Article

A Situation Assessment Method with an Improved Fuzzy Deep Neural Network for Multiple UAVs

Lin Zhang ^{1,*}, Yian Zhu ¹, Xianchen Shi ¹ and Xuesi Li ^{1,2}

¹ School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; zhuya@nwpu.edu.cn (Y.Z.); shixianchen@mail.nwpu.edu.cn (X.S.); lixuesi@mail.nwpu.edu.cn (X.L.)

² China Academy of Launch Vehicle Technology, Beijing 100076, China

* Correspondence: zhanglin_npu@outlook.com

Received: 8 March 2020; Accepted: 1 April 2020; Published: 4 April 2020

Abstract: To improve the intelligence and accuracy of the Situation Assessment (SA) in complex scenes, this work develops an improved fuzzy deep neural network approach to the situation assessment for multiple Unmanned Aerial Vehicle(UAV)s. Firstly, this work normalizes the scene data based on time series and use the normalized data as the input for an improved fuzzy deep neural network. Secondly, adaptive momentum and Elastic SGD (Elastic Stochastic Gradient Descent) are introduced into the training process of the neural network, to improve the learning performance. Lastly, in the real-time situation assessment task for multiple UAVs, conventional methods often bring inaccurate results for the situation assessment because these methods don't consider the fuzziness of task situations. This work uses an improved fuzzy deep neural network to calculate the results of situation assessment and normalizes these results. Then, the degree of trust of the current result, relative to each situation label, is calculated with the normalized results using fuzzy logic. Simulation results show that the proposed method outperforms competitors.

Keywords: Situation assessment; fuzzy deep neural network; multiple UAVs; adaptive momentum; Elastic SGD

1. Introduction

As a new type of advanced equipment, the unmanned aerial vehicle (UAV) can achieve independent flight and investigation tasks and it has attracted the attention of many scholars [1–3]. For some military missions, UAV has been proved to have potential value due to its excellent performance. Recently, UAV technology has achieved great progress due to the development of computer science and communication technology [4]. It is widely known that multiple UAVs can finish some cooperative tasks or military confrontation by making a decision under uncertain environment [5]. Meanwhile, previous works have indicated that the most critical premise of decision-making is situation assessment [6,7].

The intelligent decision-making technology for the intelligent robot system mainly includes three components: environmental perception module, situation assessment module, and behavior planning module. The framework of intelligent decision-making of the intelligent robot systems is shown in Figure 1. The perception module can acquire and represent environment information. The situation assessment module analyzes the environmental information and makes the high-level behavior strategy. Finally, the behavior planning module produces specific actions. Therefore, situation assessment is the basis of intelligent decision-making for robots, and it is an essential component for robots to conduct collaborative tasks [8]. The situation assessment system can give a complete high-level description for the current situation according to the objective model and the task scenario, which is usually used to describe the relationship between entities and events. In addition, the situation assessment system can be extended to achieve the effective prediction of scene situation information. The existing theories of

situation assessment include multi-attribute decision-making [9], fuzzy set [10], neural network [11] and Bayesian network [12].

Figure 1. The basic structure for intelligent decision-making.

The situation assessment of multiple UAVs system refers to the classification of the superiority level for situation assessment according to current multiple factors, including the combat effectiveness of both sides and the degree of superiority. The situation assessment for multiple UAVs is the premise of intelligent decision-making and collaborative confrontation for a UAV group, so scholars have done a lot of research on this. The two renowned methods are the fuzzy logic method [13] and BP neural network method [14]. There are some problems with the situation assessment method based on fuzzy logic. First of all, artificial subjective factors have great influence, which may lead to poor adaptability and flexibility of a decision-making system. Secondly, the assessment deviation is large. Due to too many calculation steps including subjective consciousness, the accuracy of an assessment for a new task is not enough. For the situation assessment method based on neural network, the conventional neural network model focuses on the layer closing to the output layer to correct the error, which will lead to the insufficient assessment for a real-time task.

The latest development of machine learning technology makes it possible to achieve the more advanced decision-making system using the perceptual data, which provides an opportunity for multiple UAVs to improve the performance of the situation assessment in a dynamic environment. Recently, deep reinforcement learning combines the superiorities of reinforcement learning and deep learning, so it can help a learning agent to perform well in high-dimensional task space, with an end-to-end approach to achieve a complex task combining perception and decision together [15]. AlphaGo robot masters the game of Go to finish an advanced situation assessment and real-time decision-making by self-play without expert experience [16]. Genetic Fuzzy Trees uses genetic methods to optimize the multiple cascading fuzzy systems and this method has been proved to have excellent performance in the complex spatiotemporal environments [17]. These fuzzy controllers based on the Genetic Fuzzy Trees can even accomplish some new and different combat tasks efficiently. In [18] proposed a Deep Q-Network method for multi-agent confrontation and cooperative tasks in video games. The game agent uses a multi-agent DQN algorithm with a classical rewarding scheme, to learn a policy mapping a state to competitive or collaborative behavior.

The deep neural network is based on an artificial neural network [19]. Its most obvious feature is that it contains multiple hidden layers and addresses the problem of the sparse correction gradient of an artificial neural network model. The deep neural network can better mine the distributed features and implicit abstract features of the data sample by combining the shallow features into more abstract high-level features [20]. Recently, the deep neural network method has been widely used in situation assessment scenarios. For example, in [21], a CNN model is proposed to fit the advanced comprehension process of the battlefield situation, so as to achieve the simulation of the commander's thinking process. However, this situation assessment method based on a CNN model only focuses on the situation comprehension module, and the situation prediction stage is achieved using the expert

experience. This method will limit the situation assessment process to expert experience. In [22], an Auto-encoder model is proposed to describe the relationship between the performance metrics of the situation assessment system for multiple UAVs. However, this method requires a large number of training data, and training data need to be preprocessed using the expert experience. The preprocessing process will lead to a lot of time costs.

In summary, two critical issues have to be considered in situation assessment problems. Firstly, in real life, for the results of the situation assessment of a task, there is often no definite situation assessment. For example, to complete a situation assessment of one team in the football field, it is too rough to give the current situation with “strengths” or “weaknesses”, and rough assessment will lead to an inaccurate real-time evaluation of the scene. Secondly, the conventional deep neural network model for situation assessment often has poor learning performance [23]. The optimization method based on the training process may be a solution to improve the performance of deep neural networks.

Previous works have the following problems. First of all, the previous situation assessment technology for multiple UAVs often does not consider that the multi-objective decision-making environment is a dynamic and uncertain environment, so the rough results of conventional methods cannot get accurate situation assessment results. Secondly, the performance of the deep neural network for the situation assessment of multiple UAVs is often limited by the training process of the neural networks. Moreover, conventional methods often focus on situation comprehension or situation awareness, while for situation prediction, the method based on expert experience is used to get results. Two conventional situation assessment methods for end-to-end prediction are the fuzzy logic method and BP neural network method. However, these two methods have some disadvantages. A more practical situation assessment method is lacking.

To address these problems, this paper develops a situation assessment method for multiple UAVs, using an improved fuzzy deep neural network. The proposed situation assessment method achieves an end-to-end mapping from environmental information to assessment results. The normalized situation value is the input of the improved fuzzy deep neural network, and the output of the network is the degree of trust for each situation label. The advantage of the fuzzy deep neural network is that the output situation assessment result has a certain fuzziness, which is more consistent with the uncertainty of the real-time system. At the same time, the deep neural network realizes a deeper feature representation, which can achieve more accurate assessment results. Secondly, in order to accelerate the convergence rate of the deep neural networks, adaptive momentum [24] and Elastic SGD (Elastic Stochastic Gradient Descent) [25] are introduced into the training process of the neural networks. The simulation results verify the effectiveness of the proposed method.

The contributions of this paper are as follows: In this paper, an improved fuzzy deep neural network method for situation assessment of multiple UAVs is proposed. An end-to-end mapping is achieved without requiring too much expert experience in the situation prediction process. Firstly, this method uses the advantages of deep feature representation of the deep neural network. Meanwhile, we use fuzzy logic to calculate the degree of trust in the output assessment results. Secondly, we introduce adaptive momentum and elastic SGD to accelerate the training speed of the deep neural network. Efficient learning can be more in line with the real-time situation assessment system.

The remainder of this work is organized as follows: Section 2 introduces the background of this work, the general process of situation assessment and the neural network model. Section 3 gives a kinematics model of UAV, which provides the UAV motion rules for the multiple UAVs situation assessment experiment. Section 4 introduces two baseline methods for the situation assessment of multiple UAVs, which are the fuzzy evaluation method and the classical BP neural network method. Section 5 introduces the proposed situation assessment method with an improved fuzzy deep neural network for multiple UAVs. Section 6 gives the simulation results calculated by the proposed method. Conclusions are drawn in the last section.

2. Background

2.1. Situation Assessment

Generally speaking, in a complete decision-making system, a situation comprehension is an important component of the situation evaluation. The situation assessment method estimates the situation effectively and then determines different strategies for different situations using the previous experience, such as attack or defense [26].

Previous work divides situation assessment into three phases: situation awareness, situation comprehension, and situation forecast, which are progressive [27], as shown in Figure 2. Firstly, according to the environmental information, the scene data is recorded by sensors, which is called situational awareness. Then, in the situation understanding phase, the current situation is evaluated after the processing of the real-time data for the current scene. Finally, an analysis of the current situation for the situation information with continuous scene data has been conducted to provide a more accurate decision-making result for the subsequent situation assessment, that is, the situation prediction process. Generally, the basic procedure for the situation assessment is shown in Algorithm 1.

Figure 2. The basic structure for the situation assessment.

Algorithm 1. The basic procedure for situation assessment

Step 1: Deduce the knowledge base of situation assessment using the expert experience in situation assessment;

Step 2: Obtain the complete knowledge map, that is, the knowledge representation via the classification and analysis of domain knowledge base;

Step 3: Record the real-time data for the real scene using the knowledge acquisition module and recorded information which is stored in the constructed knowledge base;

Step 4: Describe the rule knowledge as rules that can be recognized by the system and stored in the knowledge base according to the rule editor provided by the knowledge acquisition module, such as the data format given by the data acquisition module;

Step 5: Answer the real-time feedback question in the knowledge module using the expert experience.

The answer to the feedback question is directly sent to the knowledge acquisition module or processed and then sent to the knowledge acquisition module, waiting for a certain behavior decision.

2.2. Neural Network Model

Figure 3 gives the basic structure for the neural network. The neural network is a multilayer feed-forward neural network model, which consists of three parts: an input layer, a hidden layer, and an output layer. Each part contains an indefinite number of layers, and each layer contains one or more nonlinear neural units.

Figure 3. The basic structure for the neural network.

The connection weights of the BP neural network are updated by the back-propagation algorithm [28], which includes two processes: the signal forward transmission and the error back-propagation. In the process of the forward transmission, the input signal is input from the input layer, and then processed layer by layer through the hidden layer until it reaches the output layer, and finally, the output result is calculated in the output layer. The state of each neuron of a layer in the deep network is only affected by the state of the neurons of the previous layer, but it won't be affected by the state of the neurons of other layers. If there is a gap between the desired result and the actual result for the neural network, the error signal will be sent back to each layer via back-propagation to modify the network weights. Previous works have demonstrated that the neural network model can estimate any nonlinear function by learning a large number of learning samples [29].

Deep Neural Network (DNN) uses multiple layers of neuron units to achieve feature extraction, which is a very common machine learning method. Recently, the research of deep neural networks comes from artificial neural networks [30]. Deep Neural Network can combine the low-level features into high-level features, that represent features at a more abstract level. The similarities between deep neural networks and artificial neural networks are that they both adopt a hierarchical structure. The system consists of three parts: the input layer, an output layer, and a hidden layer. The nodes between adjacent layers are connected, and the nodes of the same layer or cross-layer are not connected. Each layer can be regarded as a logistic regression model [31]. Figure 4 shows the basic structure for the deep neural network, which is similar to the working principle of the human brain. It is a significant function for the DNN to learn multi-level representation corresponding to different levels of abstraction.

Figure 4. The basic structure for the deep neural network.

3. The Kinematic Model for UAV

In this mission, the flight altitude of each UAV is constant, that is, each UAV has been flying at a fixed altitude. In the process of flight for each UAV, the ground radar should be taken as the reference position. The movement model of each UAV is shown below.

The coordinate for radar is (x_r, y_r, z_r) . r_i is the distance from the i -th UAV to the radar. For the i -th UAV, θ_i is the pitch angle, φ_i is the azimuth angle, β_i is the heading angle, and h_i is the flight altitude. The motion relationship of UAV in the Cartesian Coordinate System is shown in Figure 5. The kinematic model of UAV flying at a fixed altitude is shown in Equation (1).

$$\begin{cases} \dot{x} = v_i \sin \beta_i \\ \dot{y}_i = v_i \cos \beta_i \\ \dot{z}_i = 0 \end{cases} \quad (1)$$

Figure 5. The kinematic model of Unmanned Aerial Vehicle (UAV).

According to the transformation relationship between the Polar Coordinate System and the Cartesian Coordinate System, the kinematic model of UAV is transformed into a Polar Coordinate System.

The kinematic model of UAV in the Polar Coordinate System is shown in Equation (2).

$$\begin{cases} x_i = r_i \cos \theta \cos \varphi \\ y_i = r_i \cos \theta \sin \varphi \\ z_i = r_i \sin \theta \end{cases} \quad (2)$$

The derivative of Equation (2) is given by,

$$\begin{cases} \dot{x}_i = \dot{r}_i \cos \theta \cos \varphi - r_i \dot{\theta} \sin \theta \cos \varphi - r_i \dot{\varphi} \cos \theta \cos \varphi \\ \dot{y}_i = \dot{r}_i \cos \theta \sin \varphi - r_i \dot{\theta} \sin \theta \sin \varphi + r_i \dot{\varphi} \cos \theta \cos \varphi \\ \dot{z}_i = \dot{r}_i \sin \theta + r_i \dot{\theta} \cos \theta \end{cases} \quad (3)$$

The expression for Equation (3) using the matrix is given by,

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \varphi & -\sin \varphi & -\sin \theta \cos \varphi \\ \cos \theta \sin \varphi & \cos \varphi & -\sin \theta \sin \varphi \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \dot{r}_i \\ r_i \dot{\varphi} \cos \theta \\ r_i \dot{\theta} \end{bmatrix} \quad (4)$$

According to Equations (2)–(4), with radar as the coordinate origin, the kinematics model of UAV in the Polar Coordinate System is shown in Equation (5).

$$\begin{bmatrix} \dot{r}_i \\ r_i \dot{\varphi} \cos \theta \\ r_i \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \varphi & \cos \theta \sin \varphi & \sin \theta \\ -\sin \varphi & \cos \varphi & 0 \\ -\sin \theta \cos \varphi & -\sin \theta \sin \varphi & \cos \theta \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} \quad (5)$$

According to Equation (1) and Equations (5) and (6) is given by,

$$\begin{cases} \dot{r}_i = v_i \cos \theta \cos \varphi \sin \beta_i + v_i \cos \theta \sin \varphi \cos \beta_i \\ r_i \dot{\varphi} \cos \theta = -v_i \sin \varphi \sin \beta_i + v_i \cos \varphi \cos \beta_i \\ r_i \dot{\theta} = -v_i \sin \theta \cos \varphi \sin \beta_i - v_i \sin \theta \sin \varphi \cos \beta_i \end{cases} \quad (6)$$

The kinematic parameters of the UAV are given by,

$$\begin{cases} \cos \beta_i = r_i (\dot{\varphi} \cos \theta \sin \theta \cos \varphi - \dot{\theta} \sin \varphi) / (v_i \sin \theta) \\ v_i = (\dot{r}_i^2 + (r_i \dot{\varphi} \cos \theta)^2 + (r_i \dot{\theta})^2)^{1/2} \end{cases} \quad (7)$$

$$\begin{cases} \theta = \arcsin[(z_i - z_r) / r_i] \\ \varphi = \arctan[(y_i - y_r) / (x_i - x_r)] \\ \dot{\theta} = \frac{((x_i - x_r)^2 + (y_i - y_r)^2)^{1/2} \dot{z}_i}{r_i^2} - \frac{(z_i - z_r)}{r_i^2} \left[\frac{(x_i - x_r)x_i^2 + (y_i - y_r)y_i^2}{((x_i - x_r)^2 + (y_i - y_r)^2)^{1/2}} \right] \\ \dot{\varphi} = \frac{(x_i - x_r)\dot{y}_i + (y_i - y_r)\dot{x}_i}{((x_i - x_r)^2 + (y_i - y_r)^2)^{1/2}} \end{cases} \quad (8)$$

Figure 6 shows the flight paths of two different UAVs. The UAV used in this experiment is based on the kinematics model of this section.

Figure 6. The flight paths for two different UAVs.

4. Conventional Methods for Situation Assessment

4.1. A Situation Assessment with BP Neural Network (SA-BP)

A situation assessment method with BP neural network is shown in Figure 7.

Figure 7. The situation assessment method with Back Propagation (BP) neural networks.

The basic procedure is given by Algorithm 2.

Algorithm 2. The conventional method with SA-BP

Step 1: Define a BP neural network model. Choose a reasonable activation function and define a neural network model from the input layer to the output layer. Meanwhile, the final assessment model can be obtained by combining it with a certain amount of data training.

Step 2: Define the input layer for the neural network. All the information obtained by the situational awareness module is used as a set for input neurons, which includes all scene information data in a certain period pushed forward from the current time. The scene data is recorded according to the evaluation factors. $\mathbf{IN} = \{I_i | i = 1, 2, \dots, m\}$ represents n pieces of scene data which have been obtained from the task scene.

Step 3: Define a situation assessment set. In order to describe the scene situation reasonably, it is necessary to set the situation label in combination with the real-time situation. n situation labels are set and the set for situation label is given by,

$$\mathbf{B} = \{b_i | i = 1, 2, 3, \dots, n\} \quad (9)$$

Step 4: Normalization of input data. By recording the data within a certain period, an approximate range of each evaluation factor can be obtained. If the minimum value in the recorded data of the evaluation factor I_j is $I_j^{\min}$ and the maximum value is $I_j^{\max}$, then the given value range $[I_j^{bo}, I_j^{to}]$ is given by Equation (10) and the normalized data $\hat{I}_j$ for the evaluation factor I_j is given by Equation (11).

$$\begin{cases} I_j^{bo} = 0.78 \times I_j^{\min}, & \text{if } I_j^{\min} \geq 0 \\ I_j^{to} = 0.78 \times I_j^{\max}, & \text{if } I_j^{\max} < 0 \\ I_j^{bo} = 0.122 \times I_j^{\min}, & \text{if } I_j^{\min} < 0 \\ I_j^{to} = 0.122 \times I_j^{\max}, & \text{if } I_j^{\max} \geq 0 \end{cases} \quad (10)$$

$$\hat{\mathbf{I}} = \left\{ \hat{I}_j = \begin{cases} 0, & \text{if } I_j \leq I_j^{bo} \\ 1, & \text{if } I_j > I_j^{to} \\ \frac{I_j - I_j^{bo}}{I_j^{to} - I_j^{bo}}, & \text{if } I_j^{bo} < I_j \leq I_j^{to} \end{cases}, j = 1, 2, \dots, m \right\} \quad (11)$$

Finally, $\hat{\mathbf{I}} = \{\hat{I}_i | i = 1, 2, \dots, m\}$ is the normalized set of input data.

Step 5: Achieve a real-time situation assessment. When the new data enters into the assessment network model, the result calculated by the neural network is a result vector for each situation label. Then, we take the output assessment corresponding to the maximum value as the final situation assessment result.

4.2. A Fuzzy Evaluation Method for Situation Assessment (SA-F)

Fuzzy evaluation is a method based on the theories of fuzzy mathematics to provide a reasonable evaluation of the actual tasks. A fuzzy evaluation based situation assessment method is to apply the fuzzy evaluation method to the assessment of specific complex scene situations. The procedure is shown in Algorithm 3.

Algorithm 3. The conventional method with SA-F

Step 1: Define the evaluation factor. Considering specific tasks and m evaluation factors obtained from the scene, the set for the evaluation factor is $\mathbf{A} = \{a_i | i = 1, 2, \dots, m\}$.

Step 2: Define the situation label. In order to describe the situation of the scene reasonably, we need to set the final situation label. The number of situation labels is n , which is $\mathbf{B} = \{b_i | i = 1, 2, \dots, n\}$.

Step 3: Define a fuzzy priority relation matrix. Firstly, the relative importance of each evaluation factor to the final situation assessment needs to be determined. Setting the expression of the relative importance of two factors a_i and a_j to d_{ij} , we can get the Equation (12).

$$\begin{cases} d_{ij} = \frac{1}{2}, & \text{if } i = j \\ d_{ij} = 1 - d_{ji}, & \text{if } i \neq j \end{cases} \quad i, j = 1, 2, 3, \dots, m \quad (12)$$

Then, the priority relationship matrix is given by $\mathbf{D} = [d_{ij}]_{m \times m}$.

Step 4: Define a fuzzy weight set. Combining the fuzzy analytic hierarchy process to transform the fuzzy matrix into the fuzzy consistent matrix $\hat{\mathbf{D}} \in \mathbb{R}^{m \times m}$.

$$\hat{\mathbf{D}} = [\hat{d}_{ij}] \Rightarrow \left[\hat{d}_{ij} = \frac{1}{2} \left(\frac{\sum_{l=1}^m d_{il} - \sum_{l=1}^m d_{jl}}{n} + 1 \right) \right] i, j = 1, 2, 3, \dots, m \quad (13)$$

Set an evaluation factor U in the m_U -th row in the fuzzy consistent matrix $\hat{\mathbf{D}}$, then the fuzzy weight value of the evaluation factor relative to the final situation assessment is given by,

$$\omega_U = \frac{1}{n} + \frac{1}{k} \left(1 + \sum_{j=1}^m \frac{\hat{d}_{Uj}}{n} \right) = \frac{1}{n} + \frac{1}{k} \left(1 + \sum_{j=1}^m \frac{1}{2n} \left(\frac{\sum_{l=1}^m d_{Ul} - \sum_{l=1}^m d_{jl}}{n} + 1 \right) \right) \quad (14)$$

where $k \geq 0.5 \times (n - 1)$ is a constant and its specific value is given according to the actual task.

Therefore, the final set of weights for the m evaluation factors can be obtained according to Equation (14) and it is recorded as

$$\boldsymbol{\omega} = (\omega_1, \omega_2, \dots, \omega_m)^T \quad (15)$$

Step 5: Achieve a fuzzy situation assessment. If the current time is t , the value for the degree of trust of each evaluation factor relative to the situation described in the first T time slices is calculated using the fuzzy method, which is given by $\eta : d_j \xrightarrow{f(x)} f(d_j) \Rightarrow \boldsymbol{\varphi}_j$, as shown in:

$$\boldsymbol{\varphi}_j = \{\phi_i^j | i = 1, 2, \dots, n\} \quad (16)$$

The final fuzzy situation assessment result using the “maximum-minimum operation” [29] is calculated, which is represented by a fuzzy vector $\mathbf{M} = \boldsymbol{\omega} \times \boldsymbol{\eta} = [m_j] \in \mathbb{R}^{1 \times n}$ as shown in:

$$m_j = \bigvee_{i=1}^m (\phi_{ji} \wedge \omega_i), i = 1, 2, 3, \dots, n \quad (17)$$

where $\wedge$ represents the smallest value in the set. $\vee$ represents the largest value in the set.

Step 6: Achieve a comprehensive assessment result. According to the maximum operation, a comprehensive assessment result of the current situation which is represented as Z , can be obtained, which is given by,

$$\max\{m_1, m_2, m_3, \dots, m_n\} = m_j^{\max} \Rightarrow b_j^{\max} = Z \quad (18)$$

5. An Improved Fuzzy Deep Neural Network for Situation Assessment of Multiple UAVs

5.1. The Framework for the Proposed Method

The basic framework of the proposed method is shown in Figure 8. Multi-UAVs system obtains current scene data in the environment using the situation awareness technology. In Figure 8, scene data is input from the input layer. Then, we normalize the input data in the normalization layer.

An improved deep neural network is used to calculate the output results, that is, the results of the situation assessment. In the fuzzy layer, we achieve the fuzzification of the output value and calculate the degree of truth for the situation assessment results. The output is the degree of truth of each situation label for multiple UAVs. In the training process for the deep neural network, we use the adaptive momentum and Elastic SGD to improve the performance of the neural network and this method can accelerate the convergence rate of the deep neural network.

Figure 8. The framework for the proposed method.

5.2. An Improved Deep Neural Network Model with Adaptive Momentum and Elastic SGD

The input of the neural network is a data vector with m samples: $\mathbf{I} = \{I_i | i = 1, 2, \dots, m\}$, and the labels of output are: $\hat{\mathbf{O}} = \{\hat{O}_i | i = 1, 2, \dots, n\}$. The error between the actual output $\mathbf{O} = \{O_i | i = 1, 2, \dots, n\}$ of the network and the label output vector $\hat{\mathbf{O}} = \{\hat{O}_i | i = 1, 2, \dots, n\}$ is used to update these weights so that the current output $\mathbf{O}$ is close to the desired value $\hat{\mathbf{O}}$. According to the decline of the slope of the error function, the weight and the deviation are determined, which makes the error function change towards the direction of the desired output, and transmit it to other layers in this way. The basic procedure for the neural network model with adaptive momentum and Elastic SGD is as follows.

Step 1: Forward transmission.

The output y_n of the hidden layer is given by,

$$a_1^i = f\left(\sum_{j=1}^r \omega_1^{ij} I_j + \theta_1^i\right), (i = 1, 2, 3, \dots, r) \quad (19)$$

The output O_n of the output layer is given by,

$$O_i = f\left(\sum_{j=1}^{s_1} \omega_2^{kj} a_1^i + \theta_1^k\right), (k = 1, 2, 3, \dots, s_1) \quad (20)$$

where $f(x) = \frac{2E}{1+e^{-Fx}} - F$ is the activation function. $E = 1.812$ and $F = 0.676$ are constant.

The error function is given by,

$$E(\omega, \theta) = \frac{1}{2} \sum_{k=1}^{s_2} (\hat{O}_k - O_k)^2 \quad (21)$$

Step 2: Back-propagation.

We use the gradient descent method to calculate the error for the weights of the output layer. The updating law with adaptive momentum for the weights from the i -th input to the k -th output is given by,

$$\begin{aligned}\Delta_t \omega_2^{ki} &= \beta \Delta_{t-1} \omega_2^{ki} - \lambda \frac{\partial E^t}{\partial \omega_2^{ki}} = \beta \Delta_{t-1} \omega_2^{ki} - \lambda \left(\frac{\partial E^t}{\partial O_k^t} \right) \cdot \left(\frac{\partial O_k^t}{\partial \omega_2^{ki}} \right) \\ &= \beta \Delta_{t-1} \omega_2^{ki} + \lambda (\hat{O}_k - O_k^t) a_1^i f' = \beta \Delta_{t-1} \omega_2^{ki} + \lambda \delta_{ki} a_1^i\end{aligned}\quad (22)$$

where $\delta_{ki} = (\hat{O}_k - O_k^t) f'$.

So, Equation (22) can be deduced.

$$\begin{aligned}\Delta \theta_1^{ki} &= -\lambda \frac{\partial E^t}{\partial \theta_1^{ki}} = -\lambda \left(\frac{\partial E^t}{\partial O_k^t} \right) \cdot \left(\frac{\partial O_k^t}{\partial \theta_1^{ki}} \right) \\ &= \lambda (\hat{O}_k - O_k^t) f' = -\lambda \delta_{ki}\end{aligned}\quad (23)$$

We use the gradient descent method to calculate the error for the weights of the hidden layer. The updating law with adaptive momentum for the weights from the j -th input to the i -th output is given by,

$$\begin{aligned}\Delta_t \omega_1^{ki} &= \beta \Delta_{t-1} \omega_1^{ki} - \lambda \frac{\partial E^t}{\partial \omega_1^{ki}} = \beta \Delta_{t-1} \omega_1^{ki} - \lambda \left(\frac{\partial E^t}{\partial O_k^t} \right) \cdot \left(\frac{\partial O_k^t}{\partial a_1^i} \right) \cdot \left(\frac{\partial a_1^i}{\partial \omega_1^{ki}} \right) \\ &= \beta \Delta_{t-1} \omega_1^{ki} + \lambda \sum_{k=1}^{s_2} (\hat{O}_k - O_k^t) \omega_2^{ki} f' \times f' \times I_j = \beta \Delta_{t-1} \omega_1^{ki} + \lambda \delta_{ij} I_j\end{aligned}\quad (24)$$

where $\delta_{ij} = \sum_{k=1}^{s_2} (\hat{O}_k - O_k^t) \omega_2^{ki} f' \times f'$ and $\Delta \theta_1^k = \lambda \delta_{ij}$.

In practice, the weight update is related to t , and the weight update is determined by its trend of the gradient change to avoid the slow convergence of the neural network. The updating law with adaptive momentum and Elastic SGD for the weight update is obtained by Equation (25).

$$\Delta_t \omega = \begin{cases} -\Delta_t \omega, & \text{if } \frac{\partial E^t}{\partial \omega} > 0 \\ +\Delta_t \omega, & \text{if } \frac{\partial E^t}{\partial \omega} \leq 0 \\ 0, & \text{if } \frac{\partial E^t}{\partial \omega} = 0 \end{cases}\quad (25)$$

Then, the weight update at the time $t + 1$ is given by $\omega_{t+1} = \Delta_t \omega + \omega_t$.

The updating rule for $\Delta_t \omega$ is given by,

$$\Delta_t \omega = \begin{cases} \mu \times \Delta_{t-1} \omega, & \text{if } \frac{\partial E^t}{\partial \omega} \times \frac{\partial E^{t-1}}{\partial \omega} > 0 \\ \varepsilon \times \Delta_{t-1} \omega, & \text{if } \frac{\partial E^t}{\partial \omega} \times \frac{\partial E^{t-1}}{\partial \omega} \leq 0 \\ \Delta_{t-1} \omega, & \text{if } \frac{\partial E^t}{\partial \omega} \times \frac{\partial E^{t-1}}{\partial \omega} = 0 \end{cases}\quad (26)$$

where $0 < \varepsilon < 1 < \mu$. When two update processes are in the same direction, the value $\Delta_t \omega$ is increased. But when the two update processes are in different directions, the value $\Delta_t \omega$ is reduced. Other situations do not change the value $\Delta_t \omega$.

5.3. The Whole Algorithm Using An Improved Fuzzy Deep Neural Network for Situation Assessment of Multiple UAVs

The procedure for the proposed method is shown below.

Step 1: Define an improved deep neural network model. The initial weights of the hidden layer are assigned randomly. The output layer is a fuzzy layer.

Step 2: Define the input layer for the improved deep neural network. The current time is t and the length for a certain period is T . If the scene data at each time has n values, the scene data gave by Equation (27) from $t - T + 1$ to t is selected as the input.

$$\mathbf{IN} = \{I_{t-i+1}^1, I_{t-i+1}^2, \dots, I_{t-i+1}^n | i = 1, 2, 3, \dots, T - 1, T\}\quad (27)$$

Step 3: Define a set for situation assessment. In order to properly describe the scenario situation, it is necessary to combine the real-time situation with the expert experience to set the final situation label. Suppose that n situation labels are finally set and the situation labels are aggregated as:

$$\mathbf{B} = \{b_j | b_j = j, j = 1, 2, 3, \dots, n\} \quad (28)$$

Step 4: Normalization of the input data. A preliminary range of values for each scene data I_i is obtained by recording the data over a certain period. The minimum value for each scene data I_i in the record data is $I_i^{\min}$ and the maximum value is $I_i^{\max}$. The updating law for calculating a range $[I_j^{bo}, I_j^{to}]$ is given by Equation (29).

The set for the normalized data is $\hat{\mathbf{I}} = \{\hat{I}_i | i = 1, 2, \dots, m\}$. The normalized data $\hat{I}_i$ for each scene data I_i is given by Equation (30).

$$\begin{cases} I_j^{bo} = 0.122 \times I_j^{\min}, & \text{if } I_j^{\min} < 0 \\ I_j^{to} = 0.78 \times I_j^{\max}, & \text{if } I_j^{\max} < 0 \\ I_j^{bo} = 0.78 \times I_j^{\min}, & \text{if } I_j^{\min} \geq 0 \\ I_j^{to} = 0.122 \times I_j^{\max}, & \text{if } I_j^{\max} \geq 0 \end{cases} \quad (29)$$

$$\hat{\mathbf{I}} = \left\{ \hat{I}_j | \hat{I}_j = \begin{cases} 1, & \text{if } I_j > I_j^{to} \\ 0, & \text{if } I_j \leq I_j^{bo} \\ \frac{I_j - I_j^{bo}}{I_j^{to} - I_j^{bo}}, & \text{if } I_j^{bo} < I_j \leq I_j^{to} \end{cases}, j = 1, 2, \dots, m \right\} \quad (30)$$

Step 5: Update of network weights using the collected data samples. The weights for the neural network are corrected using the back-propagation method with adaptive momentum and Elastic SGD. Finally, the final assessment network model can be obtained.

Step 6: Achieve the real-time situation assessment using fuzzy logic. The vector for the assessment result $\hat{\mathbf{B}} = \{\hat{b}_i | i = 1, 2, \dots, n\}$ of each situation label is calculated using an improved deep neural network model. Then the result corresponding to the maximum value is taken as the final situation assessment result. For the vector of the real-time assessment result output by the improved deep neural network, the final vector for the degree of truth for situation assessment $\mathbf{P} = \{p_i | i = 1, 2, \dots, n\}$ is given by,

$$\mathbf{P} = \left\{ p_k | p_k = \text{Averg}(\hat{b}_k) = \frac{\hat{b}_k}{\sum_{k=1}^n \hat{b}_k}, k = 1, 2, 3, \dots, n \right\} \quad (31)$$

The proposed situation assessment method using an improved fuzzy deep neural network, is shown in Algorithm 4.

Algorithm 4. The proposed situation assessment method for Multiple UAVs.

Definition

Data_Samples: = Training data with scene data and situation labels.

Num_data: = Number of training data.

I_i : = The i -th input scene data for input.

$\hat{I}_i$: = Normalized data for the input.

Training_Bylay(): = Training the hidden layer layer by layer for Improved Deep network

Net_Fc: = The activation function for Deep net

Im_Deep_Net: = Improved Deep Neural Network (DNN) model.

Normalization(): = Normalization for the input

Fuzzy_out(): = Fuzziness of output

Out_DNN(): = Calculate the output of the Improved Deep network

Training_Im_Deepnet(): = Training for the Improved Deep network

Offline training period for the Improved DNN:

$i \leftarrow 0$;

Repeat $i++$

$\hat{I}_i \leftarrow \text{Normalization}(\text{Data_Samples})$;

Training_Bylay(**Net_Fc**, $\hat{I}_i$);

Until $i > \text{Num_data}$;

Training_Im_Deepnet(**Im_Deep_Net**, **Data_Samples**);

Online training period for the Improved DNN:

$t \leftarrow 0$;

Repeat $t++$

$\hat{I}_i \leftarrow \text{Normalization}(I_i)$;

$\hat{b}_i \leftarrow \text{Out_DNN}(\hat{I}_i, \text{Im_Deep_Net})$;

$p_i \leftarrow \text{Fuzzy_out}(\hat{b}_i)$;

Until $t \geq T_{\text{max}}$

6. Simulation

6.1. Experiment on Classification of Situation Labels

The classification of situation labels is a critical step of the situation assessment for multiple UAVs, so the performance of the classification method is very important for the effectiveness of situation assessment. In order to verify the performance of the proposed method, a simulation of signal classification is conducted. For the simulation, there are four types of signals to be classified, and the input vector is 24 dimensions. The structure of the neural network is as follows: the input layer has 24 neuron units corresponding to 24 input dimensions. The learning rate of the neural network is 0.15. Each hidden layer contains 25 neural units, and the output layer has 4 neural units. In this experiment, there are 2000 groups of signals to be classified, three-quarters of which are randomly selected, that is, 1500 groups of signals are used as training samples to train the situation assessment model using BP network (BP) [14] and the situation assessment model using the proposed improved deep neural network (INN) respectively.

In this experiment, two different situation assessment models, INN and BP, are used to classify four kinds of signals. First, we train the two models on the training data set. In order to avoid the accidental result of a single experiment, we train ten times respectively and then record the convergence time of each training process. Each training process is defined as a round. Figure 9 shows the training results for the two different models. The experimental results show that the convergence time of the proposed INN model is less than that of the BP method for ten training processes. The convergence time of the proposed INN model is about 2000 rounds, while that of the original BP method is about 8000 rounds. Adaptive momentum and Elastic SGD can accelerate the convergence rate of the neural network, which makes the proposed INN model need less convergence time. After the training, the classification performance of the two models is compared to the test data set. Each test is defined as a

round. In each round, we use two models to classify signals and record the results. We selected ten rounds of classification results for performance comparison in the test phase. The simulation results are shown in Figure 10. In Figure 10, the unit of ordinate is 10^{-4} . From the results of the simulation, we can see that the classification error of the INN method is lower than that of the BP method. The INN model has higher accuracy than the BP method. The deep neural network can achieve a deep feature representation of the signal, so the accuracy of classification is higher. The simulation results show that the INN model has better learning performance than the classical BP model, which provides a new solution to improve the performance of the situation assessment for multiple UAVs.

Figure 9. The convergence time for different models.

Figure 10. The classification error for different models.

6.2. Experiment on Multiple UAVs

In this paper, a confrontation scenario for multiple UAVs is set up to verify the effectiveness of the proposed method and the confrontation scene is shown in Figure 11. There are two sides: the red side and the blue side, with four UAVs respectively. The kinematic rules of motion for each UAV are described in the third section. The confrontation rules are as follows: Each UAV is given 100 health points before the start of the round. This article defines health points as life values, similar to agents in real-time strategy games. When the health point (HP) is 0, the UAV is eliminated. Each UAV attacks the opponent with a different energy, which will consume the HP of the UAV itself. One score is given when it hits the other side.

Figure 11. The confrontation scenario for multiple UAVs.

The labels of situation assessment for multiple UAVs are as follows: the difference between the average HP of the two sides, the difference between the remaining numbers of the two sides, the difference between the minimum remaining HP of the two sides and the difference between the scores of the two sides. The situation category is described by five descriptions for situation results, which are very unfavorable to us, relatively unfavorable to us, no effect on both sides, relatively favorable to us and very favorable to us. Firstly, a certain number of training data are collected from the experiment for training, and then the effectiveness of different situation assessment methods are verified by using the validation data. 50 groups of scene data are used as validation data to verify the effectiveness of the method. Situation labels are set empirically as the standard result for comparison. Three methods of situation assessment (SA) are compared for multiple UAVs, which are the SA with the proposed method (SA-P), the SA with the BP model (SA-BP) and the SA with the Fuzzy method (SA-F).

The experimental results are shown in Figures 12–14. In the experiment, the horizontal axis represents the number of groups, and the vertical axis represents the situation assessment results. The three methods use an end-to-end approach to map environmental information to situation assessment results. However, the results of the situation assessment are different. From the experimental results, for the proposed method, the difference between the label results and the actual output results is relatively small. Meanwhile, the results of the situation assessment for the baseline methods are relatively low. In the experiment, the accuracy of the proposed method is 92%, that of method SA-BP is 84%, and that of method SA-F is 61%. The proposed method has the highest accuracy. SA-F method relies on expert experience, so it can not deal with different task scenarios well. SA-BP method is limited by the performance of the conventional neural network, and the immutable situation label cannot adapt to the continuously changing task environment. The proposed method uses the deep neural network to achieve a deeper feature representation and employs a fuzzy layer to make the output results more suitable for the real-time system.

Figure 12. The results for the situation assessment using SA-P.

Figure 13. The results for the situation assessment using SA-BP.

Figure 14. The results for the situation assessment using SA-F.

7. Conclusions

In this paper, a situation assessment method for multiple UAVs using an improved fuzzy deep neural network is developed to improve the performance of the real-time situation assessment system. The conventional rule-based approach needs to focus on different tasks to set different situation assessment rules, which not only costs a lot of time but also is limited by specific tasks. Different from the traditional deep learning methods, the proposed method realizes the end-to-end situation assessment. In the process of situation assessment, the proposed method does not require much expert experience, and will not be limited by the inflexible situation assessment results. Firstly, the normalized scene data is input into the neural network, and the corresponding output results are obtained through the cascade calculation of the neural network. Then, in the output layer, the output situation assessment results are fuzzed to get the degree of truth relative to each situation assessment label. In the training process of the neural network, adaptive momentum and elastic SGD are introduced into the back-propagation process to accelerate the learning speed of the assessment network. The proposed method is more suitable for a real-time situation assessment system because of the fuzziness and uncertainty of a real-time system. Simulation results show the effectiveness of the proposed method.

A learning agent with reinforcement learning modifies its behavior policy using the collected rewards while performing a task in the environment [32,33]. The situation assessment method, based on reinforcement learning may be able to calculate the continuous situation assessment results using the predefined state and action space. In the future, we will investigate the application of reinforcement learning in a situation assessment system for multiple UAVs [34–36]. More realistic task scenarios will be considered for execution in order to test the practicality of the proposed method. In the actual environment, we need to focus on the UAV's visual perception algorithm [37,38], flight planning method [39] and some emergency measures [40] in the face of some uncertain factors. This paper will also investigate a control system integrating the proposed situation assessment method to achieve a more advanced intelligent navigation strategy [41].

Author Contributions: Methodology, L.Z. and X.L.; Resources, L.Z. and X.L.; Software, L.Z., X.S. and X.L.; Supervision, Y.Z.; Writing—original draft, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Civil Aircraft Project under Grant No. XJ-2015-D-76, the major and key project of the Shaanxi Province key research and development plan under Grant No. 2016MSZD-G-8-1 and the ZFPR Foundation of China under Grant No. 31511070301.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vijayavargiya, A.; Sharma, A.A. Unmanned Aerial Vehicle. *Journal. Pract.* **2017**, *168*, 1554–1557.
2. Khawaja, W.; Guvenc, I.; Matolak, D.W. A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2361–2391. [CrossRef]
3. Shi, H.; Hwang, K.S.; Li, X. A learning approach to image-based visual servoing with a bagging method of velocity calculations. *Inf. Sci.* **2019**, *481*, 244–257. [CrossRef]
4. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
5. Shirani, B.; Najafi, M.; Izadi, I. Cooperative load transportation using multiple UAVs. *Aerosp. Sci. Technol.* **2019**, *84*, 158–169. [CrossRef]
6. Fan, Z.; Xiao, Y.; Nayak, A. An improved network security situation assessment approach in software defined networks. *Peer-to-Peer Netw. Appl.* **2019**, *12*, 295–309. [CrossRef]
7. Kim, T.J.; Seong, P.H. Influencing factors on situation assessment of human operators in unexpected plant conditions. *Ann. Nucl. Energy* **2019**, *132*, 526–536. [CrossRef]
8. Kang, Y.; Liu, Z.; Pu, Z. Beyond-Visual-Range Tactical Game Strategy for Multiple UAVs. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; pp. 5231–5236.

9. Hu, C.; Xu, M. Adaptive Exploration Strategy with Multi-Attribute Decision-Making for Reinforcement Learning. *IEEE Access* **2020**, *8*, 32353–32364. [\[CrossRef\]](#)
10. Ji, H.; Han, Q.; Li, X. Air Combat Situation Assessment Based on Improved Cloud Model Theory. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019; pp. 754–758.
11. Shen, L.; Wen, Z. Network security situation prediction in the cloud environment based on grey neural network. *J. Comput. Methods Sci. Eng.* **2019**, *19*, 153–167. [\[CrossRef\]](#)
12. Lin, J.L.; Hwang, K.S.; Shi, H. An ensemble method for inverse reinforcement learning. *Inf. Sci.* **2020**, *512*, 518–532. [\[CrossRef\]](#)
13. Wei, X.; Luo, X.; Li, Q. Online comment-based hotel quality automatic assessment using improved fuzzy comprehensive evaluation and fuzzy cognitive map. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 72–84. [\[CrossRef\]](#)
14. Wu, J.; Ota, K.; Dong, M. Big data analysis-based security situational awareness for smart grid. *IEEE Trans. Big Data* **2016**, *4*, 408–417. [\[CrossRef\]](#)
15. Mannucci, T.; van Kampen, E.; de Visser, C. Safe Exploration Algorithms for Reinforcement Learning Controllers. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1069–1081. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Chen, J.X. The Evolution of Computing: AlphaGo. *Comput. Sci. Eng.* **2016**, *18*, 4–7. [\[CrossRef\]](#)
17. Ernest, N.; Cohen, K.; Kivelevitch, E. Genetic fuzzy trees and their application towards autonomous training and control of a squadron of unmanned combat aerial vehicles. *Unmanned Syst.* **2015**, *3*, 185–204. [\[CrossRef\]](#)
18. Tampuu, A.; Matiisen, T.; Kodelja, D. Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE* **2017**, *12*, e0172395. [\[CrossRef\]](#)
19. Zhang, L.; Zhu, Y.; Shi, H. Adaptive dynamic programming approach on optimal control for affinely pseudo-linearized nonlinear system. *IEEE Access* **2019**, *7*, 75132–75142. [\[CrossRef\]](#)
20. Armenteros, J.J.A.; Tsirigos, K.D.; Sønderby, C.K. SignalP 5.0 improves signal peptide predictions using deep neural networks. *Nat. Biotechnol.* **2019**, *37*, 420–423. [\[CrossRef\]](#)
21. Zhang, J.; Xue, Q.; Chen, Q. Intelligent Battlefield Situation Comprehension Method Based On Deep Learning in Wargame. In Proceedings of the 2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Kunming, China, 17–19 October 2019; pp. 363–368.
22. Kwak, J.; Sung, Y. Autoencoder-based candidate waypoint generation method for autonomous flight of multi-unmanned aerial vehicles. *Adv. Mech. Eng.* **2019**, *11*, 1687814019856772. [\[CrossRef\]](#)
23. Hu, C.; Xu, M. Fuzzy Reinforcement Learning and Curriculum Transfer Learning for Micromanagement in Multi-Robot Confrontation. *Information* **2019**, *10*, 341. [\[CrossRef\]](#)
24. Shao, H.; Zheng, G. Convergence analysis of a back-propagation algorithm with adaptive momentum. *Neurocomputing* **2011**, *74*, 749–752. [\[CrossRef\]](#)
25. Zhang, S.; Choromanska, A.E.; LeCun, Y. Deep learning with elastic averaging SGD. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, Canada, 7–12 December 2015; pp. 685–693.
26. Randel, J.M.; Pugh, H.L.; Reed, S.K. Differences in expert and novice situation awareness in naturalistic decision making. *Int. J. Hum. Comput. Stud.* **1996**, *45*, 579–597. [\[CrossRef\]](#)
27. Endsley, M.R. Situation awareness global assessment technique (SAGAT). In Proceedings of the IEEE 1988 National Aerospace and Electronics Conference, Dayton, OH, USA, 23–27 May 1988; pp. 789–795.
28. Zhang, J.R.; Zhang, J.; Lok, T.M. A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Appl. Math. Comput.* **2007**, *185*, 1026–1037. [\[CrossRef\]](#)
29. Hansen, L.K.; Salamon, P. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 993–1001. [\[CrossRef\]](#)
30. Xu, M.; Shi, H.; Wang, Y. Play games using Reinforcement Learning and Artificial Neural Networks with Experience Replay. In Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 6–8 June 2018; pp. 855–859.
31. Hosmer, D.W.; Lemeshow, S. Goodness of fit tests for the multiple logistic regression model. *Commun. Stat. Theory Methods* **1980**, *9*, 1043–1069. [\[CrossRef\]](#)
32. Shi, H.; Xu, M.; Hwang, K. A Fuzzy Adaptive Approach to Decoupled Visual Servoing for a Wheeled Mobile Robot. *IEEE Trans. Fuzzy Syst.* **2019**. [\[CrossRef\]](#)
33. Shi, H.; Xu, M. A Multiple Attribute Decision-Making Approach to Reinforcement Learning. *IEEE Trans. Cogn. Dev. Syst.* **2019**. [\[CrossRef\]](#)

34. Shi, H.; Shi, L.; Xu, M.; Hwang, K. End-to-End Navigation Strategy with Deep Reinforcement Learning for Mobile Robots. *IEEE Trans. Ind. Inf.* **2020**, *4*, 2393–2402. [[CrossRef](#)]
35. Shi, H.; Xu, M. A data classification method using genetic algorithm and K-means algorithm with optimizing initial cluster center. In Proceedings of the 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 18–20 August 2018; pp. 224–228.
36. Shi, H.; Li, X.; Hwang, K.S.; Pan, W.; Xu, G. Decoupled visual servoing with fuzzy Q-learning. *IEEE Trans. Ind. Inf.* **2016**, *14*, 241–252. [[CrossRef](#)]
37. Gao, H.; Cheng, B.; Wang, J. Object Classification using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment. *IEEE Trans. Ind. Inf.* **2018**, *14*, 4224–4231. [[CrossRef](#)]
38. Gao, H.B.; Zhang, X.Y.; Zhang, T.L. Research of intelligent vehicle variable granularity evaluation based on cloud model. *Acta Electron. Sin.* **2016**, *44*, 365–373.
39. Xie, G.; Gao, H.; Qian, L. Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5999–6008. [[CrossRef](#)]
40. Hongbo, G.; Xinyu, Z.; Yuchao, L. Longitudinal Control for Mengshi Autonomous Vehicle via Gauss Cloud Model. *Sustainability* **2017**, *9*, 2259.
41. Hongbo, G.; Xinyu, Z.; Lifeng, A. Relay navigation strategy study on intelligent drive on urban roads. *J. China Univ. Posts Telecommun.* **2016**, *23*, 79–90. [[CrossRef](#)]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).