A Backprogagation Artificial Neural Network Software Program for Data Classification in Civil Engineering Developed in .NET Framework

SỬ DỤNG PHẦN MỀM MẠNG NƠ-RON THẦN KINH NHÂN TẠO CHO PHÂN LOẠI DỮ LIỆU TRONG NGÀNH XÂY DỰNG ĐƯỢC PHÁT TRIỂN TRÊN NỀN TẢNG .NET

Hoàng Nhật Đức[1], Lê Công Hải[2]

[1]Viện Nghiên Cứu Phát Triển Công Nghệ Cao, Đại học Duy Tân
Institute of Research and Development, Duy Tan University

[2]Công Ty TNHH Xây Dựng và Đầu Tư Hải Nam, Thành phố Đồng Hới, Quảng Bình
Hai Nam Construction and Investment LLC, Dong Hoi City, Quang Binh

Email các tác giả: hoangnhatduc@dtu.edu.vn (Hoàng Nhật Đức), hailecong1@gmail.com (Lê Công Hải).

Abstract

This study aims at developing an Artificial Neural Network (ANN) software program used for data classification in civil engineering. The backpropagation algorithm with momentum and regularization is used to train the ANN. The software program has been developed in Visual C# .NET and tested with several datasets. Experimental results show that the software program is a capable tool for pattern recognition.

Key words: Data Classification; Civil Engineering; Visual C#; Backpropagation.

Tóm tắt

Nghiên cứu này nhằm phát triển chương trình phần mềm Mạng thần kinh nhân tạo (ANN) được sử dụng để phân loại dữ liệu trong ngành xây dựng. Thuật toán lan truyền ngược được sử dụng để huấn luyện ANN. Chương trình phần mềm đã được phát triển trong Visual C # .NET và được thử nghiệm với hai bộ dữ liệu. Kết quả thử nghiệm cho thấy chương trình phần mềm là một công cụ tốt cho nhiệm vụ phân loại dữ liệu.

Từ khóa: Phân loại dữ liệu, Xây Dựng; Ngôn Ngữ C #; Lan Truyền Ngược.

1. Introduction

In the fields of machine learning and statistics, pattern classification is the problem of assigning a new observation to one of existing class labels. A data classification model is trained on the basis of a training data set containing observations with known categories. If there are two existing class labels, the problem is known as binary classification. With more than 2 labels, the problems are generally described as multi-class pattern classifications. Both binary and multi-class pattern classifications are widely observed in civil engineering and in other fields of study [1-8]. Examples of pattern recognition in civil engineering are assigning a given vector of input features to different types of pavement cracks [9], different behaviours of steel beam web panels subjected to concentrated loads [10], soil liquefaction [4] etc.

Among machine learning algorithms, ANN is particularly highly useful for pattern recognition. Essentially, ANN is a sophisticated network of individual learning units called neurons. By combining the information represented as connecting weights, the overall ANN model is able to generalize complicated relationships between the input features and output data of class labels. It is because ANN possesses the capability of a universal function approximator [11]. Moreover, the training algorithm of ANN can be accomplished in an online manner; this means that the ANN model can be updated gradually and ANN can be a very effective model to handle large-scale and multivariate data sets.

Although there are more advanced method of ANN training, the conventional method of gradient descent with backpropagation is still an effective way to establish an ANN model [12]. Backpropagation is capable to help ANN models to be trained quickly with acceptable accuracy in many applications [13]. Therefore, ANN models trained gradient descent with backpropagation can be a useful tool for both educational and practical purposes. Moreover, due to a limiting access to open software packages for implementing ANN based multi-class recognition, this study aims at developing an Artificial Neural Network (ANN) software program used for multi-class data classification. The backpropagation algorithm with momentum and regularization is used to train the ANN. The software program has been developed in Visual C# .NET and verified by two datasets. The rest of the paper is organized as follows: the second section briefly reviews the formulation of an ANN model; two application cases of the newly developed program are demonstrated in the third section; concluding remarks of this paper are stated in the final section.

2. Backpropagation Artificial Neural Network for Data Classification

Backpropagation Artificial Neural Network (BPANN) is a machine learning based classification method inspired from biological neural networks. An ANN model simulates the knowledge acquisition and inference processes of the human brain [14]. Based on previous works [6,15,16], BPANN is shown to be highly effective in coping with complex nonlinear data modeling problems.

A BPANN typically consists of the input, hidden, and output layers [17]. The hidden layers contain a set of artificial neurons; the interconnected artificial neurons has a crucial role of identify the structure in the data to compute the class labels of each data instance in the output layer. Using a BPANN model, data classification tasks boils down to establishing a discrimination function $f: X \in R^D \rightarrow Y \in R^O$ where D is the number of input pattern and O denotes the number of class labels [12]. A BPANN model used for pattern classification is described as follows [18]:

$$f(X) = b_2 + W_2 \times (f_A(b_1 + W_1 \times X)) \quad (1)$$

where W_1 and W_2 are weight matrices of the hidden layer and the output layer, respectively; N is the number of artificial neurons in the hidden layer; $b_1 = [b_{11}, b_{12,}\ldots, b_{1N}]$ and b_2 denotes a bias vector of the hidden layer and of the output layer, respectively. f_A denotes an activation function (e.g. log-sigmoid function).

It is noted that model parameters of BPANN, stored W_1, W_2, b_1, and b_2, are adapted via the backpropagation process [19,17]. There are three commonly employed styles for training a BPANN model: the online, the batch, and the mini-batch [20]. The first style updates the connection weights when they encounter a single training example. The second style updates all the connection weights at once with all collected training records. The mini-batch method splits the whole training data into small batches that are employed to compute the ANN model error and update its weights accordingly. The online update can help to construct a model with high accuracy; however, this method is can be slow to converge. In this study, we aim at developing ANN models to solve small and medium sized datasets; hence, we utilize the online style. Moreover, the backprogation with momentum and regularization is implemented to adapt network weights. More details of the training algorithms are provided in the previous work of Kim [21], Hoang, Tien Bui [22], and Tien Bui et al. [23].

3. Software Program Applications

The software program has been developed in Visual C# .NET Framework 4.6.2. The graphical user interface of the software program is shown in **Fig. 1**. The software implementation is divided into four modules of data loading, ANN model configuration, model training, and model testing. In the first module (see **Fig. 2**), the datasets of training input, training output, testing input, and testing output are loaded. In the next module, an ANN model configuration is specified by selecting the number of class label, the number of neurons in its hidden layer, the number of training epochs, the learning rate, the momentum coefficient, and the regularization coefficient.

The regularization coefficient is used to diminish overfitting phenomenon [21]. The numer of neurons can be roughly selected to be $2D/3 + N_o$ where D is the number of input features and N_o is the number of class labels. After the training and prediction phases of an ANN model are accomplished, the user can viewed the model parameters in the form of weight matrices and the detailed predictions of class labels for all data samples (see **Fig. 3**).

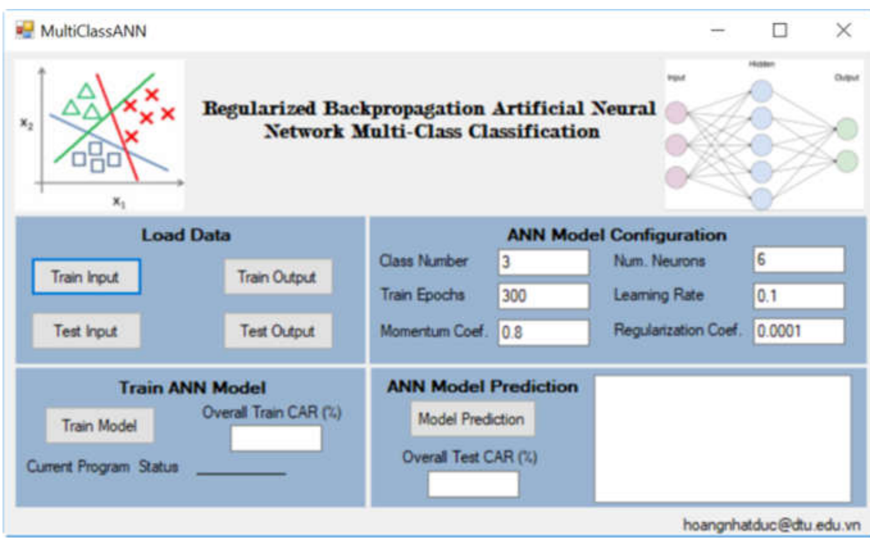


Fig. 1 Graphical user interface of the software program

In addition, before being separated into the datasets of training input, training output, testing input, and testing output, the original dataset should be normalized via Z-score or standard score data normalization method. The standardized x_N of a raw data x is calculated as follows:

$$x_N = \frac{x - M_x}{S_x} \qquad (2)$$

where M_x and S_x denote the mean and standard deviation of x.

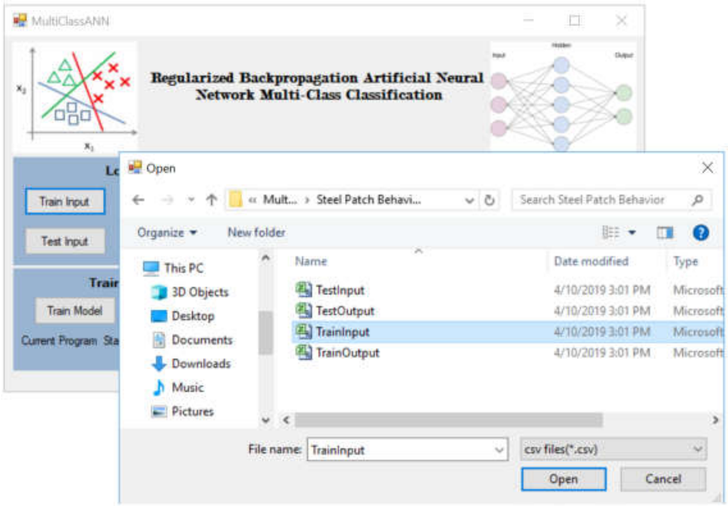
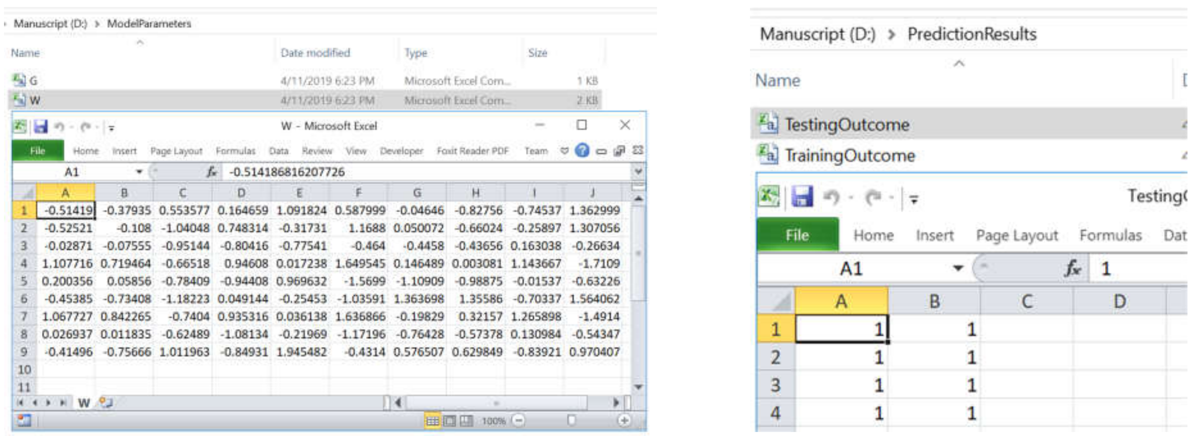


Fig. 2 Data separation



(a) (b)

Fig. 3 Training and testing results: (a) Parameters of the ANN model, (b) Actual and predicted class labels

After being developed, the ANN software program is verified with two case studies in civil engineering which are prediction of steel beams patch load behaviour and groutability prediction. The first application involves 252 data instances reported in Fonseca et al. [10]. The second application utilizes the dataset collected by Liao et al. [24]; this dataset includes 240 data samples. The ANN prediction performances in these two applications are reported in **Fig. 4** and **Fig. 5** which show the prediction accuracy for each individual class. It is observable that the developed program has attained good classification acuracy rate (CAR) results: CAR > 90% for the first application and CAR > 76% for the second application.

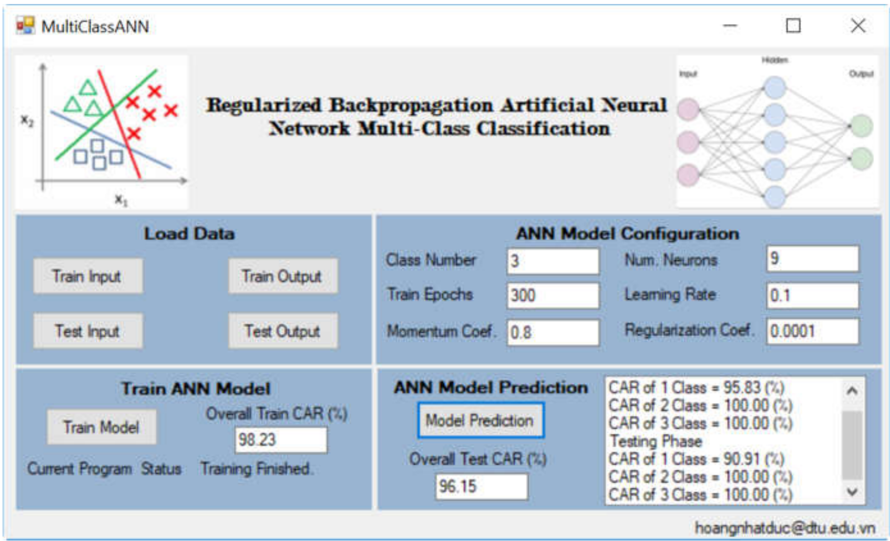


Fig. 4 Prediction performance of the first application (steel beams patch load behaviour)

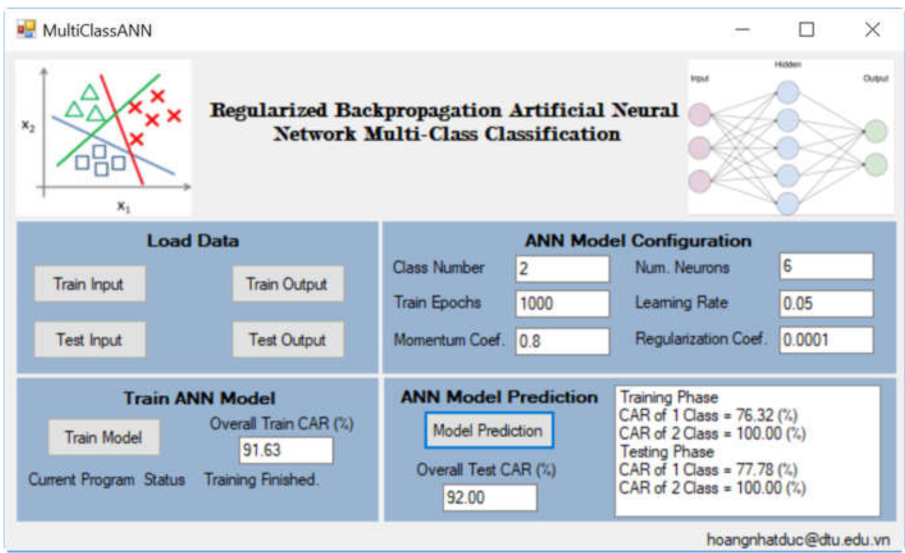


Fig. 5 Prediction performance of the second application (groutability prediction)

4. Conclusion

Pattern classification analysis is an important task in civil engineering. This study develops a software program based on BPANN machine learning method for data classification. The applicability of the software program has been demonstrated via two case studies of steel beams patch load behaviour and groutability estimation. Good classification results demonstrate that the software program can be a useful tool for researchers and engineers for modeling other problems in the field of civil engineering.

Supplementary materials

The compiled program and the experimental datasets can be asssessed via:
https://github.com/NhatDucHoang/MultiClassBPANN

References

1. Tien Bui D, Hoang N-D, Samui P (2019) Spatial pattern analysis and prediction of forest fire using new machine learning approach of Multivariate Adaptive Regression Splines and Differential Flower Pollination optimization: A case study at Lao Cai province (Viet Nam). Journal of Environmental Management 237:476-487. doi:https://doi.org/10.1016/j.jenvman.2019.01.108

2. Tran T-H, Hoang N-D (2018) Predicting algal appearance on mortar surface with ensembles of adaptive neuro fuzzy models: a comparative study of ensemble strategies. International Journal of Machine Learning and Cybernetics. doi:10.1007/s13042-018-0846-1

3. Hoang N-D, Nguyen Q-L, Bui DT (2018) Image Processing-Based Classification of Asphalt Pavement Cracks Using Support Vector Machine Optimized by Artificial Bee Colony. Journal of Computing in Civil Engineering 32 (5):04018037. doi:doi:10.1061/(ASCE)CP.1943-5487.0000781

4. Zhang W, Goh ATC (2018) Chapter 3 - Assessment of Soil Liquefaction Based on Capacity Energy Concept and Back-Propagation Neural Networks. In: Samui P, Kim D, Ghosh C (eds) Integrating Disaster Science and Management. Elsevier, pp 41-51. doi:https://doi.org/10.1016/B978-0-12-812056-9.00003-8

5. Tang XW, Bai X, Hu JL, Qiu JN (2018) Assessment of liquefaction-induced hazards using Bayesian networks based on standard penetration test data. Nat Hazards Earth Syst Sci 18 (5):1451-1468. doi:10.5194/nhess-18-1451-2018

6. Hanna AM, Ural D, Saygili G (2007) Neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data. Soil Dynamics and Earthquake Engineering 27 (6):521-540. doi:https://doi.org/10.1016/j.soildyn.2006.11.001

7. Deng S, Wang X, Zhu Y, Lv F, Wang J (2019) Hybrid Grey Wolf Optimization Algorithm Based Support Vector Machine for Groutability Prediction of Fractured Rock Mass. Journal of Computing in Civil Engineering 33 (2):04018065. doi:doi:10.1061/(ASCE)CP.1943-5487.0000814

8. Tekin E, Akbas SO (2017) Predicting groutability of granular soils using adaptive neuro-fuzzy inference system. Neural Computing and Applications. doi:10.1007/s00521-017-3140-3

9. Hoang N-D, Nguyen Q-L (2018) Fast Local Laplacian-Based Steerable and Sobel Filters Integrated with Adaptive Boosting Classification Tree for Automatic Recognition of Asphalt Pavement Cracks. Advances in Civil Engineering 2018:17. doi:10.1155/2018/5989246

10. Fonseca ET, Vellasco PCGdS, Vellasco MMBR, de Andrade SAL (2008) A neuro-fuzzy evaluation of steel beams patch load behaviour. Advances in Engineering Software 39 (7):558-572. doi:https://doi.org/10.1016/j.advengsoft.2007.07.005

11. Ojha VK, Abraham A, Snášel V (2017) Metaheuristic design of feedforward neural networks: A review of two decades of research. Engineering Applications of Artificial Intelligence 60 (Supplement C):97-116. doi:https://doi.org/10.1016/j.engappai.2017.01.013

12. Bishop CM (2011) Pattern Recognition and Machine Learning (Information Science and Statistics) Springer (April 6, 2011), ISBN-10: 0387310738,

13. Sexton RS, Gupta JND (2000) Comparative evaluation of genetic algorithm and backpropagation for training neural networks. Information Sciences 129 (1–4):45-59. doi:10.1016/s0020-0255(00)00068-2

14. Tran T-H, Hoang N-D (2016) Predicting Colonization Growth of Algae on Mortar Surface with Artificial Neural Network. J Comput Civ Eng 30 (6):04016030. doi:10.1061/(ASCE)CP.1943-5487.000059

15. Montavon G, Orr G, Müller K-R (2012) Neural Networks: Tricks of the Trade. Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-35289-8

16. Hoang N-D, Tien Bui D (2017) Chapter 18 - Slope Stability Evaluation Using Radial Basis Function Neural Network, Least Squares Support Vector Machines, and Extreme Learning Machine. In: Handbook of Neural Computation. Academic Press, pp 333-344. doi:https://doi.org/10.1016/B978-0-12-811318-9.00018-1

17. Hagan MT, Demuth HB, Beale MH, Jesús OD (2014) Neural Network Design (2nd Edition). Martin Hagan; 2 edition (September 1, 2014), ISBN-10: 0971732116,

18. Beale MH, Hagan MT, Demuth HB (2012) Neural Network Toolbox User's Guide. The MathWorks, Inc

19. Beale MH, Hagan MT, Demuth HB (2018) Neural network toolbox user's guide, https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf (Last accessed 04/28/2018). The MathWorks, Inc,

20. LeCun YA, Bottou L, Orr GB, Müller K-R (2002) Efficient BackProp. In: Grégoire Montavon GBO, Klaus-Robert Müller (ed) Lecture Notes in Computer Science vol 7700 2012 (Neural Networks Tricks of the Trade). Springer Berlin Heidelberg, Berlin, Germany,

21. Kim P (2017) MatLab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence. Apress,

22. Hoang N-D, Tien Bui D GIS-Based Landslide Spatial Modeling Using Batch-Training Back-propagation Artificial Neural Network: A Study of Model Parameters. In, Cham, 2018. Advances and Applications in Geospatial Technology and Earth Resources. Springer International Publishing, pp 239-254

23. Tien Bui D, Le HV, Hoang N-D (2018) GIS-based spatial prediction of tropical forest fire danger using a new hybrid machine learning method. Ecological Informatics 48:104-116. doi:10.1016/j.ecoinf.2018.08.008

24. Liao K-W, Fan J-C, Huang C-L (2011) An artificial neural network for groutability prediction of permeation grouting with microfine cement grouts. Computers and Geotechnics 38 (8):978-986. doi:https://doi.org/10.1016/j.compgeo.2011.07.008