Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 12, No. 3, 2025

Research on Machine Learning Optimization Algorithm Based on QUBO Model

Minyi Ye *, Lanqing Wu, Guanquan Zhu, Jiaguo Jiang

School of Mathematics and Information Science, Guangzhou University, Guangzhou Guangdong, 510006, China

*** Corresponding author:** Minyi Ye (Email: 32215150090@e.gzhu.edu.cn)

Abstract: This paper deals with the research of QUBO model optimization machine learning algorithm, including the following: first, discretize the continuous parameters of the machine learning model into binary variables, and construct the QUBO objective function; then optimize the QUBO model by using the quantum annealing algorithm; and finally solve the globally optimal solution by adjusting the annealing parameters, which include the initial temperature, the coefficient of temperature reduction, and the number of iterations. QUBO model can discretize continuous variables (including parameters such as weights and bias) in AR, SVM, CNN and other models in machine learning into binary variables, so as to better deal with nonlinear relationships and reduce the computational complexity in the training process. With the QUBO model, the regularization term can be better controlled to avoid the overfitting problem. Meanwhile, the QUBO model can be solved in parallel by quantum computing or simulated annealing algorithm, which significantly improves the computational efficiency under large-scale data.

Keywords: QUBO; AR; SVM; CNN; Quantum Computing; Simulated Annealing Algorithm.

1. Introduction

The development of quantum technology is of great scientific significance and social value, which is expected to have a major impact on traditional technology and trigger technological revolution and industrial transformation [1].

Among them, quantum computing stands out in decision problems, classification problems, as well as prediction problems. In recent years, people have discovered that a mathematical formulation known as QUBO, an acronym for a Quadratic Unconstrained Binary Optimization problem. Through special reformulation techniques that are easy to apply, the power of QUBO solvers can be used to efficiently solve many important problems once they are put into the QUBO framework [2].

According to the above description and relevant data, this paper built mathematical models to solve the following problems.

(1) Transform the Autoregressive (AR) model used in the time series forecasting problem whose primitive data ranging from January to September into a QUBO model and define the objective function and decision variables clearly. Based on this, use Kaiwu SDK's simulated annealing algorithm to solve the model and predict the demand for October.

(2) Classify the provided dataset using an SVM model, among which you should transform the optimization problem of training an SVM-based classification model into a QUBO model and define the objective function and decision variables. Besides, solve the QUBO problem using the simulated annealing algorithm in the Kaiwu SDK.

(3) Explore the integration of quantum computing and deep learning by selecting a specific application scenario like image classification or recommendation systems and design a deep learning model such as Convolutional Neural Net-works (CNNs), then convert the model's training optimization into a QUBO form, Finally use the simulated annealing algorithm to solve this optimization problem.

2. Model 1: The QUBO Model Based on Time Series Prediction

(1) Establishment of QUBO Resource Demand Prediction Model Based on AR

Firstly, based on the given problem settings, we conduct predictions by employing the autoregressive model. Autoregressive (AR) model is a model for studying time series, with wide applicability and high accuracy.

AR model is mainly based on the linear combination of past observations and present disturbance values for prediction. The general P-order autoregressive process AR (p) is[3]

$$\widehat{Y_t} = c + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \varepsilon_t \quad (1)$$

Where, $\widehat{Y_t}$ indicates the predicted value of resource demand in month t, and Y_{t-i} indicates the real value of resource demand in month (t-1); c is a constant, representing the average level of the time series; φ_i refers to the autoregressive model parameter, representing the effect of the i-th lag term on the current value; ε_t refers to random noise with a mean of zero; p is the order of the AR model, indicating how many lag terms are used to predict the current value, as determined by the Akaike information criterion (AIC) and p=3.

On this basis, We transform the above time series prediction problem into Quadratic Unconstrained Binary Optimization (QUBO) model. QUBO, whose minimum energy state is called the ground state. The QUBO model consists of multiple 0-1 binary variables, and a combination of which gives one solution. In particular, the combination of binary variables giving the minimum energy state of the QUBO model is called the ground-state solution.

A QUBO model is represented by the formula below using a binary variable x_i that takes 0 or 1($x_i = 0$ *or* $x_i = 1$).

$$f(X) = \sum_i a_i x_i + \sum_{i \neq j} b_{ij} x_i x_j \quad (2)$$

Where $X = \{x_1, x_2 \cdots\}$ is a set of binary variables; a_i is the external magnetic field coefficient representing the force on the binary variable x_i; and b_{ij} is the interaction

coefficient representing the weight of the connection between the binary variables x_i and x_j. A solution of a QUBO model is one of the combinations of binary variables. The ground-state solution of the QUBO model is the combination of binary variables that minimizes $f(X)$. The value of $f(X)$ for a given combination of binary variables is called the objective value.

Remember that Q is composed of a matrix b_{ij}, according to the QUBO Hamiltonian produced by it, the objective function is transformed into a quadratic form, then formula (2) can be simplified as:

$$Q(x) = x^T Q x + a^T x \tag{3}$$

Where x is the binary decision variable, x^T represents the transpose of x, a is the linear term coefficient vector, Q is the quadratic term coefficient matrix of the quadratic unconstrained binary optimization problem, and it is a symmetric matrix, otherwise the form of the coefficient matrix Q can be changed by the following formula:

$$\begin{cases} \text{Symmetrical Form} & q'_{ij} = \frac{q_{ij}+q_{ji}}{2} \\ \text{Upper Triangular Form} & \begin{cases} q'_{ij} = (q_{ij} + q_{ji}), & i < j \\ q'_{ij} = 0, & i > j \end{cases} \end{cases} \tag{4}$$

We take 70% of the data as the training set, 30% of the data as the test set, by determining the Sum of Squared Errors (SSE) of the predicted value of the minimum autoregressive function, predict the Computing resource demand in October. In this problem, we aim to minimize the predicted value error while satisfying the constraint that demand varies within a reasonable range. Therefore, based on the above Analysis, an objective function D with minimized predicted value is constructed:

$$min\ D = \sum_{t=p+1}^{9} (Y_t - \widehat{Y_t})^2 \tag{5}$$

Combined with formula (3), we can see that the objective function can be expressed as follows:

$$\begin{aligned} min\ D &= \sum_{t=p+1}^{9} \left[Y_t - \left(c + \sum_{i=1}^{p} \varphi_i Y_{t-i} \right) \right]^2 \\ &= \sum_{t=p+1}^{9} \left[{Y_t}^2 - 2Y_t \left(c + \sum_{i=1}^{p} \varphi_i Y_{t-i} \right) + \left(c + \sum_{i=1}^{p} \varphi_i Y_{t-i} \right)^2 \right] \end{aligned} \tag{6}$$

Based on this, we define a $(1+p) \cdot n$-dimensional vector x to discretize the continuous values c and φ_i to binary decision variables, where $x_{1:n}$ is the binary representation of c and the binary representation of φ_i is $x_{n+1:(p+1)n}$, arranged in the order of i:

$$x = [c_1, c_2, \cdots c_n, \varphi_{11}, \varphi_{12}, \cdots \varphi_{1n}, \varphi_{21}, \cdots \varphi_{pn}] \tag{7}$$

The mathematical formula for the specific binarization is as follows:

$$\begin{cases} c = \sum_{j=1}^{n} 2^{j-1} x_j \\ \varphi_i = \sum_{j=1}^{n} 2^{1-j} x_{n+(i-1)n+j} \end{cases} \tag{8}$$

Substituting formula (8) into formula (6) can simplify and expand the objective function:

$$\begin{aligned} min\ D = \sum_{t=p+1}^{9} \{ & {Y_t}^2 - 2Y_t \left(\sum_{j=1}^{n} 2^{j-1} x_j \right) - 2Y_t \left(\sum_{i=1}^{p} \sum_{j=1}^{n} 2^{1-j} x_{n+(i-1)n+j} Y_{t-i} \right) \\ & + \left(\sum_{j=1}^{n} 2^{j-1} x_j \right)^2 + 2 \left(\sum_{j=1}^{n} 2^{j-1} x_j \right) \left(\sum_{i=1}^{p} \sum_{j=1}^{n} 2^{1-j} x_{n+(i-1)n+j} Y_{t-i} \right) \\ & + \left(\sum_{i=1}^{p} \sum_{j=1}^{n} 2^{1-j} x_{n+(i-1)n+j} Y_{t-i} \right)^2 \} \end{aligned} \tag{9}$$

And since the constant term has nothing to do with the value of the decision variable when the function D gets the minimum value, for simple operation, we directly ignore the constant term y_t^2, formula (9) can be simplified to the following mathematical expression:

$$\begin{aligned} min\ D' = \sum_{t=p+1}^{9} \{ & -2Y_t \left(\sum_{j=1}^{n} 2^{j-1} x_j \right) - 2Y_t \left(\sum_{i=1}^{p} \sum_{j=1}^{n} 2^{1-j} x_{n+(i-1)n+j} Y_{t-i} \right) \\ & + \left(\sum_{j=1}^{n} 2^{j-1} x_j \right)^2 + 2 \left(\sum_{j=1}^{n} 2^{j-1} x_j \right) \left(\sum_{i=1}^{p} \sum_{j=1}^{n} 2^{1-j} x_{n+(i-1)n+j} Y_{t-i} \right) \\ & + \left(\sum_{i=1}^{p} \sum_{j=1}^{n} 2^{1-j} x_{n+(i-1)n+j} Y_{t-i} \right)^2 \} \end{aligned} \tag{10}$$

In addition, we constrain the decision variable by the following formula:

$$\begin{cases} 0 \le c \le 2^{13} - 1 \\ 2^{-12} \le \varphi_i < 1 \end{cases} \tag{11}$$

Among them, by observing the demand data from January to September, we calculate and find that the data 9000 in January is close to $2^{13} - 1$, so the value range of c is set in the interval $[0, 2^{13} - 1]$; Similarly, we find that the difference of data multiples is not more than two times, so the value range of φ_i is set in the interval $[2^{-12}, 1]$.

Among them, the QUBO matrix needs to be converted to Ising variable before using Kaiwu SDK. The Ising Model is a random process model that describes the phase transition of matter, which is abstracted into the following mathematical form:

$$H(\sigma) = -\sum_{i,j} J_{ij} \sigma_i \sigma_j - \mu \sum_i h_i \sigma_i \tag{12}$$

Where, σ is the spin variable to be found, the value is {-1,1}, H is the Hamiltonian, J is the quadratic term coefficient,

μ and h is the linear term coefficient, are known quantities[4]

We adopt the following formula as the conversion formula between the binary variable and the Ising variable:

$$x_i = \frac{1+\sigma_i}{2} \tag{13}$$

Where, $\sigma_i = -1$ correspond $x_i = 0$, $\sigma_i = +1$ correspond $x_i = 1$.

Here, we summarize the solution of this model:

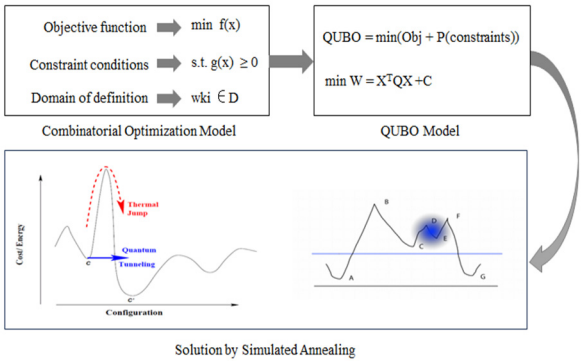

Figure 1. The solution of model 1

(2) Solution of Resource Demand Prediction Model Based on SA

In order to solve the resource demand corresponding to October when the error of the predicted value is minimized, we adopted the simulated annealing solver built in Kaiwu SDK to solve it. The following is the flow chart of the simulated annealing algorithm solver.

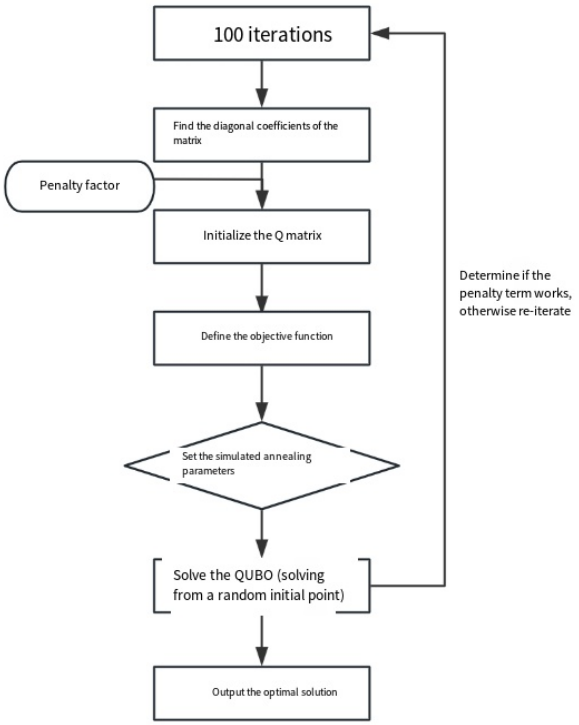

Figure 2. Flow chart of QUBO model solved by simulated annealing

For the parameters in the simulated annealing solver in the Kaiwu SDK, we set them as follows:

Select the initial temperature T_0 and the end temperature T_{end}, and select the appropriate cooling coefficient α during the annealing process, so that the iterative depth of each temperature in the annealing process does not exceed 100.

Table 1. Simulated annealing parameter table

Parameters	Select parameter values
Initial temperature T_0	3000
Cutoff temperature T_{end}	10^{-3}
Cooling factorα	0.90
Iterations per temperature	100
Size limit	100

By solving the simulated annealing solver, we get the following results:

Table 2. Optimal prediction parameter values table

Optimal prediction parameters	Value
c	1393
φ_1	0.6484
φ_2	0.1016
φ_3	0.1351

At this point, the mean square error D' has reached its minimum value of 30,723.78. Among them, φ_i can be regarded as the weight of the real value i before the predicted value to the predicted value, then the phenomenon of φ_1 great influence on the predicted value is in line with reality, which is reasonable.

On this basis, we substitute the formula (1) to solve the predicted value of each month as follows:

Table 3. Statistics of predicted values for each month

Month	Predicted Value		Month	Predicted Value
April	9784.98		August	10443.38
May	10030.58		September	10613.51
June	10147.90		October	10719.36
July	10248.88			

In addition, we draw the fitting image as follows:

It can be seen from the above figure that the predicted results have a high goodness of fit. Based on this, we calculated a variety of prediction evaluation indicators to help explain the excellent performance of the results.

In addition to this, we also visualized the process of Hamiltonian change over time in the simulated annealing algorithm to reflect the energy state of the system.

As can be seen from the figure above, when the system is in the initial state, the energy is higher; Then, the Hamiltonian decreases rapidly, and the system quickly finds a lower energy state in the optimization process; When the Hamiltonian approaches 0.03 seconds, there is a small increase, which is because the algorithm is trying to jump out of the local optimal solution. After the decrease, the Hamiltonian tends to be stable, indicating that the algorithm has converged and the system has reached the global optimal state. Solving time of the QUBO model is approximately 0.07s.

3. Model 2: The QUBO Model Based on SVM Classification

(1) Establishment of QUBO-Classification Model Based on SVM

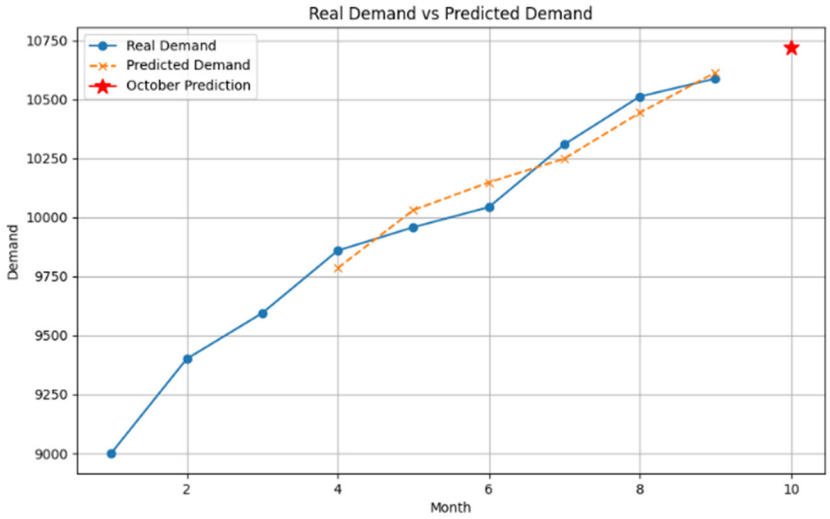

Figure 3. The predicted value compared with the true value

Table 4. Results evaluation index value statistics

Prediction Evaluation Indicators	Value
MAE	67.624
MAPE	0.667
R^2	0.933

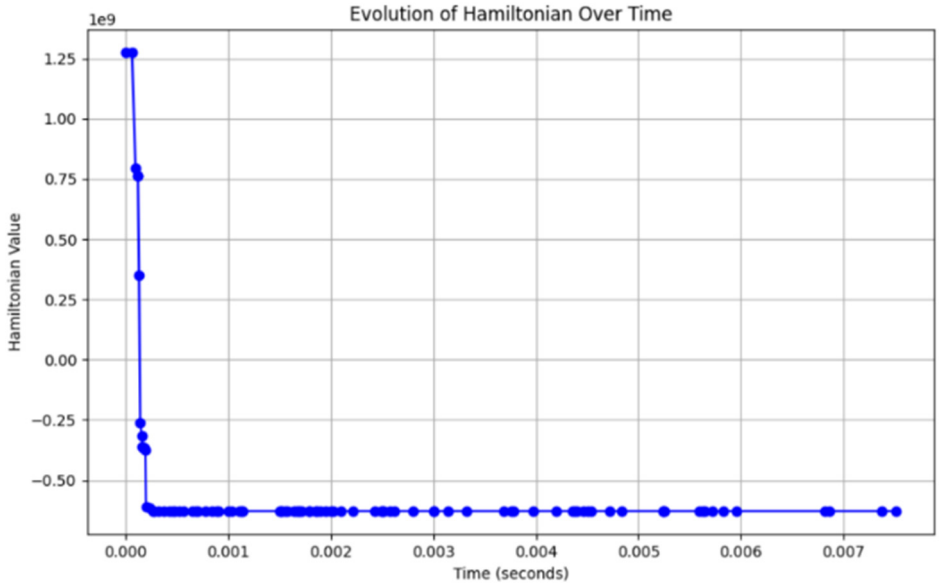

Figure 4. Visualization of Hamiltonian value over time

Support Vector Machine (SVM) is developed from the optimal classification surface in the case of linear separability. The basic idea can be illustrated by the two-dimensional situation in Figure 4. In the figure below, the solid point and the hollow point represent two types of samples respectively. If they are linearly separable, the result of machine learning is a hyperplane (also known as the discriminant function), which divides the training sample into positive and negative classes.

Obviously, there are infinite number of such hyperplanes. According to the requirement of the Structure Risk Minimization principle (SRM), the result of machine learning should be the optimal hyperplane, which can not only correctly separate the two types of training samples. but also maximizes the classification interval. Since SVM is a binary classification model, while this is a three-classification problem, we first split the three-classification problem into three different binary classification problems. including distinguishing between Setosa and Versicolor, distinguishing between Setosa and Virginica, including distinguishing between Setosa and versicolor, distinguishing between Setosa and virginica, distinguishing between Virginica and Versicolor. Distinguishing between Virginica and Versicolor. By doing this, a tripartite classification problem is transformed into three independent binary classification problems.

Suppose an *M-dimensional* hyperplane is described by the following equation [5]:

$$w \cdot x + b = 0 \quad w \in R^m, b \in R \tag{14}$$

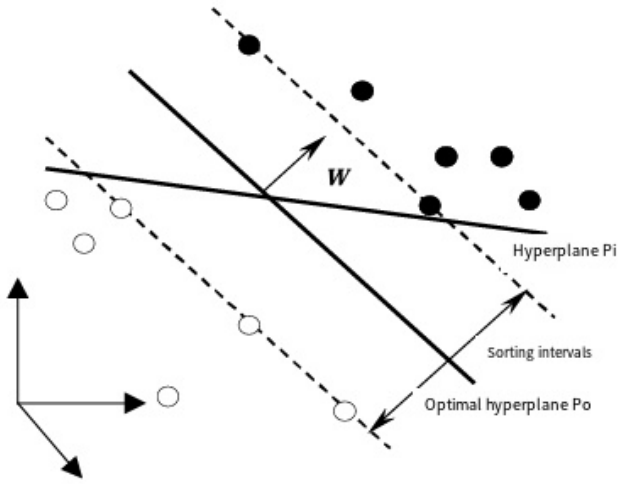

Figure 5. Classification hyperplane in the linearly separable case

Then the optimal hyperplane with the largest classification interval can be obtained by finding the minimum value of $\frac{\|w\|^2}{2}$, where the constraints are:

$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad i = 1, \cdots, n \tag{15}$$

Where, $y_i \in \{-1, +1\}$, x_i is the sample eigenvector. The constraint conditions ensure that the distance from the data point to the hyperplane is at least 1. When the above formula obtains an equal sign, it means that the data point is exactly on the interval boundary, and these data points are called support vectors.

In the case of linear indivisibility, such as the presence of noisy data, SVM introduces a relaxation term $\xi_i \geq 0$ in the above equation to achieve soft spacing, i.e.:

$$y_i(w^T \cdot x_i + b) \geq 1 - \xi_i \quad i = 1, \cdots, n \tag{16}$$

Based on this, the objective function is changed into the following formula:

$$Q = \min y(w, \xi_i) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{17}$$

Where w is the normal vector of the hyperplane, b is the bias of the hyperplane and ξ_i is the relaxation variable, C is the regularization parameter, which is used to control the model's tradeoff between classification interval and misclassification, and determines the relative importance of the two, while its value is selected by cross-validation, that is, adjusting C according to the performance on the verification set.

Based on the above analysis, we transform SVM into QUBO model, which means that formula (17) is transformed into formula (3).

First, we discretized w, b, and ξ_i in SVM by converting to binary. Suppose $w_j \in [w_{min}, w_{max}]$, w_j represented by m binary variables $z_{j,k} \in \{0,1\}$:

$$w_j = w_{min} + \Delta w \sum_{k=0}^{m-1} z_{j,k} \cdot 2^k \tag{18}$$

Of which,

$$\Delta w = \frac{w_{max} - w_{min}}{2^m - 1} \tag{19}$$

Then the mathematical expression of binary conversion of w binormal is as follows:

$$\|w\|^2 = \sum_{j=1}^{d} {w_j}^2 = \sum_{j=1}^{d} \left(w_{min} + \Delta w \sum_{k=0}^{m-1} z_{j,k} \cdot 2^k\right)^2 \tag{20}$$

In the same way, we assume that $b \in [b_{min}, b_{max}]$, represented by m_b binary variable $z_{b,k} \in \{0,1\}$:

$$b = b_{min} + \Delta b \sum_{k=0}^{m_b-1} z_{b,k} \cdot 2^k \tag{21}$$

Among them,

$$\Delta b = \frac{b_{max} - b_{min}}{2^{m_b} - 1} \tag{22}$$

And b_{min}, b_{max} respectively represent the minimum and maximum of b.

To convert formula (15) into QUBO penalty term, we have:

$$P_{constraint} = \lambda \sum_{i=1}^{n} \max\left(0, 1 - y_i(w^T x_i + b)\right)^2 \tag{23}$$

Where, $w^T x_i = \sum_{j=1}^{d} w_j x_{i,j}$, b is the bias and λ is the penalty term weight.

Secondly, we substitute the binary expressions of w and b into the above formula respectively, then we have:

$$\begin{aligned} w^T x_i + b &= \sum_{j=1}^{d} w_{\min} x_{i,j} + \Delta_w \sum_{j=1}^{d} \sum_{k=0}^{m_w-1} z_{j,k} \cdot x_{i,j} \cdot 2^k \\ &= \sum_{j=1}^{d} w_{min} x_{i,j} + b_{min} + \Delta_w \sum_{j=1}^{d} \sum_{k=0}^{m_w-1} z_{j,k} \cdot x_{i,j} \cdot 2^k + \Delta_b \sum_{k=0}^{m_b-1} z_{b,k} \cdot 2^k \end{aligned} \tag{24}$$

Where, $\sum_{j=1}^{d} w_{min} x_{i,j} + b_{min}$ is the constant term, $\Delta_w \sum_{j=1}^{d} \sum_{k=0}^{m_w-1} z_{j,k} \cdot x_{i,j} \cdot 2^k + \Delta_b \sum_{k=0}^{m_b-1} z_{b,k} \cdot 2^k$ is the part about the binary variable. Finally, substitute the above into the formula (23):

$$\begin{aligned} & 1 - y_i(w^T x_i + b) \\ &= 1 - y_i \left[\sum_{j=1}^{d} w_{min} x_{i,j} + b_{min} + \Delta_w \sum_{j=1}^{d} \sum_{k=0}^{m_w-1} z_{j,k} x_{i,j} 2^k + \Delta_b \sum_{k=0}^{m_b-1} z_{b,k} 2^k \right] \\ &= 1 - y_i \left[\sum_{j=1}^{d} w_{min} x_{i,j} + b_{min} \right] - y_i \Delta_w \sum_{j=1}^{d} \sum_{k=0}^{m_w-1} z_{j,k} x_{i,j} 2^k \\ & \quad - y_i \Delta_b \sum_{k=0}^{m_b-1} z_{b,k} 2^k \end{aligned} \tag{25}$$

Where $1 - y_i\left[\sum_{j=1}^{d} w_{min} x_{i,j} + b_{min}\right]$ is the constant term, $-y_i \Delta_w \sum_{j=1}^{d} \sum_{k=0}^{m_w-1} z_{j,k} x_{i,j} 2^k - y_i \Delta_b \sum_{k=0}^{m_b-1} z_{b,k} 2^k$ is the linear term about the binary variable, respectively remember the constant term is C_i, L_i, then the above formula can be abbreviated as follows:

$$1 - y_i(w^T x_i + b) = C_i + L_i \tag{26}$$

Then formula (23) can be written as:

$$P_{constraint} = \lambda \sum_{i=1}^{n} \max(0, C_i + L_i)^2 \tag{27}$$

Consequently, the objective function can be integrated as follows:

$$Q = \frac{1}{2}\sum_{j=1}^{d}\left[w_{\min}^2 + 2w_{\min}\Delta_w \sum_{k=0}^{m_w-1} z_{j,k}\cdot 2^k + \Delta_w^2\left(\sum_{k=0}^{m_w-1} z_{j,k}\cdot 2^k\right)^2\right] + C\sum_{i=1}^{n} max\left[0,\left(1 - y_i\sum_{j=1}^{d} w_{\min}x_{i,j} - y_i b_{\min}\right)\right. \left. - y_i\Delta_w\sum_{j=1}^{d}\sum_{k=0}^{m_w-1} z_{j,k}x_{i,j}\cdot 2^k - y_i\Delta_b\sum_{k=0}^{m_b-1} z_{b,k}\cdot 2^k\right] \tag{28}$$

(2) Solving QUBO-Classification Model Based on SA

Firstly, we do data preprocessing, which scramps the original ordered data, and uses 70% of the data for data training, and the rest of which is used for data prediction.

We still use the simulated annealing solver built in Kaiwu SDK for solving, in which the parameters in the solver are set as follows:

Table 5. Simulated annealing parameter table

Parameters	Select parameter values
Initial temperature T_0	100
Cutoff temperature T_{end}	10^{-3}
Cooling factorα	0.99
Iterations per temperature	10
Size limit	10

Select the initial temperature T_0 and the end temperature T_{end}, and select the appropriate cooling coefficient α during the annealing process, so that the iterative depth of each temperature in the annealing process does not exceed 10.

By solving the simulated annealing solver, we get 3 independent binary classification problem parameter results:

Table 6. Optimal prediction parameter values (Iris-setosa and Iris-versicolor)

Optimal prediction parameters	Value
w_1	-0.269
w_2	0.338
w_3	-0.701
w_4	-0.749
b	-0.247
λ	50

At this point, the objective function Q reaches a minimum, and w is the decision hyperplane normal vector, and b is the bias and λ is the penalty term weight.

On this basis, we draw the classification result image as follows:

It can be seen from the above figure that QUBO-SVM binary classification results are more accurate. Based on this, we calculated a variety of prediction evaluation indicators to help explain the excellent results.

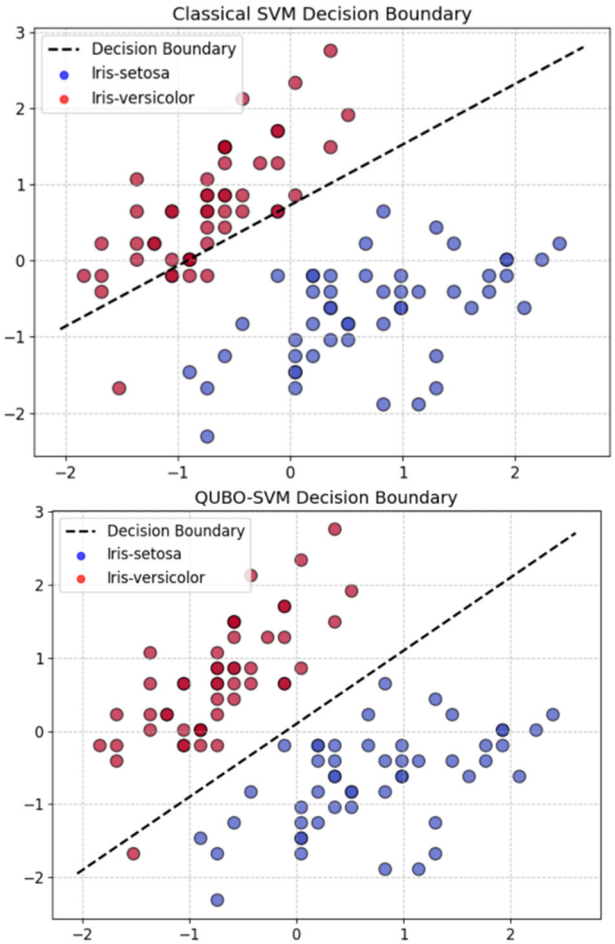

Figure 6. Comparison of classical SVM and QUBO-SVM classification (Iris-setosa and Iris-versicolor)

Table 7. Statistics of outcome evaluation indicators (Iris-setosa and Iris-virginica)

Data set	Recall rate	F1	Accuracy Rate	Precision rate
Training Set	1	1	1	1
Test Set	1	1	1	1

Based on this, we solve the hyperplane parameters of Iris-setosa and Iris-virginica:

Table 8. Optimal prediction parameter values (Iris-setosa and Iris-virginica)

Optimal prediction parameters	Value
w_1	-0.098
w_2	0.138
w_3	-0.663
w_4	-0.657
b	-0.139
λ	24

At this point, the objective function Q reaches its minimum value. Based on this, we draw the classification result image as follows:

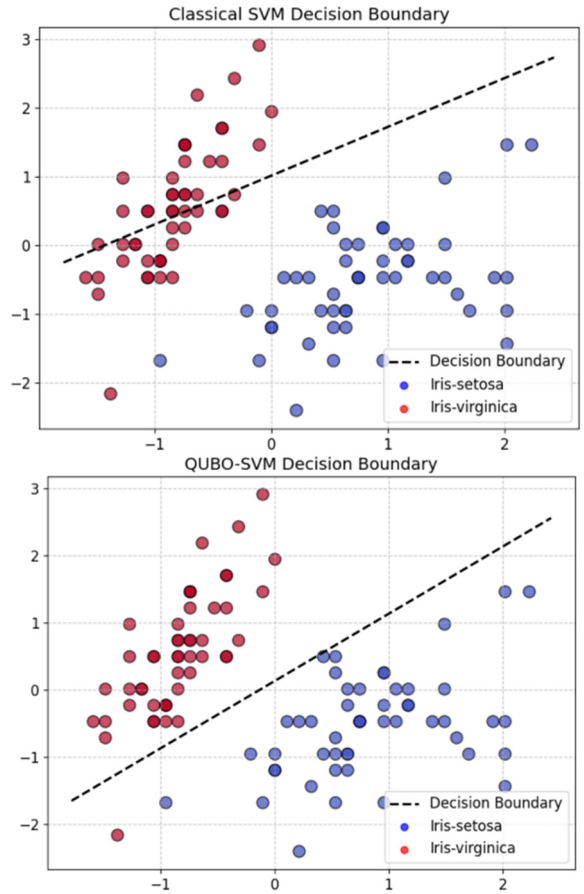

Figure 7. Comparison of classical SVM and QUBO-SVM classification (Iris-setosa and Iris-virginica)

As can be seen from the above figure, QUBO-SVM binary classification results are more accurate. Based on this, we calculate multiple prediction evaluation indicators to assist in illustrating the excellent results.

Table 9. Statistics of outcome evaluation indicators (Iris-setosa and Iris-virginica)

Data set	Recall Rates	F1	Accuracy rate	Precision rate
Training Set	1	1	1	1
Test Set	1	1	1	1

In the same way, we solve for the hyperplane parameters of Iris-versicolor and Iris-virginica:

Table 10. Optimal prediction parameter values (Iris-versicolor and Iris-virginica)

Optimal prediction parameters	Value
w_1	1.026
w_2	2.191
w_3	-5.459
w_4	-7.785
b	0.794
λ	10^7

At this point, the objective function Q reaches the minimum value, where, due to the fusion of data points of Iris-versicolor and Iris-virginica, we draw a comparison diagram of classification results of Iris-versicolor and Iris-virginica between SVM and QUBO-SVM:

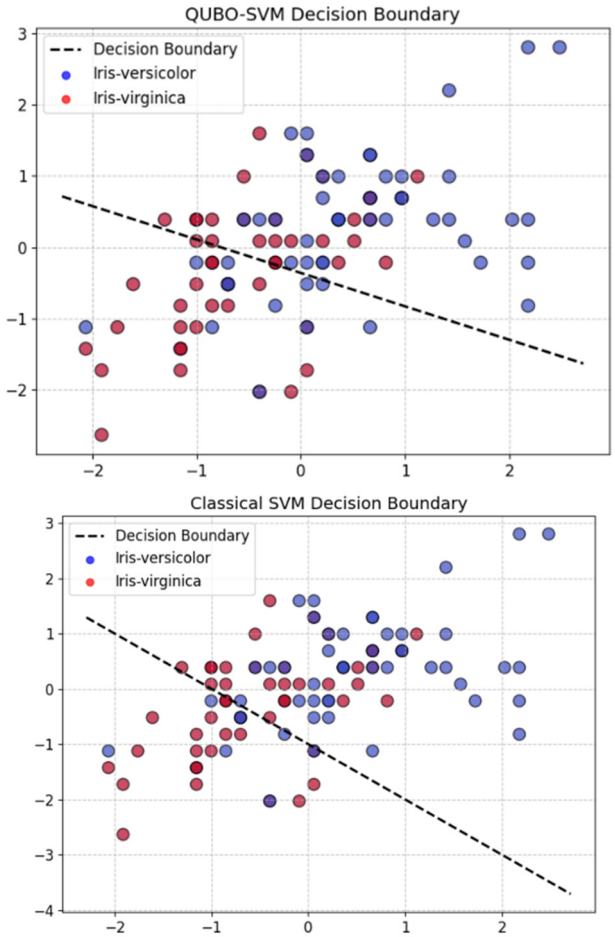

Figure 8. Comparison of classification results between classical SVM and QUBO-SVM (Iris-versicolor and Iris-virginica)

In addition, we calculated a variety of predictive evaluation indicators to assist in illustrating superior classification

performance:

Table 11. Statistics of outcome evaluation indicators (Iris-versicolor and Iris-virginica)

Data set	Recall Rates	F1	Accuracy Rate	Precision rate
Training Set	0.957	0.957	0.957	0.958
Test Set	0.942	0.933	0.933	0.933

In addition, we also visualized the change of Hamiltonian over time in the simulated annealing algorithm during the solving of 3 binary classification problems to reflect the energy state of the system.

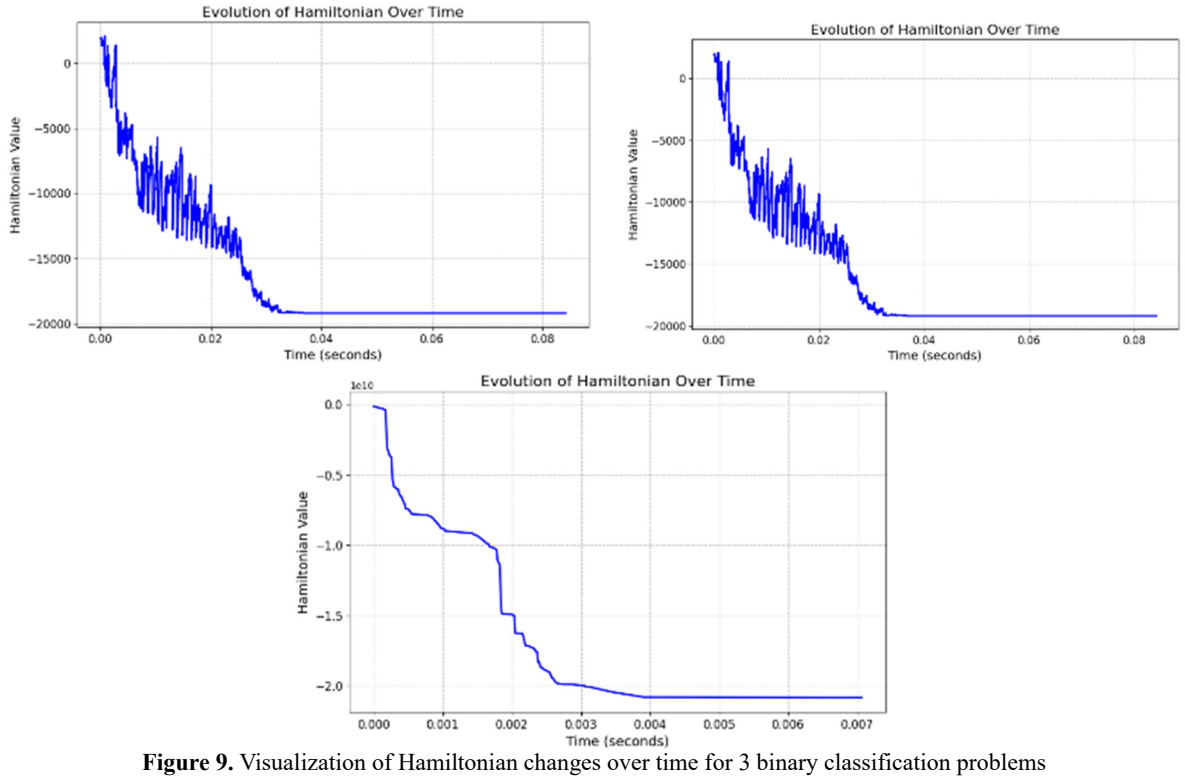

Figure 9. Visualization of Hamiltonian changes over time for 3 binary classification problems

From the figure above, we can see that the Hamiltonian drops rapidly at the beginning, which indicates that the energy of the system decreases rapidly, and then there are some fluctuations. These fluctuations indicate that during the optimization process, the system is exploring different states in order to find the state with lower energy. As time goes on, the rate of decline gradually slows down, and the Hamiltonian tends to stabilize, which may mean that the system has approached or reached a stable state. And Solving time of the QUBO model is between 0.07s and 0.09s. And solving time of the QUBO model is between 0.07s and 0.09s.

4. Model 3: The QUBO Model Based on CNN Image Classification

(1) Establish QUBO Model of CNN for Image Classification of MNIST Handwritten Numerals Dataset

So far, the pattern recognition system based on convolutional neural network is one of the systems with the best performance, especially in the field of handwritten character recognition, and has been used as the evaluation standard for the performance of machine recognition systems. By mining the spatial correlation in the data, convolutional neural network can reduce the number of trainable parameters in the network, so as to improve the backpropagation algorithm efficiency of the forward propagation network.

As shown in the following figure, after the input layer reads images of a unified size, the input image is convolved with three filters and one additive bias. After convolution, three feature maps are generated in layer C1. Then, the four adjacent pixels in the feature map are grouped together to obtain the average value, and then weighted value and bias are added. Three feature maps of layer S2 are obtained by ReLU activation function.

Among them,

$$ReLU(x) = \max(0, x) \tag{29}$$

These maps are then filtered accordingly to get the C3 layer. This layer then produces S4, as does S2. Finally, these pixel values are rasterized, connected into a one-dimensional vector input into a traditional neural network, and then enter the fully connected layer to produce an output.

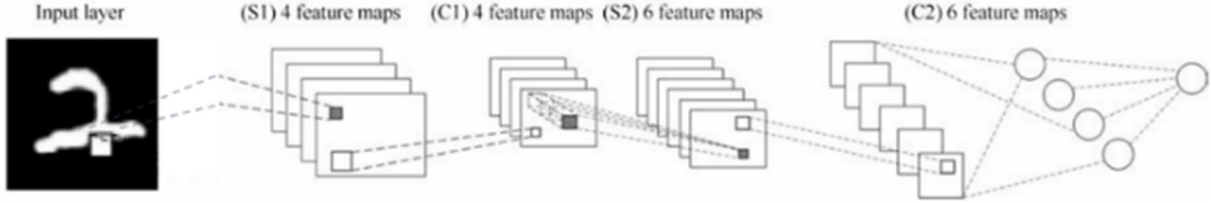

Figure 10. Illustration of convolutional neural networks

Among them, layer C is the convolutional layer, that is, the feature extraction layer. A convolutional layer usually contains multiple feature graphs with different weight vectors, so that a variety of different features can be obtained at the same position. Layer S is the pooling layer, which reduces the resolution of the feature map and reduces the sensitivity of the network output to displacement and deformation through local averaging and downsampling operations. Subsequent convolutional layers and pooling layers are alternately distributed and connected, forming a "double pyramid" structure: the number of feature maps gradually increases, and the resolution of the feature maps gradually decreases.

Convolutional layers are the core of CNN, and each convolutional layer uses several convolutional layers to scan the input image. As for every convolution kernel, the convolution operation computes the weighted sum of the local region between the input image and the convolution kernel. Assuming the size of the input image is $H \times W \times D$ (height, width, depth), the convolution operation uses a convolution kernel of size $k \times k$, depth D, and the number of convolution cores (number of output channels) is F. For each local region X_i and convolution kernel W in the input image X, the convolution operation is as follows [9]:

$$Y_i = \sum_{m=1}^{k} \sum_{n=1}^{k} \sum_{d=1}^{D} W_{m,n,d} X_{i+m-1,j+n-1,d} + b \tag{30}$$

Where, the convolution kernel $W_{m,n,d}$ representing the (m, n, d) element, $X_{i+m-1,j+n-1,d}$ is the (i+m-1, j+n-1, d) pixel of the input image, and b is the offset term.

The pooling layer is used to downsample the feature map output of the convolutional layer. The pooling operation performed here is the maximum pooling, that is, for each $k \times k$ region in the feature map, the maximum value in the region is selected as the output.

Each image in the MNIST dataset corresponds to a digital label from 0 to 9, which is composed of NIST Special Database 1 and NIST Special Database 3. The training set in MNIST selects 30,000 samples from SD-3 and SD-1 respectively. The 60,000 samples were drawn from handwritten data from approximately 250 different individuals, and the 5,000 samples each from SD-3 and SD-1 were similarly selected as the test set. The images in the MNIST dataset are all of size, and the images of some numbers in the dataset are28×28[6]

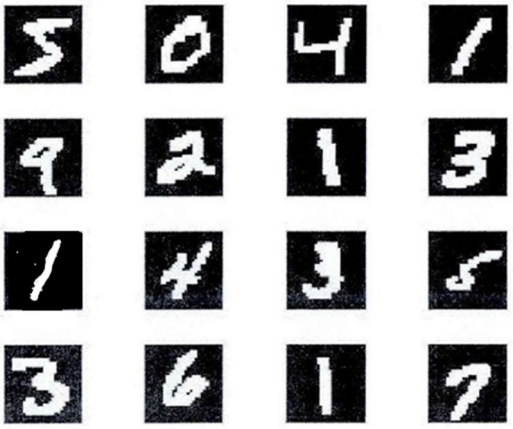

Figure 11. Images of the first 16 numbers in the MNIST data set

In the multi-class classification problem, the objective function is minimize the difference between the predicted result and the true label. Assuming that the model predicts the result is $\hat{y}_i$ and that the actual label is y_i, we use the cross entropy loss function to measure the error [7]:

$$L(\hat{y}_i, y_i) = -\sum_{k=1}^{C} y_i^{(k)} \log\left(\hat{y}_i^{(k)}\right) \tag{31}$$

Where C is the total number of categories, and for this dataset, C is the constant 10; $y_i^{(k)}$is the k-th element of the sample x_i's true label, if the sample belongs to class k, $y_i^{(k)} = 1$, then 0 otherwise; $\hat{y}_i^{(k)}$is the probability that the sample x_i predicted by the model belongs to class k. The output layer of CNN uses *softmax* activation function to calculate the probability of each class, *softmax* activation function is the mathematical expression as follows[8]:

$$\hat{y}_i^{(k)} = \frac{e^{z_i^{(k)}}}{\sum_{j=1}^{C} e^{z_i^{(j)}}} \tag{32}$$

Where, $z_i^{(k)}$ is the raw score of class k output by the fully connected layer, $\hat{y}_i^{(k)}$ is the predicted probability of class k.

For the convolutional layer l and the fully connected layer, the output can be expressed as follows:

$$Z^{(l)} = \mathrm{ReLU}\left(W^{(l)} * X + b^{(l)}\right) \tag{33}$$

Where, $W^{(l)}$ refers to the weight matrix of the convolution layer and the fully connected layer, and X is the output (or input image) of the previous layer and $b^{(l)}$ is the biased term. In addition, y_i denoted as the target value, then the error term of the convolution layer can be expressed as follows:

$$E^{(l)} = \sum_{i} (Z_i^{(l)} - y_i)^2 \tag{34}$$

In order to combine the variables involved in the above theoretical Analysis with QUBO, we defined x_i as the binary activation variable of the i-th neuron, indicating whether the neuron is activated, with the value of 0 or 1; Binary variable x_k for the model prediction class k, indicating whether the sample is classified as class k, taking the value 0 or 1; In addition, we need to replace it with binary W_{ij}, b_i to expression.

To prevent overfitting, we add regularization terms, especially L2 regularization terms.

$$L_{regularization} = \lambda_{reg} \sum_{i,j} {W_{ij}}^2 \tag{35}$$

Based on the above Analysis, we determine that the objective function is:

$$\begin{aligned} Q(x) = &\sum_i Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_i + \lambda_{\text{conv}} \sum_i \left(Z_i^{(i)} - y_i\right)^2 \\ &+ \lambda_{\text{fc}} \sum_i \left(Z_i^{(i)} - y_i\right)^2 + \lambda_{\text{crcoss-entrcopy}} \sum_k y_k \log(x_k) \end{aligned} \tag{36}$$

Where, $\lambda_{\text{conv}}, \lambda_{\text{fc}}, \lambda_{\text{crcoss-entrcopy}}$ is the penalty coefficient that controls the importance of the error term.

On this basis, we determine the constraint conditions, the classification results should meet "only one category is predicted", the activation value of neurons can only be 0 or 1, the discretization into binary decision variables in convolution operation, the calculation rules of weighted sum of neurons, the minimization of cross entropy loss function and regularization constraints, the specific mathematical expressions are as follows:

$$
\begin{cases}
\sum_{k=1}^{C} x_k = 1 \\
x_i \in \{0,1\}, \forall i \\
Z_i^{(\text{conv})} = f\left(\sum_j W_{ij} x_j + b_i\right) \\
Z_i^{(\text{fe})} = f\left(\sum_j W_{ij}^{(\text{fe})} x_j + b_i^{(\text{fe})}\right) \\
W_{ij} \in \{w_1, w_2, \ldots, w_n\}, \forall i, j \\
L_{\text{cross-entropy}} = -\sum_{k=1}^{C} y_k \log(x_k) \\
L_{\text{regularization}} = \lambda_{\text{reg}} \sum_{i,j} W_{ij}^2
\end{cases} \tag{37}
$$

Where, x_k is the predicted binary variable of class k, $w_1, w_2, \ldots, w_n$ is the weight value after discretization.

(2) Solution of Image Recognition Classification Codel Based on Simulated Annealing

We still use the simulated annealing solver built in Kaiwu SDK for solving, in which the parameters in the solver are set as follows:

Table 12. Simulated annealing parameter table

Parameters	Select parameter values
Initial temperature T_0	3000
Cutoff temperature T_{end}	10^{-3}
Cooling factorα	0.90
Iterations per temperature	100
Size limit	100

Select the initial temperature T_0 and the end temperature T_{end}, and select the appropriate cooling coefficient α during the annealing process, so that the iterative depth of each temperature in the annealing process does not exceed 100.

Through solving, we captured part of the images in the MNIST dataset and their recognition results, as shown below:

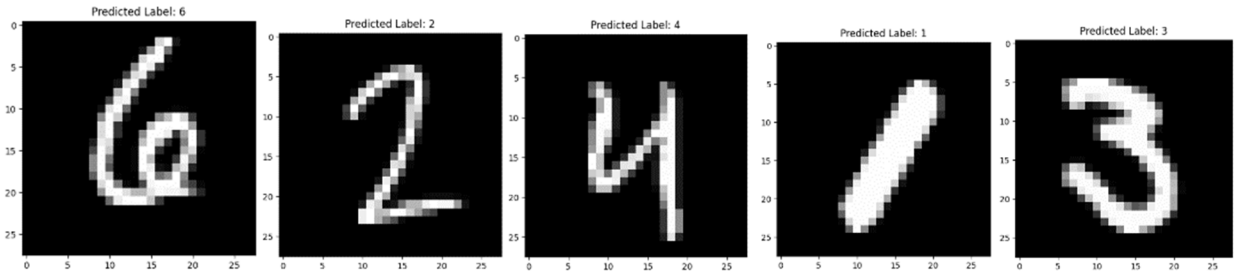

Figure 12. Part of the image recognition results

In addition to this, we visualize the changes in the loss (left) and accuracy (right) of the model training process over the training cycle:

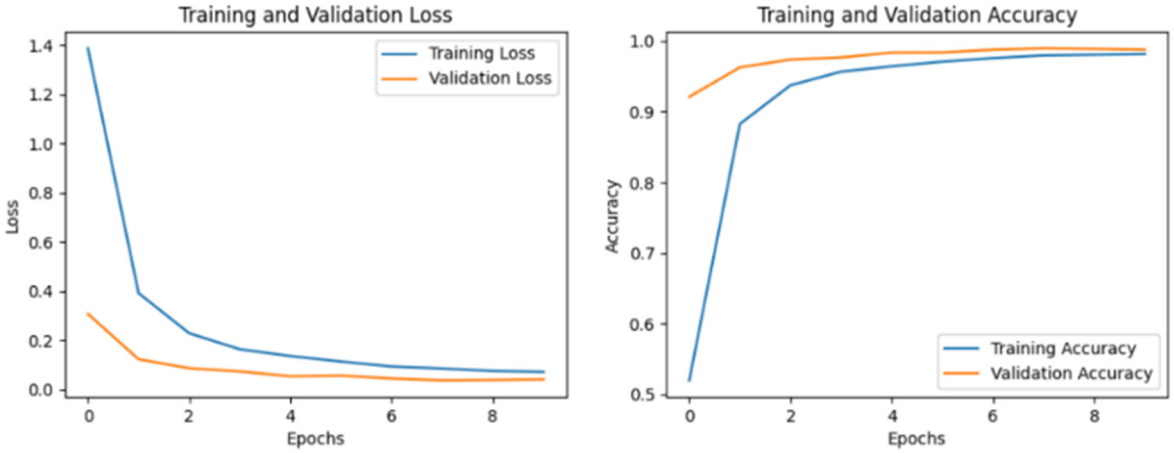

Figure 13. The change of loss and accuracy in the model training process with the training cycle

As can be seen from the figure on the left, both training losses and validation losses decrease as the training cycle increases, indicating that the model is learning and gradually reducing the prediction error. After the initial decline, the validation loss levels off and is always lower than the training loss, indicating that the model performs well on the training set without significant overfitting.

For the figure on the right, both the training accuracy and validation accuracy rise as the training cycle increases, and both approach 1.0, indicating that the model's predictions are becoming more accurate. And the training accuracy and verification accuracy are very close, which further indicates that the model is not overfitting and has good generalization ability on both the training set and the verification set.

In addition, we calculated a variety of predictive evaluation indicators to help explain the excellent performance of image recognition classification results:

Table 13. Results evaluation index value statistics (Iris-versicolor and Iris-virginica)

Data set	recall	F1-score	accuracy	support
Macro avg	0.99	0.99	0.99	10000
Weighted avg	0.99	0.99	0.99	10000

From the table above, we can see that the accuracy, recall rate, and F1 score of the model recognition results are close to 0.99 or 1.00, indicating that the model performs very well on handwritten digit recognition tasks. solving time of the QUBO model is 211s due to the huge amount of data and individual differences in hardware performance.

5. Conclusion

For model 1, we used the monthly computational resource demand data provided by the problem and combined with AIC criterion to establish an autoregressive model of order 3. To predict the data in October, we took predicted resource demand in October as the result when SSE minimises. Then AR model is transformed into QUBO, and the parameters in AR are replaced by binary expressions, and the SSE function of binary variable as the decision variable is taken as the objective function. The minimum value is obtained when SSE equals to 30723.78, and the predicted value in October is 10719.36. In addition, we calculated that the MAPE of the model is 0.709, and the R-squared is 0.933, indicating that the prediction result of the model was excellent. By using Kaiwu SDK, the solving time of the QUBO model was 0.07 seconds.

For model 2, we decomposed the tripartite classification into three binary classification tasks: Setosa vs. Versicolor, Setosa vs. Virginica, and Virginica vs. Versicolor. The objective function minimizes the norm of the normal vector of the SVM hyperplane for each task. The normal vector, bias, and relaxation variables of the hyperplane are discretized into binary variables, with a penalty term and weight introduced. The objective is then transformed into a QUBO, solved using the Kaiwu SDK, and evaluated. Here only demonstrate the normal vector for Setosa vs. Versicolor, it's [-0.269, 0.338, -0.701, -0.749], the bias is -0.247, and the penalty weight is 50. Evaluation metrics, including recall, F1-score, and accuracy, all exceeded 0.93, indicating good model performance. And the QUBO model solving time ranged from 0.07 to 0.09 seconds.

For model 3, we used the MNIST dataset for image classification, applying convolutional neural networks (CNNs) with ReLU activation and cross-entropy loss. We introduced penalty coefficients for the convolutional and fully connected layers, as well as cross-entropy, and minimized the loss for image recognition. To prevent overfitting, L2 regularization was added. The softmax activation function was used in the output layer to calculate category probabilities. Evaluation metrics, including macro avg and weighted avg recall, F1-score, and accuracy, all exceeded 0.98, with a support of 10,000, indicating strong model performance. As QUBO model solving time was around 211 seconds.

This paper fully demonstrates the application of the QUBO model in optimizing machine learning algorithms, which can be extended to other artificial intelligence algorithms and represents a future direction for research.

Acknowledgments

The authors gratefully acknowledge the financial support from 2024 Guangzhou University Student Innovation Training Project of Ministry of Education of P.R.C (202411078145).

References

[1] Wen, J., Wang, Z., Huang, Z. Optical experimental solutions for multi-channel partitioning problems and their applications in computing power scheduling. Sci. China Phys. Mech. Astron.66, 290313 (2023).

[2] Glover, F., Kochenberger, G., Hennig, R., et al. Quantum Bridge Analysis I: A tutorial on formulating and using QUBO models. Ann Opera Studies 314, 141–183 (2022).

[3] Chen Si. Based on machine learning research on electricity demand forecasting and inventory decisions [D]. Shenzhen university, 2021. The DOI: 10.27321 /, dc nki. Gszdu. 2020. 000186.

[4] Chang Yunfeng. Phase Transition and Transmission Dynamics of Ising Model on Complex Networks [D]. Huazhong Normal University,2008.

[5] Qi Hengnian. Overview of Support Vector Machines and their applications [J]. Computer Engineering, 2004, (10):6-9.

[6] Xu Ke. Research on the application of Convolutional Neural networks in image recognition [D]. Zhejiang University,2012.

[7] Zhang Xiaorong. Research on Deep Learning Algorithm Based on Convolutional Neural Network and Its Application[D]. Xidian University, 2015.

[8] Chen Hongxiang. Semantic Image Segmentation Based on Convolutional Neural Networks [D]. Zhejiang University, 2016.

[9] Yang Juyu. Research and Implementation of Object Recognition Based on Convolutional Neural Networks [D]. University of Electronic Science and Technology of China, 2016.