NANYANG TECHNOLOGICAL UNIVERSITY

Dynamic Fuzzy Neural Networks : Architectures, Algorithms, and Applications

Meng Joo Er
School of Electrical and Electronic Engineering
Nanyang Technological University
Email: emjer@ntu.edu.sg

Outline

- Introduction
- Dynamic Fuzzy Neural Networks
- Enhanced Dynamic Fuzzy Neural Networks
- Generalized Dynamic Fuzzy Neural Networks
- Applications
- Conclusions

2

Outline

- Introduction
- Dynamic Fuzzy Neural Networks
- Enhanced Dynamic Fuzzy Neural Networks
- Generalized Dynamic Fuzzy Neural Networks
- Applications
- Conclusions

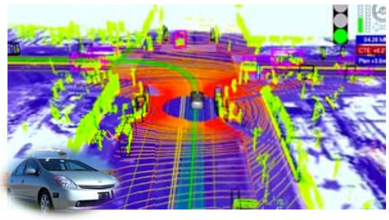
3

Introduction

Artificial Intelligence

(http://www.eka-pr.com/)

Face Recognition



(http://spectrum.ieee.org)

Self-Driving Car

(https://www.theguardian.com)

IBM Watson

(http://www.hk01.com/)

AlphaGo

4

Introduction

Review of Intelligent System Theory

- Fuzzy Systems
- Neural Networks
- Evolutionary Computation
- Expert System
- Data Mining
- … …

Simpler, faster, and more intelligent

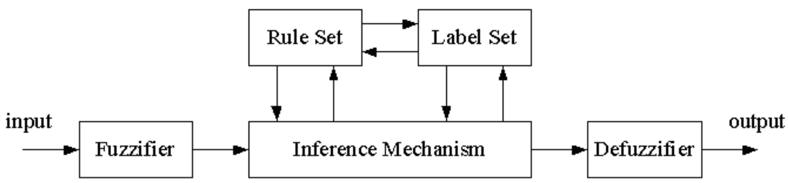
Fuzzy system + Neural network

Fuzzy Neural Network

5

Introduction

Fuzzy System



- **Rule set:** contains a number of fuzzy IF-THEN rules.
- **Label set:** defines the membership functions of the fuzzy sets used in the fuzzy rules.
- **Inference mechanism:** performs the inference operations on the rules.
- **Fuzzifier:** transforms the crisp inputs into degrees of match with linguistic values.
- **Defuzzifier:** transforms the fuzzy results of the inference into a crisp output.

6

Introduction

Fuzzy Set

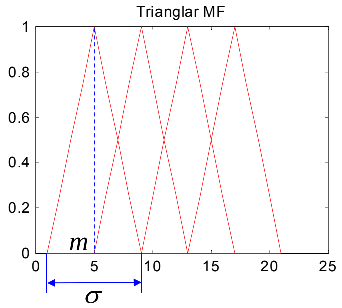
Membership function (MF) is used to describe the degree of belonging of an object

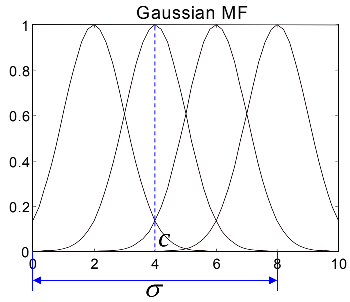
Triangular membership:

$$\mu(x)=\begin{cases}1-\dfrac{|x-m|}{\sigma} & if\ |x-m|\le\sigma \\ 0 & \text{otherwise}\end{cases}$$

Gaussian membership:

$$\mu(x)=\exp[-\frac{(x-c)^2}{\sigma^2}]$$





7

Introduction

Fuzzy Rule

Fuzzy systems are essentially rule-based expert systems, which consist of a set of linguistic rules in the form of "IF-THEN".

Fuzzy IF-THEN rules:

R^i: IF x_1 is F_1^i and …and x_r is F_r^i, THEN y_1 is G_1^i and …and y_s is G_s^i

Takagi-Sugeno-Kang model:

R^i: IF x_1 is F_1^i and...and x_r is F_r^i, THEN $y^i=\alpha_0^i+\alpha_1^i x_1+\cdots\alpha_r^i x_r$

Advantages of TSK model:

- ❑ Computational efficiency
- ❑ Works well with linear techniques
- ❑ Works well with optimization and adaptive techniques
- ❑ Guaranteed continuity of the output surface
- ❑ Better suited to mathematical analysis

8

Fuzzy Inference

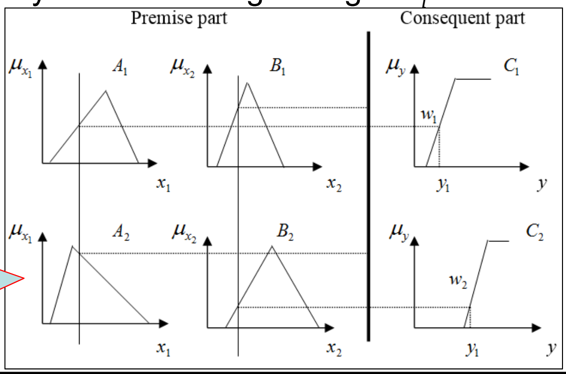
Produce a crisp output by inference operations upon fuzzy IF-THEN rules

Type I - Tsukamoto Fuzzy Models:

The overall output is the weighted average of each rule's crisp output induced by the rule's firing strength w_i :

$$y = \frac{\sum_{i=1}^{u} y^i \times w_i}{\sum_{i=1}^{u} w_i}$$

$$y = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2}$$



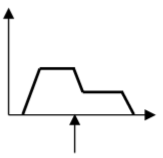
9

Fuzzy Inference

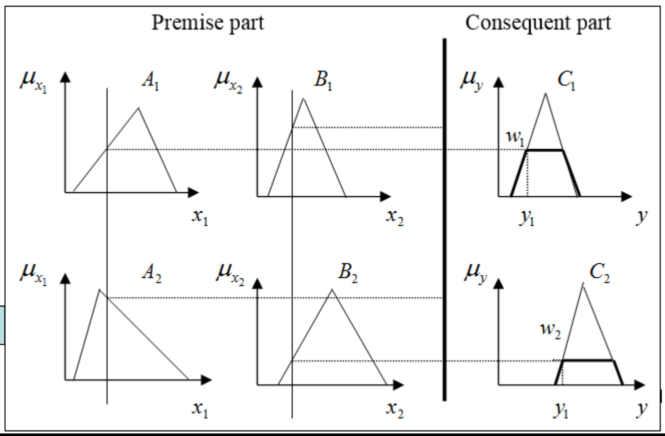
Type II - Mamdani Fuzzy Models:

The overall fuzzy output is derived by applying "maximum" operation to the qualified fuzzy output

$$y = \frac{\sum_{i=1}^{u} \mu_y(w_i) \times w_i}{\sum_{i=1}^{u} \mu_y(w_i)}$$



y (centroid of area)



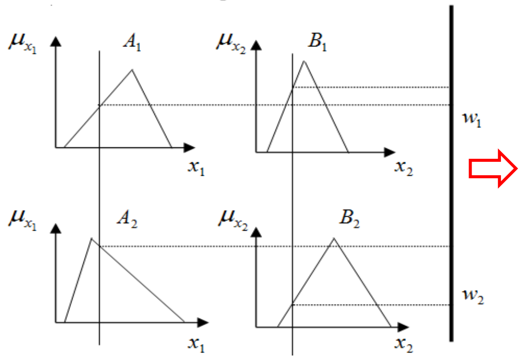
10

Introduction

Fuzzy Inference

Type III - TSK Fuzzy Models:

The output of each rule is a linear combination of input variables plus a constant term and the final crisp output is the weighted average of each rule's output



$$y_1 = \alpha_0^1 + \alpha_1^1 x_1 + \alpha_2^1 x_2$$

$$y_2 = \alpha_0^2 + \alpha_1^2 x_1 + \alpha_2^2 x_2$$

$$y = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2}$$

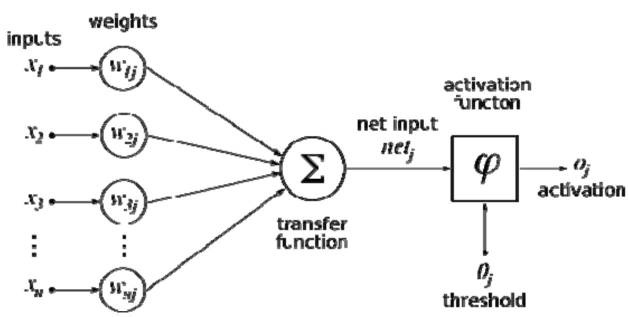
11

Introduction

Neural Networks

A neural network consists of many neurons and can be classified into feedforward (or non-recurrent) and feedback (or recurrent) neural networks

A neuron:



Each neuron performs two functions: aggregation of its inputs from other neurons or the external environment and generation of an output from the aggregated inputs.

12

Introduction

Properties of Neural Networks

- An efficient method to approximate any nonlinear mapping
- Learning ability -- be trained from sample data
- Ability of generalization -- can respond correctly to any inputs which are not learned in the sample data
- Can operate simultaneously on both quantitative and qualitative data
- Naturally process many inputs and outputs and are readily applicable to multivariable systems
- Highly parallel implementation

One of the most salient features of neural networks lies in its learning capability from samples

Introduction

Learning Algorithms

- **Supervised learning:** Data and corresponding labels are given
- **Unsupervised learning:** Only data are given, no labels provided
- **Semi-supervised learning:** Some (if not all) labels are provided
- **Reinforcement learning:** Only provides a scalar performance measure without indicating the direction in which the system could be improved

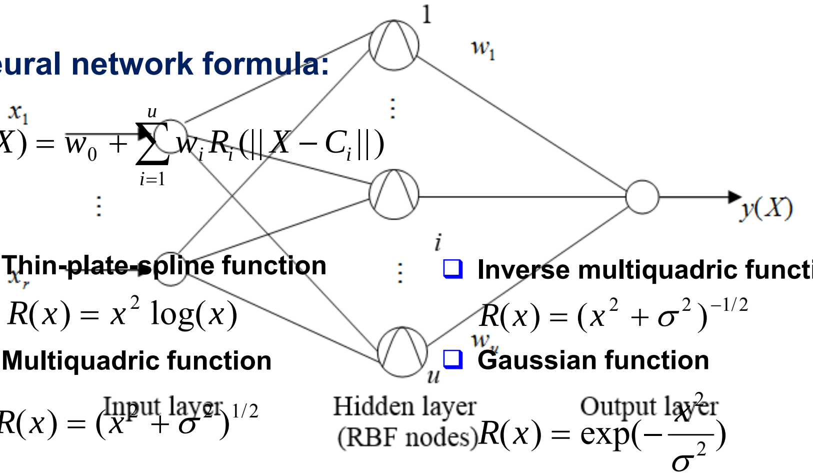
14

Introduction

RBF Neural Networks

Neural network formula:

$$y(X) = w_0 + \sum_{i=1}^{u} w_i R_i(\| X - C_i \|)$$



- Thin-plate-spline function

$$R(x) = x^2 \log(x)$$

- Multiquadric function

$$R(x) = (x^2 + \sigma^2)^{1/2}$$

- Inverse multiquadric function

$$R(x) = (x^2 + \sigma^2)^{-1/2}$$

- Gaussian function

$$R(x) = \exp(-\frac{x^2}{\sigma^2})$$

15

Introduction

Comparison

Skills		Fuzzy System	Neural Networks
Knowledge acquisition	Input	human expert	sample sets
	Tools	interaction	algorithm
Uncertainty	Information	quantitative and qualitative	quantitative
	Cognition	decision making	perception
Reasoning	Mechanism	heuristic search	parallel computation
	Speed	high	low
Adaptation	Fault-tolerance	low	very high
	Learning	induction	adjusting weights
Natural language	Implementation	explicit	implicit
	Flexibility	high	low

Introduction

Fuzzy Neural Networks

Fuzzy System

- Dependent on the specification of good rules by human experts
- No formal framework for the choice of various parameters

Neural Networks

- No easy way to predict the output
- Connecting weights invariably shed no light on what the network will do with the data

Fuzzy Neural Networks

Capture capabilities of both types of systems and overcome drawbacks of each system

FNNs were used to:

- Tune fuzzy systems
- Extract fuzzy rules from given numerical examples
- Develop hybrid systems combining neural networks and fuzzy systems

17

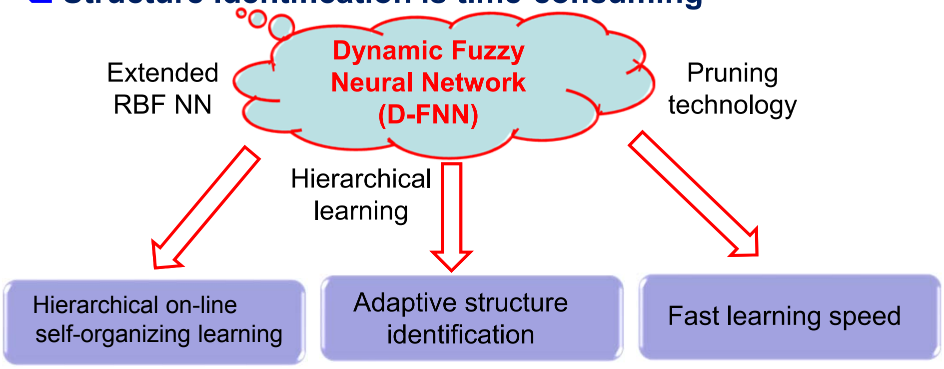
Contents

18

Dynamic Fuzzy Neural Network

Motivation

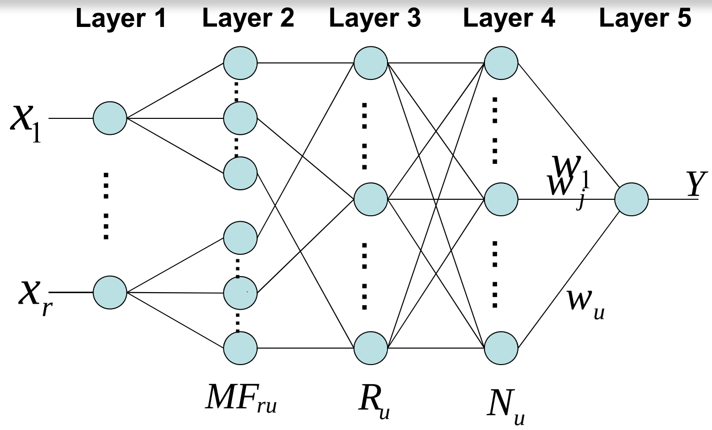
- ❑ **Lack of systematic design for membership functions**
- ❑ **Lack of adaptability for reasoning environment changes**
- ❑ **Structure identification is time-consuming**



19

Dynamic Fuzzy Neural Network

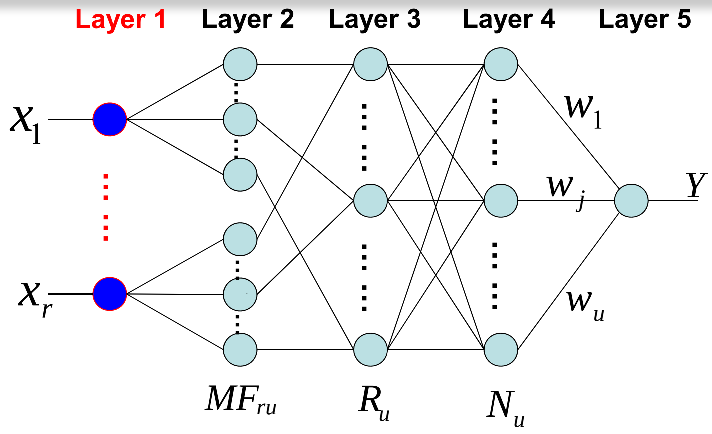
Architecture of D-FNN



20

Dynamic Fuzzy Neural Network

Architecture of D-FNN

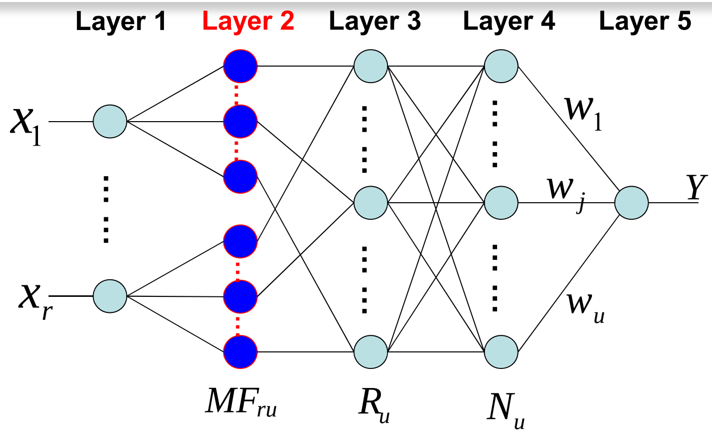


Layer 1: Each node in layer 1 represents an input linguistic variable.

21

Dynamic Fuzzy Neural Network

Architecture of D-FNN



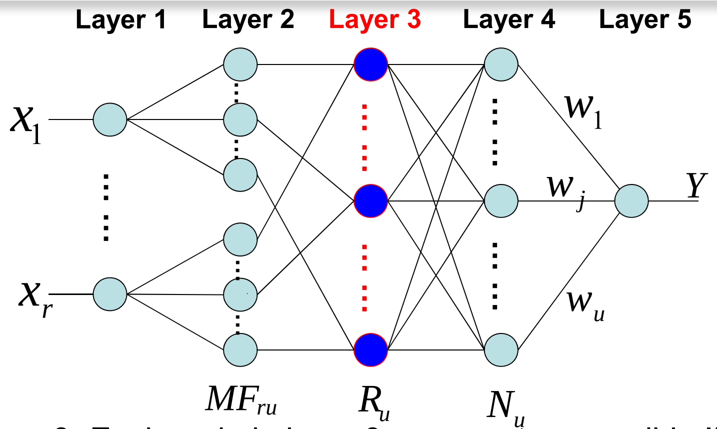
Layer 2: Each node in layer 2 represents a membership function (MF) which is a Gaussian function given by:

$$\mu_{ij}(x_i) = \exp[-\frac{(x_i - c_{ij})^2}{\sigma_j^2}] \qquad i = 1,2\cdots r,\ j = 1,2\cdots u$$

22

Dynamic Fuzzy Neural Network

Architecture of D-FNN



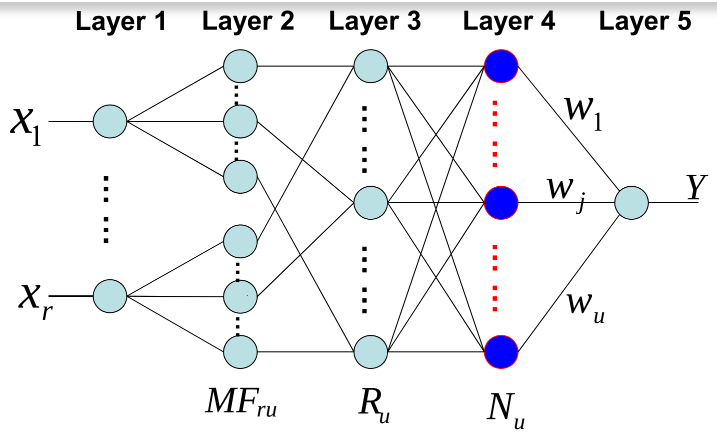
Layer 3: Each node in layer 3 represents a possible IF-part of fuzzy rules.

Output of jth RBF unit: $\phi_j = \exp\left[-\frac{\|X - C_j\|^2}{\sigma_j^2}\right]$ $\quad j = 1,2,\cdots,u$

23

Dynamic Fuzzy Neural Network

Architecture of D-FNN



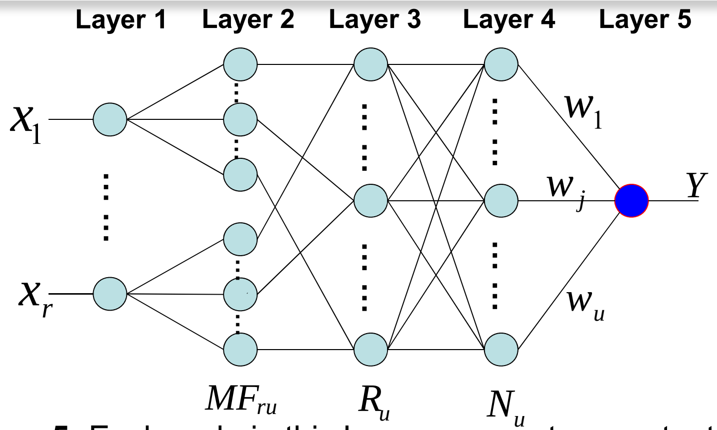
Layer 4: We refer to these nodes as N (normalized) nodes

$$\psi_j = \frac{\phi_j}{\sum_{k=1}^{u} \phi_k} \qquad j = 1,2,\cdots,u$$

24

Dynamic Fuzzy Neural Network

Architecture of D-FNN



Layer 5: Each node in this layer represents an output variable which is the summation of incoming signals

$$y(X) = \sum_{k=1}^{u} w_k \cdot \psi_k$$ TSK model: $w_k = \alpha_{k0} + \alpha_{k1}x_1 + \cdots + \alpha_{kr}x_r$

25

Dynamic Fuzzy Neural Network

Learning Algorithm of the D-FNN

Rule Generation	Allocation of RBF Unit Parameters	Weight Adjustment	Rule Pruning

26

Dynamic Fuzzy Neural Network

Learning Algorithm of the D-FNN

Rule Generation	Allocation of RBF Unit Parameters	Weight Adjustment	Rule Pruning

Error criterion: For observation (X_i, t_i) , define

$$\|e_i\| = \|t_i - y_i\| > k_e ?$$

An RBF unit should be considered

Accommodation criterion:

$$d_i(j) = \|X_i - C_i\|, \quad j = 1,2,\cdots,u$$

If $d_{\min} = \arg\min(d_i(j)) > k_d$, an RBF unit should be considered.

Hierarchical learning: Accommodation boundary is adjusted dynamically

$$k_e = \max[e_{\max} \times \beta^i, e_{\min}]$$
$$k_d = \max[d_{\max} \times \gamma^i, d_{\min}]$$

Coarse learning to fine learning

27

Dynamic Fuzzy Neural Network

Learning Algorithm of the D-FNN

Rule Generation	Allocation of RBF Unit Parameters	Weight Adjustment	Rule Pruning

Allocation of premise parameter :

$$C_i = X_i \qquad \sigma_i = k \times d_{\min}$$

Case 1: $||e_i|| > k_e,\ d_{\min} > k_d$ ⇨ Rule generation

Case 2: $||e_i|| \le k_e,\ d_{\min} \le k_d$ ⇨ Do nothing or only update consequent parameters

Case 3: $||e_i|| \le k_e,\ d_{\min} > k_d$ ⇨ Only adjust consequent parameters

Case 4: $||e_i|| > k_e,\ d_{\min} \le k_d$ ⇨ Update the width of the nearest RBF node and all the weights

For the kth nearest RBF unit: $\sigma_k^i = k_w \times \sigma_k^{i-1}$

28

Dynamic Fuzzy Neural Network

Learning Algorithm of the D-FNN

Rule Generation	Allocation of RBF Unit Parameters	Weight Adjustment	Rule Pruning

Assume nth training pattern enters the D-FNN and all these n data are memorized by the D-FNN in the input matrix P and the output matrix T, respectively, upon which the weights are determined.

Give the output in the compact form as follows:

$$Y = W \times \Psi \qquad \tilde{E} = \|T - Y\|$$

where W is the weight matrix, Ψ is the output matrix of N nodes.

The objective is to find an optimal W^* such that $\tilde{E}^T \tilde{E}$ is minimized.

This problem can be solved by the well-known Linear Least Square (LLS) method by approximating:

$$W^* = T(\Psi^T \Psi)^{-1} \Psi^T$$

29

Dynamic Fuzzy Neural Network

Learning Algorithm of the D-FNN

Rule Generation	Allocation of RBF Unit Parameters	Weight Adjustment	Rule Pruning

Error Reduction Ratio (ERR) Method:

$$D = H\theta + E$$

where $D = T^T, H = \Psi^T$, and $\theta = W^T$.

QR decomposition:

$$H = QA$$

where $Q = (q_1, q_2, \cdots q_v) \in \Re^{n \times v}$

ERR:

$$err_i = \frac{(q_i^T D)^2}{q_i^T q_i D^T D} \qquad i \le v$$

The ERR offers a simple and effective means of seeking a subset of significant regressors

30

Dynamic Fuzzy Neural Network

Learning Algorithm of the D-FNN

Rule Generation	Allocation of RBF Unit Parameters	Weight Adjustment	Rule Pruning

Determination of RBF Units:

Define the ERR matrix as

$$\Delta = (\delta_1, \delta_2, \cdots \delta_u) \in \Re^{(r+1)\times u}$$

Significance of the ith rule is defined as

$$\eta_i = \sqrt{\frac{\delta_i^T \delta_i}{r+1}}$$

If

$$\eta_i < k_{err}$$

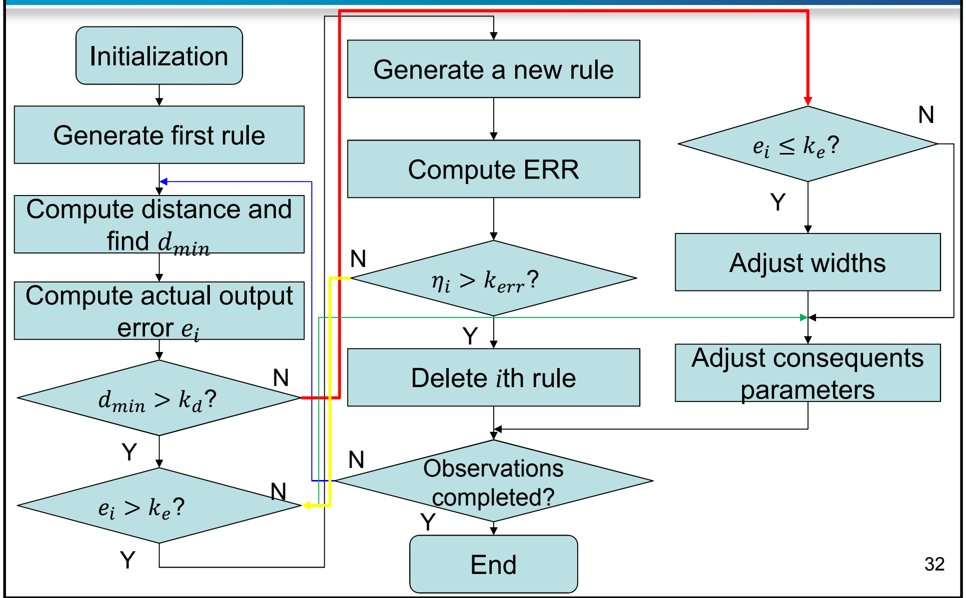
Prespecified threshold

then the ith rule is deleted.

31

Dynamic Fuzzy Neural Network

Flowchart of D-FNN Algorithm



32

Outline

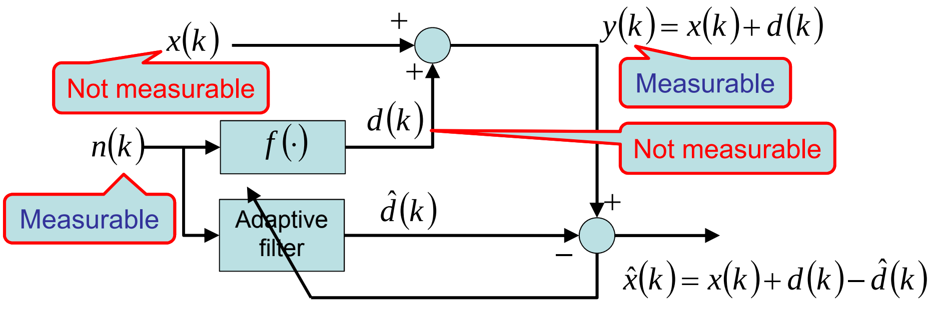
- Introduction
- Dynamic Fuzzy Neural Networks
- **Enhanced Dynamic Fuzzy Neural Networks**
- Generalized Dynamic Fuzzy Neural Networks
- Applications
- Conclusions

33

Enhanced Dynamic Fuzzy Neural Network

Motivation

Adaptive noise cancellation (ANC) using an adaptive filter:

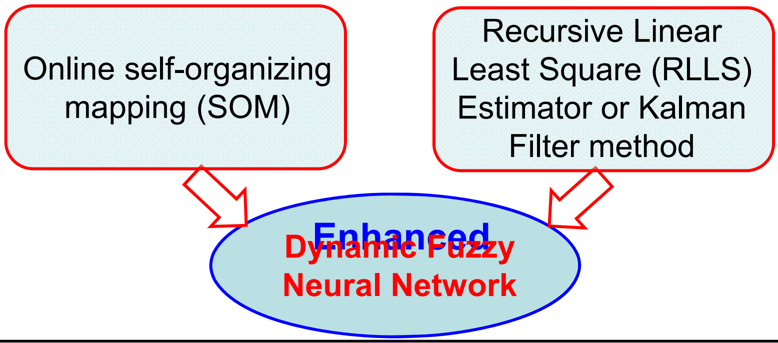


D-FNN cannot be directly applied to ANC, because the neuron generation depends on the systems error, which cannot be defined for ANC.

34

Motivation

- The method based on accommodation boundary and system error could hardly be evaluated online
- Linear Least Square (LLS) method is not very effective in processing signals online in a very noisy environment



ED-FNN Learning Algorithms

The ultimate objective is to implement the D-FNN learning algorithm in real time

Input Space Partitioning: Online SOM is performed for every training pattern (*X(k), y(k)*)

Best Matching Neuron Center:

$$\| X(k) - C_b(k) \| = \min_j \{ \| X(k) - C_i(k) \| \}$$

Update rule:

$$C_j(k+1) = C_j(k) + \alpha(k) h_{bi}(k) [X(k) - C_j(k)]$$

where k is the kth input pattern, $a(k)$ is the adaptation coefficient, and $h_{bi}(k)$, the neighborhood kernel centered on the winner unit, is given by

$$h_{bi}(k) = \exp\left(-\frac{\| C_b - C_j \|^2}{2\sigma^2(k)}\right)$$

ED-FNN Learning Algorithms

Adaptation of the Output Linear Weights: Recursive Linear Least Square (RLLS) estimator is used to adjust the weights

RLLS:

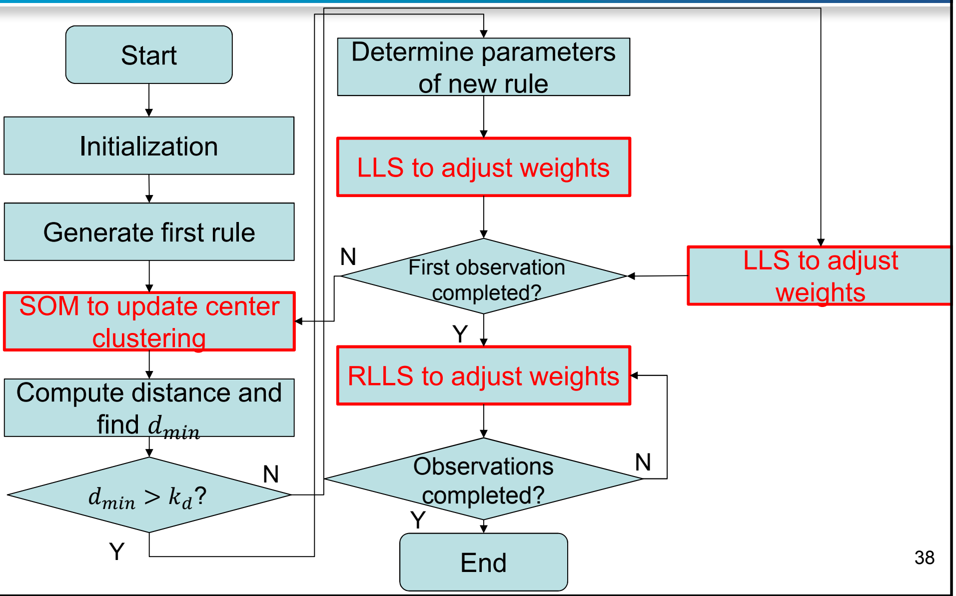
$$S_0 = (\Psi_{i-1}\Psi_{i-1}^T)^{-1}$$

$$S_i = S_{i-1} - \frac{S_{i-1}\Psi_i\Psi_i^T S_{i-1}}{1 + S_{i-1}}$$

$$W_i = W_{i-1} + S_{i-1}\Psi_i(T_i - \Psi_i^T W_{i-1}) \qquad i = 1,2,\cdots,n$$

where S_i is the error covariance matrix for the ith observation.

Flowchart of ED-FNN Algorithm



Outline

- Introduction
- Dynamic Fuzzy Neural Networks
- Enhanced Dynamic Fuzzy Neural Networks
- **Generalized Dynamic Fuzzy Neural Networks**
- Applications
- Conclusions

39

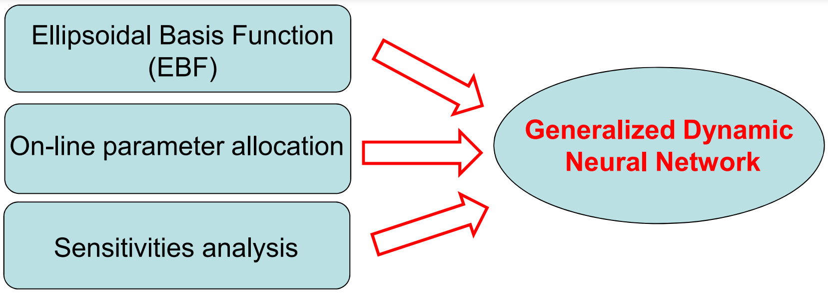
Generalized Dynamic Fuzzy Neural Networks

Motivation

Widths of Gaussian MFs are the same	Do not coincide with the reality
The number of MFs is irrespective of MF distribution	Result in significant overlapping and be opaque for users to understand
The width of a new Gaussian MF is only determined by d_{min} and k	If k and d_{min} is too large, the widths will be very large.
Several prespecified parameters are selected randomly	Not easy for users to implement

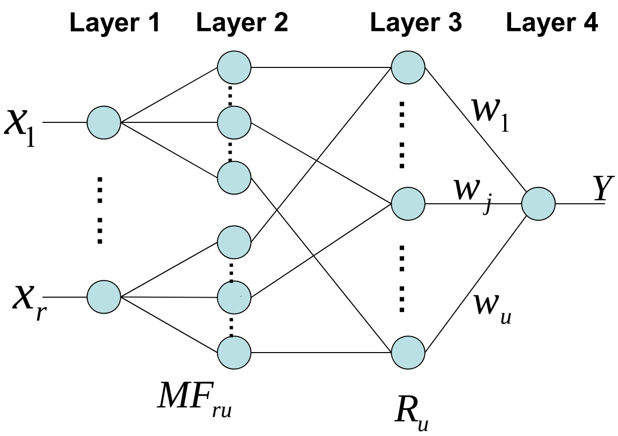
0

GD-FNN



- Structure and parameters identification are performed automatically and simultaneously without partitioning input space and selecting initial parameters a priori
- Fuzzy rules can be recruited or deleted dynamically
- Fuzzy rules can be generated quickly without resorting to the BP iteration learning

Architecture of GD-FNN



Generalized Dynamic Fuzzy Neural Networks

Equivalence of GD-FNN

Corollary 1: The proposed GD-FNN is functionally equivalent to the TSK fuzzy system if the following conditions are true:

- The number of receptive field units of the GD-FNN is equal to the number of fuzzy IF-THEN rules
- The output of each fuzzy IF-THEN rule is a linear combination of input variables plus a constant term
- The membership functions within each rule are chosen as Gaussian functions
- The T-norm operator used to compute each rule's firing strength is multiplication
- Both the GD-FNN and the TSK fuzzy inference system under consideration use the same method, i.e. either weighted average or weighted sum to derive their overall outputs

43

Generalized Dynamic Fuzzy Neural Networks

Learning Algorithm of the GD-FNN

Rule Generation → Propose criteria of rule generation

Premise Parameter Estimation → Allocate parameters of new rules

Sensitivity Analysis → Analyze sensitivity of input variables and fuzzy rules and prune rules

Width Modification → Adjust width to obtain a better local approximation

Consequent Parameter Determination → Determine the optimal parameters W^*

44

Rule Generation

❑ System Errors

$$\| e_k \| = \| t_k - y_k \|$$

If $\| e_k \| > k_e$ a new rule should be considered, where

$$k_e = \begin{cases} e_{\max} & 1 < k < n/3 \\ \max[e_{\max} \times \beta^k, e_{\min}] & n/3 \le k \le 2n/3 \\ e_{\min} & 2n/3 < k \le n \end{cases}$$

where $e_{\min}$ is the desired accuracy of the output of the GD-FNN, $e_{\max}$ is the maximum error chosen, k is the learning epoch, and $\beta \in (0, 1)$, called convergence constant is given by

$$\beta = \left(\frac{e_{\min}}{e_{\max}}\right)^{3/n}$$

45

Rule Generation

❑ ε-Completeness of Fuzzy Rules

Definition (ε-Completeness of Fuzzy input in the operating range, there exis fuzzy rule so that the match degree (no less than ε.

$$\Sigma_j^{-1} = \begin{pmatrix} \frac{1}{\sigma_{1j}^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_{2j}^2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\sigma_{rj}^2} \end{pmatrix}$$

Calculate Mahalanobis distance (M-dis

$$md_k(j) = \sqrt{(X - C_j)^T \Sigma_j^{-1} (X -$$

Find $J = \arg \min_{1 \le j \le u} (md_k(j))$

$$k_d = \begin{cases} d_{\max} = \sqrt{\ln(1/\varepsilon_{\min})} & 1 < k < n/3 \\ \max[d_{\max} \times \gamma^k, d_{\min}] & n/3 \le k \le 2n/3 \\ d_{\min} = \sqrt{\ln(1/\varepsilon_{\max})} & 2n/3 < k \le n \end{cases}$$

If $md_{k,\min} = md_k(J) > k_d$

which implies that the existing system is not satisfied with ε-completeness and a new rule should be considered

*C. C. Lee, *IEEE Trans. Syst, Man and Cybern*, Vol. SMC 20, pp. 404-436, 1990

46

Generalized Dynamic Fuzzy Neural Networks

Premise Parameter Estimation

Definition (Semi-Closed Fuzzy Sets): Let $U = [a, b]$ be the universe of discourse of input x. If each fuzzy set $A_i = \{(x, \mu_i(x)) | x \in U\}$ $(i = 1,2,\cdots,m)$ is represented as the Gaussian function $\mu_i(x) = \exp\left[-\frac{(x-c_i)}{\sigma_i^2}\right]$ for all $x \in U$, and satisfies the following boundary conditions simultaneously:

$$\mu_i(x) = \exp\left[-\frac{(x-a)}{\sigma_i^2}\right] \quad \text{if } |c_i - a| \le \delta$$

$$\mu_i(x) = \exp\left[-\frac{(x-b)}{\sigma_i^2}\right] \quad \text{if } |c_i - b| \le \delta$$

$$\mu_i(x) = \exp\left[-\frac{(x-c_i)}{\sigma_i^2}\right] \quad \text{if } |c_i - a| > \delta \text{ and } |c_i - b| > \delta$$

where δ is a tolerable small value, and the widths of Gaussian functions are selected as

$$\sigma_i(x) = \frac{\max\{|c_i - c_{i-1}|, |c_i - c_{i+1}|\}}{\sqrt{\ln(1/\varepsilon)}}, \quad i = 1,2,\cdots,m$$

where c_{i-1} and c_{i+1} are the two centers of neighboring MFs of ith membership function. Then, we call A_i semi-closed fuzzy sets.

47

Generalized Dynamic Fuzzy Neural Networks

Premise Parameter Estimation

Theorem: The semi-closed fuzzy set A_i satisfy ε-completeness of fuzzy rules, i.e., for all $x \in U$, there exists $i \in \{1,2,\cdots,m\}$, such that $\mu_i(x) \ge \varepsilon$.

Calculate E-distance:

$$ed_i(j) = \left|x_i^k - \Phi_i(j)\right| \quad j = 1,2,\cdots,u+2$$

where $\Phi_i \in \{x_{imin,} c_{i1}, c_{i2}, \cdots, c_{iu}, x_{imax}\}$ and find

$$j_n = \arg \min_{j=1,2,\cdots,u+2} (ed_i(j))$$

If $ed_i(j_n) \le k_{mf}$, x_i^k can be completely represented by the existing fuzzy set $A_{ij_n}(c_{ij_n}, \sigma_{ij_n})$ without generating a new MF. Otherwise, a new Gaussian MF is allocated with its width and center respectively being determined by

$$\sigma_i(x) = \frac{\max\{|c_i - c_{i-1}|, |c_i - c_{i+1}|\}}{\sqrt{\ln(1/\varepsilon)}}, \qquad c_{i(u+1)} = x_i^k$$

48

Sensitivity Analysis

❑ Error Reduction Ratio

Linear regression model:

$$D = H\theta + E$$

where $D = T^T$, $H = \Psi^T$, and $\theta = W^T$.

QR decomposition:

$$H = QA$$

where $Q = (q_1, q_2, \cdots q_v) \in \Re^{n \times v}$

ERR:

$$err_i = \frac{(q_i^T D)^2}{q_i^T q_i D^T D} \qquad i \le v$$

Generalized Dynamic Fuzzy Neural Networks

Sensitivity Analysis

❑ Sensitivity of Fuzzy Rules

Define the ERR matrix

$$\Delta = (\delta_1, \delta_2, \cdots \delta_u) \in \Re^{(r+1) \times u}$$

Significance of the ith rule is defined as

$$\eta_i = \sqrt{\frac{\delta_i^T \delta_i}{r+1}}$$

If

$$\eta_i < k_{err} \qquad i = 1, 2, \cdots, u$$

then the ith rule is deleted.

Generalized Dynamic Fuzzy Neural Networks

Sensitivity Analysis

Sensitivity of Input Variables

Define

$$B_j = \sum_{k=2}^{r+1} \rho_j(k) \qquad j = 1,2,\cdots,u$$

$$B_{ij} = \frac{err_{ij}}{B_j} \qquad i = 1,2,\cdots,r$$

B_j -- total ERR related to input variables in jth rule
err_{ij} -- ERR corresponding to ith input variable in jth rule
B_{ij} -- The significance of ith input variable in jth rule

If B_{ij} is small, ith input variable is not sensitive to the system output, and consequently, width of the hyperellipsoidal region in ith input variable could be reduced without significant effect of system performance.

Generalized Dynamic Fuzzy Neural Networks

Width Modification

In the case $\|e_k\| > k_e$ and $md_{k,min} \le k_d$, which implies the significance of the rule is not good to accommodate all the patterns, and the ellipsoidal region should be reduced.

Width modification:

$$\sigma_{ij}^{new} = \zeta \times \sigma_{ij}^{old}$$

where

$$\zeta = \begin{cases} \dfrac{1}{1 + k_w (B_{ij} - 1/r)^2} & B_{ij} < 1/r \\ 1 & B_{ij} \ge 1/r \end{cases}$$

52

Generalized Dynamic Fuzzy Neural Networks

Consequent Parameter Determination

Give the output in the compact form as follows:

$$W\phi = Y$$

Assume that the desired output is :

$$T = (t_1, t_2 \cdots t_n) \in \Re^n$$

Minimizing

$$\|W\phi - T\|_2$$

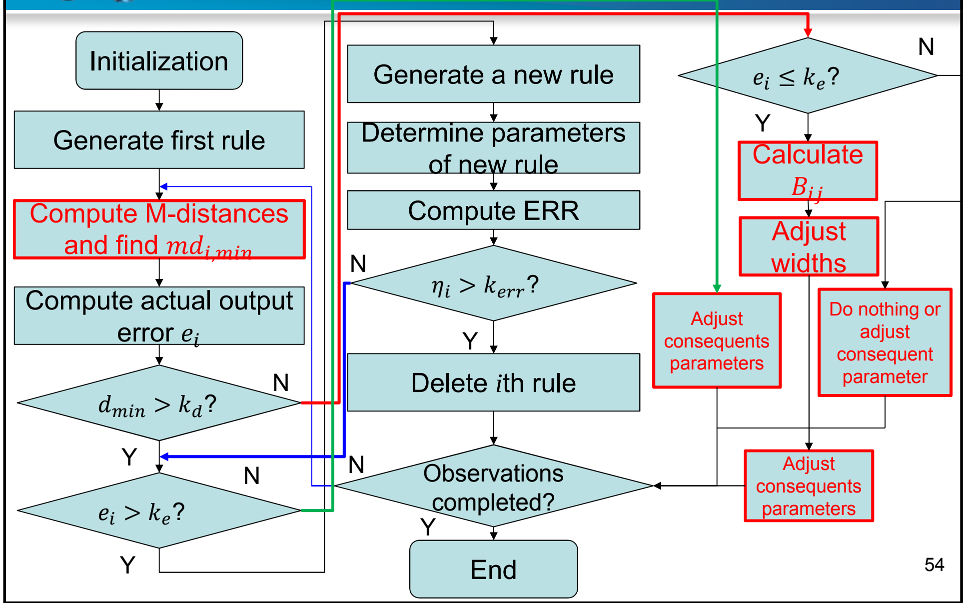
The optimal W^* can be determined by

$$W^* = T(\phi^T\phi)^{-1}\phi^T$$

53

Generalized Dynamic Fuzzy Neural Networks

Flowchart of GD-FNN Algorithm



54

Outline

- Introduction
- Dynamic Fuzzy Neural Networks
- Enhanced Dynamic Fuzzy Neural Networks
- Generalized Dynamic Fuzzy Neural Networks
- **Applications**
- Conclusions

55

Applications

Application 1: Mackey—Glass Time-Series Prediction

Time-series prediction is widely applied in economic and business planning, inventory and production control, weather forecasting, signal processing, control etc.

System model:

$$x(t+1)=(1-a)x(t)+\frac{bx(t-\tau)}{1+x^{10}(t-\tau)}$$

where $a = 0.1$, $b = 0.2$, $\tau = 17$.

Prediction model:

$$x(t+p)=f[x(t),x(t-\Delta t),x(t-2\Delta t),x(t-3\Delta t)]$$

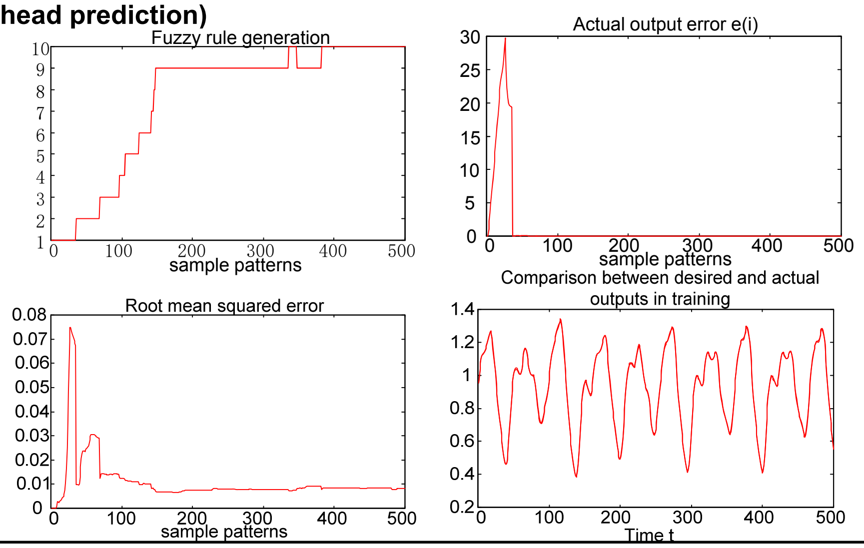
6000 exemplar samples were generated between $t = 0$ and $t = 6000$ with initial conditions $x(t) = 0$ for $t < 0$ and $x(0) = 1.2$.

56

Applications

Application 1: Mackey—Glass Time-Series Prediction

Mackey—Glass time-series prediction (from t = 118 to 617 and six-step-ahead prediction)

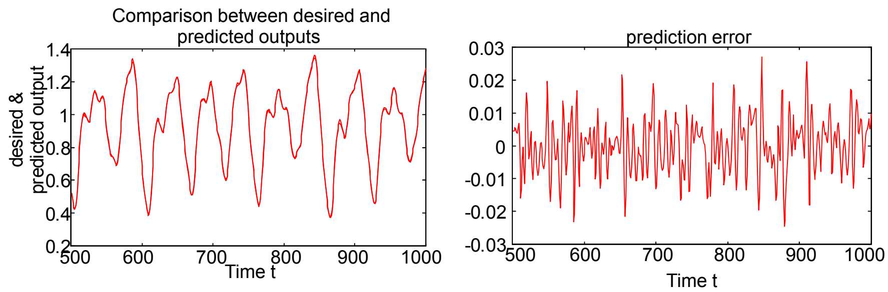


57

Applications

Application 1: Mackey—Glass Time-Series Prediction

Generalization test of the D-FNN for Mackey—Glass time-series prediction (from t = 642 to 1141 and six-step ahead prediction)



58

Applications

Application 1: Mackey—Glass Time-Series Prediction

Comparisons of structure and performance of different algorithms (Training case: n = 500, p = 6)

Technique	Number of rules	Number of parameters	RMSE of training	RMSE of testing
D-FNN	10	100	0.0082	0.0083
ANFIS	16	104	0.0016	0.0015
OLS	35	211	0.0087	0.0089
RBF-AFS	21	210	0.0107	0.0128

Applications

Application 1: Mackey—Glass Time-Series Prediction

Generalization comparison between the D-FNN and the ANFIS for p=85

Technique	Pattern length	Number of rules	Number of parameters	NDEI
D-FNN	500	14	140	0.3375
ANFIS	500	16	104	0.036

Remark: It is shown that the performance of the D-FNN is not so good as that of the ANFIS, even the D-FNN has more adjustable parameters. The reason is that the ANFIS is trained by iterative learning so that an overall optimal solution can be achieved, whereas the D-FNN only acquires a sub-optimal result.

Applications

Application 1: Mackey—Glass Time-Series Prediction

Generalization comparison between D-FNN and RAN, RANEKF and M-RAN for *p*=50

Technique	Pattern length	Number of rules	Number of parameters	NDEI
D-FNN	2000	25	250	0.0544
RAN	5000	50	301	0.071
RANEKF	5000	56	337	0.056
M-RAN	5000	28	169	—*

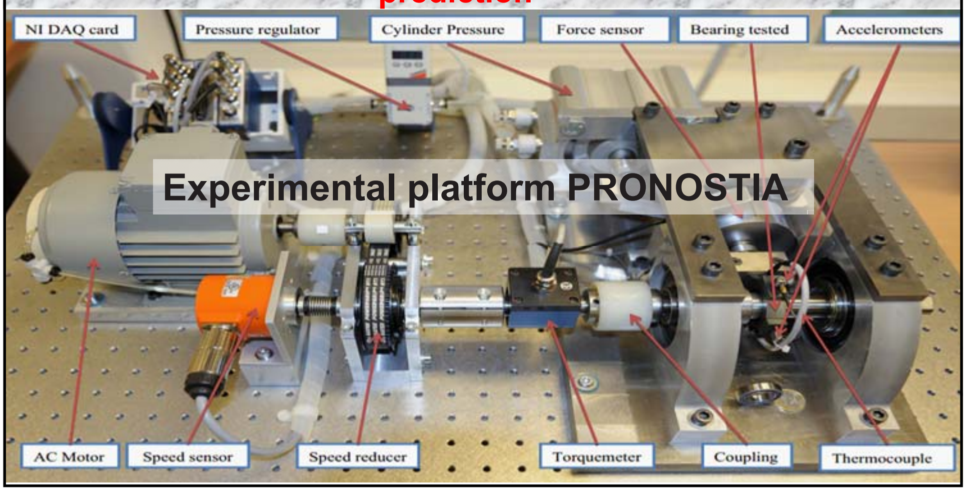
*The generalization error is measured by weighted prediction error (WPE)

61

Applications

Application 2: Prediction of Machine Health Condition

An online dynamic fuzzy neural network (OD-FNN) is developed for prediction of machine health condition prediction



Applications

Application 2: Prediction of Machine Health Condition

Prediction model:

$$y_s(k+r)= f\big(y_s(k), y_s(k-r), y_s(k-2r), \cdots, y_s(k-nr),$$
$$x_s(k), x_s(k-r), x_s(k-2r), \cdots, x_s(k-nr)\big)$$

where x_s and y_s are the input and target signals respectively, r is the prediction step, and $n+1$ is maximum lag.

500 data are applied to initialize the OD-FNN and 500 data pairs are applied to test the OD-FNN performance for online prediction.

63

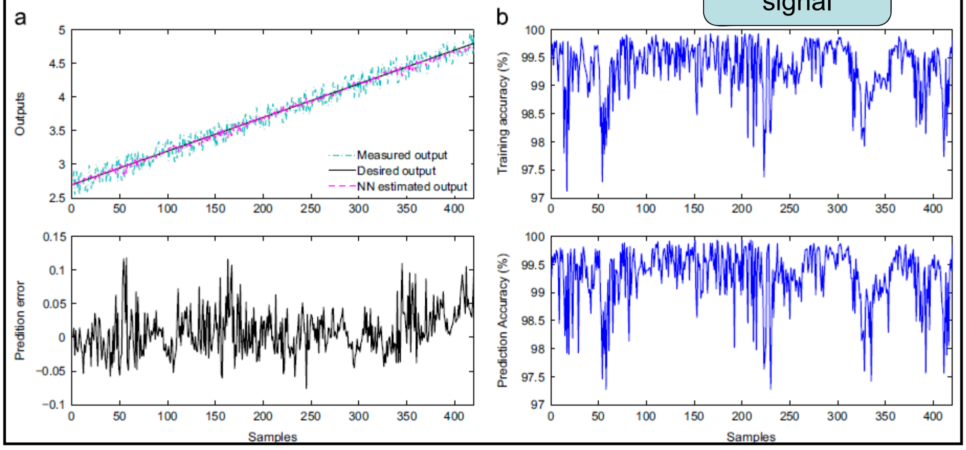
Applications

Application 2: Prediction of Machine Health Condition

Linear model online prediction results: (a) Prediction output with error curves. (b) Accuracies of online training and prediction.

$$\boxed{y_s(i) = 5x_s(i) + rand(t)}$$

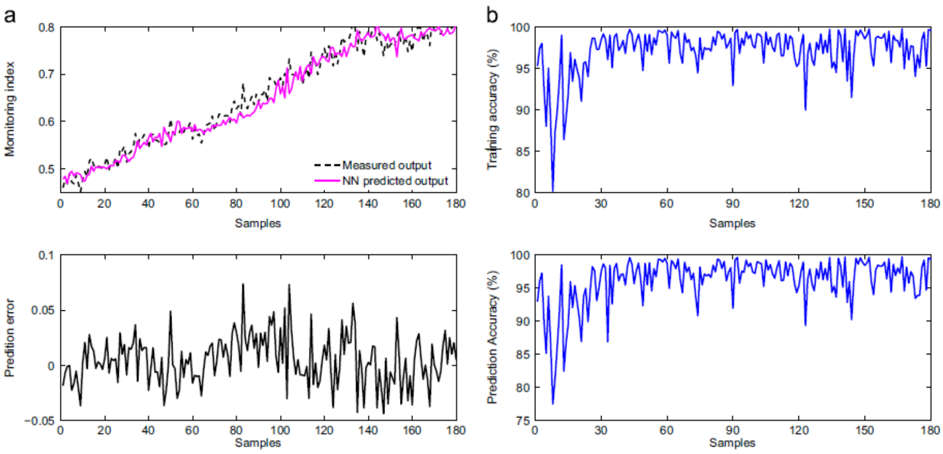
White noise signal



Applications

Application 2: Prediction of Machine Health Condition

Bearing MHC online prediction results: (a) Prediction output with error curves. (b) Accuracies of online training and prediction.



Applications

Application 2: Prediction of Machine Health Condition

Performance comparisons of all prediction methods

Achieve a good trade-off between prediction accuracy and computational cost

Best online prediction accuracy

Case		t_{com}	φ_{train}	φ_{pred}
Linea[illegible] mod[illegible]	[illegible]	8.2471	**99.52**	99.31
	[illegible]N	**1.3841**	99.23	98.28
	RRLS[illegible]NN	5.2681	99.43	**99.34**
	OD-FNN	1.6833	99.31	99.18
Bea[illegible]	[illegible]	4.0210	**98.38**	96.01
	[illegible]N	**0.6765**	97.31	95.35
	RRL[illegible]NN	2.5951	97.34	95.89
	OD-FNN	1.6833	97.94	**97.54**

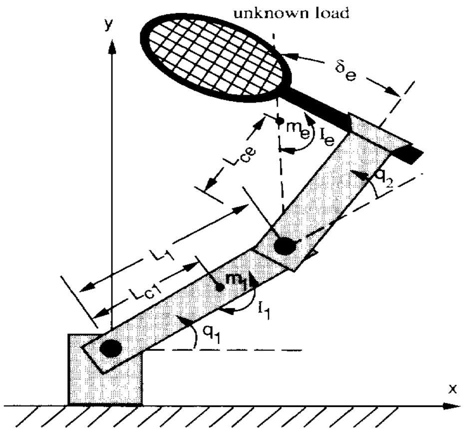
t_{com}-- total computational time φ_{train}--online training accuracy φ_{pred}--online prediction accuracy

66

Applications

Application 3: Online Manipulator Control

Articulated two-link robot manipulator



67

Applications

Application 3: Online Manipulator Control

Manipulator dynamics:

$$M(q)\ddot{q} + Q(q,\dot{q}) + \tau_d = \tau = [\tau_1 \quad \tau_2]^T$$

where

$$\tau_1 = [(I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2) + 2(m_e l_1 l_{ce}\cos\delta_e)\cos q_2 + 2(m_e l_1 l_{ce}\sin\delta_e)\sin q_2]\ddot{q}_1$$
$$+[(I_e + m_e l_{ce}^2) + (m_e l_1 l_{ce}\cos\delta_e)\cos q_2 + (m_e l_1 l_{ce}\sin\delta_e)\sin q_2]\ddot{q}_2$$
$$-[(m_e l_1 l_{ce}\cos\delta_e)\sin q_2 - (m_e l_1 l_{ce}\sin\delta_e)\cos q_2]\dot{q}_1\dot{q}_2$$
$$-[(m_e l_1 l_{ce}\cos\delta_e)\sin q_2 - (m_e l_1 l_{ce}\sin\delta_e)\cos q_2](\dot{q}_1 + \dot{q}_2)\dot{q}_2$$
$$+\tau_{d1}$$

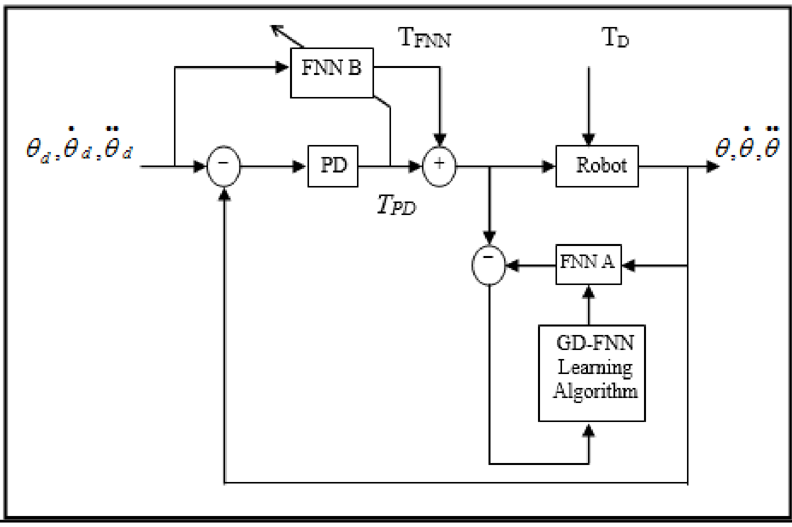
$$\tau_2 = [(I_e + m_e l_{ce}^2) + (m_e l_1 l_{ce}\cos\delta_e)\cos q_2 + (m_e l_1 l_{ce}\sin\delta_e)\sin q_2]\ddot{q}_1 + (I_e + m_e l_{ce}^2)\ddot{q}_2$$
$$+[(m_e l_1 l_{ce}\cos\delta_e)\sin q_2 - (m_e l_1 l_{ce}\sin\delta_e)\cos q_2]\dot{q}_1^2$$
$$+\tau_{d2}$$

68

Applications

Application 3: Online Manipulator Control

Online adaptive fuzzy neural control system:



69

Applications

Application 3: Online Manipulator Control

Performance comparison of GD-FNN with PD and (adaptive fuzzy controller) AFC controllers:

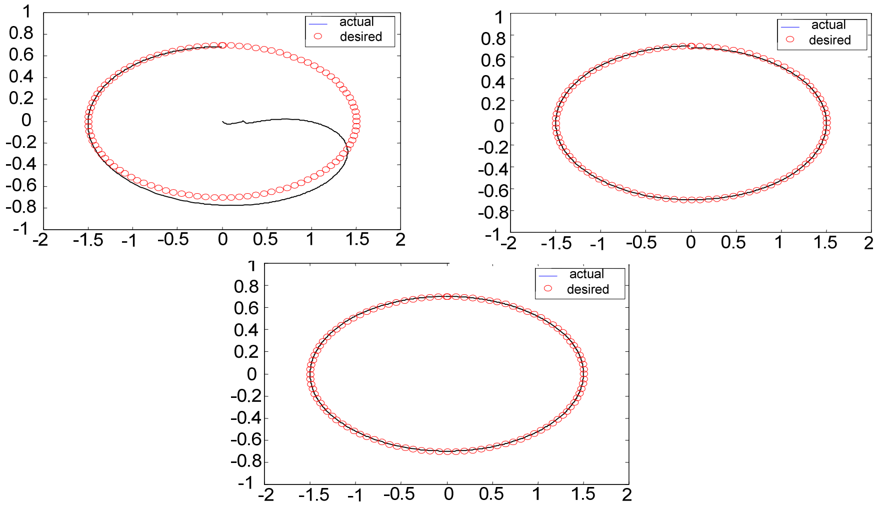
		Steady State Error (rad)	Overshoot (%)	Rise Time (sec)	Maximum Control Torque (Nm)
PD	q_1	0.050	9.3	0.287	312.7
K_p =6000; K_v =100	q_2	0.018	7.1	0.254	109.9
AFC	q_1	0.022	0	0.237	425.9
K_p =25; K_v =35	q_2	0.025	0	0.218	255.6
GD-FNN	q_1	**0.001**	0	0.01	280.3
K_p =25; K_v =7	q_2	**0.001**	0	0.01	110.0

70

Applications

Application 3: Online Manipulator Control

Trajectory tracking results of adaptive online control

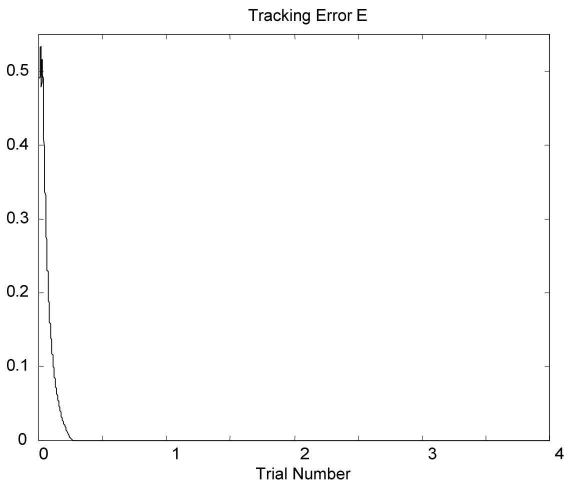


71

Applications

Application 3: Online Manipulator Control

Convergence of tracking error



72

Outline

- Introduction
- Dynamic Fuzzy Neural Networks
- Enhanced Dynamic Fuzzy Neural Networks
- Generalized Dynamic Fuzzy Neural Networks
- Applications
- **Conclusions**

73

Conclusions

- A D-FNN based on extended Radial Basis Function (RBF) neural network is introduced, which has the following properties: online learning, hierarchical learning, dynamic self-organizing structure, fast speed in learning, and generalization in learning.
- An Enhanced Dynamic Fuzzy Neural Network (ED-FNN) learning algorithm based on SOM and RLLS Estimator is developed, which is suitable for real-time applications and it outperforms other approaches

74

Conclusions and Recommendations

Conclusions

- A GD-FNN based on Ellipsoidal Basis Function (EBF) is introduced, which can provide a simple and fast approach to configure a fuzzy system so that
  - some meaningful fuzzy rules could be acquired for knowledge engineering, and
  - the system could be used as a system modeling tool for control engineering, pattern recognition, etc..

75

Thank You!