Deep Learning Techniques for Credit Scoring

Le Quy Tai and Giang Thi Thu Huyen

***Abstract*—Credit scoring is a crucial phase in the risk management process of financial organizations and commercial banks. Deep Learning algorithms are the powerful techniques with significant improvements in classification performances in many applications of machine learning fields such as image processing and speech recognition. The key benefit of Deep Learning models is the ability to analyze and learn massive amounts of data. In this paper, we apply two deep learning architectures for credit scoring problem: 1) Sequential Deep Neural Network; 2) Convolutional Neural Network. Our experiments of three different datasets show that all the neural network performance better than traditional approaches.**

***Index Terms*—Deep learning, neural network, credit scoring, CNN.**

I. Introduction

Financial risk forecasting plays a crucial role in all financial organizations especially commercial banks. In that process, credit scoring for each customer is an important task. It helps financial organizations decide whether or not to grant credit to customers who apply to them [1]. In data mining field, basically, credit scoring is modeled as a classification task with the target is to distinguish between "good" or "bad" customers.

Credit scoring and internal customer rating are the processes of accessing the ability of performing financial obligation of a customer against a bank, i.e., paying interest or an original loan on due date, or other credit conditions for the evaluation and classification of risks in the commercial bank during its credit procedures. The credit score of a customer is changed over time and recognized through the evaluation process. It is based on the existing customers' financial and non-financial data at the point time of credit scoring and customer rating.

David Durand [2], in 1941, introduced measurements to distinguish the good or bad loans by analysis the applicants' characteristics. Later, researchers continued proposing different methods in order to score the credit ability of customers. Basically, there are three main techniques used for credit scoring: 1) expert scoring model; 2) statistical models and 3) artificial intelligence (AI) methods [1]. There are some statistical models used for credit scoring: LDA (linear discriminate analysis) which uses linear discriminate function (LDF) passing through the centroids of the two classes to classify the customers; LR (logistic regression) which has the ability to predict default probability of an applicant and identify the variables related to his/her behaviours; MARS (multivariate adaptive regression splines) - a non-linear and non-parametric regression method; Bayesian model; Decision tree which constructs a decision tree, its common methods are C4.5 and CART. In recent years, there are several approaches using artificial intelligence methods such as Artificial Neural Network (ANN), k-Nearest neighbour, Support Vector Machine (SVM) and Genetic algorithms [1]. There methods have shown promising results in term of prediction accuracy.

In this paper, we propose a new method which adapts two effective models of deep neural network (DNN): Sequential neural network; Convolutional neural network (CNN) and do the experiments on three different datasets: German credit dataset, Australian credit approval dataset and a large dataset, the Kaggle credit dataset.

This paper is structured as follow: In Section II, we provide a very brief introduction to deep leaning technique with deep neural networks and its application in big data era. Section III describes how we prepare the training data for our networks. In the next section, we illustrate how we adapt two neural network architectures to the credit scoring problem. Next, Section V is the experiments and credit scoring results from these neural networks. Finally, we offer our conclusions and describe some possible future directions for research in Section VI.

II. Overview of Deep Learning in Big Data Analytics

The main concept of deep learning algorithms is automating the extraction of representations (abstractions) from the data [3]. Deep learning algorithms use a huge amount of supervised data to automatically extract complex representation. In fact, deep learning algorithms are deep artificial neural networks (ANN) with consecutive layers. Each layer applies a nonlinear transformation on its input and provides a representation in its output. Architecturally, an artificial neural network is a directed graph where the nodes are artificial neurons and the edges allow information from one neuron to pass to another neuron (or the same neuron in a future time step). Connections among neurons in different layers are weighted and can be changed during the ANNs learn. ANNs have an input layer and an output layer, any layers in between is called hidden layers. The data is fed to the input layer, consequently, the output of each layer is provided as an input to its next layer. The forward pass of an ANN is where information flows from the input layer, through any hidden layers, to the output [4].

Each artificial neuron computes a weighted sum of its inputs adds a learnt bias and passes this sum through an activation function. Let's consider a neuron receiving I inputs. The value of each input is represented by input vector x. The weight on the connection from input i to neuron h is denoted by w_ih (w is called the 'weights matrix')

Manuscript received February 28, 2019; revised May 21, 2019.

Le Quy Tai and Giang Thi Thu Huyen are with Banking Academy of Vietnam, Vietnam (e-mail: quytai3985@gmail.com, gianghuyenhvnh@gmail.com).

doi: 10.18178/joebm.2019.7.3.588

of the inputs into neuron h can be written $a_h = \sum_{i=1}^{I} x_i w_{ih}$. The network input a_h is then passed through an activation function θ to produce the neuron's final output b_h where $b_h = \theta(a_h)$. We use the these activation functions: linear: $\theta(x) = x$; rectified linear (ReLU): $\theta(x) = max(0, x)$; hyperbolic tangent (tanh): $\theta(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$; sigmoid: $\theta(x) = (1 + e^{-x})^{-1}$

Regarding to the four-Vs of Big Data characteristics, i.e., Volume, Variety, Velocity and Veracity, Deep learning algorithms and their architectures are more aptly suited to address issues related to Volume and Variety of Big Data analytics [3]. Deep learning has the ability to exploit the availability of massive amounts of data, where algorithms with shallow learning fail to explore and understand the higher complexities of data patterns. In addition, Deep learning can deal with data abstraction and representations, i.e., extract features from every new data types with different formats and/or form a diversity of sources.

III. Training Data

Deep neural networks need a huge amount of training data because they have a large number of trainable parameters (the network weights and biases). It is also common practice in deep learning to increase the effective size of training set by duplicating the training data many times and applying realistic transformations to each copy. For example, in image classification, we might flip the image horizontally or apply sight affine transformations. In credit scoring, to take the experiments with our deep neural networks, we use three different data sets: German Credit Data[1]; Australian Credit Approval[2] and Kaggle Credit Data[3].

A. German Credit Data set

This data set was published by Professor Dr. Hans Hofmann[4] in 1994. It is a good dataset with 20 attributes (in category format) and 24 attributes (in numeric format) and contains observations for 1000 past applicants for credit. Each applicant was rated as 'good credit' (700 cases) or 'bad credit' (300 cases). In our paper, to train the neural networks, we use this data set with numeric format.

B. Australian Credit Approval

There are 690 instances in this data set with 14 attributes (6 numerical and 9 categorical attributes) and 2 class (307 instances for class 2 and 383 instances for class 1). Similar to the previous data set, all categorical features need to be transferred into numeric data before training the deep neural networks.

C. Kaggle Credit Data

This data set consist 250,000 borrowers with labeled 150,000 and 100,000 unlabeled instances. There are 10 attributes with numeric format and we do not need a pre-processing phrase before applying our deep learning models in this data set. Each instance is labeled (0/1) corresponding to 'bad credit' (139,974 instances) or 'good credit' (10,026 instances), respectively.

IV. Deep Learning Architectures for Credit Scoring

In this section, we describe how we adapted these different deep neural networks into credit scoring.

A. Deep Sequential Neural Network

Deep Sequential Neural Network [5] is a simple type of Deep Learning model with a linear stack of layers. In order to improve the performance of neural networks, we construct a dense network with different density of neurons for each layer. We use these activation functions: rectified linear (ReLU): $\theta(x) = max(0, x)$; hyperbolic tangent (tanh): $\theta(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$; sigmoid: $\theta(x) = (1 + e^{-x})^{-1}$ depending on each data set in our experiments. To penalize the deviation between the predicted and true labels, in our neural networks, we use the binomial cross-entropy loss function:

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(p_i) + (1 - p_i)\log(1 - p_i)] = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log(p_{ij})$$

where i indexes observations and j indexes classes, y is the sample label and $p_{ij} \in (0,1)$: $\sum_j p_{ij} = 1, \forall i, j$ is the prediction for a sample.

The exact architecture is as follow (used for Kaggle Credit Data):

1. Input (length = 10)
2. Dense layer (60, input dim = 10, kernel initializer = 'uniform', activation function ='relu')
3. Dense layer (5, kernel initializer = 'uniform', activation function = 'relu')
4. Fully connected (N=1, activation function = 'sigmoid')

B. Convolutional Neural Network

Convolutional Neural Network (CNN) [6] is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. CNN has wide applications in image and video recognition, recommender systems and natural language processing. CNN architecture is formed by a stack of distinct layers [7] that transform the input volume into an output volume through a differentiable function. We use several layers in our neural network architectures:

1) Convolutional layer

This layer is the core building block of a CNN and contains a set of learnable filters (or kernels) which have a small receptive field but extend through the full depth of the input volume. Each filter is convolved across the width and height of the input volume during the forward pass and computes the dot product between the entries of the filter and the input and producing an activation map of that filter (in our case, the activation map has one dimension). The

[1] https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

[2] https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval)

[3] https://www.kaggle.com/c/GiveMeSomeCredit/data

[4] Institut fur Statistik und Okonometrie, University of Hamburg

number of filters is adjusted, then the best values are selected through experiments with data sets. In our CNN models, the filter values were set to 10 with Kaggle Credit Data, 35 with German Credit Data and 100 with Australian Credit Approval Data.

2) Pooling layer

Pooling layer, a form of non-linear down-sampling, is another important component of a CNN. In our CNN, we use max pooling function to implement pooling, this function partitions the input data into a set of non-overlapping rectangles, for each sub-region, outputs the maximum.

3) ReLU (Rectified Linear Units) layer

In this layer, neurons apply the non-saturating activation function $\theta(x) = max(0, x)$. After applying activation function, the non-linear properties of the decision function and the overall network are increased without affecting the receptive fields of the convolution layer.

4) Full connected layer

The high-level reasoning in the neural network is done via fully connected layers after several convolutional and max pooling layers. Neurons in a fully connected layer have full connections to all activated neurons in the previous layer.

5) Loss layer

The loss layer provides the method to penalize the deviation between the predicted and true labels and is located in the last layer in CNNs. We use the binomial cross entropy function (described in section IV.A).

In general, we experiment with CNNs with the following architecture (in case of Australian Credit Approval Data), input of CNNs is 14 features of instances in the data set and output is the predicted class label

1. Input (length=14, shape(14,1))
2. 1D Convolution layer (filters = 100, kernel size=1)
3. Max pooling layer (pool size = 1)
4. Fully connected (N=14, activation function='relu')
5. Fully connect (N=1, activation function='sigmoid')

V. Experiments and Results

We implemented our neural networks in Python using the Keras library. Keras is high-level neural networks API, written in Python and capable of running on top of either TensorFlow or Theano.

We trained our networks on an computer with CPU Intel Core i5-3320M, RAM 8GB. On this CPU, out nets typically took between 5 minutes to 6 hours to train, the trainable parameters are variated between 120 to 1600 (depending on the input size).

To evaluate the performance of our network, we calculate four metrics: Accuracy, Precision, Recall and F1.

TP = number of true positives

FP = number of false positives

TN = number of true negatives

FN = number of false negatives

P = number of positives in ground truth

N = number of negatives in ground truth

$$accuracy = \frac{TP + TN}{P + N}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

We use four typical classifiers: k-Nearest Neighbor (kNN), Classification and Regression Trees (CART), Naive Bayes (NB) and Support Vector Machine (SVM) as the baseline results to compare the performance of our deep neural networks. We do the experiments on three data sets and use 10-fold cross validation to obtain the average values. Below is the results of our methods compared with other classifiers experimented with different data sets, bold values are the best scores. In most cases, DSNN and CNN work better than other algorithms.

Table I illustrates the performance of classifiers on the German credit data set. DSNN and CNN show their remarkable performances with approximately 76% on accuracy measurement. Measuring by F1 score, DSNN is the best classifier with 81.5%. Interestingly, SVM has the greatest score on Recall measurement but its F1 score is less than DSNN's performance. Experiment on Australian credit data proves the great performance of DSNN and CNN methods, detailed results are noted in Table II. Table III describes the performance of our methods on Kaggle credit data set. On accuracy measurement, every classifier has great performance with about 90%. However, our methods are sensitive with imbalance data, the highest F1 score of our methods is only 50.1%.

TABLE I: Credit Scoring Performance on German Credit Data Set

	kNN	**CART**	**NB**	**SVM**	**DSNN**	**CNN**
Accuracy	67.71%	64.71%	70.57%	69.42%	75.90%	**76.00%**
Precision	73.74%	74.97%	80.89%	70.43%	**81.00%**	79.00%
Recall	83.36%	74.64%	75.05%	**96.30%**	82.00%	80.00%
F1	78.01%	75.01%	77.77%	81.25%	**81.50%**	79.50%

TABLE II: Credit Scoring Performance on Australian Credit Data Set

	kNN	**CART**	**NB**	**SVM**	**DSNN**	**CNN**
Accuracy	68.51%	83.44%	80.51%	56.53%	**87.54%**	81.86%
Precision	69.14%	81.11%	**85.43%**	15.00%	85.03%	81.84%
Recall	51.03%	82.63%	67.30%	0.98%	85.48%	86.77%
F1	57.92%	81.84%	74.53%	1.83%	**83.47%**	82.13%

TABLE III: CREDIT SCORING PERFORMANCE ON KAGGLE CREDIT DATA SET

	kNN	**CART**	**NB**	**SVM**	**DSNN**	**CNN**
Accuracy	92.87%	89.20%	92.88%	93.01%	**93.63%**	93.18%
Precision	30.31%	25.15%	35.85%	36.70%	**54.00%**	35.90%
Recall	1.72%	27.86%	2.34%	28.40%	**47.70%**	37.90%
F1	3.25%	26.44%	4.38%	32.05%	**50.10%**	32.00%

VI. CONCLUSION AND FUTURE WORK

In this paper, we have adapted two Deep Neural Network architectures to credit scoring problem. Deep neural nets have a high performance compared with other classifiers in most of experiments although they express several not good results in data set which has unbalanced classes.

We believe that deep neural networks show great promise for credit scoring. There are several approaches to improve performance of Deep learning methods in credit scoring: 1) Re-construct the architecture of neural nets with suitable parameters; 2) Apply feature selection methods to reduce feature space of data set; 3) Apply other deep neural networks such as Long Short-Term Memory neural network, Deep Belief Neural Network; 4) Do the experiments on GPU or cloud to reduce the training time; 5) Collect and do the experiment on the real data set in the commercial bank.

REFERENCES

[1] X.-L. Li and Y. Zhong, "An overview of personal credit scoring: Techniques and future work," *International Journal of Intelligence Science*, vol. 2, no. 4, pp. 181–189, 2012.

[2] D. Durand, *Risk Elements in Consumer Instatement Financing*, National Bureau of Economic Research, Incorporated Publication, 1941.

[3] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, Feb. 2015.

[4] J. Kelly and W. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy-Efficient Built Environ.*, 2015, pp. 55–64.

[5] L. Denoyer and P. Gallinari, "Deep sequential neural network," University Pierre et Marie Curie - Paris, France 2014.

[6] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," *Feature Grouping*, 1999.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., pp. 1097–1105, 2012.

Lê Quý Tài is a lecturer of Management Information System Faculty, Banking Academy of Vietnam. He had a bachelor degree of education in informatics from Thai Nguyen University of Education in 2007 and received master degree of informatics at College of Technology, Vietnam National University Hanoi in 2011. From 2016, he becomes a PhD student in L3S at L3S Research Center, Leibniz University Hannover, Germany. His research interest is data mining and related problems. Currently, his research is concentrated on deep learning techniques on mobility data mining.

Giang Thị Thu Huyền received the bachelor degree from Ha Noi University of Science and Technology and the master degree from College of Technology – National University Ha Noi, Viet Nam, all in information systems. She is a lecturer of Management Information System Faculty, Banking Academy of Viet Nam. Her research interests include data mining, knowledge based systems, machine learning.