COMPUTER SCIENCE, COMPUTER ENGINEERING AND MANAGEMENT

DOI - 10.32743/UniTech.2024.122.5.17487

ANALYSIS OF INTELLIGENT METHODS AND ALGORITHMS FOR DECISION MAKING UNDER UNCERTAINTY BASED ON NEURAL NETWORKS AND EVOLUTIONARY MODELING

Gulnora Buronova
Associate Professor,
Department of Information Systems and Digital Technologies,
Bukhara State University,
Uzbekistan, Bukhara
E-mai: evrikiy@list.ru

АНАЛИЗ ИНТЕЛЛЕКТУАЛЬНЫХ МЕТОДОВ И АЛГОРИТМОВ ПРИНЯТИЯ РЕШЕНИЙ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ И ЭВОЛЮЦИОННОГО МОДЕЛИРОВАНИЯ

Буронова Гулнора Ёдгоровна
доц. кафедры иинформационных систем и цифровых технологий,
Бухарский государственный университет,
Республика Узбекистан, г. Бухара

ABSTRACT

Neural network models are one of the effective tools for making predictions and solving large-scale problems. The database can provide high results in decision support systems working with incomplete, difficult to formalize and uncertain data. Currently, various models of artificial neural networks have been created, which have been successfully developed, improved and implemented in solving various problems. is hampered by difficulties inherent in the neural network paradigm, but this problem can be overcome by hybridizing it with evolutionary modeling. the main difficulties in training artificial neural networks are related to the task of finding the global extremum of the error function, and attempts to solve it have contributed to the development of neuroevolutionary theory, which studies hybrid forms of neural network tuning using evolutionary algorithms.

АННОТАЦИЯ

Нейросетевые модели являются одним из эффективных инструментов для прогнозирования и решения масштабных задач. База данных может обеспечить высокие результаты в системах поддержки принятия решений, работающих с неполными, трудно формализуемыми и неопределенными данными. В настоящее время созданы различные модели искусственных нейронных сетей, которые успешно разрабатываются, совершенствуются и реализуются при решении различных задач, что затруднено трудностями, присущими нейросетевой парадигме, но эту проблему можно преодолеть путем гибридизации ее с эволюционным моделированием. Основные трудности обучения искусственных нейронных сетей связаны с задачей поиска глобального экстремума функции ошибок, а попытки ее решения способствовали развитию нейроэволюционной теории, изучающей гибридные формы настройки нейронных сетей с помощью эволюционных алгоритмов.

Keywords: neural network, decision support systems, evolutionary modeling, neuroevolutionary theory, Adam optimization algorithm, RMSprop optimization algorithm.

Ключевые слова: нейронная сеть, системы поддержки принятия решений, эволюционное моделирование, нейроэволюционная теория, алгоритм оптимизации Адама, алгоритм оптимизации RMSprop.

1. Introduction

Currently, various models of artificial neural networks have been created, which have been successfully developed, improved and implemented in solving various problems. is hampered by difficulties inherent in the neural network paradigm, but this problem can be overcome by hybridizing it with evolutionary modeling. the main difficulties in training artificial neural networks are related to the task of finding the global extremum of the error function, and attempts to solve it have contributed to the development of neuroevolutionary theory, which studies hybrid forms of neural network tuning using evolutionary algorithms. Most of the known Euro-evolutionary methods are applicable to certain types

Библиографическое описание: Buronova G.Y. ANALYSIS OF INTELLIGENT METHODS AND ALGORITHMS FOR DECISION MAKING UNDER UNCERTAINTY BASED ON NEURAL NETWORKS AND EVOLUTIONARY MODELING // Universum: технические науки : электрон. научн. журн. 2024. 5(122). URL: *https://7universum.com/ru/tech/archive/item/17487*

of neural networks. When placing a number of input restrictions on artificial neural networks and changing all its parameters in the process of evolution, the parameters of the neural network are selected for each specific task. Empirically, this does not always lead to optimal results and requires a lot of time and the involvement of specialists. In this regard, issues of automatization of topology and process selection, setting the parameters of neural networks provide an opportunity to solve the problems of implementation of the neuroevolutionary approach based on practical and scientific research [1].

Scientific and applied research in this field may include the development of new optimization methods for adjusting the parameters of neural networks, the study of effective strategies for the evolution of neural networks, and the application of neuroevolutionary approaches to various machine learning and artificial intelligence problems[2].

Neural network parameter tuning is an important aspect of neural network training, which involves choosing optimal values for various parameters such as learning rate, number of training cycles, network architecture, etc. This can be done using optimization methods such as gradient descent or optimization algorithms such as Adam or RMSprop[3].

2. Material and methods

Many famous scientists, including V. McCulloch, V. Pitts, M. Minsky, D. Hebb, F. Rosenblatt, participated in the creation of the theory of artificial neural networks in different periods.

Various ANN models are authored by T. Kohonen, A. Galushkin, K. Fukushima, D. Hopfield, S. Bartsev, V. Okhonin and others. Evolutionary modeling methods J.G. are presented in the works of scientists such as Holland, N.A. Barrichelli, L.J. Vogel, A. Fraser. The issues of ANN hybridization and evolutionary modeling methods are covered in the works of the following scientists: V. Dobrynin, S. Ulyanov, A. Mishin, G. Beni, D.E. Rumelhart, L. Wang, I. Rechenberg, J. Miller, K. Stanley, R. Miikkulainen. The authorship of the most effective neuroevolutionary methods belongs to such famous scientists as F.Pazeman, P.Angelin, G.Saunders, G.Scher, L.Schaeffer, F.Gruau, Ks. Yao, Y. Li, H. Kitano, S. Nolfi, D. Parisi, L. Elman. Despite the existence of a large amount of work in this field, there is a need to develop new neuroevolutionary methods and algorithms that significantly expand the possibilities of neuroevolution and increase the efficiency of decision support systems.

Adjusting the parameters of neural networks is very important for training neural networks. It is necessary to select the necessary values for the learning speed of neural networks, the number of training cycles, network architecture parameters. This is done using optimization methods such as gradient descent or optimization algorithms such as Adam or RMSprop.

As for the difficulties in implementing the neuroevolutionary approach, they may include efficiently finding optimal hyperparameters for evolutionary algorithms, choosing suitable fitness functions, processing large amounts of data, and other technical and computational problems.

The Adam optimization algorithm is one of the gradient optimization algorithms and is widely used in studying the parameters of neural networks. This algorithm helps with open source optimization and typically uses a smoothed gradient.

Adam's algorithm is distinguished by its open-source optimization capabilities and generally produces good results for entry-level steps. Among its main advantages are advantages such as calculation of automatic stages (learning rate), speed of the process and adaptation of variables in optimization.

This algorithm is known as one of the widely used gradient optimization algorithms and has been successfully used in many types of practice.

RMSprop (Root Mean Square Propagation) optimization algorithm is one of the widely used algorithms in gradient optimization. This algorithm helps in open source optimization and is used to optimize the same parameters as Adam's algorithm.

The RMSprop algorithm has the following main features:

1. Adaptive stages (Learning Rates): The RMSprop algorithm calculates individual adaptive stages for each parameter. This helps in finding the optimal steps for each parameter.

2. Root Mean Square: The algorithm looks at the root mean square of the previous gradient and compares the new gradient with these values.

3. Epsilon additional value: An additional value called epsilon is used in the formulas of the algorithm, which prevents division by zero when calculating the average value of the squares of the gradient.

The Adam and RMSprop algorithms are one of the two gradient optimization methods, with their main features and comparison results.

The Adam algorithm has the following properties:

1. Adaptive steps (Adaptive Learning Rates): Adam's algorithm calculates separate adaptive steps for each parameter. This helps in finding the optimal steps for each parameter.

2. Momentum: The algorithm supports momentum, that is, it keeps the average values of the previous gradient and connects to the new gradients.

3. Bias correction: The Adam algorithm supports bias correction, which reduces problems associated with previous steps in the optimization process.

Comparison results:

- Adam's algorithm Compared to the RMSprop algorithm, adham has advantages in calculating adaptive steps and supporting momentum and bias correction during optimization. This increases the wide applicability of Adam's algorithm.
- The RMSprop algorithm takes into account the mean values of the squares of the previous gradients and thereby optimizes the new gradients. This ensures that the RMSprop algorithm has its own methods of optimization.

Each algorithm has its own advantages and disadvantages, and it is recommended to test them in practice to decide in which case they should be used.

The RMSprop algorithm is widely used in open source optimization and helps to find optimized gradient steps. This algorithm achieves the same goals as Adam's algorithm, but its optimization formulas and parameters are different from Adam's.

To code Adam and the RMSprop algorithm in Python, we use high-level libraries such as TensorFlow and PyTorch:

For TensorFlow:

```
import tensorflow as tf
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
```

For PyTorch:

```
import torch
import torch.optim as optim
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

These codes are suitable for running the Adam optimization algorithm in TensorFlow and PyTorch. Adjust `learning_rate` to the desired value. In general, these codes are useful for learning neural networks.

Conclusion

Forecasting methods in complex systems are subject to errors, and we always try to reduce these errors. One of the main disadvantages of using ANN to solve nonlinear problems is the weakness of learning a large number of patterns. This drawback is overcome by using this method, mainly due to the ability to select good data. The presented ones are mainly suitable for cases where the structure of the problems is not predetermined and the use of classical methods based on existing patterns (theories) is not recommended. Due to the stochastic search algorithm of GA and the ability of ANN to identify the pattern of complex systems, this model can be used to determine the optimal combination in DSS systems, to reduce modeling variables in engineering processes, to solve robot decision-making problems, etc.

References

1. Alvaro, V., A hybrid linear-neural model for time series forecasting, IEEE Transactions on Neural Networks, V (11), p.p. 1402-1412. 2000.
2. Demuth H., Beale M., Neural Network Toolkit for Use with MATLAB User's Guide, Version 4, MathWorks, Inc. 2003.
3. Goldberg DE. Genetic algorithm in search, optimization and machine learning. Addison Wesley Publishing Co. Annual International Conference on Industrial Engineering Theory 28-31 December 1998.
4. Kohzadi, N., Boyd, M., Kermanshahi, B. and Kaastra, I., Artificial Neural Network and Time Series for Commodity Price Forecasting comparison of models. , Neurocomputing, vol. 15, 1996.
5. Menhaj MB. Fundamentals of neural networks. Amirkabir University. Pub. (in Persian). 2002.
6. Thierens D & Goldberg D. Elitist recombination: Integrated selection recombination GA. IEEE

4 Conclusions

Forecasting methods in complex systems tie with errors and we always try to reduce these errors. On of the main drawbacks for application of ANN in solving nonlinear problems is it’s weakness in learning with large amount of patterns. This drawback is mainly overcome by using this method because of its ability in selecting Good

Data. The presented is basically appropriate for the cases where the structure of problems is not predetermined and the use of classic methods that are based on preexisting patterns (theories) is not recommended. Due to stochastic search algorithm of GA and capability of ANN in pattern recognition of complex systems, this model can be used for determination of optimal combination in DSS systems, by reducing the modeling variables in engineering processes, robots decision making problems, etc.

Reference:

1. Alavi H, Ghaffari Saadat M.H., Using Genetic Algorithm in Neural Network Based Feature Selection for Vibration Monitoring of a Gear Train, proceeding of vetomac 03 and ACSIM 2004
2. Alvaro, V., A hybrid linear- neural model for time series forecasting, IEEE Transactions on Neural Network, V (11), p.p. 1402-1412. 2000
3. Demuth H., Beale M., Neural Network Toolbox for Use with MATLAB User’sGuide, Version 4 , MathWorks, Inc. 2003
4. Goldberg DE. Genetic algorithm in search, optimization and machine learning. Addison Wesley publishing Co. 1989
5. Karokawa, T., Ikeda, Y., Nomura, Sh., Hybrid Method of Neural Network and Genetic
6. Algorithm for Stuck Trading, The 3rd. Annual International Conference on Industrial Engineering Theories December 28-31, 1998
7. Kohzadi, N., Boyd , M., Kermanshahi, B. and Kaastra, I., A Comparison of Artificial Neural Network and Time Series Models for Forecasting Commodity Prices, Neurocomputing, Vol. 15, 1996.
8. Menhaj MB. Fundamentals of neural networks. Amirkabir Univ. Pub. (in Persian). 2002
9. Thierens D & Goldberg D. Elitist recombination: An integrated selection recombination GA. IEEE