A neural network application for the estimation of the probability of leaving a working place

A Ene¹ and C Stirbu¹,*

¹University of Pitesti, Electronics, Computers and Electrical Engineering Department, Târgul din Vale Street No.1, Romania

Email: cosmin.stirbu@upit.ro

Abstract. This article is about a topical theme, namely that of leaving a workplace, by an employee. The paper proposes that on the basis of input data, to make an estimate of the probability of leaving a job by a particular employee. In order to make such a prediction, the authors of the paper propose the use of artificial intelligence elements, namely artificial neural networks. Artificial neural networks have the ability to generalize, based on a history. The probability of abandonment of a job, is determined empirically, because there is not an exact mathematical formula. When we do not know the equation after which a particular process takes place, to make an estimation, neural networks are excellent tools.

1. Introduction

The costs involved for replacing an employee from a working place, depends highly on the nature of his job. In [1] is shown that this cost is around 50% to 70% of the employee’s annual salary, on average. As shown in [2] there are also hidden costs such as productivity loss, workplace safety issues, and morale damage. The conclusion is that employee turnover is costly.

In this paper we developed a neural network application that is able to make an estimation on the probability of leaving a working place by an employee.

Neural networks are excellent tools that are able to identify the regularities from a data set, and based on these regularities they can make predictions. In [3] and [4] we used feed forward neural network for making such predictions in some industrial applications.

The application presented in this paper is useful in identifying the employees that have a high risk of leaving their current job, and so this application could be very useful for the Human Resource department of a company, in order to take more informed decisions.

The study of employee turnover using neural networks, was also addressed in [1] and [5]. In [5] the authors used a genetic algorithm to train the NN in order to predict the turnover rate for a small mid-west manufacturing company.

In [1] the author uses deep learning algorithm in order to train a neural network to predict employee turnover risk for a one-year period of time, in a Mexican company.

He used the following eight features that had, for his study, the highest correlation with employee turnover:
- age
- distance from home
- overtime
- education
- marital status
- number of companies worked at
- total working years
- monthly income

For the training of the neural network the authors used a big set of training patterns (1470), but, the problem is that it is not a real dataset, is a fictional dataset created by IBM scientists. The author recognizes in [1], that would be great to have a real data dataset, coming from a company.

In our paper we used a real dataset, coming from a Romanian financial institution. The advantage is that the dataset is real, not fictional, but, the disadvantage is that we had only 20 training examples, which are sufficient only to draw the conclusion that the neural network is capable to identify a person with a high risk of leaving her job, in a specified interval of time.

Our paper is different from [1] also in the model of neural network that we used. In [1], having a big data set, the author used deep learning, but we used feed forward neural networks, with backpropagation algorithm.

We made also an exploratory data analysis to identify the factors that have an impact on the probability of leaving a working place in that financial institution. The most important factors we found were the following:
- monthly income
- the relation with the manager (stressful, good, etc.)
- the relation with the colleagues at work
- the employee load level
- the family situation
- time necessary to arrive at the job
- the monotony of the job
- total working years
- numbers of companies worked at

As can be noticed, some factors are similar with those used in [1], but some are not.

2. Feed forward neural networks

Feed forward neural networks form the most widely used class of neural networks. These networks are composed of interconnected artificial neurons organized on three or more layers (layers). First layer neurons are the only ones receiving external signals. External signals may be analog signals (continuous values in an input range) and/or digital signals (with binary values symbolized by "0" or "1").

In terms of inputs, we can classify the feed forward neural networks in three classes:
- binary networks (have only binary inputs)
- analog networks (only analog inputs)
- mixed input networks (also have binary inputs and analog inputs)

Intermediate layers neurons are called hidden layers, and the last neural layer is called output layer.

The neurons in a layer are totally connected to the neurons in the next layer. There are no connections between the neurons at the same level.

In figure 1, a 3:3:1 feed forward neural network is presented (three input neurons, one hidden layer with three neurons, one neuron in the output layer).

The most important phase in designing and building an artificial neural network of any type is learning (training). The most known and most applied learning algorithm in feed forward neural networks is the backpropagation algorithm. Its discovery in the 1980s led to a massive interest the field of artificial intelligence (neural networks).

In the learning phase, the neural network is trained to learn a number of patterns (input/output pairs) with a specified error. The number of patterns required depends on the type and complexity of the problem. Current pattern inputs impose certain values for the output of the neural network (values calculated in the forward step of the training). These calculated outputs are compared to the expected output vector (given by the current pattern outputs), and the difference actually representing the current
error, propagates back into the network and in this phase (the backward step of the training) all network weights are adjusted according to delta rule (some mathematical equations) that minimize the overall learning error of the entire set of patterns.

![Feed forward neural network (3-3-1)](image)

**Figure 1.** A feed forward neural network (3-3-1)

When the network achieved the desired learning error, we say that the neural network has converged. Convergence speed is an important feature of the neural network used in a specific application. It depends on a number of factors such as:
- network architecture (number of hidden layers and number of neurons in each layer)
- the number of learning patterns
- the global error imposed
- parameters chosen for the backpropagation algorithm
- the pattern presentation order

The patterns have to be well defined. They have not to contradict one another. If there are contradictions in the set of patterns, the network will never converge.

As a matter of fact, the ability of the trained network to recognize new patterns, depends mostly by the quality of the patterns set.

A training epoch is determined by the presentation of the whole set of patterns to the network with the appropriate weight adjustment (for each pattern there is a forward propagate and a back propagate stage). Thus, the convergence speed of a neural network is measured in the number of epochs required for the entire set of training examples to be learned with the desired error.

3. Implementation

Firstly, we had to pre-process the dataset that we had from the financial institution about the employees that left their jobs in a determined period of time. Because feed forward neural networks train faster when we have digital input values rather than analogue, we converted the data that were analogue (for example monthly income) in digital values, in the following manner:
- monthly income: we used four classes of income: low, moderate, high and very high. The figures are compared with the medium national monthly income. In this way, these four classes can be binary coded
using two bits: 00 – for low income, 01 – for moderate income, 10 – for high income and 11 for very high.
- the relation with the manager: we used also four classes (stressful, moderate, good and very good). Same binary coding: 00, 01, 10 and 11.
- the relation with the colleagues at work: four classes, too.
- the employee load level: four classes (easy, moderate, high, very high)
- the family situation: two possibilities
- time necessary to arrive at the job: two possibilities
- the monotony of the job: two classes
- total working years: four classes
- numbers of companies worked at: four classes

The feed forward neural network was simulated using Java programming language. We used the following neural network architecture:
- input neurons number: 15
- one hidden layer with 10 neurons
- output neurons number: 1. A maximum value of 1 at the output of the network means in this application a high probability for an employee to leave in the specified interval of time (high risk of leaving). The value 0 at the output means a low probability for an employee to leave.

The architecture of the feed forward neural network is: 15 - 10 - 1

The pre-processed data were saved in a text file. Here is a sample from this text file:

1 0 1 0 1 0 0 1 1 0 1 0 1 0 0 0

Then we split this text file in two parts: 90% from the data we used for the training of the neural network, and 10% for the testing.

4. Results
We choose the global training error 1% (the error with which the neural network will learn all the training patterns). A lower training error (such as 0.001) will determine the network to learn the training patterns with a better accuracy, but also the speed of convergence will be lower, and, more important, the ability of the network to generalize will be diminished. The network converged fast, in about 100 epochs, because we had not many examples in the training set. Due to the random nature of initial weights of the network, the speed of convergence (the number of training epochs) is variable, so, if we repeat the training, we will obtain typically another number of epochs.

Here is a sample example from the running of the simulation program:

Epoch 92: total error = 0.010262057526630546
Epoch 93: total error = 0.010094954183130687
Epoch 94: total error = 0.009932785963805

The network converted successfully!
The weights had been saved successfully!

In order to test the network’s ability to estimate the probability for a worker to leave his current working place, we tested it with two different sets of input data. These input data were new for the network. They were not used when we trained the neural network.
Here are the results obtained when we tested in this way the network:

test 1
Ideal result: 1.0
Neural network calculated value: 0.9980087644530876

test 2
Ideal result: 0.0
Neural network calculated value: 0.00408351151171822

As seen from these running logs, the network answered correctly, in both cases, giving a result very close to the ideal result.

5. Conclusions
In this paper, the authors propose a new method of estimating the probability of an employee of leaving his working place in a determined period of time, using feed forward neural networks.

The necessary data for training and for testing the neural network were obtained from a HR department of private financial company.

The neural network was simulated using a Java software application. The correctness of the answers offered by the neural network depends on the on the correctness of training data.

The results obtained from the neural network are useful to HR department companies and industrial enterprises.

Selective References
[1] Torres M 2018 Employee turnover prediction with deep learning, https://dzone.com/articles/employee-turnover-prediction-with-deep-learning / accessed on 19.12.2018
[2] O’Connell M and Chuan Kung M 2007 Employee Turnover & Retention: Understanding the True Costs and Reducing them through Improved Selection Processes, Industrial Management. 49 14-19.
[3] Ene A and Stirbu C 2016 A Java application for the failure rate prediction using feed forward neural networks, ECAI2016, vol 8, 159-162, WOS:000402541200133
[4] Ene A and Stirbu C 2016 Natural gas consumption prediction with feed forward neural networks, University of Pitesti Scientific Bulletin (Series: Electronics and Computers Science, Pitesti, 2016), vol 16, issue 2, 8-11
[5] Sexton R S, McMurtrey S, Michalopoulos J O and Smith A M 2005 Employee turnover: a neural network solution, Comput. Oper. Res. 32 2635–2651