

Neural Networks for Financial Market Risk Classification

Narek Abroyan

Division of Computer Systems and Informatics, National Polytechnic University of Armenia, Yerevan, Armenia
Email: n.abroyan@polytechnic.am

Abstract. During the last several years machine learning started to revolutionize many industrial fields by replacing human intellectual work with recent technologies. Machine learning has started to be used in financial sphere as well for predicting stock prices, detecting fraud actions etc. In this work, we are focusing on financial market risk classification, which is a part of fraud action detection problem. Although artificial intelligence researchers and specialists achieved notable results in visual, voice signal and natural language processing tasks by using new methods and approaches of deep learning, such as convolutional and recurrent neural networks, not many results are in the sphere of elaboration of real-time non-stationary data, such as financial data. Moreover, methods which are used in industry usually are not published. The goal of this work is exploring, experimenting and providing new and more effective methods of classification of financial non-stationary risk data by using neural networks.

Keywords: Deep learning, convolutional neural networks, recurrent neural networks, financial data, risky transaction, classification.

1 Introduction

In general, financial data is considered to be real-time data. By saying real-time data, we mean that its total correctness depends not only upon its logical correctness, but also upon the time in which it is used [1]. A good example of such data could be stock prices that rapidly change over time. Another example of real-time system can be stock market transaction (buy/sell) fraud detection system. That is, all orders in stock exchange market are processed and tested for not being fraud. One way of doing that, is using Gaussian distribution [2]. When an alert is received about suspicious transaction, that transaction is suspended and categorized as a risky transaction, after which a qualified business analyst must decide whether that transaction is normal and should be resumed or fraud and rejected. In fact, that person solves a real-time classification problem, which can be modelled into a machine learning problem. In scope of this work, the later type of problem is being considered. In such example, correctness of data and approval decision strongly depend on current time. Data of a market order sometime later can be changed qualitatively and may not be relevant to the training data. For now, we are going to train our model based on historical data and do classification in real time. We will not take into account the fact that historical data may become irrelevant over some time (this is going to be a topic for further research). The basic structure of described system is presented in Fig. 1.

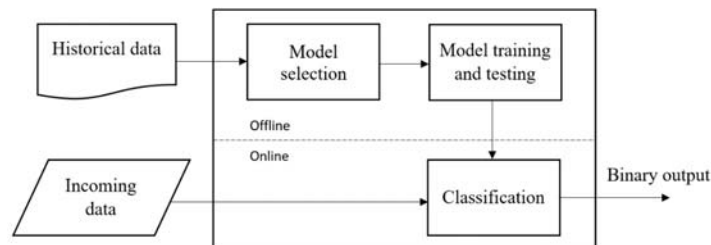


Figure 1. The illustration of real-time classification system

During the past several years, the techniques and methods that have been developed from deep learning research started a new era of machine learning and artificial intelligence. The usage of convolutional, recurrent neural networks, restricted Boltzmann machines, and deep belief networks hugely improved the results of machine and deep learning problems such as visual, natural language processing, speech recognition. However, researchers do not often talk about real-time machine learning, which has some specific properties [1]. Moreover, financial companies actively use machine learning for user action fraud detection problems, but usually they prefer not to publish their methods and technologies. Thus, not much public research is available in this particular field and there is a need to develop it. Since convolutional and recurrent neural networks gave quite superior results for supervised learning problems, we are concentrating on these neural networks and try to use them for financial market risk classification.

In this work, we continue earlier started research [3]. As it was turned out, in case of non-stationary data just by switching from classical feed forward to convolutional and recurrent neural networks mainly keeping the structure of neural network, the performance will not be improved [3]. Hence, there is a need to take extra steps and pick a correct architecture for the neural network. Evenmore, as no free lunch theorem states, no algorithm performs universally better than any other [4]. For that reason the architecture should be picked up individually for each specific problem. In this work, we are concentrating on convolutional and recurrent neural networks and try to use them for classification of real-time non-stationary data.

2 Related Works

2.1 Using Convolutional Neural Networks

Convolutional neural networks are a specific type of neural networks that are best known for processing data with grid-like structure [4]. However, they can be used also for one dimensional data such as time-series (i.e. one-dimensional grid, whose samples are at regular time intervals). In case of images, data can be represented as a two-dimensional grid of pixels [4]. The name “convolutional neural network” is used, because the network uses a mathematical operation of convolution [4]. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution operation instead of general matrix multiplication in at least one of their layers [4].

In addition, convolutional networks recognize two-dimensional shapes with a high degree of invariance to translation, scaling, and other forms of distortion [5]. That task is performed as a supervised problem with the help of specific structure, which has some restrictions including the following forms [5]

- Feature extraction. Each neuron takes its inputs from a local receptive field in the previous layer, thus forcing it to extract local features based on that local field. When a feature has been extracted, its exact location becomes not important [5].
- Feature mapping. Each computational layer of the network consists of several feature maps. Each feature map has the same dimensionality as the input data and each neuron within the single feature map uses the same shared weights for input [5].
- Subsampling/pooling. After each convolution layer, usually (but not mandatorily) there is a special computational layer, which performs local averaging or maximizing subsampling/pooling operation. In a result of that operation, the resolution of the feature map is reduced. Also, due to this operation the sensitivity of the feature map’s output is reduced to some forms of distortion [5].

Eventually, the convolutional neural network learns to extract its own features automatically [5] and more abstract features are learned from the lower level ones [1]. Typically, the most computationally expensive part of convolutional network training is learning the features. The fully connected network with output layer is usually relatively inexpensive, because after passing through several layers of pooling/subsampling, already the small number of features provided as input to the fully connected network.

Some typical examples of a convolutional networks for image processing are represented in [5] for handwritten characters recognition and an implementation in [6] for handwritten digits recognition.

After all this we can assume that convolutional neural networks is a good option for classification of credit card data, because they are fast, use less parameters, and extract features themselves. Wherein, the last property is quite important, because it can free us from feature hand crafting.

There is already some research for a similar problem, where credit card transaction fraud detection is done by using convolutional neural networks [7]. It is stated, that experimental results from the real transaction data of a commercial bank show that proposed method performs better than other well-known state-of-art methods [7]. However, input data for each training example were reshaped from one dimensionality to two for making possible of using two dimensional convolutional neural networks. Besides, it was used quite big neural network, which requires a lot for resource for training. In scope of this work, we try to perform a similar experiment for our problem, but with smaller convolutional neural networks. Besides, one dimensional convolutional neural networks are used in this work. On one hand by using one dimensional neural networks we will gain more performance. On the other hand, as each example is just a one dimensional vector of double numbers it is possible to use one dimensional convolutional neural network, without reshaping input data. The high-level structure of the used neural network is presented in Fig. 2.

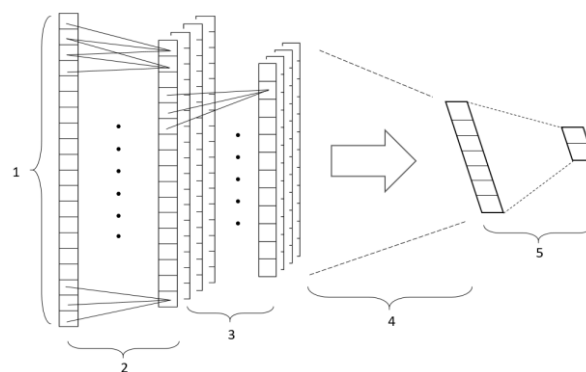


Figure 2. High-level structure of one dimensional convolutional neural network

The explanation of enumerations in Fig. 2 is given below

1. A single training sample with all features
2. Convolutional operation
3. Max or average pooling/subsampling operation
4. Visualization of the fact, that more than one convolution-pooling layers can be present in a single neural network
5. Fully connected network

So, as it is shown in the Fig. 2, a single training sample is being used for every training iteration. Convolution operation is being performed on features of each training sample with multiple filters. In a result of this several feature maps are being formed. For each of those feature maps max or average pooling operation is performed. There could be more than one convolution-pooling operation used in a single neural network. A simple fully connected network follows after all convolution-pooling operations.

2.2 Using Recurrent Neural Networks

Recurrent neural networks are special type of networks, which are being used for processing sequential data [4]. A neural network is considered to be a recurrent neural network if it has at least one feedback loop [5]. For instance, it is possible to be a recurrent network which has a single layer, where each neuron feeds its output signal back to the inputs of all the other neurons [5]. Recurrent networks can have different architectural layouts. Often, for making recurrent networks visually more understandable and look like to feedforward networks the unfolded version is represented. An example of unfolded version is shown in Fig. 3.

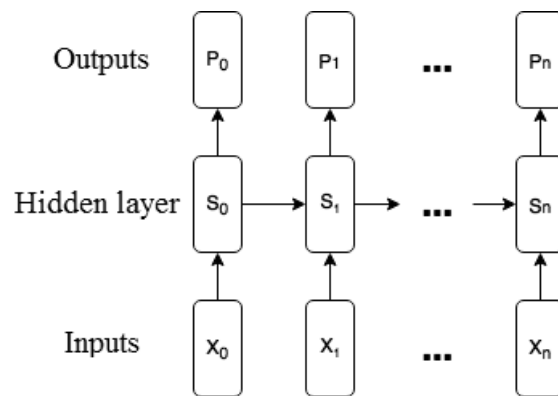


Figure. 3. An example of an unfolded recurrent neural network with one hidden layer

However, researchers started to use recurrent neural networks widely only several years ago, because before that it was difficult to train them because of long-term dependencies, giving rise to vanishing or exploding gradient. That problem is solved through using gated recurrent neural networks, such as LSTMs (long short-term memory). Gates allow the network to accumulate information over a long duration [4]. After using that information, it might be useful for the neural network to forget the old state i.e. set to zero [4]. Here a question arises – when to clear the state? Of course, one solution is clearing manually. However, instead of that it would be effective that the neural network can learn to decide when to do it. This is what gated recurrent neural networks do [4].

Recurrent neural networks have already been used again for a credit card transaction fraud detection problem [8]. In that work, it is shown that recurrent networks outperform well known Support Vector Machines (SVM) algorithm. In this work, we propose a method for classification for stock market risky transactions using recurrent neural networks and provide comparison to above mentioned methods.

3 Experiments and Results

For implementation and experiments of neural networks, Keras library with TensorFlow backend was used [6]. Experiments were performed on a private brokerage company's real data of risky transactions. Training set has about ten thousand samples and 250 features, which includes data like user's account data, order details, transaction time market data etc. Training set is not unbalanced and has approximately equal number of both labels. K-fold cross validation algorithm is being used for choosing hyperparameters [4], K was chosen 10. Binary cross-entropy was chosen as a loss function [4]. Rectified linier unit activation function was used for hidden layers [4]. As we are solving binary classification problem we are using sigmoid activation function instead of softmax for the output unit [4]. For making models to generalize well, dropout regularization method was used with chosen value of 0.5 [4]. All mentioned hyperparameters have been picked up by considering the best practices with empiric origins [2, 4, 9]. For comparing our experiments with already known result, we are going to measure our model performance with F1 score [2]. As it is known there should be a tradeoff between precision and recall and those values can be adjusted by changing decision threshold.

Thirty experiments were performed for each neural network after choosing the optimal hyperparameters. In a result of experiments up to 0.8 F1 score was achieved with convolutional neural networks just by using one convolution-pooling layers and one hidden layer from fully connected network. This result is better than the best result in [7]. And for LSTMs up to 0.91 F1 score was achieved just by using two hidden layers. In already known experiment of recurrent neural networks results are not represented through F1 score, but instead represented via FAR, TPR parameters and ROC curve [8]. So, on one hand our experimental results of convolutional neural networks outperform already known result with the same type of networks. On the other hand, by doing similar experiment with LSTMs as in [8], we get a value of F1 score, which is even better than our previous experimental result with convolutional networks. Thus, we can conclude, that using LSTM is better option than all mentioned methods.

4 Conclusion

In this work, we tried to solve stock market risky transactions classification problem by using one dimensional convolutional neural networks and LSTMs. In a result of experiments, up to 0.8 F1 score was achieved in case of convolutional networks, and up to 0.91 F1 score was achieved in case of LSTMs. Both results are better than already known results of similar classification experiments, which are better than other state-of-the art methods. Thus, we can conclude, that by using LSTMs it is possible to reach the best results for classification of stock market risky transactions.

References

1. N. H. Abroyan, R. G. Hakobyan, "A review of the usage of machine learning in real-time systems", Proceedings of NPUA, Information technologies, Electronics, Radio engineering, № 1, Yerevan, Armenia, pp. 46–54, 2016.
2. A. Ng, "CS 229 machine learning course materials", Stanford University, 2016, Available: <http://cs229.stanford.edu/materials.html>
3. N. H. Abroyan, "Classification of real-time data using deep learning", Proceedings of the 13th International Conference of Science and Technology, New Information Technologies and Systems, Penza, Russia, pp. 109-112, 2016.
4. I. Goodfellow, Y. Bengio, A. Courville, "Deep learning", Cambridge, Massachusetts, The MIT Press, 2016.
5. S. Haykin. "Neural Networks and Learning Machines", 3rd ed., McMaster University, Hamilton, Ontario, Canada, 2009.
6. F. Chollet, "Keras: deep Learning library for Theano and TensorFlow", Github, 2015, Available: <https://github.com/fchollet/keras>
7. K. Fu, D. Cheng, Y. Tu, L. Zhang, "Credit card fraud detection using convolutional neural networks", Proceedings of 23rd International Conference, ICONIP, Part III, Kyoto, Japan, pp. 483-490, 2016.
8. B. Wiese, C. Omlin, "Credit card transactions, fraud detection, and machine learning: modelling time with lstm recurrent neural networks", Innovations in Neural Information Paradigms and Applications, pp. 235-272, 2009.
9. L. Deng, D. Yu, "Deep learning: methods and applications", Foundations and Trends in Signal Processing, Vol. 7, Nos. 3–4, pp. 197–230, 2014.