Bankruptcy prediction based on financial ratios using Jordan Recurrent Neural Networks: a case study in Polish companies

Lingga Hardinata¹, Budi Warsito¹, Suparti¹

¹Departement of Statistics, Faculty of Science and Mathematics, Diponegoro University, Prof. Soedharto, SH. Street, Semarang 50275, Indonesia

Email : linggahardinata@gmail.com

Abstract. Complexity of bankruptcy causes the accurate models of bankruptcy prediction difficult to be achieved. Various prediction models have been developed to improve the accuracy of bankruptcy predictions. Machine learning has been widely used to predict because of its adaptive capabilities. Artificial Neural Networks (ANN) is one of machine learning which proved able to complete inference tasks such as prediction and classification especially in data mining. In this paper, we propose the implementation of Jordan Recurrent Neural Networks (JRNN) to classify and predict corporate bankruptcy based on financial ratios. Feedback interconnection in JRNN enable to make the network keep important information well allowing the network to work more effectively. The result analysis showed that JRNN works very well in bankruptcy prediction with average success rate of 81.3785%.

1. Introduction

Bankruptcy prediction has been an important and widely studied topic in accounting and finance because it’s significant impact on management, employees, stockholders, and nation. Accuracy is one of crucial performance due to its significant economic impact, numerous statistical techniques have been used for improving the performance of bankruptcy prediction models, such as univariate analysis, discriminant analysis, logistic models and probit [1]. Studies of bankruptcy prediction continuously developed by academician and companies by using various models. The artificial neural network was one of the model conducted.

Basically, there are two approaches to predict the companies bankruptcy: univariate analysis and multivariate analysis. Univariate analysis used to predict financial distress which is the distribution of financial variables for companies that experiencing financial distress are different from companies that don’t have financial distress. Deficiency of this model is contradiction between the predicted variables. To solve this problem, multivariate models was developed. The independent variables in this model are the financial ratios that expected to affect bankruptcy, while the dependent variable is the prediction results. But till now, few theoretical discussion only that leads to bankruptcy research, e.g. in the selection of variables that are considered relevant. With at least the theory, bankrupt prediction is more directed to the search for variables that are considered relevant to the trial and error methods [2].

Artificial Neural Networks (ANNs) is considered more appropriate for prediction in data mining, because ANNs has the ability to extract important information from large data sets. For more than sixty years, since ANNs was found, ANNs has hundreds of network models that can be used to predict
the company's financial condition, pattern recognition, economic management, control and decision-making systems, health, agriculture, and many others [3].

In 1986s, Michael Jordan first introduced a recurrent network with feedbacks from output units. That is, the output units are connected to input units but with time delay, so that the network outputs at time \( t - 1 \) are also the input information at time \( t \). Feedback Interconnections within JRNNs are able to make the network keep the information and allow the networks to perform inference tasks such as prediction and classification [4]. This paper presents a comprehensive evaluation result of Jordan Recurrent Neural Networks (JRNNs) to classify Polish Companies into bankrupt or non-bankrupt categories. The model obtained is then used to predict corporate bankruptcy in the next period. The author hopes that JRNNs can be considered as an effective method of classifying companies in the category of bankrupt or not.

2. Methodology
In this section, Basic Neural Networks, Jordan Recurrent Neural Network (JRNN), Activation functions, and Rule of Thumb methods will be introduced. Each topics will be presented and discussed in the following subsection.

2.1. Artificial neural networks
Neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest; the network is usually implemented by using electronic components or is simulated in software on a digital computer [5]. ANN is an information processing system that has characteristics similar to neural networks in biological organism. The principle of ANN is developed from the characteristics and workings of the human brain, which in processing human brain information consists of a number of neurons that perform simple tasks. Due to the interrelationships between neurons, the brain can perform quite complex processing functions. Information processing can only be done after the previous learning process. The processing of information in humans is adaptive, which means that relationships between neurons occur dynamically, the strength of relationships between neurons may change over time, and always have the ability to learn new information [6].

The networks architecture consists of input layers unit with the number of neurons \( p \), hidden layers with \( n \) units and output layers unit consists a neuron can be written as the following model:

\[
y = \psi_0 \left( w_0 + \sum_{n} w_n \psi_n \left( w_n + \sum_{i} w_{ni} x_i \right) \right)
\]

where \( \psi_n, \psi_0 \) is activation function for hidden layers and output layers.
2.2. Jordan Recurrent Neural Networks

In 1986, Michael Jordan first introduced a recurrent network with feedbacks from output units. That is, the output units are connected to inputs but with time delay, so outputs at time \( t - 1 \) become input information at time \( t \). The outputs of the Jordan Recurrent Neural Network (JRNN) is defined as:

\[
y = \psi_0 \left( w_{j,0} + \sum_{h=1}^{n} w_{j,h} \psi_h \left( w_{h,0} + \sum_{r=1}^{r} w_{h,r} y_r + \hat{y}_{j-1} \delta_h \right) \right), \quad j = 1, \ldots, n
\]

where \( \hat{y}_{j-1} \) is output at \( t - 1 \) and \( \delta_h \) is the vector of the connection weight between the \( h \)-th hidden units and the input units that receive lagged output \( y_{t-1} \).

![Network structure of Jordan Recurrent Neural Networks](image)

**Figure 3.** Network structure of Jordan Recurrent Neural Networks

*Activation function*

In this paper, the activation function used to transfer the sum of input signals in the hidden layer leading to the output layer is a logistic sigmoid function. While the activation function used for the output signal is linear function. Both of these function graphs are shown in Fig. 4 and Fig. 5.

![Logistic Sigmoid Function](image)

**Figure 4.** Logistic Sigmoid Function

![Linear Function](image)

**Figure 5.** Linear Function
3. Data selection and model development

The data used in this study is obtained from sites: https://archive.ics.uci.edu/ml/machine-learning-databases/00365/ that provided by University of California at Irvine (UCI). The data set consists 1000 Polish companies. 19.4% companies went bankrupt during 2000-2012. Initially 15 financial ratios categorized as liquidity, leverage, activity, profitability, growth, and valuation. Data is divided into two groups: data that served as input for the learning process (called training data) and data that served as testing (testing data). Training data consists of 700 data with 15 variable financial ratios of the company then transformed into matrix with size 15x700 and 300 data used as data testing as matrix size 15x300.

Training result analysis
Training networks using similar architectures produces different outputs by changing the number of neurons in the hidden layers. In this paper, number of hidden neurons determined by rule of thumb methods.

Determine the number of hidden units using rule of thumb methods
Deciding the number of neurons on the hidden layer is important in constructing neural networks architecture. The hidden layer does not directly interact with the external environment of the network, but it has significant effect on the result. There are some various approaches to find out number of hidden nodes in hidden layer, such as Trial and Error Methods, Rule of thumb Methods, Simple Methods, Two Phase Methods and Sequential Orthogonal [14]. In this paper, the number of neurons on the hidden layer determined by Rule of Thumb Methods (Baum-Haussler rule) with the following formula:

\[
N_h = \frac{N_s}{(\alpha (N_i + N_o))}
\]  

(3)

where \(\alpha\) is the best multiplier factor (2-10), \(N_s\) is total sample in training data, \(N_i, N_o\) is number of input and output.
Following table presents the average of classification accuracy over 10 runs for each numbers of neurons on hidden layer:

| Number of neurons on hidden layers | Average of MSE for training data | Average of classification accuracy (%) |
|-----------------------------------|----------------------------------|----------------------------------------|
| 4                                 | 0.91357                          | 83.71438                               |
| 5                                 | 0.75100                          | 84.92856                               |
| 6                                 | 0.85865                          | 84.28571                               |
| 7                                 | 0.78063                          | 84.59999                               |
| 9                                 | 0.84438                          | 84.17142                               |
| 11                                | 0.83268                          | 83.74285                               |
| 15                                | 0.76309                          | 84.78571                               |
| 22                                | 0.75575                          | 84.31428                               |

![Figure 7](image_url)

**Figure 7.** Average classification and MSE over 10 runs

The optimal number of hidden layer is 5 neurons with the average of MSE value is 0.751 and the average of classification accuracy is 84.9286%.

The classification accuracy for training and testing data with 5 neurons in the hidden layer are presented in the Fig 8.

![Figure 8](image_url)

**Figure 8.** Average classification accuracy with 5 neurons of hidden layer over 50 runs
4. Conclusion
This paper has achieved important function, it extends the body of knowledge about Jordan Recurrent Neural Networks and their applications. The result showed the best performance is when the number of neurons in the hidden layer is 5 with average classification accuracy is 81.3785%. Our results suggest that the JRNNs model may be successful in improving forecasts of bankruptcy traditional forecasting models.

References
[1] Kim, M.J. & Kang, D.K. 2010. J. Expert Systems with Applications 3373–3379
[2] Hanafi, M. & Halim, A. 2005. Analisis Laporan Keuangan. Yogyakarta : UPP AMP YKPN
[3] Santosa, B. 2007. Data Mining : Tek. Pemanf. Data untuk Kep. Bisnis. Yogyakarta : Graha Ilmu
[4] Kuan, C.M. 2007. Artificial Neural Networks. Taipei : Academica Sinica
[5] Haykin, S. 1999. Neural Networks a Comprehensive Foundation. New Jersey : Prentice Hall
[6] Warsito, B. 2009. Kapita Selektta Statistika Neural Network. Semarang : BP UNDIP
[7] Bernhardsen. 2001. A Model of Company Bankruptcy Prediction. Norway : Norges Bank
[8] Carcano, E. et al. 2008. Jordan Recurrent Neural Network Versus IHARCES in Modelling Daily Streamflows. J. of Hydrology 291-307
[9] Handayani, L. & Fitriandini. 2015. Prediksi Kebangkrutan Perusahaan Menggunakan Support Vector Machine (SVM). Int. J. of Electrical and Computer Engineering (IJECE)
[10] Harahap, S. 2010. Analisis Kritis atas Laporan Keuangan. Jakarta : Rajawali Pers
[11] Kim, M.J. & Kang, D.K. 2010. Ensemble with neural networks for bankruptcy prediction. Expert Systems with Applications 3373–3379
[12] Kusumadewi, S. 2004. Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB dan Excel Link. Yogyakarta : Graha Ilmu
[13] Mostafa, YSA., & Atiya, AF., 1996. Introduction to Financial Forecasting. Applied Intelligence, No.6, Hal.205-213
[14] Panchal F.S. & Panchal, M. 2014. Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network. Int. J. of Computer Science and Mobile Computing 455-464
[15] Senbet, T. & Wang, T. 2012. Corporate Financial Distress and Bankruptcy : A Survey
[16] Siang, J.J. 2005. Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab. Yogyakarta: Penerbit Andi
[17] Wang, H. & Song, G. 2014. Innovetive NARX Recurrent Neural Network Model for Ultra-thin Shape Memory Alloy Wire. Neurocomputing 289-295