Supplementary Materials

S1 Baseline datasets

28 real regression data sets with varying numbers of features (2 to 40) and sample sizes (337 to 40,768) are used as baseline datasets. According to the feature dimension, the datasets are divided into two categories: 11 low-dimensional data sets and 17 high-dimensional data sets. TABLE 1 and TABLE 2 show the detailed information of the above two datasets, respectively, including the names, abbreviations, feature numbers, and sample sizes.

TABLE S1. The low-dimensional data sets

Data sets	Abbreviation	No. of features	No. of samples
Electrical Length	ELE1	2	495
Plastic Strength	PLA	2	1650
Quake	QUA	3	2178
Electrical Maintenance	ELE2	4	1056
Friedman	FRIE	5	1200
Auto MPG6	MPG6	5	398
Delta Ailerons	DELAIL	5	7129
Daily Electricity Energy	DEE	6	365
Delta Elevators	DELELV	6	9517
Analcat	ANA	7	4052
Auto MPG8	MPG8	7	398

TABLE S2. The high-dimensional data sets

Data sets	Abbreviation	No. of features	No. of samples
Abalone	ABA	8	4177
California Housing	CAL	8	20640
Concrete Concessive Strength	CON	8	1030
Stock prices	STP	9	950
Weather Ankara	WAN	9	1609
Weather Izmir	WIZ	9	1461
MV Artificial Domain	MV	10	40768
Forest Fires	FOR	12	517
Mortgage	MOR	15	1049
Treasury	TRE	5	1049
Baseball	BAS	16	337
House-16H	HOU	16	22784
Elevators	ELV	18	16559
Computer Activity	CA	21	8192
Pole Telecommunications	POLE	26	14998
Pumadyn	PUM	32	8192
Ailerons	AIL	40	13750

S2 FNNR-M

The FNNR using Mamdani fuzzy rules is illustrated in Figure 1.

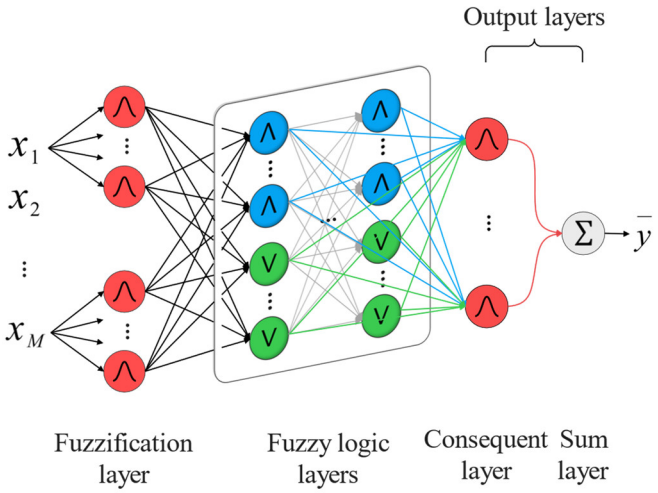

Figure S1. The structure of the FNNR-M

It can be observed that the output layers of FNNR-M consist of one consequent layer and one sum layer. The nodes of the consequent layer represent the fuzzy sets of the output variable, and the number of nodes represents the fuzzy partition number of the output variable, which is set to H_M. The parameters of the consequent layer are continuous real values in the interval [0,1], representing the weight of each rule. The consequent layer is fully connected with the fuzzy logic layers, and the output of each consequent layer node is the weighted sum of firing strengths of the rules whose consequent is the corresponding fuzzy set. The output of the sum layer is shown in Equation (1):

$$\bar{y} = \frac{\sum_i \hat{c}_i u_i}{\sum_i u_i} \tag{1}$$

where $\hat{c}_i$ is the parameter of the edge connecting the node of the sum layer and the i^{th} node of the consequent layer, which represents the center of the consequent fuzzy set, that is, the mean value of the Gaussian membership functions (MFs). u_i is the output of the i^{th} consequent layer. Mamdani fuzzy rules can be directly extracted from the trained FNNR-M. For the FNNR-M, the parameter scale of the output layers is $H_M \cdot K$, where K is the number of nodes in fuzzy logic layers.

S3 FNNR-F

Replace the output layers of the FNNR with fully connected layers, and the model after the replacement is called the FNNR-F.

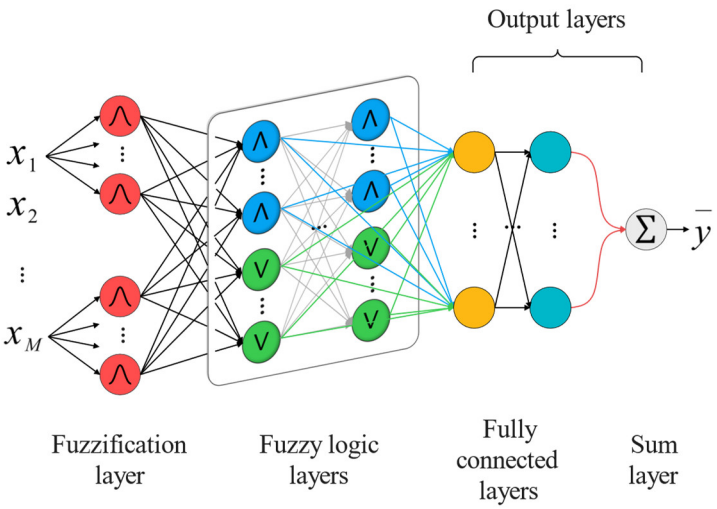

Figure S2. The structure of the FNNR-F

In the FNNR-F, the output layers are composed of the fully connected layers and one sum layer, where the fully connected layers are fully connected with fuzzy logic layers. The layer number of fully connected layers is set as $I_F (I_F \geq 1)$, and the number of nodes of each layer is set as n_F. The ReLU function is utilized as the activation function. The parameter scale of the output layers is $I_F \cdot (n_F + 1) + n_F \cdot K$.

S4 Time complexity analysis of the three models

Figure 3 reveals the normalized training time of the three models. In the figure, before and after the dotted line are the training time of each model on low- dimensional and high-dimensional data sets, respectively. It can be observed that on most low-dimensional data sets, there are little differences in the time consumptions of the three models, but generally, the training time of FNNR-F is relatively longer (the FNNR-F spends the longest time on 6 of 11 data sets). This is because for the FNNR-F, the number of parameters is large and the complexity is high. The training time of the FNNR-T is advantageous on data sets with relatively large sample sizes like DELAIL, DELELV, and ANA.

On high-dimensional data sets, the FNNR-T has an advantage in time consumption on data sets with relatively large sample sizes like CAL and MV, while the consumed time of the FNNR-M is long. As the number of features increases, the time consuming of the FNNR-T also increases, which is the result of the positive correlation between the model complexity of the FNNR-T and its feature dimension. The running time of FNNR-M is overall middle-ranking. The FNNR-F takes the longest time on most data sets, and the time on datasets with large features and large sample sizes is slightly shorter than that of the FNNR-T.

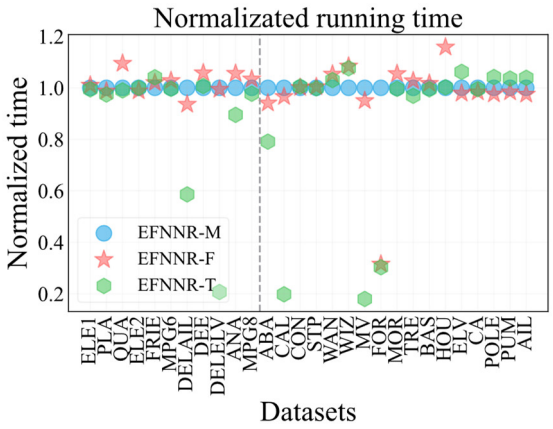

Figure S3. The normalized training time of the FNNR-M, FNNR-F, and FNNR-T on 28 data sets

S5 The experimental results of the three models on the ablation study of the alternate training strategy

TABLE III shows the average MSEs of the three models on 28 data sets using the alternate training strategy and the normal training method. To highlight the role of alternate training strategy, parameters of MFs and fuzzy partitions are fixed during the training, and the whole training is only divided into two stages: the stage of joint training and the stage of the fixed fuzzy logic layer. As can be observed from TABLE III, regardless of the model, on most data sets, the regression errors obtained by utilizing the alternate training strategy are lower (2% to 80%) than that obtained by using the normal training method.

TABLE S3. The average prediction errors of the three models with alternate training strategy and normal training method on 28 data sets

Data sets	FNNR-M		FNNR-F		FNNR-T	
	Alternate	Normal	Alternate	Normal	Alternate	Normal
ELE1	1.622	1.774	1.361	1.642	1.504	1.526
PLA	1.080	1.116	1.062	1.157	1.095	1.129
QUA	0.018	0.0190	0.0184	0.0189	0.0192	0.0193
ELE2	5882	8147	2585	12831	3922	7586
FRIE	0.608	0.699	0.682	0.784	0.605	0.630
MPG6	3.618	3.755	3.623	3.769	3.696	3.759
DELAIL	1.513	1.498	1.396	1.625	1.456	1.897
DEE	0.077	0.081	0.0767	0.0876	0.080	0.083
DELELV	1.017	1.021	0.928	1.012	1.015	1.042
ANA	0.003	0.004	0.003	0.005	0.004	0.004
MPG8	3.402	3.593	2.700	3.180	3.095	3.215
ABA	2.080	2.087	1.973	2.162	2.050	2.050
CAL	1.714	2.088	1.582	2.060	1.674	2.061
CON	17.08	27.54	16.375	42.13	15.42	21.88
STP	0.299	0.449	0.279	0.500	0.288	0.362
WAN	0.729	0.857	0.834	2.216	0.741	1.533

WIZ	0.697	0.743	0.669	0.828	0.655	0.708
MV	0.031	0.135	0.048	0.432	0.019	0.125
FOR	4018	4018	3998	4083	4010	4010
MOR	0.009	0.009	0.004	0.018	0.003	0.006
TRE	0.032	0.035	0.025	0.036	0.025	0.031
BAS	1.796	1.866	1.721	1.591	1.896	1.886
HOU	6.990	9.218	6.729	7.267	6.850	7.317
ELV	2.519	2.602	1.987	2.815	2.500	5.963
CA	3.769	3.838	3.432	4.739	2.766	4.125
POLE	36.16	58.62	8.788	82.42	25.11	46.11
PUM	0.326	1.027	0.157	1.447	0.198	0.680
AIL	1.342	1.338	1.309	1.364	1.303	1.399

Of course, in a few data sets, the regression MSEs of the model using the normal training method are not much different from that using the alternate training strategy, and on some data sets, the formers are even smaller, which are underlined in the table. This is because, under the alternate training strategy, different parameters are trained separately, and all kinds of parameters need more time to “run in” with each other to achieve the optimal solution. This means that although the alternate training strategy can help the model converge eventually, it also prolongs the time of convergence, while for the normal training method, the global optimal solution may be found in advance on occasion although it can cause the constant oscillation of parameters.