

Neural Networks Ancillaries

Mohamed A. El-Sharkawi

Mohamed A. El-Sharkawi

Computational Intelligence Applications (CIA) Lab.

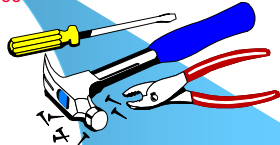
Department of EE, Box 352500

University of Washington

Seattle, WA 98195-2500

elsharkawi@ee.washington.edu

<http://cialab.ee.washington.edu>



Why Computational Intelligence?

- *Complex nonlinear mapping through a set of input/output examples.*
- *No structured model.*
- *Variables can be easily include or excluded.*
- *Superior noise rejection capability.*
- *Fast executions.*

Challenges: System

- *Explicit mathematical models*
- *Intensive computations*
- *Topology and operation changes*
- *Repeated solutions*
- *Noisy operating conditions*
- *Available knowledge in historical examples*



Challenges: NN

- ❑ *NN architecture*
- ❑ *Learning protocol*
- ❑ *Range of training data*
- ❑ *Data spanning in the operational space*
- ❑ *Data statistical properties.*
- ❑ *Correlated Features.*



Challenges: NN

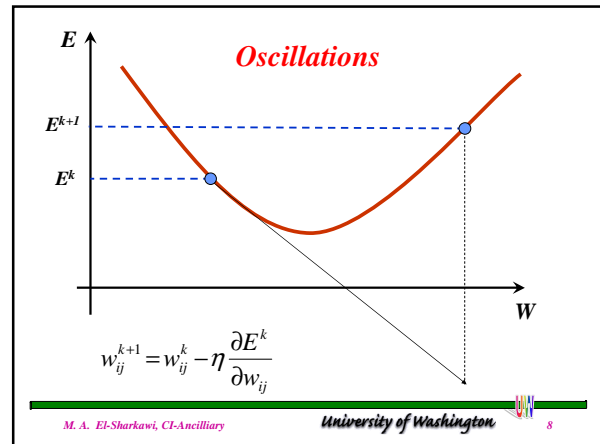
- ❑ *Sensitivity to variation in topology, operation, and control characteristics.*
- ❑ *Memorization.*
- ❑ *Saturation.*
- ❑ *Adaptation to system dynamics.*

1. Speed Training

Potential Problems with BEP

$$w_{ij}^{k+1} = w_{ij}^k - \eta \frac{\partial E^k}{\partial w_{ij}}$$

- Slow convergence:
 - When η is small
 - Near the minimum since the gradient is very small
- Oscillations can occur when η is large
- BEP can be trapped in local minima



Vogl Method

$$\eta(k+1) = \begin{cases} \alpha \eta(k); & \text{if } E(k+1) < E(k) \\ \beta \eta(k); & \text{if } E(k+1) > E(k) \\ \eta(k) & \end{cases}$$

Where

$$\alpha > 1$$

$$\beta < 1$$

Delta-Bar-Delta

$$\eta(k+1) = \eta(k) + \Delta \eta(k)$$

$$\Delta \eta(k+1) = \begin{cases} \alpha; & \text{if } \frac{\partial E^k}{\partial w} \frac{\partial E^{k+1}}{\partial w} > 0 \text{ Same side of error function} \\ -\beta \eta(k); & \text{if } \frac{\partial E^k}{\partial w} \frac{\partial E^{k+1}}{\partial w} < 0 \text{ Switched sides of error function} \end{cases}$$

Otherwise $\Delta \eta(k+1) = 0$

Where $\alpha < 1$

$\beta < 1$

Resilient Propagation (Rprop)

$$\Delta w_{ij}(k+1) = \Delta_{ij}(k)$$

$$\Delta w_{ij}(k+1) = \begin{cases} -\Delta_{ij}; & \text{if } \frac{\partial E^k}{\partial w_{ij}} > 0 \\ \Delta_{ij}; & \text{if } \frac{\partial E^k}{\partial w_{ij}} < 0 \end{cases} \quad \Delta_{ij}(k+1) = \begin{cases} \alpha \Delta_{ij}(k); & \text{if } \frac{\partial E^k}{\partial w_{ij}} \frac{\partial E^{k+1}}{\partial w_{ij}} > 0 \\ \beta \Delta_{ij}(k); & \text{if } \frac{\partial E^k}{\partial w_{ij}} \frac{\partial E^{k+1}}{\partial w_{ij}} < 0 \\ \Delta_{ij}(k) & \end{cases}$$

Otherwise $\Delta w_{ij}(k+1) = 0$

$\alpha > 1$
 $\beta < 1$

Quickprop

- Second order method using the second derivative of the error function
- Taylor Expansion

$$E^* = E + E_w \Delta w; \quad \Delta w = w^* - w$$

Where E^* is the error at the minimum point

$$E_w^* = E_w + E_{ww} \Delta w = 0$$

$$\text{Hence, } \Delta w = -\frac{E_w}{E_{ww}}$$

Quickprop

$$\Delta w(k) = -\frac{E_w(k)}{E_{ww}(k)}$$

$$E_{ww}(k) = \frac{E_w(k-1) - E_w(k)}{w(k-1) - w(k)} = \frac{E_w(k-1) - E_w(k)}{-\Delta w(k-1)}$$

$$\Delta w(k) = -\frac{E_w(k)}{\frac{E_w(k-1) - E_w(k)}{-\Delta w(k-1)}} = \frac{E_w(k)}{E_w(k-1) - E_w(k)} \Delta w(k-1)$$

Momentum

- To stabilize the weight trajectory
- Weight is changed based on BEP plus the previous change

Momentum

BEP

$$v_{ij}^{k+1} = v_{ij}^k + \delta v_{ij}^k$$

$$w_{ij}^{k+1} = w_{ij}^k + \delta w_{ij}^k$$

BEP with momentum

$$v_{ij}^{k+1} = v_{ij}^k + \delta v_{ij}^k + \alpha \delta v_{ij}^{k-1} \quad \alpha < 1$$

$$w_{ij}^{k+1} = w_{ij}^k + \delta w_{ij}^k + \alpha \delta w_{ij}^{k-1}$$

Weight decay

- Weights are allowed to be decayed
- Decay allows the NN to explore other regions

Variations of BEP: Weight Decay

BEP

$$v_{ij}^{k+1} = v_{ij}^k + \delta v_{ij}^k$$

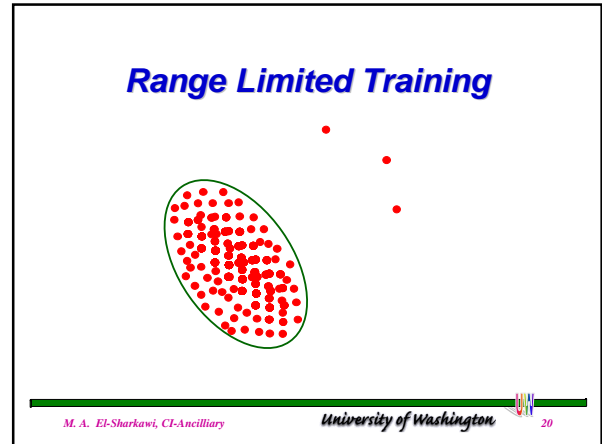
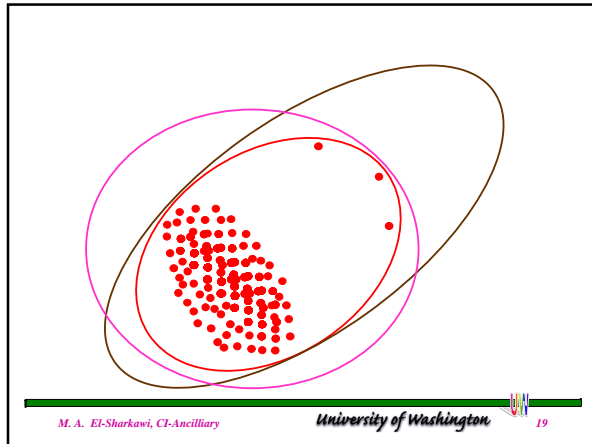
$$w_{ij}^{k+1} = w_{ij}^k + \delta w_{ij}^k$$

BEP with momentum

$$v_{ij}^{k+1} = v_{ij}^k + \delta v_{ij}^k - \alpha v_{ij}^{k-1} \quad \alpha < 1$$

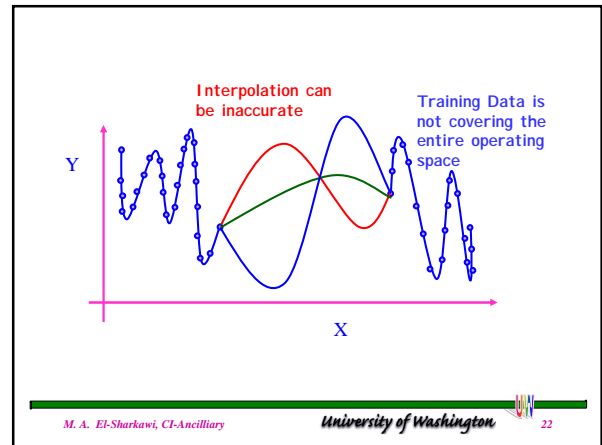
$$w_{ij}^{k+1} = w_{ij}^k + \delta w_{ij}^k - \alpha w_{ij}^{k-1}$$

2. Range of Training Data



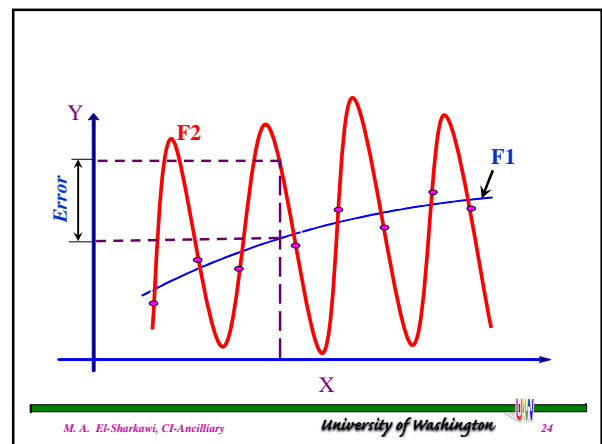
3. Distribution of Training Data

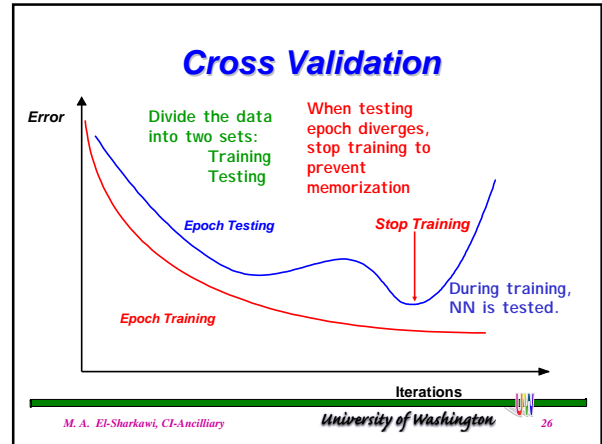
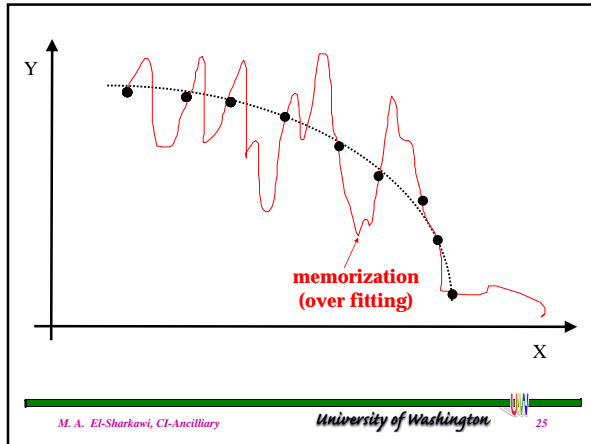
M. A. El-Sharkawi, CI-Ancillary University of Washington 21



4. Learning versus Memorization

M. A. El-Sharkawi, CI-Ancillary University of Washington 23

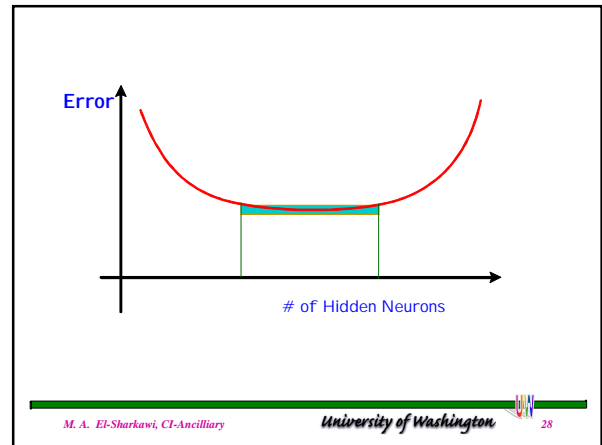




5. Neural Net size

- The number of hidden neurons, must be matched to the complexity of the classification boundary.
- Cross validation among neural networks can provide good approximation of the net size.
- The best NN structure is where the error of the NN is relatively unchanged.

M. A. El-Sharkawi, CI-Ancillary University of Washington 27



6. Data Normalization

M. A. El-Sharkawi, CI-Ancillary University of Washington 29

Data Normalization

- Neural Networks perform better when inputs are appropriately scaled
 - input x has a dynamic range that is 100 times greater than input y , then the network will typically pay more attention to x .
- Normalization Methods:
 - Range Normalization
 - Variance Normalization

M. A. El-Sharkawi, CI-Ancillary University of Washington 30

Data File: Input

	Feature 1	Feature 2	Feature k
Pattern 1 (X_1)	x_{11}	x_{12}	x_{1k}
\vdots	\vdots	\vdots	\vdots	\vdots
Pattern n (X_n)	x_{n1}	x_{n2}		x_{nk}

Data File: Output

	Output 1	Output 2	Output p
Pattern 1 (O_1)	o_{11}	o_{12}	o_{1p}
\vdots	\vdots	\vdots	\vdots	\vdots
Pattern n (O_n)	o_{n1}	o_{n2}		o_{np}

Range Normalization

	Feature 1	Feature 2	Feature k
Pattern 1 (X_1)	x_{11}	x_{12}	x_{1k}
\vdots	\vdots	\vdots	\vdots	\vdots
Pattern n (X_n)	x_{n1}	x_{n2}		x_{nk}

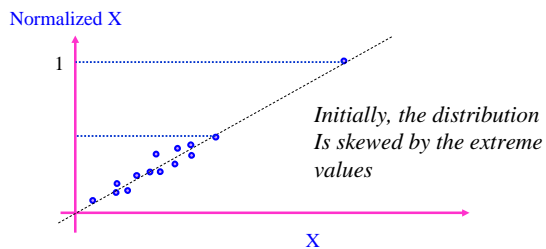
$$Max_1 = \max(x_{11} : x_{n1})$$

$$Min_1 = \min(x_{11} : x_{n1})$$

Range Normalized Input

	Feature 1	Feature 2	Feature k
Pattern 1 (X_1)	$\frac{x_{11} - Min_1}{Max_1 - Min_1}$		$\frac{x_{1k} - Min_k}{Max_k - Min_k}$
\vdots	\vdots	\vdots	\vdots	\vdots
Pattern n (X_n)	$\frac{x_{n1} - Min_1}{Max_1 - Min_1}$			$\frac{x_{nk} - Min_k}{Max_k - Min_k}$

Problems with Range Normalization



Problem with Range Normalization

- Range normalization explicitly uses outliers.
- NN tends to be affected by outliers

Variance Normalization

Mean

$$\mu_i = \sum_{j=1}^n \frac{x_{ji}}{n}$$

Standard Deviation

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^n (x_{ji} - \mu_i)^2}{n-1}}$$

Variance Normalized Input

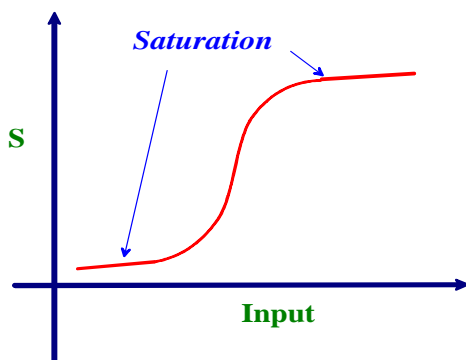
	Feature 1	Feature 2	Feature k
Pattern 1 (X_1)	$\frac{x_{11} - \mu_1}{\sigma_1}$		$\frac{x_{1k} - \mu_k}{\sigma_k}$
⋮	⋮	⋮	⋮	⋮
Pattern n (X_n)	$\frac{x_{n1} - \mu_1}{\sigma_1}$			$\frac{x_{nk} - \mu_k}{\sigma_k}$

Range Normalization

Sample mean and sample standard deviation are less sensitive to outliers.

7. Network Saturation

- Nonlinear functions reaches its limits
- A wide change in the input produce minimal change in the output
- With large number of saturated neurons, the NN can be paralyzed.
- If saturated, neurons must be randomly perturbed.

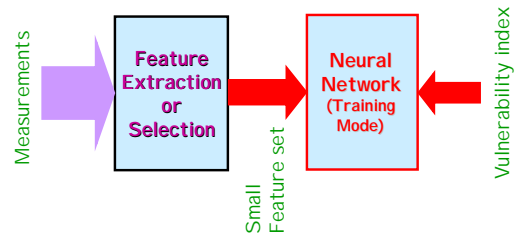


8. Feature Selection and Feature Extraction

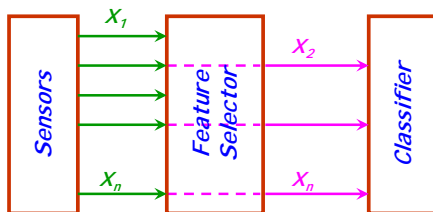
Feature Selection/Extraction

- Eliminates curse of dimensionality.
- Enhances class separability.
- Reduces pattern dimension
- Maintains classification accuracy.
- Reduces training time

Overall System Design: Training

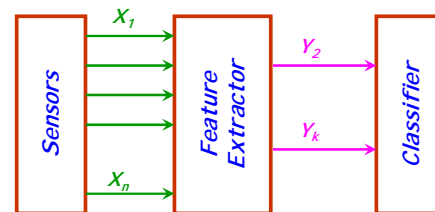


Feature Selection



Most important features are selected
 Techniques: Fisher Discriminate, Genetic Algorithm, Particle Swarm Optimization

Feature Extraction

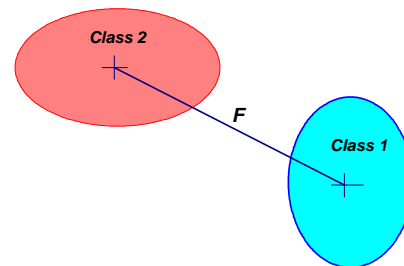


All features are combined to form a new reduced set of features
 Techniques: Principal components, NN

Feature Selection/Extraction

- Class mean Selection (Fisher Selection)
- Karhunen-Loe`ve expansion
- Encoder

Fisher Classifier



Fisher extraction

Original Pattern $Y_j = [y_{1j} \dots y_{dj}]^T$

Inter Class Distance (F)

$$F_i = \frac{|\mu_i(1) - \mu_i(2)|}{|\sigma_i(1)^2 - \sigma_i(2)^2|}, \quad 0 < i \leq d$$

$$\mu_i(\cdot) = \frac{1}{N(\cdot)} \sum_{j=1}^{N(\cdot)} y_{ij}(\cdot)$$

$$\sigma_i(\cdot) = \frac{1}{N(\cdot)} \sum_{j=1}^{N(\cdot)} [y_{ij}(\cdot) - M_i(\cdot)]$$

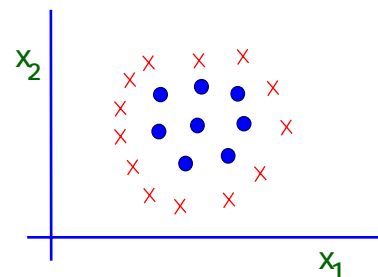
Fisher Classifier

- A heuristic measure of inter-class distance
- Dominant indices are selected
- Dimension of pattern vectors can be substantially reduced.
- Pattern vectors for each class occupies a distinct region in the observation space.

Class-Mean Feature extraction

- **Advantages:**
 - First order statistics (Mean); Fast!
 - Feature variables retain their physical identity
- **Drawbacks:**
 - Data must have an inter-class distance
 - Fails for concentric or near concentric data
 - Best n features is not the same as n best features

Concentric data



Karhunen-Loe`ve expansion

- **Original pattern:**
 $[x_{i1} \ x_{i2} \ \dots \ x_{in}]^T; \quad i=1,2, \dots, M$
- **Reduced pattern:**
 $[y_{i1} \ y_{i2} \ \dots \ y_{id}]^T \quad d \ll n$

Karhunen-Loe`ve expansion

Original pattern of one class

$$X_i = [x_{i1} \ \dots \ x_{in}]^T, \quad i = 1, 2, \dots, M$$

Pattern Mapping $X_i = \sum_{j=1}^n y_{ij} \Phi_j$

Φ is orthonormal function, y_{ij} is feature variable set

$$\Phi_i^T \Phi_j = \begin{cases} 1 & \text{for } i=j \\ 0 & \text{for } i \neq j \end{cases}$$

Selection of Φ

Error Index

$$J = E \left\{ \left(X_i - \sum_{j=1}^n y_{ij} \Phi_j \right)^2 \right\}$$

Minimum Index

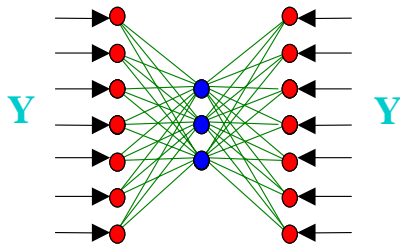
$$J = \sum_{j=1}^n \lambda_j$$

IFF λ_j and Φ_j are the eigenvalues and eigenvectors of $E_i [X_i X_i^T]$

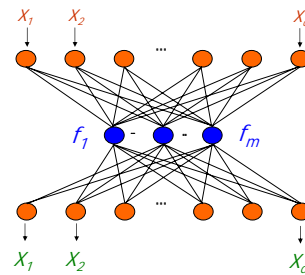
Karhunen-Loe`ve expansion

- **Reduction is second order statistics (Variance)**
- **Linearly combines the original set to form a set with better separable features**
- **New features are not physically meaningful**

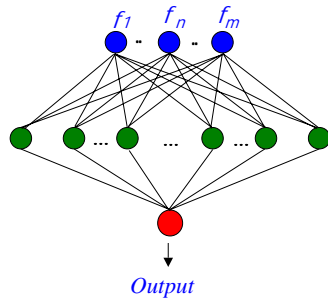
NN Encoder



Feature Extraction: Step 1



Feature Extraction: Step 2



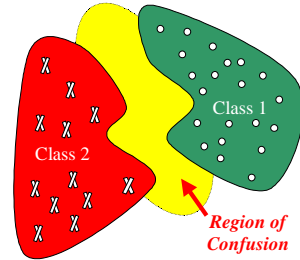
9. Inversion of NN

Inversion of NN

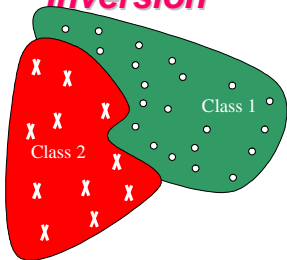
- The accuracy of a classifier is dependent on the amount and quality of training data.
- Training data should be sufficient to cover the entire space of system operation.



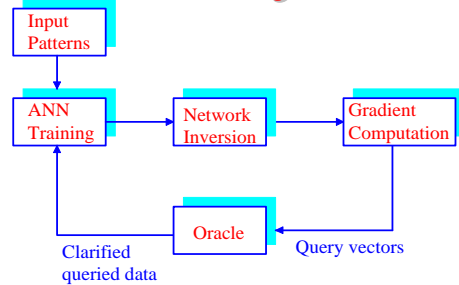
Class Separation without Inversion



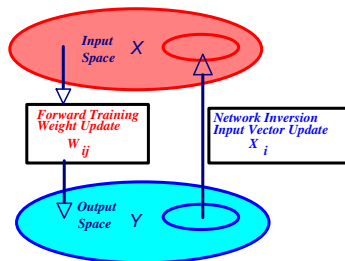
Class Separation with Inversion



Inversion Algorithm



Network Inversion



Inversion Algorithm

Error function

$$E(\bar{W}) = \frac{1}{2} \sum_i (d_i - y_i)^2$$

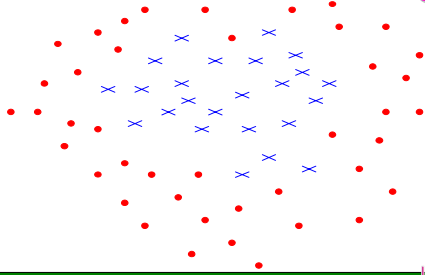
Weight Update

$$\bar{W}(k+1) = \bar{W}(k) - \eta \frac{\partial E[\bar{W}(k)]}{\partial \bar{W}(k)}$$

Input Update

$$\bar{X}(k+1) = \bar{X}(k) - \eta \frac{\partial E[\bar{X}(k)]}{\partial \bar{X}(k)}$$

Border Marking



Border Marking

