2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

A Model for Dynamic Knowledge Representation and Learning

Xinyuan Wang, Wenbiao Zhang, and Weilin Wang

Abstract— Knowledge of complex systems is often imprecise and subject to frequent modifications. Consequently, creating a knowledge representation and inference model that can adapt to changes in information is crucial. In this paper, we integrate the functional link into the fuzzy Petri net and propose a generalized model called functional link fuzzy Petri net (FLFPN). This model retains the explanatory ability of fuzzy Petri nets while acquiring the powerful learning ability of functional link neural networks. Finally, since the forming of congestion is highly sensitive to traffic situations in peak hours, we use FLFPN to predict traffic situations of an expressway ramp, which shows a significant improvement in the prediction accuracy, as compared with the traditional FPN model.

I. INTRODUCTION

For many years, researchers have developed knowledge bases that permit non-experts to solve complex problems that are typically solved only by experts [1]. In order to facilitate the processing of real-world data by computers, a variety of knowledge representation techniques have been created, including production rules and fuzzy Petri nets [2]. Fuzzy Petri Nets (FPNs) are Petri nets that have been adjusted to model fuzzy reasoning processes with propositional logic [3]. Based on fuzzy production rules (FPRs), FPN could deal with imprecise, vague or fuzzy information in complex systems [4][5][6].

The FPN's parameters are learnable, and there have been studies on enhancing its learning ability [7][8][9][10][11][12]. By training the parameters with historical data, it can more accurately make future decisions. Moreover, the prediction accuracy of an FPN is dependent on the rules to build it. The rules are proposed and formulated by experts and should not be changed. If we want to improve its learning ability further while keeping the rules fixed, we need to change its mathematical formula—its computational formula for transferring the logical value.

The FPN's reasoning process can be interpreted as a linear transfer of truth values for propositions. While the traditionalFPN model can effectively capture linear inputoutput relationships, it proves inadequate for dealing with complex nonlinear relationships. Learning these nonlinearities is difficult by relying solely on base parameters and expressions. To improve the situation, we need a model with better representation power while introducing few parameters and changes in structure as possible.

The main structure of the FPN should remain unaltered to ensure interpretability, given that it is defined by rules. With this constraint in mind, we introduce additional higher-order inputs similar to the functional link net (FLN). The functional link net methodology was initially introduced in [13][14], as a type of higher-order neural network boasting faster convergence and lower computational requirements than traditional neural networks. The FLN removes the hidden layer and adds higher-order units to the input layer to preserve its non-linearity. This grants it the impressive ability to capture non-linear relationships between inputs and outputs, while also resulting in a simpler structure that facilitates the determination and retention of network parameters. This effective structure is appropriate for our requirements, thus we propose a new model called functional link fuzzy Petri net (FLFPN). The introduction of higher-order terms can significantly expand the solution space, ultimately improving the original model's capacity to represent the data.

To test the improvement of FLFPN over traditional FPN, we apply both models to predict the traffic situations of a highway ramp during peak hours. The forming of traffic congestion is highly sensitive to traffic situations when the traffic volume approaches the capacity. The tests show that FLFPN achieves a one-third reduction in min square error with respect to the actual observations compared to traditional FPN in predictions.

The rest of the paper is organized as follows. In Section II, we give the detailed definition, the reasoning and learning algorithms of FLFPN. In Section III, to validate the model, we conduct experiments applying FLFPN to predict the traffic situations of an expressway ramp.

II. FUNCTIONAL LINK FUZZY PETRI NET

In this section, we first present the representation techniques of FLN and incorporate them into FPN by defining FLFPN. Then, we present the techniques for applying FLFPN for making decisions and training the model.

A. Functional Link Net

The FLN model is a single-layer neural network. It removes the hidden layer while maintaining nonlinearity, provided that the input layer has higher-order units. The

*This work is supported in part by PowerChina Grants KY2020-JT-12-01-2022 and HDY-CGHT20-20231262Y.

Xinyuan Wang and Weilin Wang (the corresponding author) are with the Department of Control Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093 merewxy15@gmail.com; wlwang@usst.edu.cn.

Wenbiao Zhang is with Huadong Engineering Corporation Limited, Hangzhou 311122 zhang_wb@hdec.com.

computational cost is shifted from the hidden layer to selecting a suitable input layer in FLNs [15][16].

There are different input patterns available for functional link nets, including random vectors, tensor representations, and function expansion representations [17]. According to [13], the use of second-order tensor representations proves to be effective; therefore, this paper will adopt the second-order tensor model as the input pattern for the simple storage of structural parameters.

For example, the tensor representation pattern of the inputs $\mathbf{x} = (x_1, x_2)$ can be $\left(1, x_1, x_2, x_1^2, x_1x_2, x_2^2\right)$, where the constant term is usually omitted. The extended input layer is equivalent to projecting the original $d-$dimensional input space into a new space $\mathbf{x} = (x_1, x_2, \ldots, x_d) \rightarrow (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_h(\mathbf{x}))$, where h is the new dimension that takes into account both the first and second order polynomial inputs.

We integrate the representation techniques of FLN into FPN and propose the FLFPN model as follows.

B. Definition of FLFPN

Definition 1: An FLFPN structure is defined as a 12-tuple:

$$FLFPN = (P, T, D, I, O, Th, W, CF, CD, \alpha, \beta, M) \quad (1)$$

where $P = \{p_1, p_2, \ldots, p_m\}$, $T = \{t_1, t_2, \ldots, t_n\}$ and $D = \{d_1, d_2, \ldots, d_m\}$ are finite sets of places, transitions, and proposition, with $|P| = |D|$. I and O are $m \times n$ incidence matrices defining the directed arcs from places to transitions. $Th = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$, $CF = [\mu_1, \mu_2, \ldots, \mu_n]^T$, and $M = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T$ are vectors denoting the thresholds, certainty factors, and markings. α and β are association functions. The former assigns a truth value to each place and the latter defines the mapping between each place and proposition. $W = \{W_1, W_2, \ldots, W_n\}$ is a set of weight matrices. W_i is a $(m+1) \times (m+1)$ upper triangular matrix, i.e. the weight matrix of transition t_i, and its element, w_{jk}, denotes the learned weight of the functional link term of $\alpha_j\alpha_k$, for $j = 1, 2, \ldots, m$, $k = 1, 2, \ldots, m$, $j \leq k$. Element in last column, $w_{j,m+1}$, denotes the weight of the first-order term α_j, for $j = 1, 2, \ldots, m$. CD is an $m \times n$ matrix, and the value of cd_{ij} shows the relative importance of an input p_i to its output t_j.

The FLFPN representation of a typical fuzzy production rule is shown in Fig. 1. To make the model more intuitive, instead of simply adding inputs to the input layer, we treat the functional link inputs as an additional hidden layer, which is denoted as the link layer. The projection from input layer to link layer is presented as $\boldsymbol{\alpha}_i \rightarrow (\phi_1(\boldsymbol{\alpha}_i), \phi_2(\boldsymbol{\alpha}_i), \ldots, \phi_h(\boldsymbol{\alpha}_i)) = \Phi_j(\boldsymbol{\alpha}_i)$, where $\boldsymbol{\alpha}_\mathbf{i}$ denotes the ith truth value set of input places, and $\Phi_j(\cdot)$ is the projection function of transition t_j. The dimension of the tensor-represented inputs is h, and the encapsulated unit $\phi_l(\boldsymbol{\alpha}_i)$ represents the lth functional link inputs generated by the initial inputs $\boldsymbol{\alpha}_i$, for $l = 1, 2, \ldots, h$.

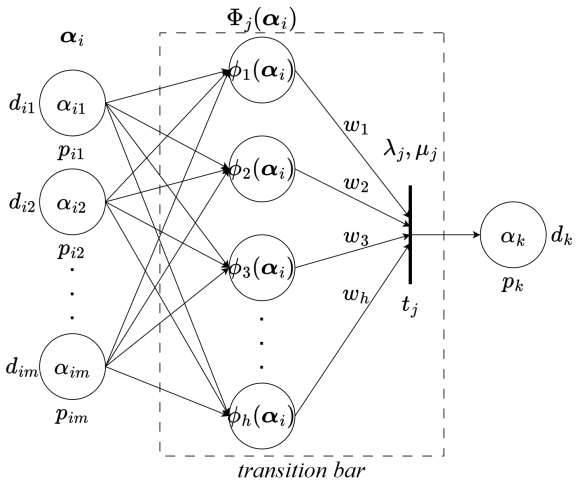

Fig. 1. Internal structure of FLFPN.

C. Fuzzy Reasoning Using FLFPN

The basic definitions are given to better explain the reasoning process with the FLFPN model. For a transition $t_k \in T$, let $I(k) = \{p_{I1}, p_{I2}, \ldots, p_{Id}\}$ be the initial input set and for the sake of simplicity, define $\mathbf{x}_k = [x_1, x_2, \ldots, x_d]^T = [\alpha_{I1}, \alpha_{I2}, \ldots, \alpha_{Id}]^T$. The expanded polynomial terms are represented as $\phi_{k1}(\mathbf{x}_k)$, $\phi_{k2}(\mathbf{x}_k), \ldots, \phi_{kh}(\mathbf{x}_k)$, and the corresponding weights are $w_{k1}, w_{kw}, \ldots, w_{kh}$. Let $O(k) = \{p_{O1}, p_{O2}, \ldots, p_{Ol}\}$ be the output set. The threshold value and certainty factor of this transition are λ_k and μ_k, respectively.

Definition 2: A transition $t_k \in T$ is enabled and fired if $\sum_{i=1}^{h} w_{ki}\phi_{ki}(\mathbf{x}_k) \geq \lambda_k$. Let $s_k = \sum_{i=1}^{h} w_i\phi_i(\mathbf{x})$ be the weighted summation.

A sigmoid function is used to approximate the threshold judgment. The sigmoid function is denoted by $\sigma(\cdot)$, and $\sigma(\beta x) = \frac{1}{1+e^{-\beta x}}$. We usually set β to a relatively large value to keep the slope of the function large so that it can act like a gating unit. So the output truth value can be represented by $f(s_k) = s_k\sigma[\beta(s_k - \lambda_k)]$.

If β is big enough, when $s_k > \lambda_k$, $f(s_k) \approx s_k$, and when $s_k < \lambda_k$, $f(s_k) \approx 0$. This approximation of the step function using a sigmoid function enables the smooth transition from a discrete to a continuous problem, simplifying our subsequent derivation.

Definition 3: After transition $t_k \in T$ is fired, the token in its input places will be transmitted to its output places $O(k)$, produced by its certainty factor. The new truth values of t_k's output places are $\alpha_{Oi} = \mu f(s_k) = \mu s_k\sigma[\beta(s_k - \lambda_k)]$. Let $G(s_k) = \mu f(s_k)$ denotes the final output value of a transition t_k. If a place has more than one input transitions fired, then its new real value is decided by the biggest truth value.

$\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ are all m-dimensional vector and a_i, b_i, c_i are their elements respectively. Then we define new operaters:

1) $\mathbf{a} : \mathbf{b} = \mathbf{c}$

where $\mathbf{c}$ is the concatenation of $\mathbf{a}$ and $\mathbf{b}$.

2) $\mathbf{a} \circ \mathbf{b} = \mathbf{c}$
where $c_i = a_i \cdot b_i$, for $i = 1, 2, \ldots, m$.

3) $\max(\mathbf{a}) = d$
where $d = \max_{1 \le i \le m} \{a_i\}$.

The reasoning process of FLFPN is shown in Algorithm 1.

Algorithm 1: Reasoning Algorithm of FLFPN

Input: P, T, I, O, Th, W, CF, and initial marking M_0
Output: Final marking M_k

Let $k \leftarrow 1$, where k means this is the kth iteration;
Let I_i be the ith column of I, and O_j be the jth row of O;
while $M_k \neq M_{k-1}$ do
  foreach $t_i \in T$, $i \in \mathbb{N}^+, 1 \le i \le n$ do
    $\mathbf{x}_i \leftarrow I_i \circ M_{k-1} : 1$;
    $s_i \leftarrow \mathbf{x}_i^T W_i \mathbf{x}_i$;
    $e_i \leftarrow \sigma\left[\beta\left(s_i - \lambda_i\right)\right]$;
  Set weighted summation vector $\Gamma \leftarrow [s_1, s_2, \ldots, s_n]^T$;
  Set enable vector $E \leftarrow [e_1, e_2, \ldots, e_n]^T$;
  $\Psi \leftarrow \Gamma \circ E$;
  foreach $p_j \in P$, $j \in \mathbb{N}^+, 1 \le j \le m$ do
    $\alpha_j \leftarrow \max\left(O_j \circ \Psi \circ CF\right)$;
  Set $M_k \leftarrow [\alpha_1, \alpha_2, \ldots, \alpha_m]^T$;
  $k \leftarrow k + 1$;

D. Learning Algorithm of FLFPN

We employ a nested structure consisting of a genetic algorithm (GA) and a backpropagation algorithm (BP) to learn parameters. Parameters are categorized into basic parameters and structural parameters, representing the precise combination of higher-order terms introduced. We consider the thresholds, certainty factors, and structural parameters as network hyperparameters, which are determined by the GA encoding. Evaluation of fitness can occur following weight training, leading to the determination of overall parameter performance. The overall procedure is shown in Fig. 2 and Algorithm 2.

1) Genetic Algorithm: GA is a global search optimization that based on evolutionary theory [18]. For the FLFPN model, there are three parameters required to be coded: structural parameters, thresholds and certainty factors.

A tensor representation pattern without the constant term, such as $(x_1, x_2, x_1^2,\ x_1x_2, x_2^2)$, can be coded by a binary chromosome. For example, code $(1, 0, 1, 1, 0)$ denotes that the x_1, x_1^2 and x_1x_2 terms are selected, so a linear network with three weights will be constructed. For a transition with d input places, there is a binary chromosome of length $\left(\frac{(d+r)!}{d!r!} - 1\right)$ to represent its functional link terms, where $r = 2$ in this paper.

Algorithm 2: Learning Algorithm of FLFPN

Input: Sets of data, population size p_s, maximum number of generations g_{max}
Output: A set of FLFPN parameters with the best performance

Generate initial population of p_s chromosomes;
Let $g \leftarrow 1$, where g means this is the gth generation;
while $g \le g_{max}$ do
  foreach chromosomes C_i in gth generation, $i = 1, \ldots, p_s$ do
    train the weights using BP;
    compute fitness value;
    if C_i has better fitness value than the current optimal parameter set then
      $C_{best} \leftarrow C_i$;
      Update the best weights;
  Selection, Crossover, and Mutation;
  Generate a new population;
  $g \leftarrow g + 1$;

The values of λ and μ for a transition depend on their domain length and the required precision [19]. For a variable with domain length τ and required precision p, then b, the number of bits required for encoding, needs to satisfy the following condition $2^{b-1} < \tau \cdot 10^p < 2^b$. Since the range of λ and μ is $[0, 1]$, we have $\tau = 1$. If the required precision is two decimal places, then $p = 2$. $b = 7$ will satisfy the contidion, which means we need 7 binary bits to code each variable. A complete chromosome in the population contains n such binary fragments of each transition in FLFPN model.

Since the GA requires a fitness function, we will calculate the mean squared error (MSE) at the end of BP's training, based on the training data's error according to $MSE = \frac{\sum_{i=0}^{N}(\hat{y}-y)^2}{N}$, where N represents the number of samples. The fitness function will then use the inverse of MSE.

2) Back Propagation Algorithm: For ease of weight learning calculation, the FPN model for each rule can be divided into four layers, as illustrated in Fig. 3.

The first step of back propagation algorithm is to propagate the input forward through the network. For each transition, we have $\mathbf{a}^{(0)} = \boldsymbol{\alpha}$, $\mathbf{a}^{(1)} = \Phi\left(\mathbf{a}^{(0)}\right)$, $\mathbf{a}^{(2)} = G\left(s\right) = \mu s \sigma\left[\beta\left(s - \lambda\right)\right]$, where $s = \sum_{i=1}^{h} w_i \phi_i(\boldsymbol{\alpha}) = W^T \mathbf{a}^{(1)}$, and $\mathbf{a}^{(3)} = \max_j \mathbf{a}_j^{(2)}$, where all fired t_j have a commom output place.

The weights we need to learn are located between the link layer and transition layer, so the weight update rule

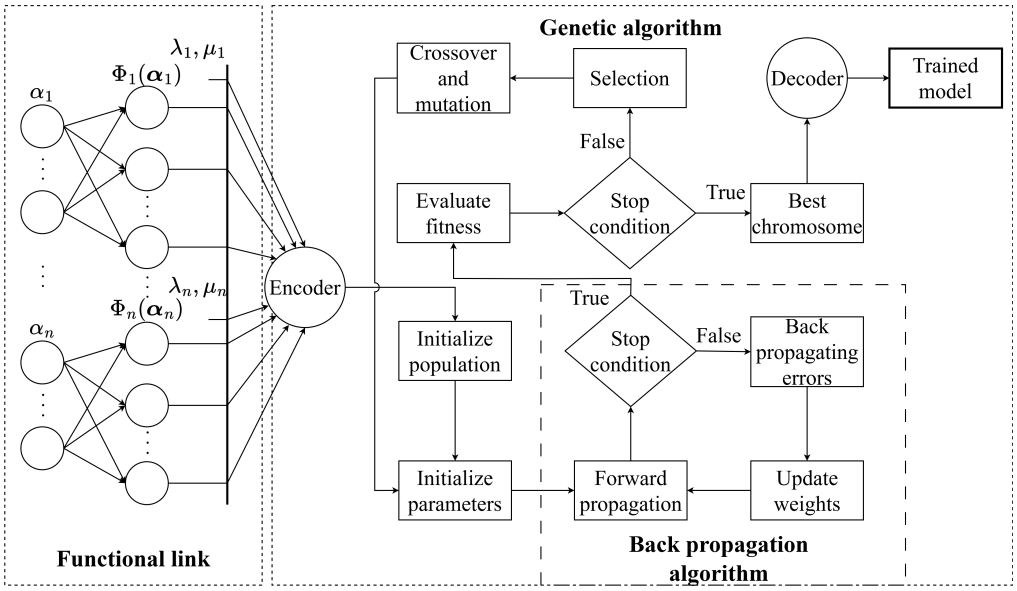

Fig. 2. The training process of FLFPN.

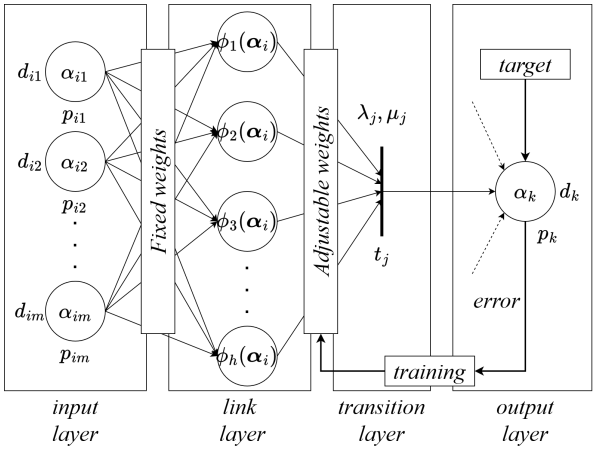

Fig. 3. Layered Structure of FLFPN.

can be expressed as the following equation

$$W^{(m)}(k+1) = W^{(m)}(k) - \alpha \boldsymbol{\delta}^{(m)} \left(\mathbf{a}^{(m-1)}\right)^T \tag{2}$$

where k denotes the number of iterations; α is the learning rate; $W^{(m)}$ is the weights of the mth layer; $\boldsymbol{\delta}^{(m)}$ is the sensitivities of the mth layer; $\mathbf{a}^{(m-1)}$ is the input items from the $(m-1)$th layer.

Note that since the FLFPN differs from a neural network in that the transition layer is not fully connected, it is necessary to set to zero the positions where no connection actually occurs after updating the weight matrix of this layer.

The back propagation of sensitivities needs to be divided into the following cases in accordance with different layers.

As for the terminal output layer: $\boldsymbol{\delta}^{(m)} = \mathbf{e}$, where $\mathbf{e} = \mathbf{a} - \mathbf{t}$. $\mathbf{t}$ is the target values, and $\mathbf{a}$ is the output values.

As for the other layers:

$$\begin{aligned}\boldsymbol{\delta}^{(m)} &= \left(\frac{\partial \mathbf{a}^{(m+1)}}{\partial \mathbf{a}^{(m)}}\right)^T \boldsymbol{\delta}^{(m+1)} \\ &= \begin{cases} B^T \boldsymbol{\delta}^{(m+1)} & \text{if } L_{transition} \\ \dot{G}^{(m)}(\mathbf{s}) \left(W^{(m+1)}\right)^T \boldsymbol{\delta}^{(m+1)} & \text{if } L_{link} \\ D^T \boldsymbol{\delta}^{(m+1)} & \text{if } L_{input} \end{cases}\end{aligned} \tag{3}$$

where

B is a matrix with its elements b_{ij} satisfying

$$b_{ij} = \begin{cases} 1 & \text{if single fired transition to } p_j \\ & \text{or } t_i \text{ offers the biggest output to } p_j \\ a_i^{(m)} & \text{if } t_i \text{ is not the biggest output to } p_j \\ 0 & \text{if no relation between } t_i \text{ to } p_j \end{cases}$$

Here we use the smooth derivative introduced in [20], which is defined as $\frac{\partial \max(y,p)}{\partial y} = \begin{cases} 1 & \text{if } y \geq p \\ y & \text{if } y < p \end{cases}$

This enhanced derivative can capture the real meaning of $y \geq p$ in a vague context, and can serve as a measure of fuzzy truth degree of the proposition "y is greater than or equal to p".

D is a matrix with its elements d_{ij} satisfying $d_{ij} = \partial \phi_i \left(\mathbf{a}^{(m)}\right) / \partial a_j^{(m)}$

$\dot{G}^{(m)}(\mathbf{s})$ is a diagonal matrix, and its elements $\dot{G}_i(s_j)$ can be calculated by the following equation. Let $G = G_i(s_j)$, then $\dot{G} = \frac{\partial}{\partial s_j}\{\mu s_k \sigma[\beta(s_k - \lambda_k)]\} = \beta G + \sigma[\beta(s_j - \lambda_i)](1 - \beta G)$

where the subscript i denotes that the G expression is for the ith transition on this layer, and the subscript j denotes that s_j is the jth input of this transition. A detailed justification of back propagation can be found in [21].

Due to the introduction of functional link inputs, the interpretability of the FLFPN model is reduced. In traditional FPNs, the weights reflect the relative importance of each antecedent proposition [22]. In FLFPN, the newly defined weights are essential for its learning ability, allowing the model to capture highly non-linear and complex relationships, however, FLFPN cannot explicitly explain the extent to which the input contributes to the output.

Similar to the black box issue in neural networks, it is arduous to discern the relative importance of each variable in a neural network [23]. In this paper, we opt to use the perturbation method. Following perturbation tests on all inputs, we compare the MSEs of outputs achieved with different levels of perturbation. The perturbation applied to each input will be divided into five levels: 10%, 20%, 30%, 40%, and 50% of the input's original value.

III. APPLICATION IN PREDICTING TRAFFIC SITUATIONS

A case study of local ramp control system mainly consists of the following set of rules.

$$R1: \text{IF } d_1 \text{ AND } d_2 \text{ THEN } d_5 \ (CF = \mu_1), \lambda_1, cd_{11}, cd_{21}$$
$$R2: \text{IF } d_3 \text{ AND } d_4 \text{ THEN } d_6 \ (CF = \mu_2), \lambda_2, cd_{32}, cd_{42}$$
$$R3: \text{IF } d_5 \text{ AND } d_6 \text{ THEN } d_7 \ (CF = \mu_3), \lambda_3, cd_{52}, cd_{63}$$

where

- d_1 The local speed is small.
- d_2 The local occupancy is big.
- d_3 The downstream speed is small.
- d_4 The downstream occupancy is big.
- d_5 The current road is congested.
- d_6 The downstream road is congested.
- d_7 The ramp flow is low.

In accordance with the proposed rules, we could establish the FLFPN model. We selected real traffic data collected from detectors around Sunnyside and Johnson Creek, Oregon, during the evening peak period of all working days in a month for model learning [24], and these data are processed by the corresponding fuzzy membership functions. We utilize the collected speed and occupancy data to forecast the ramp flow based on our model and established rules, and compare it to the actual data. The training results of the model parameters are as follows: $Th = [0.103, 0.052, 0.032]^T$, $CF = [0.977, 0.883, 0.963]^T$, weights on the first layer: $w1 = 0.324$, $w2 = 0.444$, $w3 = 0.29$, $w4 = 0.36$, $w5 = 0.244$, $w6 = 1.363$, $w7 = -1.201$, weights on the second layer: $w1 = 1.533$, $w2 = -0.511$, $w3 = -0.555$, where the code of functional link inputs is $(1, 1, 1, 0, 1)$, $(1, 1, 0, 0, 1)$, and $(0, 1, 0, 1, 1)$, which denotes that the tensor representation pattern of inputs to each transtion is: $\left(\alpha_1, \alpha_2, \alpha_1^2, \alpha_2^2\right)$, $\left(\alpha_3, \alpha_4, \alpha_4^2\right)$, and $\left(\alpha_6, \alpha_5\alpha_6, \alpha_6^2\right)$.

The trained model's MSE value is 0.042, compared to an MSE value of 0.063 when training the FPN using a same set of data without incorporating functional link terms. Given the exclusive selection of data during peak hours, even slight changes in traffic volume can lead to road congestion. As a result, minor gains in accuracy are crucial due to the heightened sensitivity to congestion. The incorporation of FLFPN in ramp control resulted in a 33% reduction in the MSE. This enhancement in flow prediction accuracy is essential for avoiding ramp congestion.

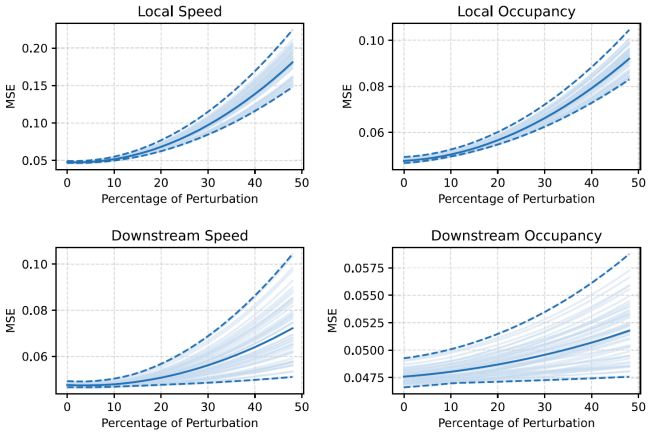

Fig. 4. The average and boundary values of MSE of the variable perturb method.

As for the contribution degree parameter, we analyze the relative importance of each input by the perturb method. Based on the 50 models trained with varied initial weights, each input variable produced 50 curves to reflect the fluctuation of MSE with noise in the experiment. As depicted in Fig. 4, the upper and lower boundaries are represented by the dashed line, and the mean is denoted by the solid line at the center. The contribution required is obtained by calculating the difference between the maximum and minimum values of the mean. The larger the obtained value, the more the MSE is affected by the noise and the higher the contribution of the variable to its final output. Normalizing the data leads to a CD matrix denoted by: $cd_{11} = 0.748$, $cd_{21} = 0.252$, $cd_{32} = 0.84$, $cd_{42} = 0.16$, $cd_{53} = 0.772$, $cd_{63} = 0.228$,

It can be seen that speed is more important than occupancy, and local traffic conditions have a greater impact on ramp traffic than downstream traffic conditions.

After the model is trained, we can conduct inference. Given a set of input values, the FLFPN parameters are as follows.

Since I, O, W_1, W_2, and W_3 are all sparse matrices, we simply write down their corresponding nonzero terms.
I: $I_{11} = I_{21} = I_{32} = I_{42} = I_{53} = I_{63} = 1$
O: $O_{51} = O_{62} = O_{73} = 1$
W_1: $w_{11} = 0.29$, $w_{22} = 0.36$, $w_{18} = 0.324$, $w_{28} = 0.444$
W_2: $w_{44} = -1.201$, $w_{38} = 0.244$, $w_{48} = 1.363$
W_3: $w_{56} = -0.511$, $w_{66} = -0.555$, $w_{68} = 1.363$
$Th = [0.103, 0.052, 0.032]^T$, $CF = [0.977, 0.883, 0.963]^T$

$M_0 = [0.106, 0.284, 0.059, 0.194, 0, 0, 0]^T$

The inference process is as follows.

$\mathbf{x}_1 = I_1 \circ M_0 : 1 = [0.106, 0.284, 0, 0, 0, 0, 0, 1]^T$

$s_1 = \mathbf{x}_1^T W_1 \mathbf{x}_1 = 0.193$

Similarly, we can obtain $s_2 = 0.324, s_3 = 0$, then

$\Gamma = [s_1, s_2, s_3]^T = [0.193, 0.324, 0]^T$

$e_1 = \frac{1}{1+e^{-\beta(s_1-\lambda_1)}} = 1$

Then we have $e_2 = 1, e_3 = 0$, and $E = [e_1, e_2, e_3]^T = [1, 1, 0]^T$,

$\Psi = \Gamma \circ E = [0.193, 0.324, 0]^T$

The new marking can be updated as:

$M_1 = [0.106, 0.284, 0.059, 0.194, 0.189, 0.286, 0]^T$

Similarly, we can get the marking for the next iteration:

$M_2 = [0.106, 0.284, 0.059, 0.194, 0.189, 0.286, 0.352]^T$

The calculated $M_3 = M_2$, therefore, we can judge that the termination condition has been reached and the inference is completed.

IV. CONCLUSIONS

Our proposed FLFPN model, augmented with functional link terms, exhibits enhanced learning performance compared to traditional FPN while preserving representational efficacy. This assertion is validated through experiments on highway traffic flow prediction. However, indiscriminate incorporation of such terms may lead to heightened computational demands. Particularly in the context of a chain structure, identifying the optimal combination of function linking terms becomes challenging. Thus, the future challenge lies in devising an optimal architectural framework tailored for meaningful applications, aimed at enhancing the model's learning efficiency while mitigating the computational overhead incurred during training.

References

[1] D. Yeung and E. Tsang, "Fuzzy knowledge representation and reasoning using petri nets," Expert Systems with Applications, vol. 7, no. 2, pp. 281–289, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0957417494900442

[2] S.-M. Chen, J.-S. Ke, and J.-F. Chang, "Knowledge representation using fuzzy petri nets," IEEE Transactions on Knowledge and Data Engineering, vol. 2, no. 3, pp. 311–319, 1990.

[3] C. Looney, "Fuzzy petri nets for rule-based decisionmaking," IEEE Transactions on Systems, Man, and Cybernetics, vol. 18, no. 1, pp. 178–183, 1988.

[4] H.-C. Liu, J.-X. You, Z. Li, and G. Tian, "Fuzzy petri nets for knowledge representation and reasoning: A literature review," Engineering Applications of Artificial Intelligence, vol. 60, pp. 45–56, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197617300222

[5] H.-C. Liu, L. Liu, Q.-L. Lin, and N. Liu, "Knowledge acquisition and representation using fuzzy evidential reasoning and dynamic adaptive fuzzy petri nets," IEEE Transactions on Cybernetics, vol. 43, no. 3, pp. 1059–1072, 2013.

[6] S.-M. Chen, "A new approach to handling fuzzy decision-making problems," IEEE Transactions on Systems, Man, and Cybernetics, vol. 18, no. 6, pp. 1012–1016, 1988.

[7] X. Li, W. Yu, and F. Lara-Rosano, "Dynamic knowledge inference and learning under adaptive fuzzy petri net framework," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 30, no. 4, pp. 442–450, 2000.

[8] W.-M. Wang, X. Peng, G. niu Zhu, J. Hu, and Y.-H. Peng, "Dynamic representation of fuzzy knowledge based on fuzzy petri net and genetic-particle swarm optimization," Expert Systems with Applications, vol. 41, no. 4, Part 1, pp. 1369–1376, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413006465

[9] M. Amin and D. Shebl, "Reasoning dynamic fuzzy systems based on adaptive fuzzy higher order petri nets," Information Sciences, vol. 286, pp. 161–172, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025514007014

[10] M. Gong, H. Song, J. Tan, Y. Xie, and J. Song, "Fault diagnosis of motor based on mutative scale back propagation net evolving fuzzy petri nets," in 2017 Chinese Automation Congress (CAC), 2017, pp. 3826–3829.

[11] L. Feng, M. Obayashi, T. Kuremoto, and K. Kobayashi, "A learning fuzzy petri net model," IEEJ Transactions on Electrical and Electronic Engineering, vol. 7, 05 2012.

[12] X. Li and F. Lara-Rosano, "Adaptive fuzzy petri nets for dynamic knowledge representation and inference," Expert Systems with Applications, vol. 19, no. 3, pp. 235–241, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417400000361

[13] Y. Pao, "Adaptive pattern recognition and neural networks," Reading: Addison Wesley, 1989, 1 1989. [Online]. Available: https://www.osti.gov/biblio/5238955

[14] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," Computer, vol. 25, no. 5, pp. 76–79, 1992.

[15] A. Sierra, J. Macias, and F. Corbacho, "Evolution of functional link networks," IEEE Transactions on Evolutionary Computation, vol. 5, no. 1, pp. 54–65, 2001.

[16] Y.-C. Hu and F.-M. Tseng, "Functional-link net with fuzzy integral for bankruptcy prediction," Neurocomputing, vol. 70, no. 16, pp. 2959–2968, 2007, neural Network Applications in Electrical Engineering Selected papers from the 3rd International Work-Conference on Artificial Neural Networks (IWANN 2005). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523120600453X

[17] S. Dehuri and S.-B. Cho, "A comprehensive survey on functional link neural networks and an adaptive pso-bp learning for cflnn," Neural Computing & Applications, vol. 19, no. 2, SI, pp. 187–205, MAR 2010.

[18] J. HOLLAND, "Adaptation in natural and artificial systems : an introductory analysis with application to biology," Control and artificial intelligence, 1975. [Online]. Available: https://cir.nii.ac.jp/crid/1572261550376545152

[19] A. Osyczka, Evolutionary algorithms for single and multicriteria design optimization. Heidelberg: Physica-Verlag, 2002, indeks s. 214-218.

[20] A. Blanco, M. Delgado, and I. Requena, "Identification of fuzzy relational equations by fuzzy neural networks," Fuzzy Sets and Systems, vol. 71, no. 2, pp. 215–226, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0165011494002512

[21] M. T. Hagan, H. B. Demuth, and M. H. Beale, Neural network design. Distributed by Campus Pub. Service, Colorado University Bookstore, University of Colorado at Boulder. [Online]. Available: https://cir.nii.ac.jp/crid/1130282272136169216

[22] D. Yeung and E. Tsang, "Weighted fuzzy production rules," Fuzzy Sets and Systems, vol. 88, no. 3, pp. 299–313, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165011496000528

[23] J. de Oña and C. Garrido, "Extracting the contribution of independent variables in neural network models: a new approach to handle instability." Neural Computing & Applications, vol. 25, no. 3/4, pp. 859 – 869, 2014. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=97459937&lang=zh-cn&site=ehost-live

[24] P. S. U. PORTAL, "Transportation data archive for portland-vancouver," Available online:https://portal.its.pdx.edu/home, accessed on 8 May 2023.