

27244
P-31
1995 107756

FAULT DETECTION & DIAGNOSIS USING NEURAL NETWORK APPROACHES

Prof. Mark A. Kramer

ABSTRACT

Neural networks can be used to detect and identify abnormalities real-time process data. Two basic approaches can be used, the first based on training networks using data representing both normal and abnormal modes of process behavior, and the second based on statistical characterization of the normal mode only. Given data representative of process faults, radial basis function networks can effectively identify failures. This approach is often limited by the lack of fault data, but can be facilitated by process simulation. The second approach employs elliptical and radial basis function neural networks and other models to learn the statistical distributions of process observables under normal conditions. Analytical models of failure modes can then applied in combination with the neural network models to identify faults. Special methods can be applied to compensate for sensor failures, to produce real-time estimation of missing or failed sensors based on the correlations codified in the neural network.

BIOGRAPHY

Mark A. Kramer is currently Associate Professor of Chemical Engineering at the Massachusetts Institute of Technology. Professor Kramer received a Bachelor's degree at the University of Michigan in 1979 and a Ph.D. degree from Princeton University in 1983. He also has served as the Associate Director of the Laboratory for Intelligent Systems in Process Engineering at MIT. Professor Kramer has published over 45 papers in the areas of artificial intelligence in process engineering and neural networks.

)

)

)

Fault Detection & Diagnosis using Neural Network Approaches

Mark A. Kramer
Massachusetts Institute of Technology
Department of Chemical Engineering

Mission Safety Critical Systems: Research & Applications
Research Institute for Computing and Information Systems
University of Houston-Clear Lake
October 29, 1992

How to use neural networks to:

- 1) *detect*
- 2) *identify*
- 3) *rectify*

faults in processes and associated sensors.

Using neural nets involves **learning from examples.**

Two approaches:

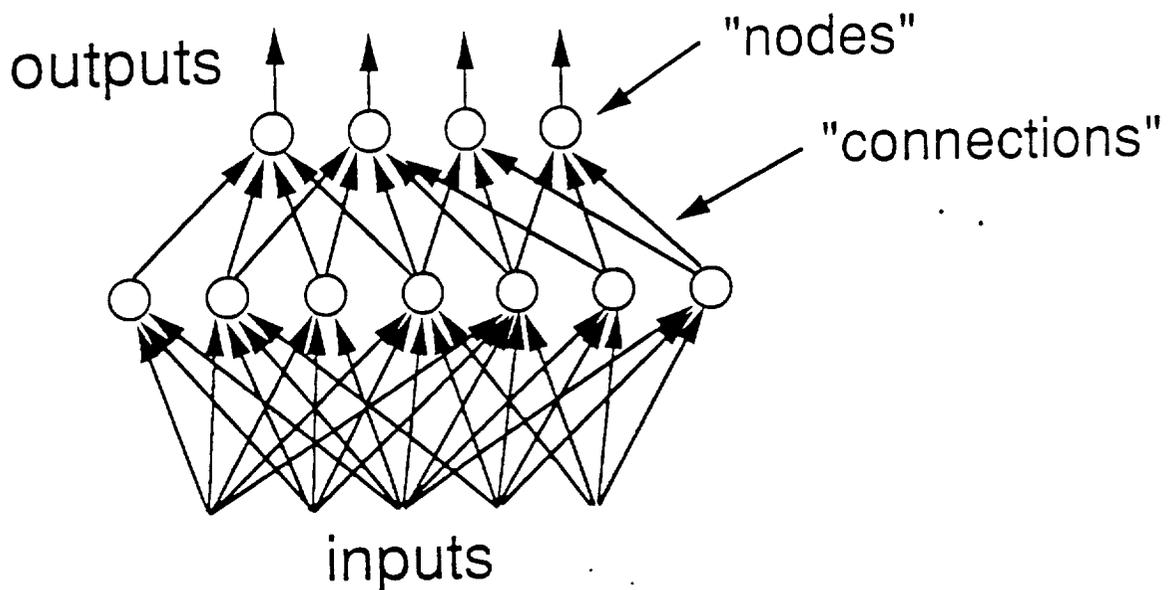
- 1) Learning with examples of normal and abnormal behavior
- 2) Learning with examples of normal behavior only

Useful in the absence of a **functional theory** of device behavior.

REVIEW OF NEURAL NETWORKS

Artificial neural networks are loosely based on how the brain carries out its low-level computations.

- Simple computational elements acting in parallel
- Neurons "fire" when excited by other neurons
- Capable of learning and responding differently to different input patterns



Input/output behavior determined by:

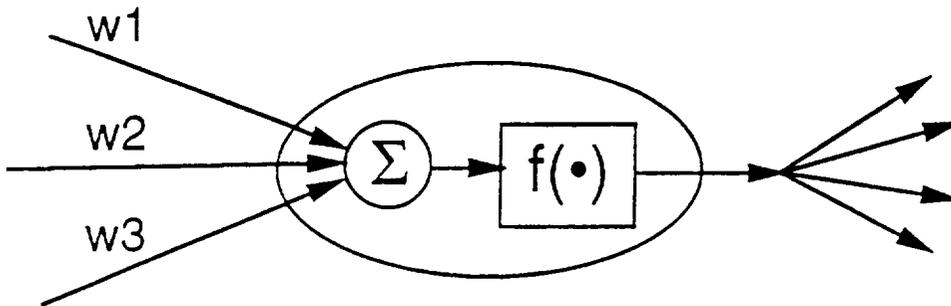
- Topology of network
- Computation of each neuron
- Adjustable parameters of connections and nodes

Two of the most important types of networks are:

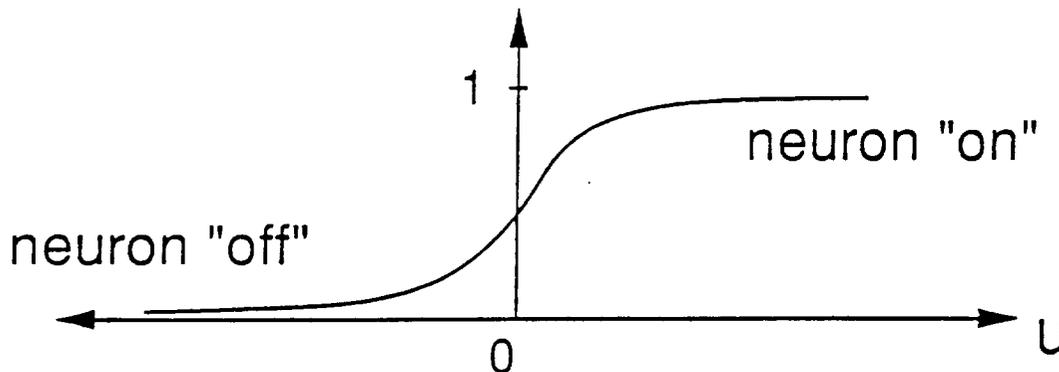
- **backpropagation networks (BPNs),**
- **radial basis function networks (RBFNs)**

Computation in Backpropagation Networks

- Layered architecture, usually input layer, output layer, plus one intermediate "hidden" layer
- Connections between nodes are weighted. Weights multiply the signal on the connection.
- Each node sums its inputs and then passes the result through a **sigmoidal** nonlinearity

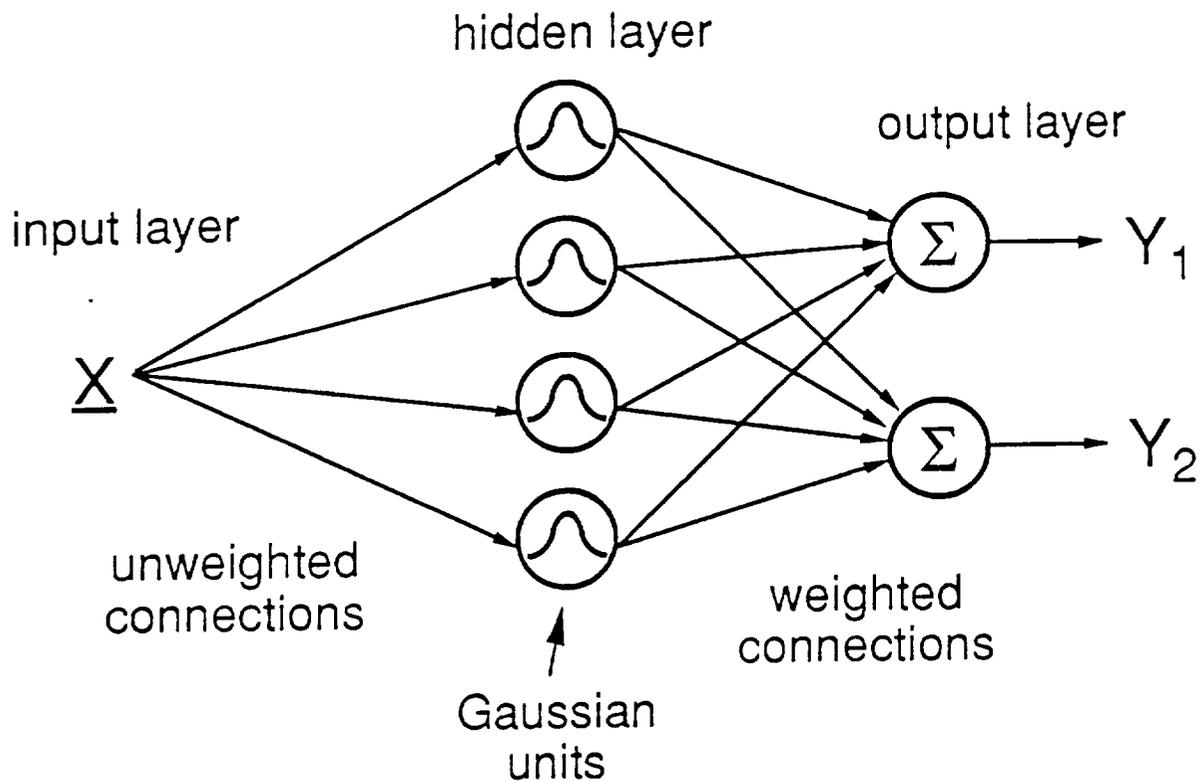


Typical sigmoid function: $f(u) = 1/(1+\exp(-u))$



Computation in Radial Basis Function Networks

- Similar to BPN but uses **Gaussian** nonlinearity in nodes
- Input/hidden layer connections not weighted, simply pass input vector \underline{X} to hidden layer



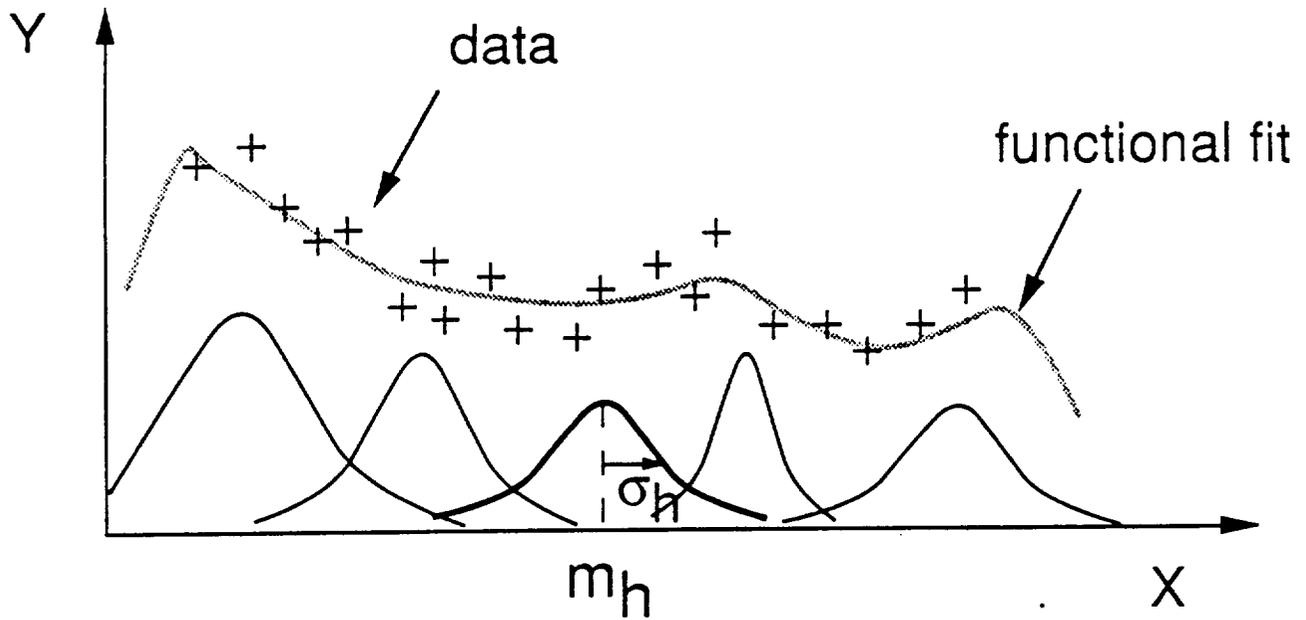
Each Gaussian unit has internal parameters representing a "unit center" \underline{m} and a "receptive width" σ

Output of Gaussian unit i , a_i , is based on the distance between inputs \underline{x} and the unit center \underline{m} :

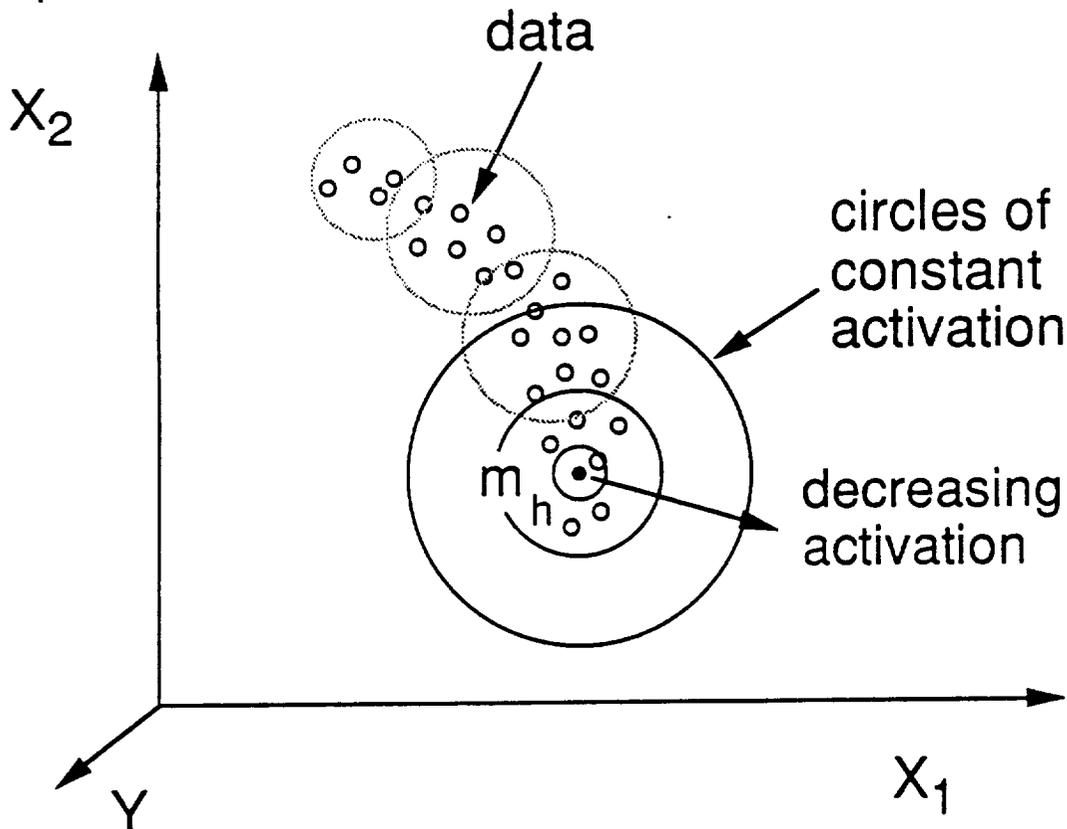
$$a_i = \exp\left(\frac{-\|\underline{x} - \underline{m}_i\|^2}{\sigma_i^2}\right)$$

Graphical interpretation of hidden nodes of RBFN:

One input dimension:



Multiple dimensions:



NEURAL NETWORK LEARNING

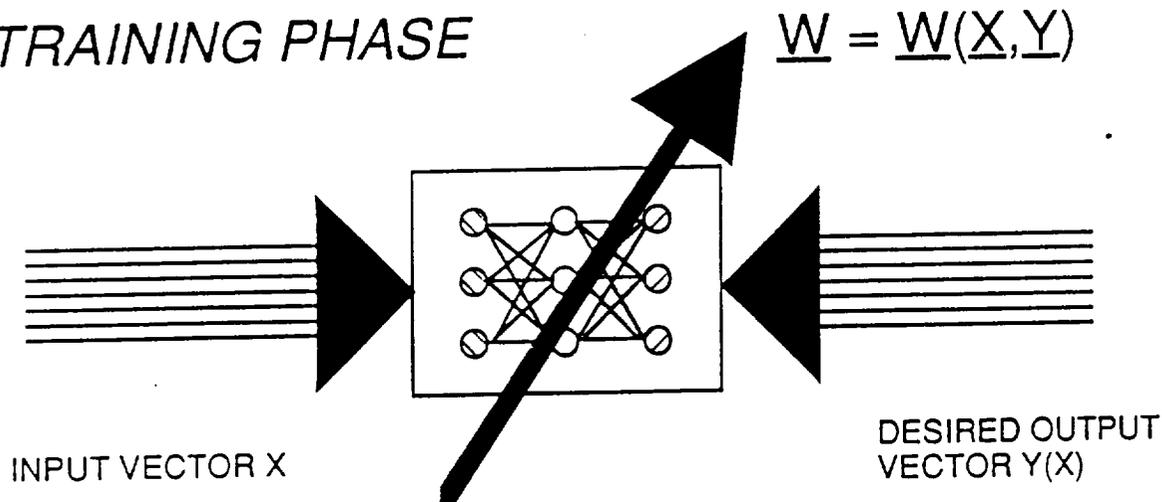
Takes place through an optimization of internal connection weights W.

A set of examples of desired input/output behavior $(\underline{x}, \underline{y})_i, i=1, \dots, K$ is required.

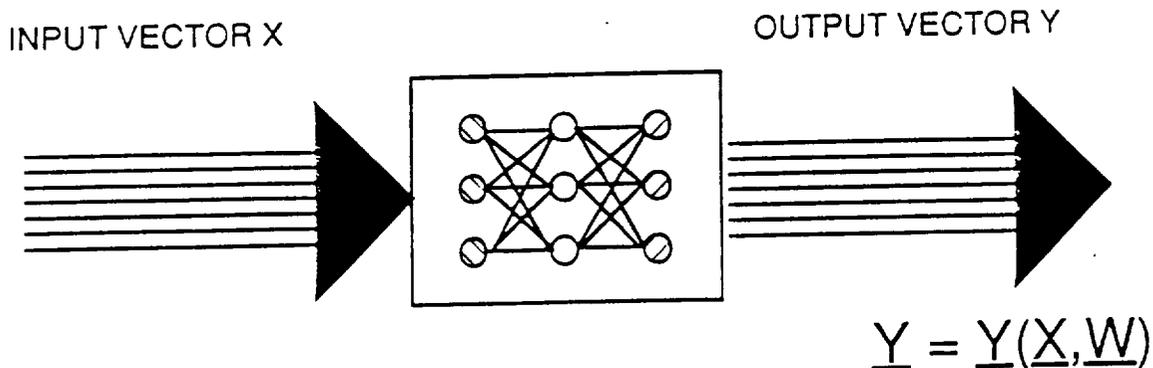
A **least-squares** fit is sought:

$$\min_w \sum_{i=1}^K [y_i - \text{Net}(x_i)]^2$$

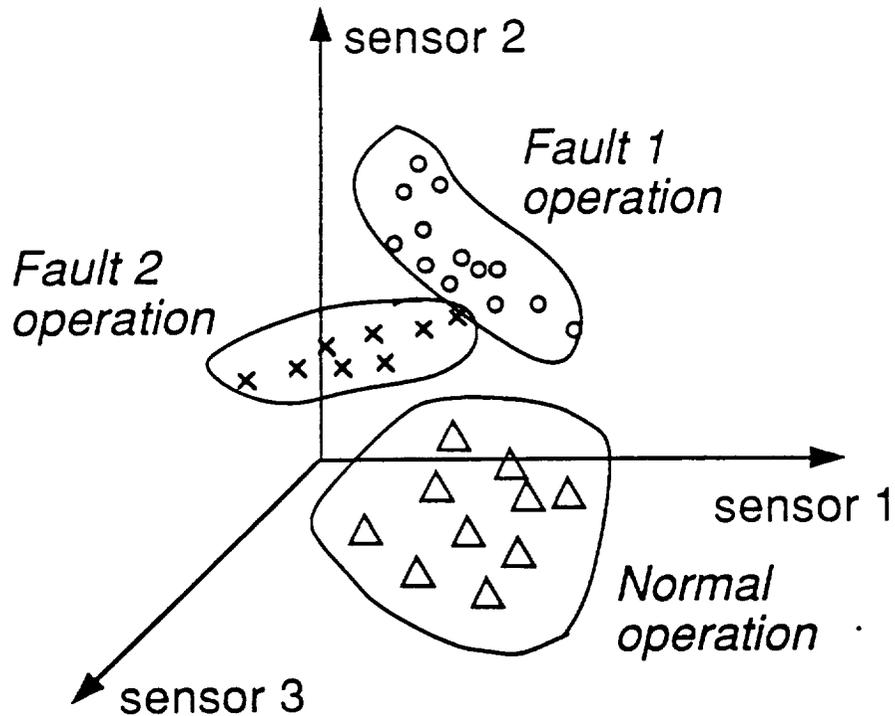
TRAINING PHASE



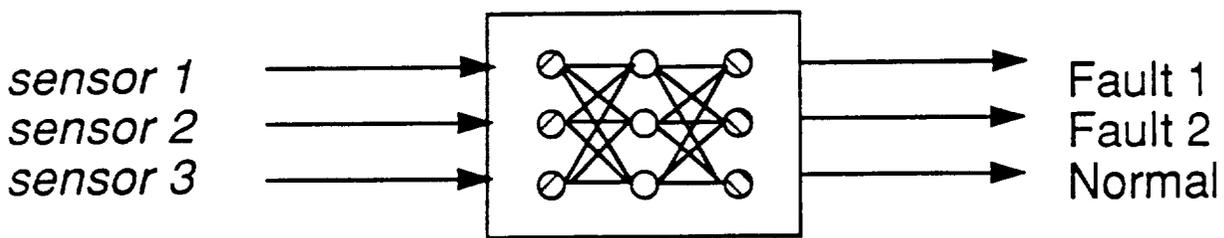
APPLICATION PHASE



Multi-Class Diagnosis Using Neural Networks



Neural network is capable of identifying more complex class regions than "high-low-normal"-style rules



NEURAL NETWORK

Network Training For Multiclass Diagnosis

<u>observables x</u>	<u>diagnosis</u>	<u>y</u>
{0.032, 0.099, -0.039}	→ fault 1	→ {1, 0, 0}
{0.016, -0.53, -0.465}	→ fault 2	→ {0, 1, 0}
{0.466, 0.022, -0.405}	→ fault 3	→ {0, 0, 1}

Example inputs x:

Feedstock characterization: {sp. grav., bubble pt., visc.,...}

Sensor data: {meas 1, meas 2,...}

Time series: {meas(t), meas(t-1), meas(t-2), ...}

Using least squares objective function, assuming:

- 1) Sufficient # of training examples
- 2) Examples in proportion to prior probabilities
- 3) Adequate network representational capacity

Then:

$$y_i = \frac{P(\text{fault } i \mid x)}{\sum_j P(\text{fault } j \mid x)} = \begin{array}{l} \text{relative} \\ \text{probability} \\ \text{of fault } i \end{array}$$

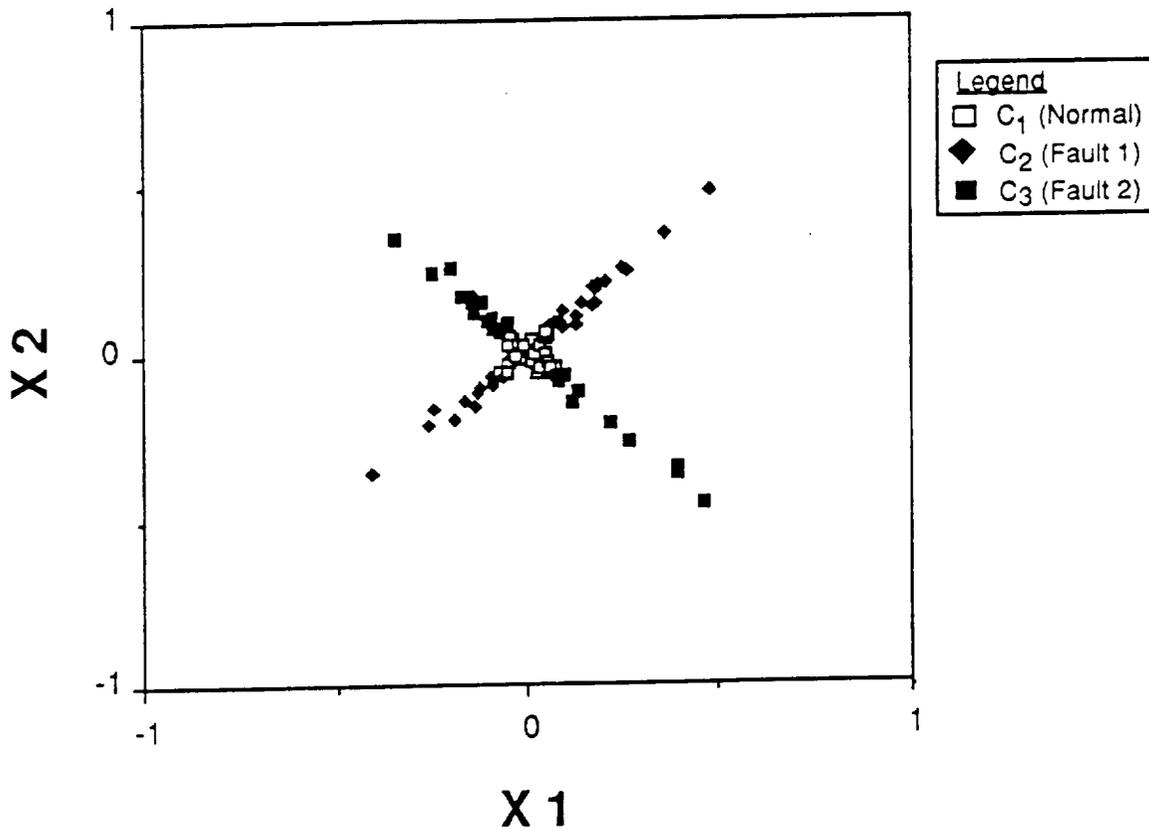
$\sum y_i \neq 1$ implies an invalid classification

RADIAL BASIS FUNCTION NETWORKS ARE
BETTER FOR DIAGNOSTIC PROBLEMS THAN
BACKPROPAGATION (SIGMOIDAL) NETS

Fault diagnosis example problem

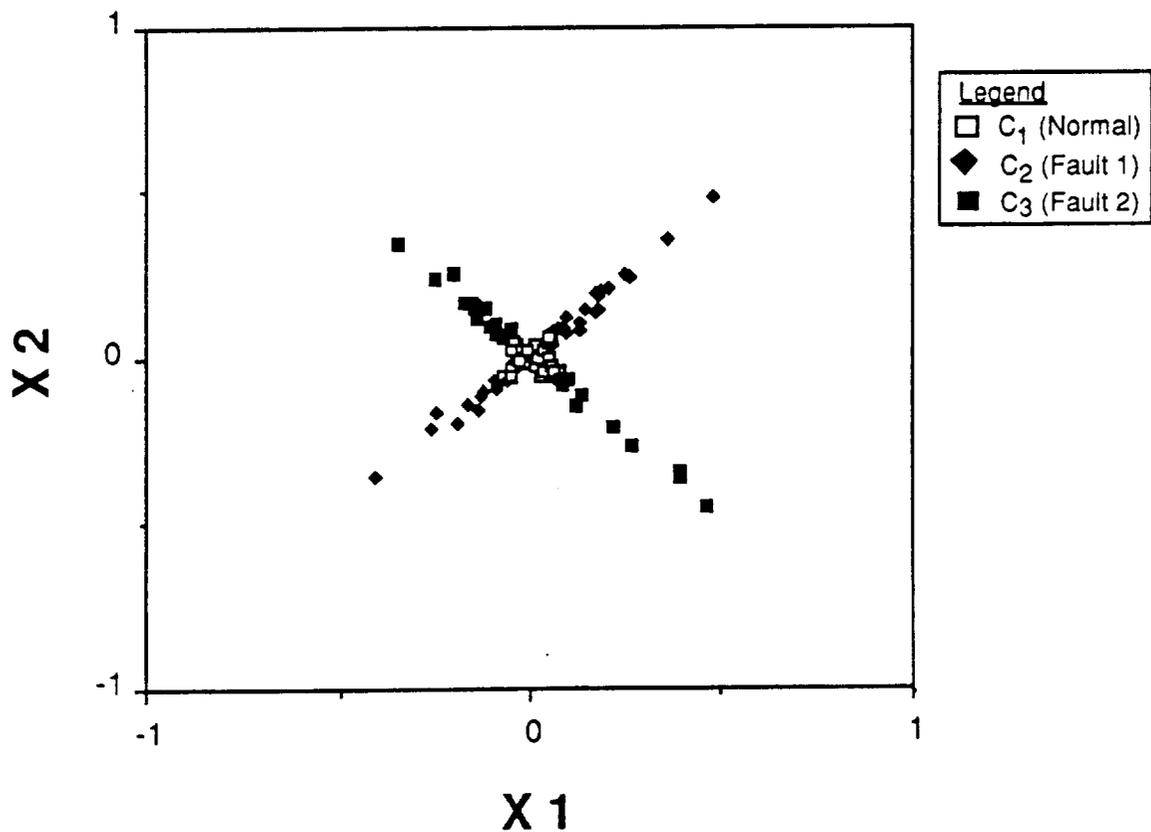
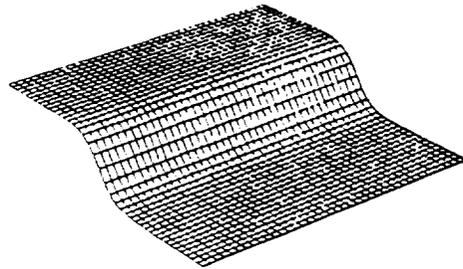
(see Kramer & Leonard, IEEE Control Systems 11, 31, April 1991)

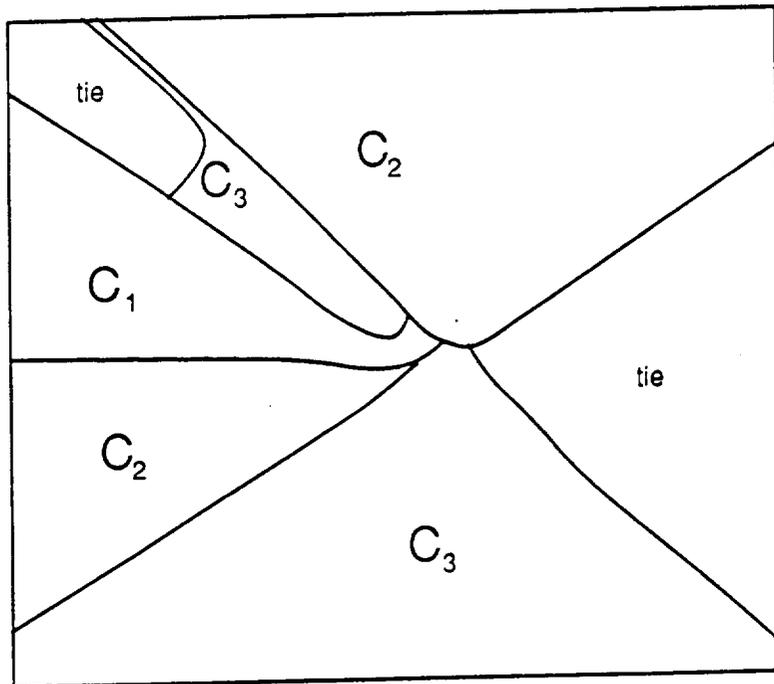
3 classes, 2 input dimensions
30 training examples of each class.



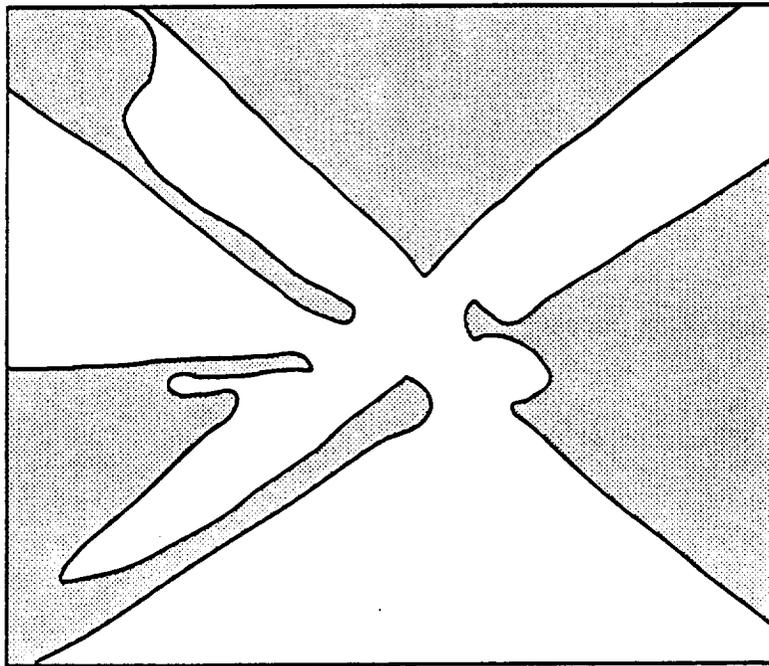
Backpropagation networks (sigmoidal nodes):

- Class regions divided with hyperplanes
- Tends to place class boundaries near "edge" of class
- Check sum $\sum Y_i \neq 1$ indicates some regions of insufficient training data (sufficient but not necessary)





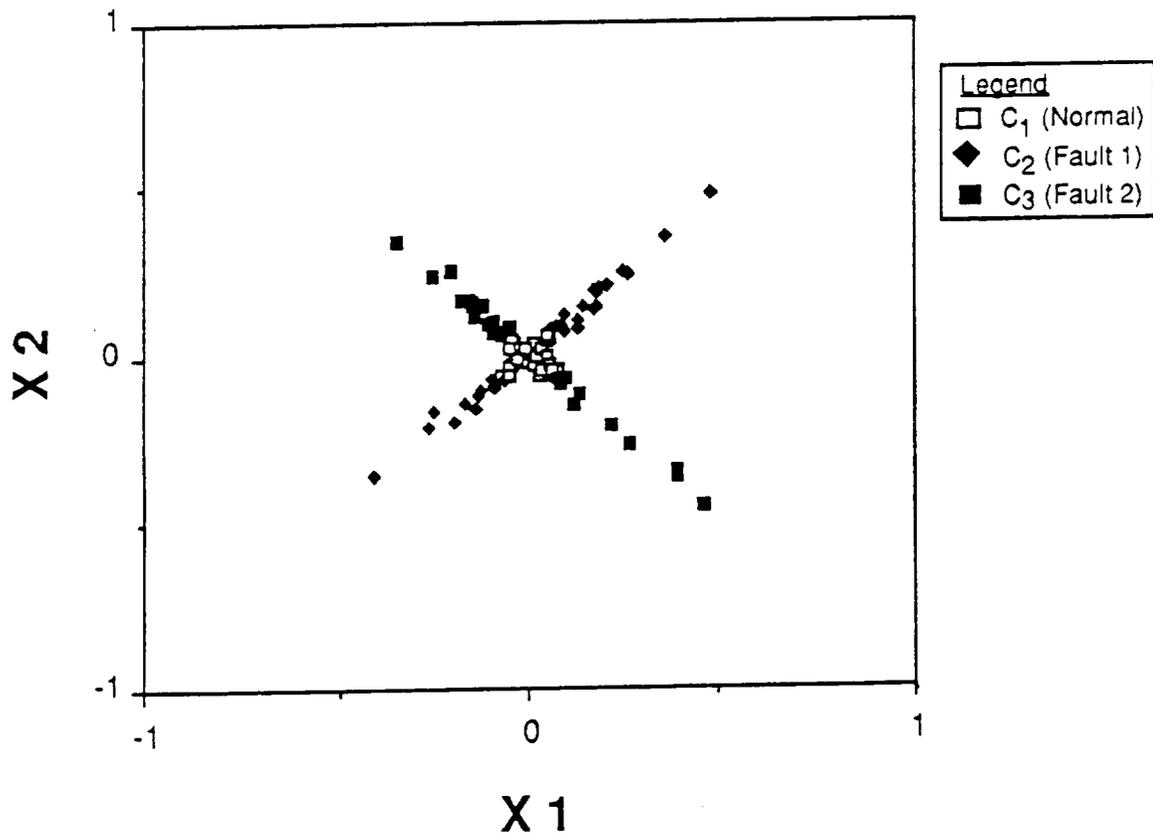
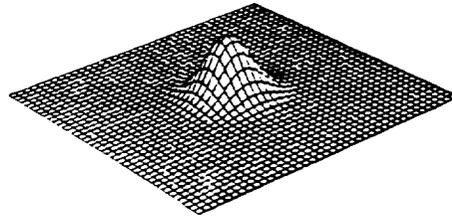
DECISION REGIONS

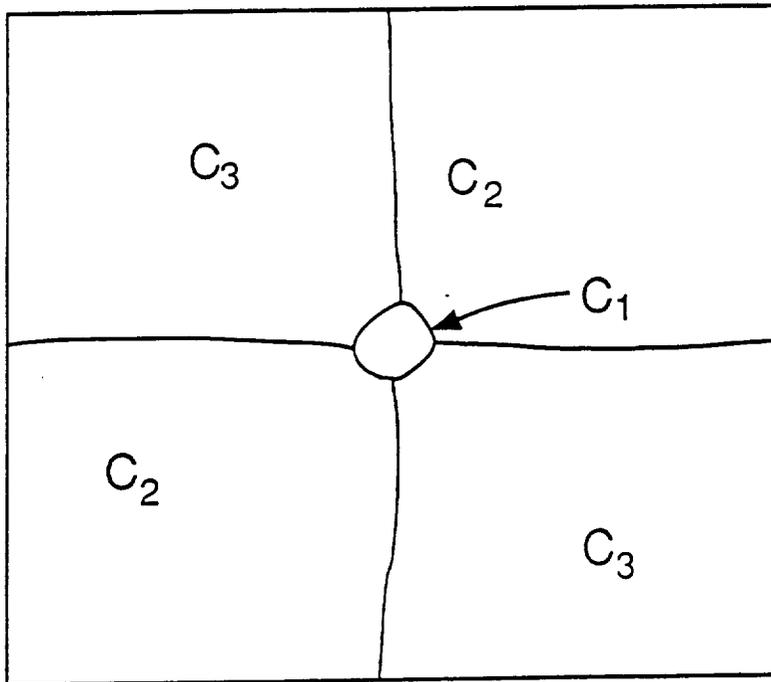


$$0.9 < \sum Y_i < 1.1$$

Radial Basis Function Networks (Gaussian nodes):

- Well-placed classification boundaries
- Check sum $\sum Y_i = 1$ tends to be satisfied *everywhere* (i.e. no regions flagged as novel)

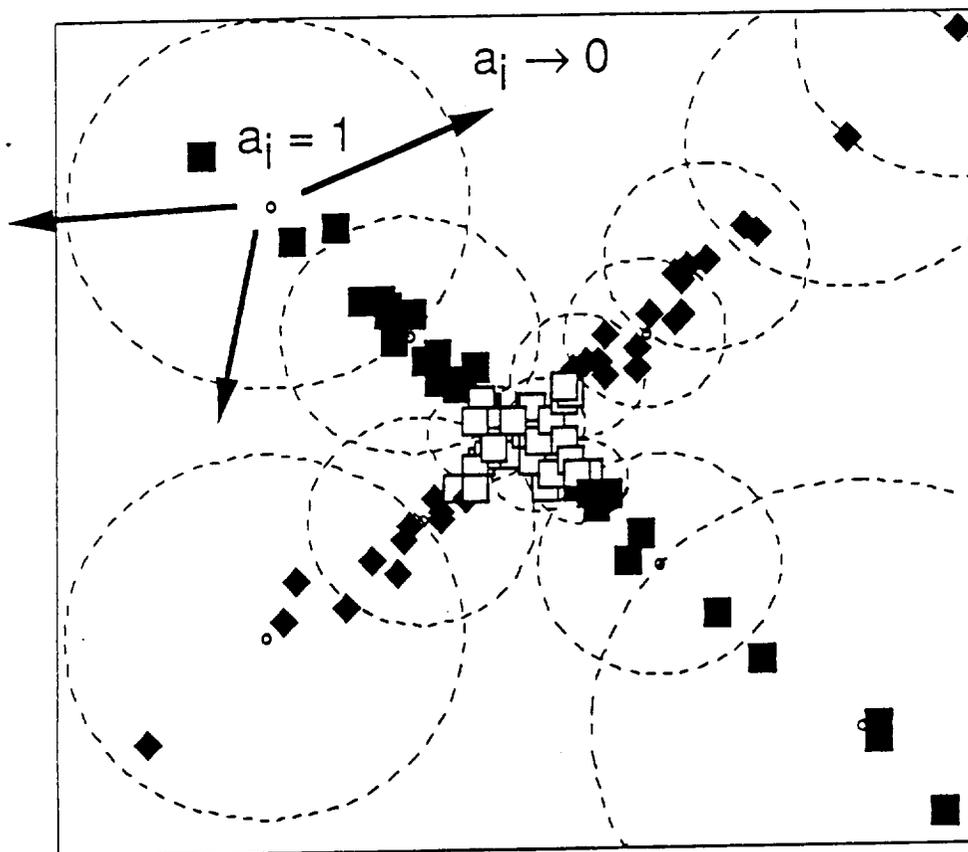




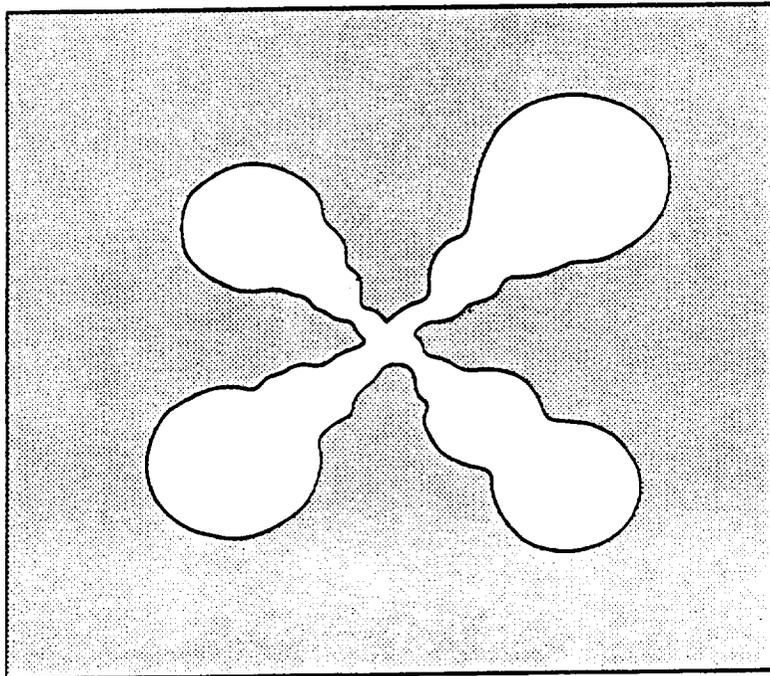
DECISION REGIONS

Closer look at novelty in RBFNs:

- Radial units are centered among groups of data by k-means clustering.
- Gaussian activation functions $a(x)$ decrease to 0 as one moves away from unit center



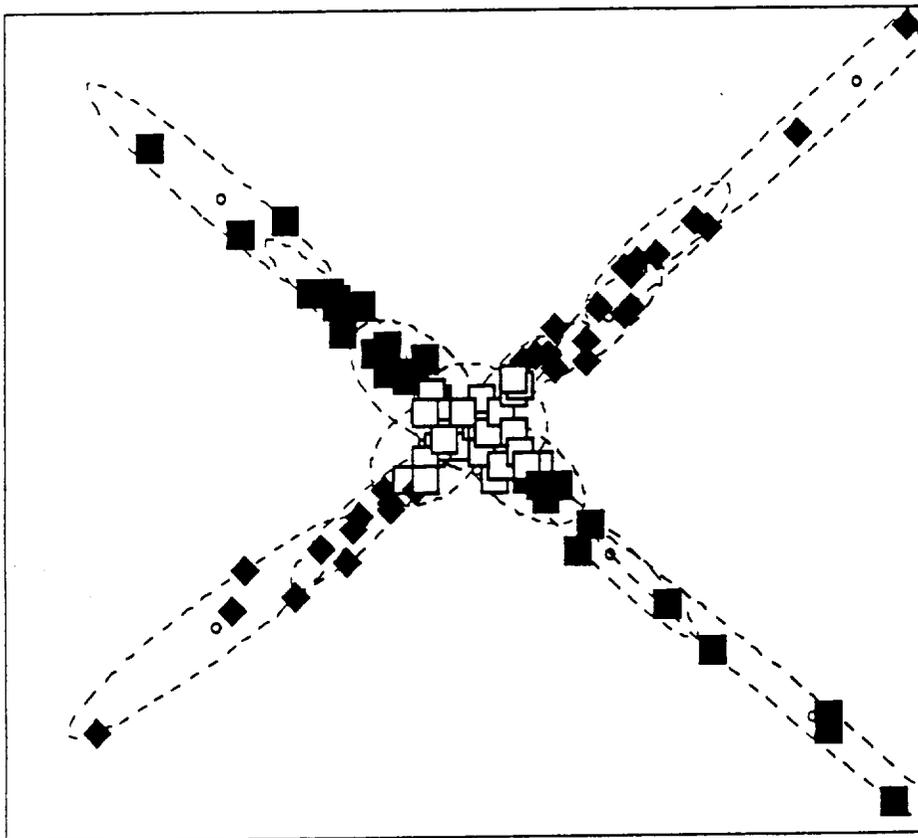
- Novelty can be indicated by $\max(a_i) < \text{cutoff value}$ (e.g. 0.5)
- Works even better with elliptical units



Hidden node
activation > 0.5

Elliptical Basis Function Networks (EBFN)

- Similar to RBFN but unit shapes can be elliptical
- Shapes determined by local covariance structure of data
- Good novelty detection and classification properties



Elliptical Basis Function Coverage of
Fault Classification Data

Data Density Estimation using RBFNs

In each radial or elliptical unit, local data density is approximately:

$$\rho_h = \frac{\text{\# data points local to unit } h}{\text{Volume of unit } h \cdot \text{total \# data points}}$$

A smooth data density estimate at every point in space then given by the interpolation formula:

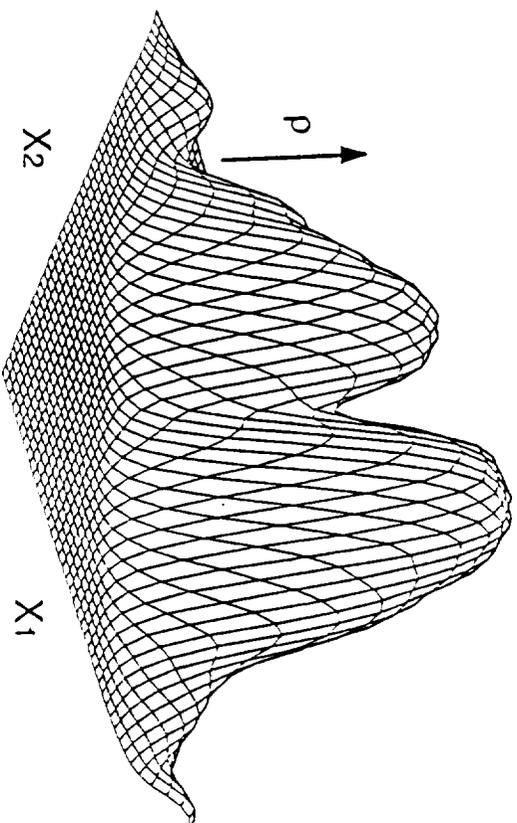
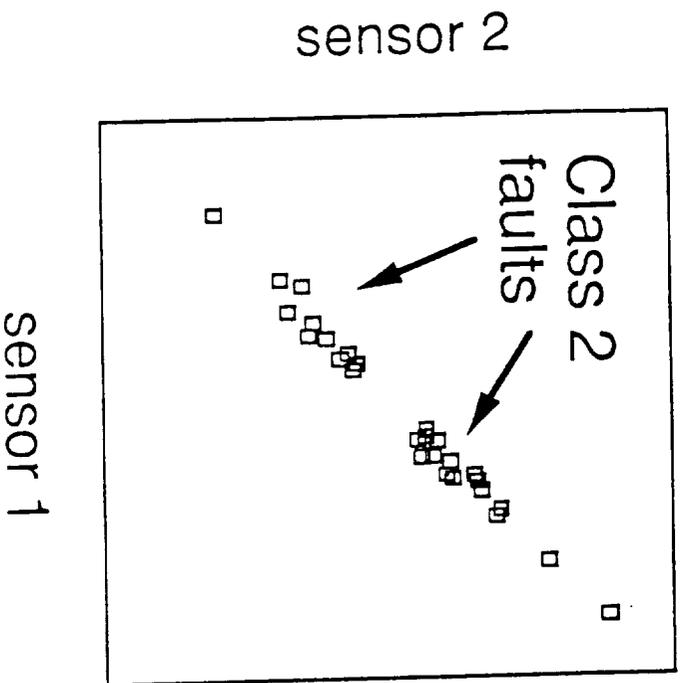
$$\rho(\underline{x}) = \frac{\sum_{h=1}^H a_h(\underline{x}) \rho_h}{1 - \max(a_h) + \sum_{h=1}^H a_h(\underline{x})}$$

Uses of probability density function:

- 1) Class-based decomposition of classifier
- 2) Fault detection using only normal data
- 3) Rectification of sensor faults

(See Leonard, Kramer & Ungar, Comput Chem Eng, vol. 16, 819, 1992)

CLASS-SPECIFIC DENSITY ESTIMATION



Estimated density

Class-Based Network Decomposition

$$p_i(\underline{x}) = P(\underline{x} | H_i)$$

Density related to posterior fault probability conditional on data via Bayes' Theorem:

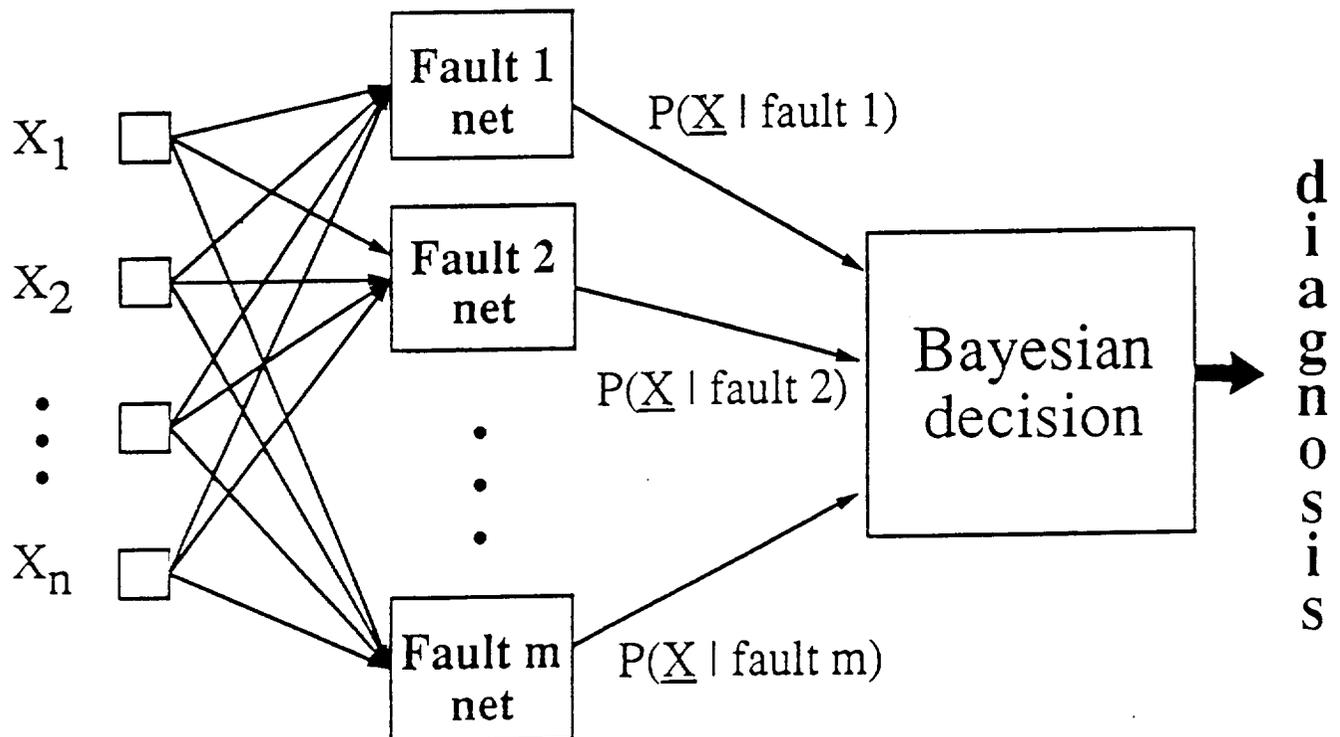
$$P(H_i | \underline{x}) = P(\underline{x} | H_i) P(H_i) / P(\underline{x})$$

$P(\underline{x})$ is pooled density function (all classes).

Relative probability, eliminate $P(\underline{x})$:

$$R_{ik} = P(\underline{x} | H_i) P(H_i) / P(\underline{x} | H_k) P(H_k)$$

Class-decomposed network:



Decomposition benefits:

- No regression of weights

Work savings = $O(HM/N)$

H = # hidden nodes

M = # faults

N = # inputs

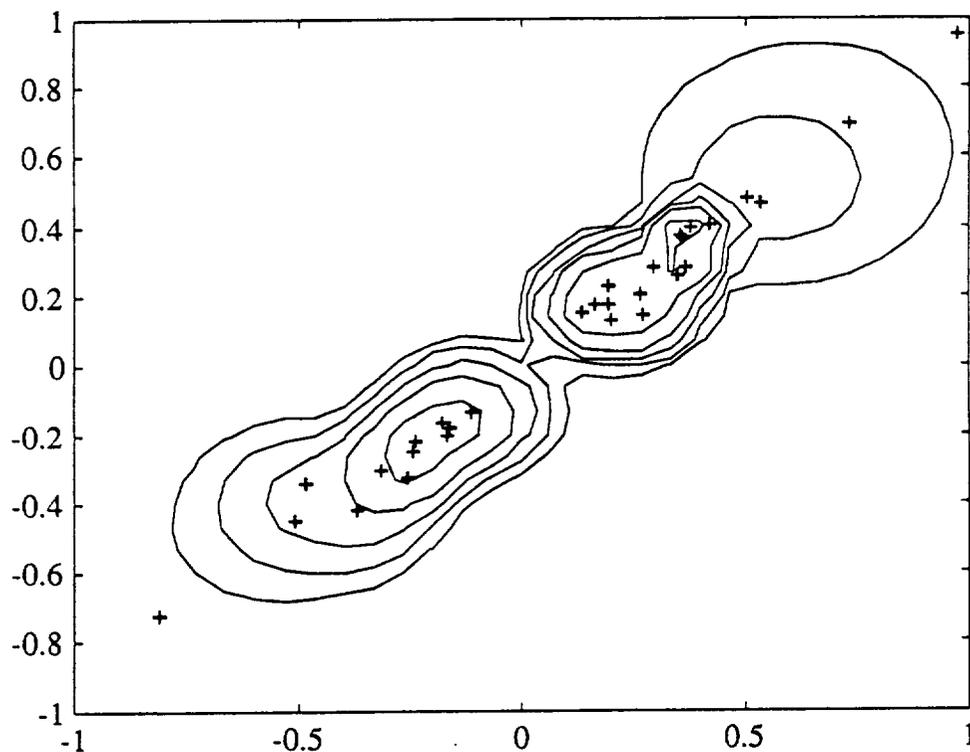
- Allows incremental development of classifier, easy incorporation of new data
- Prior probabilities and misclassification costs can be incorporated in Bayesian decision

FAULT DETECTION USING DENSITY FUNCTION

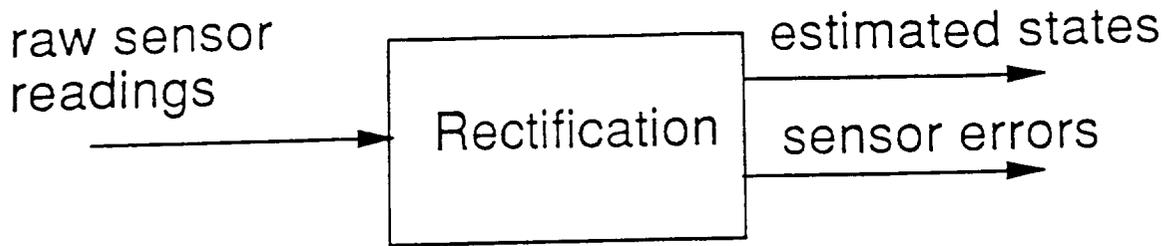
Fault detection: Is current state in the normal class, or out?

Fault data not needed.

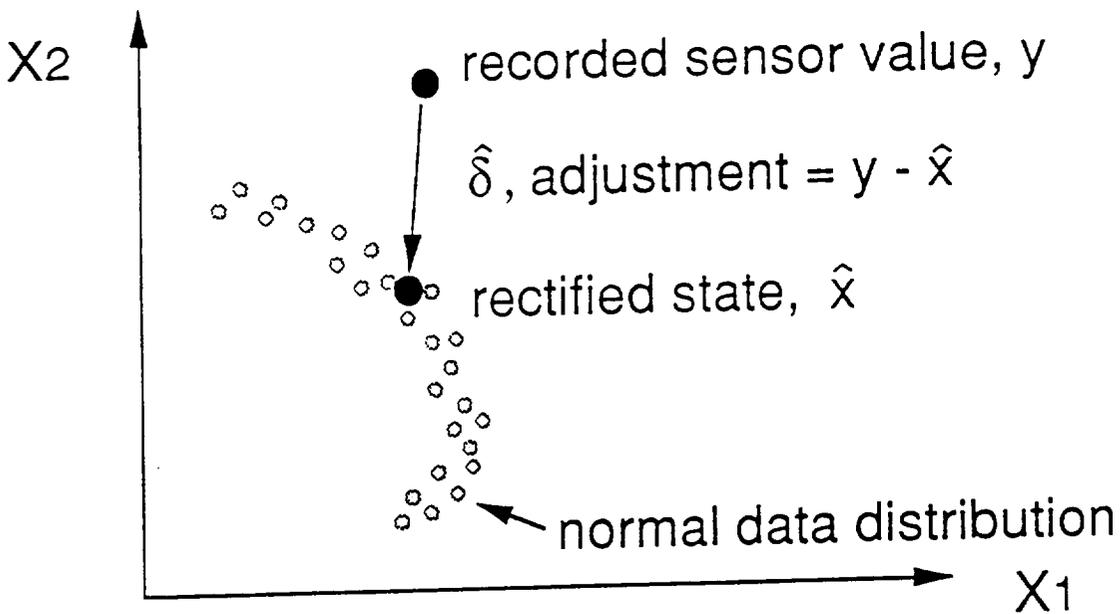
- Model probability density function of normal class
- Place probability limits, e.g. 95% for declaring fault



Rectification of sensor faults by probability optimization



- Assume model of normal probability distribution
- Hypothesize sensor failures only



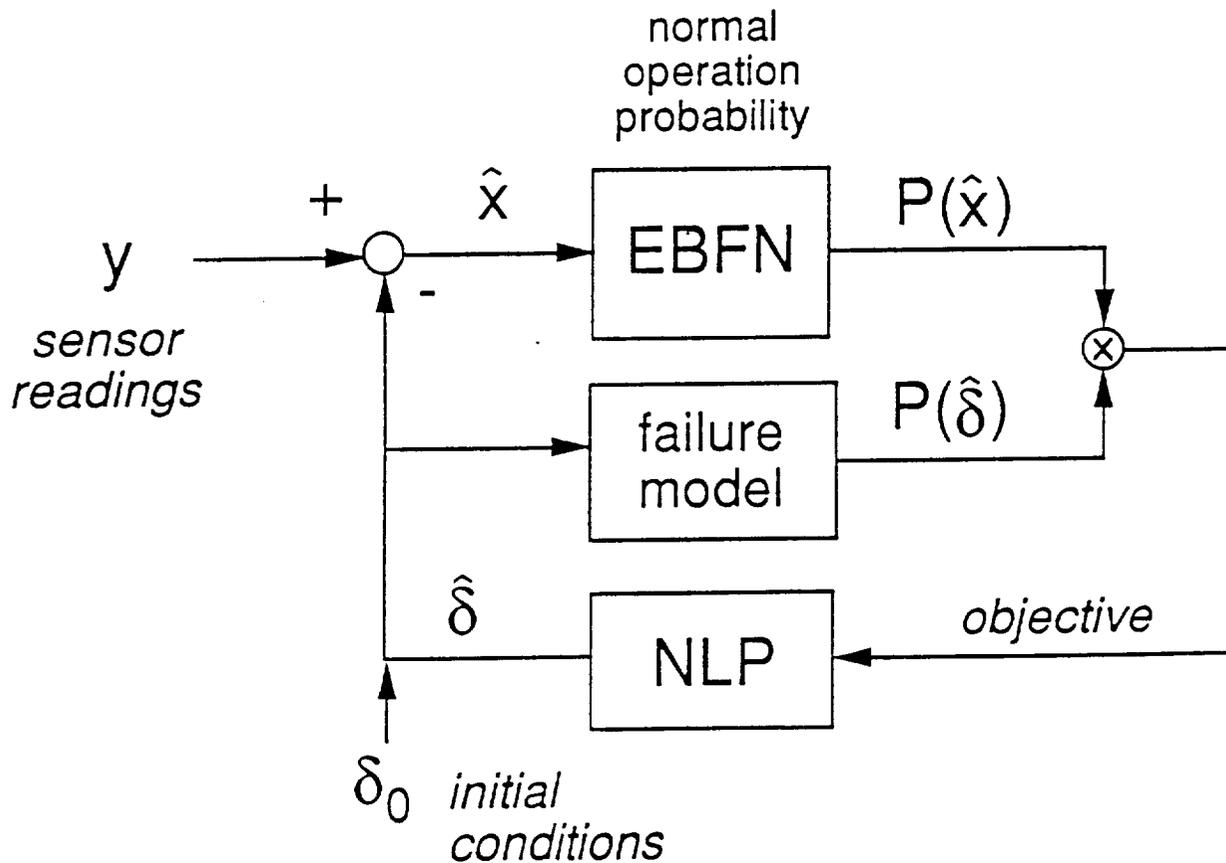
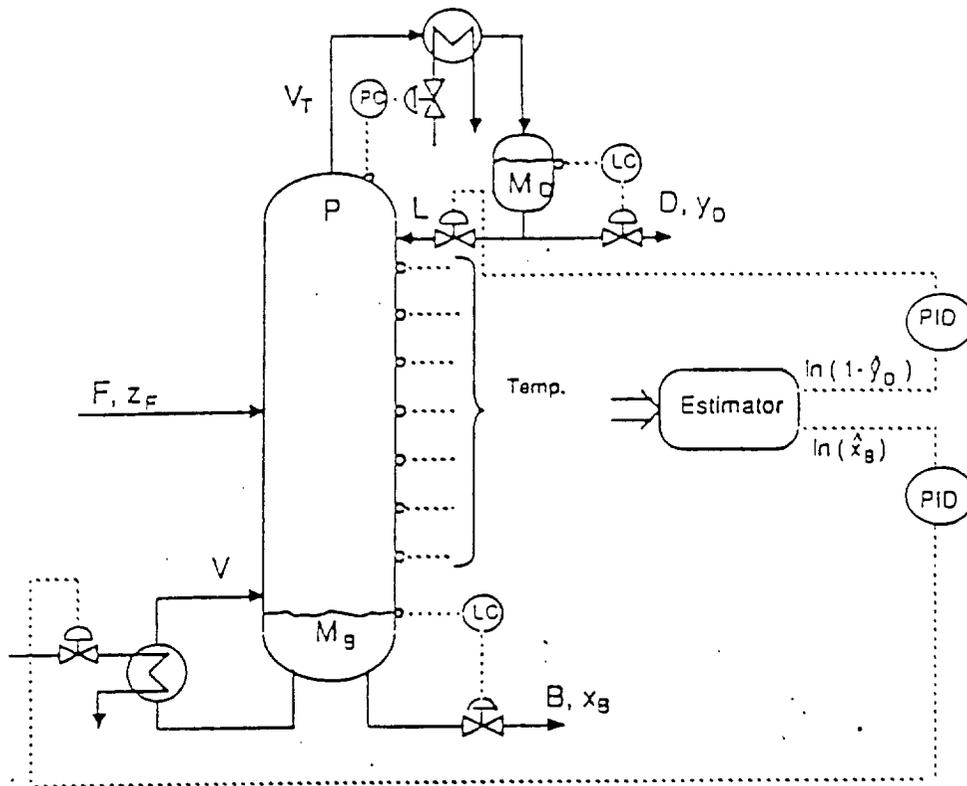
$$\text{maximize } P(\hat{x}|y) \propto P(\hat{\delta})P(\hat{x})$$

\hat{x}

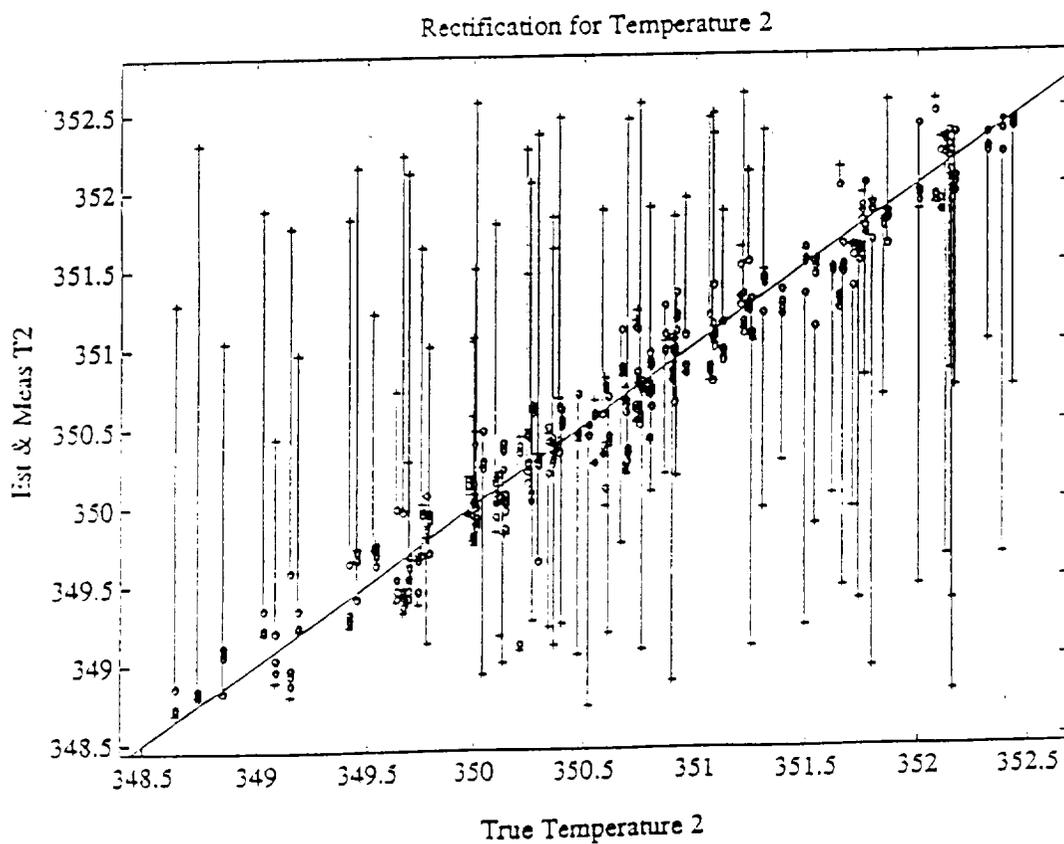
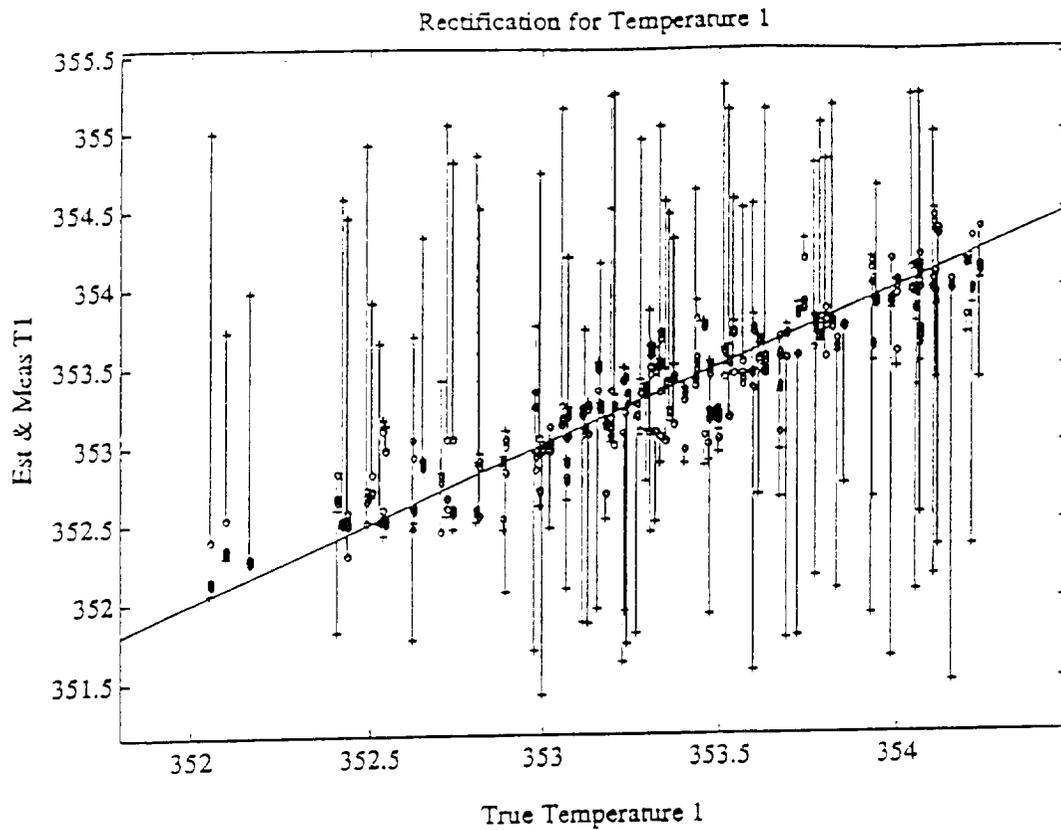
likelihood of
adjustment

likelihood of
rectified state

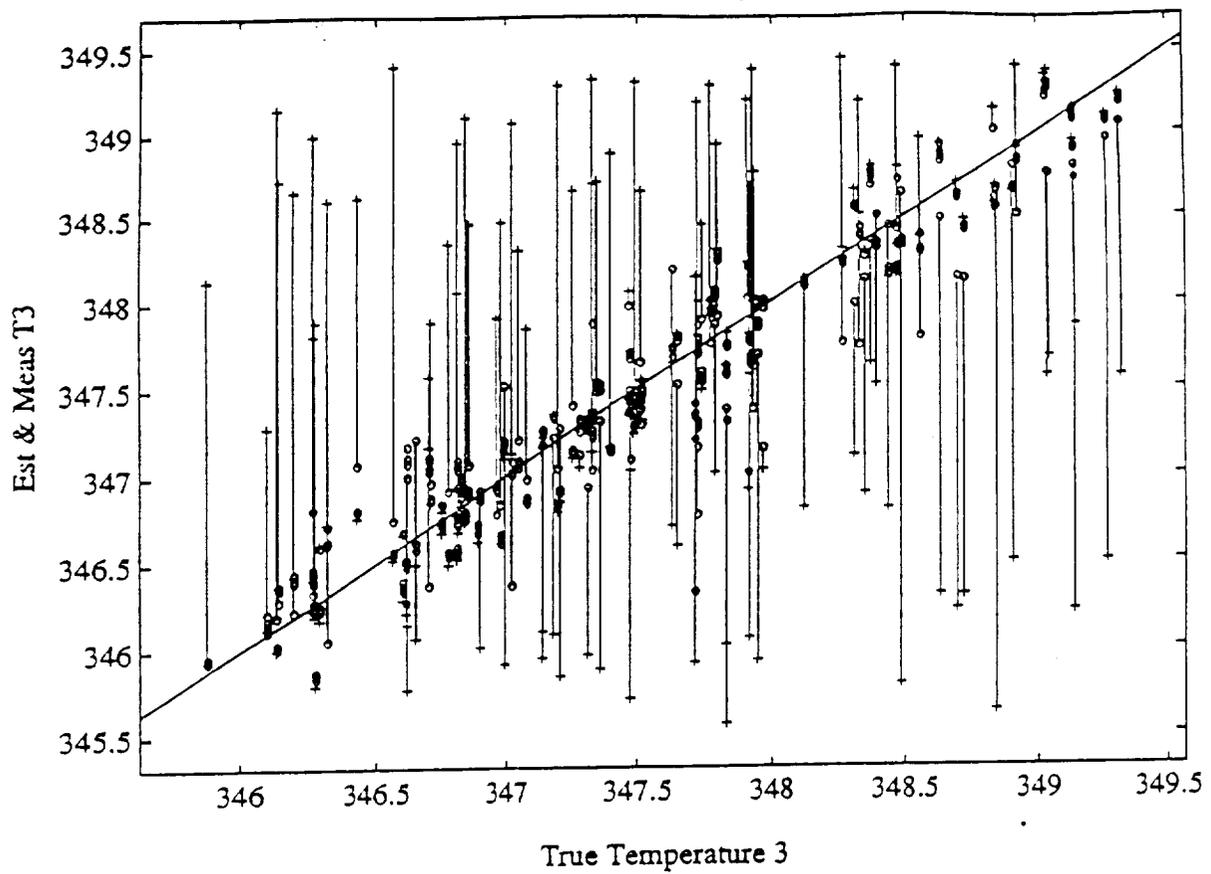
Example: Plate temperature rectification in distillation



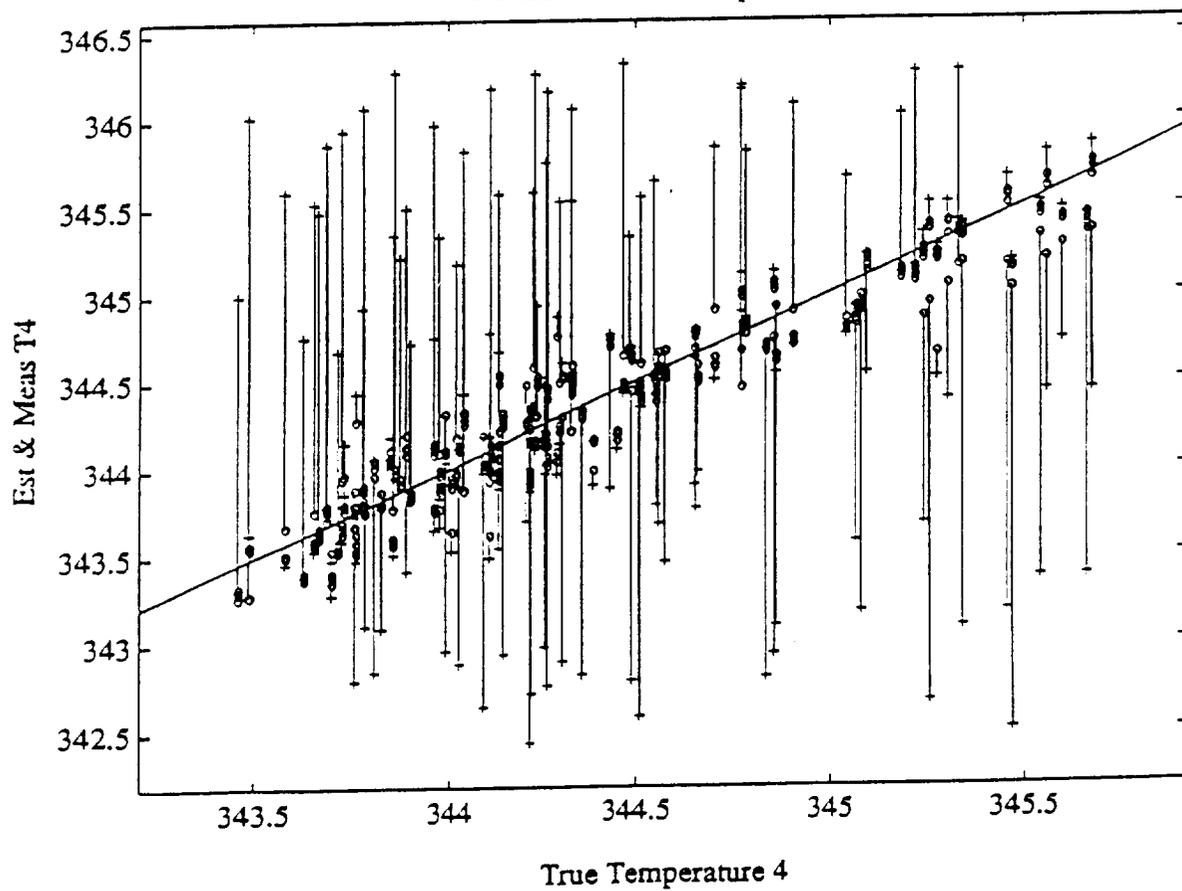
Test: Corrupt each of 5 sensors in test set of 100, yielding 500 examples with single sensor failure.



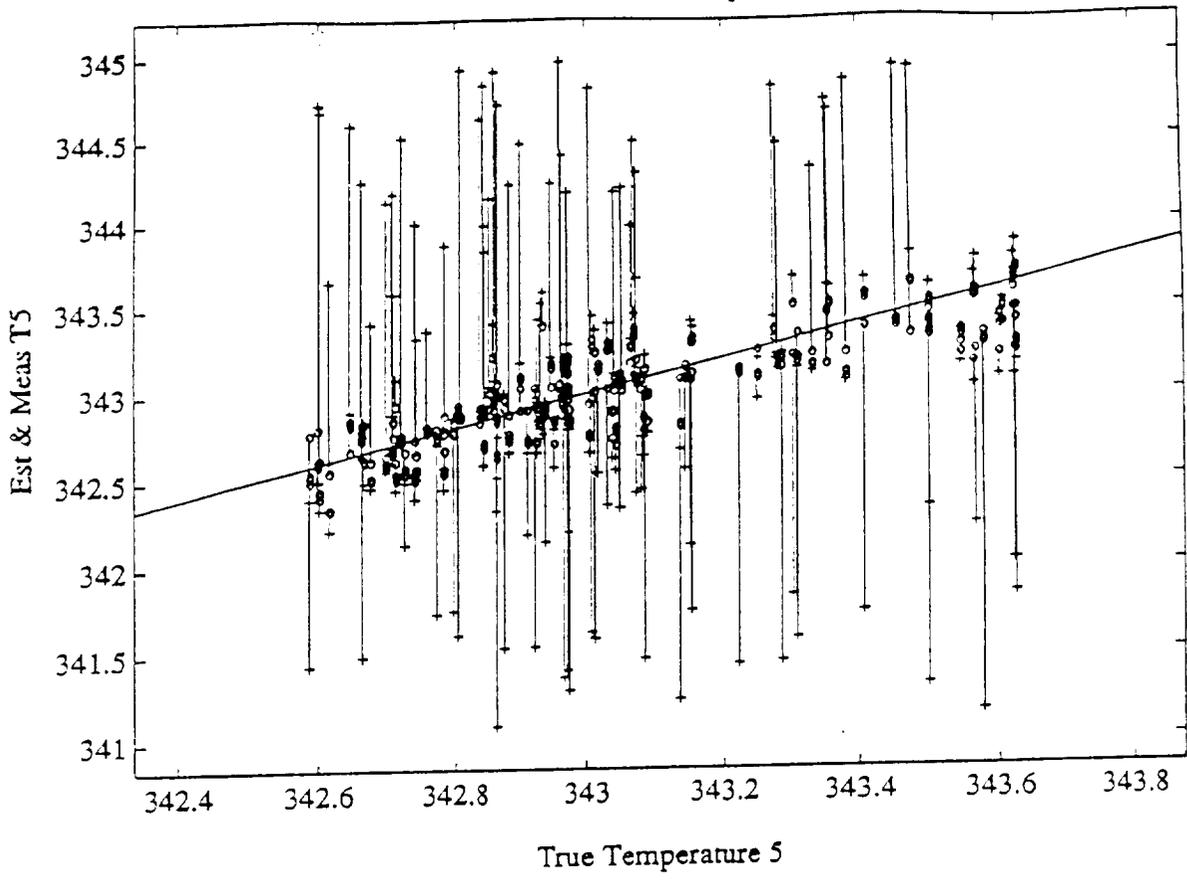
Rectification for Temperature 3



Rectification for Temperature 4



Rectification for Temperature 5



Summary & Conclusions:

Diagnosis can be approached by:

- Multi-class training
- Single-class training

Multi-class diagnosis yields relative fault probabilities

Radial basis function networks preferred approach to multi-class diagnosis

Single-class training involves extraction of statistical distribution model

Can be approach using radial basis functions

Useful for:

- Decomposing multi-class problems
- Fault detection
- Rectification of sensor data