

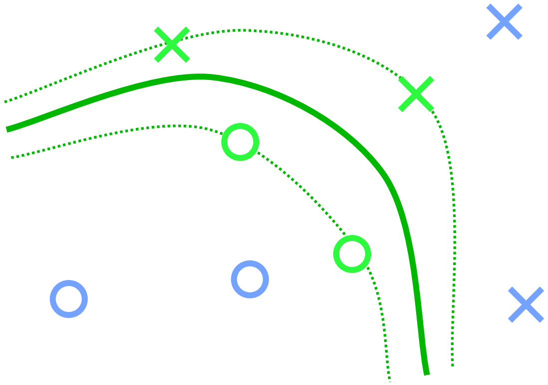




SVM classification for pedestrian and face object detection

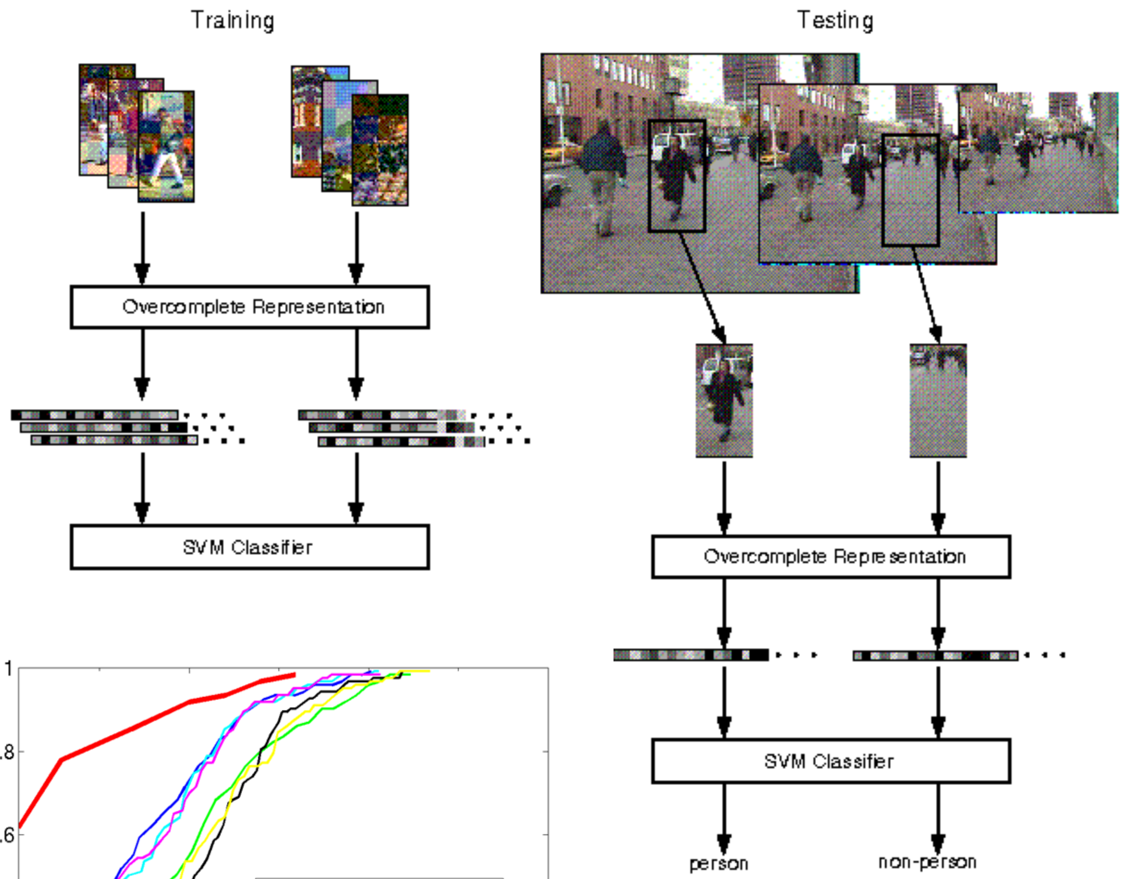
- Strong mathematical foundations in *Statistical Learning Theory* (Vapnik, 1995)

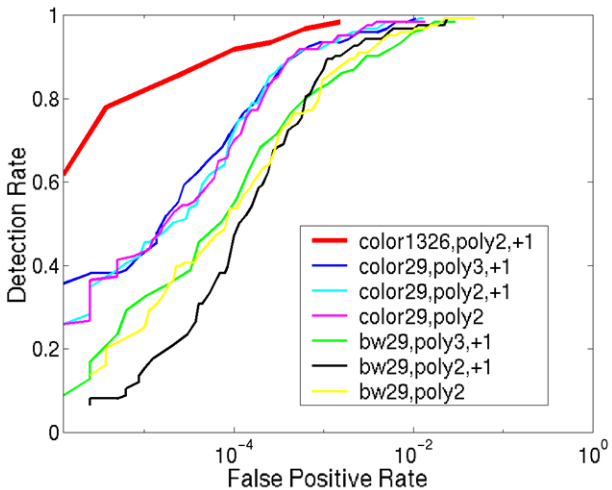
- The training process selects a small fraction of prototype *support vectors* from the data set, located at the *margin* on both sides of the classification boundary (e.g., barely faces vs. barely non-faces)



Trainable Modular Vision Systems: The SVM Approach

Papageorgiou, Oren, Osuna and Poggio, 1998



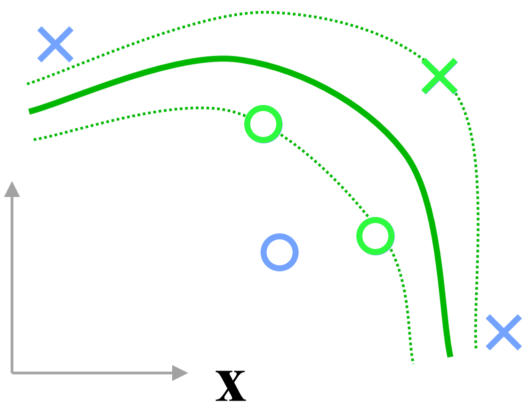


ROC curve for various image representations and dimensions

- The number of support vectors, in relation to the number of training samples and the vector dimension, determine the generalization performance
- Both training and run-time performance are severely limited by the computational complexity of evaluating kernel functions

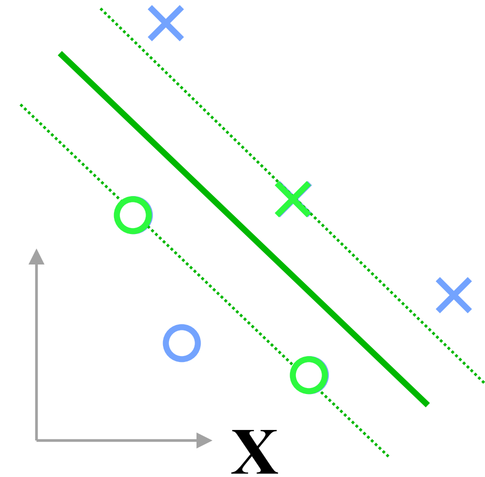
Kernels and Support Vector Machines

Mercer, 1909; Aizerman et al., 1964
Boser, Guyon and Vapnik, 1992



$$\Phi(\cdot)$$

$$\mathbf{X}_i = \Phi(\mathbf{x}_i)$$
$$\mathbf{X} = \Phi(\mathbf{x})$$
$$\mathbf{X}_i \cdot \mathbf{X} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})$$



$$y = \text{sign}\left(\sum_{i \in S} \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b\right)$$

$$y = \text{sign}\left(\sum_{i \in S} \alpha_i y_i \mathbf{X}_i \cdot \mathbf{X} + b\right)$$

$$K(\cdot,\cdot)$$

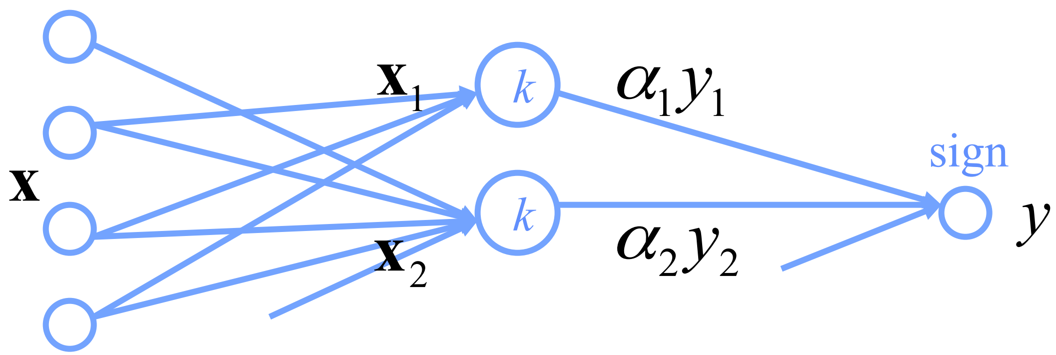
$$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) = K(\mathbf{x}_i, \mathbf{x})$$

Mercer's Condition

$$y = \text{sign}\left(\sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

Kernel Machines

$$y = \text{sign}(\sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$$



– Gaussian (Radial Basis Function Networks)

$$K(\mathbf{x}_i, \mathbf{x}) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}) \propto \exp(\frac{\mathbf{x}_i \cdot \mathbf{x}}{\sigma^2})$$

– Sigmoid (Two-Layer Perceptron)

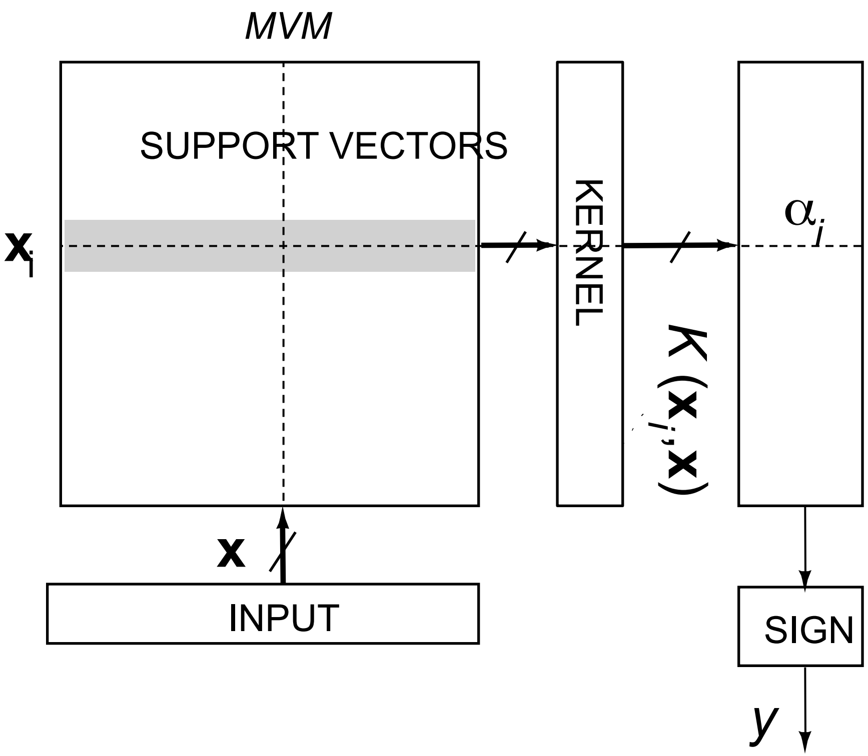
$$K(\mathbf{x}_i, \mathbf{x}) = \tanh(L + \mathbf{x}_i \cdot \mathbf{x})$$

only for certain L

– Polynomial (Splines etc.)

$$K(\mathbf{x}_i, \mathbf{x}) = (1 + \mathbf{x}_i \cdot \mathbf{x})^\nu$$

Parallel SVM Architecture

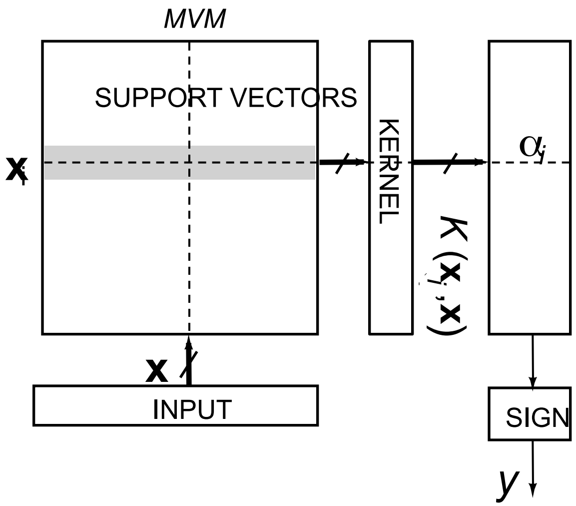


$$y = \mathrm{sign}\left(\sum_{i\in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

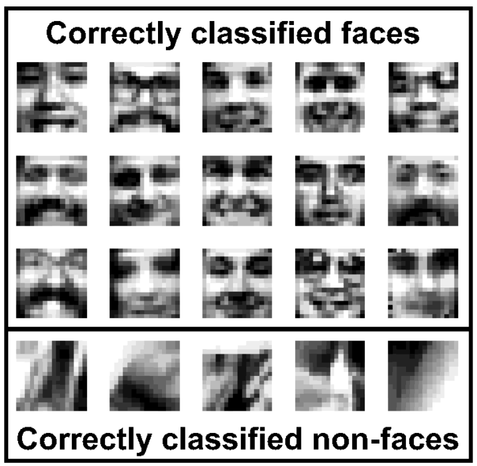
- Kernel inner-products are implemented by parallel matrix-vector multiplication (MVM).
- Silicon area and power dissipation are proportional to number of support vectors, favoring sparse SVM solutions.
- Sparsity in SVM training also guarantees proper generalization performance (Vapnik, 1995).

Kerneltron III: Adiabatic Support Vector "Machine"

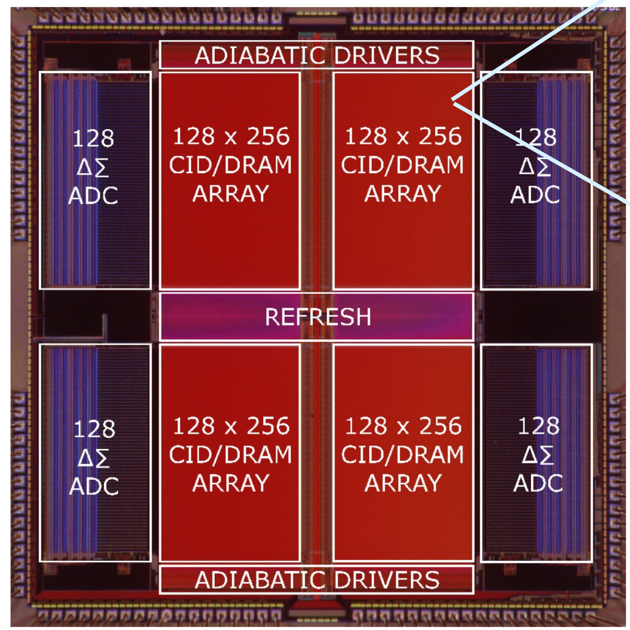
Karakiewicz, Genov, and Cauwenberghs, VLSI' 2006; CICC' 2007

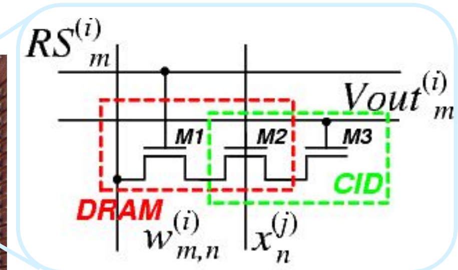


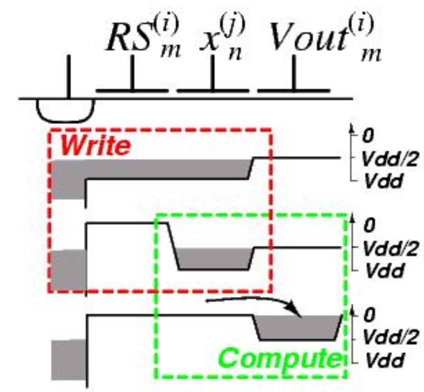
$$y = \text{sign}\left(\sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$



Classification results on MIT CBCL face detection data

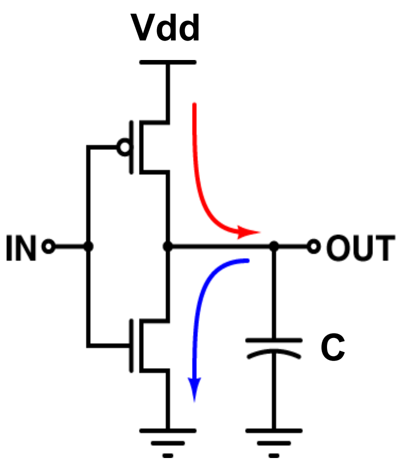




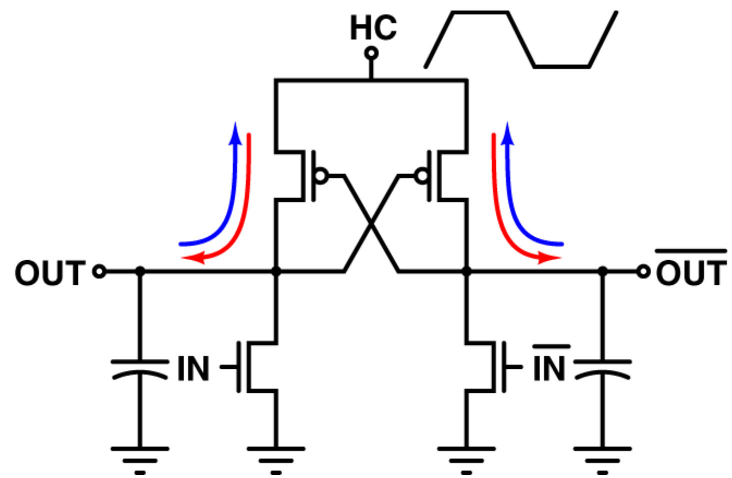


- ***1.2 TMACS / mW***
  - *adiabatic resonant clocking conserves charge energy*
  - *energy efficiency on par with human brain (10^{15} SynOP/S at 15W)*

CMOS Logic vs. Adiabatic Computing



CMOS logic



Energy recovery logic (ERL)
(Y. Moon, JSSC'96)

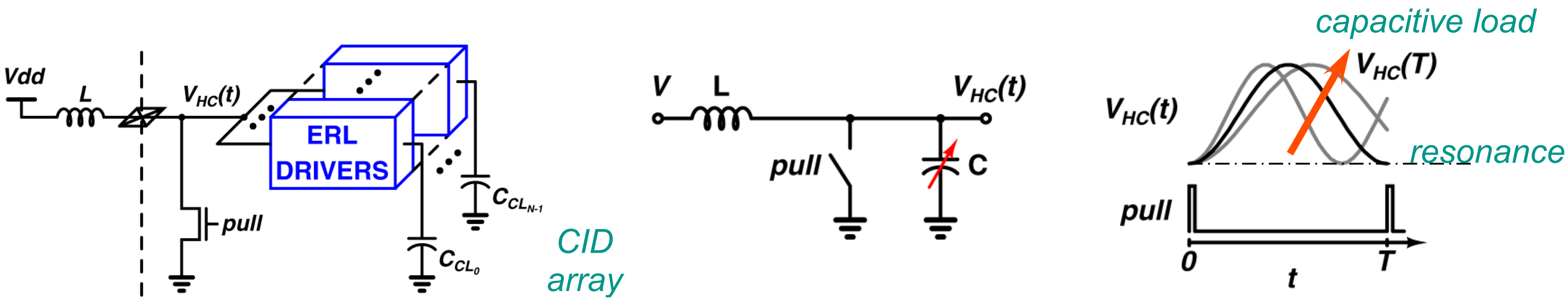
- *Dynamic energy dissipation*

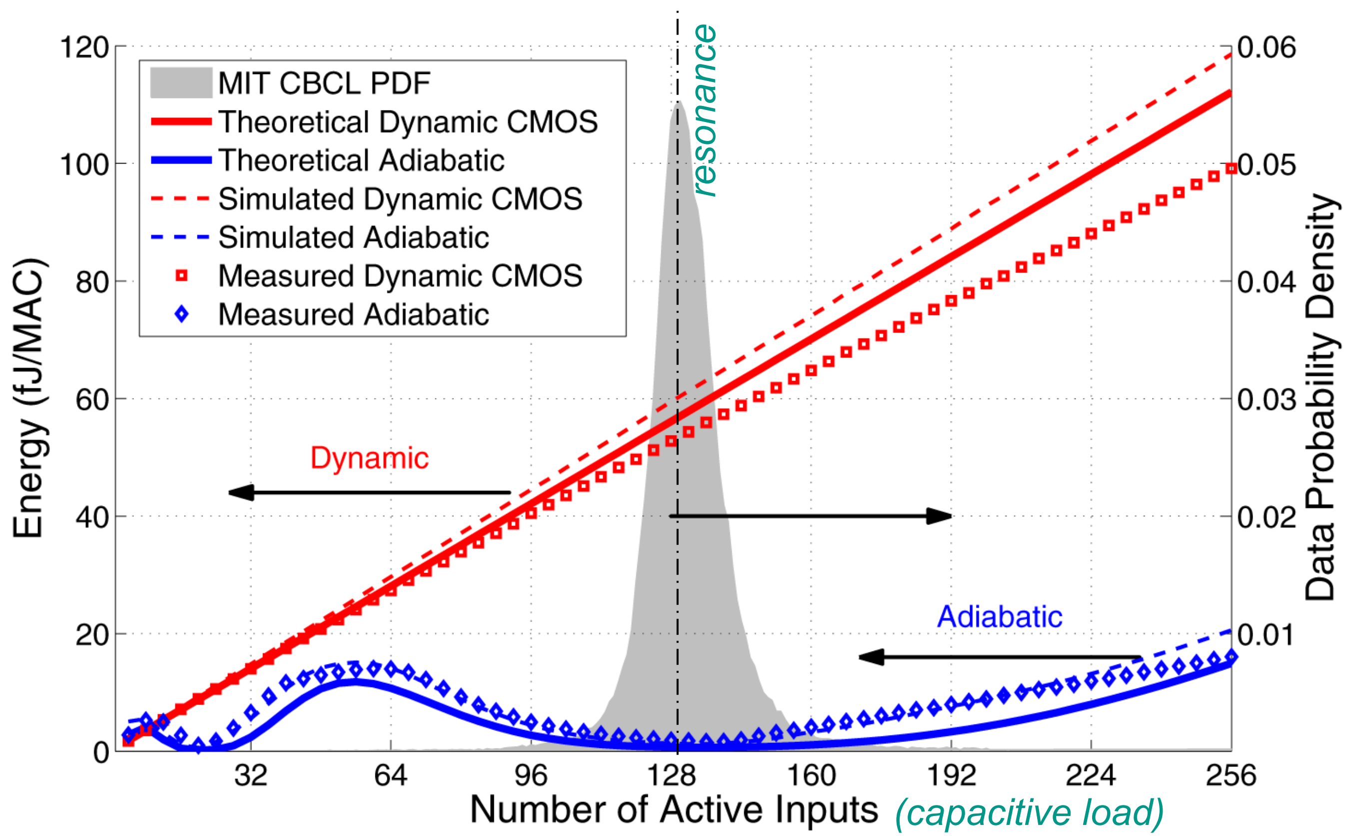
$$E_{diss.} = CVdd^2$$

- *'Hot clock' recycles energy*
  - *LC tank resonant clock*
- *Reversible computation*

Resonant Charge Energy Recovery

Karakiewicz, Genov, and Cauwenberghs, IEEE JSSC, 2007

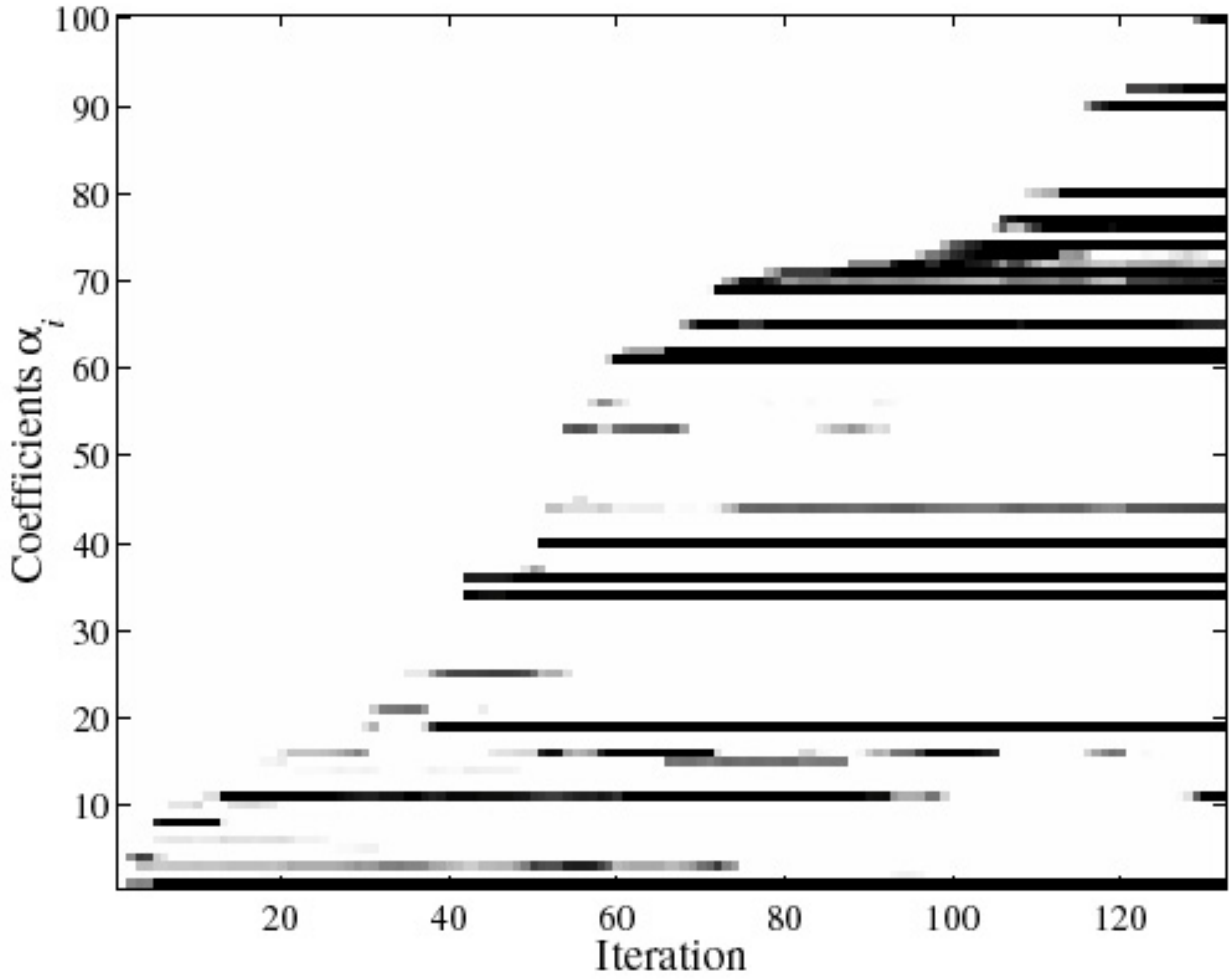




Incremental and Decremental SVM Learning

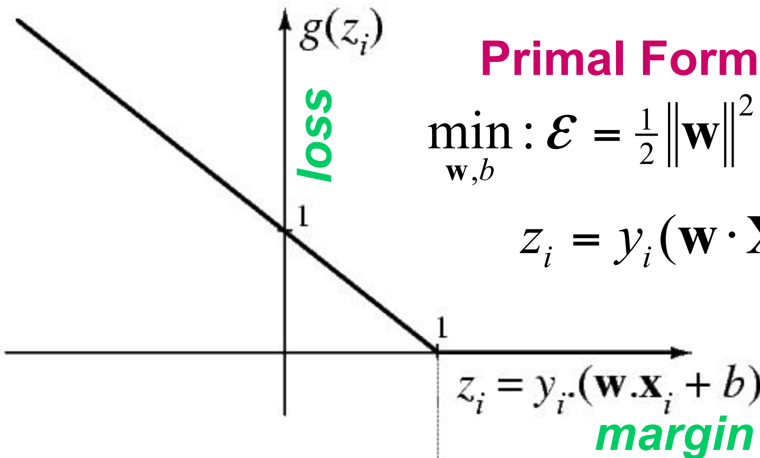
Cauwenberghs and Poggio, 2001

- Support Vector Machine training requires solving a linearly constrained quadratic programming problem in a number of coefficients equal to the number of data points.
- An incremental version, training one data point at at time, is obtained by solving the QP problem in recursive fashion, without the need for QP steps or inverting a matrix.
  - *On-line learning is thus feasible, with no more than L^2 state variables, where L is the number of margin (support) vectors.*
  - *Training time scales approximately linearly with data size for large, low-dimensional data sets.*
- Decremental learning (adiabatic reversal of incremental learning) allows to directly evaluate the exact leave-one-out generalization performance on the training data.
- When the incremental inverse jacobian is (near) ill-conditioned, a direct L1-norm minimization of the α coefficients yields an optimally sparse solution.



Trajectory of coefficients a as a function of time during incremental learning, for 100 data points in the non-separable case, and using a Gaussian kernel.

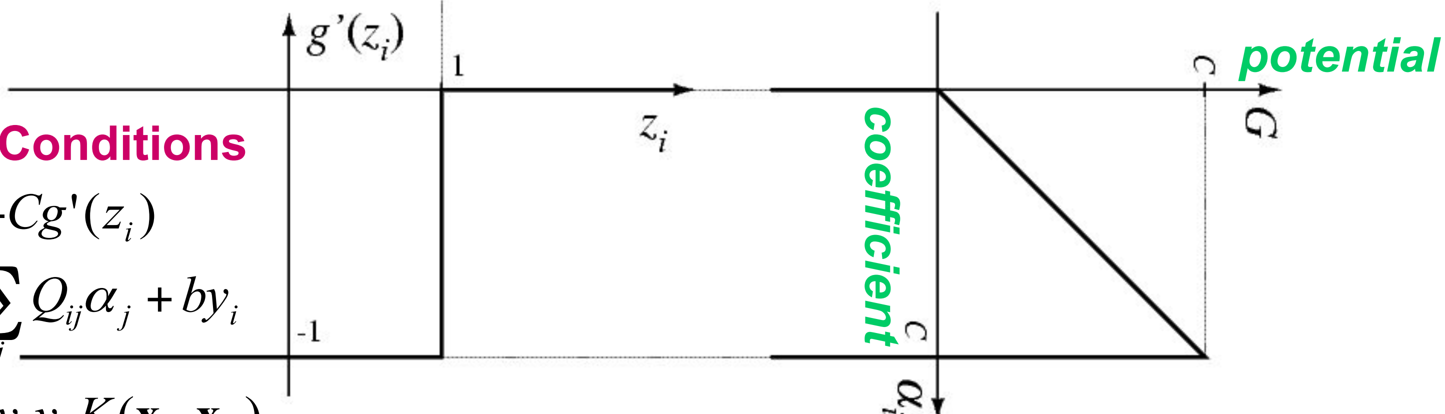
SVM Learning Revisited



Primal Formulation

$$\min_{\mathbf{w},b} : \mathcal{E} = \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_i g(z_i)$$

$$z_i = y_i(\mathbf{w}\cdot\mathbf{X}_i + b)$$



KKT Conditions

$$\alpha_i = -Cg'(z_i)$$

$$z_i = \sum_j Q_{ij}\alpha_j + by_i$$

$$Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

Dual Formulation

$$\min_{\mathbf{w},b} : \mathcal{E}_2^{SVcM} = \tfrac{1}{2}\sum_i\sum_j \alpha_i Q_{ij}\alpha_j - \sum_i \alpha_i$$

$$\text{subject to} : \sum_i y_i\alpha_i \equiv 0 \;\text{ and }\; 0 \le \alpha_i \le C, \forall i$$

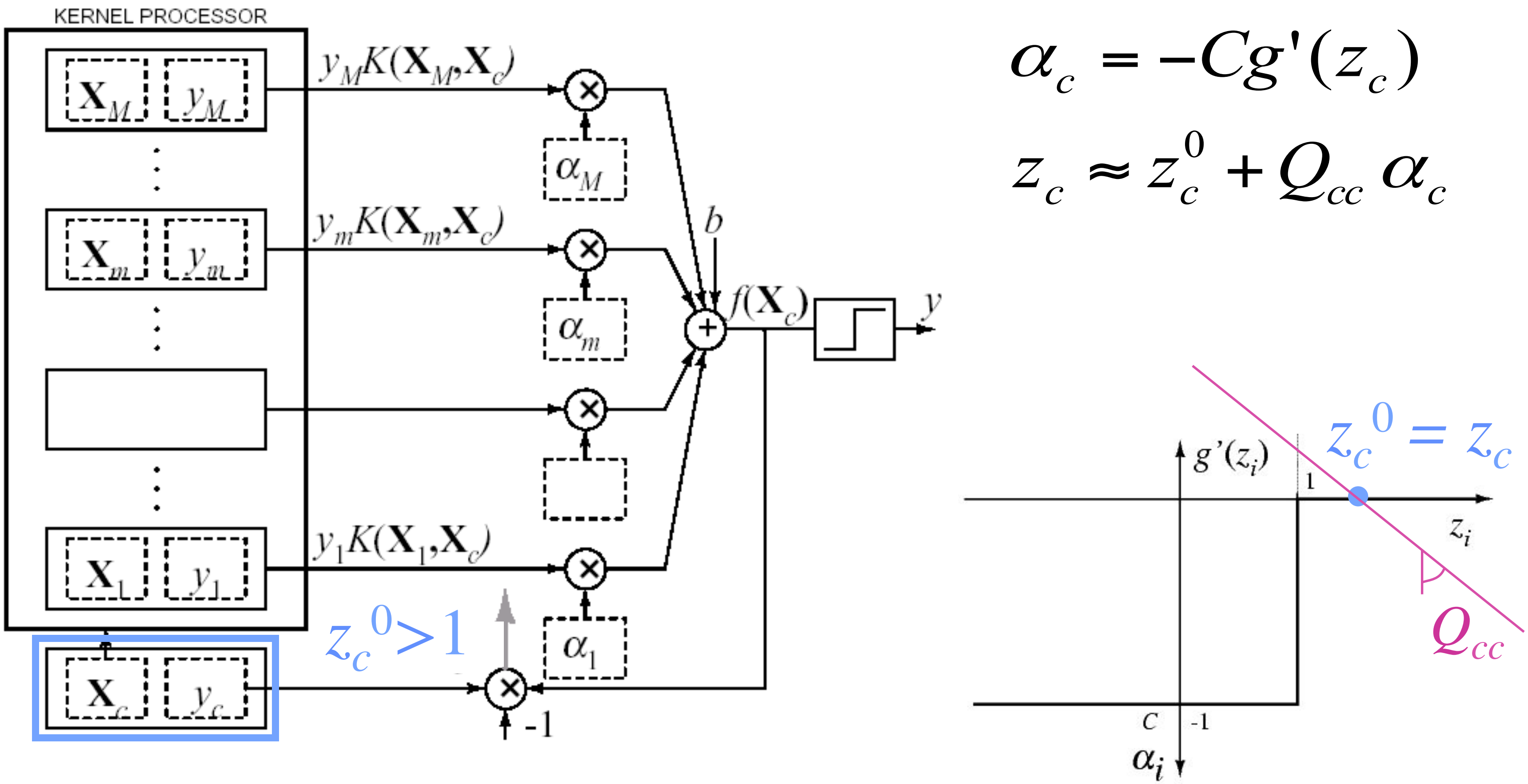
Soft-Margin SVM Classification
(Cortes and Vapnik, 1995)

Sequential On-Line SVM Learning

Chakrabartty, Genov and Cauwenberghs, 2003

$$\alpha_c = -Cg'(z_c)$$

$$z_c \approx z_c^0 + Q_{cc}\,\alpha_c$$

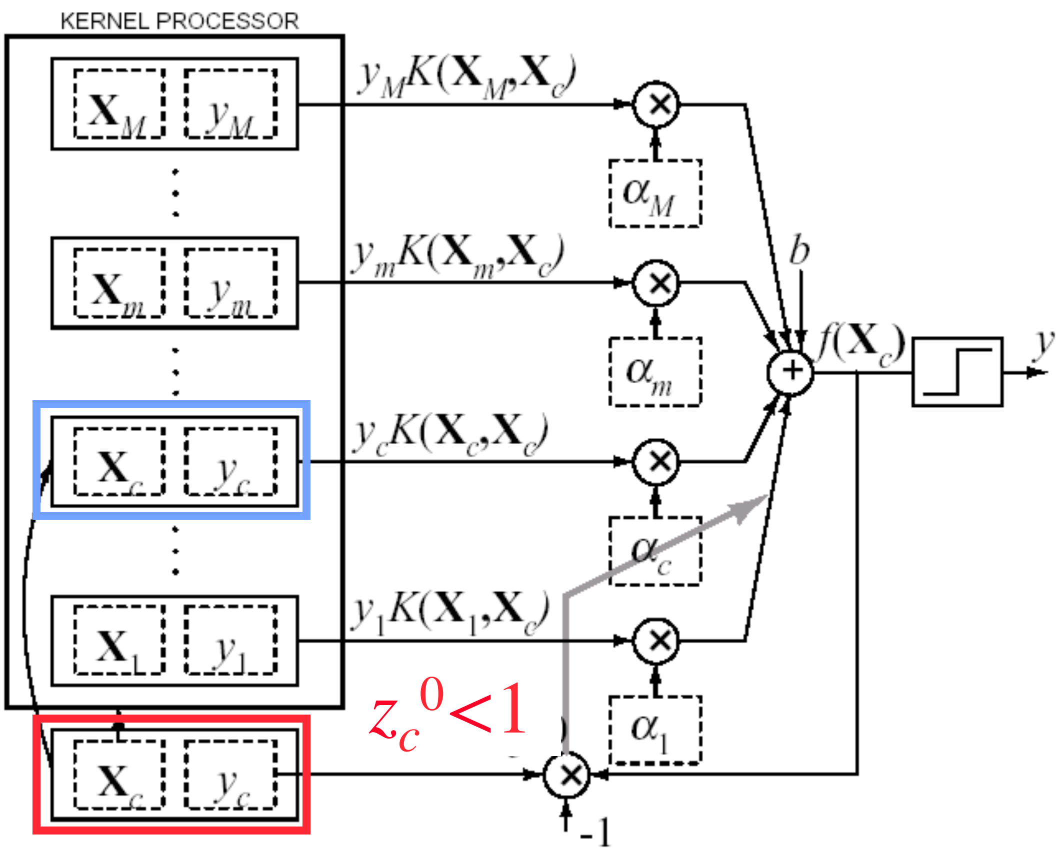


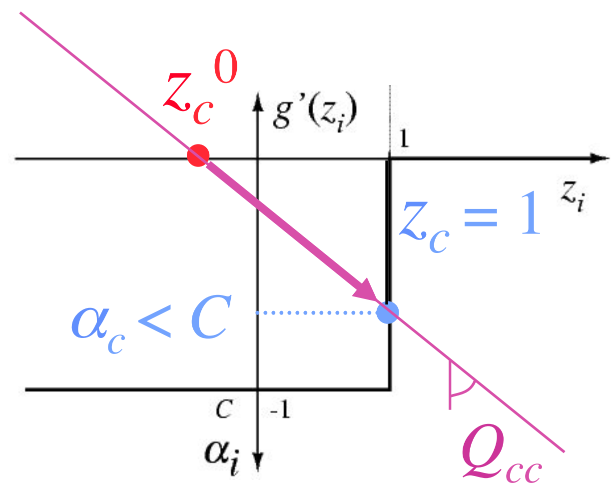
Sequential On-Line SVM Learning

Chakrabartty, Genov and Cauwenberghs, 2003

$$\alpha_c = -Cg'(z_c)$$

$$z_c \approx z_c^0 + Q_{cc}\,\alpha_c$$

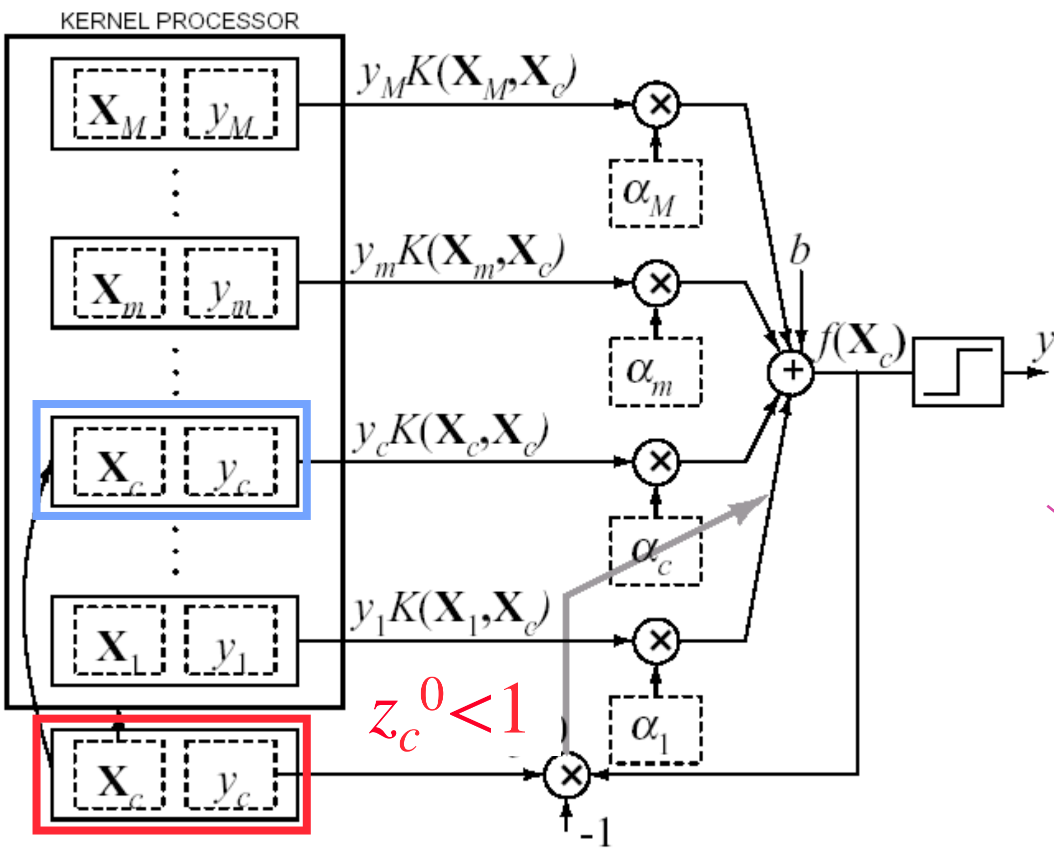




margin support vector

Sequential On-Line SVM Learning

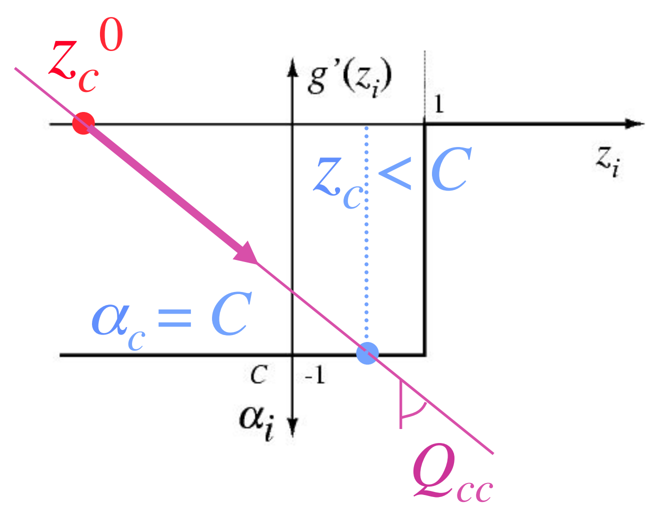
Chakrabartty, Genov and Cauwenberghs, 2003



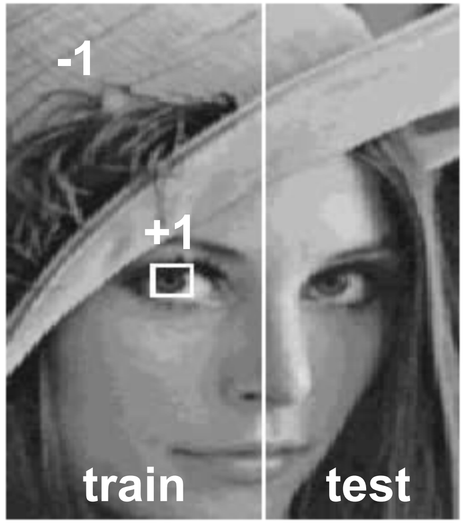
error support vector

$$\alpha_c = -Cg'(z_c)$$

$$z_c \approx z_c^0 + Q_{cc}\,\alpha_c$$

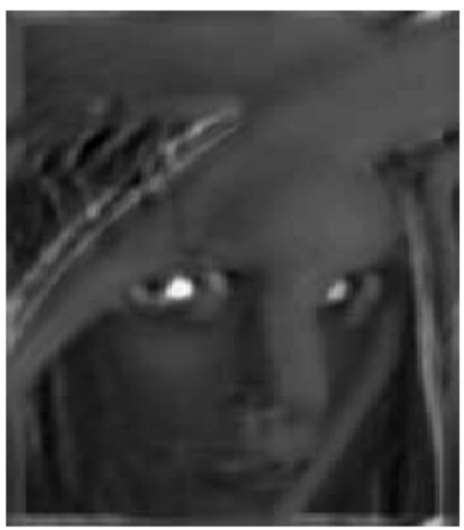


Effects of Sequential On-Line Learning and Finite Resolution



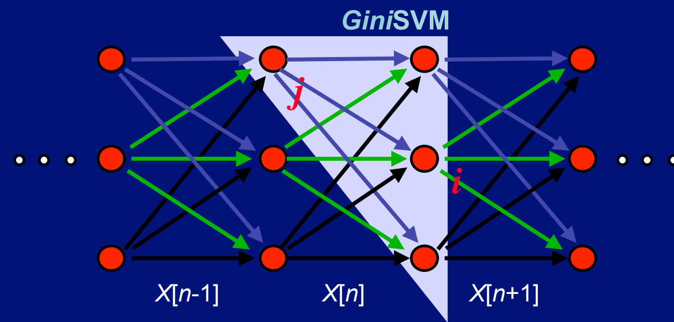
- *Matched Filter Response*

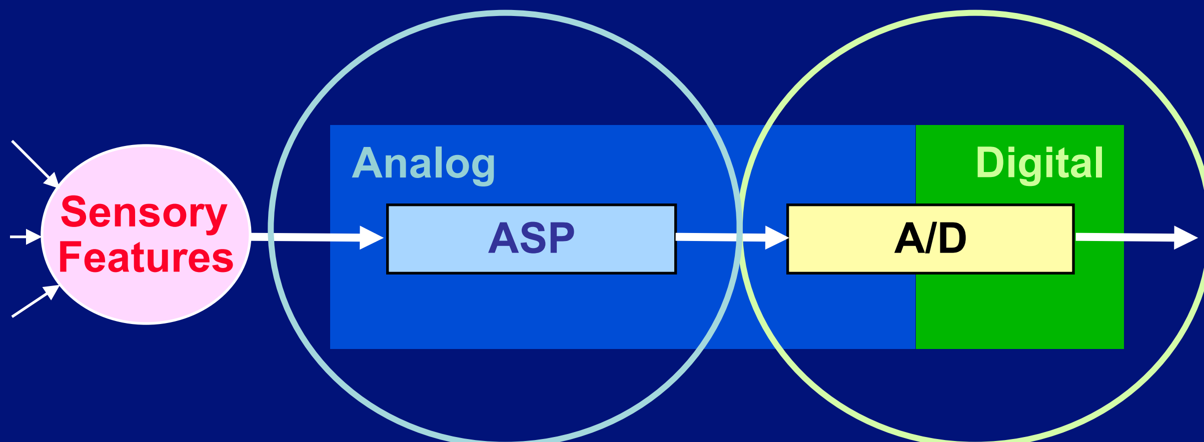
- *Batch Training*
- *Floating-Point Resolution*



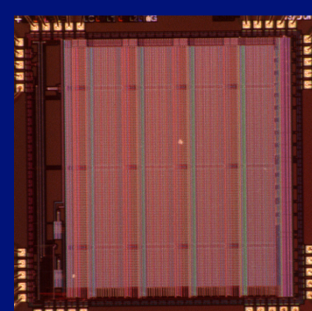
- *On-Line Sequential Training*
- *Kerneltron II*

SVM Sequence Estimation





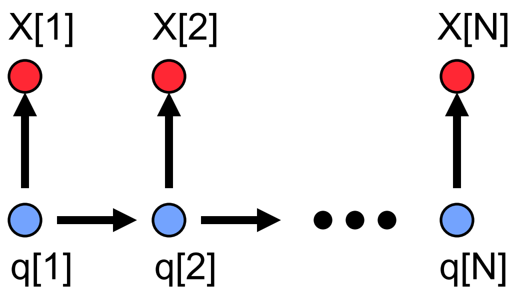
MAP Forward Decoding



Sequence Identification

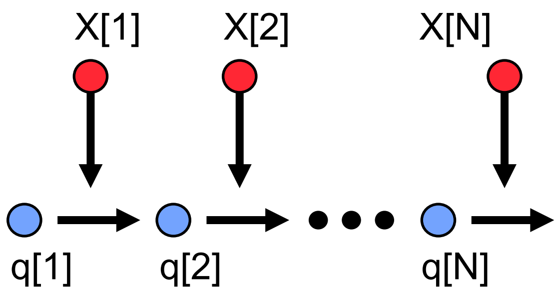
Sub-microwatt speaker verification and phoneme recognition (*NIPS'2004*)

MLE vs. MAP Sequence Estimation



Generative (MLE)
HMM

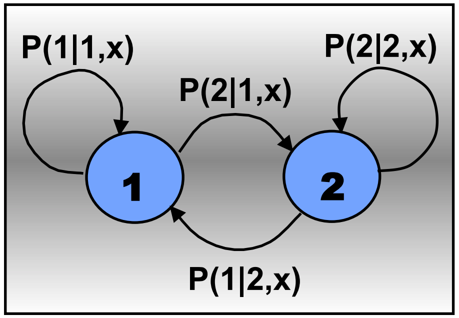
Density models (such as mixtures of Gaussians) require vast amounts of training data to reliably estimate parameters.



Discriminative (MAP)
FDKM

Transition-based speech recognition
(H. Bourlard and N. Morgan, 1994)

MAP forward decoding



Transition probabilities generated by large margin probability regressor

Forward Decoding Kernel Machines (FDKM)

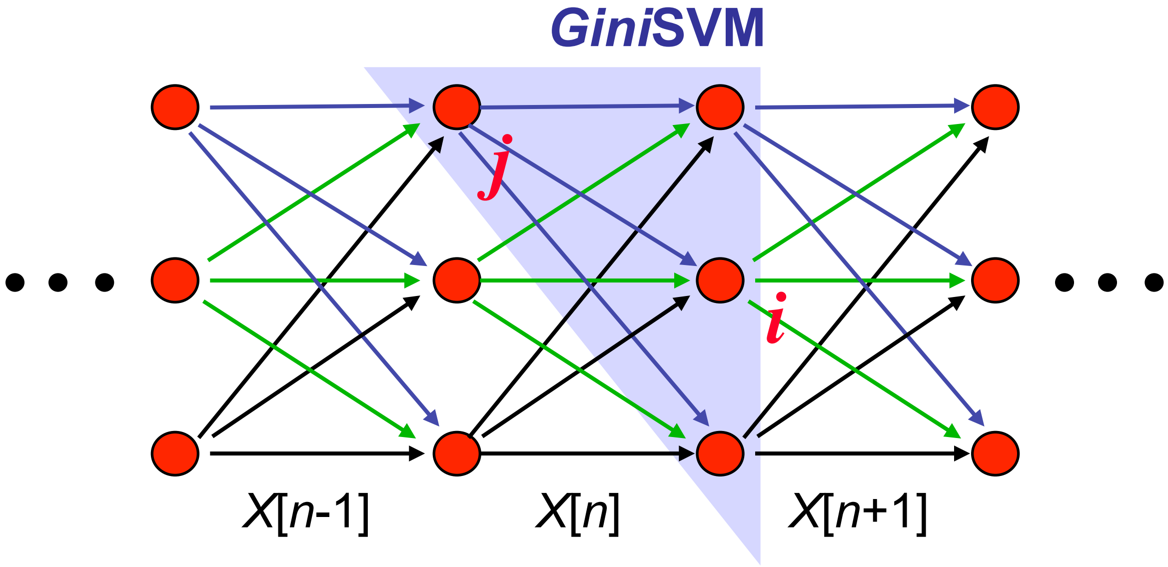
Chakrabartty and Cauwenberghs (NIPS'2002)

- Forward decoding of posterior probabilities α_i

$$\alpha_i[n] = \sum_j P_{ij}[n]\alpha_j[n-1]$$

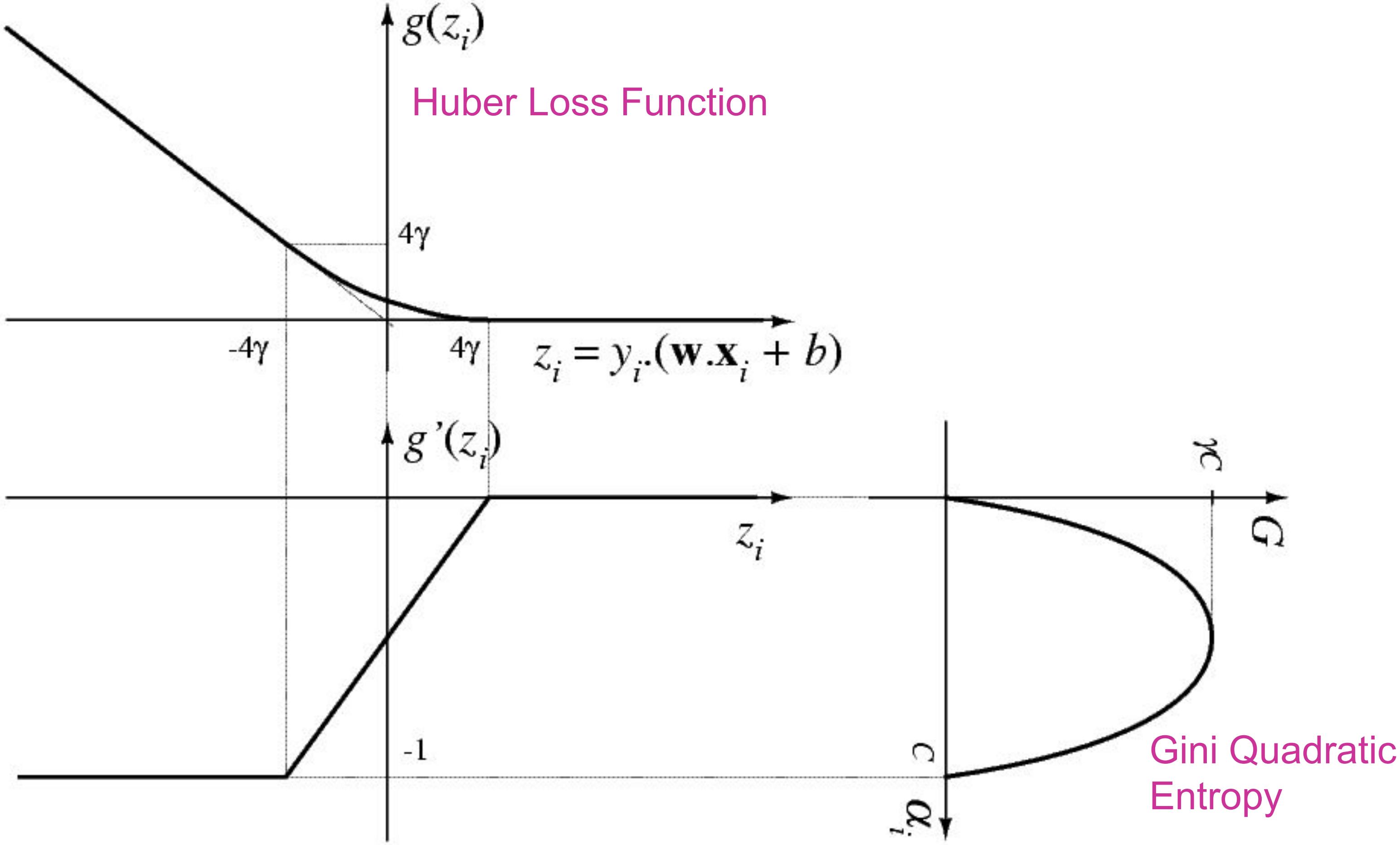
- Transition probabilities P_{ij} generated by SVM conditioned on input data X

$$P_{ij}[n] = P(i \mid j, X[n]) \propto f_{ij}(X[n])$$



*Gini*SVM Sparse Probability Regression

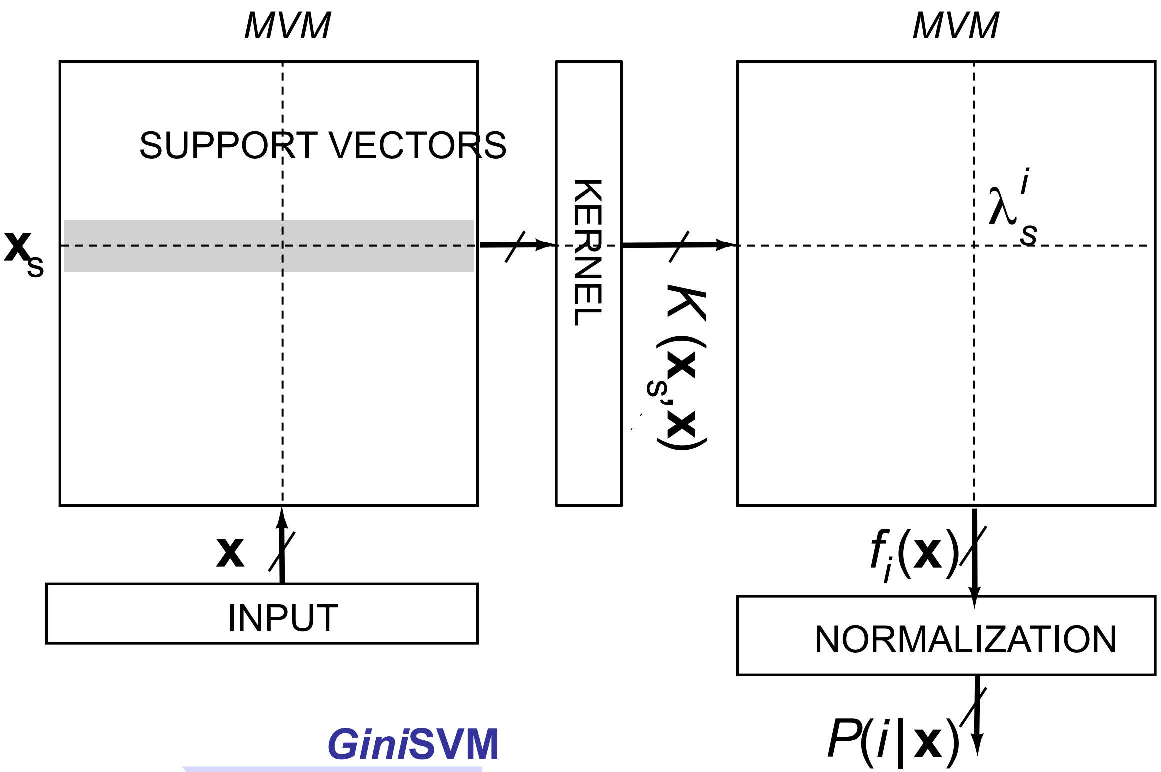
Chakrabartty and Cauwenberghs, JMLR 2007



$$\min_{\mathbf{w},b} : \varepsilon_2^{kGini} = \tfrac{1}{2}\sum_i \sum_j \alpha_i Q_{ij} \alpha_j - C \sum_i H_{Gini}(\tfrac{\alpha_i}{C})$$

$$\text{subject to} : \sum_i y_i \alpha_i \equiv 0 \;\text{ and } 0 \le \alpha_i \le \mathrm{C}, \text{ with } \; H_{Gini}(a) = 4\gamma(1-a)a$$

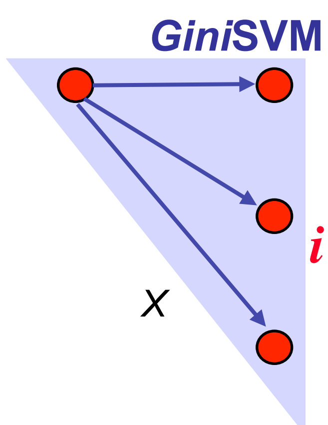
GiniSVM Probability Regression



Gini quadratic entropy in SVM training leads to sparse, large-margin regression of class probabilities (*Chakrabartty et al. 2002*)

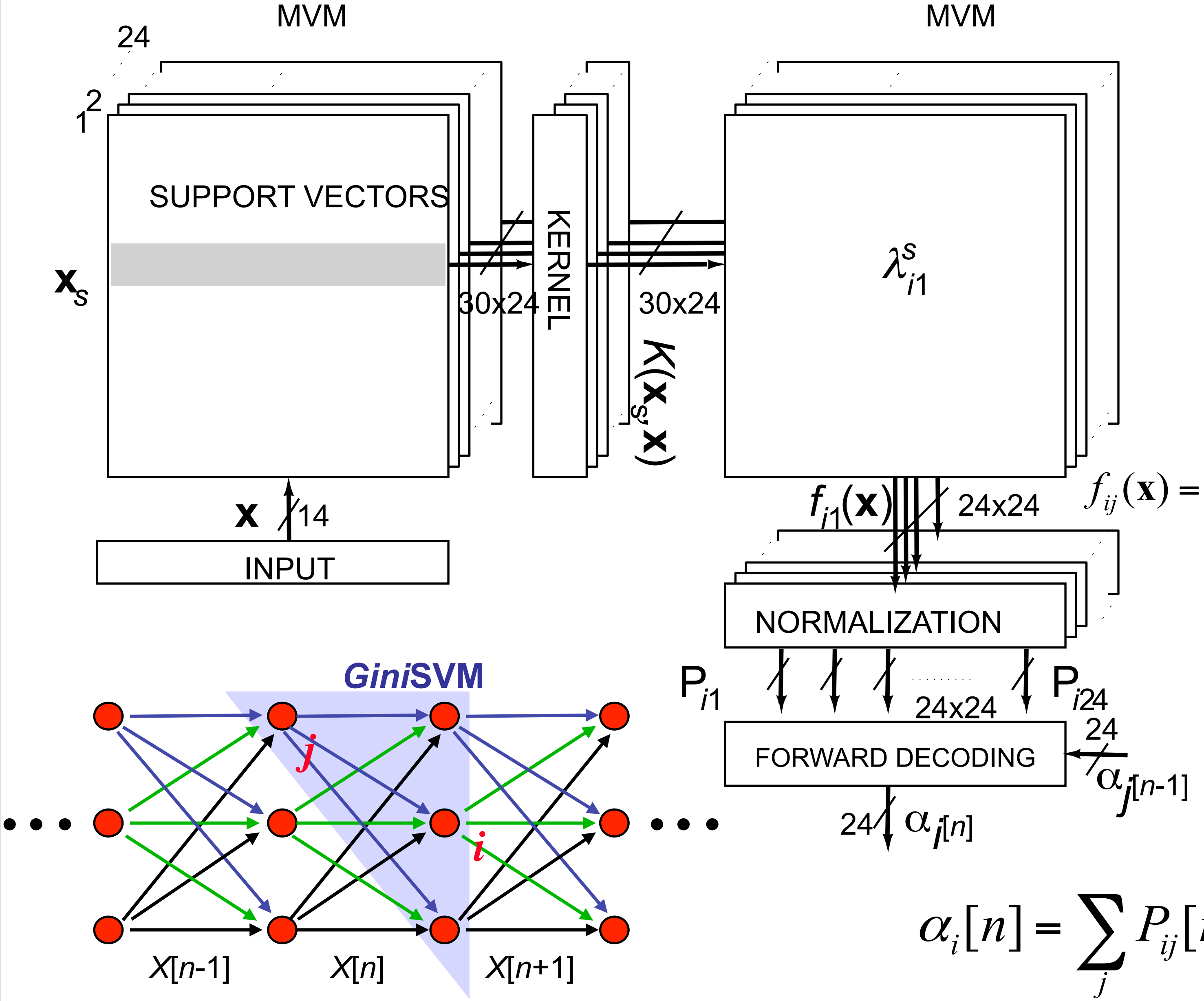
$$\sum_i P(i\,|\,\mathbf{x}) = 1$$

$$0 \le P(i\,|\,\mathbf{x}) \le 1$$



$$P(i\,|\,\mathbf{x}) \propto f_i(\mathbf{x}) = \sum_{s\in S} \lambda_s^i y_i K(\mathbf{x}_s,\mathbf{x}) + b$$

FDKM Architecture

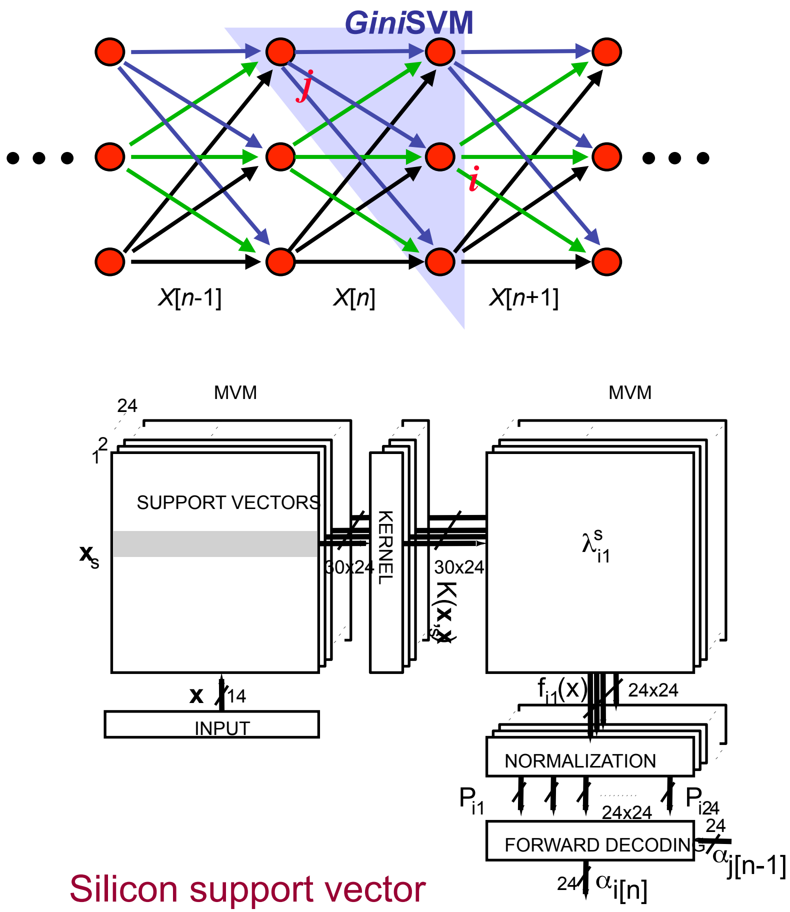


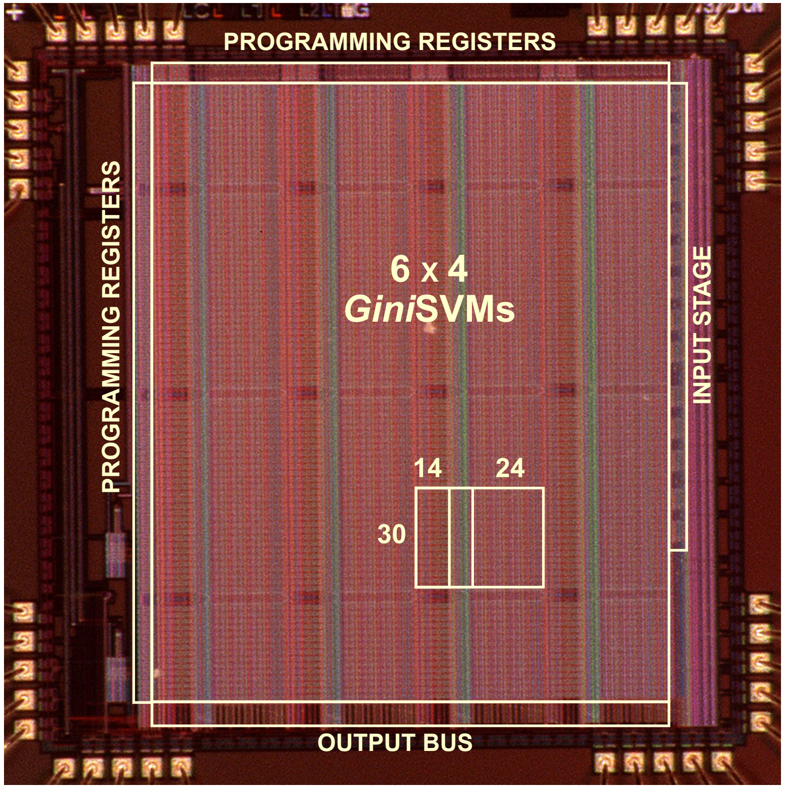
$$f_{ij}(\mathbf{x}) = \sum_{s \in S} \lambda^s_{ij} K(\mathbf{x}_s, \mathbf{x}) + b$$

$$\alpha_i[n] = \sum_j P_{ij}[n]\alpha_j[n-1]$$

*Gini*SVM/FDKM Processor

Chakrabartty and Cauwenberghs (NIPS'2004)



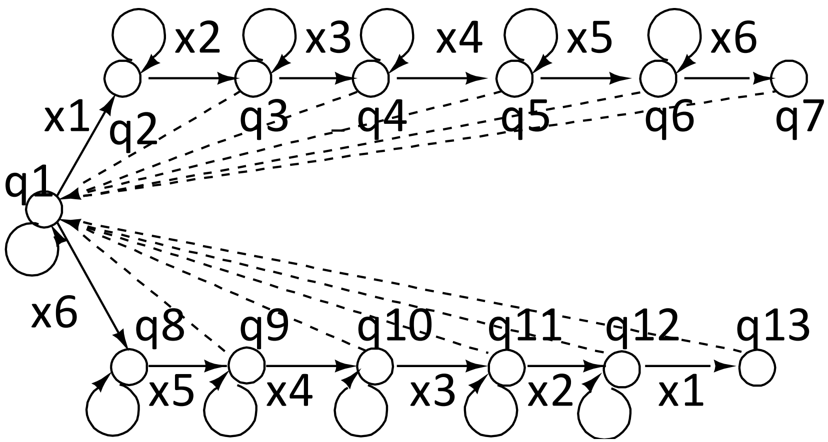


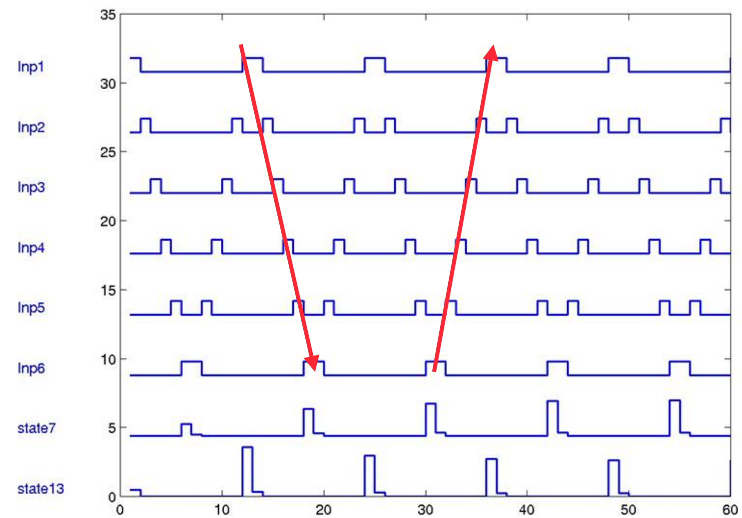
Silicon support vector machine (SVM) and forward decoding kernel machine (FDKM)

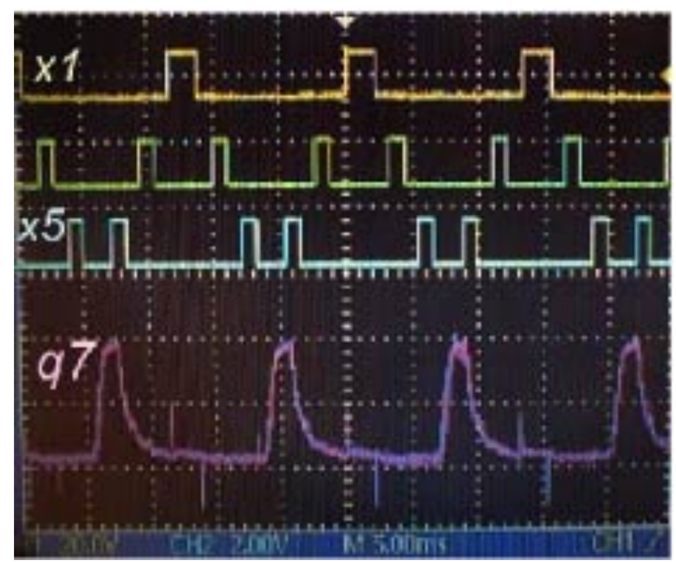
– ***Sub-Microwatt Power***
– *Subthreshold translinear MOS circuits*
– *Programmable with floating-gate non-volatile analog storage*

FDKM Dynamic Sequence Detection (*80 nW*)

Chakrabartty and Cauwenberghs (NIPS'2004)







FDKM Training Formulation

Chakrabartty and Cauwenberghs, 2002

- Large-margin training of state transition probabilities, using regularized cross-entropy on the posterior state probabilities:

$$H = C \sum_{n=0}^{N-1} \sum_{i=0}^{S-1} y_i[n] \log \alpha_i[n] - \tfrac{1}{2} \sum_{j=0}^{S-1} \sum_{i=0}^{S-1} | w_{ij} |^2$$

- Forward Decoding Kernel Machines (FDKM) decompose an upper bound of the regularized cross-entropy (by expressing concavity of the logarithm in forward recursion on the previous state):

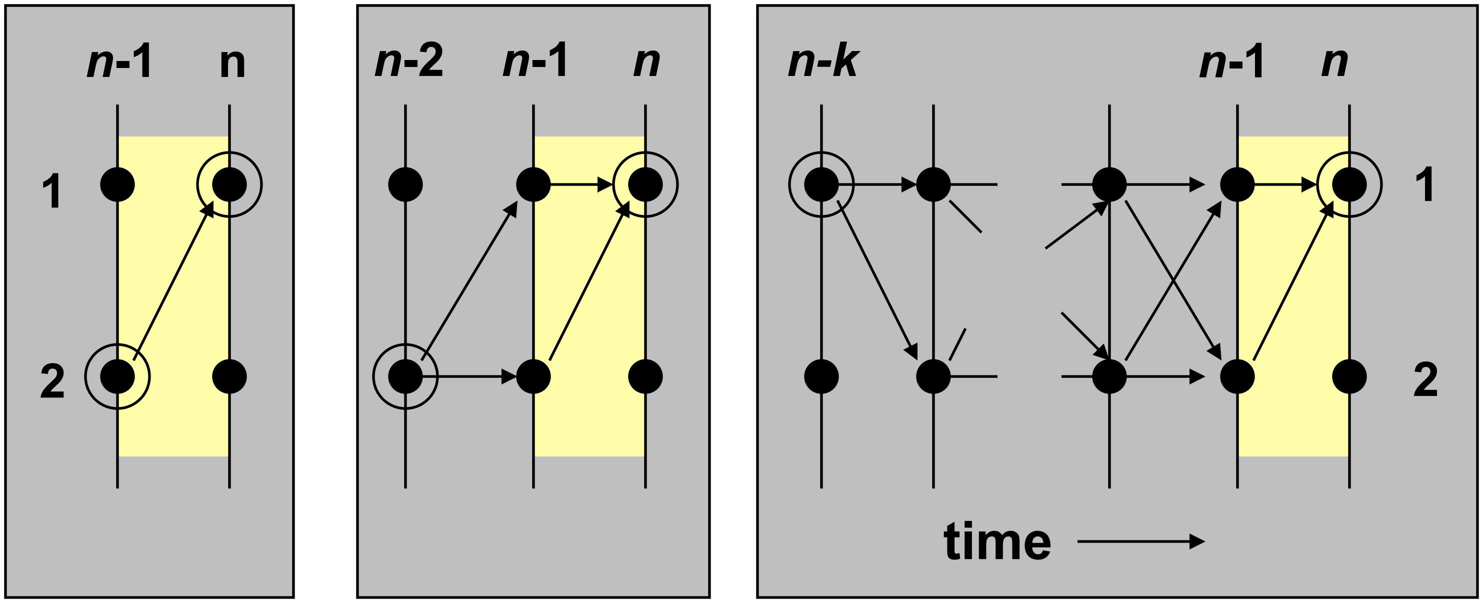
$$H \geq \sum_{j=0}^{S-1} H_j$$

which then reduces to S independent regressions of conditional probabilities, one for each outgoing state:

$$H_j = \sum_{n=0}^{N-1} C_j[n] \sum_{i=0}^{S-1} y_i[n] \log P_{ij}[n] - \tfrac{1}{2} \sum_{i=0}^{S-1} | w_{ij} |^2$$

$$C_j[n] = C \alpha_j[n-1]$$

Recursive MAP Training of FDKM

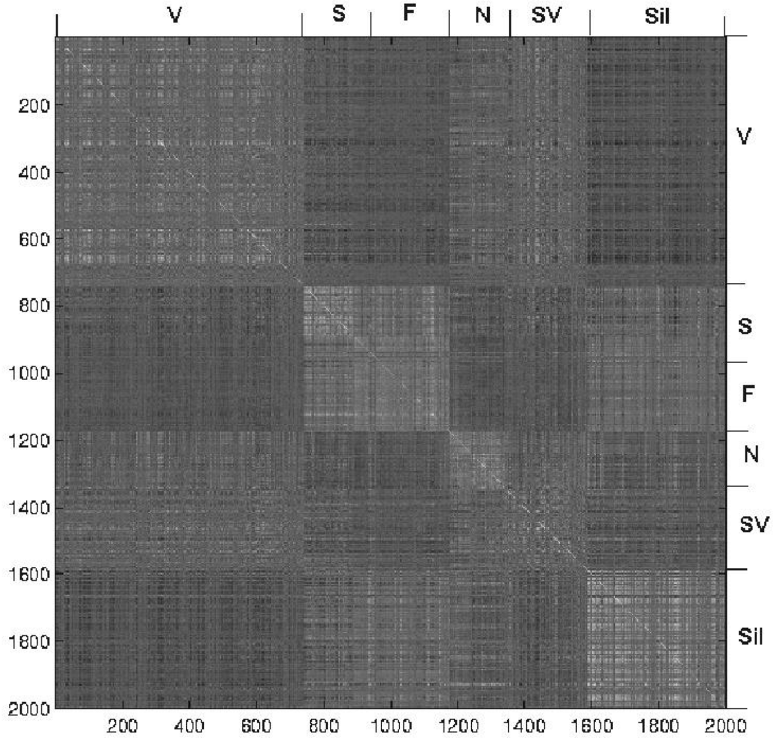


Epoch 1   Epoch 2   Epoch K

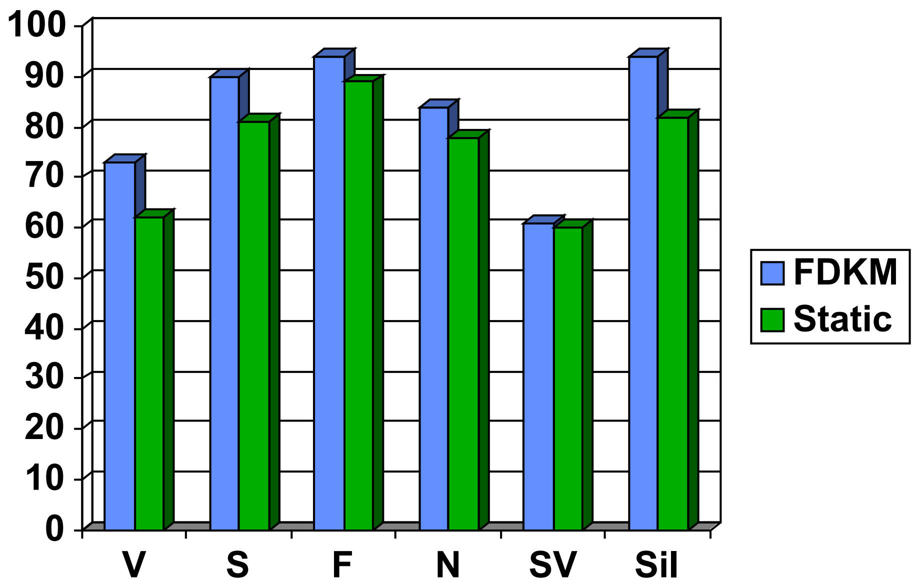
Phonetic Experiments (TIMIT)

Chakrabartty and Cauwenberghs, 2002

Features: cepstral coefficients for *Vowels*, *Stops, Fricatives*, *Semi-Vowels,* and *Silence*



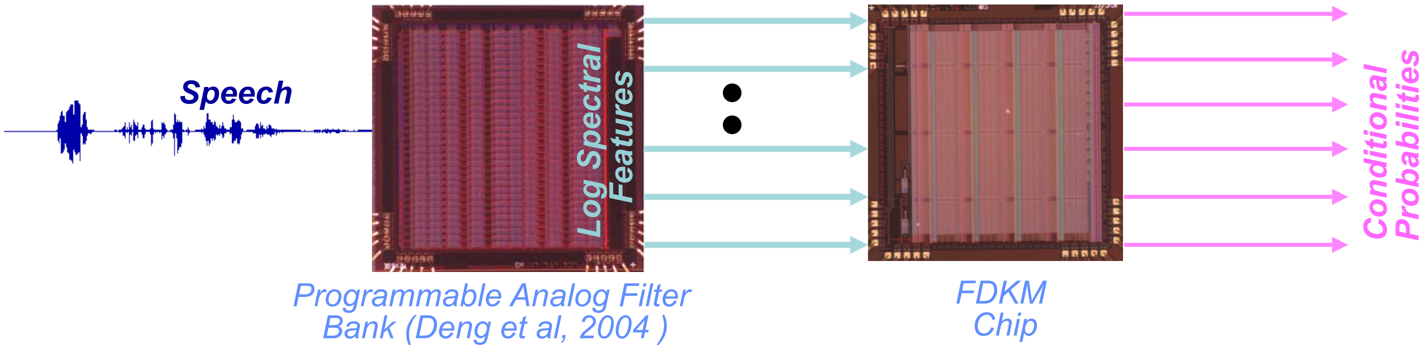
Kernel Map



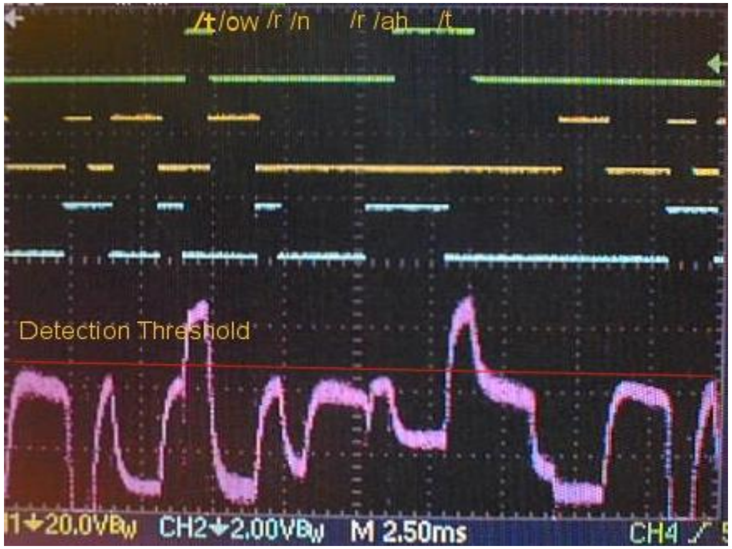
Recognition Rate

On-Chip TIMIT Phone Recognition

Chakrabartty and Cauwenberghs (NIPS'2004)



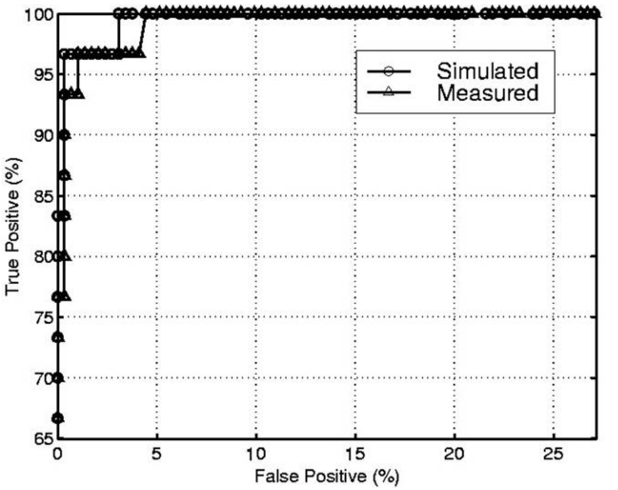
- 6 phones /t/n/r/ow/ah/eh/ from TIMIT corpus
- Thresholded Mel-cepstral features from log-compressed analog filterbank



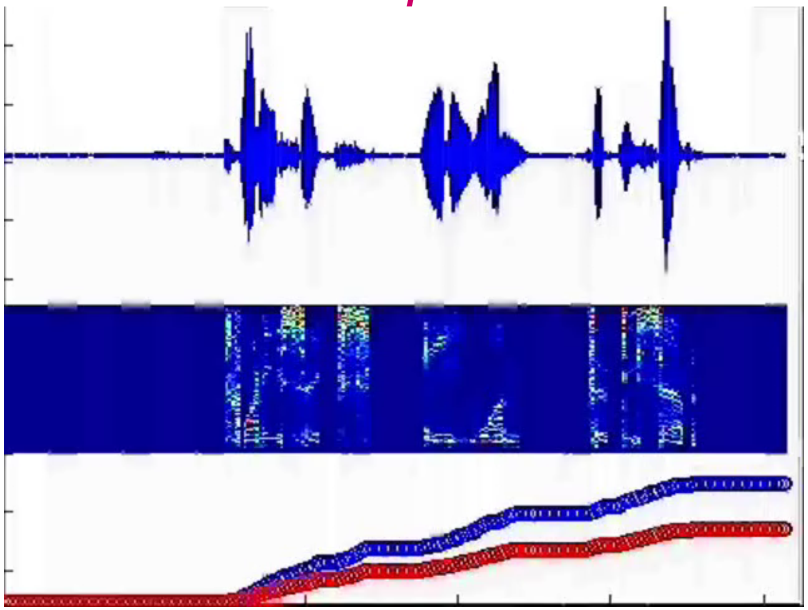
On-Chip Speaker Verification (*840nW*)

Chakrabartty and Cauwenberghs (NIPS' 2004)

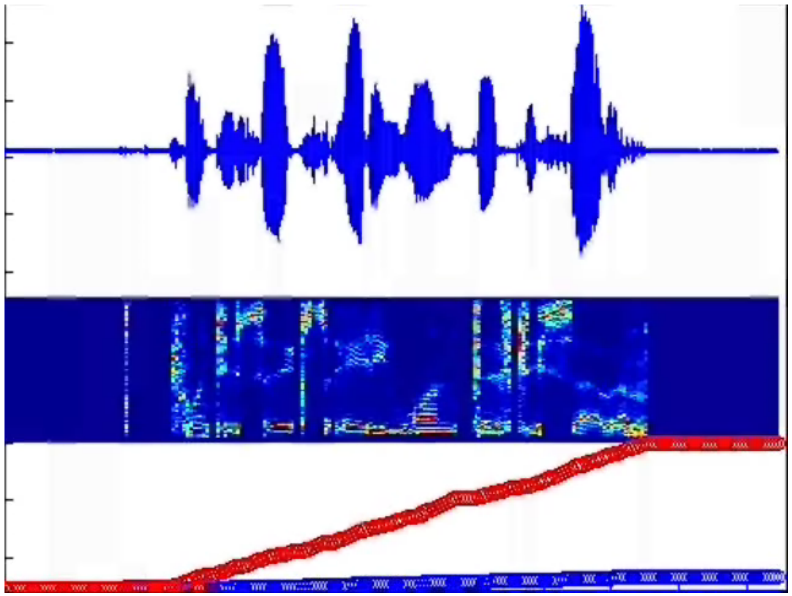
- 1 speaker and 10 imposters from YOHO dataset
- 92% recognition accuracy on 48 true and 432 imposter out-of-sample utterances
- 352 support vectors (47% FDKM chip capacity)
- 840 nW power at 25msec frame rate



Correct Speaker



Imposter



BEGN 207 Neuromorphic Integrated Bioelectronics

Date	Topic
9/27, 9/29	Biophysical foundations of natural intelligence in neural systems. Subthreshold MOS silicon models of membrane excitability. Silicon neurons. Hodgkin-Huxley and integrate-and-fire models of spiking neuronal dynamics. Action potentials as address events.
10/4, 10/6	Silicon retina. Low-noise, high-dynamic range photoreceptors. Focal-plane array signal processing. Spatial and temporal contrast sensitivity and adaptation. Dynamic vision sensors.
10/11, 10/13	Silicon cochlea. Low-noise acoustic sensing and automatic gain control. Continuous wavelet filter banks. Interaural time difference and level difference auditory localization. Blind source separation and independent component analysis.
10/18, 10/20	Silicon cortex. Neural and synaptic compute-in-memory arrays. Address-event decoders and arbiters, and integrate-and-fire array transceivers. Hierarchical address-event routing for locally dense, globally sparse long-range connectivity across vast spatial scales.
10/28, 11/1	Review. Modular and scalable design for neuromorphic and bioelectronic integrated circuits and systems. Design for full testability and controllability.
11/1, 11/3	Midterm due 11/2. Low-noise, low-power design. Fundamental limits of noise-energy efficiency, and metrics of performance. Biopotential and electrochemical recording and stimulation, lab-on-a-chip electrophysiology, and neural interface systems-on-chip.
11/8, 11/10	Learning and adaptation to compensate for external and internal variability over extended time scales. Background blind calibration of device mismatch. Correlated double sampling and chopping for offset drift and low-frequency noise cancellation.
11/15, 11/17	Energy conservation. Resonant inductive power delivery and data telemetry. Ultra-high efficiency neuromorphic computing. Resonant adiabatic energy-recovery charge-conserving synapse arrays.
11/22, 11/24	Guest lectures
11/29, 12/1	Project final presentations. All are welcome!