Proc. of The Fourth Intl. Conf. On Advances in Economics, Management and Social Study - EMS 2015

ISBN: 978-1-63248-071-2 doi: 10.15224/ 978-1-63248-071-2-25

Neural Networks For Credit Risk Management:

A case study in the car financing industry

[Martin Müller, Anna Matuszyk, Stefan Lessmann, Hsin-Vonn Seow]

***Abstract*— The paper aims at examining the degree to which artificial neural networks are a suitable approach to aid risk management in the car financing industry. More specifically, we empirically compare a classic feedforward neural network to a recently proposed extreme learning machine. To that end, we employ a real-world credit data set from a leading car financing company in Poland is used to assess each classifier’s accuracy. To systematically study the suitability of the two methods, our study comprises multiple experimental factors including the type of the neural network, whether or not it is embedded into an ensemble learning framework, and strategies to mitigate class imbalance.**

***Keywords*— credit risk, decision support, data analytics**

I. Introduction

Predictive analytics for credit scoring has received much attention in academic literature. Dozens of classifiers facilitate predicting the credit risk of a customer. The question is economically significant: With billions of dollars lend to individuals and businesses each year, an (even small) improvement of predictions can entail a huge financial premium to lenders. Many studies have examined which algorithms show the best performance in terms of predictive accuracy [1]. However, for practitioners, other features play an important role when assessing predicative models, such as ease of use, comprehensibility and computational resource consumption. This study aims to empirically compare and evaluate two different prediction models, namely artificial neural networks and a (more recent) extension called extreme learning machines, in the context of credit scoring data.

In the subsequent chapter, a brief introduction into artificial neural networks and extreme learning machines is given. Thereafter, the experimental design as well as the employed data is explained. Finally, the results are presented, followed by a brief summary of limitations and potential further research on this topic.

Martin Müller, Stefan Lessmann
Humboldt-University of Berlin
Germany

Anna Matuszyk
Warsaw School of Economics
Poland

Hsin-Vonn Seow
The University of Nottingham, Malaysia Campus
Malaysia

II. Neural Networks for Classification and Prediction

A. *Artificial Neural Networks (NN)*

The early days of neural networks date back until the 1940s. In 1958, Frank Rosenblatt presented a neural network with one neuron in the hidden layer. In the 1970s, the backpropagation algorithm was developed, which enabled the training of networks with more than one hidden neuron. Neural Networks consist of several layers: one input, one output and a number of hidden layers determined by the user. Each layer again consists of several elements, called neurons. The number of inputs in the first layer is determined by the number of predictors available. The number of output neurons is also determined by the prediction task, but is one in the case of a dual class classification problem. The number of hidden neurons is defined before the modelling process by the practitioner. They represent an activation function, which can have any nonlinear form (e.g. sigmoidal). These nodes are connected to the previous layer via synapses or, in mathematical terms, the parameters that determine the input of the activation functions. Parameters that cannot be calculated analytically, just like the beta parameters in linear regression, and therefore need to be set by the user are called meta-parameter. For neural networks, there exists a plurality of meta-parameters such as the number of hidden layers and neurons, the type of activation function, the size of the learning rate etc. In order to find optimal meta-parameters given the available data, the common approach is to empirically test several candidate values. Besides, neural networks have H*(P+1)+H+1 (regular) parameters or weights to be estimated, where H is the number of hidden neurons and P the number of predictor variables. Since this number mainly depends on the number of predictors, complexity increases especially for data sets with high dimensionality [2]. The efficient estimation of parameters is therefore important. The backpropagation algorithm introduced by Rumelhart in 1986, iteratively uses derivatives to assign the error in the resulting output prediction to previous parameters, and then re-estimates the network considering the contribution of each parameter to the overall error. The potentially large number of coefficients can make artificial neural networks very complex and thus prone to overfit on the training data. Several methods to address this issue exist, e.g. the early stopping approach or penalization methods like weight decay.

B. *Extreme Learning Machines (ELM)*

Although algorithms such as backpropagation are generally considered as very efficient, slow learning rates and high computational resource consumption remain weaknesses of standards neural network approaches. Here, a

Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-071-2 doi: 10.15224/ 978-1-63248-071-2-25

recent innovation from the last decade seems promising for mitigating these downsides. The so called extreme learning machines offer advantages such as a fast learning rate, easier implementation and less human intervention according to researchers in the field [3]. The main idea is basically that parameters at hidden neurons can be chosen randomly, leaving a linear problem to solve in order to find output weights. However, this might lead to reduced predictive accuracy.

Several studies have been conducted, which come to mixed results. Some authors present evidence that ELMs are generally superior when compared to NN with backpropagation, based on the root means squared error (RMSE). It shows better generalization performance as well as learning speed [4]. Others measure accuracy based on multiple performance metrics and find that ELMs tend to predict with lower accuracy in most cases throughout the examined data sets [5]. This underlines the need for further empirical tests on this issue, in order to identify the appropriate classifier for specific areas, such as credit scoring in the car loan industry.

III. Experimental Design

In order to evaluate the performance of the two examined classifiers, a data set from the car loan industry is used to train, evaluate and finally compare the models. It contains real-world observations and comes with the usual noise, such as missing values. The whole set encompasses contains 32,381 samples and 81 predictors, which describe the borrower’s demographic characteristics as well as specifications of credit agreements etc. Due to the nature of the underlying problem, credit scoring, the target variables are binary indicating whether someone defaulted on a given credit or not.

In the first step, the data is manipulated so that it fits the needs of the models developed. Neural networks, for example, can only handle numerical predictors, i.e. categorical variables have to be recoded. Additionally, predictors which have no or a very limited informational content are removed, such as those having zero variance or where the majority of values is missing. Avoiding the use of highly correlated variables and/or highly skewed distributions can also improve model performance. Due to their parametric nature, especially neural networks profit from the removal of uninformative predictors, since an increased number of dimensions require the model to compute an exponentially increased number of parameters. This introduces additional complexity and makes the model prone to overfitting [2]. Removing predictors can also be practically beneficial as it reduces computational time and interpretability.

For all tests, we use the ROC and the resulting area under the ROC curve (AUC) as performance metric. Although the employment of multiple metrics is considered to be more appropriate for empirical evaluations, only one is used in order to limit the scope of this work. As the name indicates, the AUC measures the area under the ROC curve. In case the classifier is able to perfectly separate cases in a given data set, the area under the curve would be exactly one. A completely useless model, such as randomly assigning classes to test cases, would result in a ROC curve stretching from the bottom left to top right in almost a straight line, with an AUC of roughly 0,5. Since the ROC curve is a function of sensitivity and specificity, it is relatively insensitive to class imbalances. One of the main disadvantages of the AUC measure can be the reduced information content of it. If two compared curves cross, for example, none is superior, and it depends on the particular section of interest, which classifier is more suitable. This information is neglected by the AUC metric [2].

The data is separated into training and test sample, using a stratified sampling technique. In a first step, meta-parameter tuning using grid search is performed, in order to find the most appropriate set of meta-parameters for the available data. Here, a grid of candidate values is created and each combination of parameters empirically tested on the training data. If needed, granularity of the grid is increased to obtain the most powerful parameter combination. Additionally, using the functionalities the caret package provides, each set of candidate values was tested on five different bootstrap samples drawn from the training data set. Secondly, an ensembling technique called bagging is used to obtain a combination of results from multiple base models of the same classifier. Homogenous ensembling (all base models stem from the same classifier) can reduce bias and variance by combining the predictions of multiple models. As for the bagging algorithm, a new bootstrap sample is created for each base model in the ensemble. This way, diversity is introduced and variance lowered by manipulating the data rather than combining different classifiers. Hence, the offside is that bias assigned to the specific classifier is not reduced using the bagging approach. As the available data sets come in part with severe class imbalance, subagging, a technique employed especially in credit scoring frameworks, is used to mitigate the problem of severe class imbalances in the data sets. Subagging works similarly to bagging, apart from the data manipulation approach. Here, the data sample for the training of each base model is drawn by down-sampling the original data set. This means, for each case in the minority class, a corresponding sample from the majority class is drawn without replacement. The procedure ensures a certain predetermined balance among the classes [5]. The comparison of the results of bagging and subagging should also provide a picture on how neural networks and ELMs perform when class imbalance is present. However, since the car loan data is highly imbalanced, down-sampling results in a very small training set (roughly 3% of the original size). Therefore, a new, slightly altered version of subagging is deployed to further reduce the negative effects of an extremely underrepresented class in the target variable. In this version, the SMOTE algorithm samples a larger data set by creating new cases from existing ones of the minority class. This can be achieved by randomly selecting cases and averaging their attribute values.

IV. Results

After repeated grid search was performed, it became clear that better results were achieved with relatively large values for the number of hidden neurons and weight decay. Due to the large number of dependent variables, testing candidate values with more than 15 hidden neurons was not possible, as this resulted in more than 1000 weights to be calculated, which exceeded limit of the used nnet R

ISBN: 978-1-63248-071-2 doi: 10.15224/ 978-1-63248-071-2-25

package. The results of the meta-parameter tuning process are depicted in Fig. 1.

The same procedure was used selecting the optimal number of hidden neurons in the ELM and the type of the employed activation function. Here, the best AUC values were achieved with a combination of ten hidden neurons and a tangent sigmoid activation function, as shown in Fig. 2.

The optimal combination for the neural network achieved a very high AUC value of above 0,925 within the tuning process, i.e. measured using the training data. Even sub-optimal combinations of meta-parameters yielded values of more than 0,9. When applied on the hold-out data, the model with the best meta-parameter combination achieved an AUC of 0,936. This indicates that neural networks are suitable, or perform well on this kind of data. As for ELM, the values are significantly lower. The best meta-parameter combination yields an AUC of about 0,75 on the training data and a mere 0,635 on the test sample. However, the results from a single evaluation may be misleading. Thus, several base models are combined with the bagging approach and the number of combined models is increased in order to examine the effect of this dimension on performance. Furthermore, the severe class imbalance is taken into account with the use of different resampling methods.

As outlined in the previous section, the meta-parameters received from the grid-search process were held constant in the subsequent experimental steps. For the neural network models combined with the bagging approach, AUC values of around 0,94 were achieved, relatively independent from the number of base models. Fig. 3 provides an overview of the results. Although the bagging approach is recommended when using neural networks as base models to account for classifier unstableness, the results barely improve after applying it. Perhaps, since accuracy is already high, the additional improvement gained from further optimization is relatively marginal. For ELM, a different picture emerges, as shown in Fig. 3. AUC rises from 0,635 to more than 0,9 for a combination of 25 base models. The results are also more volatile. For NN it seemed performance was relatively independent from number of base models, but it varies significantly for ELM combinations with ten or less models, stabilizing only for 25 and 50 combinations. Actually, this is the expected effect for neural networks and related classifiers. Unstable results for single or few base models, and this effect being mitigated by a variance reducing procedure like bagging.

Additionally, the subagging approach is used to take into account the severe class imbalance (about 1,3% of cases belong to the minority class) present in the car loan data set. This means, the training data is subsampled such that it contains all cases from the minority class and the same number of cases from the majority class, drawn randomly without replacement. This is supposed to mitigate the negative effects of class imbalance. Since only a very small fraction of cases are labeled as bad in the case of the available car loan data, the down-sampling procedure creates rather small data sets (fewer than 1000 observations) that are used to train the base models. This might weakens the positive effect of more balanced classes when it comes to performance. Therefore, another approach is employed that resembles subagging apart from the resampling method. Instead of drawing cases from the majority class until both classes are of equal size, the so-called SMOTE algorithm (synthetic minority over-sampling technique) is used to restore class balance. The algorithm produces larger training data sets, as it creates new “synthetic” cases derived from existing ones, additional to the down-sampling of the majority class. Fig. 4 summarizes the results of both approaches. For NN, subagging yields results similar to the bagging procedure, with AUC values close to 0,94. Again, the number of base models combined does not seem to have a significant influence on accuracy. Aggregating models based on SMOTE subsamples results in only marginally different AUC values of close to 0,93. The corresponding ROC curves for the ensembles with the highest accuracy can be found in the appendix. All in all, all deployed procedures only have a minor influence on the very high accuracy achieved by a NN classifier on the data set at hand. A different picture emerges when the accuracy of ELM is assessed with respect to the different approaches. First of all, the number of base models combined does seem to have a significant influence, since there is a significant upward trend visible for ELM ensembles in Fig. 3 and also a marginal one in Fig. 4. There is only mixed evidence on the effect that subagging and resampling with SMOTE have on the accuracy of ELM. For smaller subsamples of five to ten base models, accuracy is indeed higher. When 25 or more models are combined the effect vanishes. Potentially, the benefits of subsampling (or, on the other hand, the negative effects of class imbalance) become irrelevant the more base models enter the ensemble.

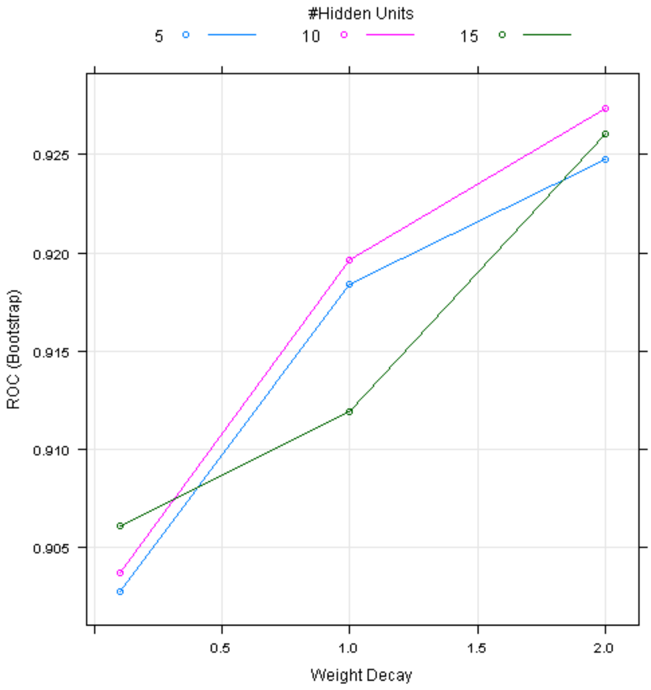

Figure 1. AUC values for different sets of NN meta-parameters.

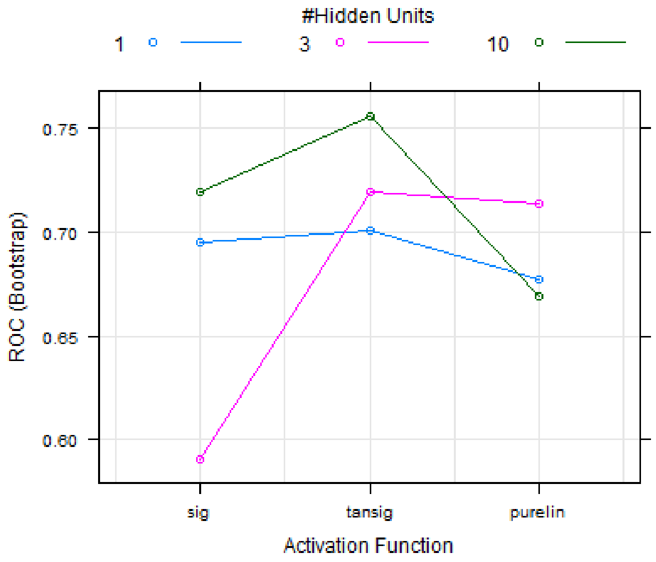

Figure 2. AUC values for different sets of ELM meta-parameters.

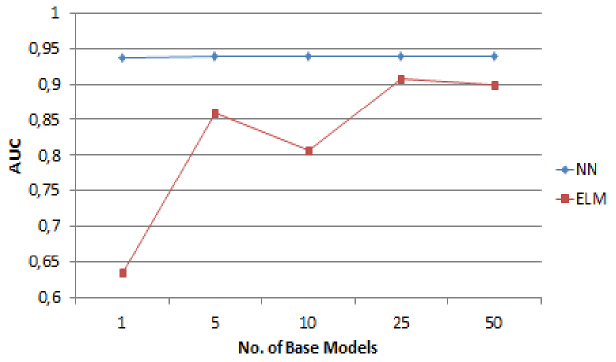

Figure 3. AUC values across different ensemble sizes for NN and ELM using bootstrap resampling for each base model.

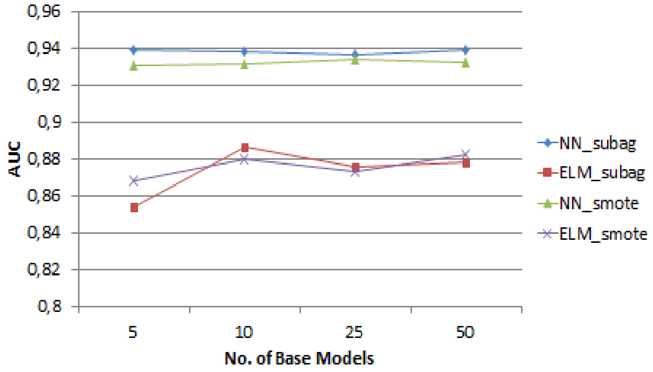

Figure 4. AUC values across different ensemble sizes for NN and ELM using *subagging* and subsampling with SMOTE.

V. Conclusions

The conducted comparison of neural network and extreme learning machines actually has a clear winner. The NN outperforms the ELM classifier in every setup. In some cases, however, when ensembling techniques are used and the number of combined base models is ten or higher, ELM could achieve AUC values close to those of NN. Under some scenarios, practitioners may decide to trade some accuracy for enhanced model building time. Although this seems implausible for credit scoring, applications that require close to real-time results may benefit from the application of ELM. Additionally, the analysis has shown that, given the data at hand, the performance of NN is relatively insensitive to both, ensembling and resampling procedures. When time is an issue, it can be advantageous to predict with NN based on just subsamples of the available data and combined to an ensemble including relatively few base models, instead of turning to the less accurate ELM.

This analysis, however, also comes with limitations. It was carried out based on just on data set and results can differ for new data with even similar characteristics. Furthermore, the assessment was based purely on the AUC as the metric of interest. Many studies are criticized when they do not take into account several performance metrics. Still, this analysis provides a picture on how NN and ELM can differ in terms of accuracy and how different ensembling and resampling techniques affect results when a severe class imbalance is present. Further research may aims to compare the classifiers using different data and experimental setups, in order to examine the validity of the results of this study.

References

[1] Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, (doi:10.1016/j.ejor.2015.05.030).

[2] M. Kuhn, and K. Johnson, “Applied predictive modeling,” New York, NY: Springer, pp. 489-490, p. 270, 2013

[3] E. Cambria, G.B. Huang, and L.L.C. Kasun, “Extreme learning machines [trends & controversies]”, IEEE Intelligent Systems, vol. 28, p. 30, December 2005

[4] G.B. Huang, Q.Y. Zhu, and C.K. Siew, “Extreme learning machine: theory and applications” Neocomputing, vol. 70, pp.489-501. December 2005

[5] G. Paleologo, A. Elisseeff, and G. Antonini, “Subagging for credit scoring models”, European Journal of Operational Research, vol. 201, pp.490-499, March 2009