Article

Forecasting economy-related data utilizing weight-constrained recurrent neural networks

Ioannis E. Livieris

Department of Computer & Informatics Engineering, Technological Educational Institute of Western Greece, GR 263-34, Greece; livieris@teiwest.gr.

Version April 18, 2019 submitted to Algorithms

Abstract: During the last decades, machine learning constitutes a significant tool in extracting useful knowledge from economic data for assisting decision-making. In this work, we evaluate the performance of weight-constrained recurrent neural networks in forecasting economic classification problems. These networks are efficiently trained with a recently proposed training algorithm, which has two major advantages. Firstly, it exploits the numerical efficiency and very low memory requirements of the limited memory BFGS matrices; Secondly, it utilizes a gradient-projection strategy for handling the bounds on the weights. The reported numerical experiments present the classification accuracy of the proposed model, providing empirical evidence that the application of the bounds on the weights of the recurrent neural network, provides more stable and reliable learning.

Keywords: Artificial neural networks; machine learning; economic data mining, classification, constrained optimization; limited memory BFGS.

1. Introduction

The rapid advances in digital technologies as well as the vigorous development of the Internet and the significant storage capabilities of electronic media, enabled economic research centers to accumulate and store large repositories of data. This enormous amount of valuable data yields information of various economic activities such as credit history of bank customers, stock market prices and movement, companies and business funds sales records and other statistics. It is a fundamental need for business, companies and banks to extract useful knowledge from these data. This knowledge constitutes a key element for better decision-making in the increasing market volatility and competition.

Economic data mining constitute an essential process where intelligent methods are applied to extract data patterns from economic databases. This research area has rapidly grown and gained popularity in the modern economic era due to its potential to assist managerial decision-making. During the last two decades, researchers and financial managers have began to analyze economic data utilizing machine learning and data mining techniques for supporting hard policy decisions (see [1–4] and the references there in). As a result, the area of economic analysis has been dramatically changed from a rather qualitative science to a more quantitative science which is also based on knowledge extraction from databases. Nevertheless, economic data are usually imbalanced and are usually characterized by complex dimensionality and enormous noise, hindering their analysis and modeling. Therefore, the process of leveraging these data constitutes an attractive and challenging task for many experts which often requires huge efforts.

Artificial Neural Networks (ANN) have been established as one of the most dominant machine learning algorithms for extracting useful knowledge; thus their value has demonstrated across an impressive spectrum of applications [5–8]. Due to their excellent capability of self-learning and self-adapting, they are very appealing to deal with economic and financial problems with poorly defined system models, noisy data and strong presence of nonlinear effects. In the literature, several neural network architectures have been proposed [9]. Recurrent Neural Networks (RNNs) constitute a class of neural networks which are well-known for their power to memorise time dependencies and model nonlinear systems.In contrast to the classical feed-forward neural networks in which all the inputs and outputs are independent of each other, RNNs allow previous outputs to be used as inputs while having hidden states. Their main advantages are their ability to deal with time varying input and output through and their own natural temporal operation [10–12].

Mathematically, the problem of efficiently training a RNN can be formulated as the minimization of an error function $E(w)$ which depends on the connection weights w of the network. Recently, Livieris [13] proposed a new approach for improving the generalization ability of neural networks. by applying additional conditions on the weights in the form of bound-constraints, during the training process. The motivation behind this approach focuses in defining the weights of the trained network in a more uniform way in order all inputs and neurons of the network to be efficiently explored and exploited. Therefore the problem of training a neural network is reformulated to a constrained optimization problem, namely

$$\min\{E(w) \,:\, w \in \mathcal{B}\}, \tag{1}$$

with

$$\mathcal{B} = \{w \in \mathbb{R}^n \,:\, l \leq w \leq u\}, \tag{2}$$

where $l \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ denote the lower and upper bounds on the weights, respectively. Moreover, for evaluating the efficacy and the efficiency of this approach, Livieris proposed a new weight-constrained neural network training algorithm. The proposed algorithm exploits the numerical efficiency and very low memory requirements of the limited memory BFGS matrices together with a gradient-projection strategy for handling the bounds on the weights.

In this work, we examine and evaluate the performance of weight-constrained recurrent neural networks for the classification of economic data. To this end, we conducted a series of experiments using in three famous economic classification problems, the bank marketing problem, the German credit approval problem and the banknote authentication dataset. Our numerical experiments demonstrate that the classification efficiency of the proposed algorithm outperforms classical neural network training algorithms, providing empirical evidence that it provides more stable, efficient and reliable learning.

The remainder of this paper is organized as follows: Section 2 briefly discusses recent studies concerning the application of machine learning in economic data. Section 3 presents the weight-constrained recurrent neural network training algorithm. Section 4 presents the numerical experiments utilizing the performance profiles Dolan and Morè [14]. Finally, Section 5 presents the conclusions and our proposals for future research.

Notations. Throughout this paper, the gradient of the error function is indicated by $\nabla E(w)$ and the vectors $s_k = w_{k+1} - w_k$ and $y_k = \nabla E(w_{k+1}) - \nabla E(w_k)$ represent the evolutions of the current point and of the error function gradient between two successive iterations.

2. Related work

Research on the predictability on economic data has a long history in economics; thus economic data mining systems have gained popularity during the last two decades. The main reason for the increasing popularity of these systems is their ability to support economic decision-making considerations such as information acquisition and decision-making error costs. A number of rewarding studies have been carried out in recent years and some useful outcomes are briefly presented below.

Chang et al. [1] proposed a novel classifier based on the artificial immune network (named AINE-based classifier) to evaluate the applicants' credit scores. Additionally, they conducted a variety of experiments, utilizing two real-world datasets of the banking industries to explore the effectiveness of their proposed model. The presented experimental results demonstrated that the AINE-based classifier outperforms state-of-the-art classifiers, in terms of prediction accuracy. Furthermore, the authors claimed that the proposed model can provide the credit card issuer with accurate and valuable information of credit scoring analysis to avoid making incorrect decisions.

Moro et al. [2] proposed a personal and intelligent decision support system which utilizes a data mining approach for the selection of bank telemarketing clients. Their primary goal was to model the success of subscribing a long-term deposit using attributes which were known before the telemarketing call was executed. For this purpose, they utilized a large dataset of 150 features related with bank client, product and social-economic attributes. In the modeling phase of their proposed framework, a semi-automatic feature selection was explored which resulted in selecting a reduced set of 22 features. In the sequel, they evaluated the classification performance of various machine learning algorithms. Their experimental results revealed that ANN demonstrated the highest classification accuracy.

Tkavc et al. [3] provided an extensive review of neural networks applications on several economic and business classification problems. Their investigation revealed that most of the research was aimed at financial distress and bankruptcy problems, stock price forecasting and decision support, with special attention to classification tasks. Furthermore, the authors claimed that most research papers argued that neural networks outperformed conventional approaches such as discriminant analysis, Bayesian classifiers and linear regression.

Zakaryazad and Duman [15] proposed an ANN model incorporating a new penalty function which gives variable penalties to the misclassification of instances considering their individual significance (profit of correct classification and/or cost of misclassification). More specifically, they have modified the sum of square errors function by changing its values with respect to profit of each instance in order to generate individual penalties. To this end, they have introduced seven versions of ANN classifiers in total where each of them consists of a modification of the original ANN classifier. The performance of their proposed framework was evaluated on two real-world datasets from fraud detection and a dataset about bank marketing. The reported numerical experiments revealed that there was no champion model for all datasets but the different versions of proposed model exhibited statistical improvement in the total net profit as compared to several classification algorithms.

In more recent works, Villuendas-Rey et al. [4] introduced a novel supervised learning model, called Naive Associative Classifier (NAC), which boosts simplicity, transparency, transportability and accuracy. Their proposed model was evaluated using finance-related datasets including bank telemarketing, credit assignment, bankruptcy and banknote authentication. The numerical experiments presented that NAC exhibited considerable capability in solving financial classification problems, highlighting the adequacy of the proposed model for decision support. Furthermore, the authors discussed in detail the advantages and limitations of the NAC, and they presented some possible improvements and extension of their framework.

Jena et al. [16] focused on predicting banking credit scoring assessment using a Predictive k-Nearest Neighbour classifier. To evaluate the performance of the proposed algorithm against traditional classification models, they have utilized two credit approval datasets: Australian credit and German credit. Based on their numerical experiments the authors claimed that "*the proposed algorithm has a potential to accurately perform credit scoring assessment in real time*".

Livieris et al. [17] evaluated the performance of two ensemble semi-supervised learning algorithms for the credit scoring problem. The proposed algorithms exploit the predictions of three of the most efficient and popular self-labeled algorithms: Self-training, Co-training and Tri-training, using different voting methodologies. Their preliminary numerical experiments demonstrated the classification efficiency of the presented algorithms on three credit scoring datasets. Thus, the authors concluded that reliable and robust prediction models could be developed by the adaptation of ensemble techniques in the semi-supervised learning framework.

3. Weight-constrained recurrent neural network training algorithm

In this section, we present the Weight-Constrained Recurrent Neural Network (WCRNN) training algorithm while a high level description is presented in Algorithm 1 for completeness.

The original BFGS method requires the storage and manipulation of an $n \times n$ matrix. Nevertheless, for large-scale problems such as neural network training, this is unwieldy. The limited-memory BFGS method attempts to alleviate this handicap by storing only a (usually) small number of m curvature pairs.

Let $\hat{m} = \min\{k, m-1\}$, then given the set of correction vector pairs $(s_i, y_i)_{i=k-1}^{k-\hat{m}}$ satisfying $s_i^T y_i > 0$. At each iteration, the algorithm approximates the error function $E(w)$ at a point w_k, utilizing a Hessian approximation B_k by a quadratic model $m_k(w)$, namely

$$m_k(w) = E_k + g_k^T(w - w_k) + \frac{1}{2}(w - w_k)^T B_k (w - w_k), \tag{3}$$

where $E_k = E(w_k)$, $g_k = \nabla E(w_k)$.

The Hessian approximation B_k is defined (in compact form) in terms of the correction matrices S_k and Y_k are $n \times m$ matrices

$$S_k = [s_{k-\hat{m}}, \dots, s_{k-1}] \qquad \text{and} \qquad Y_k = [y_{k-\hat{m}}, \dots, y_{k-1}].$$

More specifically, the limited memory matrix B_k is obtained from $\hat{m}$ updates to the basic matrix $B_0^{(k)} = \theta_k I$ by

$$B_k = \theta_k I - W_k M_k^{-1} W_k, \tag{4}$$

where

$$\begin{aligned} W_k &= [\theta_k S_k \quad Y_k], \\ M_k &= \begin{bmatrix} \theta_k S_k^T S_k & L_k \\ L_k^T & D_k \end{bmatrix}, \end{aligned}$$

θ_k is a positive scalar and D_k and L_k are the matrices

$$D_k = diag\left[s_{k-\hat{m}}^T y_{k-\hat{m}}, \dots, s_{k-1}^T y_{k-1}\right].$$

and

$$(L_k)_{ij} = \begin{cases} (s_{k-\hat{m}-1+i})^T (y_{k-\hat{m}-1+j}), & \text{if } i > j; \\ 0, & \text{otherwise.} \end{cases}$$

It is worth noticing that the computation of B_k is performed via a computationally efficient recursive technique presented by Zhu et al. [18] which requires only vector inner products with complexity $\mathcal{O}(m^2 n)$.

In the sequel, the algorithm performs a minimization procedure of the approximation model $m_k(w)$ to compute the new vector of weights, which consists of three stages: the generalized Cauchy point, the subspace minimisation; and the line search.

Stage I: Cauchy point computation. The basic aim of this stage is to approximately minimize the model $m_k(w)$ subject to the feasible domain

$$D = \{w \in \mathbb{R}^n \mid l \leq w \leq u\},$$

Therefore, the gradient projection method is utilized in order to compute the generalized Cauchy point w^C and eventually find a set of active bounds $\mathcal{A}(w^C)$. More specifically, let w_k be the current iterate and the path $w(t)$ defined by

$$w(t) = P(w_k - t\nabla E_k; l; u).$$

where P denotes the projection of the steepest descent direction on the feasible domain D. The generalized Cauchy point w^C is computed as the local minimum quadratic approximation of the error function on the path defined by $w(t)$. Next, the active set $\mathcal{A}(w^C)$ is consists of the indices of the variables whose values at w^C are at lower or upper bound; thus these variables are held fixed.

Stage II: Subspace minimization. After the active set of variables is obtained, then the quadratic model (3) is approximately minimized with respect to the non-active variables utilizing a direct primal method [19], that is

$$\overline{w}_{k+1} = \arg\min_{w \in D_S} m_k(w). \tag{5}$$

Notice that the feasibility domain is reduced to a subspace of the feasibility domain

$$D_S = \left\{ w \in \mathbb{R} \,|\, l_i \leq w_i \leq u_i,\ \forall i \notin \mathcal{A}(w^C) \right\}.$$

by considering as free variables, the variables which are not fixed on limits; while the remaining variables are fixed on their boundary value obtained during the Cauchy point calculation stage.

Stage III: Line search. In this stage, the new iterate w_{k+1} is computed by performing a line search along $d_k = \overline{w}_{k+1} - w_k$ which satisfies the strong Wolfe line search conditions, that is

$$E_{k+1} \leq E_k + c_1 \eta_k \nabla E_k^T d_k, \tag{6}$$

$$|\nabla E_{k+1}^T d_k| \leq c_2 |\nabla E_k^T d_k|. \tag{7}$$

with $0 < c_1 < c_2 < 1$. It is worth mentioning that the learning rate η_k is computed utilizing the line search procedure of Moré and Thuente [20] which employs quadratic and cubic interpolation schemes and safeguards in satisfying the strong Wolfe line search conditions.

Algorithm 1: WCRNN

Step 1. Set $k = 0$.
Step 2. **repeat**
Step 3. Calculate the error function value E_k and its gradient ∇E_k at w_k.
Step 4. Set the quadratic model (3) at w_k.
Step 5. Calculate the generalized Cauchy point w^C. (*Stage I*)
Step 6. Define the active set $\mathcal{A}(w^C)$.
Step 7. Minimize the quadratic model (3) with respect to the non-active variables, namely (*Stage II*)

$$\overline{w}_{k+1} = \arg\min_{w \in D_S} m_k(w),$$

where $D_S = \{ w \in \mathbb{R} \,|\, l_i \leq w_i \leq u_i,\ \forall i \notin \mathcal{A}(w^C) \}$.
Step 8. Set $d_k = \overline{w}_{k+1} - w_k$. (*Stage III*)
Step 9. Compute the learning rate η_k satisfying the strong Wolfe line search conditions (6) and (7).
Step 10. Update the weights $w_{k+1} = w_k + \eta_k d_k$.
Step 11. Set $k = k + 1$.
Step 12. **until** (stopping criterion).

---

4. Experimental results

In this section, we present a series of experiments in order to evaluate the performance of WCRNN training algorithm in three famous economic classification problems acquired by the UCI Repository of machine learning databases [21]: the bank marketing problem, the German credit approval problem and the banknote authentication problem.

The implementation code was written in Matlab 7.6 and the simulations have been carried out on a PC (2.66GHz Quad-Core processor, 4Gbyte RAM) running Linux operating system while the results have been averaged over 100 simulations. We have chosen the RNN architecture of the nonlinear autoregressive network with exogenous inputs which have reported very good performance in literature [22,23]. All networks and received the same sequence of input patterns, the weights were initiated using the Nguyen-Widrow method [24] and all nodes had logistic activation functions. Moreover, the categorical variables in all datasets were handled utilizing label encoding process.

The classification performance was evaluated utilizing the standard procedure called *stratified 10-fold cross-validation* and the following two performance metrics: F_1*-score* and *accuracy*. It is worth noticing that F_1-score consists of a harmonic mean of precision and recall while accuracy is the ratio of correct predictions of a classification model [13].

Our experimental analysis was obtained by conducting a two phase procedure: In the first phase, the classification performance of the WCRNN algorithm was evaluated against the state-of-the-art neural network training algorithms; while in the second phase, we compare the performance of the RNNs which were trained with the proposed algorithm against the most popular and frequently utilized supervised classification algorithms.

4.1. *Performance evaluation of WCRNN against state-of-the-art neural network training algorithms*

Next, we briefly describe each classification problem and present the performance comparison between the algorithm WCRNN against state-of-the-art training algorithms, i.e. Resilient backpropagation, scaled conjugate gradient and Levenberg-Marquardt training algorithm which were utilized with their default parameter settings. It is worth noticing that we utilized several neural networks architectures and selected the ones which presented the best average performance for each benchmark.

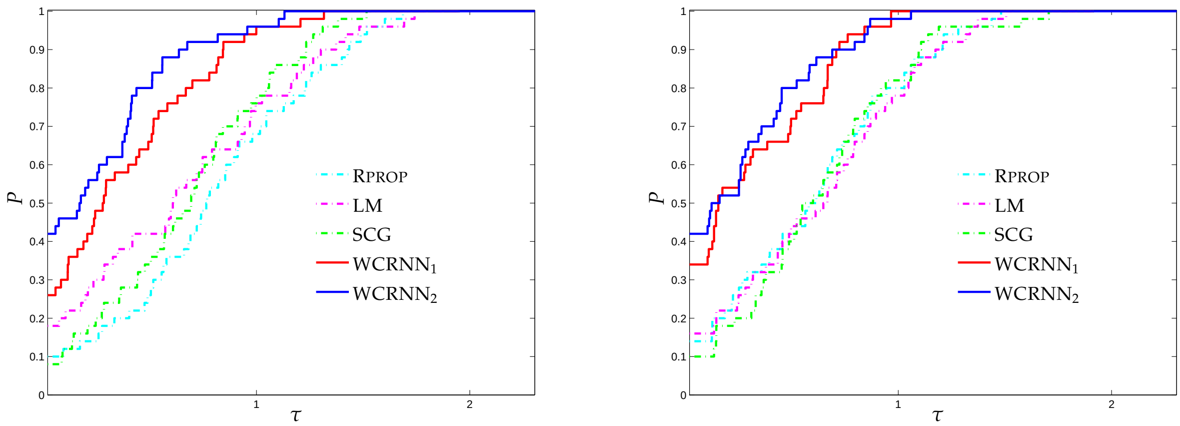
Furthermore, since a small number of simulations tend to dominate these results, the cumulative total for a performance metric over all simulations does not seem to be too informative. Therefore, similar to [13], we also utilized the performance profiles of Dolan and Morè [14] relative to both performance metrics, to present perhaps the most complete information in terms of robustness, efficiency and solution quality. The use of performance profiles eliminates the influence of a small number of simulations on the benchmarking process and the sensitivity of results associated with the ranking of solvers. The performance profile plots the fraction P of simulations for which any given algorithm is within a factor τ of the best training algorithm. The curves in the following figures have the following meaning.

- "RPROP" stands for Resilient backpropagation.
- "LM" stands for Levenberg-Marquardt training algorithm.
- "SCG" stands for scaled conjugate gradient.
- "WCRNN_1" stands for Algorithm 1 with bounds $[-1, 1]$ on all weights.
- "WCRNN_2" stands for Algorithm 1 with bounds $[-2, 2]$ on all weights.

4.1.1. Bank marketing dataset

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution and the classification goal is to predict if the client will subscribe a term deposit. Each observation represents a customer and is described by 17 attributes, both categorical and continuous corresponding to a total of 4119 contacts. During these phone campaigns, an attractive long-term deposit application, with good interest rates, was offered. For each contact, a large number of attributes was stored and if there was a success (the target variable). For the whole database considered, there were 451 successes (11% success rate) and 3668 failures (89% failure rate). The network architectures consists of 1 hidden layer with 10 neurons and an output layer of 2 neurons while the error goal was set to $E_G \leq 0.05$ within the limit of 1000 epochs.

Figure 1 presents the performance profile for the Bank marketing classification problem, investigating the performance of each training algorithm. Firstly, we note that both versions for the proposed training algorithm illustrate the highest probability of being the optimal training algorithms in terms of classification accuracy, since their curves lie on the top. More analytically, WCRNN_1 and WCRNN_2 report 26% and 42% of simulations with the best F_1-score, respectively; while the state-of-the-art training algorithms RPROP, LM and SCG present 10%, 18% and 6%, respectively. Furthermore, WCRNN_1 and WCRNN_2 report 34% and 42% of simulations with the best accuracy, respectively; while the state-of-the-art training algorithms RPROP, LM and SCG present 14%, 16% and 10%, respectively. Summarizing, we conclude that the application of the bounds on the weights of the RNNs, increased the overall classification accuracy, in most cases. However, it is worth noticing that in case the bounds are too tight this will not substantially benefit much the classification performance.



(a) F_1-score (b) Accuracy

Figure 1. Log_{10} scaled performance profiles for the bank management classification problem. The performance profile plots, for every $\tau \geq 1$, the proportion $P(\tau)$ of simulations for which any training algorithm has a performance within a factor τ of the best algorithm.

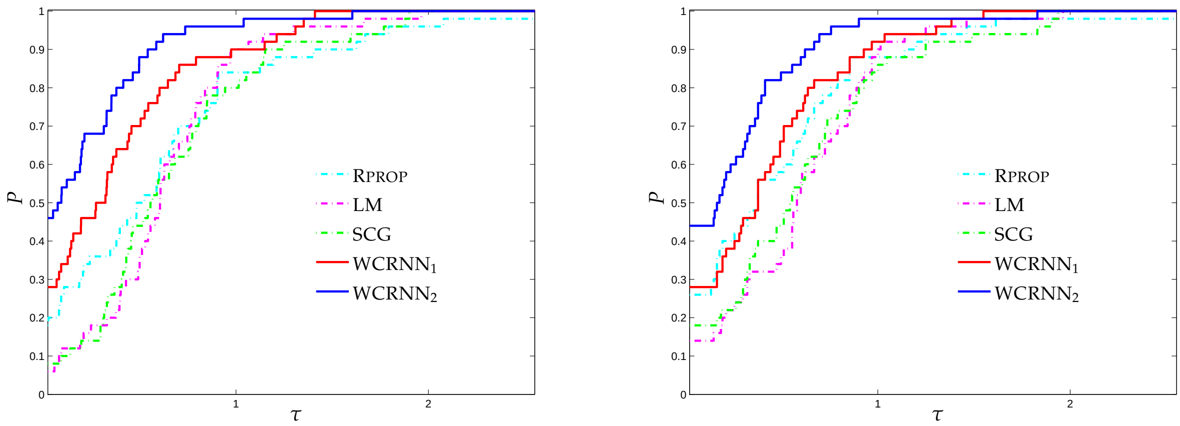
4.1.2. German credit approval problem

The German credit approval dataset contains all the details concerning approved or rejected credit card applications in Germany. This imbalanced dataset is constituted by 1000 instances (300 negative decisions and 700 positive decisions), with 20 explanatory variables (7 continuous and 13 categorical) The interesting thing about this classification problem is that the data varies and has mixture of attributes which is continuous, nominal with small numbers of values and nominal with larger numbers of values. The network architectures consists of 1 hidden layer with 30 neurons and an output layer of 2 neurons while the error goal was set to $E_G \leq 0.1$ within the limit of 1000 epochs.

Figures 2(a) and 2(b) present the performance profiles for the German credit approval classification problem, based on F_1-score and accuracy. Firstly, it is worth noticing that WCRNN_1 and WCRNN_2 outperformed the classical training algorithms, presenting the highest probabilities of being the optimal solvers, relative to both performance metrics. Regarding the F_1-score metric, WCRNN_2 exhibits the best performance, outperforming the rest training algorithms, followed by WCRNN_1. Furthermore, WCRNN_1 and WCRNN_2 report 28% and 44% of simulations with the highest classification accuracy, respectively; while RPROP, LM and SCG report 26%, 14% and 18%, respectively. Thus, the interpretation of Figure 2 demonstrates that the application of the bounds on the weights of the neural network, increased the overall classification accuracy.

4.1.3. Banknote authentication problem

The data for this classification problem were extracted from images taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print

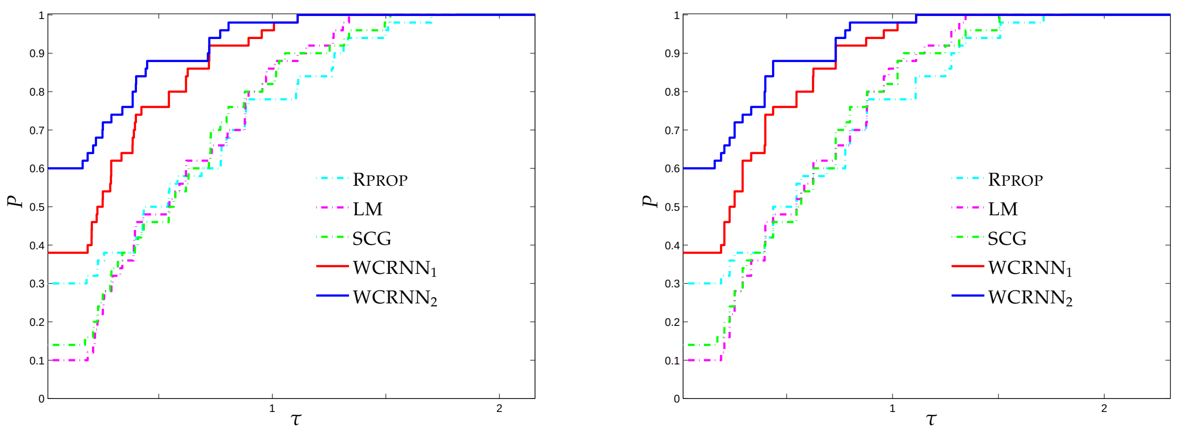


(a) F_1-score (b) Accuracy

Figure 2. Log_{10} scaled performance profiles for the German credit approval classification problem. The performance profile plots, for every $\tau \geq 1$, the proportion $P(\tau)$ of simulations for which any training algorithm has a performance within a factor τ of the best algorithm.

inspection was used. The final images have 400×400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images. The network architectures consists of 2 hidden layers with 8 and 4 neurons each and an output layer of 2 neurons while the error goal was set to $E_G \leq 0.01$ within the limit of 2000 epochs.

Figure 3 illustrates the performance profiles regarding the Banknote authentication classification problem. It is worth noticing that WCRNN$_2$ exhibit the highest probability of being the optimal solver, significantly outperforming all other training algorithms, followed by WCRNN$_1$. More specifically, WCRNN$_1$ and WCRNN$_2$ report 30% and 54% of simulations with the highest F_1-score, respectively; while the state-of-the-art training algorithms RPROP, LM and SCG present 24%, 8% and 12%, respectively. Furthermore, WCRNN$_1$ and WCRNN$_2$ report 38% and 60% of simulations with the highest classification accuracy, respectively; while the state-of-the-art training algorithms RPROP, LM and SCG present 30%, 10% and 14%, respectively. Summarizing, we conclude that the application of the bounds on the weights of the RNNs, increased the overall classification accuracy; however, in case the bounds are too tight this will not substantially benefit much the classification performance.



(a) F_1-score (b) Accuracy

Figure 3. Log_{10} scaled performance profiles for the banknote authentication classification problem. The performance profile plots, for every $\tau \geq 1$, the proportion $P(\tau)$ of simulations for which any training algorithm has a performance within a factor τ of the best algorithm.

4.2. Performance evaluation against state-of-the-art supervised algorithms

In the sequel, we evaluate the performance of the RNNs which were trained with WCRNN algorithm against the state-of-the-art supervised algorithms, Naive Bayes [25], Support Vector Machines (SVM) [26], 3NN [27] and Random Forest [28]. These classification models constitute some of the most effective and widely used data mining algorithms for classification [29].

Table 1. Performance comparison of WCRNN_1 and WCRNN_2 against state-of-the-art classification algorithms, regarding F_1-score and accuracy

Algorithm	F_1-score			Accuracy		
	Bank	German	Banknote	Bank	German	Banknote
RNN (WCRNN_1)	**0.93**	**0.77**	**0.98**	91.64%	74.73%	97.88%
RNN (WCRNN_2)	**0.93**	**0.77**	**0.98**	**91.70%**	**75.60%**	98.11%
Naive Bayes	0.88	0.72	0.79	86.76%	72.80%	79.66%
SVM	0.88	0.71	0.92	90.02%	73.60%	92.41%
3NN	0.87	0.71	0.97	88.88%	72.40%	97.44%
Random Forest	0.91	0.74	**0.98**	91.30%	75.10%	**98.32%**

Table 1 presents the performance comparison of WCRNN_1 and WCRNN_2 against state-of-the-art classification algorithms, relative to F_1-score and accuracy. Notice that the best performance is highlighted in bold for each metric. Random Forest exhibited the best performance among state-of-the art classifiers which is probably due to the fact that Random Forest is the only classifier based on an ensemble methodology. Nevertheless, the RNNs trained with WCRNN_1 and WCRNN_2 exhibited the highest average F_1-score, relative to all problems. Regarding the classification accuracy, the RNNs (WCRNN_2) exhibit the best performance for Bank marketing and German credit approval datasets, followed by RNN (WCRNN_1). Moreover, Random Forest reports the highest accuracy for banknote authentication dataset, slightly outperforming the RNN (WCRNN_2).

Summarizing, it is worth mentioning that the RNNs trained with both versions of WCRNN outperform all state-of-the-art classifiers in terms of F_1-score. Furthermore, their classification performance is superior to all single classifiers and competitive to the ensemble-based Random Forest, regarding all benchmarks.

5. Conclusions

In this work, we evaluated the classification accuracy of weight-constrained recurrent neural networks in forecasting economic data. The classification efficiency of these new prediction models is based on a recently proposed training algorithm, which exploits the numerical efficiency and very low memory requirements of the limited memory BFGS matrices together with a gradient-projection strategy for handling the bounds on the weights. By placing constraints on the values of weights, the likelihood that some weights will "*blow up*" to unrealistic values is considerably reduced. Our numerical experiments demonstrated the classification efficiency of the proposed models, as confirmed statistically by the performance profiles. Therefore, we are able to conclude that proposed algorithm appears to efficiently train RNNs with improved classification ability in domains such as forecasting economic benchmarks.

The determination of optimal bounds on the weights is a rather challenging task; therefore, more research and experiments are needed. To this end, the question of what should be the values of the bounds for each benchmark or what constraints should be applied to the weights of each layer is still under consideration. An interesting idea could be to auto-adjust the bounds during the training process utilizing a strategy based on the use of a validation set. Probably, the required research to answer these questions, may reveal additional and crucial information and questions.

Our future work is concentrated on incorporating the proposed methodology to more advanced and complex architectures such as Long Short Term Memory neural networks and deep neural networks, together with sophisticated techniques such as dropout and batch normalization. Since our experimental results are quite encouraging, a next step could be the evaluation of the proposed framework for the prediction of stock exchange index movement and for forecasting the value stock price indices and prices. Furthermore, another interesting aspect for future research could be the utilization of rule induction and discovery methods or even the use of synthetic data for further accuracy improvement based on the insights received in their training/testing periods (see [30,31] and the references there in). Finally, we intend to conduct extensive empirical experiments by applying the proposed algorithm in specific scientific fields and evaluate its performance on large real-world datasets, such as educational, healthcare, etc.

Funding: This research received no external funding.

Conflicts of Interest: The author declare no conflict of interest.

1. Chang, S.Y.; Yeh, T.Y. An artificial immune classifier for credit scoring analysis. *Applied Soft Computing* **2012**, *12*, 611–618.
2. Moro, S.; Cortez, P.; Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* **2014**, *62*, 22–31.
3. Tkáč, M.; Verner, R. Artificial neural networks in business: Two decades of research. *Applied Soft Computing* **2016**, *38*, 788–804.
4. Villuendas-Rey, Y.; Rey-Benguría, C.F.; Ferreira-Santiago, Á.; Camacho-Nieto, O.; Yáñez-Márquez, C. The naïve associative classifier (NAC): a novel, simple, transparent, and accurate classification model evaluated on financial data. *Neurocomputing* **2017**, *265*, 105–115.
5. Chen, J.F.; Hsieh, H.N.; Do, Q. Predicting student academic performance: a comparison of two meta-heuristic algorithms inspired by cuckoo birds for training neural networks. *Algorithms* **2014**, *7*, 538–553.
6. Huang, X.; Wang, Z. Multiple Artificial Neural Networks with Interaction Noise for Estimation of Spatial Categorical Variables. *Algorithms* **2016**, *9*, 56.
7. Purnamasari, P.; Ratna, A.; Kusumoputro, B. Development of filtered bispectrum for EEG signal feature extraction in automatic emotion recognition using artificial neural networks. *Algorithms* **2017**, *10*, 63.
8. Wu, F.; Fu, K.; Wang, Y.; Xiao, Z.; Fu, X. A spatial-temporal-semantic neural network algorithm for location prediction on moving objects. *Algorithms* **2017**, *10*, 37.
9. Ferri, M. Why topology for machine learning and knowledge extraction? *Machine Learning and Knowledge Extraction* **2018**, *1*, 115–120.
10. Suzuki, K. *Artificial Neural Networks-Architectures and Applications*; 2013.
11. Singh, D.; Merdivan, E.; Psychoula, I.; Kropf, J.; Hanke, S.; Geist, M.; Holzinger, A. Human activity recognition using recurrent neural networks. International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer, 2017, pp. 267–274.
12. Shanmuganathan, S.; Samarasinghe, S. *Artificial neural network modelling*; Vol. 628, Springer, 2016.
13. Livieris, I.E. Improving the Classification Efficiency of an ANN Utilizing a New Training Methodology. *Informatics* **2018**, *6*.
14. Dolan, E.; Moré, J. Benchmarking optimization software with performance profiles. *Mathematical Programming* **2002**, *91*, 201–213.
15. Zakaryazad, A.; Duman, E. A profit-driven Artificial Neural Network (ANN) with applications to fraud detection and direct marketing. *Neurocomputing* **2016**, *175*, 121–131.
16. Jena, S.K.; Kumar, A.; Dwivedy, M. Banking Credit Scoring Assessment Using Predictive K-Nearest Neighbour (PKNN) Classifier. In *Handbook of Research on Intelligent Techniques and Modeling Applications in Marketing Analytics*; IGI Global, 2017; pp. 332–350.
17. Livieris, I.E.; Kiriakidou, N.; Kanavos, A.; Tampakas, V.; Pintelas, P. On Ensemble SSL Algorithms for Credit Scoring Problem. Informatics, 2018, Vol. 5, p. 40.
18. Zhu, C.; Byrd, R.H.; Lu, P.; Nocedal, J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)* **1997**, *23*, 550–560.

19. Morales, J.L.; Nocedal, J. Remark on "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization". *ACM Transactions on Mathematical Software (TOMS)* **2011**, *38*, 7.
20. Moré, J.J.; Thuente, D.J. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software (TOMS)* **1994**, *20*, 286–307.
21. Dua, D.; Karra Taniskidou, E. UCI Machine Learning Repository, 2017.
22. Peng, C.C.; Magoulas, G.D. Nonmonotone BFGS-trained recurrent neural networks for temporal sequence processing. *Applied Mathematics and Computation* **2011**, *217*, 5421–5441.
23. Peng, C.C.; Magoulas, G.D. Nonmonotone Levenberg–Marquardt training of recurrent neural architectures for processing symbolic sequences. *Neural Computing and Applications* **2011**, *20*, 897–908.
24. Nguyen, D.; Widrow, B. Improving the learning speed of 2-layer neural network by choosing initial values of adaptive weights. *Biological Cybernetics* **1990**, *59*, 71–113.
25. Domingos, P.; Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning* **1997**, *29*, 103–130.
26. Platt, J.C. Using sparseness and analytic QP to speed training of support vector machines. Advances in neural information processing systems; Kearns, M.; Solla, S.; Cohn, D., Eds., 1999, pp. 557–563.
27. Aha, D.W. *Lazy learning*; Springer Science & Business Media, 2013.
28. Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.
29. Wu, X.; Kumar, V. *The top 10 algorithms in data mining*; CRC press, 2009.
30. Kolias, V.; Kolias, C.; Anagnostopoulos, I.; Kayafas, E. RuleMR: Classification rule discovery with MapReduce. 2014 IEEE International Conference on Big Data (Big Data). IEEE, 2014, pp. 20–28.
31. Kolias, V.; Anagnostopoulos, I.; Kayafas, E. A Covering Classification Rule Induction Approach for Big Datasets. 2014 IEEE/ACM International Symposium on Big Data Computing. IEEE, 2014, pp. 45–53.