METHODS OF SOLVING OPTIMAL SOLUTIONS OF MATHEMATICAL PROBLEMS WITH ARTIFICIAL INTELLIGENCE METHODS

Qarshi davlat universiteti dotsenti ***Daminova Barno Esanovna****,*
https://orcid.org/0009-0001-4211-6082, barnod@mail.ru,
Kaynarov Fazliddin Zarif o'g'li*,*
Economics and Pedagogical University, Non-State Educational Institution,
Mathematics Department, 3rd year student,
Orcid ID: 0009-0009-9677-1849; Kaynarov.fazliddin@gmail.com

Annotation. *The article shows that solving mathematical problems can be more effectively implemented using artificial intelligence (AI) than traditional methods. Artificial intelligence, in particular neural networks and deep learning (Deep Learning) methods, significantly simplify and accelerate the process of solving mathematical and statistical problems. This work considers the application, capabilities and practical approaches of artificial intelligence methods in solving mathematical problems.*

Keywords: *Artificial intelligence, neural networks, deep learning (Deep Learning), regression, classification, optimization, simulation, differential equations, machine learning, genetic algorithms, mathematical modeling, error functions, physics-Informed Neural Networks (PINNs).*

Аннотация. *В статье показано, что решение математических задач может быть более эффективно реализовано с использованием искусственного интеллекта (ИИ), чем традиционными методами. Искусственный интеллект, в частности нейронные сети и методы глубокого обучения (Deep Learning), значительно упрощают и ускоряют процесс решения математических и статистических задач. В данной работе рассматриваются применение, возможности и практические подходы методов искусственного интеллекта при решении математических задач.*

Ключевые слова: *Искусственный интеллект, нейронные сети, глубокое обучение (Deep Learning), регрессия, классификация, оптимизация, моделирование, дифференциальные уравнения, машинное обучение, генетические алгоритмы, математическое моделирование, функции ошибок, физико-информированные нейронные сети (PINN).*

Complex problems, such as regression, classification, and optimization, can be solved using neural networks, evolutionary algorithms, genetic algorithms, and other machine learning methods. These methods help to identify ambiguous and complex relationships between input and output data. Artificial intelligence methods are also effectively used in complex problems such as simulation, optimization, and differential equations.

Methods for solving mathematical problems using artificial intelligence are currently very popular and are used in many different fields. These methods can be divided into the following main areas:

• Genetic algorithms and simulation methods can be effective for solving mathematical problems. For example, in complex optimization problems, the best solution can be found using genetic algorithms or simulation methods (Monte Carlo simulation).

Simulation and optimization are widely used methods in the fields of artificial intelligence and mathematics, which help in solving many real-world problems. Let's take a closer look at each of these two concepts:

Simulation is the process of creating a real or theoretical model of a system or process and testing its behavior or performance on a computer. This method is used to understand systems or processes, predict them, or make decisions.

• Monte Carlo Simulation: This method allows you to test the behavior of a system using probability distributions. Monte Carlo simulation is a method for modeling the behavior of a system using random numbers. It is used, for example, to model or predict random processes.

• Agent-based simulation: In this method, the system is represented by different agents (such as people, machines, or other objects). Each agent learns its own behavior to create the overall behavior of the system. The agents interact with each other and simulate the system.

• Dynamic simulation: This method aims to simulate how a system changes over time. For example, it is used to model economic or ecological systems. Differential equations and integrals can be used to analyze the change in a system over time.

Helps to understand the behavior of complex systems.

Allows you to study processes or problems that are difficult to test in practice.

Allows you to quickly test many alternative solutions.

Can be computationally and time-consuming.

Models and approaches may not fully reflect reality.

In some cases, the necessary data for creating a model may not be available.

Optimization is the process of finding the best (optimal) solution for a system to achieve a specific goal. This process requires the use of various algorithms and mathematical models. Optimization is used in many areas, such as manufacturing, transportation, marketing, economics, and others.

Linear Optimization. In this type of optimization, the objective function and constraints are linear. For example, it is used in the allocation of resources in production, minimizing costs, or maximizing profits. The simplex algorithm is one of the most popular methods for linear optimization.

Nonlinear Optimization: If the objective function or constraints are nonlinear, then nonlinear optimization methods are used. Solutions to such optimization problems are found using gradient methods or heuristic methods.

Integer Optimization: In this type of optimization, the variables must be integers. For example, it is used in logistics problems when choosing freight transport routes or in the allocation of resources in the production process.

Genetic Algorithms: Genetic algorithms (GAs) are heuristic methods that mimic the processes of natural selection and heredity. These algorithms are useful in solving complex optimization problems because they help solve global optimization problems better than local optimization.

ISSN 3060-4567

Helps find the most efficient and economical solution to the problem.

Allows for efficient allocation of resources and maximum benefit.

Effective in developing solutions for complex systems.

Solutions may not be guaranteed to be exact or global, especially in nonlinear and discrete optimization.

Longer computation time and increased resource requirements when working with large data sets.

Integration of simulation and optimization

The combination of simulation and optimization is used in many practical problems. For example, by simulating complex systems or processes, it is possible to study their behavior and then improve the system using optimization algorithms.

Examples:

• Optimization of transportation networks: When optimizing large transportation networks or logistics systems, simulation can be used to test different scenarios, and then the best transportation routes can be selected using optimization.

• Manufacturing processes: In manufacturing systems, simulation can be used to create different production situations to allocate resources efficiently, and production can be effectively controlled using optimization methods.

Simulation and optimization are very effective tools in artificial intelligence and mathematical modeling. Simulation helps to analyze the behavior of systems, while optimization is used to develop solutions to achieve maximum system efficiency. Using these two methods together gives effective results in solving many complex problems.

Artificial neural networks (ANNs) or decision trees can be used to model and solve linear and nonlinear problems. For example, ANNs can be used to find a solution to a system.

Linear and nonlinear algebra are important branches of mathematics, widely used in many fields, including physics, economics, engineering, and computer science. Both concepts are fundamental tools in analyzing systems and finding solutions. Let's take a closer look at these two areas.

ISSN 3060-4567

Linear algebra is the branch of mathematics that studies vectors, matrices, and linear functions. The most common concepts in this field are vectors, matrices, determinants, and vector fields. Linear algebra is widely used in mathematical modeling and optimization problems.

Objects that represent the direction and length of a line. Vectors often exist in n-dimensional spaces such as R^n.

In linear algebra, matrices are rectangular arrays of numbers or other objects. Matrices can be used to find solutions to systems, transform vectors, and represent linear transformations.

They are used to determine the properties of a matrix. For example, the determinant helps determine the inverse of a matrix.

Linear algebra solves equations such as $Ax = b$, where A is a matrix, x is a vector, and b is a resultant vector.

Gaussian elimination: A method used to solve a system of linear equations. This method helps to find a solution by reducing matrices to their simplest form.

Eigenvalues and Eigenvectors: Eigenvalues and eigenvectors are fundamental concepts in linear algebra and physics. These concepts are used to analyze the properties of a matrix. An eigenvalue is a scalar value that leaves the matrix unchanged, while an eigenvector is a vector that leaves the matrix unchanged.

• Linear transformations: Used to transform vectors or spaces. For example, operations such as rotation or scaling are considered linear transformations.

Optimization problems: Linear algebra is very important in linear optimization problems (such as the Simplex algorithm).

Physics: Linear algebra is used to model quantum mechanics, electrodynamics, and other physical systems.

Machine learning: Models are built using vectors and matrices in regression analysis and classification problems.

Nonlinear Algebra

Nonlinear algebra, on the other hand, is a more complex and broader field than linear algebra, and focuses on the study of nonlinear behavior of systems. It is used to analyze nonlinear equations, nonlinear functions, and complex systems.

Nonlinear equations: Unlike linear equations, nonlinear equations study nonlinear relationships between variables and objective functions. For example, they may contain nonlinear elements such as polynomials, exponential functions, or logarithms.

Nonlinear functions: Many functions in nonlinear algebra describe nonlinear relationships. For example, functions such as $y = x^2$ or $y = \sin(x)$ are considered nonlinear.

Gradient and optimization: The concept of gradient (e.g., gradient descent algorithms) is used to optimize nonlinear systems. These methods lead systems to the minimum or maximum value (minimization or maximization).

Newton's methods: These methods are used to find solutions to systems of nonlinear equations. Newton's method allows you to quickly find a solution if the initial value is close.

Gradient methods: In nonlinear optimization, for example, the descent method is used. In this method, the lowest point is reached along the gradient of the objective function.

Gauss-Seidel methods: Approximating methods for nonlinear systems Stability analysis of systems: Nonlinear algebra is used to analyze the stability of systems, that is, how they respond to changes over time. This is important in modeling physical systems or economic systems.

Machine learning: Nonlinear algebra is used in nonlinear regression, artificial neural networks (ANNs), and other complex models. For example, the activation function for a neural network is often nonlinear.

Physics and Engineering: Nonlinear algebra is used to model many physical processes, such as the deformation of various materials or temperature changes.

Linear algebra provides easy and accurate solutions to systems and equations. It often expresses simple relationships between systems.

Nonlinear algebra, on the other hand, models complex systems, such as dynamical systems and many physical processes. In this area, solutions are more iterative and approximate.

Linear and nonlinear algebra are two main areas of mathematics and are used in many fields. Linear algebra is usually used to work with simple systems and equations, while nonlinear algebra is used to model and optimize complex and dynamical systems. Both areas are essential tools for mathematical modeling and problem solving.

Deep learning can be used to learn mathematical functions and generate solutions based on them. For example, deep learning technologies can be used to solve differential equations or matrices.

Learning mathematical functions using neural networks is one of the few interesting and practical applications of machine learning. Neural networks can be used to learn mathematical functions, that is, to predict the output of a mathematical function based on its input values. This process is especially useful in regression problems, since regression models are designed to predict continuous values based on input variables. The main goal of learning a mathematical function using neural networks is to determine the output of a function given the input data. This means, for example, imagining some functions that resemble mathematical equations or formulas.

The architecture of a neural network for learning a mathematical function can be in the following simple form:

Input Layer: This layer contains the input values of the function, i.e., x (for example, f(x) or y = f(x)).

Hidden Layers: These layers perform the learning of the neural network. They help to extract complex features from the input data.

Output Layer: In this layer, the neural network produces its final output, i.e., the result of the mathematical function.

When learning mathematical functions, the neural network learns the relationship between the input data (x) and the output data (y). This process usually consists of the following steps:

Input data is prepared: The input values, i.e., the x values, and their associated output values, y, are collected.

How a neural network works:

The neural network processes the input data receives and passes it through hidden layers.

In each layer, the input values are processed using weights and biases.

In the output layer, the network produces an output of a mathematical function.

Backpropagation: The network tries to find the initial solution. Then the error is calculated and the weights are updated using backpropagation. This process is needed to train the network.

Final output: The output of the network is the mathematical function learned with the updated weights after the backpropagation process.

Simple Function (For example, $f(x) = x^2$)

Suppose we need to learn the function $f(x) = x^2$. The neural network training for this function can be of the following form:

• Input data: x (for example, 1, 2, 3, 4, 5)

• Output data: $y = x^2$ (1, 4, 9, 16, 25)

A neural network can be configured as follows:

1. The input layer contains the values of x.
2. Through hidden layers, the network learns patterns of the function x^2.
3. The output layer produces the result $y = x^2$.

If our goal is to learn the function sin(x), a neural network can help in learning complex functions such as sin(x). For example:

• Input data: x (e.g. 0, π/2, π, 3π/2, 2π)

• Output data: y = sin(x)

The learning process of a neural network consists of determining the relationship between x and sin(x).

3. Training the model and evaluating the results

To train a neural network to learn a mathematical function, the following steps must be performed:

1. Training set: Create a training set containing the input and output data.

2. Training the model (Training): Train the neural network using the training set. In this process, the network optimizes the weights and learns the relationships between the data.

3. Testing the model (Testing): Evaluate the individual results of the model using the test set and check its generalization capabilities.

4. Error Analysis: Evaluate the accuracy of the model and, if necessary, re-optimize or retrain the network.

4. Practical Examples and Construction

For example, using Python and the TensorFlow or Keras libraries, you can train a neural network to learn mathematical functions. In the following example, we create a neural network trained to learn the function $f(x) = x^2$:

Python Example (with TensorFlow or Keras):

```
import jumpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

# Creating training data
X = np.linspace(-10, 10, 1000)  # Incoming x values
y = X ** 2  # Output y = x^2
# Model creation
model = keras.Sequential([
    layers.Dense(64, activation='relu', input_dim=1),  # 1-layer
    layers.Dense(64, activation='relu'),  # 2-layer
    layers.Dense(1)  # Output layer
])# Compiling the model
model.compile(optimizer='adam', loss='mean_squared_error')
# Model training
model.fit(X, y, epochs=100, batch_size=32)
```

```
# Forecasting using a model
predictions = model.predict(np.array([5]))  # x = 5
print(f"f(5) = {predictions[0][0]} (model output)")
```

Network evaluation

Evaluate the model and analyze its error rate using the test set:

```
# Creating test data
X_test = np.linspace(-10, 10, 100)
y_test = X_test ** 2
# Testing the model
test_loss = model.evaluate(X_test, y_test)
print(f"Test Loss: {test_loss}")
```

The process of learning mathematical functions using neural networks can be very effective in predicting complex functions. A neural network helps to identify complex relationships between data by learning mathematical relationships. This method is widely used in regression problems and analyses based on complex mathematical models.

• Simulation methods and neural networks can be used to solve complex differential equations using artificial intelligence. This method is especially used in physics and engineering problems.

Solving differential equations using artificial intelligence is one of the interesting and practical applications of modern technologies. Differential equations are widely used to express mathematical models, but solving them using traditional methods can sometimes be difficult. Artificial intelligence (AI), in particular neural networks and deep learning methods, are offering new and effective approaches to solving differential equations.

Traditional methods for solving differential equations:

• Analytical solutions: It is possible to find exact solutions for some simple differential equations. For example, there are analytical solutions (formulas) for them.

• Numerical methods: Numerical methods (e.g., Euler's method, Runge-Kutta method) are widely used to solve differential equations. These methods are very efficient, but often require a lot of calculations for high accuracy.

However, sometimes differential equations are complex and uncertain, and it is difficult to find solutions using traditional methods. This is where artificial intelligence and neural networks come to the rescue.

2. Solving Differential Equations Using Neural Networks

Neural networks use the function learning method to solve differential equations. Artificial intelligence, especially using deep learning, is used to model and learn the solution to a differential equation.

Physics-Informed Neural Networks (PINNs)

PINNs are a type of artificial intelligence used to solve differential equations. PINNs find solutions by studying physical laws, that is, the structural properties of differential equations. They usually work in conjunction with differential equations and constraint conditions.

The main principle of PINN:

• In PINN, a neural network is trained to find the exact solutions to a differential equation. The network is based on learning the differential equation and its constraint conditions.

• The network learns time and space coordinates as input data, and the solution to the equation (for example, u(t,x)u(t, x)) as output data.

• To train the model, an error function is constructed, which is used to approximate the exact solution to the equation. The error is calculated mainly using the differential equation, initial and boundary conditions.

The following steps are performed to solve differential equations using PINN:

1. Define the differential equation: For example, an ordinary, especially inverse, differential equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad (1)$$

This equation models the distribution of temperature in time and space.

2. Build a neural network: Build a neural network architecture in the model. The neural network takes time (t) and spatial coordinates as inputs and produces the solution of the equation u(t, x) as output.

3. Train the neural network: Train the model using backpropagation. During the training process, the PINN calculates errors and optimizes the differential equation, initial and constraint conditions.

4. Solve: After training and training the model, the solution can be predicted in time and space.

• Multidimensional systems: PINNs are very effective in solving complex, multidimensional differential equations, as they model higher-dimensional systems better than traditional methods.

• Constraint handling: NNNs work effectively with initial and constraint conditions, which is important for solving differential equations.

• Accuracy: Neural networks can learn well from ambiguous and complex solutions to differential equations.

For example, let's try to solve the diffusion equation as follows:

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} \quad (2)$$

This equation represents the diffusion of the material, where u(x,t) is the concentration of the substance, and D is the diffusion coefficient.

A PINN model can be built as follows:

1. Input: coordinates tt and xx.

2. Output: concentration u(x,t).

3. Error function:

Residual error: The output of the model is u(x,t)u(x,t) and this value is approximated to zero.

Constraint conditions: Initial and constraint conditions (for example, u(0,t)=0, u(x,0)=f(x)).

To train the model and solve the differential equation, we start the training process by providing input data to PINN.

1. Build PINN using Keras or TensorFlow

ISSN 3060-4567

```python
python
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np
# Model creation
class PINN_Model(tf.keras.Model):
    def __init__(self):
        super(PINN_Model, self).__init__()
        self.dense1 = layers.Dense(50, activation='tanh')
        self.dense2 = layers.Dense(50, activation='tanh')
        self.dense3 = layers.Dense(1)
    def call(self, inputs):
        x, t = inputs
        x = tf.concat([x, t], axis=1)
        x = self.dense1(x)
        x = self.dense2(x)
        return self.dense3(x)
# Model creation
model = PINN_Model()
# Compilation
model.compile(optimizer='adam', loss='mean_squared_error')
# Training the created model and solving differential equations
```

Solving differential equations using artificial intelligence, in particular Physics-Informed Neural Networks (PINNs), is an effective and powerful approach. These methods help solve complex systems, multidimensional problems, and constraint conditions compared to traditional numerical methods. PINNs can provide accurate and efficient solutions to differential equations, for example, in the calculation of physical models or in scientific research.

REFERECEN:

1. Kaynarov F. Z. THEORETICAL FOUNDATIONS FOR THE CREATION OF ELECTRONIC TEXTBOOKS FOR DISTANCE EDUCATION //Экономика и социум. – 2024. – №. 2-2 (117). – С. 169-175.

2. Zarif o‘g‘li K. F. CREATING A TEST FOR SCHOOL EDUCATIONAL PROCESSES IN THE ISPRING SUITE PROGRAM //BOSHLANG ‘ICH SINFLARDA O ‘ZLASHTIRMOVCHILIKNI. – С. 84.

3. O‘G‘Li K. F. Z. CREATING A TEST FOR SCHOOL EDUCATIONAL PROCESSES IN THE ISPRING SUITE PROGRAM //Yosh mutaxassislar. – 2023. – Т. 1. – №. 8. – С. 84-87.

4. Kaynarov F. APPLICATION OF MODERN INFORMATION TECHNOLOGIES IN MEDICINE //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.

5. Кайнаров Ф. З. ИННОВАЦИОННЫЕ МЕТОДЫ ПРЕПОДАВАНИЯ ПРИКЛАДНОЙ МАТЕМАТИКИ //Экономика и социум. – 2023. – №. 1-2 (104). – С. 619-622.

6. Raximov N., Primqulov O., Daminova B. Basic concepts and stages of research development on artificial intelligence //2021 International Conference on Information Science and Communications Technologies (ICISCT). – IEEE, 2021. – С. 1-4.

7. Якубов М. С., Даминова Б. Э. Совершенствование системы образований на основе применение цифровых технологий //Евразийский журнал математической теории и компьютерных наук. – 2022. – Т. 2. – №. 2. – С. 4.

8. Даминова Б. Э. Сравнительный анализ состояния организации многоуровневых образовательных процессов //Экономика и социум. – 2023. – №. 1-2 (104). – С. 611-614.

9. Daminova B. E. GAUSS AND ITERATION METHODS FOR SOLVING A SYSTEM OF LINEAR ALGEBRAIC EQUATIONS //Экономика и социум. – 2024. – №. 2 (117)-1. – С. 235-239.

ISSN 3060-4567

10. Daminova B., Tolipova M., Axadilloyeva Z. Chiziqli algebraik tenglamalar sistemasini gauss va iteratsion yechish usullari //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.

11. Даминова Б. Э. и др. ОБРАБОТКА ВИДЕОМАТЕРИАЛОВ ПРИ РАЗРАБОТКЕ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ //Экономика и социум. – 2024. – №. 2-2 (117). – С. 435-443.

12. Daminova B. E., Oripova M. O. METHODS OF USING MODERN METHODS BY TEACHERS OF MATHEMATICS AND INFORMATION TECHNOLOGIES IN THE CLASSROOM //Экономика и социум. – 2024. – №. 2 (117)-1. – С. 256-261.

13. GAUSS D. B. E. ITERATION METHODS FOR SOLVING A SYSTEM OF LINEAR ALGEBRAIC EQUATIONS //Экономика и социум. – 2024. – №. 2. – С. 117.

14. Даминова Б. Э., Якубов М. С. Развития познавательной и творческой активности слущателей //Международная конференция" Актуальные проблемы развития инфокоммуникаций и информационного общества. – 2012. – С. 26-27.06.

15. Student M. D. et al. THE ROLE OF MODERN INFORMATION AND COMMUNICATION TECHNOLOGIES IN TEACHING LESSONS IN MATHEMATICS AND COMPUTER SCIENCE //Экономика и социум. – 2024. – №. 2-2 (117). – С. 88-93.