Adaptive Learning Machines

Dr Varun Ojha

Department of Computer Science,
University of Reading

April 2019

Part I: Introduction

Machine Learning Tasks

Part II: Heterogeneous flexible neural tree (HFNT)

Neural tree construction

Neural tree performance on benchmark data

Neural tree performance on pharmaceutical application

Input feature analysis

Part III: Hierarchical fuzzy inference tree (HFIT)

Fuzzy tree construction

Fuzzy tree performance on benchmark data

Fuzzy tree performance on pharmaceutical application

Part IV: Conclusions

Task I: Function approximation

Function $f(\mathbf{x}, \mathbf{w}^*)$

x input vector
w unknown parameter vector

Goal:

- To find an appropriate function
- To find unknown parameter

Such that a cost function [commonly an error function such as Root Mean Squared Error (**RMSE** or **Error rate**)] is reduced.

Task II: Feature selection

Feature set $Z = \{z_1, z_2, \ldots, z_p\}$

z_i i-th feature

Goal:
To find an appropriate set of features Z^*.

Such that a cost function (commonly an error function such as RMSE/Error rate) is reduced.

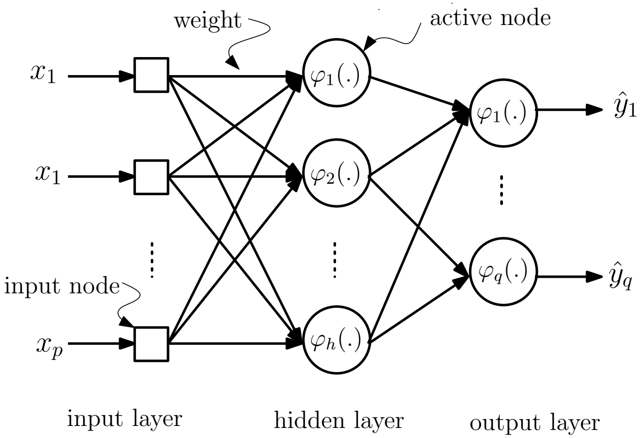
Machine learning objectives

- Create a model to fit data
- Optimize model for effectiveness:
  - to adapt the topology and learning parameters,
    which could lead to a low approximation error and a less complex model.
- Enabling adaptation:
  - simultaneous feature selection and function approximation.
- Validate the models:
  - select benchmark datasets and two industrial problems.

Feedforward neural network (FNN)

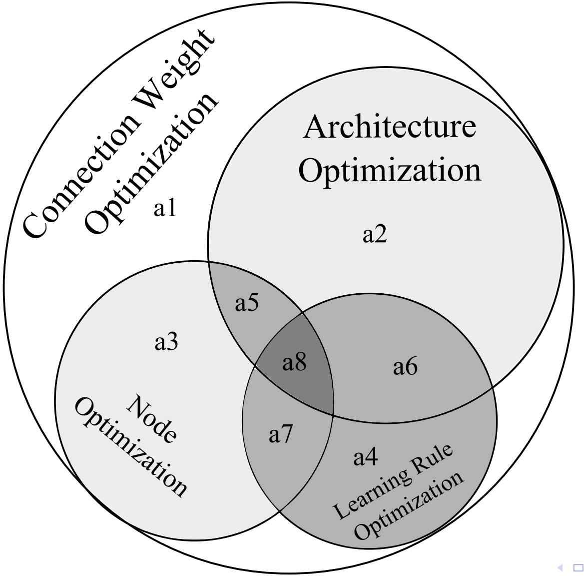
FNN components:

- Weights
- Architecture
- Activation functions
- Learning algorithms
- Inputs

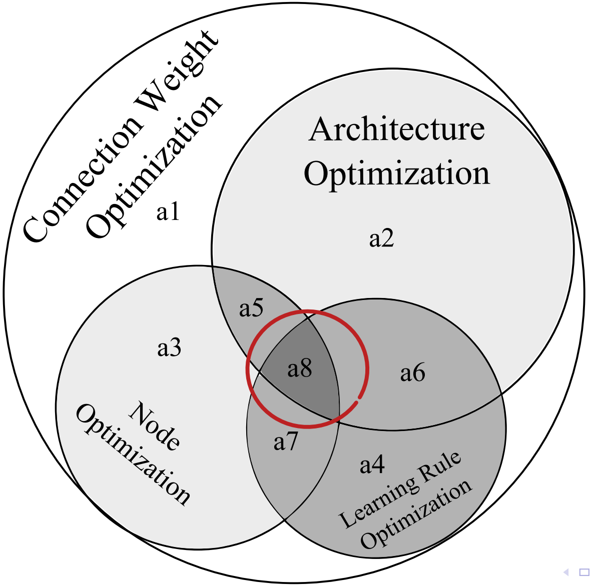


Three layered feedforward neural network (FNN)

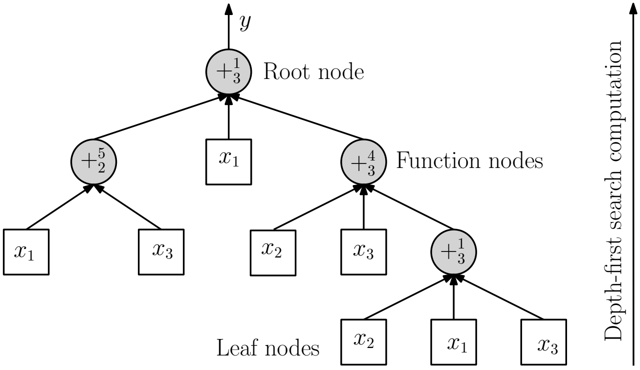
Neural network optimization spectrum



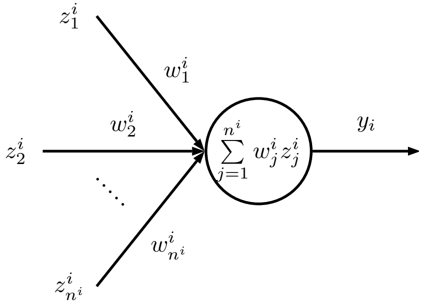
Neural network optimization spectrum



Heterogeneous Flexible (Adaptive) Neural Trees (HFNT)



Typical tree-like structure



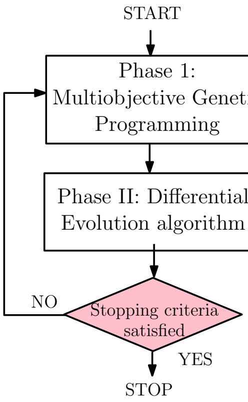
Typical computational [neural] node

Neural tree construction: Two-phase learning

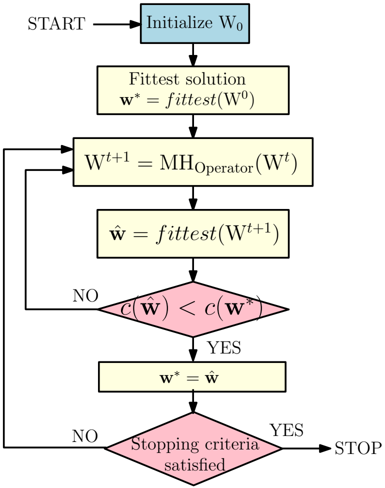
Structure learning
Algorithms: Multiobjective genetic programming
Operators: (Crossover and mutation)
Objectives: Tree size, Approximation error, diversity.

Parameter tuning
Differential evolution (or any other meta-heuristic algorithm)

Tree construction: Two-phase learning

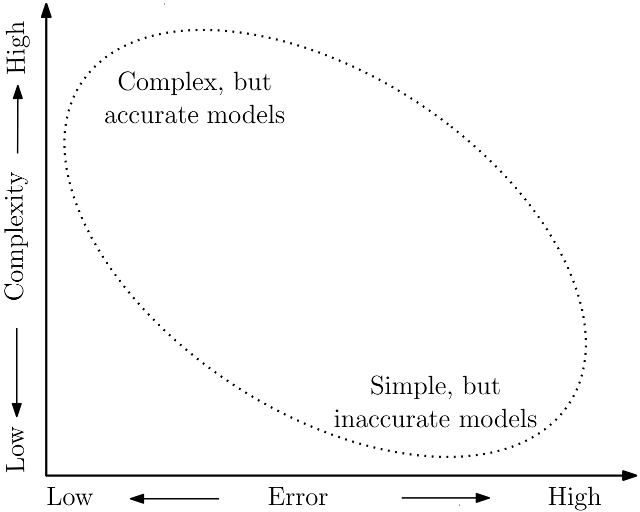


General two phase optimization

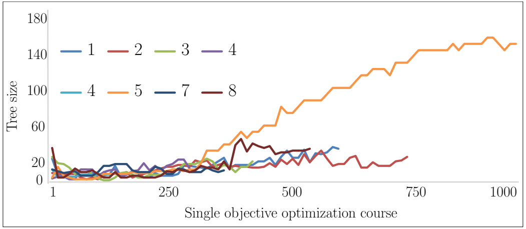


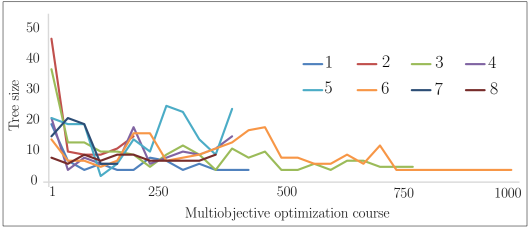
Metaheusitic basic framework

Advantage of multiobjective (1)



Advantage of multiobjective (2)





Single objective optimization (average tree size 39.265 nodes)

Multiobjective optimization (average tree size 10.25 nodes)

Neural tree (HFNT) performance evaluation

- Classification problems (five datasets).
- Regression problems (five datasets).
- Time-series problems (two datasets).
- Industrial Case Study: pharmaceutical die-filling.

Neural tree (HFNT) performance: classification

Friedman ranking test results over five data sets

Algorithm	Ranking
HFNT^{M}	1.0
HDT	2.5
FNT	2.5

Holm's post-hoc test results ($\alpha = 0.05$)

i	algorithm	z	p	α/i	Hypothesis
2	HDT	2.12132	0.033895	0.05	rejected
1	FNT	2.12132	0.033895	0.1	rejected

Neural tree (HFNT) performance: regression

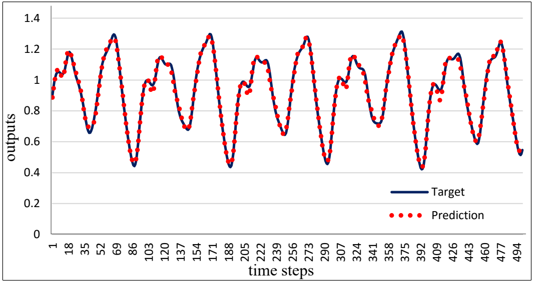
Friedman ranking test results over five data sets

Algorithm	Ranking
HFNT^M	1.5
METSK-HD^e	2.75
LEL-TSK	3.25
LINEAR-LSM	3.5
MLP	4.5
ANFIS-SUB	5.5

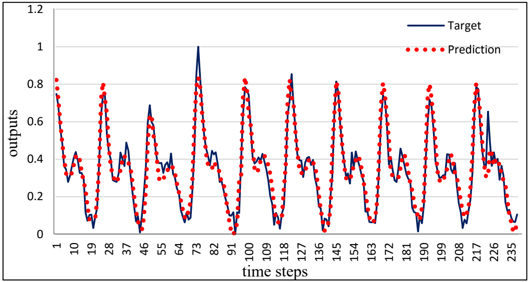
Holm's post-hoc test results ($\alpha = 0.1$)

i	algorithm	z	p	α/i	Hypothesis
5	ANFIS-SUB	3.023716	0.002497	0.02	rejected
4	MLP	2.267787	0.023342	0.025	rejected
3	LINEAR-LSM	1.511858	0.13057	0.033	
2	LEL-TSK	1.322876	0.185877	0.05	
1	METSK-HD^e	0.944911	0.344704	0.1	

Neural tree (HFNT) performance: time-series

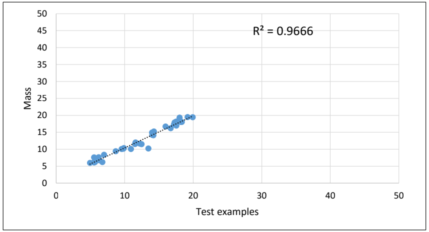


Mackey glass time series (correlation coefficient $r = 0.99$)

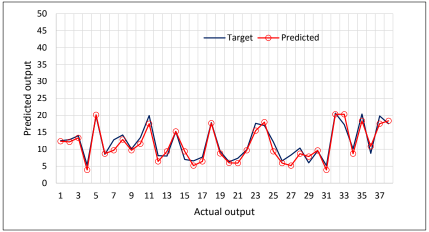


Waste water time series (correlation coefficient $r = 0.99$)

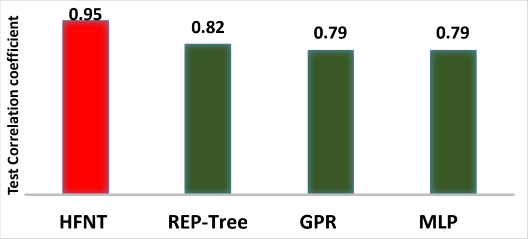
Neural tree performance: pharmaceutical die-filling



Target and predicted outputs)



Target versus predicted outputs



Performance of Neural tree (HFNT) compared to other predictors

Input feature analysis

Two parameters:

- Selection rate R_j: the rate of selection of an input feature set $Z_j \in \mathbf{Z}$ within a total of M experiments.
  Selection rate: $0 \leq R_j \leq 1$.

- Predictability score P_j: the predictability of an input feature set $Z_j \in \mathbf{Z}$ within a total of M experiments.
  Predictability score: $0 \leq P_j \leq 1$.

Feature analysis results (die-filling problem)

Selection rate and Predictability score of individual features

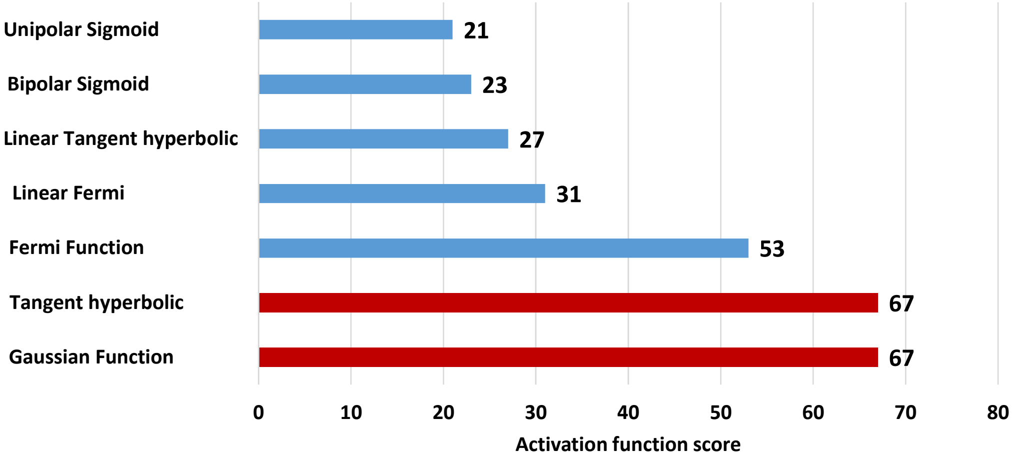
#	Input Features set	Selection Rate	Predictability Score
1	Z_1= True density	0.55173	0.541356
2	Z_2= d50	0.62069	0.586262
3	Z_3= Granule size	1	1
4	Z_4= Shoe speed	0.86207	0.92563

Feature analysis results (die-filling problem)

Selection rate and Predictability score of input feature sets

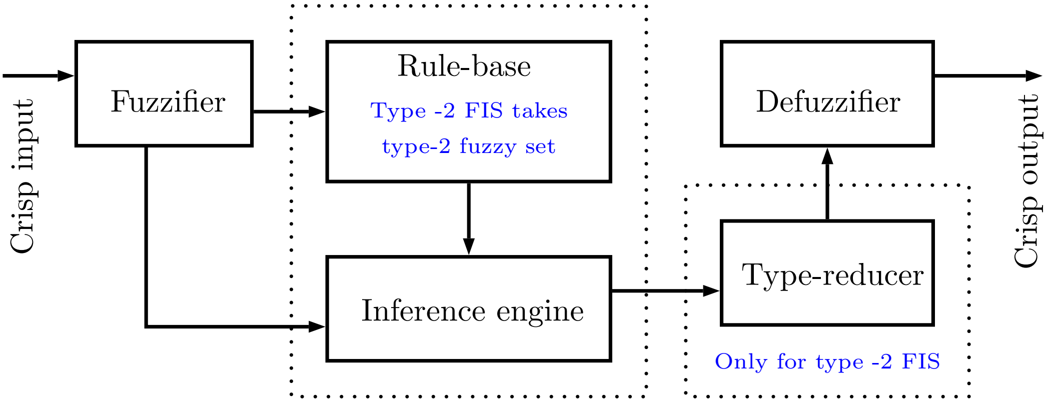
#	Input Feature set	Selection Rate	Predictability Score
1	Z_1= True density, d50, Granule size, Shoe speed	0.31035	0.969497
2	Z_2= d50, Granule size, Shoe Speed	0.17242	0.941601
3	Z_3= True density, Granule size, Shoe speed	0.13793	1
4	Z_4= Granule size, Shoe speed	0.24138	0.979663
5	Z_5= True density, d50, Granule size	0.10345	0.493741
6	Z_6= d50, Granule size	0.03448	0.470451

Activation function scores



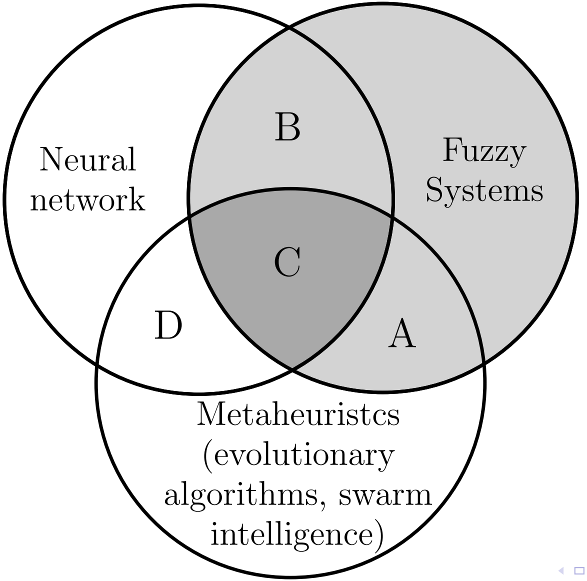
Score 67 is the best performance and score 21 is the worst performance

Fuzzy inference system (FIS)

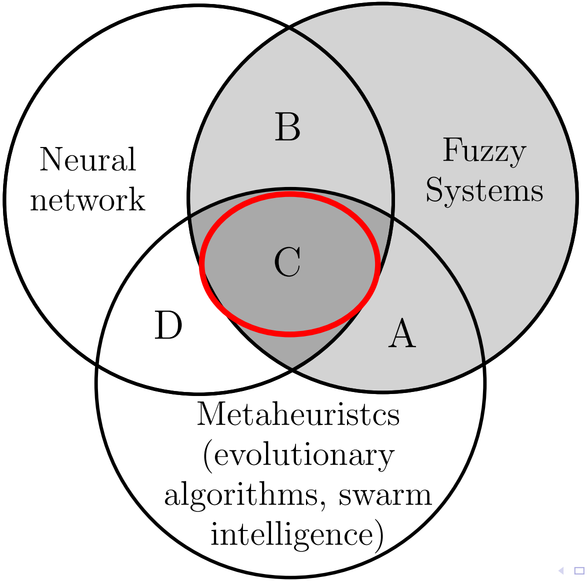


Typical fuzzy inference system (FIS)

Fuzzy Inference Systems optimization spectrum



Fuzzy Inference Systems optimization spectrum



Fuzzy rules

IF–THEN rule of the form:

$$R^i : \text{IF } x_1 \text{ is } A_1^i \text{ AND } \ldots \text{AND } x_{p^i} \text{is } A_{p^i}^i \text{THEN } y^i \text{ is } B^i$$

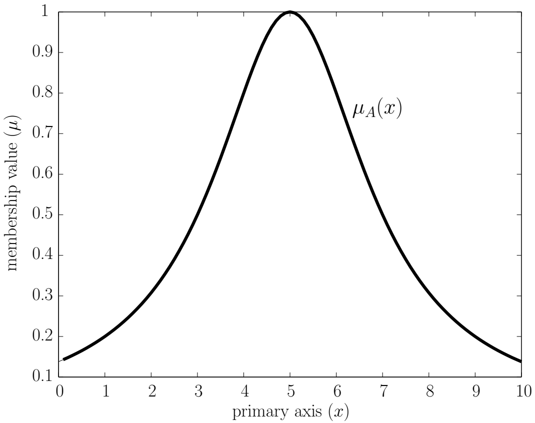
Function B^i for type-1:

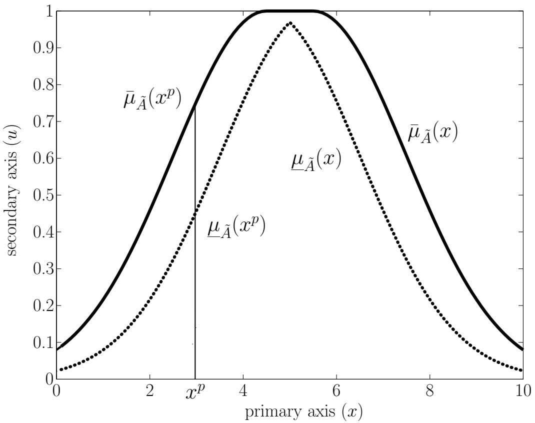
$$B^i = c_0^i + \sum_{j=1}^{p^i} c_j^i x_j, \tag{1}$$

Function B^i for type-2:

$$B^i = [c_0^i - s_0^i, c_0^i + s_0^i] + \sum_{j=1}^{p^i} [c_j^i - s_j^i, c_j^i + s_j^i] x_j, \tag{2}$$

Type-1 and Type-2 membership function

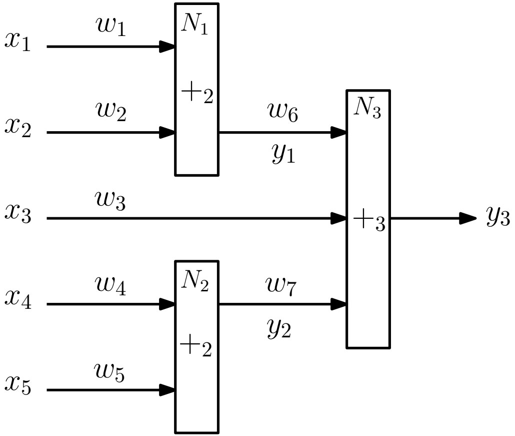




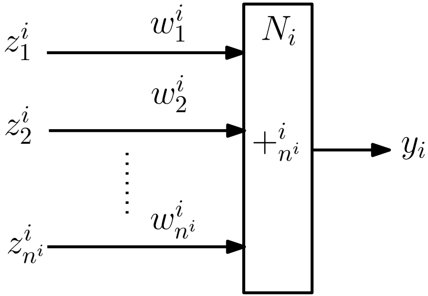
Type-1 membership function

Interval Type-2 membership function

Hierarchical fuzzy inference tree construction (HFIT)



Typical fuzzy inference tree



Typical computational [fuzzy] node

Rule formation at a node

Rules at a node:

- A node receives n inputs.
- Each input is allowed to have 2 fuzzy sets.
- Maximum rule formed at a node is 2^n.

Fuzzy tree (HFIT) performance evaluation

1. Example 1: System identification
2. Example 2: Mackey-glass time series
3. Example 3: Abalone age prediction
4. Example 4: Box-Jenkins gas furnace
5. Example 5: PLGA dissolution rate prediction (Industrial Case Study)

Fuzzy tree-HFIT (Type-1 Fuzzy tree) Performance

Example 1		Example 2		Example 3		Example 4	
Algorithm	RMSE	Algorithm	RMSE	Algorithm	RMSE	Algorithm	RMSE
SaFIN	0.012	NNT1FW	0.055	HS	3.16	T1-NFS	0.4074
SONFIN	0.0085	AFRS	0.0256	General	3.15	GNN-1	0.3114
T1HFIT^S	0.0043	IFRS	0.0253	CCL	2.65	GNN-2	0.2983
T1HFIT^M	0.0041	HTS-FS1	0.0129	Chen	2.59	**T1HFIT^S**	0.2455
		HTS-FS2	0.0151	**T1HFIT^S**	2.126	**T1HFIT^M**	0.2838
		RBF-AFA	0.0128	**T1HFIT^M**	2.348		
		HyFIS	0.01				
		D-FNN	0.008				
		SuPFuNIS	0.0057				
		T1HFIT^S	0.0122				
		T1HFIT^M	0.0119				

Fuzzy tree-HFIT (Type-2 fuzzy tree) performance

Example 1		Example 2		Example 3		Example 4	
Algorithm	RMSE	Algorithm	RMSE	Algorithm	RMSE*	Algorithm	RMSE
T2TSKFNS	0.0324	T2FLS	0.043	RIT2NFS-WB	2.4047	SEIT2FNN	0.269
T2FNN	0.0281	T2FLS (TSK)	0.043	McIT2FIS-UM	2.3481	RIT2NFS-WB	0.353
SIT2FNN	0.0241	NNT2FW	0.039	SEIT2FNN	2.3388	McIT2FIS-UM	0.314
RIT2NFS-WB	0.0151	SEIT2FNN1	0.003	McIT2FIS-US	2.3357	McIT2FIS-US	0.318
MRI2NFS	0.0051	SEIT2FNN2	0.005	**T2HFIT^S**	2.1154	**T2HFIT^S**	0.277
T2FLS-G	0.0379	**T2HFIT^S**	0.009	**T2HFIT^M**	2.1275	**T2HFIT^M**	0.284
SEIT2FNN	0.0022	**T2HFIT^M**	0.006				
T2HFIT^S	0.0034						
T2HFIT^M	0.0028						

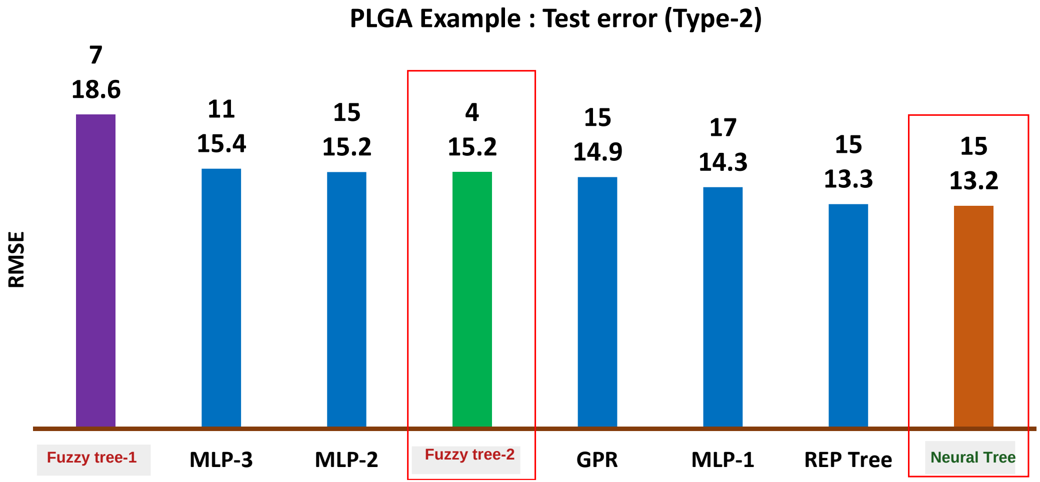
*training error

Average performance of Fuzzy tree (HFIT) versions

Average performance of hierarchical fuzzy inference tree

	Fuzzy tree-1		Fuzzy tree-2	
	Error	Size	Error	Size
	Single Objective			
Average	**1.078**	**95.5**	**1.014**	**210.7**
	Multibjective			
Average	**1.086**	**55.1**	**0.997**	**148.9**

Fuzzy tree (HFIT) performance: PLGA dissolution rate prediction



Comparison between Neural tree and Fuzzy tree

- **Approximation ability**: Neural tree is better than Fuzzy tree.
- **Feature selection ability**: Fuzzy tree is better than Neural tree.
- **Model size**: Fuzzy tree is lighter than Neural tree.
- **Interpretability**: Fuzzy tree is interpretable and Neural tree is not.

Conclusions

- Two computational models were developed for the simultaneous feature selection and function approximation and adaptive learning.
- Performance analysis on of the Neural tree and Fuzzy tree models on benchmark datasets reveled the proposed model's competitiveness with models chosen for comparison.
- Fuzzy tree and neural tree models are offer good results for the real-world industrial problems (die-filing performance and PLGA drug dissolution prediction).

Thank You!

References:

J1. **Ojha, V. K.**, Abraham, A., and Snášel, V. (2019). Heuristic Design of Fuzzy Systems : A Review of Three Decades of Research, *Engineering Applications in Artificial Intelligence*.

J2. **Ojha, V. K.**, Abraham, A., and Snášel, V. (2018). Multiobjective Programming for Type II Hierarchical Fuzzy Trees, *IEEE Transaction on Fuzzy Systems*. Under review (**IF:** 8.75)

J3. **Ojha, V. K.**, Schiano, S., Wu, C.Y., Snášel, V., and Abraham, A. (2018) Predictive Modeling of the die filling process of the pharmaceutical granules using Flexible Neural Tree. *Neural Computing Application*. (Accepted) (**IF:** 1.57)

J4. **Ojha, V. K.**, Abraham, A., and Snášel, V. (2017). Metaheuristic Design of Feedforward Neural Networks: A Review of Two Decades of Research, *Engineering Applications in Artificial Intelligence*.

J5. **Ojha, V. K.**, Abraham, A., and Snášel, V. (2016). Ensemble of Heterogeneous Flexible Neural Trees Using Multiobjective Genetic Programming, *Applied Soft Computing*. (Accepted) (**IF:** 2.81)