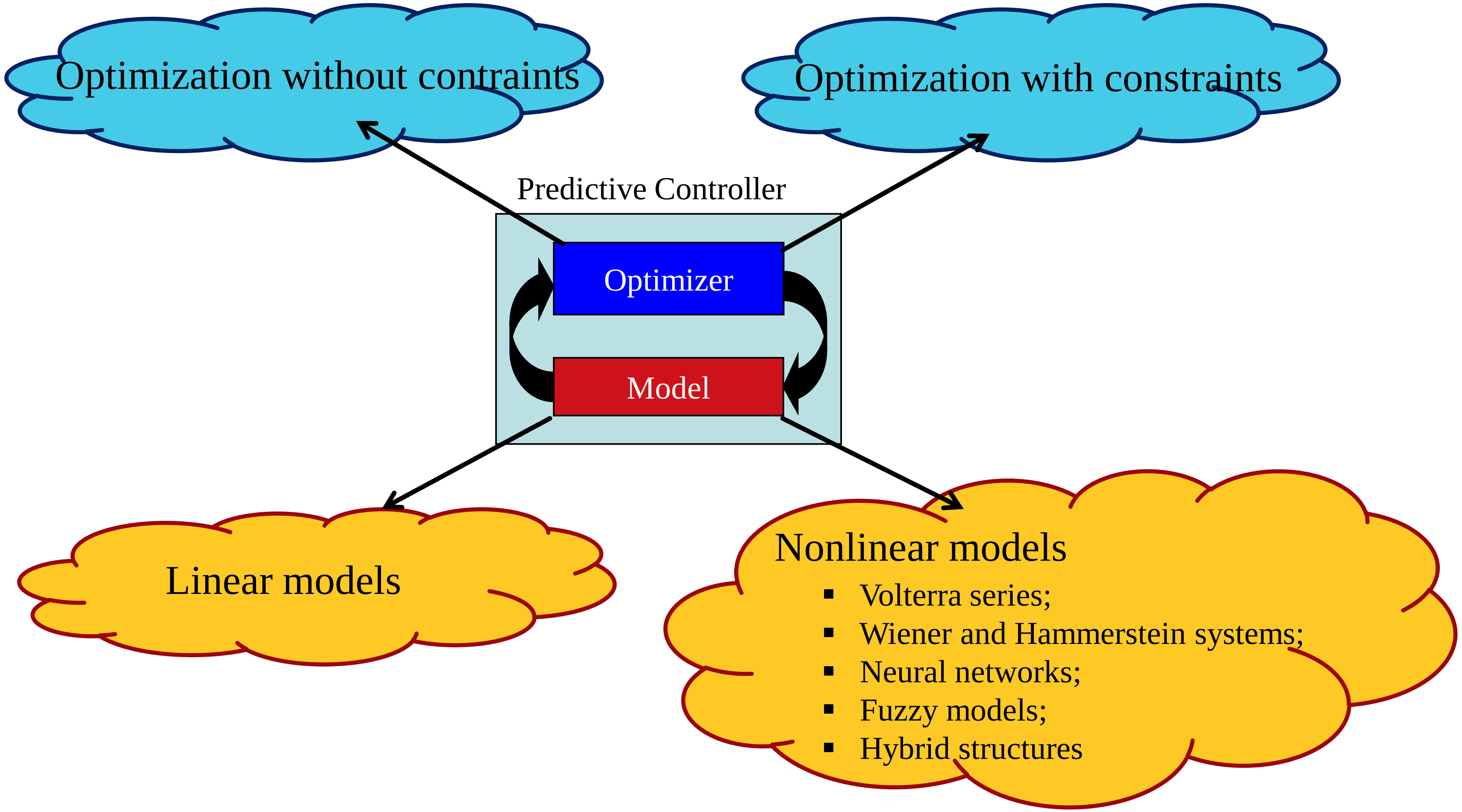
REDUCED RULE-BASE FUZZY-NEURAL NETWORKS

MARGARITA TERZIYSKA, YANCHO TODOROV


Optimization without contraints
Optimization with constraints
Predictive Controller
Optimizer
Model
Linear models
Nonlinear models
Volterra series;
Wiener and Hammerstein systems;
Neural networks;
Fuzzy models;
Hybrid structures

Why fuzzy-neural networks?

- **Universal approximators**
- **Adaptive structures with learning abilities**

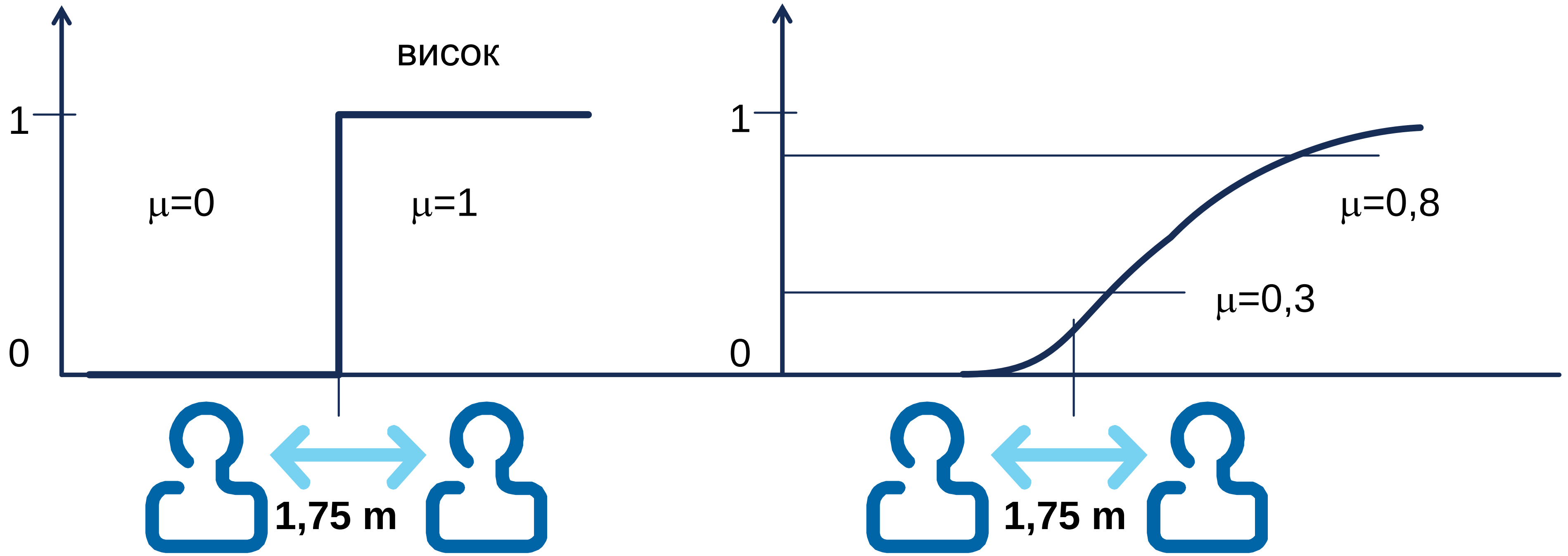
Disadvantages

- **A severe computational burden, due to the large number of generated fuzzy rules and their associated parameters**

Solution?

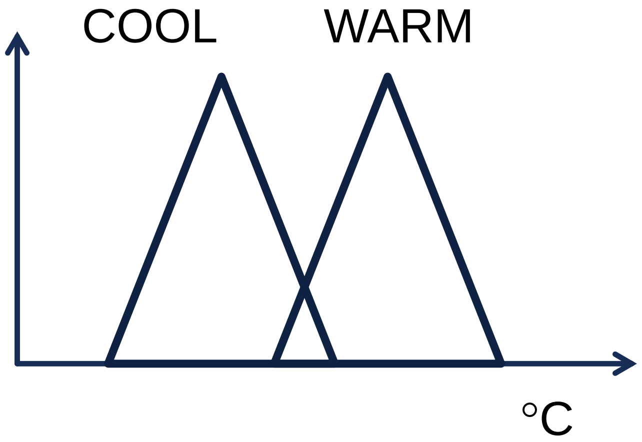
- **Simplified fuzzy-neural structures**
- **Neuroprogramming for optimization**

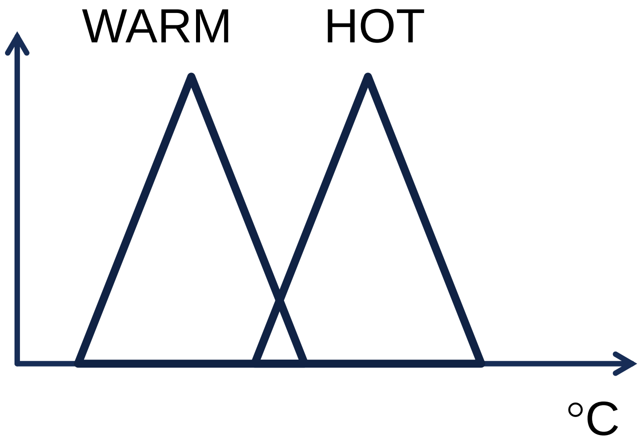
If we separate a group of people assuming that every person of height above 1.75 m is TALL :

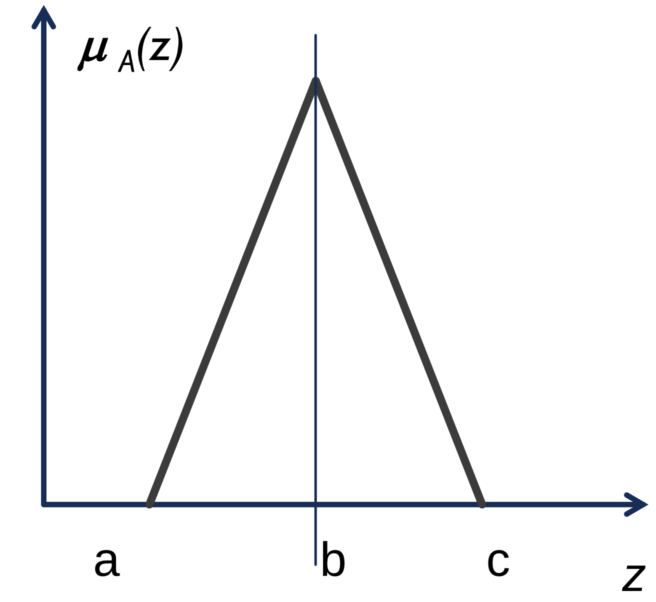


COLD	COOL	WARM	HOT
0 – 15 °C	15-25 °C	25 – 38 °C	40-100 °C

COLD	COOL	WARM	HOT
0 – 20 °C	20-27 °C	27 – 45 °C	45-100 °C







$$\mu_A(z) = \begin{cases} \dfrac{z-a}{b-a} & b > z > a \\ 1 & z = b \\ \dfrac{c-z}{c-b} & c > z > b \end{cases}$$

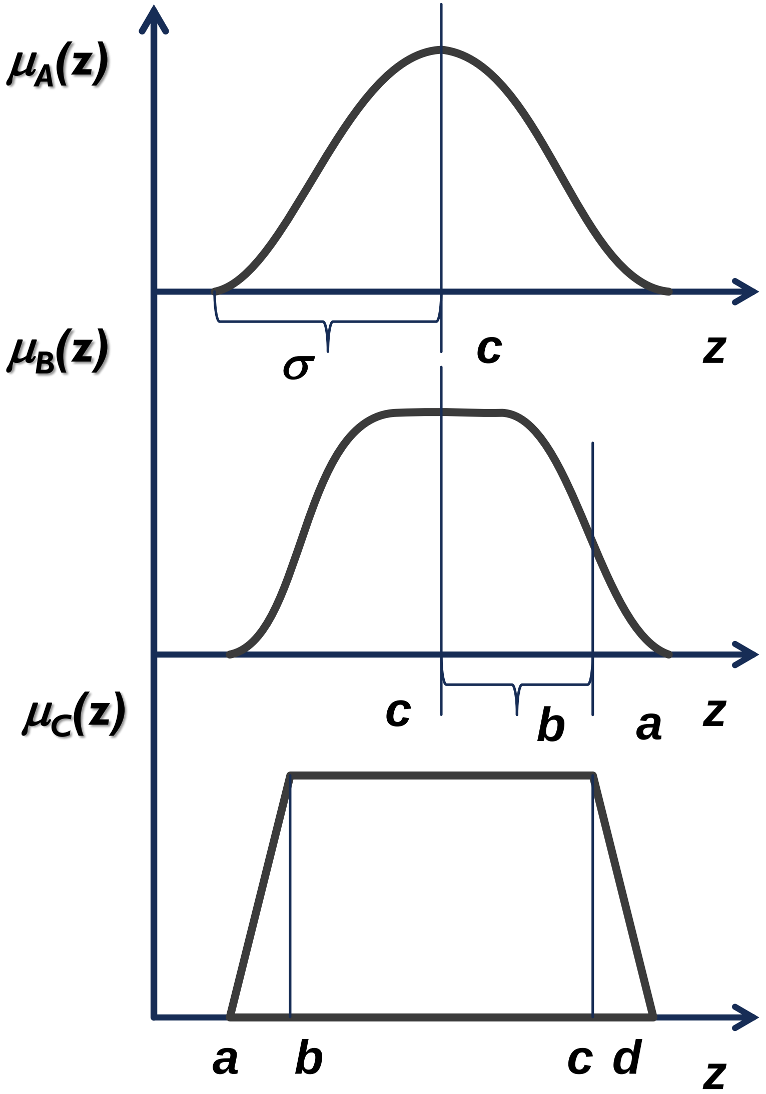
If **Z** is a space of elements, with main component **Z** described by **z**, such that:

Z = {**z**}

Thus a fuzzy set **A** in **Z** is characterized by a **membership function** $\mu_{\mathbf{A}}(\mathbf{z})$, which associate each point in **z** with real number in **[0,1]**, along with the elements $\mu_{\mathbf{A}}(\mathbf{z})$ for **z**, which are called **degree of membership** of **z** in **A**.

If the value of $\mu_{\mathbf{A}}(\mathbf{z})$ is closer to **one**, then a greater value of the degree of membership of **z** to **A is assigned**.

A={z, μ(z)}, z ∈ Z, $\mathbf{\mu_A(z)} \in$ **[0,1]**

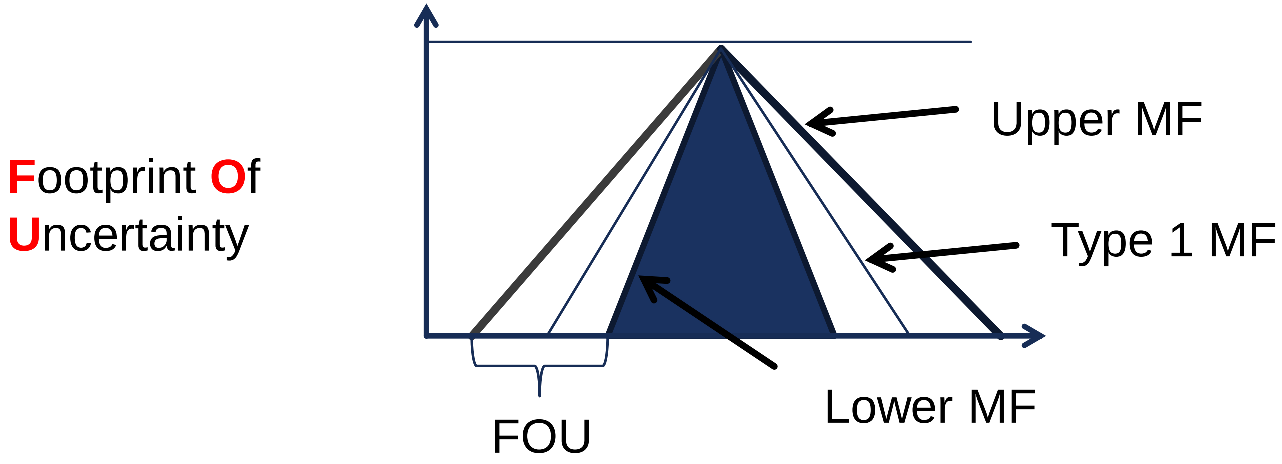


$$\mu_A(z) = \exp\left(-\frac{z-c}{2\sigma^2}\right)$$

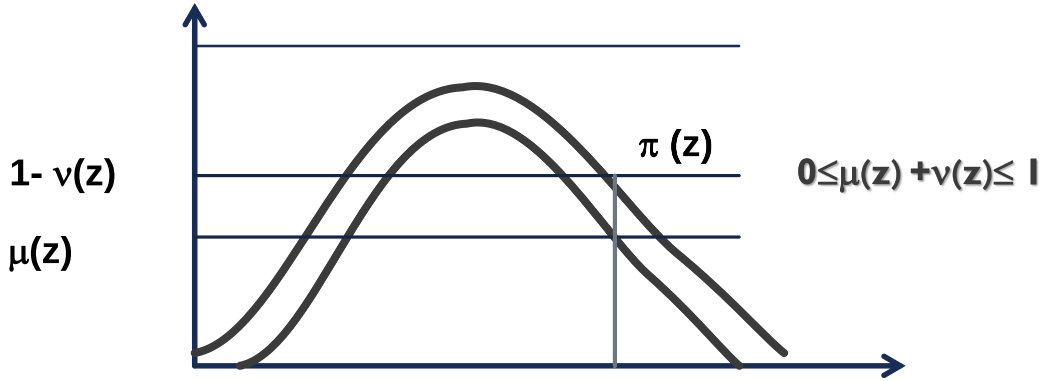
$$\mu_B(z) = \frac{1}{1+\left|\frac{z-c}{a}\right|^{2b}}$$

$$\mu_C(z) = \begin{cases} \frac{z-a}{b-a} & b > z > a \\ 1 & c > z > b \\ \frac{d-z}{d-c} & c > x > b \end{cases}$$

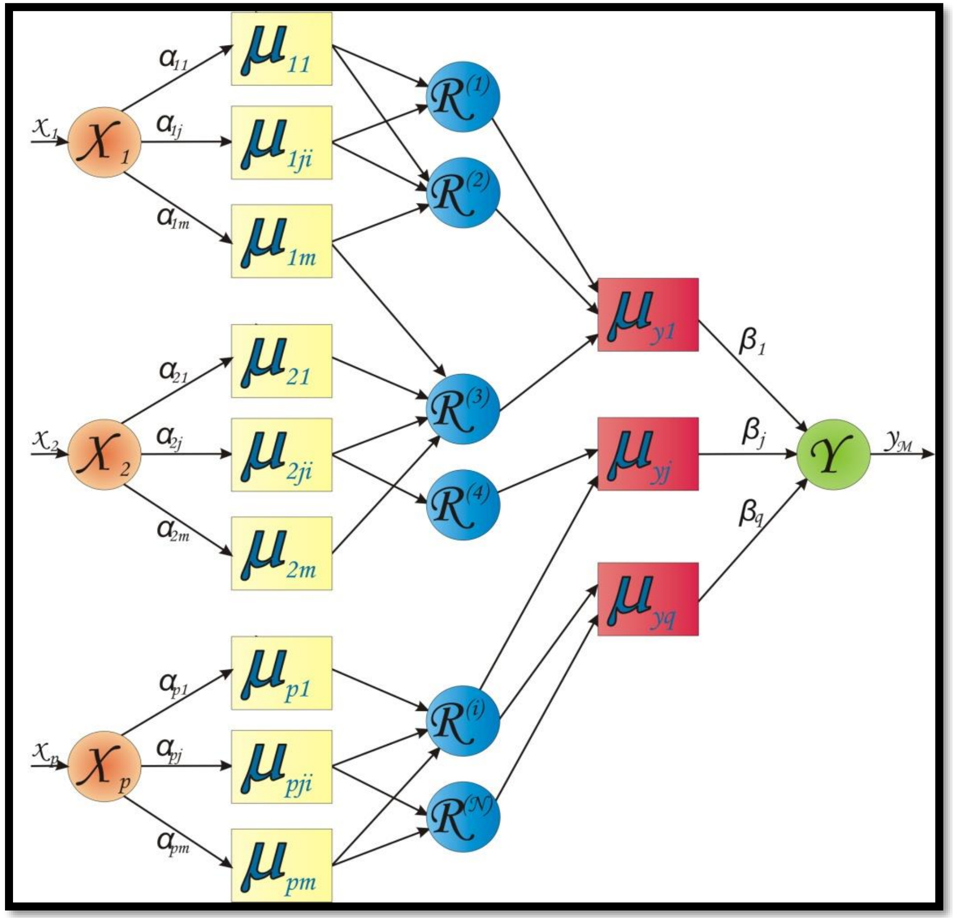
Type 2 Interval Fuzzy Set



Intuitionistic Fuzzy Set



TAKAGI-SUGENO FUZZY-NEURAL SYSTEM



Layer 1: Nonparametric layer distributing the input vector space X(p)

Layer 2: Parametric layer, performing fuzzification using Gaussian MF

$$\mu^{(n)}_{Xp,m} = \exp\frac{-(x_p - c_{Xp,m})^2}{2\sigma^2_{Xp,m}}$$

Layer 3: Rules generator with consequents:

$$f_y^{(N)}(k+j) = a_1^{(N)} y(k+j-1) + \ldots + a_{ny}^{(N)} y(k+j-n_y) + \ldots$$
$$+ b_1^{(N)} u(k+j) + \ldots + b_{mu}^{(N)} u(k+j-n_u) + b_0^{(N)}$$

Layer 4: Fuzzy implication:

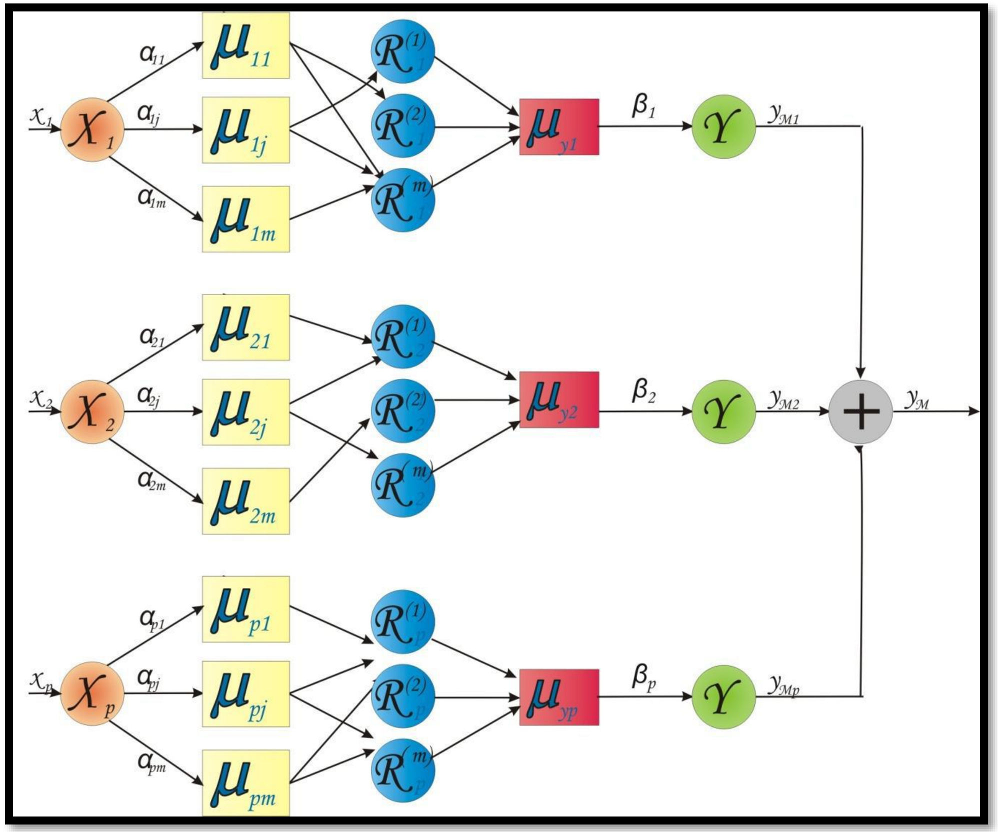
$$\mu^{(n)}_{yq}(k+j) = \mu^{(n)}_{x_1,m}(k+j) * \mu^{(n)}_{x_2,m}(k+j) * \ldots * \mu^{(n)}_{x_p,m}(k+j)$$

Layer 5: Output of the network:

$$y_M = \left(\sum_{i=1}^{q} f_y^{(i)} \mu_y^{(i)}\right) \Big/ \left(\sum_{i=1}^{q} \mu_y^{(i)}\right)$$

$N = m^p = 81$ $P = 1053!$

DISTRIBUTED ADAPTIVE NEURO-FUZZY ARCHITECTURE- DANFA



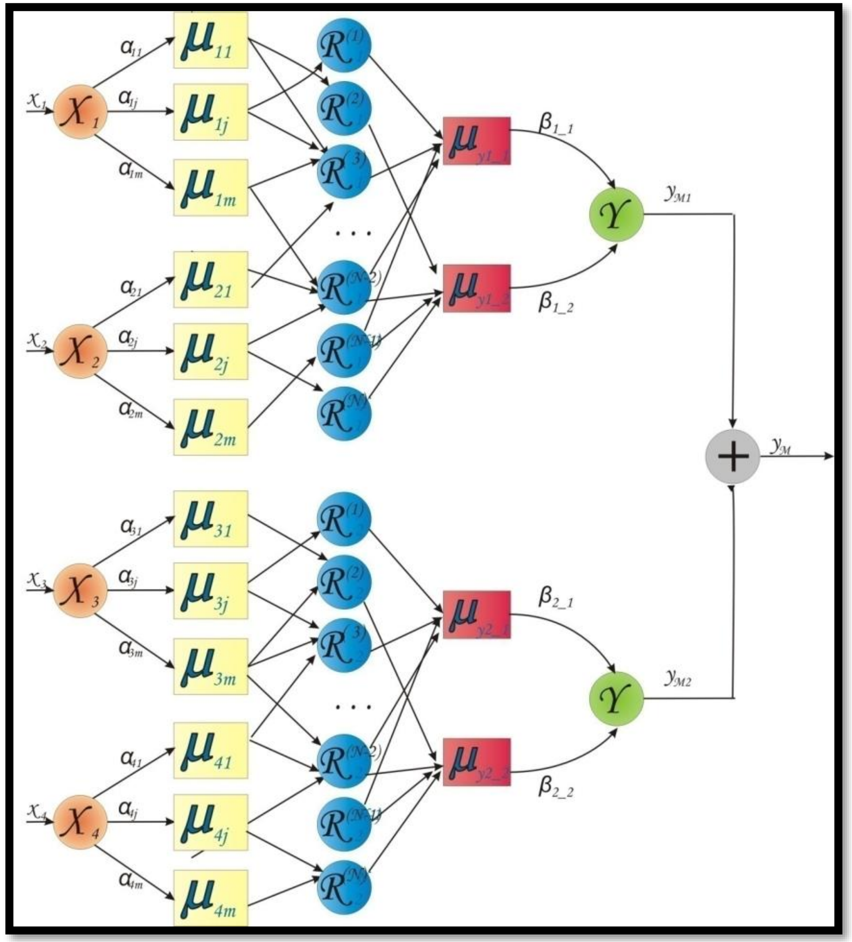
(Distributed Adaptive Neuro Fuzzy Architecture - DANFA)

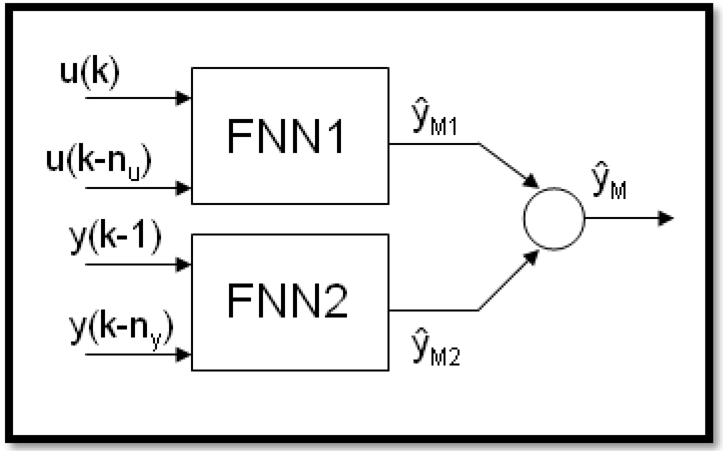
MAIN IDEA: "Distribution of the input space along small and simple fuzzy inferences!

RESULT : The network is a simple structure of q sub-models!

$$N = m_1^{p_1} + m_2^{p_2} + \ldots + m_q^{p_q}$$

DISTRIBUTED ADAPTIVE NEURO-FUZZY ARCHITECTURE- DANFA



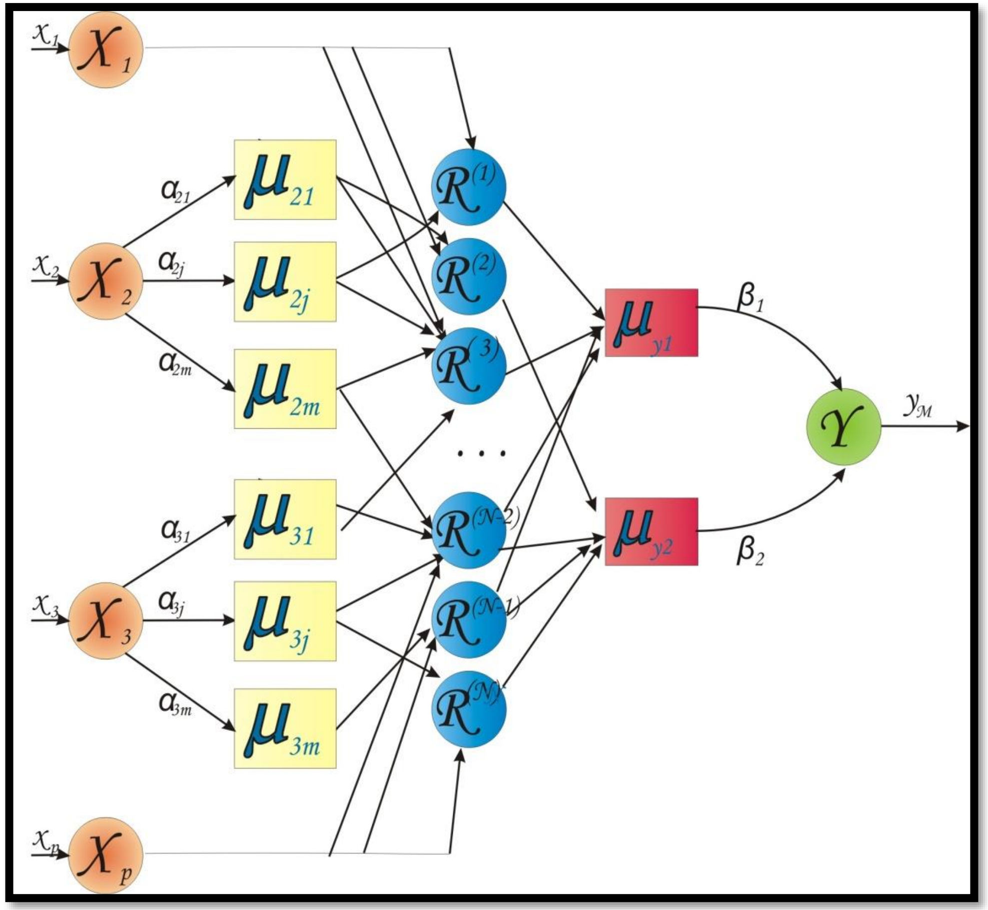


Generalized structure of the DANFA approach

Number of the fuzzy rules: 18

Number of the trained parameters: 126

SEMI FUZZY-NEURAL NETWORK



(Semi Fuzzy Neural Network - **SFNN**)

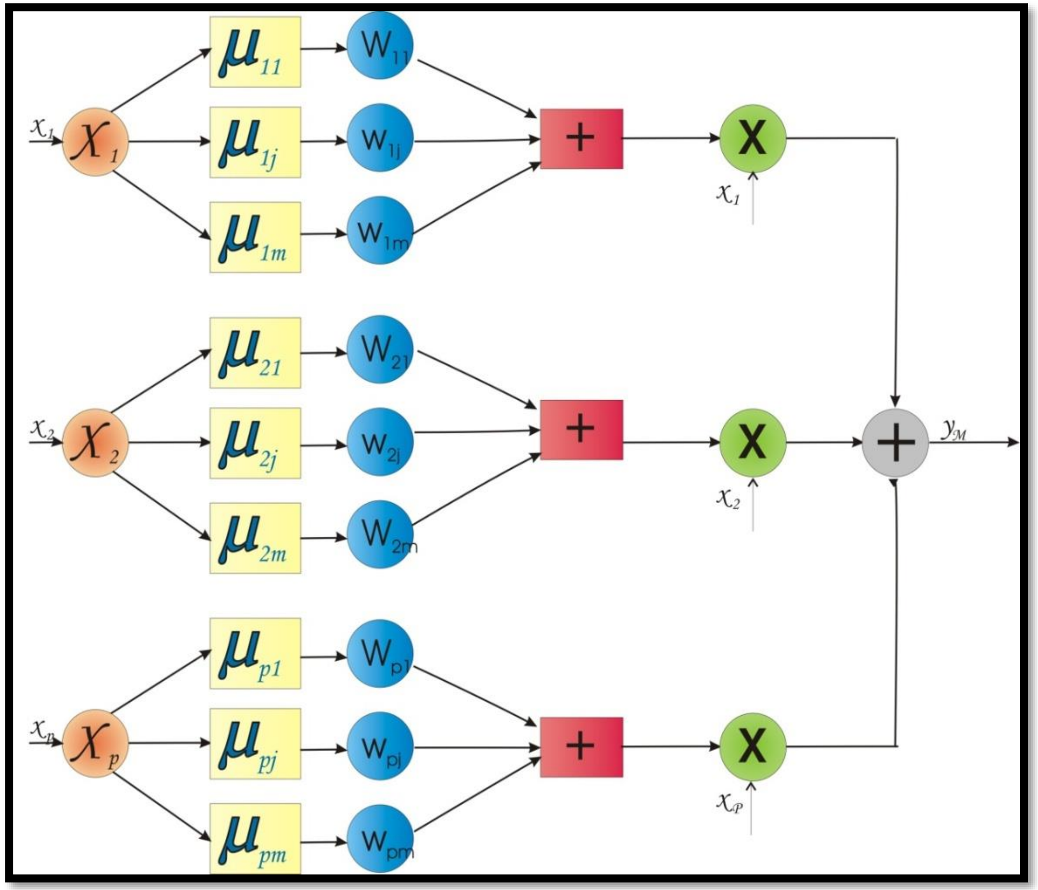
AIM: To fuzzify partially the input space!;

A major question?

How to choose which inputs to be fuzzified!

Number of the Fuzzy rules: 9
number of the parameters: 63

TYPE-2 NEO-FUZZY NEURAL NETWORK



Neo-Fuzzy Neural Network

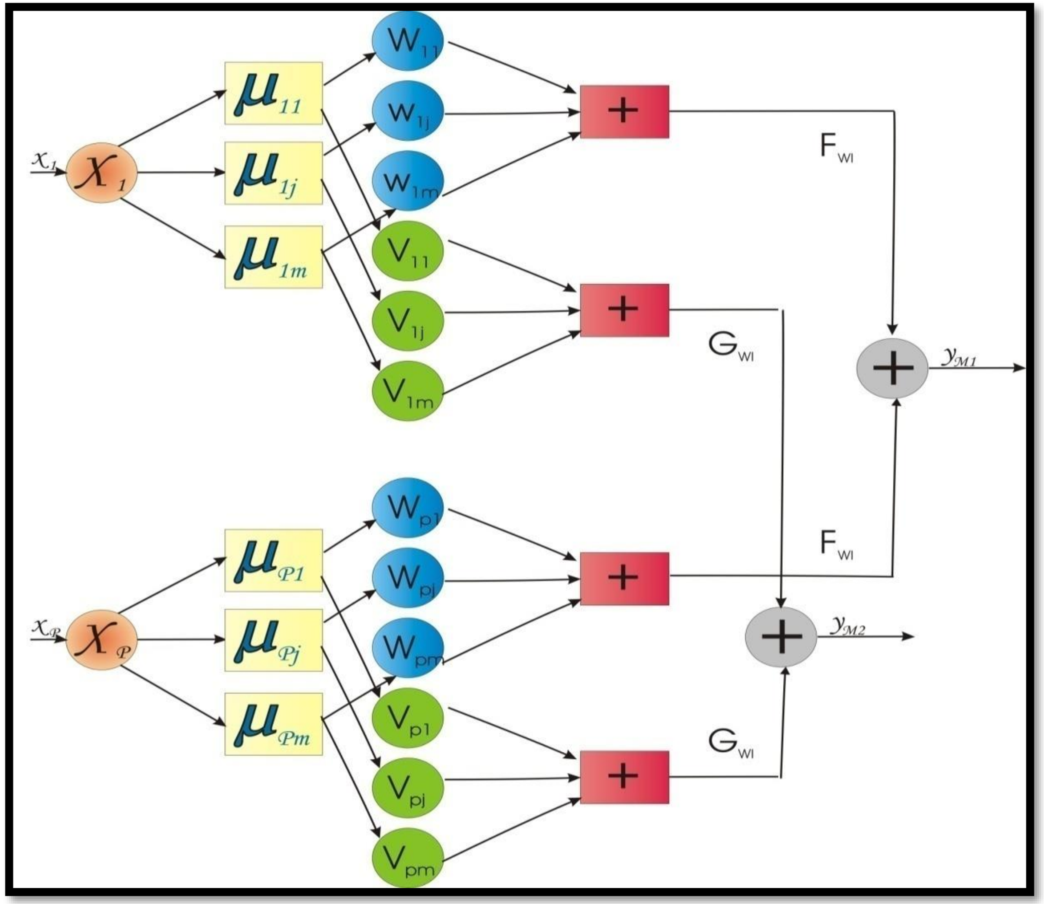
Structure: RBF neural network with fuzzy inference mechanism

$$y_m(k) = \sum_{i=1}^{p} f_i(x_i(k))$$

$$f(x) = \sum_{j=1}^{m} \mu_j(x(k))w_j$$

number of the fuzzy rules: 12
number of the parameters: 36

INTUITIONISTIC NEO-FUZZY NEURAL NETWORK



$$y_{m1} = \sum_{j=1}^{p} F_{wj}(x) \qquad y_{m2} = \sum_{j=1}^{p} G_{wj}(x)$$

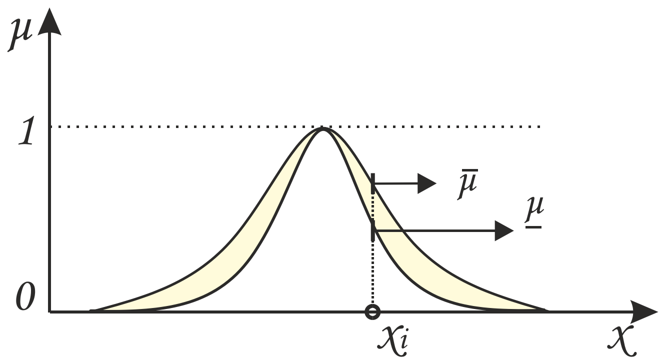
$$w_{ij}(k+1) = w_{ij}(k) + \Delta w = w_{ij}(k) - \eta\left(\frac{\partial E_1(k)}{\partial w(k)}\right) =$$

$$= w_{ij}(k) + \eta e(k)\mu_{ij}(x_i(k))$$

$$v_{ij}(k+1) = v_{ij}(k) + \Delta v = v_{ij}(k) - \eta\left(\frac{\partial E_2(k)}{\partial v(k)}\right)$$

$$= v_{ij}(k) + \eta e(k)\mu_{ij}(x_i(k))$$

Type-2 Neo-Fuzzy Neural Network



Type-2 Gaussian membership function

$$\mu_{ij}(x_i) = \exp\left(\frac{-(x_i - c_{ij})^2}{2\sigma_{ij}^{\ 2}}\right) = \begin{cases} \bar{\mu}_{ij} \ \text{за} \ \sigma_{ij} = \bar{\sigma}_{ij} \\ \underline{\mu}_{ij} \ \text{за} \ \sigma_{ij} = \underline{\sigma}_{ij} \end{cases}$$

$$\mu_{ij}^{\ *}(x_i) = \begin{cases} \bar{\mu}_{ij}^{*} = \prod_{i=1}^{n} \bar{\mu}_{ij} \\ \underline{\mu}_{ij}^{*} = \prod_{i=1}^{n} \underline{\mu}_{ij} \end{cases}$$

Calculation of the model output:

$$\hat{y}_M(k+j) = \frac{1}{2}\left(\frac{\sum_{i=1}^{q} \bar{f}_y^{(i)}(k+j)\bar{\mu}_y^{(i)}(k+j)}{\sum_{i=1}^{q} \bar{\mu}_y^{(i)}(k+j)} + \frac{\sum_{i=1}^{q} \underline{f}_y^{(i)}(k+j)\underline{\mu}_y^{(i)}(k+j)}{\sum_{i=1}^{q} \underline{\mu}_y^{(i)}(k+j)} \right)$$

Advantages: can estimate accurately in presences of uncertain input variations!

Disadvantages: greater number of parameters for adjustment!

Intuitionistic Neo-Fuzzy Neural Network

Particularities:

- We need to define the degree of membership μ(x) :

$$\mu_{ij}(x_i) = \exp\left(\frac{-(x_i - c_{ij})^2}{2\sigma_{ij}^{\ 2}}\right)$$

- We need to define the degree of membership ν(x):

$$\nu_{ij}(x_i) = \left(1 - \exp\left(\frac{-(x_i - c_{ij})^2}{2\sigma_{ij}^{\ 2}}\right)\right)^k, k \geq 1$$

- The hesitation bound is defined as:

$$\pi(x) = 1 - \mu(x) - \nu(x)$$

Calculation of the network output:

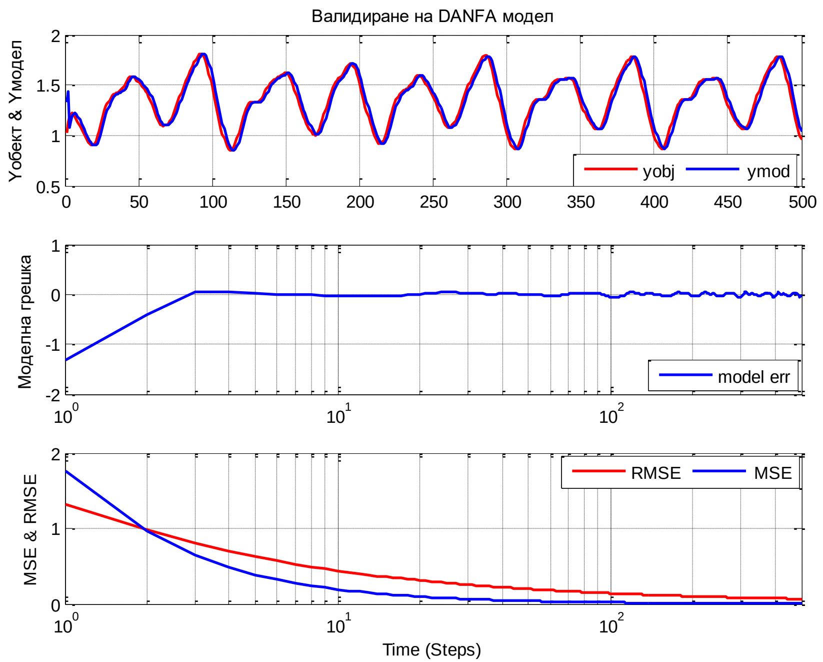
$$\hat{y}_M(k+1) = (1 - \pi(x))y_\mu + \pi(x)y_\nu$$

Where y_μ and y_ν are:

$$y_\mu(k+j) = \frac{\sum_{i=1}^{q} f_\mu^{(i)}(k+j)\mu_y^{(i)}(k+j)}{\sum_{i=1}^{q} \mu_y^{(i)}(k+j)}$$

$$y_\nu(k+j) = \frac{\sum_{i=1}^{q} f_\nu^{(i)}(k+j)\nu_y^{(i)}(k+j)}{\sum_{i=1}^{q} \nu_y^{(i)}(k+j)}$$

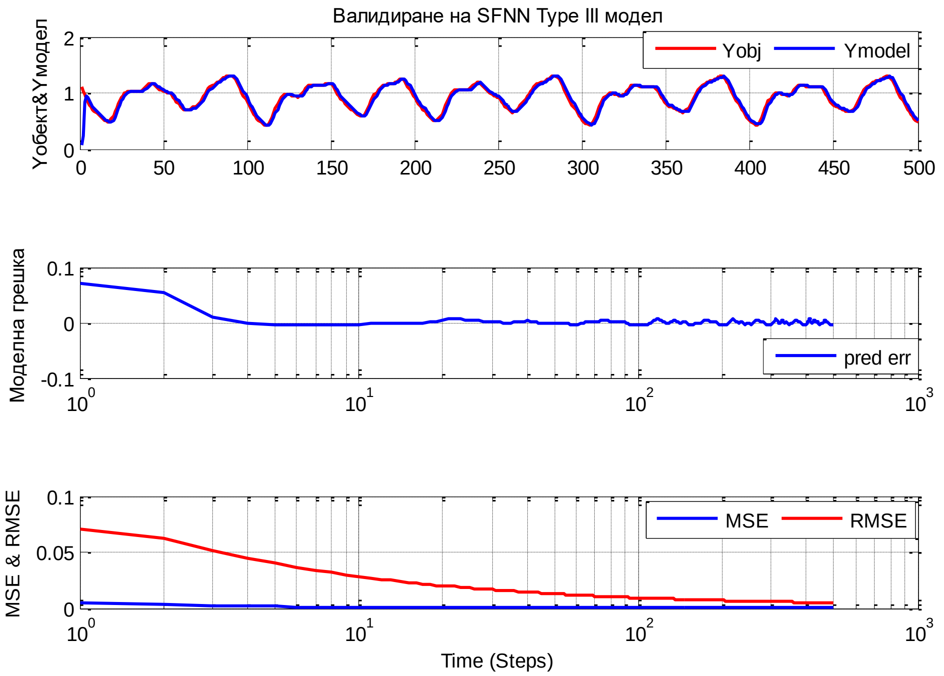
Modeling of chaotic time series with DANFA



DANFA model when modeling Mackey-Glass chaotic time series

steps	DANFA model		Classical TS	
	Model_err	MSE	Model_err	MSE
50	-0.019613	0.053553	-0.04702	0.0091595
100	-0.067288	0.027756	-0.15122	0.0077459
150	-0.01297	0.01893	-0.051306	0.0079593
200	-0.058875	0.014637	-0.12744	0.077992
250	-0.02409	0.012033	-0.051261	0.0077583
300	-0.04841	0.010343	-0.12544	0.0079122
350	-0.039738	0.0090874	-0.10179	0.0077017
400	-0.045243	0.0081819	-0.11864	0.007838
450	-0.03445	0.0074437	-0.095164	0.007659
500	-0.040763	0.0068844	-0.11223	0.007769

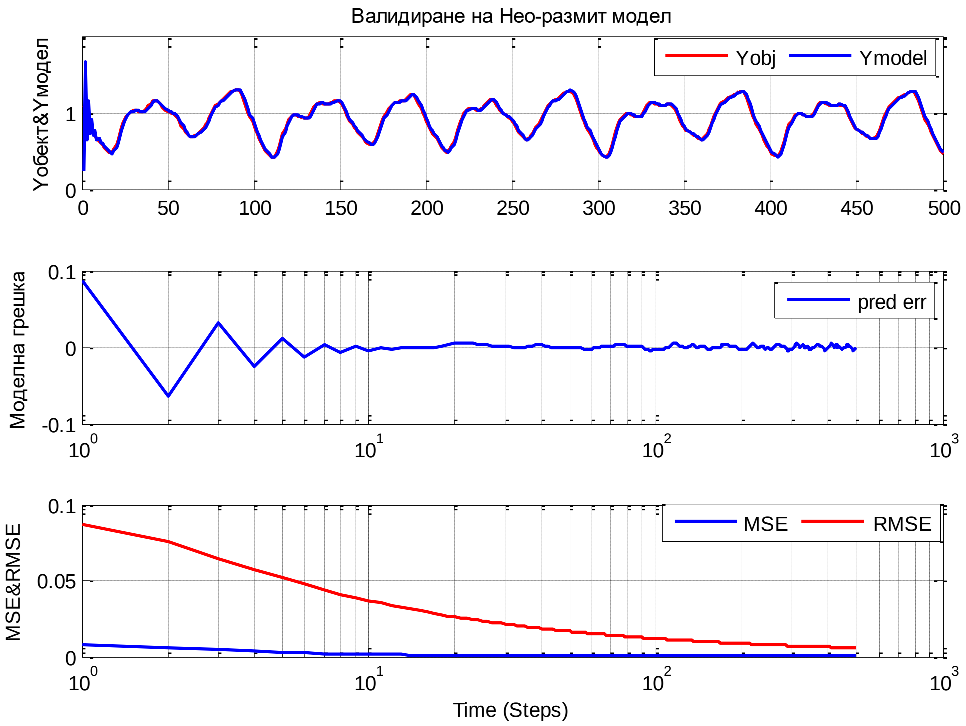
Modeling of chaotic time series with SFNN



SFNN Type III modeling Mackey-Glass chaotic time series

steps	SFNN Type I модел	SFNN Type II модел	SFNN Type III модел
	MSE	MSE	MSE
50	2,4e^{-4}	2,9e^{-4}	1,6e^{-4}
100	1,09e^{-4}	1,2e^{-4}	8,7e^{-5}
150	7,6e^{-5}	8,5e^{-5}	6,04e^{-5}
200	5,9e^{-5}	6,6e^{-5}	4,8e^{-5}
250	4,9e^{-5}	5,4e^{-5}	3,97e^{-5}
300	4,2e^{-5}	4,7e^{-5}	3,5e^{-5}
350	3,7e^{-5}	4,1e^{-5}	3,1e^{-5}
400	3,4e^{-5}	3,7e^{-5}	2,8e^{-5}
450	3,1e^{-5}	3,4e^{-5}	2,6e^{-5}
500	2,8e^{-5}	3,2e^{-5}	2,3e^{-5}

Modeling of chaotic time series with Type-2 NEO-FN



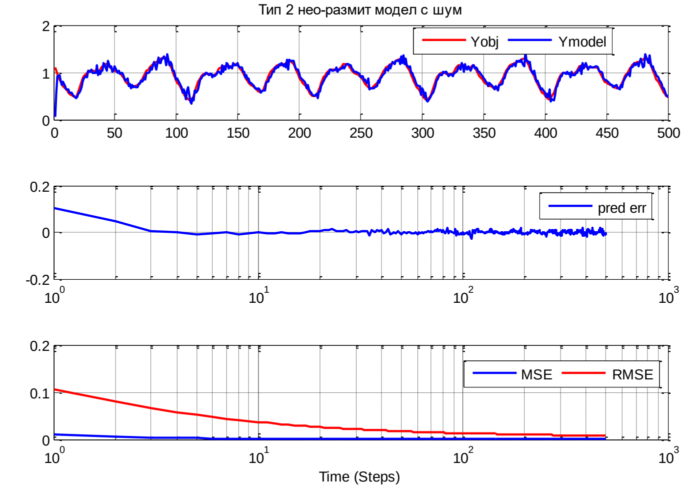
Type-2 Neo-Fuzzy model estimating Mackey-Glass Chaotic time series.

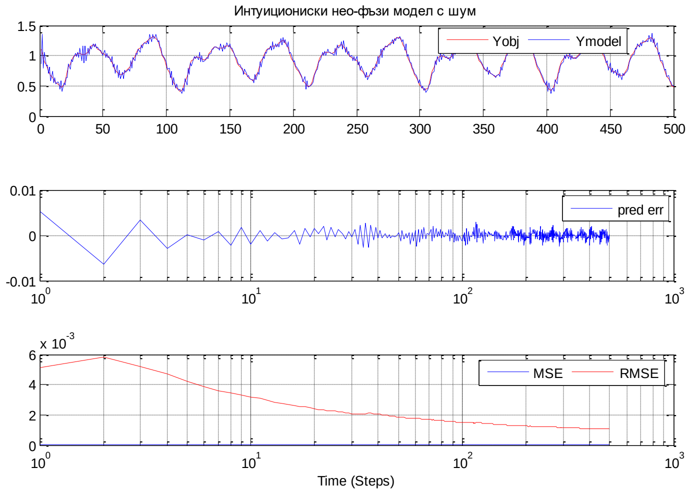
steps	NEO-fuzzy without MF adjustment	NEO-fuzzy with MF adjustment
	MSE	MSE
50	$9{,}5e^{-5}$	$9{,}35e^{-5}$
100	$7{,}3e^{-5}$	$7{,}06e^{-5}$
150	$5{,}9e^{-5}$	$5{,}6e^{-5}$
200	$4{,}8e^{-5}$	$4{,}5e^{-5}$
250	$4{,}1e^{-5}$	$3{,}9e^{-5}$
300	$3{,}5e^{-5}$	$3{,}4e^{-5}$
350	$3{,}2e^{-5}$	$3{,}12e^{-5}$
400	$2{,}9e^{-5}$	$2{,}92e^{-5}$
450	$2{,}5e^{-5}$	$2{,}33e^{-5}$
500	$2{,}25e^{-5}$	$2{,}17e^{-5}$

Comparison between the proposed FN networks

Стъпки	**DANFA model**				**SFNN Type III**				**Type-2 Neo-fuzzy with MF adjustment**				**Type-2 Neo-fuzzy without MF adjustment**				**Classical TS model**			
	N	P	MSE	t[**ms**]	N	P	MSE	t[**ms**]	N	P	MSE	t[**ms**]	N	P	MSE	t[**ms**]	N	P	MSE	t[**ms**]
50	18	126	5,3e^{-2}	1,83	9	63	1,6e^{-4}	1,58	12	36	9,35e^{-5}	1,67	12	12	9,5e^{-5}	1,14	81	1053	9,2e^{-2}	2,82
100			2,7e^{-2}				8,7e^{-5}				7,06e^{-5}				7,3e^{-5}				7,7e^{-2}	
150			1,9e^{-2}				6,04e^{-5}				5,6e^{-5}				5,9e^{-5}				7,9e^{-2}	
200			1,4e^{-2}				4,8e^{-5}				4,5e^{-5}				4,8e^{-5}				7,8e^{-2}	
250			1,2e^{-2}				3,97e^{-5}				3,9e^{-5}				4,1e^{-5}				7,6e^{-2}	
300			1,03e^{-2}				3,5e^{-5}				3,4e^{-5}				3,5e^{-5}				7,65e^{-2}	
350			9,1e^{-3}				3,1e^{-5}				3,12e^{-5}				3,2e^{-5}				7,62e^{-2}	
400			8,2e^{-3}				2,8e^{-5}				2,92e^{-5}				2,9e^{-5}				9,59e^{-3}	
450			7,4e^{-3}				2,6e^{-5}				2,33e^{-5}				2,5e^{-5}				8,47e^{-3}	
500			6,9e^{-3}				2,3e^{-5}				2,17e^{-5}				2,25e^{-5}				7,56e^{-3}	

Comparison between the proposed Type-2 and Intuitionistic models





steps	Type-2 Neo-fuzzy network		INFN	
	With noise	Without noise	With noise	Without noise
	MSE	MSE	MSE	MSE
50	$2.38e^{-4}$	$2.98e^{-4}$	$1.26e^{-6}$	$3.42e^{-6}$
100	$1.26e^{-4}$	$1.69e^{-4}$	$7.05e^{-7}$	$2.25e^{-6}$
150	$8.85e^{-5}$	$1.31e^{-4}$	$5.19e^{-7}$	$1.87e^{-6}$
200	$7.01e^{-5}$	$1.07e^{-4}$	$4.32e^{-7}$	$1.61e^{-6}$
250	$5.89e^{-5}$	$8.4e^{-5}$	$3.75e^{-7}$	$1.48e^{-6}$
300	$5.19e^{-5}$	$7.22e^{-5}$	$3.41e^{-7}$	$1.37e^{-6}$
350	$4.63e^{-5}$	$6.72e^{-5}$	$3.13e^{-7}$	$1.26e^{-6}$
400	$4.26e^{-5}$	$5.91e^{-5}$	$2.96e^{-7}$	$1.19e^{-6}$
450	$3.92e^{-5}$	$5.33e^{-5}$	$2.78e^{-7}$	$1.13e^{-6}$
500	$3.69e^{-5}$	$4.76e^{-5}$	$2.67e^{-7}$	$1.09e^{-6}$

The needed time for performing all the needed calculations:
IFNN: **1.48 ms** noisless case, **1.50 ms** noisy case.
Type-2 NFN **1.93 ms** noiseless case, **1.96 ms** noisy case.

Thank you for your attention!