ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

Quantitative evaluation of machine learning capability based on a differential game problem

Haolong Wei[1], Wang Yan[2], Yueyang Wang[3]

[1] Westa College，Southwest University，Chongqing,Beibei,400715

[2] Westa College，Southwest University，Chongqing,Beibei,400715

[3] Westa College，Southwest University，Chongqing,Beibei,400715

Abstract

With the development of artificial intelligence, various machine learning algorithms have emerged, but in the past, it was very inefficient to use machine learning methods to solve problems such as optimal control and problems of game theory, so a method that can quantitatively evaluate the machine learning capability is proposed, which can significantly improve the efficiency of existing algorithms. In this paper, a differential game problem based on the sheep-dog two-dimensional motion model is established and five related questions are answered according to the constraints in the problem. Meanwhile, the algorithm is analyzed quantitatively and the corresponding results are obtained. The paper also verifies that several models, equations of motion and machine learning algorithms established in this paper are reliable through sensitivity analysis.

Keywords

optimal strategy, adversarial neural network, game theory model, machine learning, quantitative evaluation.

1. Problem Overview

1.1. Background of the problem

With the development of technology and the increasing popularity of artificial intelligence, various machine learning algorithms have emerged and are continuously optimized, but since the efficiency (effectiveness/arithmetic power) is very low when solving problems with continuous actions and states such as optimal control and differential games by machine learning methods, the special differential game problem will be a competitive case for testing machine learning methods as a way to generate adversarial conditions, which in turn makes a set of generators with different purposes play constantly with the adversaries, making the algorithm more efficient. The following problem is designed through a realistic context.

An old sheep-dog problem of game theory: A sheep has a constant rate v within a circle of radius R and an arbitrary turning ability satisfying the following restriction: the distance between each point on the escape path and the center of the circle is monotonically non-decreasing in time. The sheep wins if it escapes from the circle. The dog is blocked along the circumference of the circle at a constant rate V to prevent the sheep from escaping and has the ability to choose one of the two directions of the circle at any moment.

1.2. Problem Overview

The following five problems are solved by combining the sheep-dog game with a realistic context.

1. solving the dog's optimal containment strategy by kinematically accurate modeling.

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

2. to solve the conditions under which the sheep can escape to win based on exact modeling assuming that the dog is besieged with an optimal strategy.

3. assuming that the sheep understands its capabilities, limitations and the goal of escaping from dog containment, but does not have kinematic-based knowledge of optimal decision making, and that the condition in Assumption 2 that the sheep can escape is satisfied, give a machine learning method that enables the sheep to escape through learning training.

4. design an evaluation system to quantitatively evaluate the learning ability of the machine learning method given in 3.

5. propose and quantitatively evaluate more sheep escape machine learning methods.

2. model assumptions

1. Assume that the dog's roundup range is the range of the circle with itself as the origin, and the sheep's escape range is also the circle with itself as the origin, and the dog's half pound is greater than the sheep's half pound.

2. Assume that the escape speed of the sheep is less than the roundup speed of the dog.

3. Because the escape of sheep and dog roundup are in the two-dimensional plane, assuming that sheep and dogs are in the two-dimensional plane of movement, do not consider the three-dimensional jump problem, as well as the three-dimensional height of the dog and sheep problems on the impact of the experimental results.

4. It is assumed that both sheep and dogs are healthy in this confrontation exercise, and no individual disease or death occurs.

5. Assume that the data in the questions are correct and can objectively reflect the nature of the problem.

3. Symbols

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

Table 1 Symbol description data

symbol	explanation
R	Game field radius
r_y	Sheep's range radius
r_q	Dog's range radius
pdist	Distance function
d	Euclidean distance
θ	angle
G	Builder
D	Discriminator
j	Number of iterations
APP	Accuracy
P	Precision value

4. Model building and solving

4.1. Model I sheep-dog movement model

First, a sheep escape motion simulation model is established. In this paper, we use known motion constraints and data to build a sheep escape motion model to speculate the motion change of the predated target in the future period. Second, a dog pursuit motion simulation model is established and superimposed on the sheep escape motion model. Through motion physics and mathematical formulas, this paper describes the correlation between dog pursuit motion and sheep escape motion, and then determines the program strategy of dog pursuit motion based on the escape motion route of the target sheep in a future period of time. Finally, an optimized canine pursuit and containment motion model is established. And the optimal strategy of canine roundup is solved based on the randomly generated sheep movement data, and the changes in response to such strategy under certain objective conditions are discussed.

4.1.1. Modeling

Since both the escape motion of sheep and the canine roundup motion can be described in the two-dimensional plane, and the three-dimensional heights of sheep and dogs can be neglected, they have no effect on the problem. Therefore, this paper transforms the problem into a two-dimensional plane model and establishes a two-dimensional motion model.

First of all, this paper sets the motion field, and this paper takes a fixed point as the center of circle and R as the radius to restrict the roundup and escape motion field. Again, this paper defines the dog's range of containment and the sheep's range of escape, assuming that the sheep's range of escape radius is: r_y, the dog's range of containment radius is: r_q, and there is a

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31
connection between the two range radii, and there is a connection with the range radius of the field R as shown in the following equation：

$$6r_q = 8r_y$$

$$R = 10r_y$$

The corresponding two-dimensional motion diagram is shown in Figure 1.

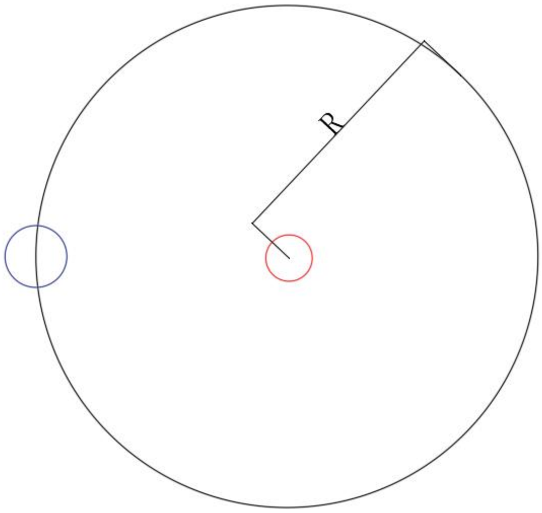


Figure 1 Two-dimensional diagram of motion

In which, the large black circle in the figure is the gaming field of radius R, the small red circle of radius R is the effective radius of the sheep's escape activity, and the blue one is the effective radius of the dog's roundup activity range is: r_q. And where the dog can only move on the circumference of the circle, and the direction is only clockwise and counterclockwise; while the sheep can move in any direction in the activity field within the radius of R, and the initial position of the sheep within the radius is also randomly generated, not fixed. When any point within the radius r_y of the sheep is out of the circumference of the field R (i.e., when it starts to tangent), the sheep is considered to have won the escape, and when any point within the radius r_q of the dog's activity touches the sheep's activity range, when the two circles are just tangent, the dog is considered to have won the roundup.

After the model is established, we start to initialize the model and carry out the solution.

1. Randomly initialize the positions of the dog and sheep, detect several times, and return the best position of the individual dog to prepare for the roundup, and the detection formula is as follows.

$$X_{i+1} = X_i \pm Mc \cdot r$$

is the maximum distance of each activity, $r \in [-1,1]$.

2. Grouping, using the FCM algorithm to make the grouping as uniform as possible, to get the following Figure 2 shows.

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

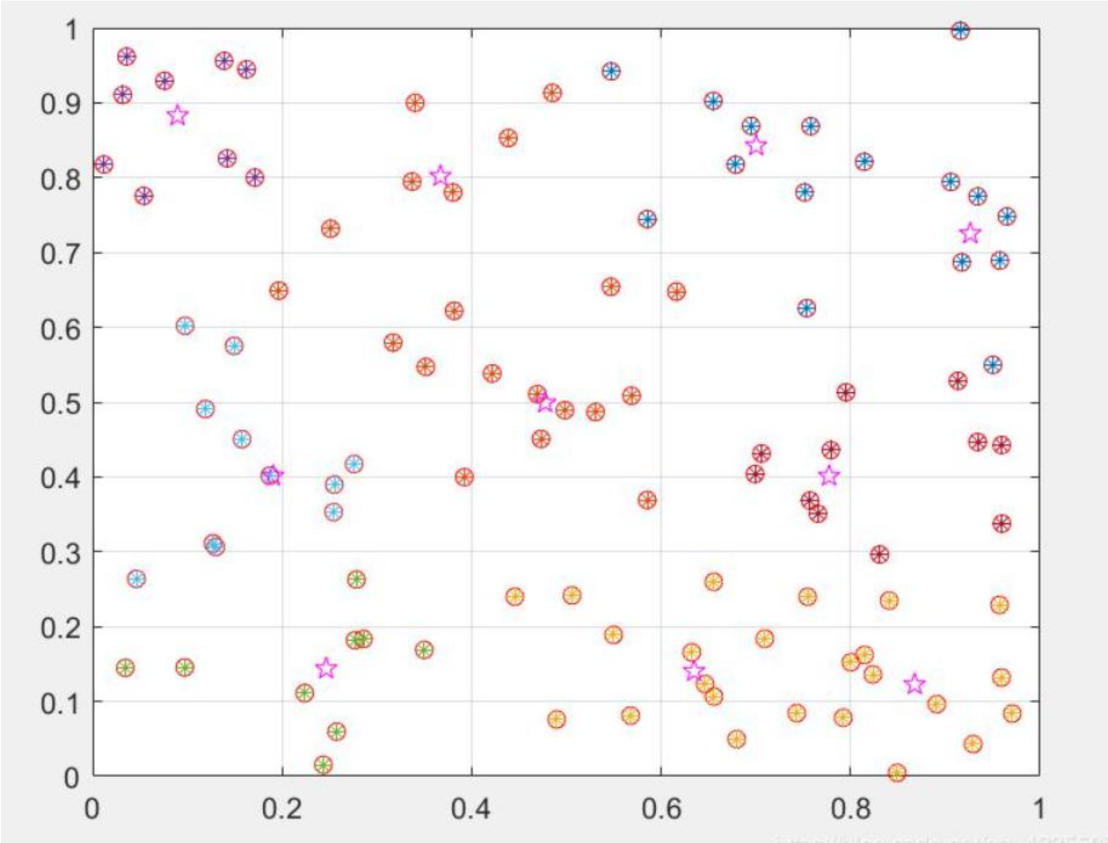


Figure 2 FCM clustering

The clustering center formula.

$$P(u_i \backslash x_j) = \frac{(1/d_{ij})^{1/(b-1)}}{\sum_{r=1}^{c} P(1/d_{rj})^{1/(b-1)}}$$

$$u_i = \frac{\sum_{j=1}^{N} P(u_i \backslash x_j)^b x_j}{\sum_{j=1}^{N} P(u_i \backslash x_j)^b}$$

Where, d is the Euclidean distance, P is the affiliation matrix, μ is the clustering center update formula.

3. start of roundup

The impatience coefficient of sheep, pdist is the distance function.

$$d = pdist(best, x)$$

$$z = e^{-\frac{d_{\min}}{Mc}}$$

Using Monte Carlo simulation dogs to find the optimal roundup solution.

Optimal displacement equation.

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

$$V_{i+1} = c_1 \cdot rand \cdot (bestx - X_i) + c_2 \cdot rand \cdot (bestc - X_i)$$

$$w = w_{\min} + \frac{(Y_i - Y_{\min})(w_{\max} - w_{\min})}{Y_{ave} - Y_{\min}}$$

$$X_{i+1} = X_i + w \cdot V_{i+1}$$

c_1 c_2 are the learning factor, $w_{\max}$ $w_{\min}$ are the inertia weight, $bestx$ is the current prey position, and is the prey escape position.

The results of the optimal function search are shown in the figure.

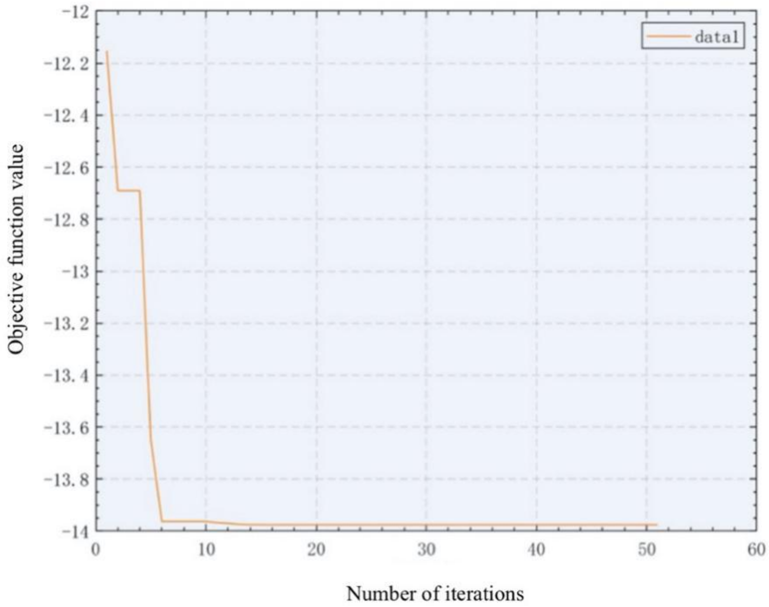


Figure 3 The objective function optimization diagram

The displacement formula of the wolf pack algorithm and the displacement formula of the particle swarm algorithm:

$$C_{ij} = rand$$

$$A_{ij} = 2 \cdot a \cdot rand - a$$

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c_j} X_i - A_{ij} \cdot C_{ij} (X_i - X_{ij})$$

$$V_{i+1} = c_1 \cdot rand \cdot (\overline{X} - X_i) + c_2 \cdot rand \cdot (bestc - X_i)$$

$$X_{i+1} = X_i + V_{i+1}$$

Combined with the bat algorithm displacement formula:

$$v = (bestc - X_i)$$

$$w = w_{\min} + (w_{\max} - w_{\min}) e^{-2(\frac{\det_j}{\det})^2}$$

$$X_{i+1} = X_i + w \cdot v \cdot rand$$

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

det is the maximum number of iterations, $\det_j$ is the current jth iteration.

Using Matlab software to perform iterative calculations, the specific results are shown in Figure 4:

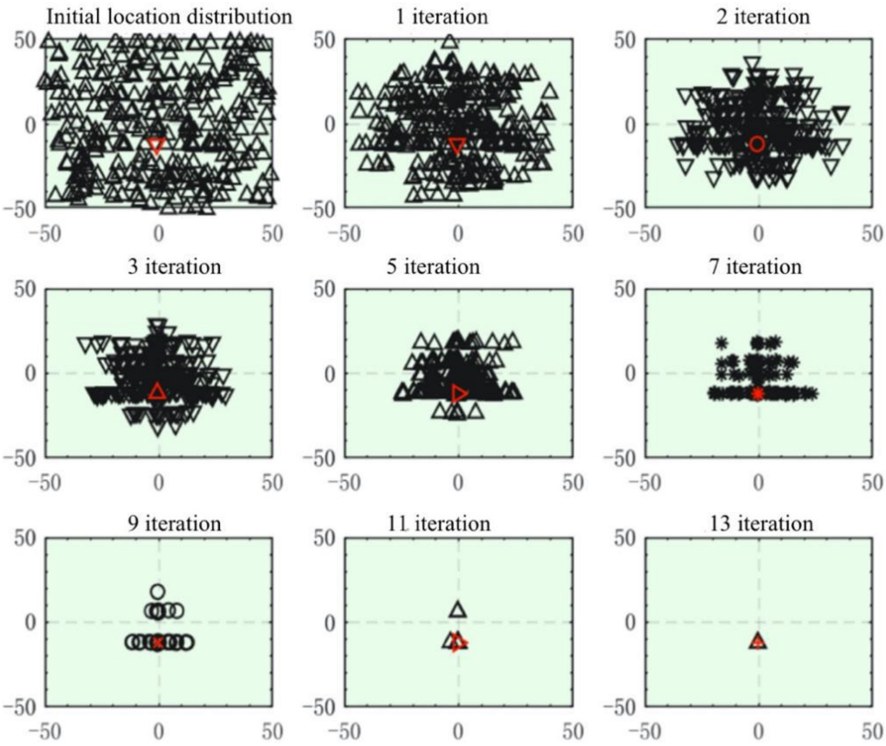


Figure 4 Graph of iteration results

4.1.2. Solving the model

Using Matlab software, this paper first randomly generated the position of the dog and the sheep, and then used the Monte Carlo model to find the best hunting strategy for the dog. Using Matlab, this paper was used to find the location of the sheep and its escape probability when the sheep was in the range of motion. The relationship diagram is shown in Figure 5:

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

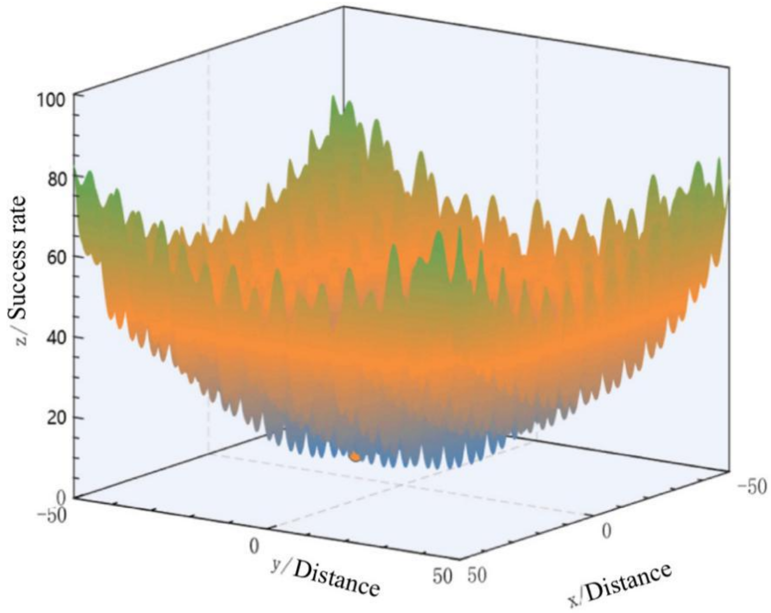


Figure 5 Sheep escape probability diagram

From the figure, it can be seen that when the initial position of the sheep is on the circumference of the circle, the sheep's escape probability is the highest, corresponding to the dog's round-up probability is the lowest. In the same way, this article can see that when the flock is located at the center of the circle, the dog's rounding-up probability is higher. When the sheep is on the border, the escape probability is generally higher than 80%, which means that when the sheep is on the border of the range of motion, the dog has almost no way to round up the sheep. Unless the dog is very close to the sheep, the dog has a certain probability to round up the sheep. sheep.

Through Matlab, the simulation operation is carried out to obtain the optimal containment strategy plan diagram of dog to sheep under the condition of unlimited conditions, as shown in Figure 6.

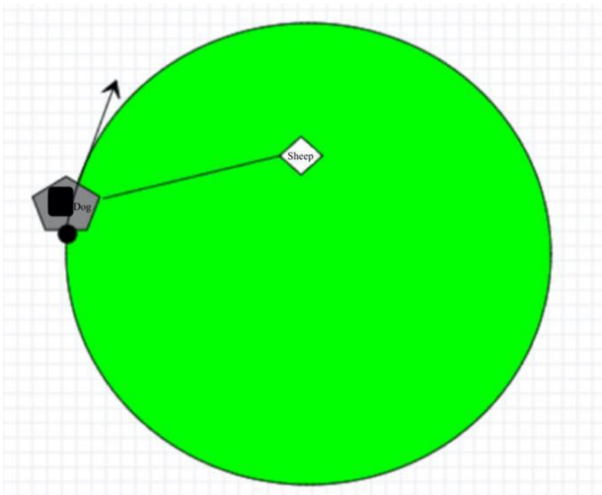


Figure 6 The optimal containment strategy for dogs

In the figure, the green area is assumed to be the set sports field, the white diamond represents the individual sheep, the gray pentagon represents the individual dog, and the arrow indicates the direction of the dog's rounding movement at the current moment.

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

Because the individual sheep has any direction in the field Movement ability, so the direction of the individual sheep is arbitrary. Therefore, the escape direction of the sheep cannot be one direction, but because of this article, the sheep does not have an optimal escape plan, and the sheep does not know their own abilities and cannot learn, so the sheep escapes at will. The hunting of dogs is indeed carried out in the optimal way, because the hunting direction of dogs is fixed, and they can only escape clockwise or counterclockwise along the circle, so the direction of the optimal hunting strategy for dogs is not fixed. It changes with the movement of the sheep. The optimal hunting direction of the dog is to use the dog and the sheep as the two mass points to connect the line to solve the tangent to the center of the circle where the dog is at the moment, and to compare the two tangents to obtain The direction with the smaller angle is the direction of the dog's optimal hunting strategy. And when the direction of movement of the sheep is inconsistent, the direction of movement of the sheep shall prevail.

4.2. Improved sheep-dog movement model

For problem two, on the basis of problem one, set the dog's round-up as the optimal round-up plan for problem one, use this optimal strategy to enclose and substitute it into the model, and make further improvements and updates to the model to obtain the corresponding The escape probability of the sheep under this model and the corresponding escape conditions can then be based on accurate modeling to solve the conditions under which the sheep can escape and win.

In this model, using randomly generated quantities as the initial states of dogs and sheep and performing iterative calculations within the same length of time, such a closed-loop system can effectively compensate for uncertainty and interference. In the iterative process, the system model changes with time, which can ensure a small deviation from the actual landing situation.

4.2.1. Model establishment

Aiming at the mathematical model proposed in the first question, the model predictive control law is used to produce piecewise smooth constant value control. Set the length of the estimation time of each step as N, then the prediction state at the next moment in each iteration is:

$$x(t \setminus i+1) = A_2 x(t \setminus i) + B_2 u(t \setminus i) + C_2,$$
$$i = 0, \dots, N-1$$

Among them: A_2 B_2, and C_2 are the values of A_2 (t), B_2 (t) and C_2 (t) at the current time in each iteration; $x(t \setminus i)$ and $u(t \setminus i)$ are the state vector and control vector of the i-th step in a prediction time domain respectively; $x(t \setminus i+1)$ are the i+1th The predicted state of the step.

In order to easily calculate the state sequence and control sequence in each prediction, the state variables and control variables of all steps in the time domain of each prediction can be integrated into new state variables X and control variables U:

$$X = [x_1, x_2, x_3, \cdots, x_N]^T_{6N\times1}$$
$$U = [u_0, u_1, u_2, \cdots, u_{N-1}]^T_{3N\times1}$$

Then the above formula can be rewritten as:

$$X = A_3 x_0 + B_3 U + C_3$$

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

Using the new state quantity and control quantity formula, the performance index formula can be rewritten as:

$$J = X^T Q' X + U^T R' U$$

Among them: Q' and R' are the weight values of the state quantity and the control quantity.

For the distance L_1 between the sheep-to-dog and the distance for sheep L_2 the boundary size of the movement area in the model, the distance relationship satisfies:

$$L_2 = L_1 \cdot e^{-l}$$

Where l is the speed of the dog's movement.

The dog's movement formula is circular, which makes it easier to traverse to find a better point. The principle of impatience coefficient is that the farther the prey is from the hunter, the smaller the coefficient and the smaller the range of prey movement. It is possible to allow hunters to traverse more; when the hunter approaches the prey, the movable range of the prey will also increase. If the best point is not retrieved, it can also jump out of the local optimum; therefore, in the large range and for the optimization problem, the result is very likely to have a local optimum. Therefore, the establishment of this step is also to reduce the probability of a local optimum:

$$d = pdist(best, x)$$

$$z = e^{-\frac{d_{\min}}{Mc}}$$

At the same time, an adaptive coefficient is added to the dog's movement. Generally speaking, if the individual moves too far, it is easy to ignore the best advantage, but the individual's moving speed is too slow, which will also affect the overall efficiency of the algorithm, especially It is the optimization of a large range. This paper finds that the larger the interval, the larger the population is needed to achieve better results, or when the population remains unchanged, reducing its moving speed and increasing the number of iterations, the effects are consistent with each other; And the greater the density, the easier it is to find the optimal position, and the speed of approaching the optimal can be increased. Comprehensive considerations can be used to consider the individual's position update formula. When the distance from the current optimal distance is farther, the individual distribution density is smaller. Speed limit, certain rewards will be given when the distance is the most superior. In the program, for this algorithm, the main control of the displacement speed is the maximum number of laps k. The following adjustment mechanism for k is given in the realization of the algorithm program in this paper:

$$k' = \begin{cases} k \cdot e^{-2} & d \geq 0.5 \\ k \cdot e^{-d/5} & d < 0.5 \end{cases}$$

The dog's position update formula:

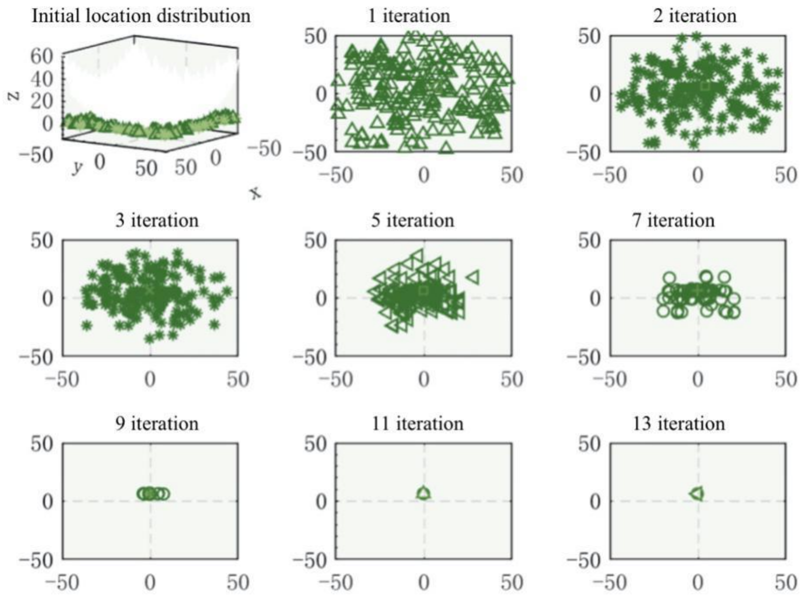
$$y = d \cdot e^{-k' \cdot rand}$$

$$\theta' = \theta_o + \Delta\theta = \theta_o + (k' \cdot ramd) \cdot 2\pi$$

$$x = [y \cdot \cos\theta', y \cdot \sin\theta']$$

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

Model iteration diagram:



The improved model optimization result graph:

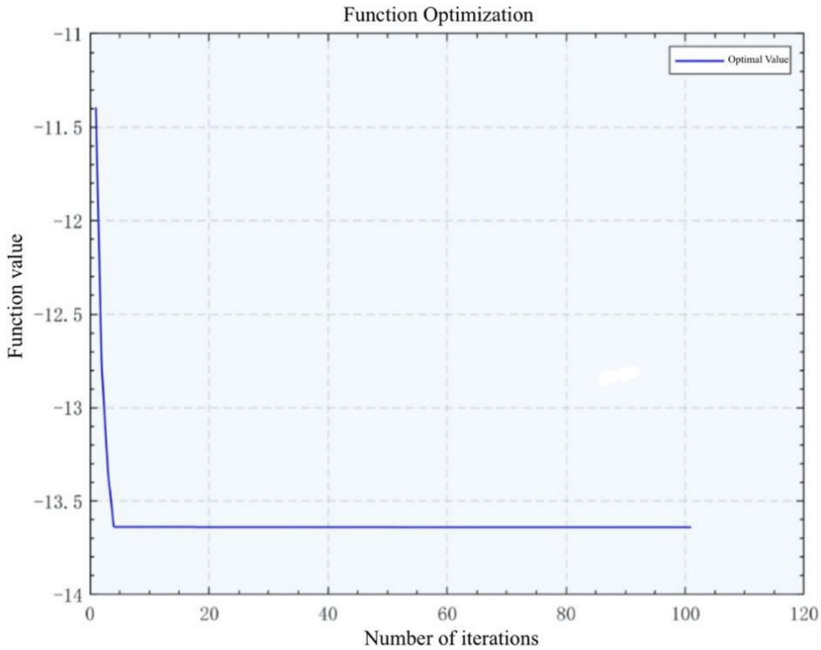


Figure 8 Function optimization result graph

4.2.2. Solving of the model

In order not to lose the generality of the model, and to verify the correctness and efficiency of the MPC method proposed in the article, the simulation of the exact mathematical model of the autonomous escape of sheep rounded up by the examination dog with the optimal strategy is given here. Among them, the MPC algorithm is used to solve the control volume by using the fmincon algorithm in MATLAB software, where the functions are set by Matlab default.

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

The distribution of sheep escape probability in the final improved model is plotted as:

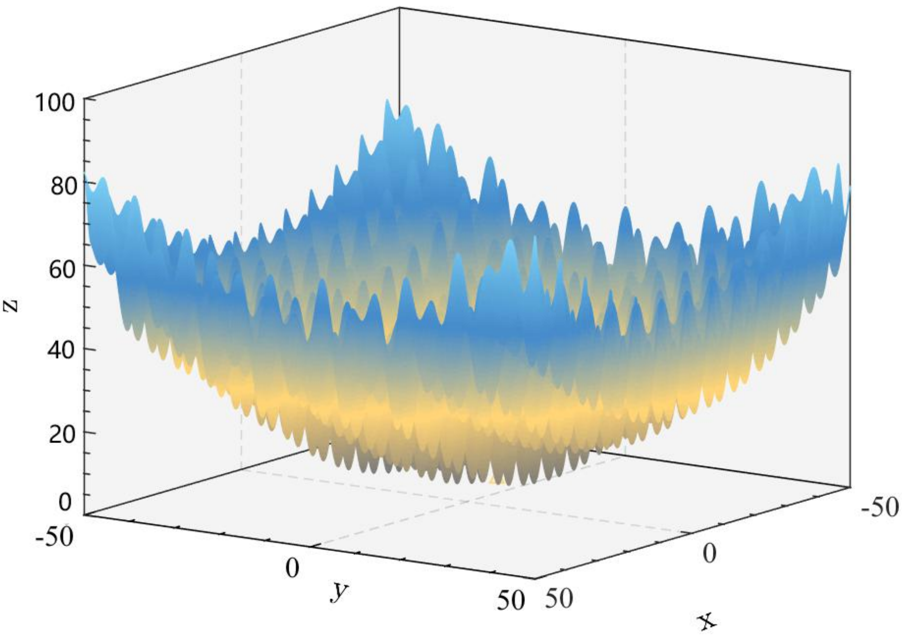


Figure. 9 Improved escape probability

Finally, under the assumption that the dog is enclosed with an optimal strategy and based on exact modeling to solve for the conditions under which the sheep can escape to win, the results are shown in Figure 10.

Figure. 10 Graph of escape tuning results

In other words, when the sheep is on the opposite side of the dog and close to the boundary with the circumference, the escape of the sheep can be completed 100%, the specific conditions are the initial position, when the distance between the dog and the wolf is more than $\sqrt{2}R$, and the escape direction of the sheep is close to the circumference and far from the dog, it can be guaranteed that the sheep can achieve successful escape without changing

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

the direction of movement; when the distance between the sheep and the dog is $\frac{3}{2}e^{-1}\sqrt{2R}$, when the angle relationship is $\theta' < 135^o$, the sheep can change the direction of escape in the process of escape. the direction of escape, she can also achieve a successful escape.

4.3. Adversarial neural network model

Since there is no machine learning data in this paper, the training difficulty is very high for general machine learning, and the training accuracy is also very low. However, in recent years, with the development of artificial neural networks and the in-depth study of game theory, people put the adversarial scenarios in game theory into neural networks, and then developed into adversarial neural networks, which do not need a large amount of data, and have an internal generator and an adversary, each of which learns and confronts each other to improve their accuracy, and thus improve the accuracy of self-generated conditional adversarial neural networks, and For this reason, this paper uses the self-generating conditional adversarial network to train and learn the sheep so that they can escape accurately.

4.3.1. Modeling

Generative adversarial networks have recently been introduced as an alternative framework for training gen-ad iterative models to evade the difficult computation of approximating many tricky probabilities. The advantages of adversarial networks are that Markov chains are never needed, only backpropagation is required for obtaining gradients, no inference is required during learning, and a wide variety of factors and interactions can be easily incorporated into the model. Moreover, it can produce up-to-date log-likelihood estimates and realistic samples. In an unconditional generative model, there is no control over the pattern of the generated data. However, data generation can be guided by conditioning the model based on additional information. Such a condition can be based on category labels, used to repair certain parts of the data, or even data from different modalities. In this work, this paper shows how to construct conditional adversarial networks and obtain empirical results, and two sets of experiments are demonstrated in this paper.

Conditional generative adversarial networks are developed based on GANs, and the unsupervised learning of GANs is turned into supervised learning by adding additional information y to the inputs of the generator G and the discriminator D. In the training phase, the random noise z and the conditional variable y are input to G at the same time to obtain the generated data G(z | y) that obeys the distribution P_{data} of real data as much as possible; then the real data x, the conditional variable y and the generated data G(z | y) are input to D at the same time, and finally a scalar is output to estimate the probability that the input data comes from the real data. Its objective function is:

$$\min_G \max_D V(D,G) = E_{x\sim Pdata(x)}[\log D(x \setminus y)] + E_{z\sim Pz(z)}[\log(1 - D(G(z \setminus y)))]$$

The objective of D is to achieve a binary discrimination of data sources, and the objective of G is to maximize the probability of D judging the input data incorrectly. Thus, these two processes are iteratively optimized against each other to improve the performance of both G and D. Finally when D is unable to correctly discriminate between true and false data, it is assumed that G has learned the true data distribution and the two have reached an equilibrium state.

Conditional batch normalization (CBN) processes the data not by using the entire batch statistics, but by normalizing within each type of data feature map, relying on the category labels to restore the original data of each type:

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

$$\mu_B = \frac{1}{m}\sum_{i-1}^{m} x_i$$

$$\delta_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$

$$x_i = \frac{x_i - \mu_B}{\sqrt{\delta_B^2 + \varepsilon}}$$

$$y_i = r(c)x_i + \beta(c)$$

$$\omega(i) = \frac{\omega_0 - \omega_n}{m}(m - t) + \omega_n + 0.2s$$

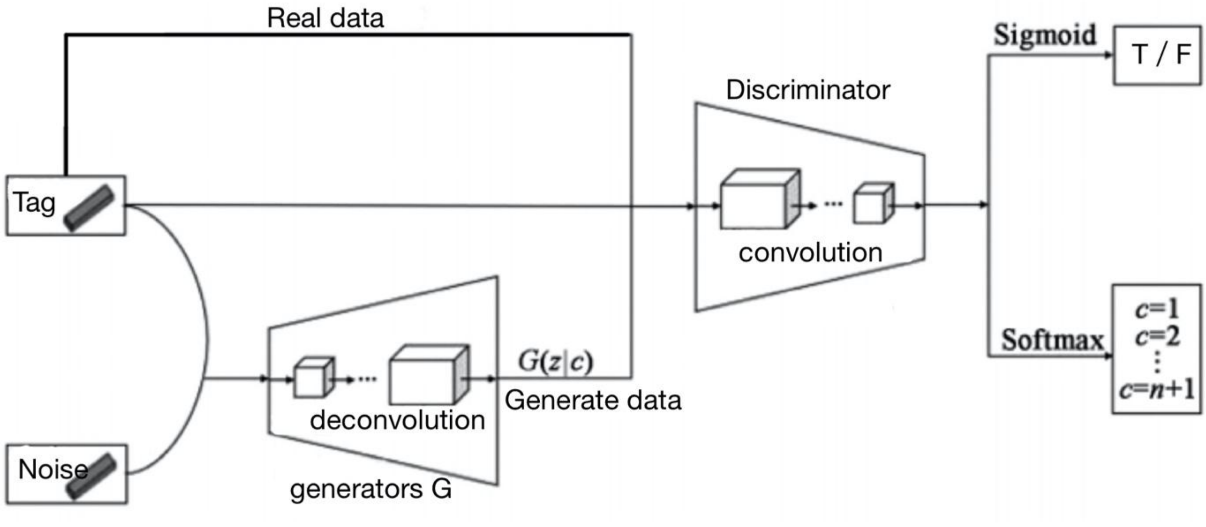


Figure 11 CBN-CGAN network structure diagram

The structural diagram of the neural network is shown in Figure:

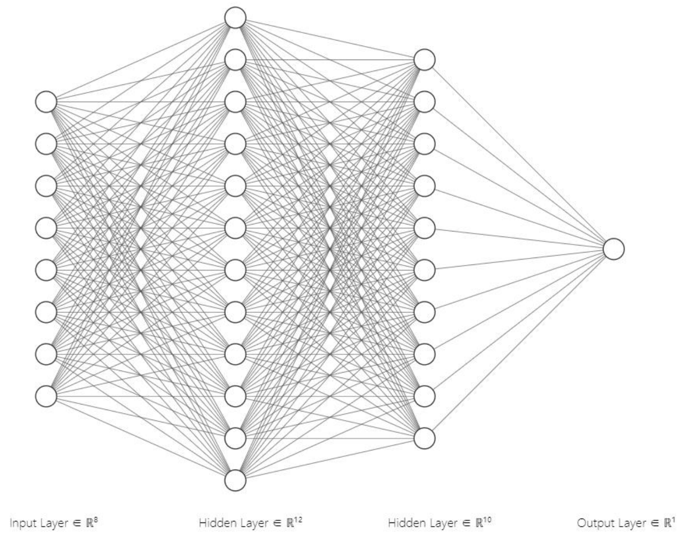


Figure. 12 Structure of neural network

The network structure of the generator and discriminator is changed to a fully convolutional network structure to learn the features of the image better with the powerful feature

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31
extraction ability of the convolutional neural network (CNN). A Softmax classifier is added at the end of the discriminator output to make the model suitable for multiple classification tasks.

The CBN is added to the network layer of the generator and discriminator to make full use of the category labels to batch each category of data, so that the network can fully learn the features of each category in the feature function and improve the quality and recognition accuracy of the generated data. At the same time, the stability of the model is improved, the convergence speed is accelerated, and the problem of gradient disappearance is alleviated.

4.3.2. Model solving

By combining the algorithm of the conditional adversarial neural network model established in this paper with repeated iterative calculations in the numerical simulation software using Matlab, it finally makes the sheep can accurately achieve the escape through learning, and the result graph of the network training is shown in Figure 13 below.

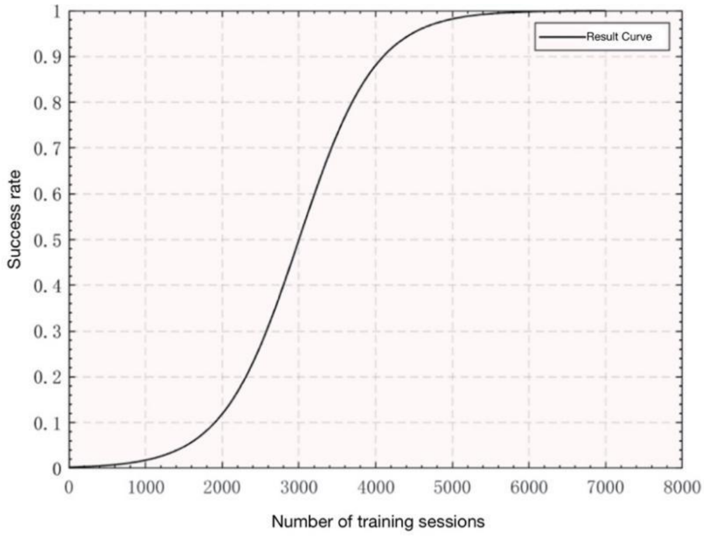


Figure 13 Correctness curve

The relationship between the training curve and the correct rate is obtained. From this paper, we can see that when the training confrontation result reaches about 2000 times, the correct rate increases sharply, and when the training reaches about 5000 times, the correct rate has reached about 98%, at this time, the training result is very complete, which indicates that the sheep can already have 98% probability of successfully escaping after training. When it reaches about 6000 times, the correct rate can almost reach 100%, which shows that the adversarial neural network model established in this question is reliable.

4.4. Evaluation of the adversarial neural network model

For problem four, this paper needs to design a set of evaluation system to quantitatively evaluate the learning ability of the adversarial neural network machine learning algorithm given in problem 3. For this, this paper firstly selects four common indexes for the evaluation of machine learning, such as accuracy, precision, error distribution rate, learning efficiency, and PR curve. Secondly, the decision problem is divided into three levels, O, C and P. By analyzing the above four metrics, multiple sets of data are approximated to obtain the positive and negative matrices, which are implemented using Matlab and passed the consistency test of the positive and negative matrices. Finally, this paper quantitatively evaluates the machine learning method given in the above question by plotting the PR curve.

First of all, it is necessary to quantify overfitting, which is the performance that is excellent on

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

the training set and drops sharply on the test set, most likely due to treating local features as general features. However, it is often difficult to identify exactly which local features have been learned, and in some special cases, it is possible that a model can learn parameters very close to the global optimal solution with relatively few training samples. Determining the appropriate number of samples is the more difficult problem. The effort to learn as comprehensive features as possible using as few samples as possible is the direction being worked on at the moment.

In this paper, four metrics are chosen for evaluation, including accuracy.

$$P = \frac{TP}{TP + FP}$$

Accuracy.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Error distribution rate.

$$1 - ACC = \frac{FP + FN}{TP + FP + TN + FN}$$

The full name of the ROC curve is "Receiver Operating Characteristic" curve. Its horizontal axis is the False Positive Rate (FPR) and its vertical axis is the True Rate (TPR), which are defined as:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{TN + FP}$$

AUC (Area Under Curve) is defined as the area under the ROC curve, which is obviously not greater than 1. A positive sample and a negative sample are randomly selected, and the probability that the classifier determines that the positive sample has a higher value than the negative sample is the AUC value.

The larger the AUC value, the higher the correctness of the classifier.

In addition, most algorithms have some parameters that need to be set, and the performance of the learned models often differs significantly with different parameter configurations. Therefore, in the model evaluation and selection, in addition to the selection of the applicable learning algorithm, the parameters of the algorithm also need to be set, which is commonly referred to as "tuning".

Based on the above indicators, this paper quantitatively analyzes the model in question 3 and obtains the ROC curve for the model as shown in the following figure:

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

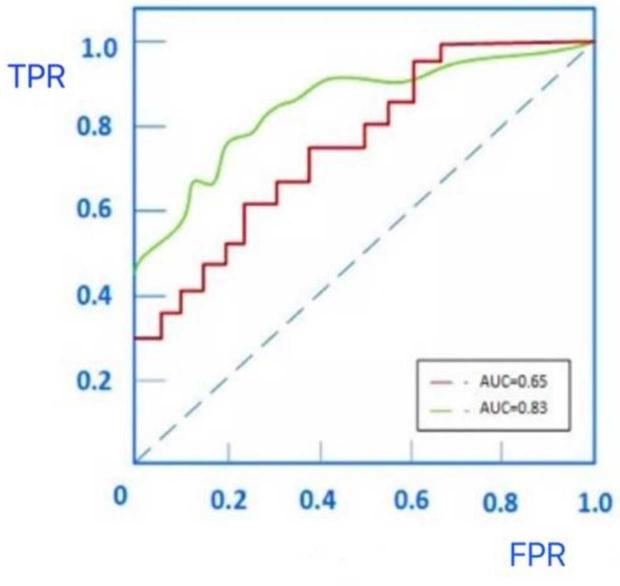


Figure 14 ROC graph

Finally, this paper concludes that the model has a high accuracy rate and runs at a high speed, and it can train the intrinsic connection between data models and find the interrelationship between data in a short time, and the learning efficiency and effect is its powerful, when the training reaches about 4000 times, the correct rate of the model can already reach about 98%, which basically meets most of the requirements of machine learning algorithms on the market at present, and when the training is continued and the training times reach 6000, the output correct rate of the model can be close to 100%, which shows that the machine learning model in this paper is very feasible and the model's excellence is excellent.

4.5. Evaluation of sheep escape machine learning method

For problem 5, firstly, this paper needs to give another sheep escape machine learning method, based on the analysis in problem 3, here this paper chooses genetic algorithm for learning, because genetic algorithm has similarity with neural network in some aspects, therefore, this paper chooses genetic algorithm as the machine learning algorithm model for this problem. Secondly, this paper needs to point out the quantitative evaluation of the proposed algorithm, and here this paper takes the quantitative analysis in Problem 4 as the benchmark to analyze the algorithm quantitatively and get the corresponding results. Finally, this paper quantitatively evaluates the strengths and weaknesses of the algorithm and improves the weaknesses, so that the whole machine learning model is optimized and supplemented, and can be reasonably promoted and used.

4.5.1. Algorithm proposal

Again this question, this paper uses genetic algorithm for learning, and its specific process is shown in Figure 15 as follows:

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31

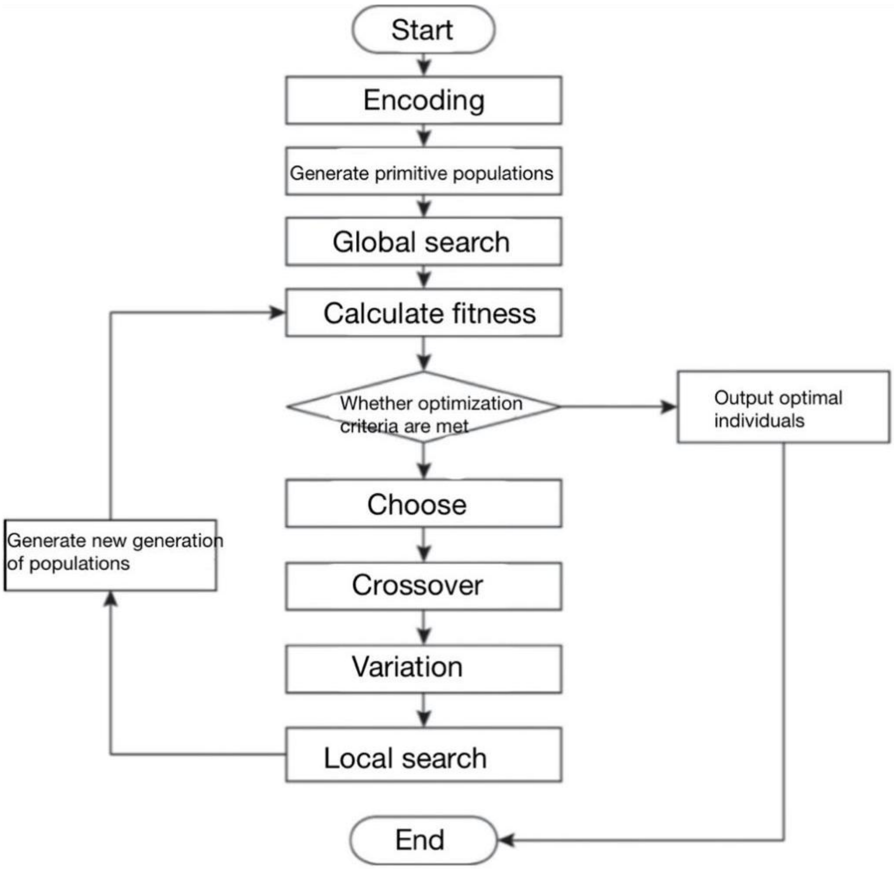


Figure 15 Flow chart of the genetic algorithm

The genes of each individual of the initial population can be generated with uniformly distributed random numbers, for example, to find the optimal solution in the interval $x \in [a, b]$, then the length of the chromosome encoding and the accuracy requirement of x are crucial.

$$2^{n-1} \leq (b-a)10^m \leq 2^n - 1$$

When solving for the individual adaptive function value, it is necessary to decode the chromosome encoding (decode), for a binary string of length n, the decoded x value is

$$x = a + \frac{b-a}{2^n - 1} \sum_{k=1}^{n} b_k 2^{k-1}$$

where $b_1, b_2, b_3, \cdots, b_n$ is the number (0 or 1) encoded in the kth bit (from right to left) of the individual's binary code, and the above definition of encoding and decoding enables the transformation between the decoding and gene-space elements.

4.5.2. Evaluation of the algorithm

The model in Problem 3 is replaced by a genetic algorithm. In this paper, we evaluate the genetic algorithm using the evaluation metrics in Problem 4 and find that the genetic algorithm can directly use the objective function value as search information. It only uses the fitness function value to measure the goodness of an individual, and does not involve the process of differentiating the objective function value. Since in reality many objective functions are difficult to derive or even do not have derivatives, this also makes the genetic algorithm show a high degree of superiority. However, since a single genetic algorithm code cannot fully represent the constraints of the optimization problem, it is necessary to consider the use of thresholds for infeasible solutions, which in turn increases the workload and solution time. In addition, it is found that the genetic algorithm is also less efficient compared

ISSN: 2790-170X DOI: 10.56028/ijcit.1.1.31
to other transmission machine learning algorithms by calculating this paper.

Reference

[1] Zhao, Y. M., Gu, S. K. An adversarial attack defense model incorporating residual dense self-attentive mechanism and generative adversarial networks [J/OL]. Computer Applications:1-12 [2021-08-12].

[2] Si X.F., Zhang, Q.G. Weakly supervised adversarial data augmentation for fine-grained visual classification algorithms[J]. Electronic Design Engineering,2021,29(11):160-165.

[3] Ling, W.T., Ni, J.J., Chen Yan, Tang, G.Y. Multi-UAV seizure based on improved whale optimization algorithm[J]. Computers and Modernization,2021(06):1-5+11.

[4] You, H.H., Yu, M.J., Lv Yan, Yang, H.Y., Han, Q.S. Application of UKF in air combat trajectory prediction based on improved gray wolf algorithm optimization[J]. Tactical Missile Technology. 2020(01)

[5] Zhuo, J.W., Application of MATLAB in mathematical modeling. 2nd edition [M]. Beijing University of Aeronautics and Astronautics Press, 2014.