Hindawi
Complexity
Volume 2020, Article ID 1764594, 9 pages
https://doi.org/10.1155/2020/1764594

Research Article

Study on the Strategy of Playing Doudizhu Game Based on Multirole Modeling

Shuqin Li,[1,2] **Saisai Li**,[1,2] **Hengyang Cao**,[1,2] **Kun Meng**,[1,2] **and Meng Ding**[1,2]

[1]*School of Computer, Beijing Information and Science and Technology University, Beijing 100101, China*
[2]*Sensing and Computational Intelligence Joint Lab, Beijing Information and Science and Technology University, Beijing 100101, China*

Correspondence should be addressed to Shuqin Li; lishuqin_de@126.com

Received 30 June 2020; Accepted 30 July 2020; Published 20 October 2020

Guest Editor: Zhile Yang

Doudizhu poker is a very popular and interesting national poker game in China, and now it has become a national competition in China. As this game is a typical example of incomplete information game problem, it has received more and more attention from artificial intelligence experts. This paper proposes a multirole modeling-based card-playing framework. This framework includes three parts: role modeling, cards carrying, and decision-making strategies. Role modeling learns different roles and behaviors by using a convolutional neural network. Cards carrying can calculate reasonable rules especially for "triplet" by using an evaluation algorithm. Decision making is for implementing different card strategies for different player roles. Experimental results showed that this card-playing framework makes playing decisions like human beings, and it can to some extent learn, collaborate, and reason when facing an incomplete information game problem. This framework won the runner-up in the 2018 China Computer Game Competition.

1. Introduction

As one important branch of artificial intelligence (AI), computer game is a challenging problem in the broad and deep logical AI decision-making field. It has long been an important verification scenario for various data mining and machine learning algorithms and is known as AI "fruit fly" [1].

The field of computer games is divided into two branches: complete information and incomplete information machine games. Characteristics of the complete information game are that the player can obtain all the situation information completely, for example, in Go [2], chess [3], Chinese chess [4], and Tibetan chess [5]. Characteristics of incomplete information games are that players cannot obtain all or credible situation information during the game. In incomplete information games, the true state of the game environment is often unknowable, and the information held by the players involved is asymmetric and incomplete, which makes the study of incomplete information games more complicated and challenging, for example, poker games such as Texas Hold'em [6], mahjong [7], and Doudizhu [8]. Most applications in the real world are incomplete information games, such as business strategy negotiations, financial investments, bidding strategies, political activities, autonomous driving, medical planning, network security, and military applications.

The traditional computer game mostly focuses on chess games with complete information. Initially, the maximum and minimum search based on depth-first search was used as a general method of game state tree search in computer game systems. Subsequently, the famous Alpha-Beta pruning [9] was proposed and widely used. The search for the maximum and minimum of Alpha-Beta pruning is called Alpha-Beta search. Based on Alpha-Beta search, some excellent and improved algorithms are derived, such as PVS [10], MTD (f) [11], and other algorithms [12, 13] that optimize the search window based on the local principle of search space and data quality-sensitive various heuristic or nonheuristic permutation table optimization methods. In the absence of a full search, the actual effect of Alpha-Beta search is highly dependent on its situation evaluation function. In order to avoid the dependence of the Alpha-Beta search process, especially its situation evaluation process, the Monte Carlo Tree Search (MCTS) algorithm [14, 15] came into being. It uses a large number of random matches to simulate the objective game's winning rate and then solves the game problem. It has good versatility and controllability.

With the breakthrough development of deep learning, models such as deep confidence network (DBN) [16], deep automatic encoding machine (DAE) [17, 18], and deep convolutional neural network (CNN) [19] have been used to successfully solve many problems in the field of computer vision. Especially CNN's superior performance in the field of image pattern recognition and the relatively easy and purely supervised learning and training process make it quickly popular [20–23]. Deep learning technology is known for its powerful map expression ability and excellently completed various regression and classification tasks. No matter in the laboratory or in various practical application scenarios, deep learning technology has the potential to be a core component to optimize the quality and efficiency of computer game systems. The most famous deep learning computer game model belongs to the AlphaGo series of Go computer game systems of the DeepMind team. In 2015, AlphaGo defeated the European Go champion Fan Yu [24], and then in 2016, its reinforcement version AlphaGo Lee defeated world-class Go Master Li Shishi; in 2017, AlphaGo Master defeated World Go champion KeJie in the Open; in the same year, the computer game system AlphaGo Zero [25] and the general chess computer game system AlphaZero were fully trained by unsupervised reinforcement learning methods [26]. It was announced that AlphaZero defeated the strongest existing computer game system in Go, chess, and shogi. AlphaGo is the first integrated deep learning computer game system with remarkable success. It uses both strategy network and value network. A deep convolutional neural network provides reference opinions for decision making and situation evaluation. In addition, these two CNN models first use a large number of professional game data for supervised training and then use a reinforcement learning algorithm based on the DQN algorithm [27].

In a game with incomplete information as opposed to a complete information game, game players have private information, and neither party can get all the state information of the current situation. Therefore, it is impossible to reasonably evaluate the game situation by artificially extracting features. It is difficult to determine the range of actions that the opponent can perform. In addition, the game tree of the incomplete information game is extremely large. Although the Monte Carlo method will search for the optimal path to a certain extent, it still makes the original game algorithm inapplicable to the game with incomplete information.

At present, there are three main ideas about incomplete information games: the first one is based on game theory, through various methods to shrink and create a game tree [28, 29], using a search method similar to complete information game to traverse the game tree and find the best strategy obtained at the equilibrium point [30–32]; the second is based on reinforcement learning and multiagent cooperation, through self-play, learning to formulate game strategies [33–35]. The third is a knowledge-based method. By learning the behavioral characteristics of a large number of professional human players and combining the rules of artificially joining information, the game strategy is finally formulated [36–38].

In this paper, the second multiagent is combined with the third knowledge-based method. Each character is regarded as an agent, which is modeled separately to design and implement different card-playing strategies for different characters. Relying on large-scale historical data, the deep learning method is applied to the Doudizhu poker game.

In Section 2, we will introduce the rules of the Doudizhu game and the overall framework of the Doudizhu game system based on multirole modeling. We will explain each component in Sections 3, 4, and 5 including detailed information on character modeling, carrying cards strategies, and decision making. In Section 6, we will show how to prepare for the experiment and the results of competition with human players. Finally, in Section 7, a summary and problems to be improved will be given.

2. Design of Card Game System Based on Multirole of Doudizhu

2.1. Rules and Common Terms in Doudizhu. Doudizhu is a simple and entertaining traditional Chinese poker game and is usually played by three players. A standard game includes dealing, bidding, playing, and scoring. Three players use a 54-card deck (with two jokers), in which every player can get 17 cards. Three cards are left as hole cards. There are two sides in the game, the Dizhu and the Farmers. After dealing, players bid according to their hand; the player who bids the highest score becomes a Dizhu (the attacker, a game of three contains only one Dizhu). A player becomes the Dizhu by bidding and gets the hole cards, and the other two players become Farmers (defenders, they are allies) to compete with the Dizhu. Then, the players take turns playing cards according to the rules (about the played cards). The side that gets rid of all their cards first wins. Dizhu gets more score than Farmers if he wins. Terms in this paper are defined as follows:

Game: the whole process including dealing, bidding, playing, and scoring is called a game.

Round: several games played by three players are called a round.

Hands: the number of plays to play all according to the rules when the other two players choose to pass every time.

Suit pattern: the suit patterns, patterns for short, are certain combinations of cards that are legal to play in the game such as pass, rockets, bombs, and standard patterns.

2.2. The Overall Framework of the Card Game System of Doudizhu. Each player needs to constantly change judgments and dynamically choose his own strategy based on his role, the relationship of other participants relative to himself, and the actual actions he observes of other participants. The design of the playing system of the three characters in the Doudizhu is shown in Figure 1. The Doudizhu framework is divided into three parts: role modeling, carry cards strategy, and decision making.

In Figure 1, the "history data," which are based on the human poker player provided by a well-known website, are first divided into two different datasets according to Dizhu and Farmers. Those data are used for subsequent model training and verification; the "role modeling" uses convolutional neural networks to model Dizhu, Farmer 1, and Farmer 2 according to different training data and learns the behaviors of different characters; the "banding strategy" is mainly for the "three belts" card type, and it is reasonable to use valuation algorithms to learn. The licensing rules of different roles are the same; the "decision making" is to give different strengths of playing strategies for three different roles of different Dizhu, Farmer 1, and Farmer 2 to reflect a higher level of cooperative confrontation. The following sections introduce the design and implementation of role modeling, licensing strategy, and decision making.

3. Modeling and Design of Doudizhu Game Based on Convolutional Neural Network

The modeling of multirole in this paper includes two aspects: (1) separation of training data and (2) different decision-making methods of playing cards. This article divides the historical card-playing data of the platform into two parts according to the role of the final winning player, which are the data of Dizhu win and Farmer win, respectively. Use the Dizhu winning data to train the Dizhu model and use the Farmers' winning data to train Farmer 1 and Farmer 2 models, respectively. See Section 5 for the realization of card strength. Multirole modeling is implemented using deep convolutional neural network (CNN).

3.1. CNN Model Input Format Design. In the Doudizhu game, the game participants have private information and cannot get all the status information of the current situation. Although they only know the characteristics, strategy space, and income function information of some other participants, they are not aware of their opponents' cards. It is a state that is not fully understood, and as players make various operations, the information that can be learned will gradually increase, and the estimates of other players' hands will be gradually accurate.

The information provided to the neural network model should be complete and not redundant. If playing cards do not consider suits, there are 15 different card information, namely, the numbers A, 2–10, J, Q, K, black joker, and red joker. In this article, enter "A23456789TJQKXD" in the following order to enter the player's hand information, where "T" means "10," "X" means "black joker," and "D" means "red joker".

In order to fully reflect the advantages of convolutional neural networks, in the representation of the model input data, it is necessary to not only show the current game state but also include the historical sequence of operations and reflect the player's confrontation relationship. To this end, this article contains the following five aspects of information in each model input (game state) of a single character.

$$N_{\text{in}} = \left\{N_{\text{all}}, N_{\text{player}}, N_{\text{rest}}, N_{\text{history}}, N_{\text{rounds}}\right\}, \qquad (1)$$

where N_{all} represents the total card of the Dizhu's game; N_{player} represents the remaining hands of the current player; N_{rest} represents the unknown hands (the sum of other players' hands); N_{history} represents the total number of historical cards; and N_{rounds} represents the number of rounds from the current state forward card data. This article uses the first 5 rounds of data, a total of 9 sets of data.

The confrontation and cooperation of the game are reflected in the input channel and are arranged in the order of the Dizhu, Farmer 1, and Farmer 2.

Therefore, the input size of the CNN model is a $[9 \times 15 \times 3]$ matrix, where "9" means N_{in}, "15" means card information, and "3" means three different player data of Dizhu, Farmer 1, and Farmer 2.

3.2. CNN Model Output Format Design. The model's output is the way to play. This section mainly considers the 8 kinds of action, such as "pass," "bomb," "single," "pair," "triplet," "single sequence," "double sequence," and "triplet sequence." The carrying card type is more complicated and will be discussed separately in Section 4.

Corresponding to 8 kinds of action, this paper further divides the way to play cards into 182 types, as shown in Table 1. Each card-playing method is represented by a single vector, and the corresponding position of 15 cards is marked with "1," and the remaining positions are marked with "0." The way of playing cards is manifested in the form of different probability distributions, and the one with the highest probability value is the strategy of the round of playing cards.

3.3. Role Model Design Based on CNN. The model uses a convolutional neural network, which consists of 9 convolutional layers, 2 fully connected layers, one batch normalization (BN) [39] layer, and an output layer. As shown in Figure 2, after the data sample is input, it will pass through 9 convolutional layers.

The number of convolution kernels from the first layer to the third layer is 64, 128, and 196, the size of the convolution kernels is 5×5, 3×3, and 3×3, the number of the remaining 6 layers of convolution kernels is 256, the size of the convolution kernels is 3×3, the horizontal and vertical movement steps of the convolution kernels are 1, and 0 is added around the output matrix after the convolution operation to maintain the input time. The data size of 9×15 remains unchanged. The activation function after the convolution operation uses ReLu (rectified linear units) [40]. No downsampling operation is performed after the convolution operation. After 9 convolutional layers, the data enter 2 fully connected layers. The number of neurons in each fully connected layer is 256, and the nonlinear activation function still uses the ELU function. Finally, the data enter the

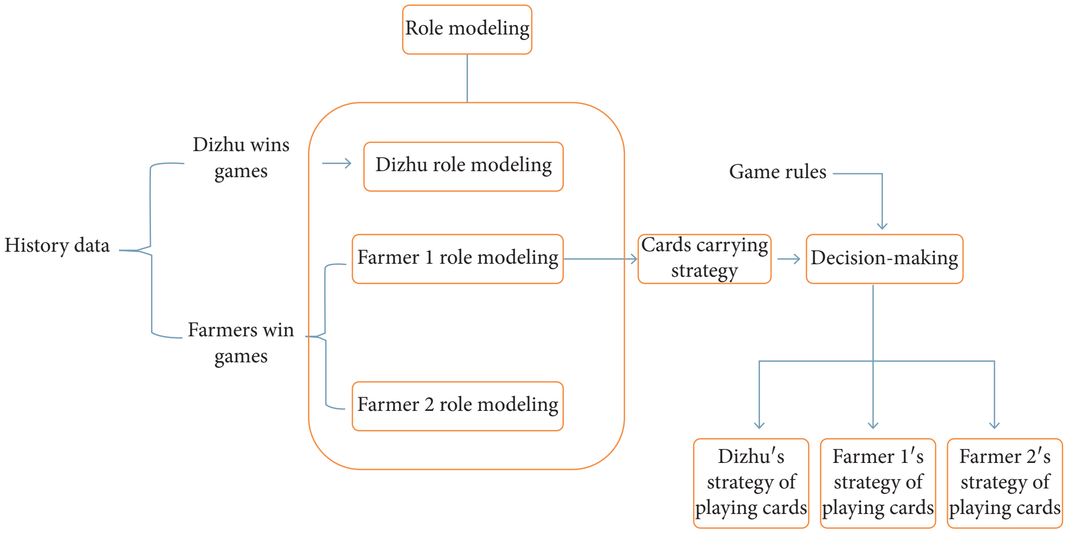

Figure 1: The overall framework of the Doudizhu playing card system.

Table 1: List of ways to play cards.

Suit pattern	Description	Counter
Pass	Choose not to play card this turn. It is also called as a trivial pattern.	1
Rocket	Same as the joker bomb, both jokers (red and black) are the highest bombs.	1
Bomb	Four cards with same points (e.g., AAAA).	13
Single	One single card (e.g., A).	15
Pair	Two cards with same points (e.g., AA).	13
Triplet	Three cards with same points (e.g., AAA).	13
Single sequence	Five of more singles in sequence excluding 2 and jokers (e.g., ABCDE or ABCDE...).	36 (=8 + 7+6 + 5+4 + 3+2 + 1)
Double sequence	Three of more pairs in sequence excluding 2 and jokers (e.g., AABBCC or AABBCC...).	52 (=10 + 9+8 + 7+6 + 5+4 + 3)
Triplet sequence	Two of more triplets in sequence excluding 2 and jokers (e.g., AAABBB or AAABBBCCC...).	38 (=11 + 10+9 + 8)

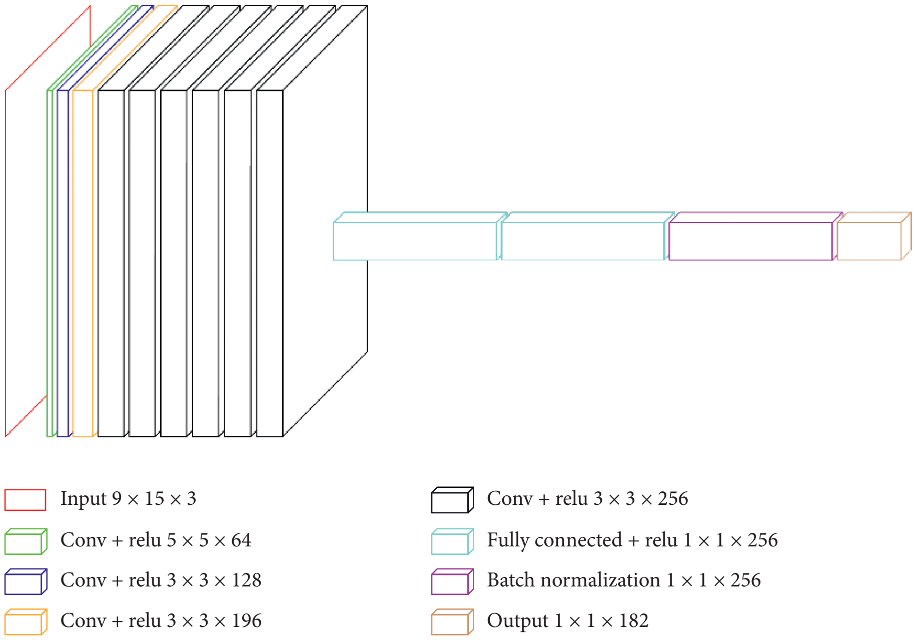

Figure 2: Role-based CNN network structure model.

output layer after entering a BN layer, which contains 182 neuron structures. The data entering the output layer do not need to go through a nonlinear activation function. The Adam algorithm [41, 42], which is more stable than the stochastic gradient descent algorithm, is selected, and it iteratively optimizes the convolution kernels of all convolutional layers in the network and the connection weight values of neurons in the fully connected layer according to the error between the network output and the expected value. The model output is normalized by the sigmoid function and falls in the interval [0, 1].

4. Carry Cards Strategy Design

There is a special card type in the Doudizhu game, which is divided into three carry cards and four carry two cards. The specific explanation is shown in Table 2.

From the explanation in Table 2, it can be seen that the card type is more complicated. It is based on the "triplet" and "triplet sequence" card types and fully considers the current hand card information to decide which card types are more appropriate.

First, split the opponent cards according to the "rocket," "bomb," "sequence," "pair," "single," and other card types and then count the number of various card types and use the valuation algorithm. Calculate the estimated size returned by multiple split branch nodes and select the maximum node as the final card type.

The valuation algorithm mainly considers the following points:

(1) Consider whether you can finish your hand after you carry the card. If you can finish the hand, choose this operation directly.

(2) Consider the degree of threat to opponents by different card types; the "bomb," "sequence," and other card types are assigned different value weights from high to low.

(3) Consider that the number of "three belts" can offset the number of "single cards" and "pairs"; the more the number of offsets, the better.

(4) It is stipulated that the number of "three belts" cannot offset the number of "single cards" generated by the licensed system.

(5) When calculating the value of "straight," consider some special split situations. For example, splitting "3455667789" into "34567" and "56789" is the best calculation method.

(6) Consider the value of the single card of the large digital board, that is, the value of "A," "2," "X," and "D."

The value V of each card type in the hand is calculated as shown in formula (2), that is, the square of the card type coefficient α is multiplied by the number of card types.

$$V_i = \alpha_i^2 * N_i, \tag{2}$$

where N_i represents the number of different cards, α_i is the coefficient of different cards, and $i \in \{\text{bomb}, \text{sop}, \text{seq}, \text{tri}, \text{sc}, \text{pair}, \text{zhu}\}$. In this paper, the α_{bomb} value of the "bomb" is set to 8; the "Lian Pair" α_{sop} value is 6; "straight" α_{seq} value is 5; "three belts" α_{tri} value is 3; separate "pair," "single card," and the main card's a value is 1.

The face value of different hands is the sum of the included card values. The calculation is shown in the following equation:

$$\begin{aligned}\text{Value} = V_{\text{over}} + V_{\text{bomb}} + V_{\text{sop}} + V_{\text{seq}} + V_{\text{tri}} \\ - V_{\text{sc}} - V_{\text{pair}} + V_{\text{zhu}},\end{aligned} \tag{3}$$

where V_{over} indicates whether the hand is finished after the card is brought, and the default value is 0. If the hand can be played, a larger value is returned directly, such as 9999.

For the different characters in the Doudizhu game, the strategy of carrying cards is the same, and the value with the highest valuation is selected.

5. Decision-Making Design Based on Multirole Modeling

The output of the CNN model we gave in Section 3 is the probability distribution of the players' different card-playing strategies, and the maximum probability play card is directly used as the final card-playing strategy. Sometimes, certain errors will occur. For example, the previous player played the "straight" card type and the current player's maximum card probability may be "connected pairs," but it is against the rules. Therefore, according to the probability value, this paper selects 5 card types from large to small and chooses the one that satisfies the rules of the game as the strategy, instead of only considering the card types with maximum probability.

On the basis of multirole modeling, this section further refines the strategy for different players. Combining role modeling and card-bearing strategies, plus the confrontational and cooperative relationship between players, different levels of playing strength are used to generate final playing decisions for different characters. The specific settings are as follows:

(1) For the role of Dizhu, the strategy is to directly select the maximum probability of playing cards as the final playing strategy.

(2) For the role of Farmers, because the Farmers' strategy contains a large number of "pass" operations, although the cooperation relationship is reflected to a certain extent, too many such operations will cover some correct methods, especially when the Farmers have fewer cards, and when the card power is small, the "pass" operation of other players cannot effectively increase the player's chance of playing cards. For example, in a situation where Farmer 2 has a remaining "4" card, if the Dizhu plays a "5" card, the strategy probability of Farmer 1 is [0.32, 0.25, 0.19, 0.11, 0.09], representing ["pass," "2," "9," "Q," "6"]; at this time, the maximum probability is the "pass" operation, and the second probability is within 0.1 of the maximum probability, so choose to play the "2" plate as the best strategy. Therefore, the

Table 2: Ways to carry cards.

No.	Suit pattern	Description
1	Triplet carry one cards	Three cards of the same point with 1 single card or 1 pair of cards (such as AAA + B or AAA + BB)
2	Triplet carry cards	Triplets carry the same number of single cards or the same number of pairs (types such as AAABBB + C + D or AAABBB + CC + DD or AAABBB... +...+Y + Z or AAABBB... +...+YY + ZZ)
3	Bomb carry cards	Four cards of the same point with 2 single cards or 2 pairs of cards (types like AAAA + B + C or AAAA + BB + CC)

final strategy selection method for Farmers is shown as follows:

$$\text{play} = \begin{cases} \text{Max}\,(x - \delta, y), \\ z. \end{cases} \tag{4}$$

where x represents the maximum probability,y represents the second probability for a "pass" strategy, z indicates the maximum probability that the strategy is not a "pass", and δ represents the card strength (δ is taken as 0.1 in this paper). When the maximum probability card strategy is "pass" and if the difference between the second probability and it is within 0.1, the card strategy represented by the second probability is selected. The larger the d value, the more the Farmer's card strategy which tends to avoid "pass" operations.

To sum up, this paper combined multirole modeling with the card-carrying strategy, considered the antagonism and cooperation between players, used different levels of card strength, and generated the final card strategy of different roles.

6. Experimental Results and Analysis

The server configuration of the training environment is Ubuntu 16.04.2 LTS operating system, NVIDIA GeForce GTX TITAN X graphics card, 12 GB video memory, and Tensorflow version 1.0.0. The data come from a real-time game record of a live-action platform on a well-known website in China, including the initial hand of the game and the detailed card process. Of the 5 million Games selected, 3 million were won by Dizhu, and 2 million were won by farmers.

This paper, respectively, conducted experiments on multirole model, multirole card-playing performance, and card strategy performance, analyzed the effects, and proposed ways to improve the problems.

6.1. Implementation of Multirole Modeling. The experiment of multirole modeling mainly shows the training effect of different role models of Dizhu, Farmer 1, and Farmer 2. The model is trained on a high-performance graphics card with a batch size of 100 and a learning rate of 0.001. The training results are shown in Figures 3–5, which show the changes in the accuracy rate of the Dizhu model, Farmer 1 model, and Farmer 2 model as the training data increase. The horizontal axis is the number of iterations, and the vertical axis is the similarity of the network output strategy and the actual player strategy, that is, the correct rate.

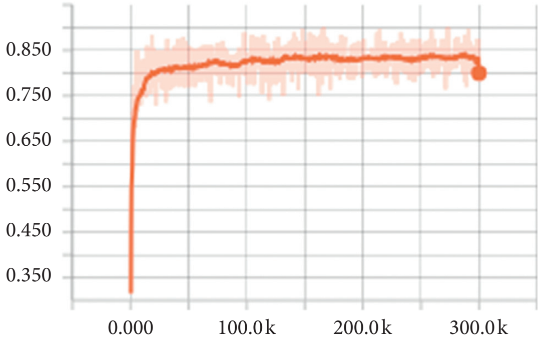

Figure 3: Changes in the accuracy rate of the Dizhu model.

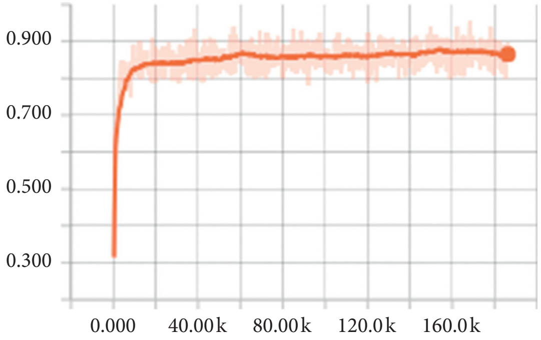

Figure 4: Changes in the accuracy rate of Farmer 1 model.

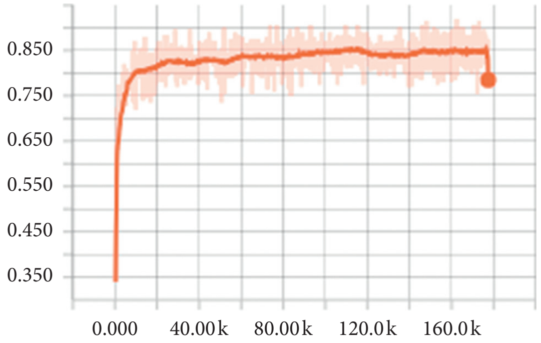

Figure 5: Changes in the accuracy rate of Farmer 2 model.

The experimental results show that the similarity between the output strategy of the three player models and the real player strategy is around 85%. It shows that the model has extracted certain game state features, and the selection of the current playing strategy is to some extent similar to real players. In addition, the statistics of the three character

Table 3: Game flow.

Cards information (0, 1, 2)	Self-gaming process	AI is the Dizhu, and humans are the Farmers	Human is the Dizhu, and AI are the Farmers

334566789JJQQKAXD;
34456789TTKAAA222;
345577889TTJQQKK2; 69J
	0, 33; 1, TT; 0, QQ; 2, KK; 2, 55; 0, 66; 2, 77; 0, 99; 2, TT; 2, 88; 2, QQ; 2. 3; 0, A; 2, 2; 0, X; 0, 45678; 1, 56789; 1, 3; 2, 4; 0, D; 0, JJJK	0, 33; 1, TT; 0, QQ; 2, KK; 2, J; 0, K; 1, 2; 1, 56789; 1, 44; 2, TT; 2, Q; 0, A; 1, 2; 1, K; 1, AAA3; 0, XD; *0, JJJ66*; 0, 456789; 0, 9	0, 456789; 1, 56789T; 1, 3; 2, 2; 2, 55; 0, 66; 2, 77; 0, QQ; 1, 22; 1, T; 2, K; 0, A; 1, 2; 0, X; 0, K; 1, A; 0, D; 0, JJJ33; 0, 9

35577789JJJQKA22D;
44566889TTTJQKK2X;
3334567899TQQKAAA; 462
	0, 3456789; 2, 456789T; *2, 3339; 0, JJJ5*; 2, AAAK; *0, 222Q*; 0, 77; 1, KK; 1, 4; 2, Q; 0, A; 1, 2; 0, D; 0, K	0, 3456789; 2, 456789T; **2, 3339;** 0, JJJ5; 2, AAAK; 0, 222Q; 0, K; 1, 2; 1, 44; 2, QQ	0, Q; 1, 2; 1, 44; 2, QQ; 2, 456789T; **2, 3339;** 0, JJJ5; 2, AAAK

678899TJQQKKAA22X;
34455566779TTQA22;
33345679TJJJQKKAD; 488
	0, 4; 1, 9; 2, K; 0, 2; 0, 6; 1, Q; 0, K; 1, A; 0, 2; 0, 7; 1, 2; 0, X; 2, D; 2, 34567; 0, 9TJQK; 2, TJQKA; 0, 8888; 0, 9; 1, 2; 1, 34567; 1, 55; 2, JJ; 0, AA; 0, Q	0, 4; 1, 9; 2, K; 0, 2; 2, D; 2, 34567; 2, 9TJQKA; 2, 33; 0, AA; 1, 22; 1, 44; 2, JJ	0, 4; 1, 9; 2, K; 0, A; 1, 2; 0, X; 2, D; 2, 34567; 0, TJQKA; 0, 6; 1, Q; 0, K; 1, A; 0, 2; 0, 7; 2, A; 0, 2; 0, 99; 0, 8888; 0, Q

Table 4: Carrying card strategy.

Position	Situation	Carry cards type	Output strategy
Table 4 item 1 data “JJJ66”	456667899	Single or pairs	66
Table 4 article 2 data “3339”	AAA9QQK	Single	9
Table 4 item 2 data “JJJ5”	A222577QKD	Single	5
Table 4 article 2 data “222Q”	A77QKD	Single	Q

players’ playing cards are found, and it is found that the Farmer strategy has more “checking” operations than the Dizhu strategy, indicating that the two Farmers are cooperative, and the Farmer players often provide better teammates.

6.2. Multirole Card Performance Test. This experiment mainly tests whether the three card-playing models (hereinafter referred to as AI) can draw an appropriate card-playing strategy according to the current situation, which reflects the confrontation and cooperation relationship. In order to test the intelligence of AI, the same game was set up in three ways to play: (1) three-role AI Program for self-game; (2) AI for the Dizhu and human for Farmers to game; and (3) human for the Dizhu and two AIs for Farmers to game. The similarity between AI and human strategies is observed in a particular game. The game situation is shown in Table 3.

“Cards information” in Table 3 indicates the initial situation of the game in the order of “Dizhu’s initial hand, Farmer 1’s initial hand, Farmer 2’s initial hand, and the bottom card,” where “0” indicates Dizhu, “1” indicates Farmer 1, and “2” means Farmer 2. In different game processes, “0, 33” indicates that the Dizhu played “33.” If the player chooses the “pass” strategy, it will not be recorded.

The game process shows that AI has the characteristics of cooperation, card removal, and card combination:

(1) In the wavy part of the second data, when the player at position 2 plays “QQ,” even if the player at position 1 has a larger card, the AI chooses “pass” to increase the chance of playing at position 2.

(2) In the thick line part of the second data, human players and AI face the same situation and play the same card type.

(3) In the wavy part of the third data, the player at position 2 will disassemble the “JJJ” and give priority to the “TJQKA” card type.

The data show that the game program implemented based on the method in this paper is very similar to the playing habits of human players and can perform some reasonable combination of card operations, as well as co-operation between the two Farmers.

6.3. Carrying Card Strategy Performance Implementation and Testing. According to the game process listed in Table 3, further analysis of the implementation of the card strategy is mentioned in this article. We selected four cases from Table 3 (see double-underlined position) , focusing on AI's strategy, summarized in Table 4.. Among them, “situation” means the current player’s hand situation when faced with the use of the card strategy; “type of card” is the type of the card that can be selected; “output strategy” is the type of card that is finally recommended by the card strategy.

The analysis found the following:

(1) In ordinary situations, such as the situation where some “single cards” or some weak cards exist alone, the card strategy can find such cards well, such as when the “3” card exists alone. The strategy with license will be output first.

(2) In a special situation, such as a situation with a combination of “sequence” and “triplet”, the card-licensing strategy still has good performance, as shown in Table 4.

(3) In a special situation, the card-licensing strategy will give priority to the output of scattered cards and will not destroy the combination of key cards such as "sequence" and "triplet". For example, in the first data of Table 3, the game situation faced by AI is "456667899", which includes the "sequence" card type. The output of the strategy with the card does not destroy this card type and even "intentionally" uses other card types to play and create "sequence" cards.

In a word, the experiment shows that the card-licensing strategy can make a more reasonable strategy when facing different situations.

7. Conclusion

From the perspective of incomplete information games, this paper proposes a complete game framework for Doudizhu, fully considering the confrontation and cooperation in the Doudizhu game, models separately according to the player's role, and fully reflects the game information and rules on the CNN model input representation. This article elaborates on the complete game method of the Doudizhu game of "player modeling strategy with card decision making," supplemented by specific examples. In the final decision-making section, this paper discusses a number of key factors that affect decision making and uses different levels of card strength for different players. This program has won the runner-up in the 2018 China Computer Game Contest, which shows that the multirole modeling strategy proposed in this paper is feasible.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by key potential projects of promoting research level program at Beijing Information Science and Technology University (no. 5212010937), by Normal Projects of General Science and Technology Research Program (no. KM201911232002), and by Construction Project of Computer Technology Specialty (no. 5112011019).

References

[1] X. Xinhe, Z. Deng, and J. Wang, "Challenging issues facing computer game reseach," *CAAI Transactions on Intelligent Systems*, vol. 3, no. 4, pp. 288–293, 2008, in Chinese.

[2] S.-J. Yen, T.-N. Yang, C. Chen, and S.-C. Hsu, "Pattern matching in go game records," in *Proceedings of the Second International Conference on Innovative Computing, Information and Control (ICICIC 2007)*, p. 297, Washington, DC, USA, October 2007.

[3] Y. HaCohen Kerner, "Learning strategies for explanation patterns: basic game patterns with application to chess," in *Proceedings of the International Conference on Case-Based Reasoning*, pp. 491–500, Sesimbra, Portugal, October 1995.

[4] D. Meng and Z. Yipeng, "Optimization methods for boundary judgment in Chinese chess game program," *Journal of Beijing Information Science & Technology University 2016*, vol. 31, no. 6, pp. 19–22, 2016.

[5] X. Li, Z. Lv, B. Liu, L. Wu, and Z. Wang, "Improved feature learning: a maximum-average-out deep neural network for the game go," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1397948, 6 pages, 2020.

[6] S. Wu, Research on the Opponent Model in Texas Hold'em *Harbin Institute of Technology*, Harbin, China, 2013.

[7] N. Mizukami and Y. Tsuruoka, "Building a computer Mahjong player based on Monte Carlo simulation and opponent models," in *Procerdings of the 2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 275–283, Tainan, Taiwan, September 2015.

[8] S. Li, S. Li, M. Ding, and K. Meng, "Research on fight the landlords' single card guessing based on deep learning," in *Procerdings of the International Conference on Artificial Neural Networks*, pp. 363–372, Cham, Switzerland, October 2018.

[9] J. Pearl, "The solution for the branching factor of the alpha-beta pruning algorithm and its optimality," *Communications of the ACM*, vol. 25, no. 8, pp. 559–564, 1982.

[10] A. Reinefeld, "An improvement to the scout tree search algorithm," *ICGA Journal*, vol. 6, no. 4, pp. 4–14, 1983.

[11] A. Plaat, J. Schaeffer, W. Pijls, and A. De Bruin, "Best-first fixed-depth minimax algorithms," *Artificial Intelligence*, vol. 87, no. 1-2, pp. 255–293, 1996.

[12] Z. Feng and C. Tan, "Subgame perfect equilibrium in the rubinstein bargaining game with loss aversion," *Complexity*, vol. 2019, Article ID 5108652, 23 pages, 2019.

[13] J. Lee and Y.-H. Kim, "Epistasis-based basis estimation method for simplifying the problem space of an evolutionary search in binary representation," *Complexity*, vol. 2019, Article ID 2095167, 13 pages, 2019.

[14] G. Chaslot, S. Bakkes, and I. Szita, "Monte-Carlo tree search: a new framework for game AI," in *Procerdings of the AIIDE*, Palo Alto, CA, USA, October 2008.

[15] T. Pepels, M. H. M. Winands, M. Lanctot, and M. Lanctot, "Real-time Monte Carlo tree search in ms pac-man," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 6, no. 3, pp. 245–257, 2014.

[16] G. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[17] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[18] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.

[20] S. Hochreiter, Y. Bengio, and P. Frasconi, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Networks*, pp. 237–243, Wiley-IEEE Press, Hoboken, NJ, USA, 2001.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] D. Steinkrau, P. Y. Simard, and I. Buck, "Using GPUs for machine learning algorithms," in *Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pp. 1115–1119, Seoul, South Korea, August 2005.

[23] F.-P. An, "Pedestrian re-recognition algorithm based on optimization deep learning-sequence memory model," *Complexity*, vol. 2019, Article ID 5069026, 16 pages, 2019.

[24] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[25] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[26] D. Silver, T. Hubert, and J. Schrittwieser, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," 2017, http://arxiv.org/abs/11712.01815.

[27] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[28] N. Brown and T. Sandholm, "Simultaneous abstraction and equilibrium finding in games," in *Proceedings of the International Conference on Artificial Intelligence*, pp. 489–496, 2015.

[29] T. Sandholm, "Abstraction for solving large incomplete-information games," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 4127–4131, Austin, TX, USA, January 2015.

[30] N. Brown and T. Sandholm, "Libratus: the superhuman AI for no-limit poker," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 5226–5228, Melbourne, Australia, August 2017.

[31] N. Brown and T. Sandholm, "Safe and nested subgame solving for imperfect-information games," 2017, http://arxiv.org/abs/1705.02955.

[32] S. Ganzfried and T. Sandholm, "Improving performance in imperfect-information games with large state and action spaces by solving endgames," pp. 46-47, 2013.

[33] Microsoft research Asia, "Microsoft super Mahjong AI Suphx, crack the imperfect information game (DB/OL)," 2019, https://www.msra.cn/zh-cn/news/features/mahjong-ai-suphx.

[34] M. Wang, T. Yan, M. Luo, and W. Huang, "A novel deep residual network-based incomplete information competition strategy for four-players Mahjong games," *Multimedia Tools and Applications*, vol. 78, pp. 1–25, 2019.

[35] B. Liu, N. Xu, H. Su, L. Wu, and J. Bai, "On the observability of leader-based multiagent systems with fixed topology," *Complexity*, vol. 2019, Article ID 9487574, 10 pages, 2019.

[36] L. F. Teófilo, N. Passos, L. P. Reis, and H. L. Cardoso, "Adapting strategies to opponent models in incomplete information games: a reinforcement learning approach for poker," *Autonomous and Intelligent System*, vol. 7326, pp. 220–227, 2012.

[37] M. Moravčík, M. Schmid, N. Burch et al., "DeepStack: expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, 2017.

[38] S. Li, R. Wu, and B. Jianbo, "Study on the play strategy of Doudizhu poker based on convolution neural network," in *Proceedings of the 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, pp. 702–707, Shenyang, China, October 2019.

[39] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[40] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323, Fort Lauderdale, FL, USA, April 2011.

[41] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, http://arxiv.org/abs/1412.6980.

[42] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CA, USA, August 2016.