ITM Web of Conferences 72, 05006 (2025) https://doi.org/10.1051/itmconf/20257205006
HMMOCS-III 2024

Credit approval classification through a WASD neuronet

Spyridon D. Mourtas[1,2*]

[1]Department of Economics, Division of Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece
[2]Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia

Abstract. Because the banking business is growing, more people are seeking for bank loans, although banks can only lend to a certain number of applicants because they have a limited amount of assets to lend to. Hence, in order to save a lot of bank resources, the industry of banking is particularly concerned in developing ways to lower the risk element involved in selecting the safe applicant. These days, selecting the safe applicant requires a lot less work thanks to machine intelligence. In light of this, a new weights and structure determination (WASD) neuronet has been developed to address the two issues of credit approval mentioned above, as well as to manage its particular features. We improve the learning process of the WASD algorithm with a novel activation function for optimal adaptation to the credit approval model, motivated by the finding that WASD neuronets perform better than traditional back-propagation neuronets in terms of slow training speed and trapping in a local minima. An experimental study with an insurance company dataset demonstrates superior performance and adaptability to issues.

1 Introduction

Financial organizations such as banks have been making loans since the year 2000. Since credit risk arises mainly when borrowers are unwilling or unable to make payments, conducting a thorough background check on a customer before approving a loan is crucial to maintaining oneself in this line of work. Remember that there are a lot of non-performing loans in the economy since they reduce bank profits and deplete important resources, which makes it harder for banks to make new loans. Issues in the banking industry have the potential to quickly extend to other economic sectors, endangering jobs and economic expansion. Better methods for deciding whether or not to provide a loan must therefore be developed immediately.

Today, the amount of work required to do such jobs is significantly reduced by developing technologies such as natural language processing and machine learning [1]. Neural networks (NNs), or neuronets, that are mainly employed for classification and regression problems, have been efficiently used in healthcare, engineering, economics, social science research, and

* Corresponding author: spirmour@econ.uoa.gr

finance, among other fields. Measurements of solar systems and alloy behavior analysis are two common uses for them in engineering [2]. Likewise, NNs are widely utilized in medical diagnostics to identify several types of cancer, including lung and breast cancer [3]. On the other hand, NNs are mainly employed in the domains of economics and finance for time series forecasting, portfolio optimization, and macroeconomic factor prediction [4]. Additionally, NNs have been efficiently employed in social science research, usually for multiclass classification issues like occupational classification [5], teleworking assessment, and occupational mobility definition [6].

This work's main objective is to develop a model for loan acceptance prediction using innovative NNs augmented with cutting-edge methods. To do this, we will apply a feed-forward NN (FNN) capable of handling binary classification problems. The popular back-propagation method for training FNNs will be replaced with a weights and structure determination (WASD) training algorithm. The WASD strategy calculates the optimal set of weights directly utilizing the weights direct determination (WDD) algorithm, in contrast to the back-propagation technique, which iteratively alters the NN's topology. Ultimately, this keeps the system from being stuck in local minima, which lowers computing complexity [4]. In order to train a three-layer FNN for binary classification problems, we present a power Sigmoid Linear Unit (SiLU) activated WASD algorithm, called SWASD. The SWASD model outperforms some of the most advanced classification models in MATLAB's classification app in every way, according to experimental results using a dataset from insurance companies.

The following succinctly describes the main concepts of this work: Using a publicly available dataset from insurance companies, the performance of a novel three-layer FNN based on the power SiLU activated WASD for binary classifications, called SWASD, is contrasted to some of the most advanced classifiers of MATLAB's classification app and the power Gaussian Error Linear Units WASD (GWASD) model from [5].

2 Methods: the SWASD neuronet model

A three-layer FNN model trained using the SWASD technique is covered in this section. The model has m inputs and n hidden layer neurons. Specifically, Layer 1 is the input layer which receives and supplies the corresponding neuron in Layer 2 with equal weight 1 with the input values $A_1, A_2, \dots, A_m$. Up to n active neurons may be present in Layer 2, which houses hidden layer neurons. Lastly, there is only one active neuron in Layer 3, which is the output layer. The weights $w_i, i = 1,2, \dots, n-1$, in the neurons connecting Layer 2 and Layer 3 are obtained through the WDD process. Low hidden layer usage can be achieved by the NN model by the use of the SWASD algorithm.

2.1 The WDD procedure for binary classification

A key component of any WASD algorithm is the WDD process, which only takes real numbers as input data. Before being entered into the FNN, the data must additionally be normalized to a range of $[-0.5, -0.25]$. In this way, the FNN manages over-fitting. If necessary, we can use the linear transformation in [5] to do it. Here, important theoretical underpinnings and research for the creation of the SWASD model are thoroughly justified. Think of the target vector $D \in \mathbb{R}^r$ and the input $A = [A_1, A_2, \dots, A_m] \in \mathbb{R}^{r\times m}$, where r is the number of samples. Consider that $U(A_1, A_2, \dots, A_m) = D$ due to an underlying relationship U. To approximate U, the FNN employs the following element-wise power SiLU activation function:

$$G_h(A) = \frac{A^{\odot h} . e^{\odot A^{\odot h}}}{1+e^{\odot A^{\odot h}}} \qquad (1)$$

where h is both the hidden layer neurons number with $h = 0,1,\dots,n-1$, h is the power value, and $(\,)^{\odot}$ is the element-wise exponential. Note that (1) is a linear scheme of n activation functions G_h. For each activation G_h, let k_h imply the image of A under G_h. Hence, $k_h \in \mathbb{R}^{r\times m}$ for $h = 0,1,\dots,n-1$. Also, let $W = [w_0, w_1, \dots, w_{n-1}] \in \mathbb{R}^{mn}$ be a weights vector. A linear combination of all n images may be represented as $\sum_{h=0}^{n-1} k_h w_h = \breve{D}$, where $K = [k_0, k_1, \dots, k_{n-1}] \in \mathbb{R}^{r\times mn}$ and $\breve{D}$ is the FNN's output. In order to convert $\breve{D}$ to binary, an element-wise function $B(\cdot)$ is employed. Thus, the FNN's final output is

$$B(\breve{D}_i) = \begin{cases} 1 & , \breve{D}_i \geq -0.375 \\ 0 & , \breve{D}_i < -0.375 \end{cases}, \quad \text{for } i = 1,2,\dots,r. \tag{2}$$

When a credit is granted, it is represented by 1, and when it is not, it is 0. It is important to take note of the next Theorem and Proposition concerning the NN's convergence.

Theorem 1 Let Θ *be a nonnegative integer and* $U(\cdot)$ *be a target function. When* $U(\cdot)$ *has the* $(\Theta+1)$*-order continuous derivative inside the range* $[\gamma_1,\gamma_2]$*, it holds:*

$$U(\zeta) = B_\Theta(\zeta) + C_\Theta(\zeta), \quad \zeta \in [\gamma_1,\gamma_2], \tag{3}$$

where $C_K(\zeta)$ *is the error and* $B_K(\zeta)$ *is the* Θ*-order Taylor approximation (TA) of* $U(\zeta)$*.*

Let $U^{(\delta)}(\beta)$ be the δ-order derivative's value of $U(x)$ at point β. The approximate representation of $U(\zeta)$ is:

$$U(\zeta) \approx B_\Theta(\zeta) = \sum_{\delta=0}^{\Theta} \frac{f^{(\delta)}(\beta)}{\delta!}(\zeta-\beta)^\delta, \quad \beta \in [\gamma_1,\gamma_2]. \tag{4}$$

Proposition 1 The approximation of multivariable functions can be done using Theorem 1. Suppose there are v variables. In an origin's neighborhood $(0,\dots,0)$, let $U(\zeta_1,\zeta_2,\dots,\zeta_v)$ be the target multivariable function with $(\Theta+1)$-order continuous partial derivatives. It follows the Θ-order TA $B_\Theta(\zeta_1,\zeta_2,\dots,\zeta_v)$ about the origin:

$$B_\Theta(\zeta_1,\zeta_2,\dots,\zeta_v) = \sum_{h=0}^{\Theta} \sum_{\delta_1+\cdots+\delta_v=h} \frac{\zeta_1\cdots\zeta_v}{\delta_1\cdots\delta_v}\left(\frac{\partial^{\delta_1+\cdots+\delta_v} U(0,\cdots,0)}{\partial\zeta_1^{\delta_1}\cdots\partial\zeta_v^{\delta_v}}\right), \tag{5}$$

where $\delta_1,\delta_2,\dots,\delta_v$ *are nonnegative integers.*

Following that, instead of applying the iterative weight training scheme seen in ordinary FNNs, the Θ-order TA FNN's weights are immediately calculated using the WDD strategy explained next [7]:

$$W = K^\dagger D, \tag{6}$$

where $(\,)^\dagger$ implies pseudoinversion.

2.2 The SWASD algorithm

Since (6) handles the calculation of the optimal weights, the next step is to figure out the best size for the NN. Cross validation is used in conjunction with the SWASD algorithm to secure that the FNN generalizes beyond the training set. The SWASD algorithm may be explained in the 3 steps listed below.

1) Assume the normalized input matrix A. The training and test sets are identified using a random 75%-25% split. From the training set, two sets of samples are again extracted: one for validation and one for model fitting. The user-specified parameter $c \in (0,1] \subseteq R$ deputizes the samples' percentage to be allotted for model fitting. We identify the first $S_1 = c\cdot S$ samples as being utilized for model fitting and the last $S_2 = S - S_1$ samples as undergoing validation, as the training set consists of $S = 0{,}75\cdot r$ samples.

2) The SWASD algorithm starts with a power $h = 0$ and an empty matrix K. Then, when h increases, it adds the matching columns k_{h+1} to build K. As demonstrated in (6), the ideal weights are determined for every iteration in respect to the training set. On the validation set, the FNN's performance is assessed using the mean-absolute-error (MAE $= \frac{1}{S_2}\sum_{l=1}^{S_2} |\breve{D}_l -$

$D_l|$). New columns are only kept in place if they have enhanced performance or reduced the MAE. This method is continued until h reaches a predetermined threshold, that we set at $h_{\max} = 59$.

3) Upon reaching $h_{\max}$, the training and validation sets are merged, and the terminal weights are calculated with respect to all samples (i.e., training and validation samples).
In conclusion, the SWASD algorithm guarantees optimal weights and facilitates computation in subsequent runs by reducing the FNN's structure to the smallest dimension possible. The SWASD training method is a very successful process when combined with cross validation, as was previously described.

3 Results and discussion

The terminal step is to apply the FNN's final structure to the test set. Notably, the 13-variable credit approval dataset is available at https://www.kaggle.com/datasets/ninzaami/loan-predication?resource=download. This dataset, which includes customer information from an online loan application form, was used to evaluate the NNs' performance. Since the WDD technique takes only input data in the shape of real numbers, the dataset's non-numerical data is converted into real numbers using the data preparation methodology described in [8]. 471 numerical samples will be included in the dataset following data preprocessing. Additionally, the first 236 samples are used to create the training set, while the last 235 samples are used to create the testing set. The two subfigures in Fig. 1 display the test set classification results (right) and the training error path (left).

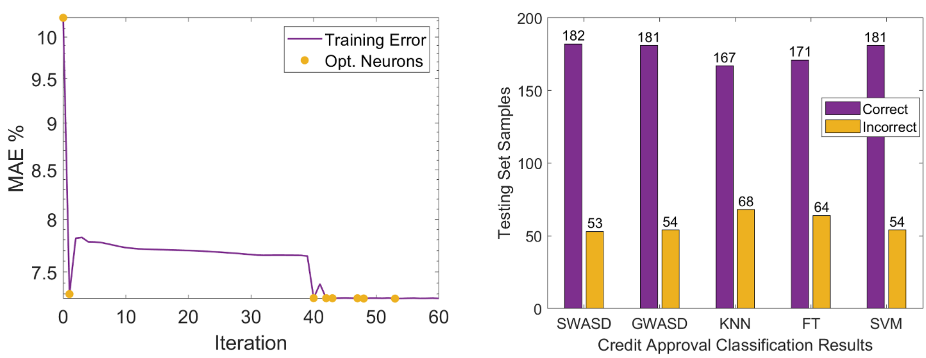

Fig. 1. Results on the test set (right) and path of the training error (left) for the SWASD.

To make pertinent inferences regarding the performance of the suggested SWASD model, a contrast to other high-performing classifiers is required. To do this, we chose the GWASD model in addition to three popular models from MATLAB's classification app: fine tree (FTree), k-nearest neighbors (KNN) and support vector machines (SVM). The performance metrics taken into consideration in our analysis are the MAE, precision, recal, accuracy, F-score, true positive (TP), false negative (FN), false positive (FP) and true negative (TN). For further information and a detailed examination of these measures, see [9].

Fig. 1 (left) shows that SWASD needs less than 60 iterations to find the optimal structure of the FNN, which includes 8 hidden layer neurons. On the testing set, Fig. 1 (right) presents that SWASD has the best ratio correct/incorrect outcomes and KNN has the worst. The aggregate test results for the credit approval dataset are presented in Tab. 1. The SWASD distinguishes itself from the other classifiers with the highest scores for TP, Precision, F-score and Accuracy, and the lowest MAE. In the TN, FN, and Recal scenarios, it is outperformed by Ftree, and in the FP scenario, by KNN. However, in contrast to the Ftree model, exceptional achievement on some metrics does not seem to be accompanied with unjustifiably subpar performance on the other criteria. For this reason, the SWASD proves to be a reliable classifier that does exceptionally well on all parameters instead of just one.

Table 1. Performance comparison between classifiers.

Statistics	SWASD	GWASD	KNN	Ftree	SVM
MAE	**0.2255**	0.2297	0.2893	0.2723	0.2297
FP	0.0243	0.0243	**0.1646**	0.1707	0.0243
TP	**0.9756**	**0.9756**	0.8353	0.8292	**0.9756**
TN	0.3098	0.2957	0.4225	**0.4929**	0.2957
FN	0.6901	0.7042	0.5774	**0.5070**	0.7042
Precision	**0.9756**	**0.9756**	0.8353	0.8292	**0.9756**
Recal	0.5856	0.5807	0.5912	**0.6205**	0.5807
F-score	**0.7319**	0.7281	0.6924	0.7098	0.7281
Accuracy	**0.7744**	0.7702	0.7106	0.7276	0.7702

To precisely ascertain whether it is feasible to make inferences regarding a FNN that is more suited for forecasting credit acceptance in the previously mentioned dataset, we incorporatea mid-p-value McNemar test [10] (i.e., a statistical element) into the previous investigation. Whether 2 classifiers forecast the proper class with similar accuracy (i.e., the null hypothesis) or not (i.e., the alternative hypothesis) is the goal of this test. We looked at every possible combination of the SWASD and one of the other 4 models using the mid-p-value McNemar test. The complete outcomes are presented in Tab. 2. At the 5% significant level, the null hypothesis is strongly rejected by the evidence, with the exception of the pair SWASD and KNN. The higher accuracy of SWASD is thus widely proven. The null hypothesis is not disproved in the pairs SWASD and SVM, SWASD and Ftree, and SWASD and GWASD. In other words, the accuracy of the SWASD, GWASD, Ftree, and SVM models is comparable. However, the data in Tab. 1 make it evident that the SWASD performs better overall than the GWASD, Ftree, and SVM. As a result, SWASD's superior performance is widely recognized.

Table 2. Outcome McNemar's test.

SWASD	**McNemar test**	
vs.	Null Hypothesis	p-value
GWASD	Not Rejected	0.5000
KNN	Rejected	0.0090
Ftree	Not Rejected	0.0730
SVM	Not Rejected	0.5000

4 Conclusion

In this study, a SWASD NN is introduced to categorize credit approval. Using a freely accessible credit approval dataset, the SWASD exhibited either greater or very equivalent performance abilities contrasted to other well-known and successful classifiers. We have been able to demonstrate that the predictive accuracy of the KNN model and the SWASD model differs statistically significantly. This validates the results' credibility, which is further supported by a McNemar test's findings. Nonetheless, the statistical test unequivocally demonstrates that the SWASD outperforms the GWASD, KNN, and SVM in terms of overall performance. As a result, we can confirm that the SWASD classifier is a reliable classifier capable of handling the credit approval classification problem.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

References

1. B. P. Lohani, M. Trivedi, R. J. Singh, V. Bibhu, S. Ranjan, P. K. Kushwaha, Machine learning based model for prediction of loan approval, in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM),* 465-470, IEEE, 27-29 April 2022, London, United Kingdom (2022)
2. N. Premalatha, A. V. Arasu, J. Appl. Res. Technol., **14**, 206-214 (2016)
3. R. J. S. Raj, S. J. Shobana, I. V. Pustokhina, D. A. Pustokhin, D. Gupta, K. Shankar, IEEE Access, **8**, 58006-58017 (2020)
4. Y. Zhang, Z. Xue, M. Xiao, Y. Ling, C. Ye, *Ten-quarter projection for Spanish central government debt via WASD neuronet,* in: *International Conference on Neural Information Processing, ICONIP*, November 14-18, Guangzhou, China (2017)
5. D. Lagios, S. D. Mourtas, P. Zervas, G. Tzimas, Mathematics, **11**, 629 (2023)
6. I. N. Generalao, The Philippine Review of Economics, **58**, 92-127 (2021)
7. Y. Zhang, X. Yu, L. Xiao, W. Li, Z. Fan, W. Zhang, Weights and structure determination of articial neuronets, in *Self-Organization: Theories and Methods* (Nova Science, New York, NY, USA, 2013)
8. T. E. Simos, V. N. Katsikis, S. D. Mourtas, Appl. Soft Comput. **127**, 109351 (2022)
9. A. Tharwat, Appl. Comput. Inform., **17**, 168-192 (2020)
10. M. W. Fagerland, S. Lydersen, P. Laake, BMC Med. Res. Methodol., **13**, 1-8 (2013)