Customer churn classification through a weights and structure determination neural network

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, Athens, 10559, Greece
[2]Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, Krasnoyarsk, 660041, Russia

Abstract. In today's corporate world, acquiring and keeping clients are the most important priorities. Every business's market is expanding quickly, which is increasing the number of subscribers. Because neglect could result in a drop in profitability from a major standpoint, it has become imperative for service providers to limit churn rates. These days, identifying which customers are most likely to leave a business requires a lot less work thanks to machine learning. Taking this into account, a novel weights and structure determination (WASD) neural network has been built to meet the aforementioned challenge of customer churn classification, as well as to handle its unique characteristics. Motivated by the observation that WASD neural networks outperform conventional back-propagation neural networks in terms of slow training speed and trapping in a local minimum, we enhance the WASD algorithm's learning process with a new activation function for best adapting to the customer churn model. Superior performance and flexibility to problems are demonstrated in an experimental investigation using a dataset from a telecommunications provider.

1 Introduction

Today's business environment is highly competitive, and each client is precious. It's critical to comprehend the client, which includes being able to comprehend their behavioral patterns. The pace at which a commercial customer leaves a commercial business and transfers their funds to another is known as customer churn. In particular, when it comes to market dynamics and competitiveness, the telecommunications sector is active and expanding quickly. It produces new goods and technology that give consumers more choices and opportunities to meet their needs and specifications. However, losing important clients to rivals is a significant issue that affects commercial businesses generally and telecommunications businesses specifically. As a result, there is an urgent need to develop better models for determining whether or not customers are likely to leave, in order to make the required decisions before departing consumers choose to do business with a rival.

These days, emerging technologies like machine learning and natural language processing greatly reduce the amount of work needed to do such tasks [1]. Primarily utilized

* Corresponding author: spirmour@econ.uoa.gr

for classification and regression issues, neural networks (NNs) have been successfully applied in a number of disciplines, including medicine, engineering, economics, social science research and finance. In engineering, they are widely used for alloy behavior analysis and solar systems measurements [2]. Additionally, NNs are frequently used in medical diagnostics to diagnose various cancers, such as breast and lung cancer [3]. In contrast, NNs are typically used in the fields of economics and finance for macroeconomic factor prediction [4], time series forecasting and portfolio optimization [5, 6]. Furthermore, NNs have been effectively used in social science research, typically for multiclass classification tasks like classifying occupations [7], assessing the possibility for teleworking in jobs and defining occupational mobility [8].

The primary goal of this work is to create a model for predicting customer churn utilizing novel NNs enhanced with state-of-the-art techniques. We will use a feed-forward NN that can handle binary classification tasks in order to accomplish this. A weights and structure determination (WASD) training algorithm will be employed in place of the well-known back-propagation approach for training feed-forward NNs. Unlike the back-propagation technique, which iteratively changes the network's structure, the WASD approach uses the weights direct determination (WDD) procedure to compute the optimal set of weights directly. In the end, this reduces computational complexity by preventing the system from becoming trapped in local minima [9]. In this way, we introduce a power Gaussian Error Linear Unit (GELU) activated WASD algorithm, termed PG-WASD, for binary classification tasks to train a 3-layer feed-forward NN. Results from experiments conducted on a dataset from a telecommunications provider demonstrate that the PG-WASD model performs in every manner better than some of the most sophisticated classification models of MATLAB's learner app.

The primary ideas of this work can be summed up as follows: a novel 3-layer feed-forward power GELU activated WASD NN for binary classifications, termed PG-WASD, is presented; the PG-WASD model's performance is compared to some of the most sophisticated classification models of MATLAB's learner app, as well as the Fresnel Cosine Integral WASD (FCI-WASD) model from [10], on a publicly accessible dataset from a telecommunications provider.

2 Methods: The PG-WASD neural network model

This section discusses a 3-layer feed-forward NN model trained employing the PG-WASD algorithm. The model consists of n hidden layer neurons and m input. In particular, Layer 1 is the input layer that receives and assigns the input values $A_1, A_2, \dots, A_m$ to the matching neuron in Layer 2 with equal weight 1. Layer 2 contains the hidden layer neurons and there can be up to n active neurons. Finally, Layer 3 is the output layer with one activated neuron. The WDD procedure yields the weights $w_i, i = 1,2, \dots, n-1$ in the neurons that link Layer 2 and Layer 3. By utilizing the PG-WASD algorithm, the NN model can attain low hidden layer employment.

2.1 A novel WDD process for binary classification tasks

The WDD process is a crucial part of any WASD algorithm and only accepts input data in the form of real numbers. Additionally, the data must also be normalized to a range of $[-0.5, -0.25]$ before being fed into the NN model. In this way, the NN can handle over-fitting. If required, we can accomplish that by using the linear transformation shown in [6]. Here, thorough justifications of significant theoretical foundations and research are offered for the development of the PG-WASD model.

Consider the input $A = [A_1, A_2, \dots, A_m] \in \mathbb{R}^{r\times m}$ with r denoting the number of the samples and the target vector $D \in \mathbb{R}^r$. Consider that there exists an underlying relationship U such that $U(A_1, A_2, \dots, A_m) = D$. The NN uses a linear combination of n activation functions G_h to approximate U, where

$$G_h(A) = \frac{A^{\odot h}}{2}\left(1 + \operatorname{erf}\left(\frac{A^{\odot h}}{\sqrt{2}}\right)\right) \tag{1}$$

is an element-wise power GELU activation function, $()^{\odot}$ signifies the element-wise exponential and h implies both the power value and the hidden layer neurons number with $h = 0,1, \dots, n-1$. For each activation G_h, let k_h denote the image of A under G_h. Thus, $k_h \in \mathbb{R}^{r\times m}$ for $h = 0,1, \dots, n-1$. Assuming a weights vector $W = [w_0, w_1, \cdots, w_{n-1}] \in \mathbb{R}^{mn}$, a linear combination of all n images can be formulated as $\sum_{h=0}^{n-1} k_h w_h = \breve{D}$, where $K = [k_0, k_1, \dots, k_{n-1}] \in \mathbb{R}^{r\times mn}$. Notice that $\breve{D}$, the NN's output, need to be converted to binary. In order to achieve this, an element-wise function B is used, resulting in $B(\breve{D})$ as the NN's final output with

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

where a customer's likelihood of churn is indicated by 1, and it is 0 otherwise. The following Proposition 1 and Theorem 1 should be noticed regarding the convergence of the NN.

Theorem 1 When a target function, $U(\cdot)$, has the $(\Theta + 1)$-order continuous derivative on the range $[\gamma_1, \gamma_2]$ and Θ is a nonnegative integer, the following is true:

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

where $C_K(\zeta)$ and $B_K(\zeta)$ signify the error term and Θ-order TA of $U(\zeta)$, respectively.

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

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

Proposition 1 Theorem 2.1 may be utilized for multivariable functions approximation. Consider $U(\zeta_1, \zeta_2, \dots, \zeta_v)$ be the target function with v variables and $(\Theta + 1)$-order continuous partial derivatives in an origin's neighborhood $(0, \dots, 0)$. Below is shown 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.

Then, using the WDD technique described below, the weights of the Θ-order TA NN are directly generated as opposed to utilizing the iterative weight training methods used in conventional NNs [11]:

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

where $()^{\dagger}$ signifies pseudoinversion.

2.2 The PG-WASD algorithm

The next task is to determine the ideal size for the NN, since (6) takes care of the computation of the optimal weights. To guarantee that the model generalizes outside of the training set, cross validation is utilized together with the PG-WASD algorithm. The three steps below can be used to describe the PG-WASD algorithm.

Step 1: A random $75\% - 25\%$ split is used to identify the training and test sets while taking into account the normalized input matrix A. Once more, two sets of samples are taken from the training set: one for model fitting and the other for validation. The samples' percentage to be assigned for model fitting is represented by the user-specified parameter $c \in$

$(0,1] \subseteq \mathbb{R}$. Given that $S = \frac{75}{100} r$ samples make up the training set, we designate the first $S_1 = cS$ samples as being used for model fitting, and the remaining $S_2 = S - S_1$ samples as going through validation.

Step 2: The PG-WASD algorithm begins with an empty matrix K and a power $h = 0$. It then builds K by adding the corresponding columns k_{h+1} while increasing h. The optimal weights are calculated for each iteration in relation to the training set as shown in 6. The mean-absolute-error ($\text{MAE} = \frac{1}{S_2}\sum_{l=1}^{S_2} |\breve{D}_l - D_l|$) is the chosen measure for evaluating the NN's performance on the validation set. Only if the addition of a column has improved performance, or decreased the MAE, are new columns retained. Until h crosses a certain threshold, which we set at $h_{\max} = 9$, this process is repeated.

Step 3: The training and validation sets are combined when $h_{\max}$ is reached, and the final set of weights is then calculated in relation to all training and validation samples that are currently available.

In summary, by shrinking the NN's structure to the shortest size feasible, the PG-WASD algorithm not only ensures that the weights are optimal, but it also makes calculation easier in future runs. When used in conjunction with cross validation, as previously mentioned, the PG-WASD training algorithm is a highly effective procedure.

3 Results and discussion

Applying the final structure of the NN to the test set is the last remaining step. It is important to note that the customer churn dataset, which consists of 21 variables and 1 target, can be found at https://www.kaggle.com/datasets/blastchar/telco-customer-churn/. This dataset was used to assess the performance of the NNs and contains information about a hypothetical telecommunications provider that offered home phone and internet services to 7043 consumers in California. The data preparation algorithm outlined in [12] is used to transform the non-numerical data in the dataset into real numbers because the WDD method only accepts input data in the form of real numbers. In order to achieve a generally balanced distribution between churn and non-churn consumers, we finally randomly picked and eliminated enough indices, since only 934 (or 13.26%) of the samples corresponded to customers who had left the telecommunications provider. The training error path (left) and the test set classification results (right) are shown in the two subfigures in Figure 1.

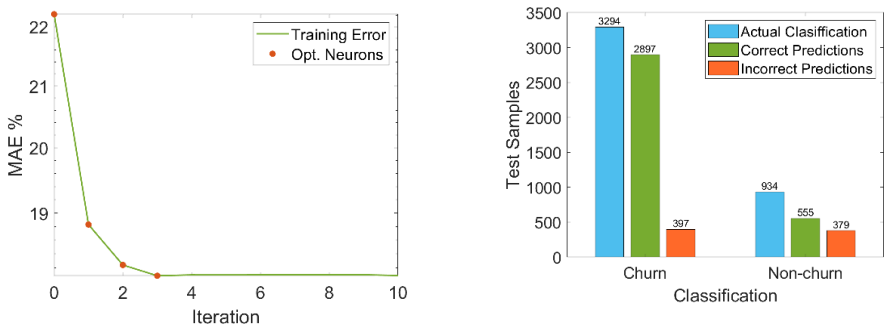

Fig. 1. Path of the training error and predictions on the test set for the PG-WASD.

A contrast with other high-performing classifiers is necessary to draw relevant conclusions about the competency of the suggested PG-WASD model. In order to achieve this, we selected three well-liked models of MATLABs' classification learner app – fine tree (FTree), kernel naive Bayes (KNB) and fine K-nearest neighbors (FKNN) – as well as the

FCI-WASD model [10]. The MAE, true positive (TP), false positive (FP), true negative (TN), false negative (FN), precision, recal, accuracy and F-score are the performance measures considered in our analysis. See [13] for more details and in-depth analysis of these measures. In Table 1, We show the aggregate test results on the customer churn dataset. With the lowest MAE and the highest marks for Recal, Accuracy, and F-score, the PG-WASD stands out from the rest of the classifiers. It is surpassed by Ftree in the Precision situation. Unlike the Ftree model, though, great success at some metrics does not appear to come at the expense of unjustifiably poor results on the remaining criteria. As a result, the PG-WASD establishes itself as a trustworthy classifier that performs remarkably well across the board rather than only excelling at one parameter.

Table 1. Performance comparison between classifiers.

Statistics	**PG-WASD**	**FCI-WASD**	**FKNN**	**Ftree**	**KNB**
MAE	0.1835	0.1837	0.2883	0.2346	0.1925
FP	0.4057	0.4068	0.4753	0.3672	0.4304
FN	0.1205	0.1205	0.2352	0.1970	0.1250
TP	0.5942	0.5931	0.5246	0.6327	0.5695
TN	0.8794	0.8794	0.7647	0.8029	0.8749
Precision	0.5942	0.5931	0.5246	0.6327	0.5695
Recal	0.8313	0.8311	0.6903	0.7625	0.8199
Accuracy	0.8164	0.8162	0.7116	0.7653	0.8074
F-score	0.6930	0.6922	0.5961	0.6916	0.6722

We add a statistical component – a mid-p-value McNemar test [14] – to the earlier study in order to accurately determine if it is possible to draw conclusions about a model that is more appropriate for predicting customer churn in the aforementioned dataset. This test seeks to determine if two classifiers predict the correct class with identical accuracy (null hypothesis) or not equal accuracy (alternative hypothesis). We used the mid-p-value McNemar test to examine every conceivable pairing of the PG-WASD and one of the other four models. Table 2 presents the results as a whole. With the exception of the pair PG-WASD and FCI-WASD, there is substantial evidence to reject the null hypothesis at the 5% significant level. Thus, PG-WASD's greater accuracy is well established. In the pair PG-WASD and FCI-WASD, the null hypothesis is not rejected. That is, the PG-WASD and FCI-WASD models have similar accuracy. Nonetheless, it is clear from the results in Table 1 that the PG-WASD outperforms the FCI-WASD in terms of overall performance.

Table 2. McNemar's test outcome.

PG-WASD	**McNemar test**	
vs.	**p-value**	**Null Hypothesis**
FCI-WASD	0.81452	Not Rejected
FKNN	2.7x10-50	Rejected
Ftree	1.8x10-19	Rejected
KNB	0.0337	Rejected

4 Conclusion

In order to classify customer churn, a PG-WASD NN is introduced in this research. In comparison to other well-known and effective classifiers, the PG-WASD showed either higher or very comparable performance capabilities when applied to a publicly available customer turnover dataset. We have been able to assert that there is a statistically significant difference in the predictive accuracy between the PG-WASD and the other models, except for the FCI-WASD. This confirms the validity of the results, which are further corroborated by the results of a McNemar test. However, the statistical test clearly shows that, in terms of overall performance, the PG-WASD performs better than the FCI-WASD. Thus, we are able to affirm that the PG-WASD classifier is a trustworthy classifier that may handle the challenge of classifying customer churn.

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. S.D. Mourtas, J. Forecast. **14**, 1512-1524 (2022)
6. S.D. Mourtas, E. Drakonakis, Z. Bragoudakis, AIMS Math. **8**, 24254-24273 (2023)
7. D. Lagios, S.D. Mourtas, P. Zervas, G. Tzimas, Mathematics **11**, 629 (2023)
8. I.N. Generalao, The Philippine Review of Economics **58**, 92-127 (2021)
9. Y. Zhang, D. Chen, C. Ye, *Deep Neural Networks: WASD Neuronet Models, Algorithms, and Applications* (CRC Press: Boca Raton, FL, USA, 2019)
10. H. Alharbi, O. Alshammari, H. Jerbi, T.E. Simos, V.N. Katsikis, S.D. Mourtas, R.D. Sahas, Mathematics **11**, 1506 (2023)
11. 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)
12. T.E. Simos, V.N. Katsikis, S.D. Mourtas, Appl. Soft Comput. **127**, 109351 (2022)
13. A. Tharwat, Classification assessment methods, Appl. Comput. Inform. **17**, 168-192 (2020)
14. M.W. Fagerland, S. Lydersen, P. Laake, BMC Med. Res. Methodol. **13**, 1-8 (2013)