ANFIS: Adaptive Neuro-Fuzzy Inference System

Outline

Soft computing

Fuzzy logic and fuzzy inference systems

Neural networks

Neuro-fuzzy integration: ANFIS

- ANFIS: Adaptive Neuro-Fuzzy Inference Systems
- Learning methods for parameter ID

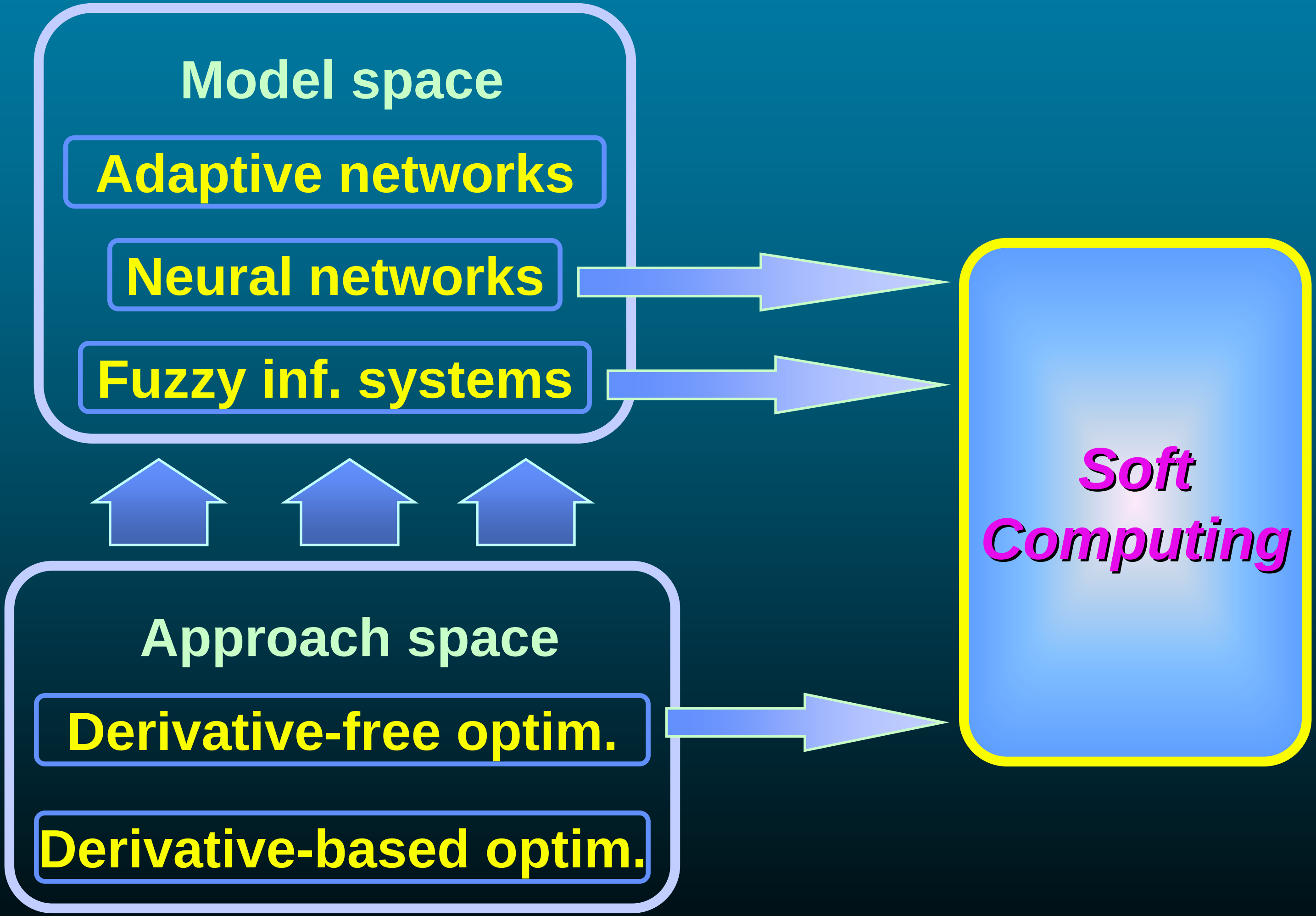
Input selection for ANFIS modeling

- Heuristic and exhaustive searches
- Performance index

Application examples

- Hair dryer modeling
- Box-Jenkins furnace data

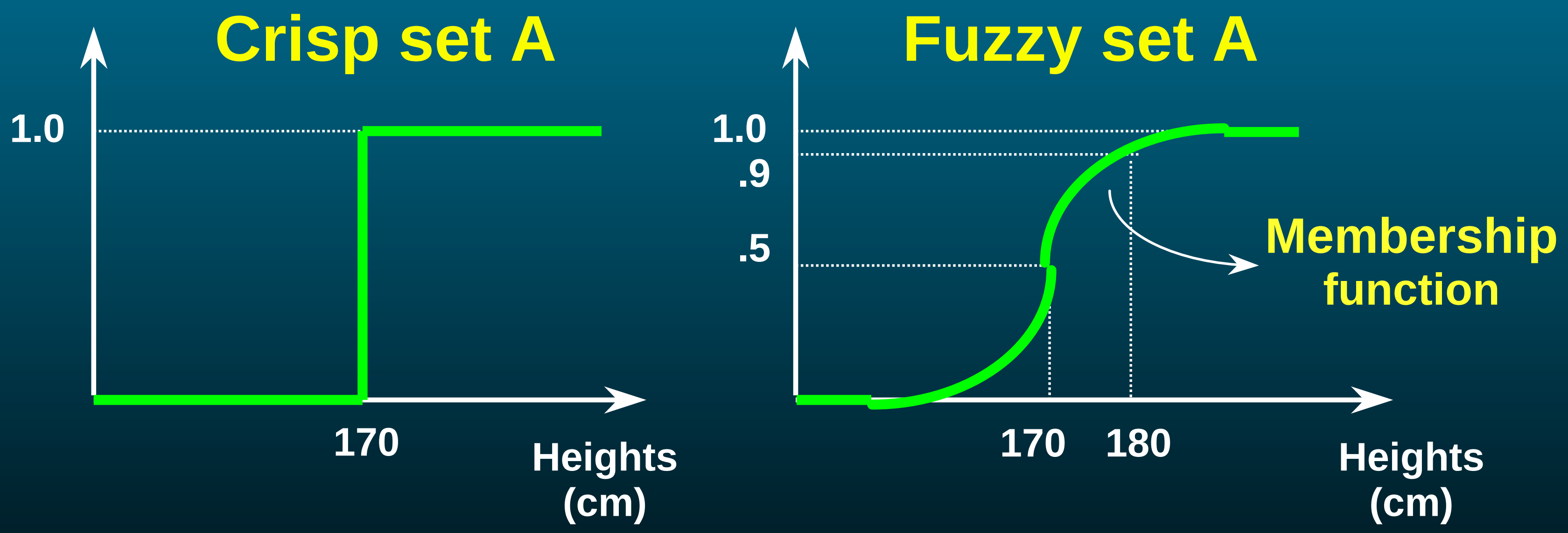
Neuro-Fuzzy and Soft Computing



Fuzzy Sets

Sets with fuzzy boundaries

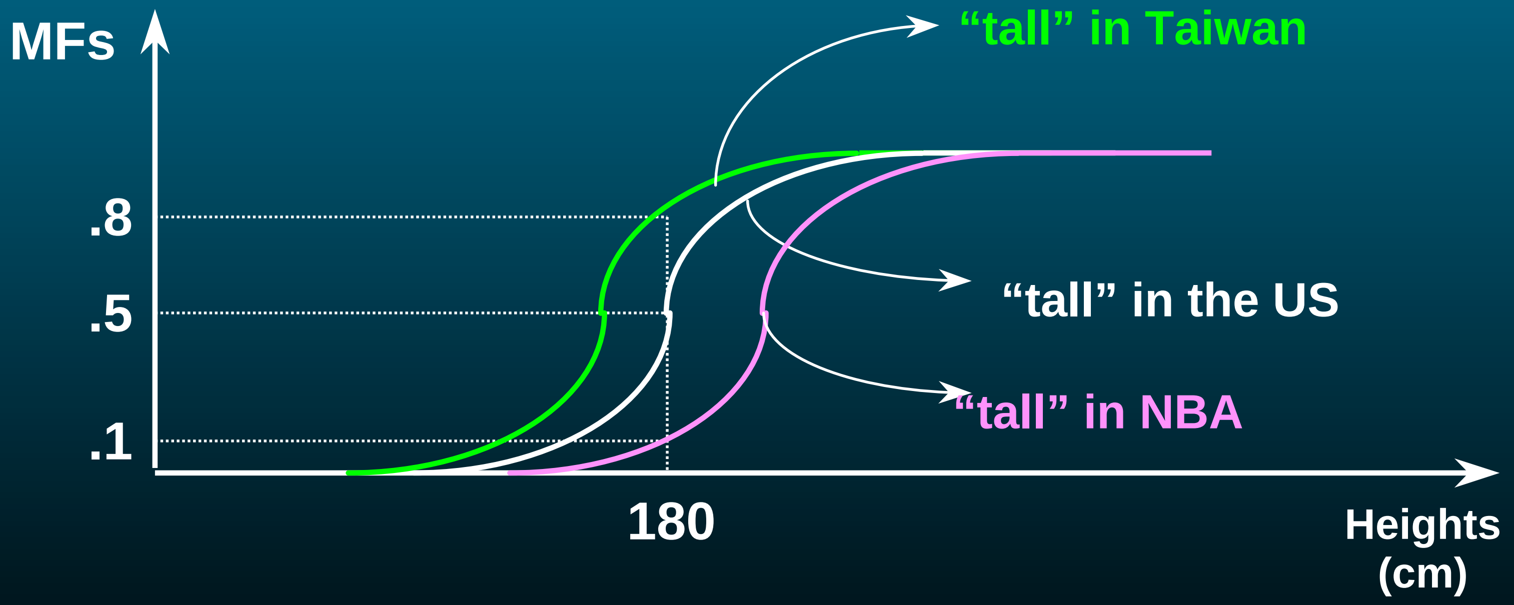
A = Set of tall people



Membership Functions (MFs)

About MFs

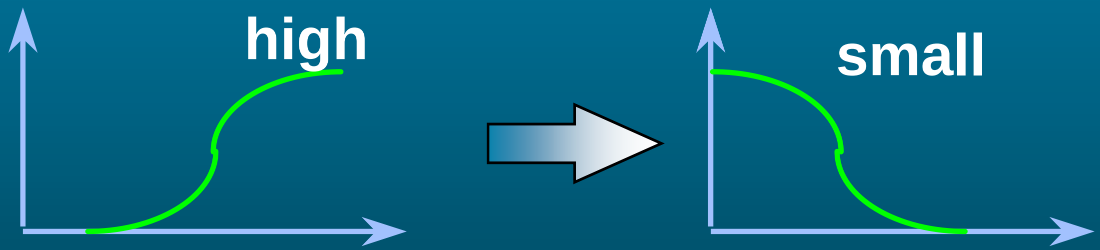
- Subjective measures
- Not probability functions



Fuzzy If-Then Rules

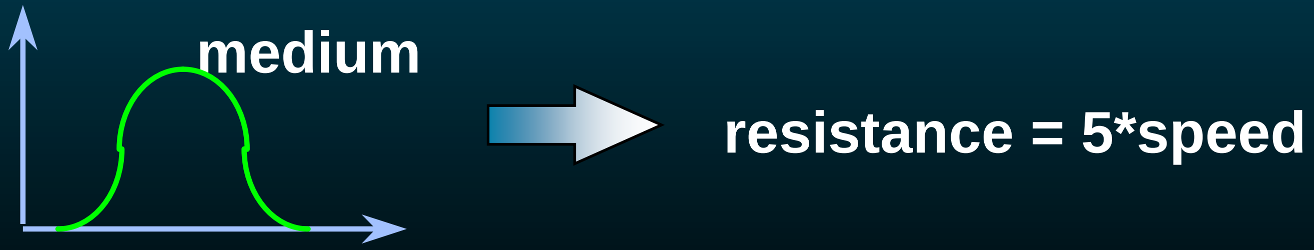
- **Mamdani style**

  If pressure is high then volume is small



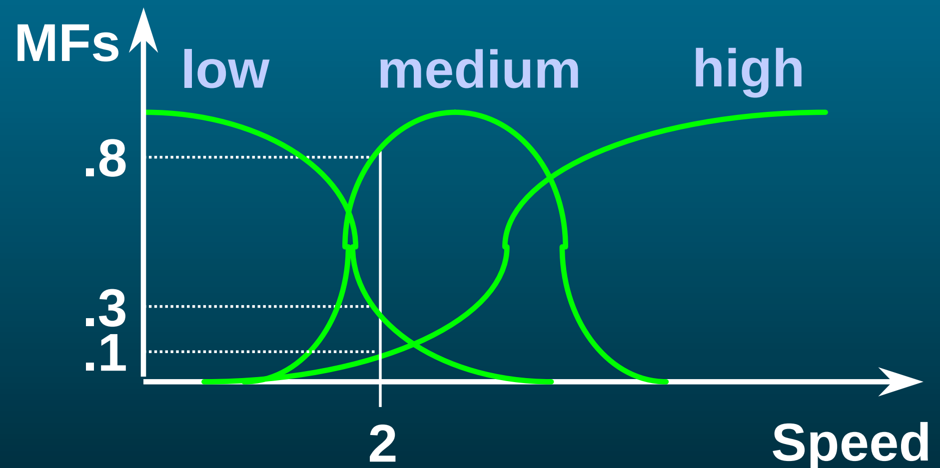
- **Sugeno style**

  If speed is medium then resistance = 5*speed



Fuzzy Inference System (FIS)

If speed is low then resistance = 2
If speed is medium then resistance = 4*speed
If speed is high then resistance = 8*speed



Rule 1: w1 = .3; r1 = 2
Rule 2: w2 = .8; r2 = 4*2
Rule 3: w3 = .1; r3 = 8*2

Resistance = $\Sigma(wi*ri) / \Sigma wi$
= 7.12

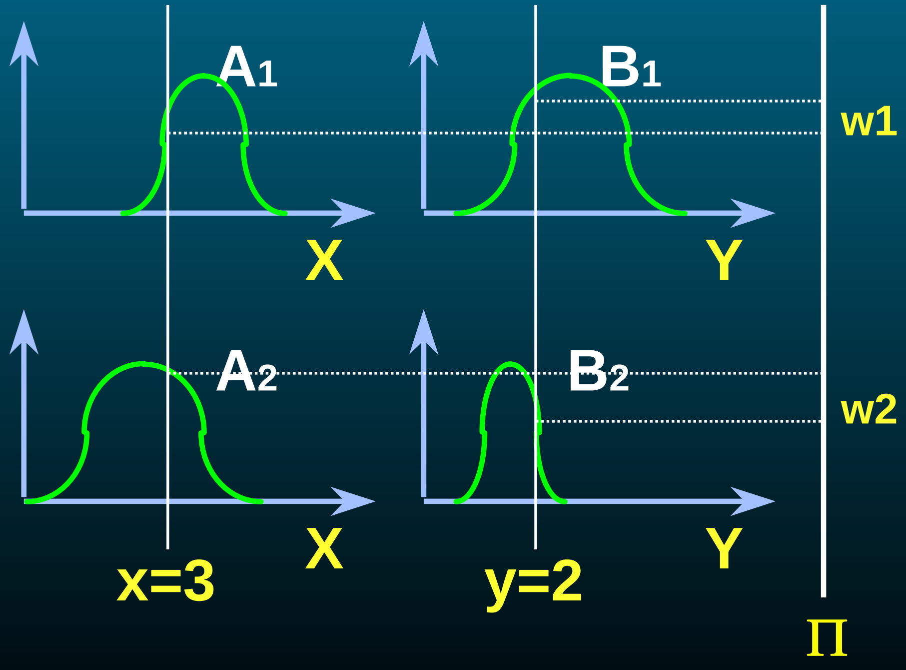
First-Order Sugeno FIS

- **Rule base**

  If X is A_1 and Y is B_1 then $Z = p_1*x + q_1*y + r_1$

  If X is A_2 and Y is B_2 then $Z = p_2*x + q_2*y + r_2$

- **Fuzzy reasoning**



$z_1 = p_1*x+q_1*y+r_1$

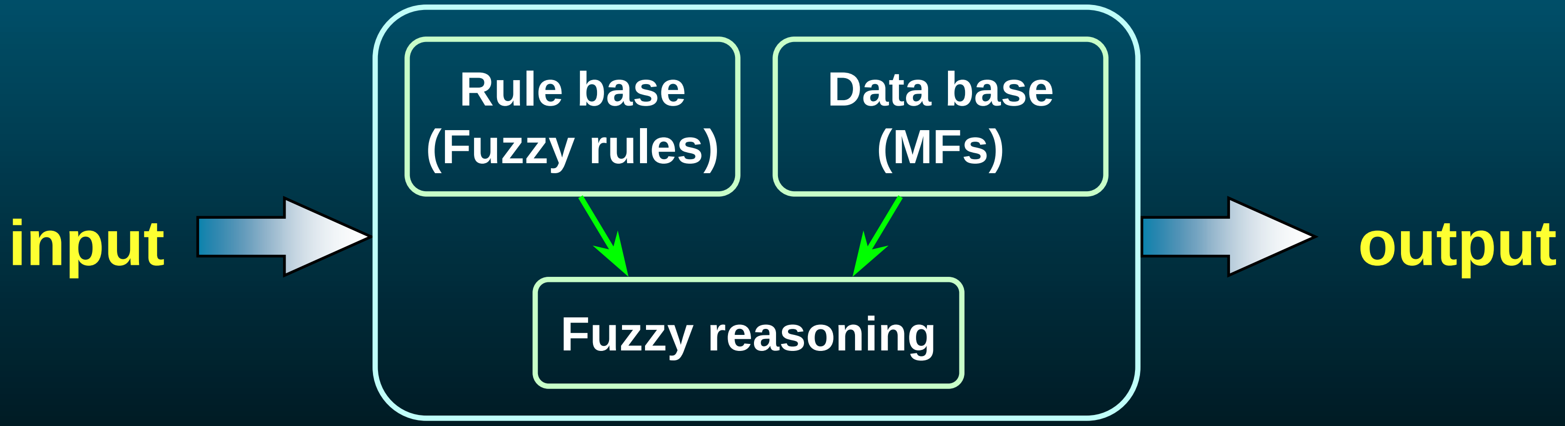
$z_2 = p_2*x+q_2*y+r_2$

$$z = \frac{w_1*z_1+w_2*z_2}{w_1+w_2}$$

Fuzzy Inference Systems (FIS)

Also known as

- Fuzzy models
- Fuzzy associate memories (FAM)
- Fuzzy controllers



Neural Networks

Supervised Learning

- Multilayer perceptrons
- Radial basis function networks
- Modular neural networks
- LVQ (learning vector quantization)

Unsupervised Learning

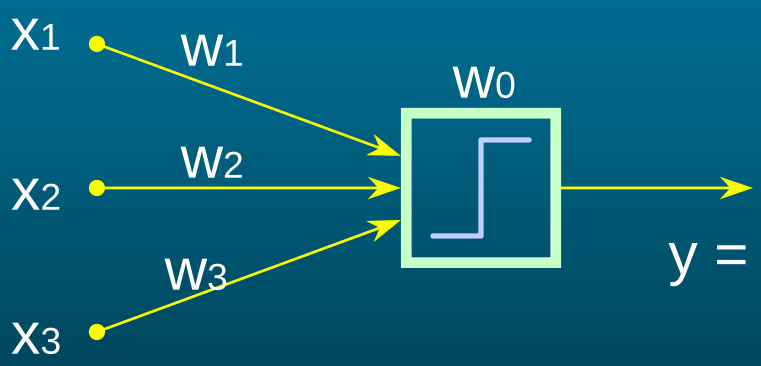
- Competitive learning networks
- Kohonen self-organizing networks
- ART (adaptive resonant theory)

Others

- Hopfield networks

Single-Layer Perceptrons

Network architecture



$y = \text{signum}(\Sigma w_i\, x_i + w_0)$

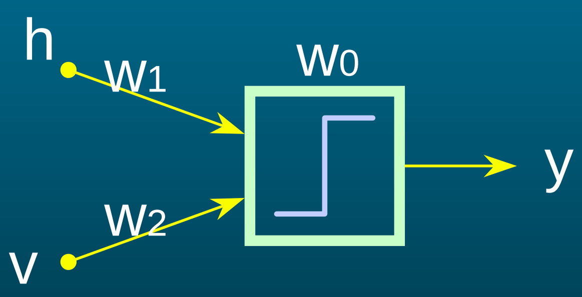
Learning rule

$\Delta w_i = \kappa\, t\, x_i$

Single-Layer Perceptrons

Example: Gender classification

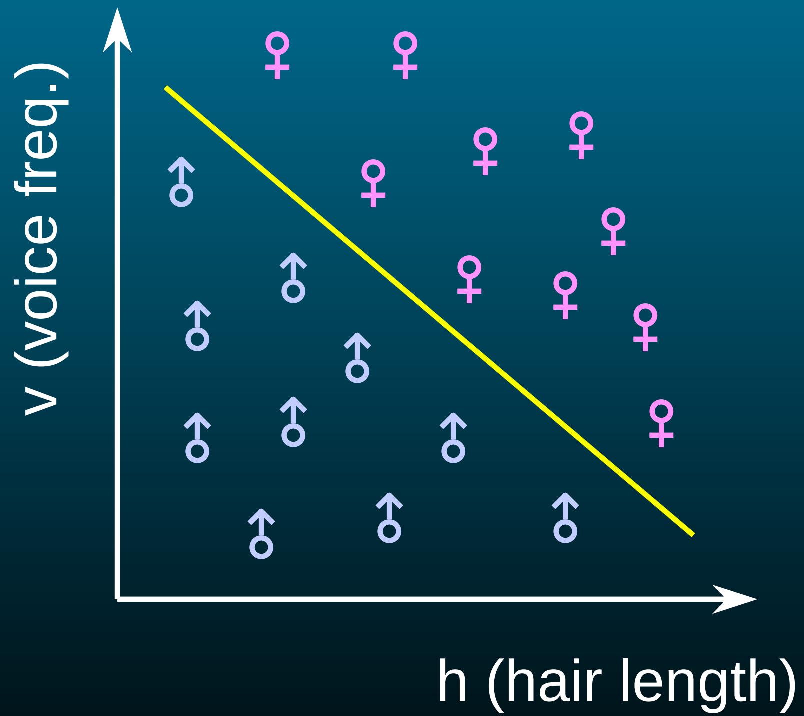
Network Arch.



$$y = \text{signum}(hw_1 + vw_2 + w_0)$$

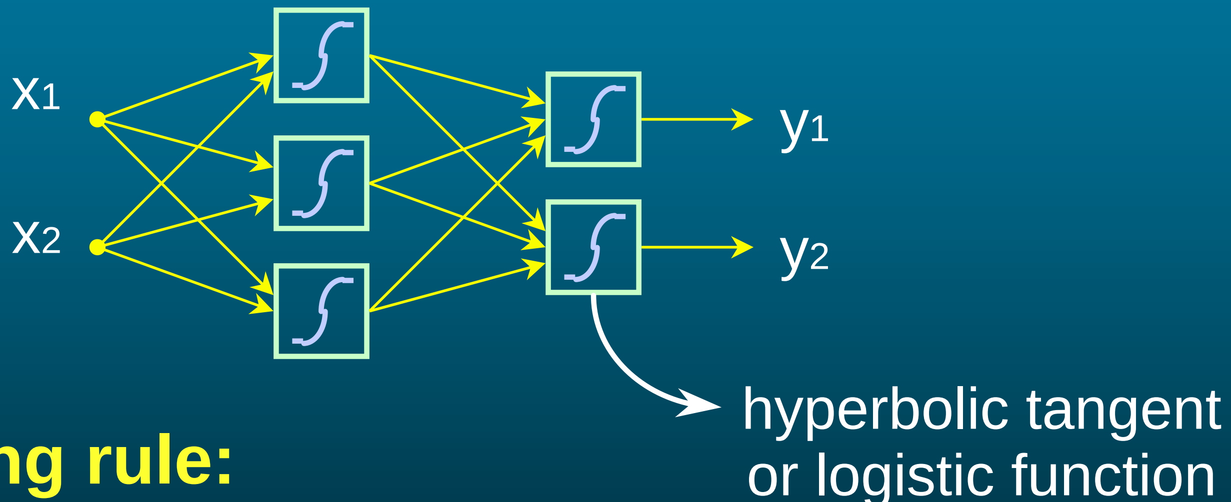
$$= \begin{cases} -1 & \text{if female} \\ 1 & \text{if male} \end{cases}$$

Training data



Multilayer Perceptrons (MLPs)

Network architecture



Learning rule:

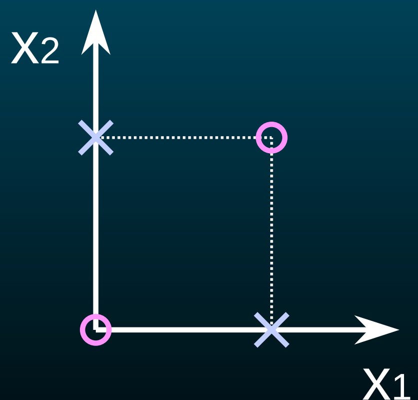
- Steepest descent (Backprop)
- Conjugate gradient method
- All optim. methods using first derivative
- Derivative-free optim.

Multilayer Perceptrons (MLPs)

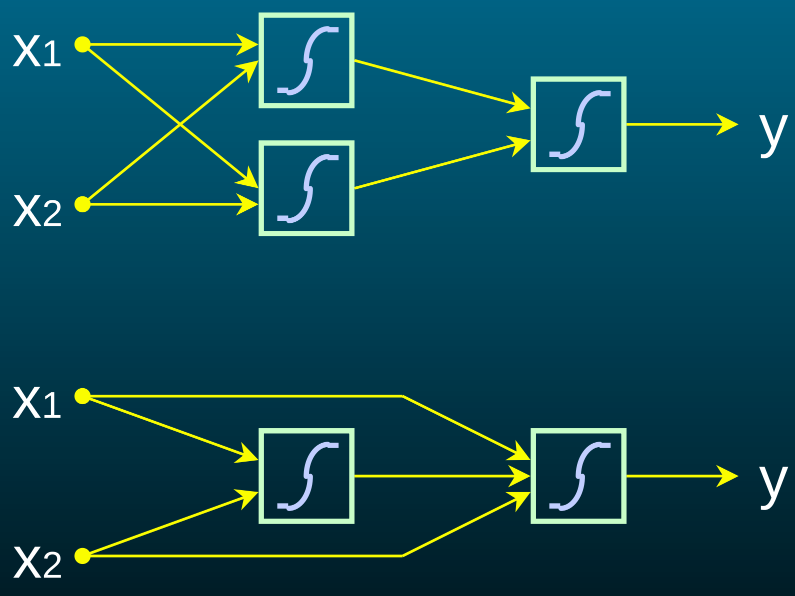
Example: XOR problem

Training data

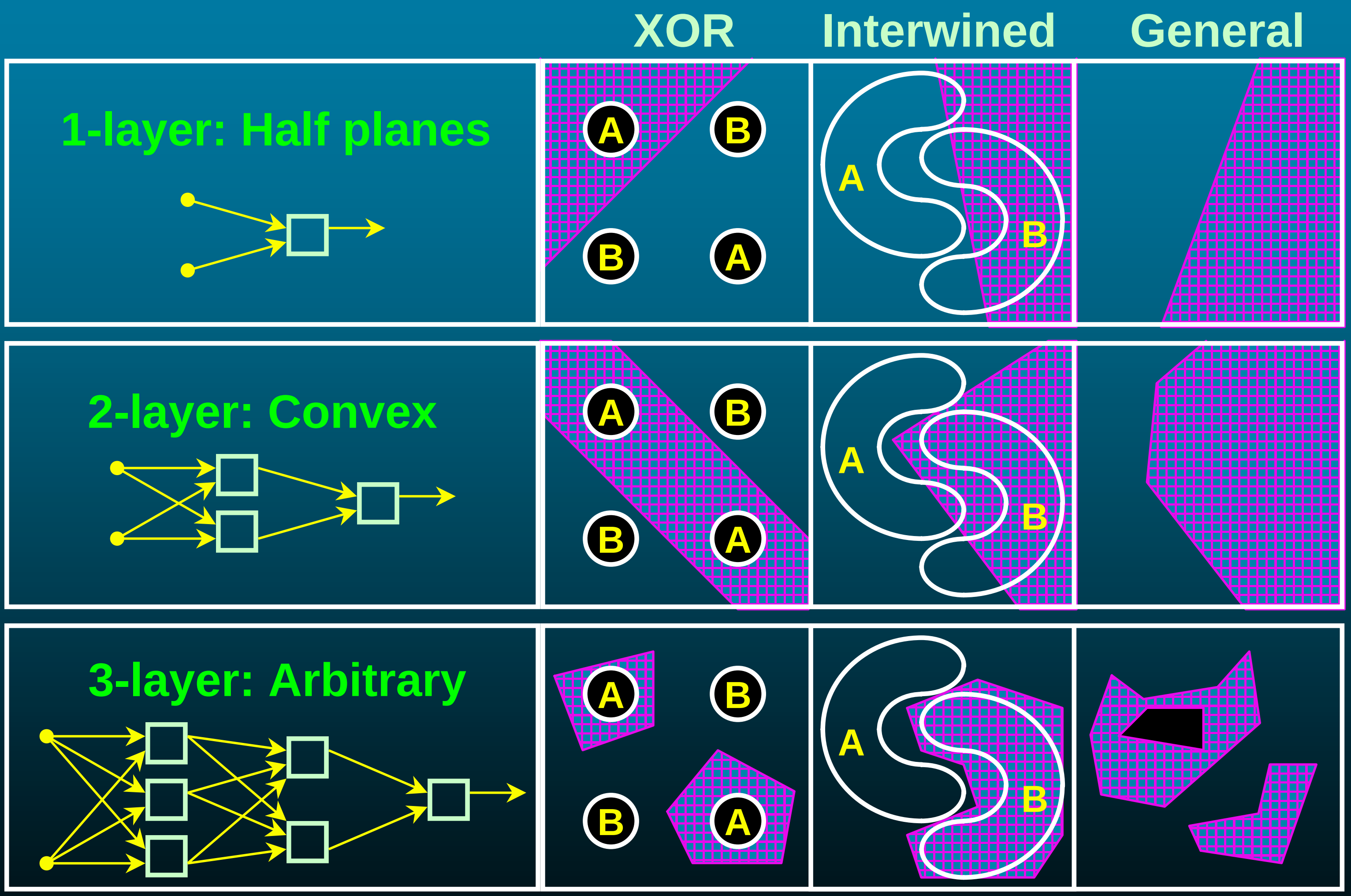
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0



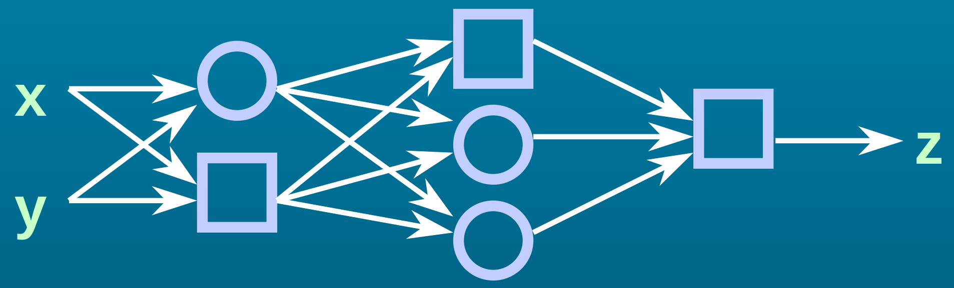
Network Arch.



MLP Decision Boundaries



Adaptive Networks



Architecture:

- Feedforward networks with diff. node functions
- Squares: nodes with parameters
- Circles: nodes without parameters

Goal:

- To achieve an I/O mapping specified by training data

Basic training method:

- Backpropagation or steepest descent

Derivative-Based Optimization

Based on first derivatives:

- Steepest descent
- Conjugate gradient method
- Gauss-Newton method
- Levenberg-Marquardt method
- And many others

Based on second derivatives:

- Newton method
- And many others

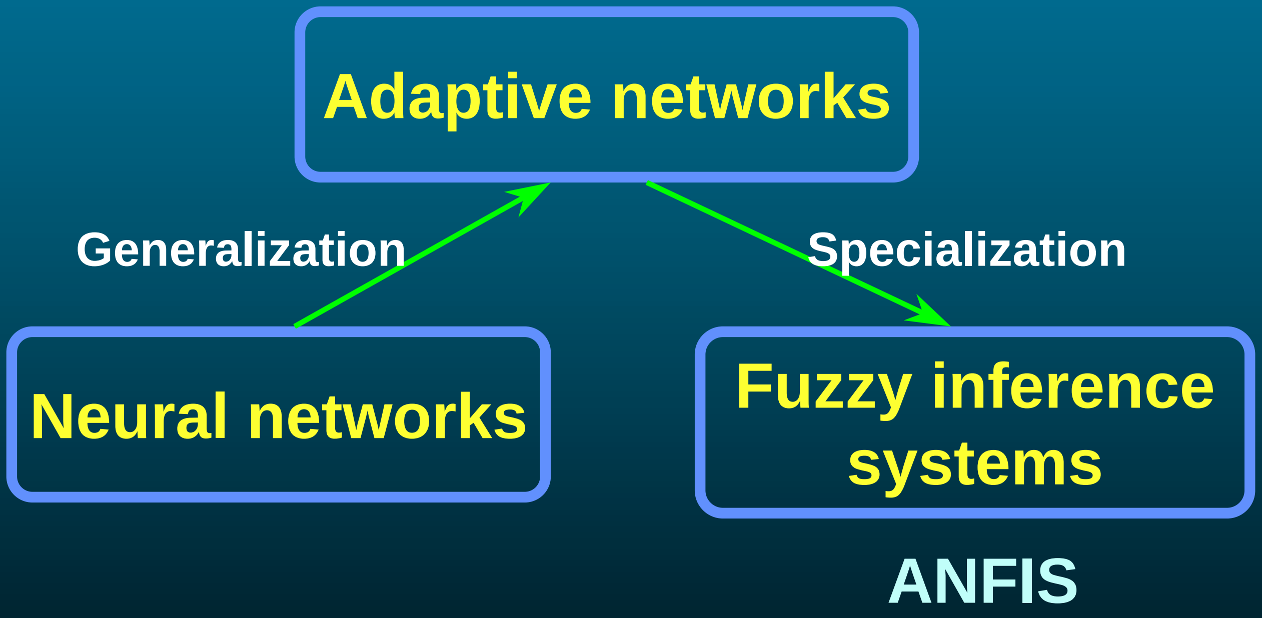
Fuzzy Modeling

- Given desired i/o pairs (training data set) of the form (x1, ..., xn; y), construct a FIS to match the i/o pairs

- Two steps in fuzzy modeling
  structure identification --- input selection, MF numbers
  parameter identification --- optimal parameters

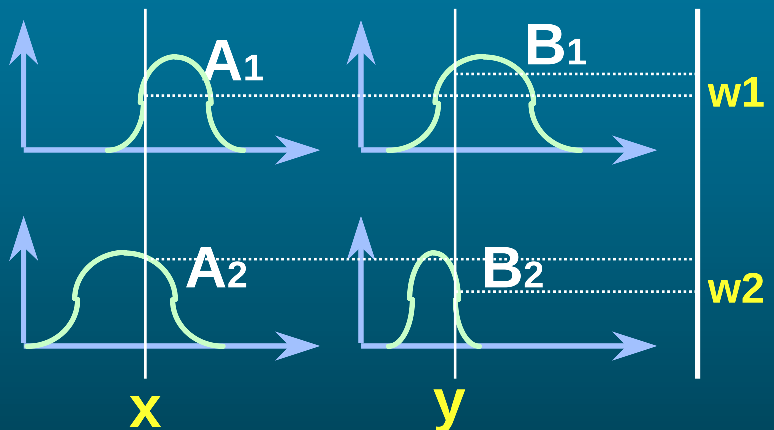
Neuro-Fuzzy Modeling

Basic approach of ANFIS



ANFIS

- Fuzzy reasoning

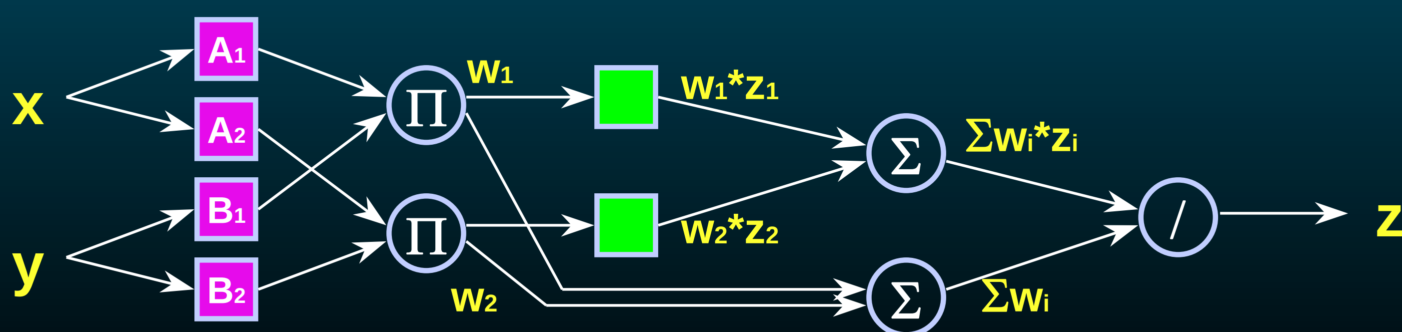


$z_1 = p_1{*}x+q_1{*}y+r_1$

$z_2 = p_2{*}x+q_2{*}y+r_2$

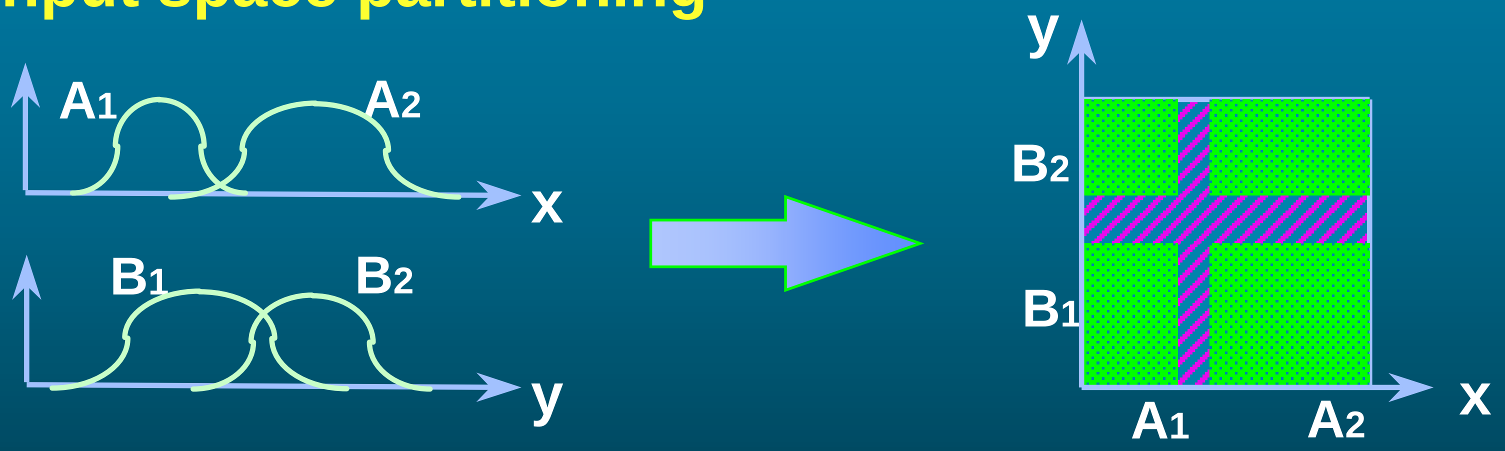
$$z = \frac{w_1{*}z_1+w_2{*}z_2}{w_1+w_2}$$

- ANFIS (Adaptive Neuro-Fuzzy Inference System)

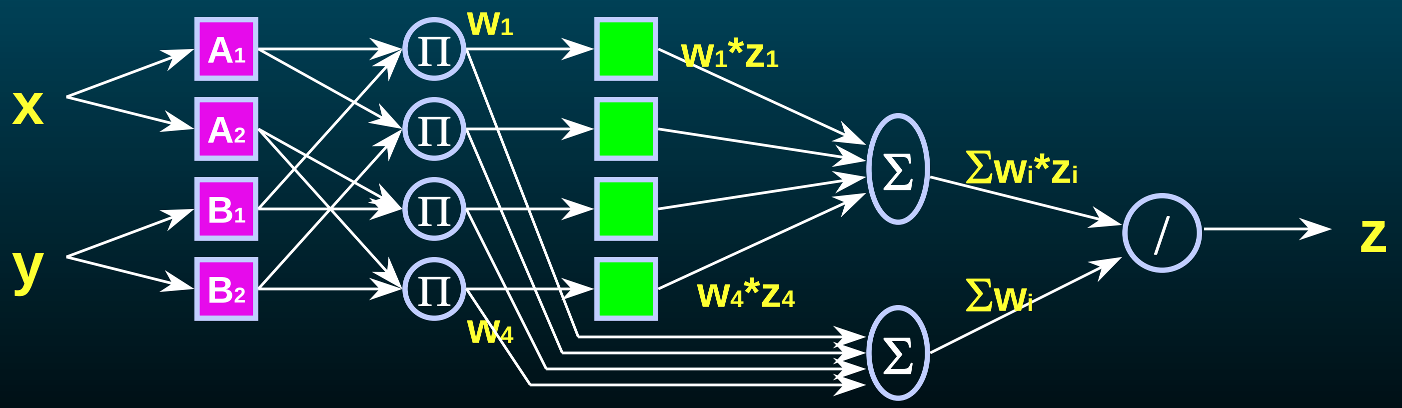


Four-Rule ANFIS

- Input space partitioning

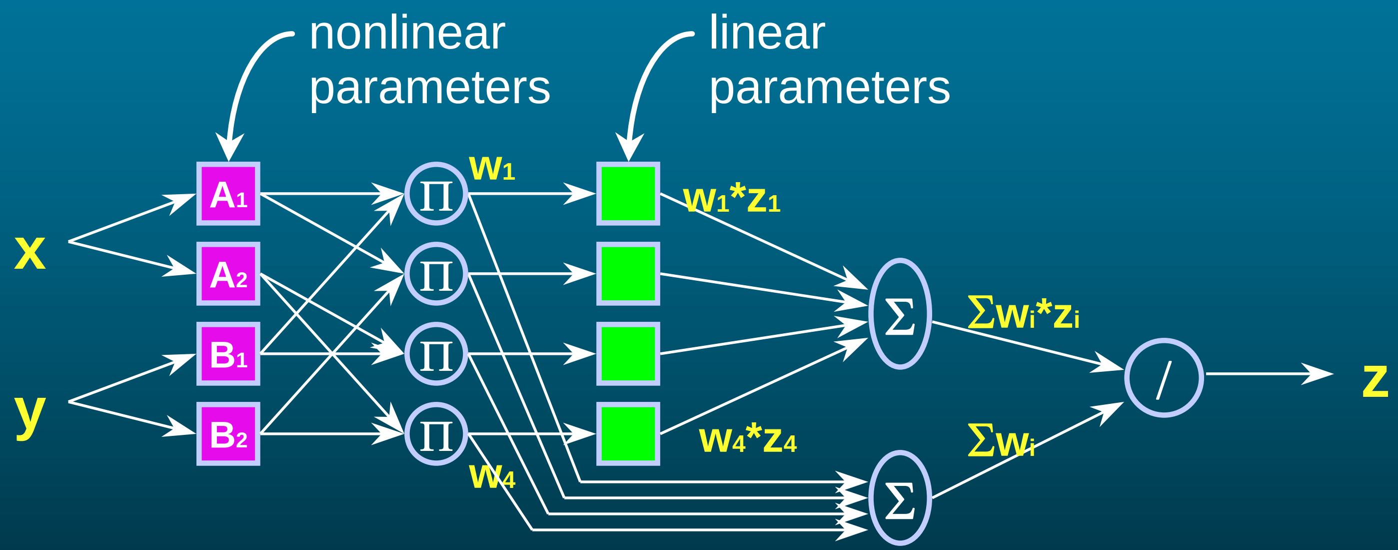


- ANFIS (Adaptive Neuro-Fuzzy Inference System)



ANFIS: Parameter ID

Hybrid training method



	forward pass	backward pass
MF param. (nonlinear)	fixed	steepest descent
Coef. param. (linear)	least-squares	fixed

Parameter ID: Gauss-Newton Method

Synonyms:

- linearization method
- extended Kalman filter method

Concept:

general nonlinear model: $y = f(x, \theta)$

linearization at $\theta = \theta_{now}$:

$$y = f(x, \theta_{now}) + a_1(\theta_1 - \theta_{1,now}) + a_2(\theta_2 - \theta_{2,now}) + \ldots$$

LSE solution:

$$\theta_{next} = \theta_{now} + \eta(A^T A)^{-1} A^T B$$

Param. ID: Levenberg-Marquardt

Formula:

$$\theta_{next} = \theta_{now} + \eta(A^T A + \lambda I)^{-1} A^T B$$

Effects of λ:

- λ small → Gauss-Newton method
- λ big → steepest descent

How to update λ:

greedy policy → make λ small

cautious policy → make λ large

Param. ID: Comparisons

Steepest descent (SD)

- treats all parameters as nonlinear

Hybrid learning (SD+LSE)

- distinguishes between linear and nonlinear

Gauss-Newton (GN)

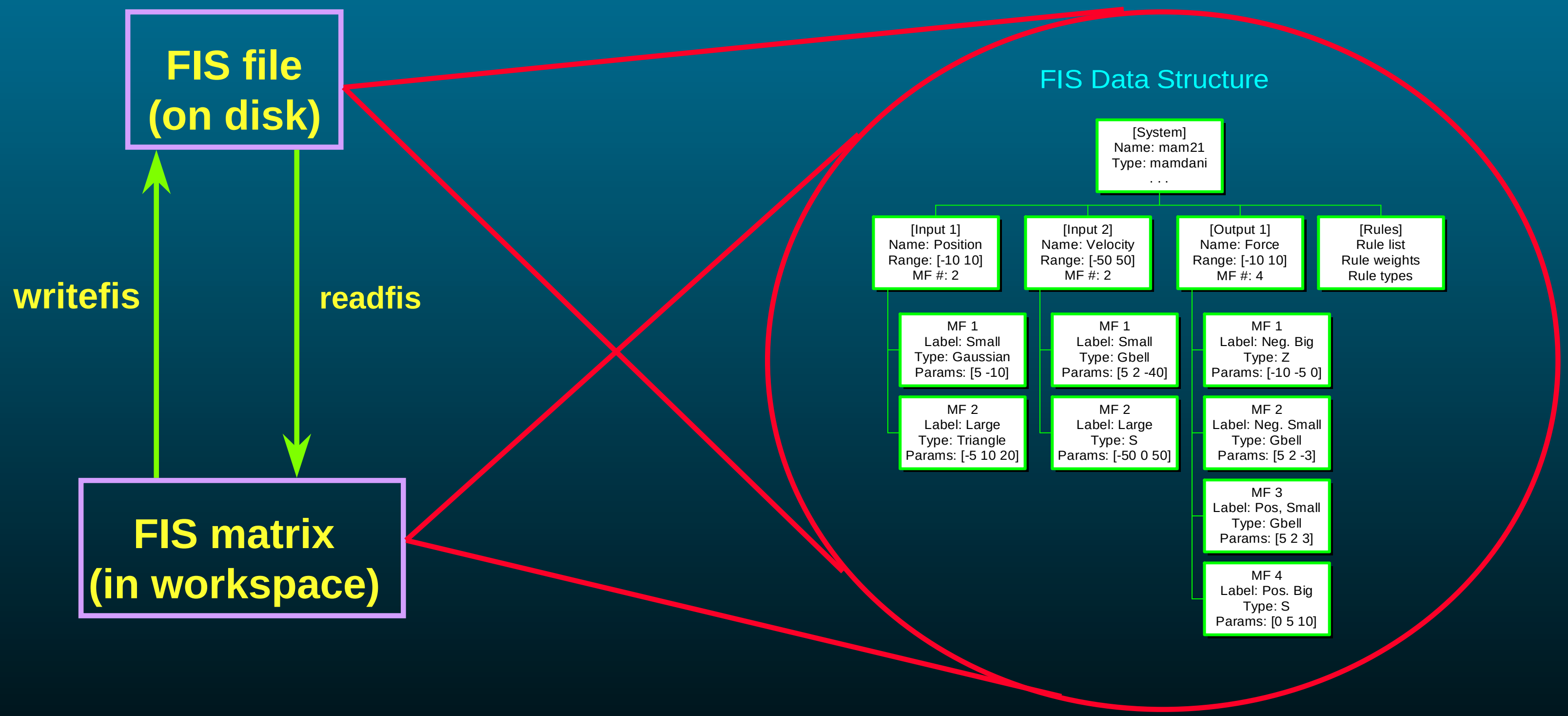
- linearizes and treats all parameters as linear

Levenberg-Marquardt (LM)

- switches smoothly between SD and GN

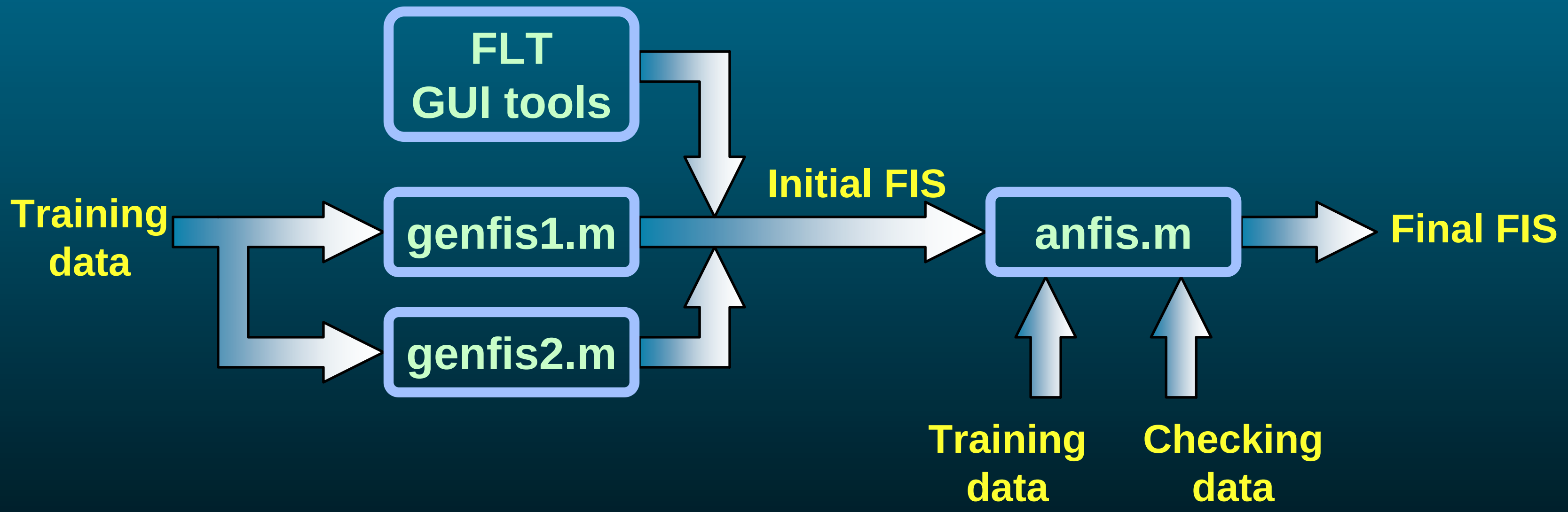
FIS data structure in FLT

FIS file and FIS matrix



From Data Sets to FIS

Flow chart: From data sets to FIS



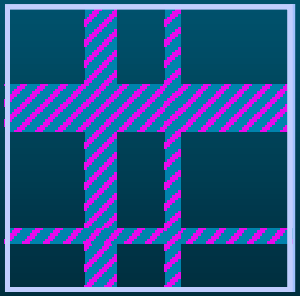
ANFIS: Structure ID

Input selection

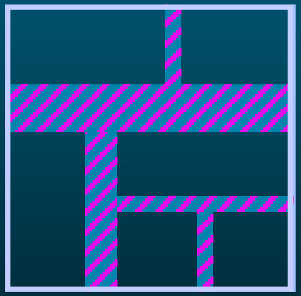
To select relevant input for efficient modeling

Input space partitioning

Grid partitioning

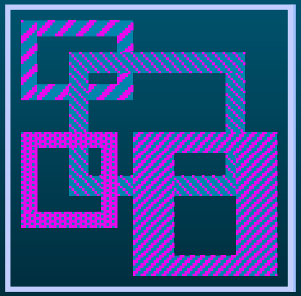


Tree partitioning



- CART method

Scatter partitioning



- C-means clustering
- mountain method

Input Selection

Optimal search: direct exhaustive search

1 input → x1 | x2 | x3 | x4 | x5

2 inputs → x1, x2 | x1, x3 | x1, x4 | x1, x5 | x2, x3 ...

3 inputs → x1, x2, x3 | x1, x2, x4 | x1, x2, x5 | x1, x3, x4 ...

4 inputs → x1, x2, x3, x4 | x1, x2, x3, x5 | x1, x2, x4, x5 ...

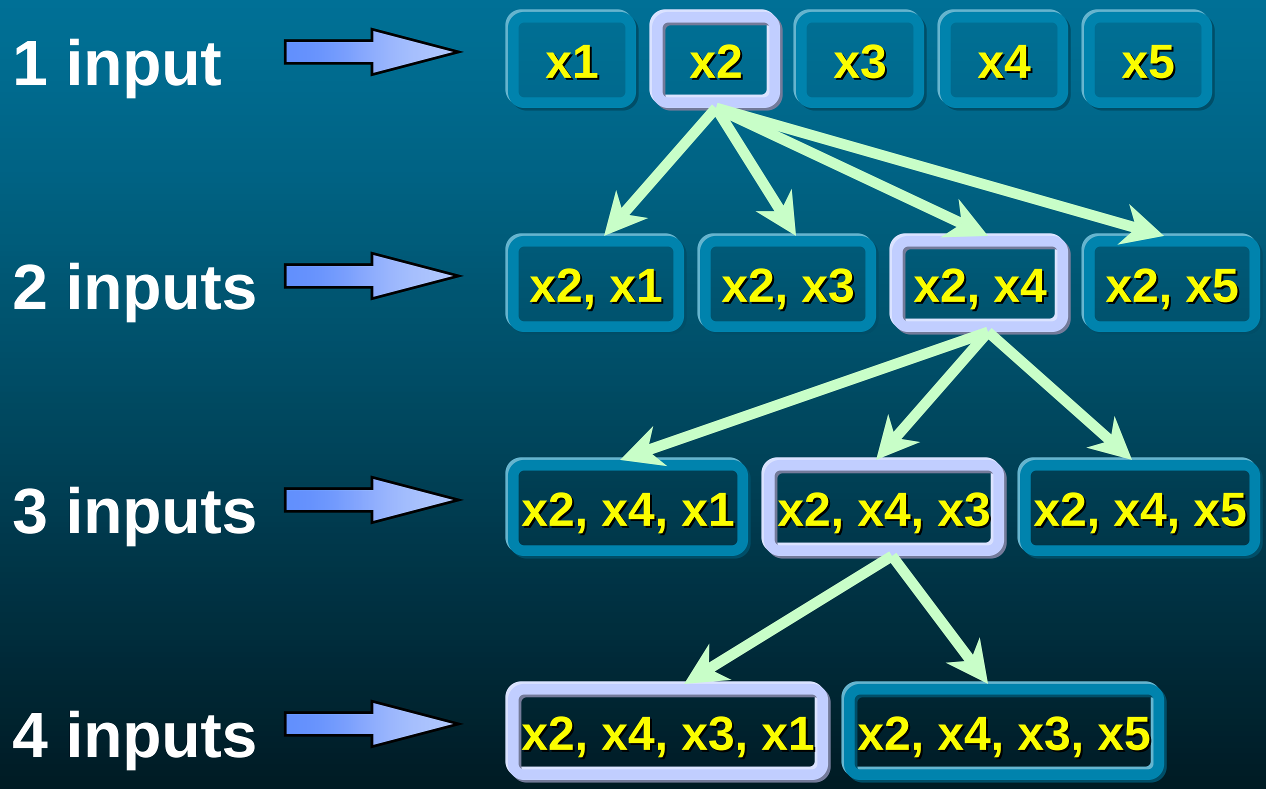
Input Selection

Suboptimal search

- One-pass ranking
- Sequential forward selection
- Generalized sequential forward selection
- Sequential backward selection
- Generalized sequential backward selection
- ‘Add m, remove n’ selection
- Generalized ‘add m, remove n’ selection

Input Selection

Sequential forward selection



Performance Index

How to effectively contruct a model and evaluate it properly?

- **Model construction:**

  ANFIS with one-epoch training

  (Use least-squares method only once)

- **Model evaluation:**
  - Training RMSE (root-mean-squared error)
  - Bipartite regularity criterion
  - Leave-one-out regularity criterion

Performance Index

Bipartite regularity criterion:

$$RC = \left\{ \frac{1}{|A|} \sum_i \left[t_i^A - F_B(\vec{x}_i^A) \right]^2 + \frac{1}{|B|} \sum_i \left[t_i^B - F_A(\vec{x}_i^B) \right]^2 \right\} \Big/ 2$$

$$\text{Data Set A} = \left\{ \left(\vec{x}_i^A ; t_i^A \right) \right\}$$

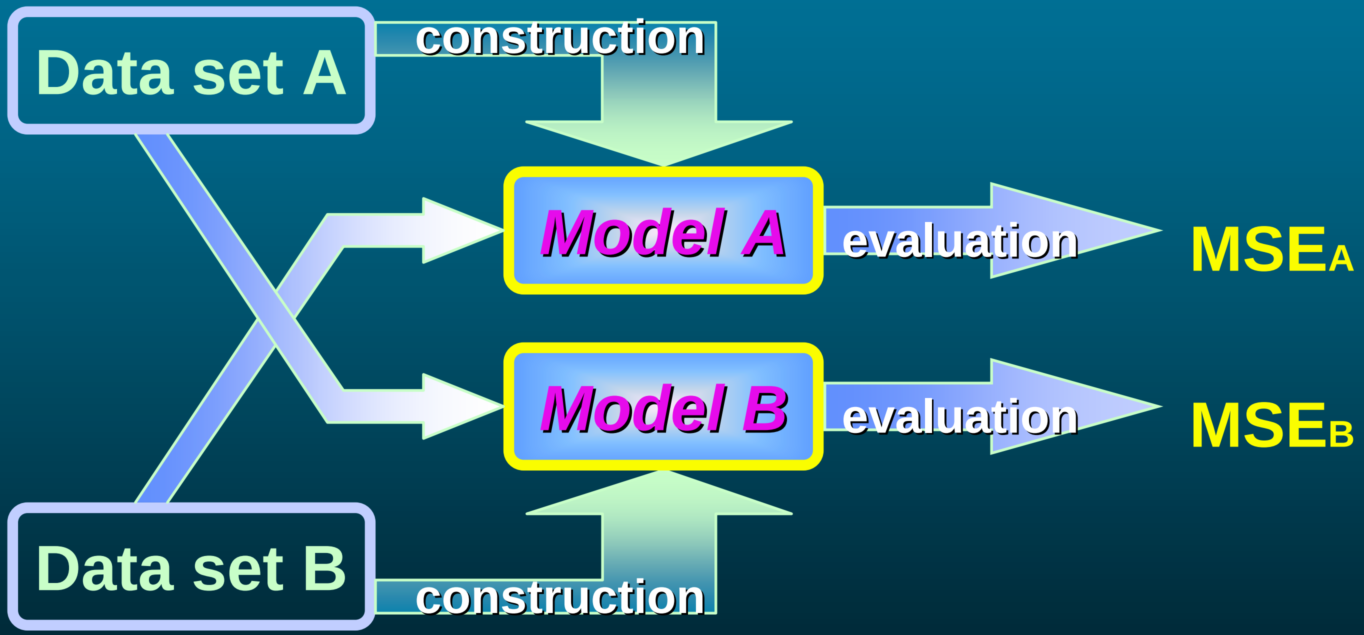
$$\text{Data Set B} = \left\{ \left(\vec{x}_i^B ; t_i^B \right) \right\}$$

$F_A(\bullet)$: Model identified using data set A

$F_B(\bullet)$: Model identified using data set B

Regularity Criterion

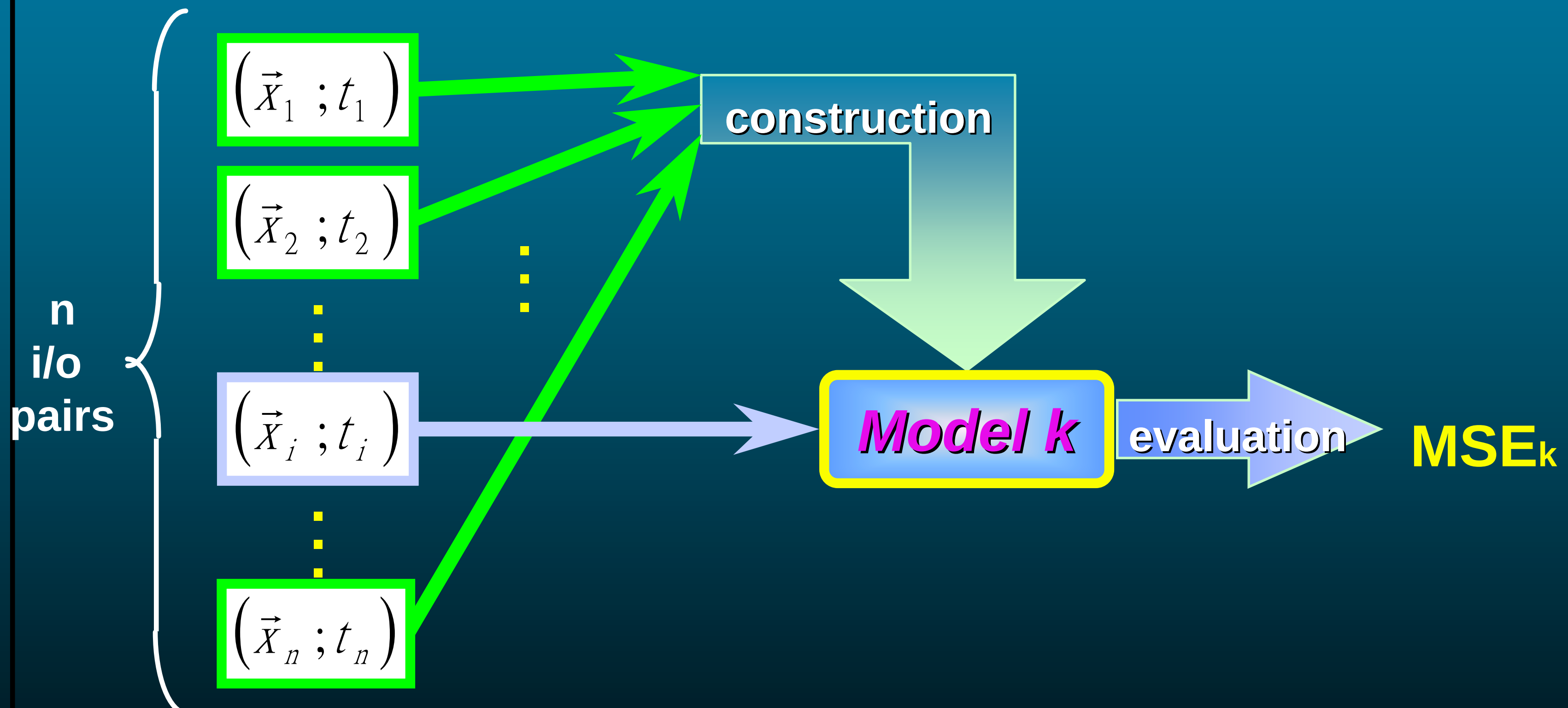
Bipartite RC (used in GMDH)



$$RC = (MSE_A + MSE_B)/2$$

Regularity Criterion

Leave-one-out RC



$$RC = (\Sigma_k MSE_k)/n$$

Regularity Criterion

Computational complexity of ANFIS:

- Bipartite RC: $T_{BRC} = n*(t_L+t_E)$
- Leave-one-out RC: $T_{LRC} = 2n*t_L + n*t_E$

  (t_L: time for one-entry sequential LSE update)

  (t_E: time for one-entry model computation)
- If $t_L = 4t_E$, then $T_{LRC}/T_{BRC} = 9/5$ ~= 2.

Input Selection Based on RC

Test example: $y = (1 + x_1^{-2} + x_2^{-1.5})^2$

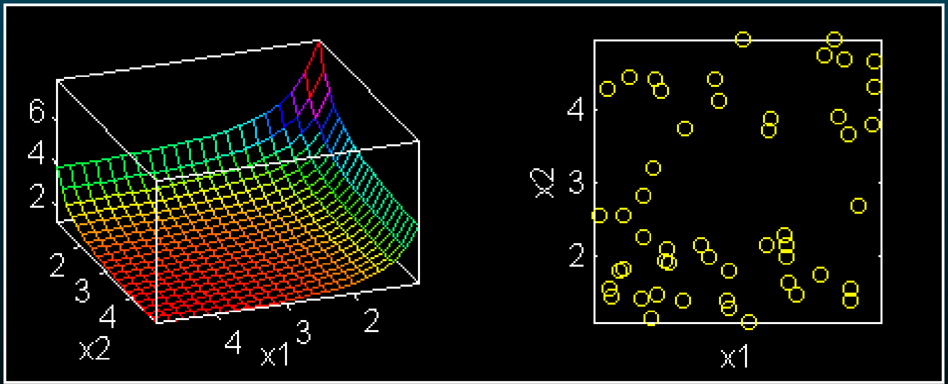
Data source:

- Sugeno and Yasukawa, Tran. on Fuzzy Systems

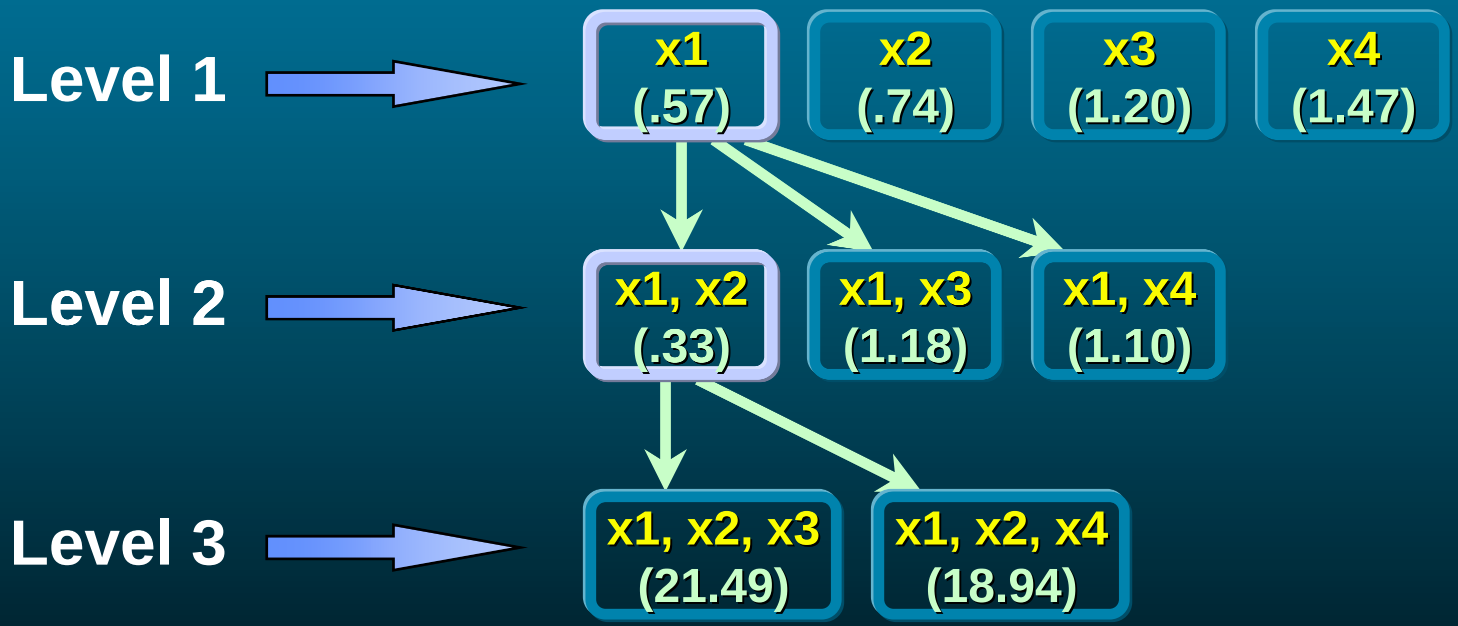
Data size:

- Total: 50, data set A: 25, data set B: 25

Two dummy input variables: x3 and x4



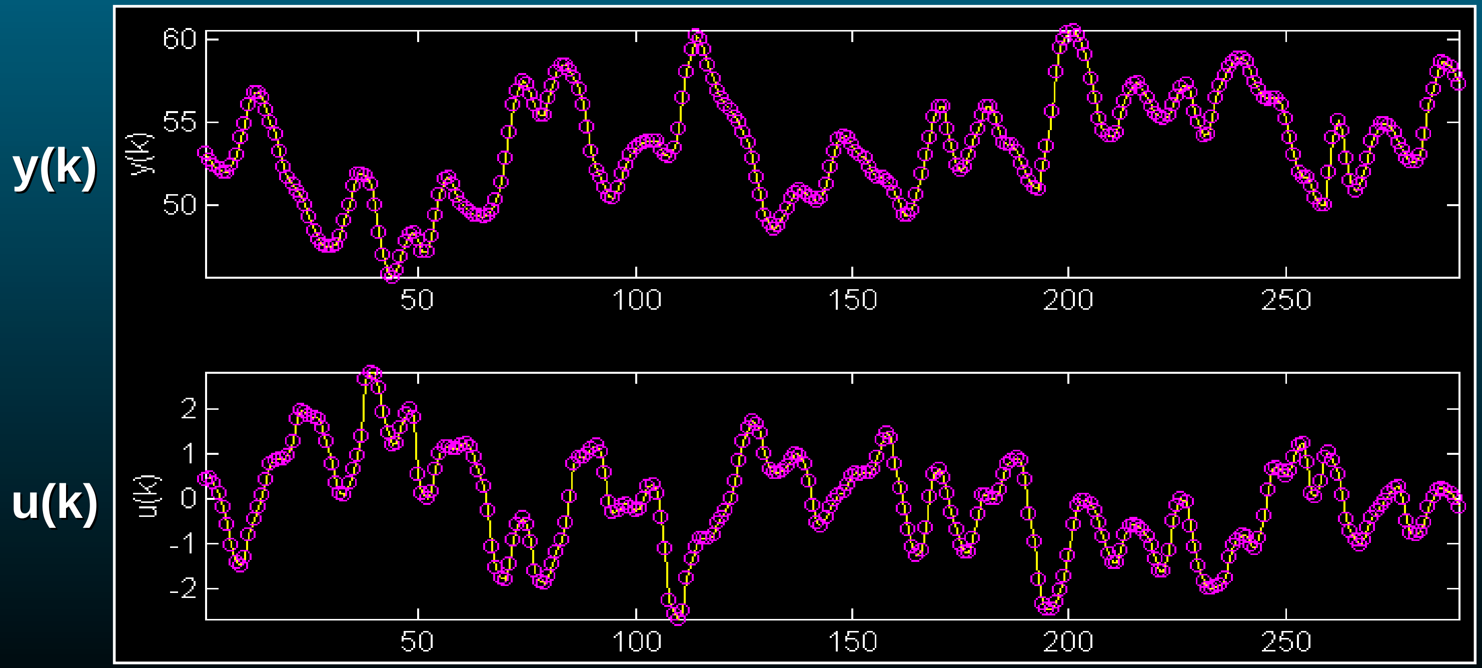
Input Selection Based on RC



Gas Furnace Modeling

Data source: Box-Jenkins gas furnace data

Sampling time: 9 seconds



Gas Furnace Modeling

10 potential inputs for ANFIS:

- Group 1: y(k), y(k-1), y(k-2), y(k-3)
- Group 2: u(k), u(k-1), u(k-2), u(k-3), u(k-4), u(k-5)

1 output: y(k+1)

System model:

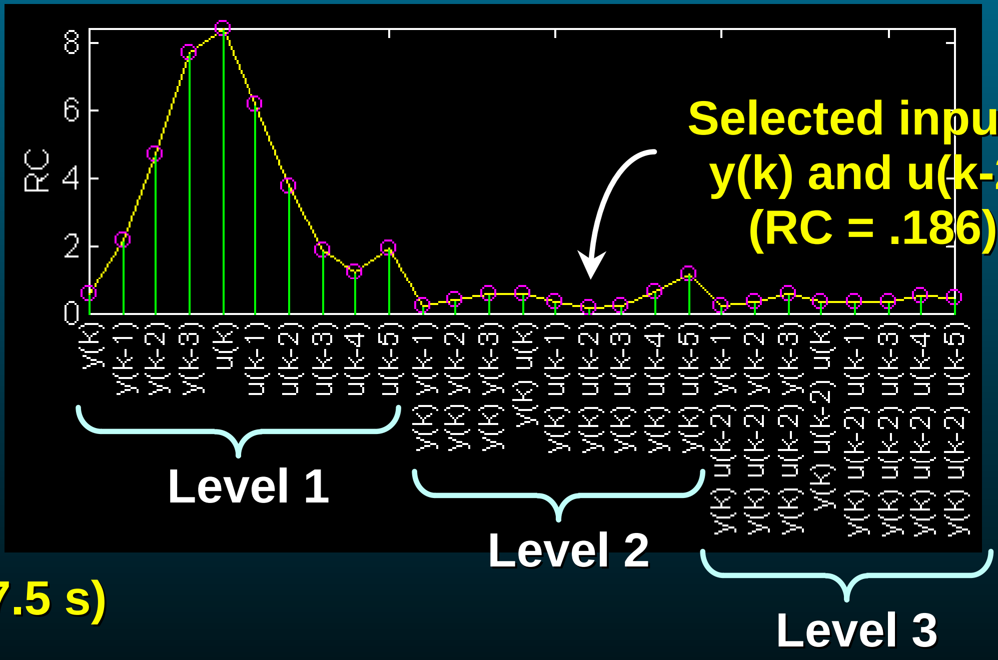
$$y(k+1) = F(y(k), \ldots y(k-3), u(k), \ldots u(k-5))$$

Data size: 296

Gas Furnace Modeling

Sequential forward selection with bipartite RC:

- data set A: the first 148 entries
- data set B: the second 148 entries

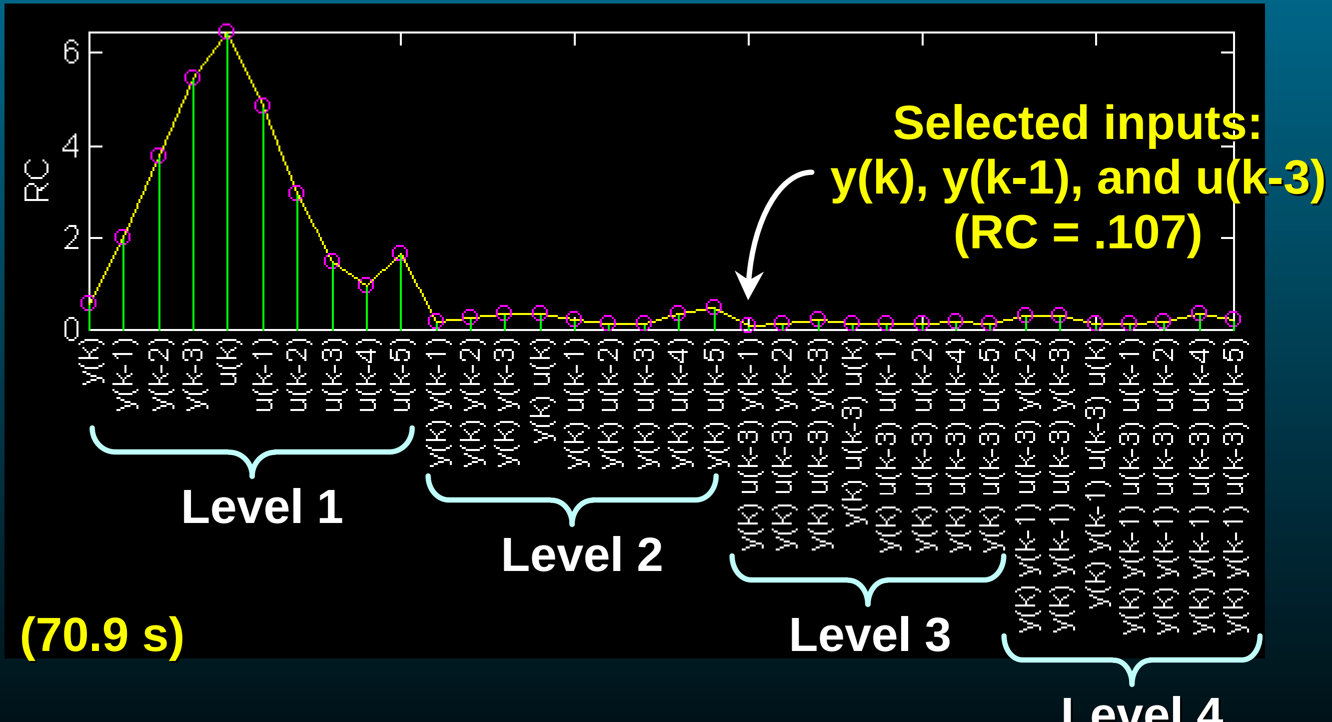


(27.5 s)

Gas Furnace Modeling

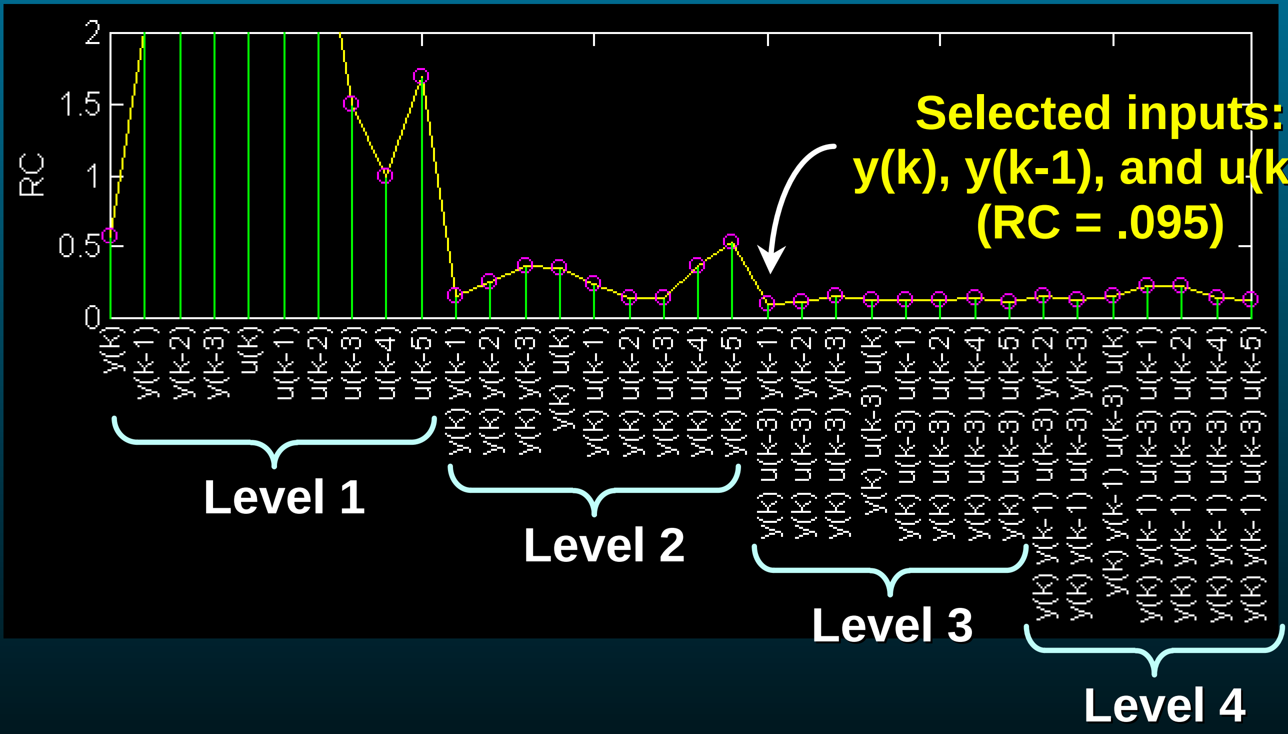
Sequential forward selection with bipartite RC:

- data set A: odd-indexed 148 entries
- data set B: even-indexed 148 entries



Gas Furnace Modeling

Sequential forward selection with leave-one-out RC:

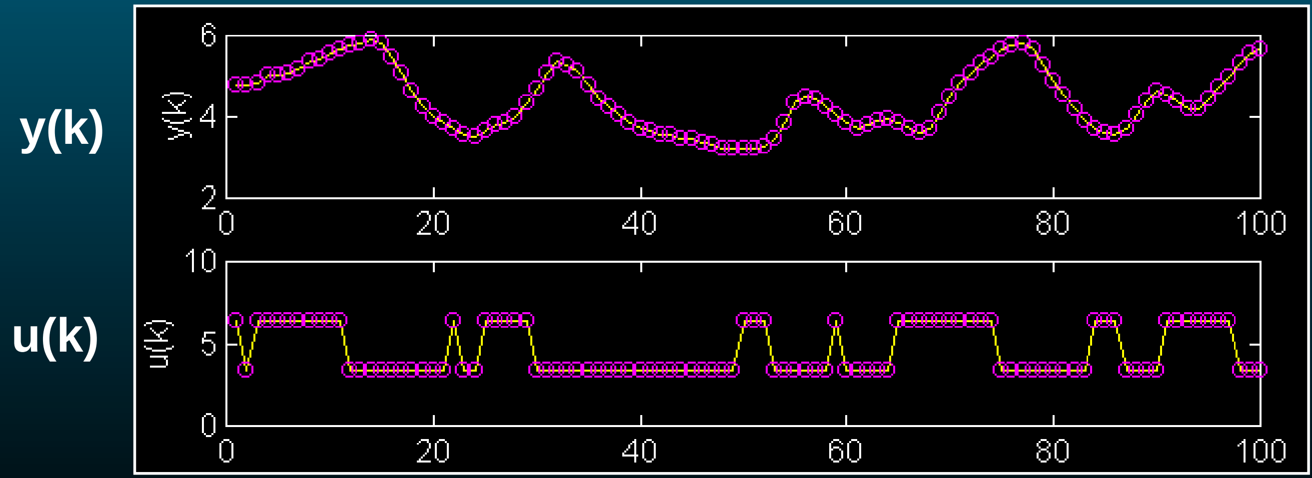


Hair Dryer Modeling

Data source: Chap. 17 of System Identification., Ljung, 1984

u(k): binary random signal shifting between 3.41 and 6.41 V

sampling time: 0.08s



Hair Dryer Modeling

10 potential inputs for ANFIS:

- Group 1: y(k), y(k-1), y(k-2), y(k-3)
- Group 2: u(k), u(k-1), u(k-2), u(k-3), u(k-4), u(k-5)

1 output: y(k+1)

System model:

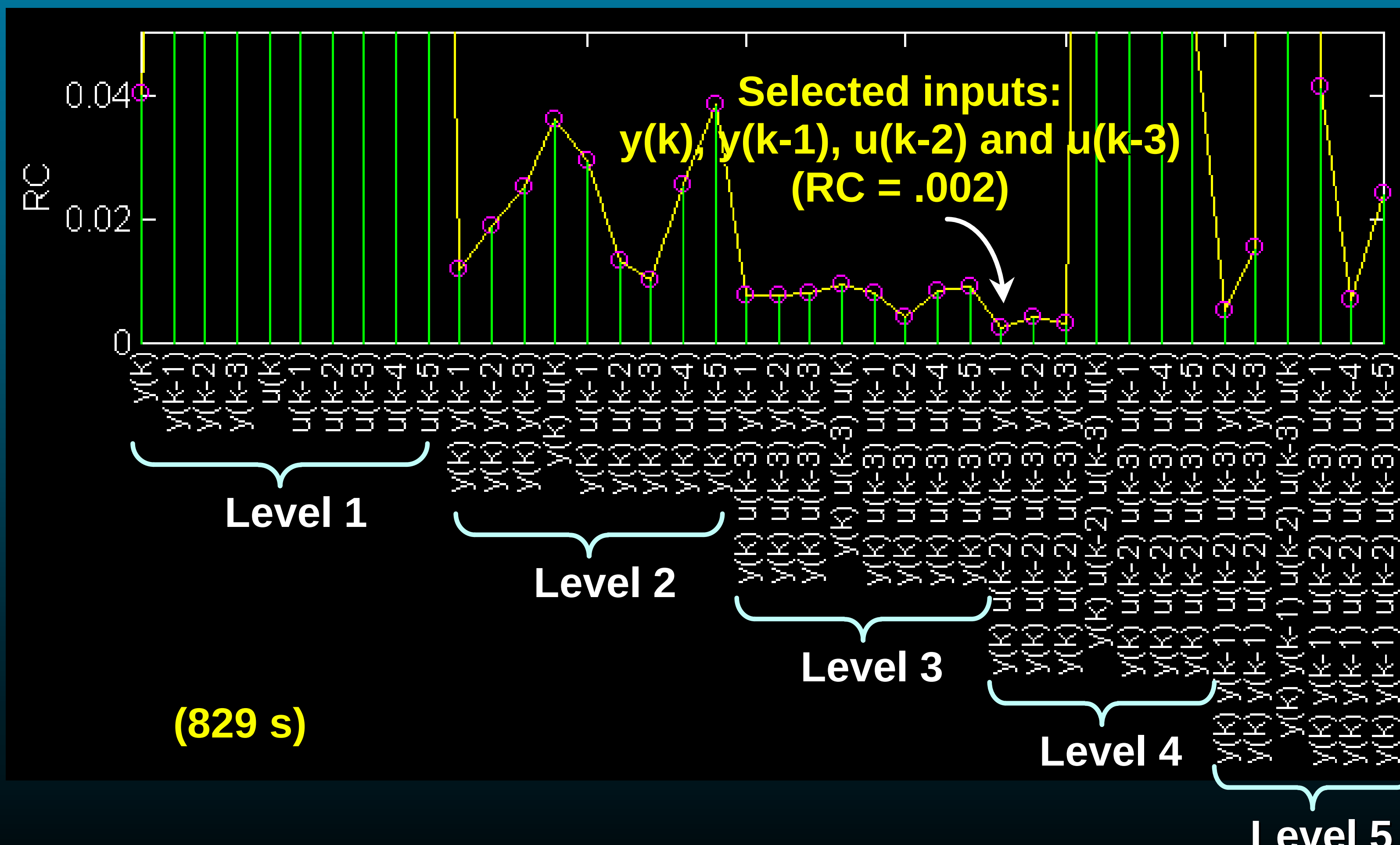
y(k+1) = F(y(k), ... y(k-3), u(k), ... u(k-5))

Data size: 600

- data set A: 300
- data set B: 300

Hair Dryer Modeling

Sequential forward selection with bipartite RC:

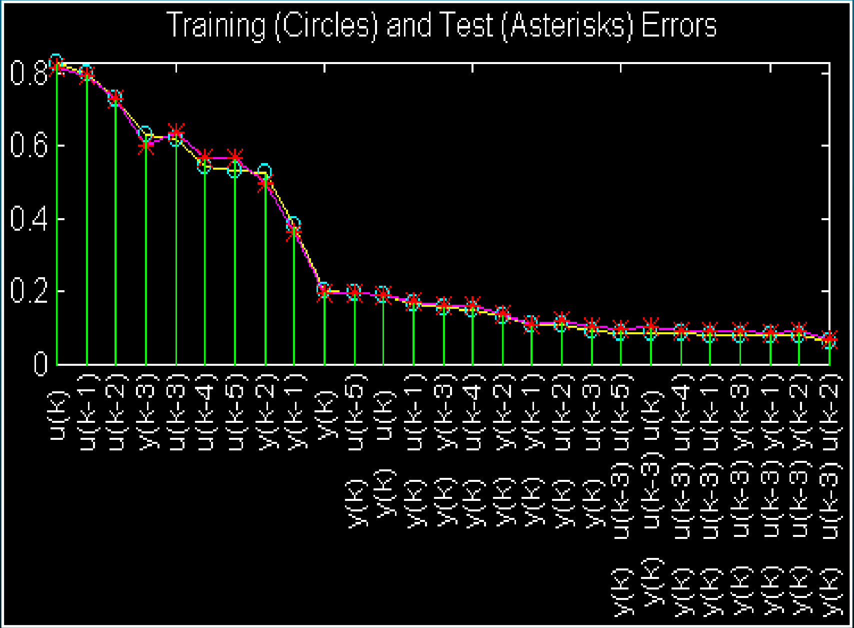


Hair Dryer Modeling

To select 3 inputs out of ten:

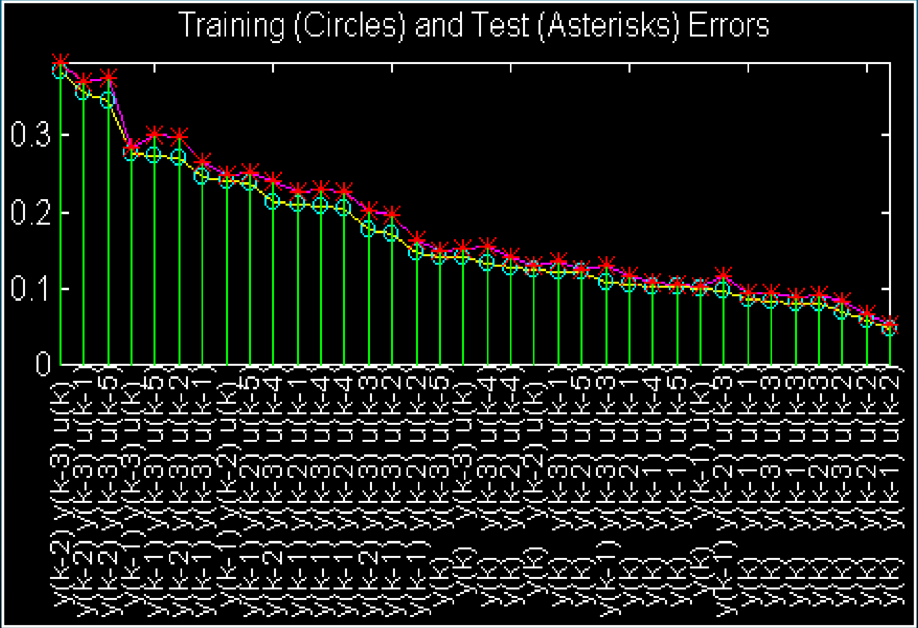
Sequential forward selection

(22.2 s)



Direct exhaustive search

(54.7 s)



Hair Dryer Modeling

ARX model

training RMSE = .114

test RMSE = .072

ANFIS model

training RMSE = .038

test RMSE = .044