

Greedy Algorithm Based Deep Learning Strategy for User Behavior Prediction and Decision Making Support

Kumar Attangudi Perichiappan Perichappan

Regional Development Center, KPMG US, Roseland, NJ, USA
Email: kumarattangudiperich@kpmg.com

How to cite this paper: Perichappan, K.A.P. (2018) Greedy Algorithm Based Deep Learning Strategy for User Behavior Prediction and Decision Making Support. *Journal of Computer and Communications*, 6, 45-53.

<https://doi.org/10.4236/jcc.2018.66004>

Received: May 13, 2018

Accepted: June 24, 2018

Published: June 27, 2018

Copyright © 2018 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we suggest a deep learning strategy for decision support, based on a greedy algorithm. Decision making support by artificial intelligence is of the most challenging trends in modern computer science. Currently various strategies exist and are increasingly improved in order to meet practical needs of user-oriented platforms like Microsoft, Google, Amazon, etc.

Keywords

Machine Learning, Big Data Analysis, Decision Making, Artificial Intelligence, Computer Science, Tensorflow, Prediction

1. Introduction

Artificial intelligence becomes more and more powerful due to increasing development of neural algorithms, as well as computer technologies. Various techniques involve tools of artificial intelligence in text/object/face/voice/path recognition [1]-[11], big data analysis and prediction [13] [14], healthcare and diagnosis [12], flight control [15], market analysis and data mining [16] [17] [18], signal processing [19], and so on [20] [21]. See also references therein. In order to be capable to meet all the requirements from all areas of application, the existing methods of artificial intelligence are constantly improving, as well as new challenging methods are developed. Choice of a particular method or technique strongly depends on the specific problem that one aims to deal with. For instance, in deterministic problems it is more convenient to use symbolic or sub-symbolic methods, while in non-deterministic problems statistical methods are more suitable. One of the simplest techniques for studying non-deterministic systems uses so-called classifiers, functions using pattern matching for determi-

nation of closest matches, and controllers, functions choosing output of classifiers. For further details refer to [22] [23] [24].

The most known and used type of classifiers are the so-called neural networks and their various modifications, such as deep feedforward neural networks, recurrent neural networks, etc. Being called after human brain cells-neurons, those networks are “trained” to replicate the main function of neurons, i.e. generate specific outputs from incoming signals [25]. In general, artificial neural networks consist of neuron layers responsible for input information (input layer), main function of the network (is programmed in hidden layers) and output information (output layer), see **Figure 1**.

The main function of neurons at input layer is to transmit the input information to the hidden layers. Depending on specific problem under consideration there might be one or more hidden layers, the main function of which is to process the input information received from the input layer as required. The processed information is transmitted to the output layer, which converts this information into user acceptable format. All or some neurons may have so-called weights, by which the input information is accepted and is transmitted to proceeding neurons. In practice, those tools are used in a smart configuration to handle the problem.

Another important aspect of artificial neural networks is neurons learning, i.e. the algorithm modifying the neural network parameters, e.g. weights, interconnections, number and structure of hidden layers, etc., for derivation of required output from given input. For further details refer to [25].

Other examples of techniques of artificial intelligence include decision trees and its modifications such as regression trees, incremental decision trees, alternating decision trees, logistic model trees, etc. [26] [27].

One of the very promising and important fields of nowadays needs that can be supported by artificial intelligence is the so-called decision making support (see, for instance, [28] [29] [30] and references therein). Decision making is the process of selecting an option among several alternatives. This concept is widely used in contemporary management, financial operations, trade, bargain, etc. One of the challenging applications of decision making is in e-commerce, where an online user needs to choose an item from a really wide variety. Therefore, it is of great help to the e-shop to have such an artificial online agent that can suggest users items perfectly matching their search and preferences, based on their previous experience, that is searches, clicks, add-to-cards, keywords, etc. There exist several verified attempts to integrate artificial intelligence in decision making process [31] [32] [33] [34]. Moreover, currently such companies as Google, Amazon, etc., provide open access to their artificial intelligence based decision making support platforms¹. Despite the amount of existing references, there still exist some aspects and parts of artificial intelligence based decision making support that can be improved.

¹Refer to <https://github.com/tensorflow/tensorflow> for Tensorflow and <https://github.com/awsml/machine-learning-samples> for Amazon.

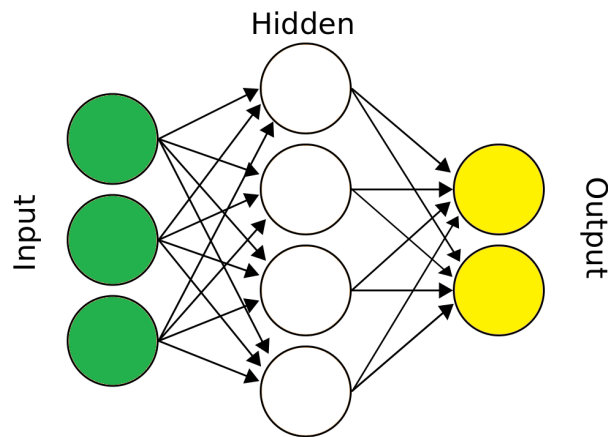


Figure 1. Schematic representation of artificial neural networks.

This paper is devoted to the study of possibilities to involve the well-known greedy algorithm at some special steps of neural network algorithm supporting online user decision making process. It should be noted that the greedy algorithm has been incorporated in artificial neural networks earlier by other authors. See, for instance, [35] [36] and references therein. The rest of the paper is organized as follows. We first describe the greedy algorithm in Section 2 and then show how specifically it can be used in neural computations in Section 2.1. Particular numerical computations show the efficacy of the algorithm in Section 3.

2. Greedy Algorithm

Greedy algorithm or search is an efficient tool that is usually applied in optimization problems. The main steps of all greedy algorithms are as follows:

- 1) Choice of a candidate set. The problem is divided into a finite set of sub-problems. At initial step a candidate set is arbitrarily chosen as a solution to the first sub-problem, so that the proceeding sub-problems are solved on the basis of this candidate set.
- 2) Choice of a selection function. A selection function is chosen to test what is the best candidate to be added to the solution.
- 3) Choice of a feasibility function. The chosen selection function involves a feasibility function, which first determines the set of candidates that can possibly contribute to the solution.
- 4) Choice of an objective function. This function allows to make choice of a candidate at each step in some sense optimal.
- 5) Choice of a solution function. Finally, a solution function is chosen appropriately to establish when a desired precision in solution approximation is achieved.

Greedy algorithms can be really efficient when dealing with large sets of data, in the sense that if the *globally* optimal solution of the problem consists of optimal solutions of *locally* optimal sub-problems, then greedy searches will find

the global solution in reasonable time. Nevertheless, in some specific problems, such as the famous traveling salesman problem, greedy algorithm may result in unique worst possible solution [37]. This failure comes from the fact that the greedy algorithm does not use all the data of the problem (recall step 1).

Note that the successful application of the algorithm, first of all, depends on success at every partial step, since any globally optimal solution consists of optimal sub-solutions. Moreover, since the selection function is different from identity, then each candidate can be involved in only one step of the algorithm, i.e. neither of previous steps (except directly previous one) plays a role in proceeding computations/searches. This is called greedy choice property and helps to save quite much computational cost.

Integration of Greedy Algorithm into Neural Model of User Behavior Prediction

Problem solving using neural computations in principle is a systematic search through given data in order to reach required solution. Therefore, from the description of the greedy algorithm above it follows that it can be efficiently involved in neural computing procedure to save computational time, especially when the analyzed data sets are quite large. However, using of greedy algorithm may *not* be a guarantee for optimality of found solution.

Let us demonstrate how a simple greedy algorithm can be involved in prediction of user behavior based on his/her clicks, search history, time spent on viewing certain items etc., within a specific online shop. Analysis of user behavior and prediction of his/her intentions are crucial for online sellers in order to e.g. make advertisement more targeted, recommend items that will be more likely bought, etc. Several approaches exist to involve artificial intelligence in real time prediction of user intentions including recent studies [38] [39] [40] and some references therein.

We are mainly interested in interactions between guaranteed purchase and

- 1) view of a product page,
- 2) view of the basket.

The set of training data (\mathcal{D}) is composed of

- 1) all sessions (\mathcal{S}) of all users,
- 2) all items (\mathcal{I}_s) that were displayed within session $s \in \mathcal{S}$,
- 3) all purchases (\mathcal{P}_s) corresponding to session $s \in \mathcal{S}$.

Thus, $\mathcal{D} = \mathcal{S} \cup \mathcal{I}_s \cup \mathcal{P}_s$. The set of sessions is split into two subsets: \mathcal{S}_p containing sessions with purchase and \mathcal{S}_{np} containing those without purchase (see Figure 2). Eventually,

$$\mathcal{D} = \mathcal{S}_p \cup \mathcal{S}_{np} \cup \mathcal{I}_s \cup \mathcal{P}_s.$$

The problem is to predict the set \mathcal{P}_s .

In this specific example, there exist $2.5 \cdot 10^4$ products with full description. Evidently, the data are very sparse and high dimensional (depend on many parameters), therefore their analysis is too costly. To explore any pattern between

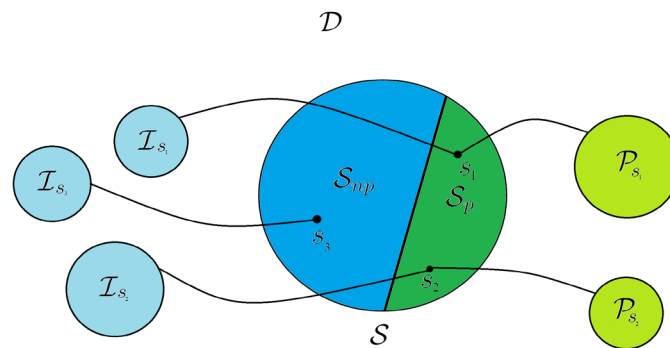


Figure 2. Schematic representation of the data set \mathcal{D} .

these data, we involve the neural network borrowed from [40] with greedy search integrated into the data analysis step in order to reduce the dimensionality. More specifically, at the initial step a candidate set is chosen heuristically. Then, a selection function

$$\sigma(s) = \chi_{\mathcal{D} \setminus (\mathcal{S}_p \cup \mathcal{S}_{np} \cup \mathcal{I}_s)}(s),$$

where $\chi_{\mathcal{D}}$ is the characteristic function of \mathcal{D} defined as follows:

$$\chi_{\mathcal{D}}(s) = \begin{cases} 1; & s \in \mathcal{D}, \\ 0; & \text{else.} \end{cases}$$

Thus, the selection function checks whether a chosen element s belongs to $\mathcal{D} \setminus (\mathcal{S}_p \cup \mathcal{S}_{np} \cup \mathcal{I}_s)$ or not. If it does, then it is out of consideration, otherwise it is a potential candidate for \mathcal{P}_s to be complemented by. Then, the objective function, chosen as the following convex functional (quadratic error function):

$$\kappa[s] = \frac{1}{n} \sum_{i=1}^n (s - s_i)^2,$$

where i is the iteration index, n is the amount of data points, s_i is the currently chosen element; allows to choose only the data which are (in the sense of quadratic error) close to the desired set.

The last step of the greedy algorithm defines the solution function as follows:

$$S(s) = \arg \min \kappa[s] = \arg \min \left[\frac{1}{n} \sum_{i=1}^n (s - s_i)^2 \right].$$

Apparently, for the above chosen simple objective function, the solution function, S , is easy to compute. However, more complicated forms of κ can be considered.

The mathematical background of the minimization problem is the same as in [40], therefore we will not bring it here to be concise. Refer to [40] for further details.

3. Numerics and Discussions

Numerical experiment shows that in the chosen particular case involvement of greedy algorithm allows to reduce the computation time significantly, compared

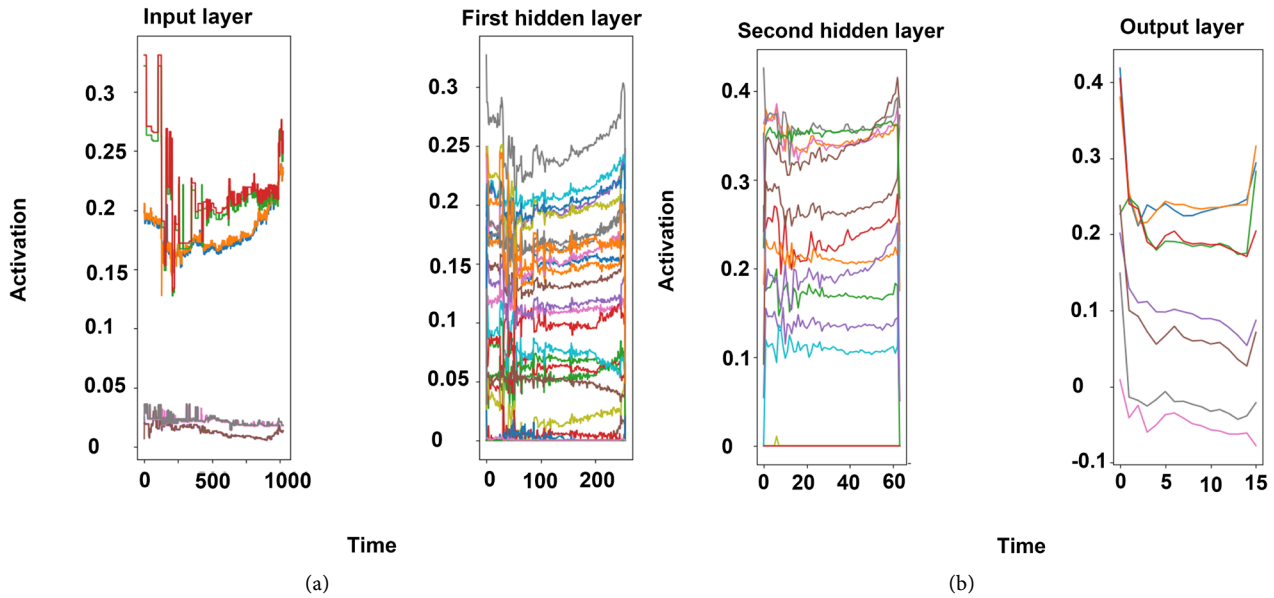


Figure 3. Activation vs Time in neural network layers.

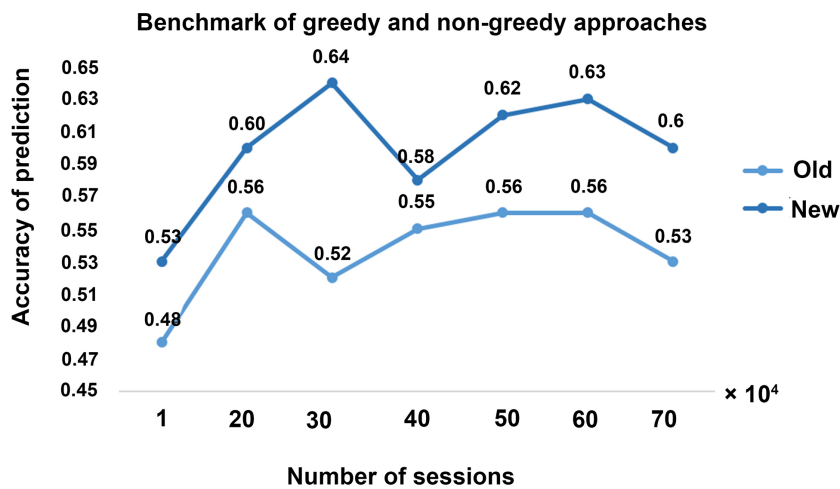


Figure 4. Benchmark of greedy and non-greedy approaches.

with [40]. Though, the relative error, i.e. the prediction error of the set \mathcal{P}_s of purchases within session $s \in \mathcal{S}_p$, is $\sim 2\% - 4\%$, which is pretty much acceptable. On Figure 3 the time spent on computations within input, first hidden layer, second hidden layer, and output. It is seen that for basic prediction of \mathcal{P}_s , only $t \approx 1400$ s, which is a good improvement as well.

Figure 4 shows the accuracy of prediction implemented using the proposed approach against number of sessions. It shows a good tendency to be applicable in time consuming predictions.

4. Conclusion

Aiming to reduce the complexity of computations (machinery time) in the problem of user behavior analysis, prediction and decision support based on

his/her online behavior, in this paper, we suggest to involve the well-known greedy algorithm. For heuristically chosen candidate set, we choose appropriate selection function, classifying the data at each iteration. Then, we choose a quadratic error function as objective function for classification. The output is used to construct the solution function. Numerical analysis of given data shows a significant decrease in computation time compared with the method without a greedy algorithm.

Acknowledgements

We express our sincere gratitude to both reviewers for their invaluable help in making the content and the presentation of the paper much better.

References

- [1] David, M., Powers, W. and Turk, Ch.C.R. (1989) Machine Learning of Natural Language. Springer-Verlag.
- [2] Tappert, C.C., Suen, C.Y. and Wakahara, T. (1990) The State of the Art in Online Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 787. <https://doi.org/10.1109/34.57669>
- [3] Tebelskis, J. (1995) Speech Recognition using Neural Networks. PhD Thesis for Doctor of Philosophy in Computer Science. School of Computer Science, Carnegie Mellon University, Pittsburgh.
- [4] Ovchinnikov, P.E. (2005) Multilayer Perceptron Training without Word Segmentation for Phoneme Recognition. *Optical Memory & Neural Networks (Information Optics)*, **14**, 245-248.
- [5] Huang, B., Zhang, Y. and Kechadi, M. (2009) Preprocessing Techniques for Online Handwriting Recognition. Intelligent Text Categorization and Clustering. *Studies in Computational Intelligence*, **164**, 25-45. https://doi.org/10.1007/978-3-540-85644-3_2
- [6] Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H. and Schmidhuber, J. (2009) A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**. <https://doi.org/10.1109/TPAMI.2008.137>
- [7] Holzinger, A., Stocker, C., Peischl, B. and Simoncic, K.-M. (2012) On Using Entropy for Enhancing Handwriting Preprocessing. *Entropy*, **14**, 2324-2350. <https://doi.org/10.3390/e14112324>
- [8] Qian, C., Zhu, Z.Y., Wang, R.H., Chen, C.E., Wang, X.G. and Tang, X.O. Deepid-Net: Multi-Stage and Deformable Deep Convolutional Neural Networks for Object Detection. <http://arxiv.org/abs/1409.3505>
- [9] Cornia, M., Baraldi, L., Serra, G. and Cucchiara, R. (2016) A Deep Multi-Level Network for Saliency Prediction. *23rd IEEE International Conference on Pattern Recognition (ICPR)*, 3488-3493. <https://doi.org/10.1109/ICPR.2016.7900174>
- [10] Mollahosseini, A., Chan, D. and Mahoor, M.H. (2016) Going Deeper in Facial Expression Recognition Using Deep Neural Networks. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1-10. <https://doi.org/10.1109/WACV.2016.7477450>
- [11] Riedl, C., Zanibbi, R., Hearst, M.A., Zhu, S., Menietti, M., Crusan, J., Metelsky, I. and Lakhani, K. (2016) Detecting Figures and Part Labels in Patents: Competi-

- tion-Based Development of Image Processing Algorithms. *International Journal on Document Analysis and Recognition*, **19**, 155-172.
<https://doi.org/10.1007/s10032-016-0260-8>
- [12] Hamscher, W., Console, L. and de Kleer, J. (1992) Readings in Model-Based Diagnosis. Morgan Kaufmann Publishers Inc., San Francisco.
- [13] Big Data, Analytics & Artificial Intelligence. The Future of Health Care Is Here. University of San Francisco, San Francisco.
- [14] IEEE Smart Grid Big Data Analytics, Machine Learning and Artificial Intelligence in the Smart Grid Working Group. Big Data Analytics, Machine Learning and Artificial Intelligence in the Smart Grid.
- [15] Farrell, J.A. and Polycarpou, M.M. (2006) Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches. Wiley, New York. <https://doi.org/10.1002/0471781819>
- [16] Wu, X. (2004) Data Mining: Artificial Intelligence in Data Analysis. *Proceedings IEEE/WIC/ACM International Conference on Web Intelligence*, Beijing, 24 September 2004, 7. <https://doi.org/10.1109/WI.2004.10000>
- [17] Fiol-Roig, G., Miró-Julià, M. and Isern-Deyà, A.P. (2010) Applying Data Mining Techniques to Stock Market Analysis. In: Demazeau, Y., *et al.*, Eds., *Trends in Practical Applications of Agents and Multiagent Systems*, Advances in Intelligent and Soft Computing, Vol. 71, Springer, Berlin, Heidelberg, 519-527.
- [18] Wierenga, B. (2010) Marketing & Artificial Intelligence: Great Opportunities, Reluctant Partners. *Marketing Intelligent Systems Using Soft Computing: Managerial and Research Applications*, **258**, 1-8. https://doi.org/10.1007/978-3-642-15606-9_1
- [19] Hu, Y.H. and Hwang, J.-N. (2001) Handbook of Neural Network Signal Processing. CRC Press, Boca Raton.
- [20] Jain, L.C. and Martin, N.M. (1998) Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications. CRC Press, Boca Raton.
- [21] Principe, J.C. (2000) Artificial Neural Networks. The Electrical Engineering Handbook. CRC Press LLC, Boca Raton.
- [22] Luger, G. and Stubblefield, W. (2004) Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 5th Edition, Benjamin/Cummings, San Francisco.
- [23] Neapolitan, R. and Jiang, X. (2012) Contemporary Artificial Intelligence. Chapman & Hall/CRC, Boca Raton.
- [24] Poole, D. and Mackworth, A. (2017) Artificial Intelligence: Foundations of Computational Agents. 2nd Edition, Cambridge University Press, Cambridge.
- [25] Nielsen, M.A. (2015) Neural Networks and Deep Learning. Determination Press.
- [26] Sumner, M., Frank, E. and Hall, M. (2005) Speeding Up Logistic Model Tree Induction. In: Jorge, A.M., Torgo, L., Brazdil, P., Camacho, R. and Gama, J., Eds., *Knowledge Discovery in Databases: PKDD 2005*, Lecture Notes in Computer Science, Vol. 3721, Springer, Berlin, Heidelberg, 1-8. https://doi.org/10.1007/11564126_72
- [27] Rokach, L. and Maimon, O. (2008) Data Mining with Decision Trees: Theory and Applications. World Scientific Pub. Co. Inc., Singapore.
- [28] Crunk, J. and North, M.M. (2007) Decision Support System and AI Technologies in Aid of Information Based Marketing. *International Management Review*, **3**, 61-86.
- [29] Phillips-Wren, G., Jain, L.C. and Ichalkaranje, N. (2008) Intelligent Decision Making: An AI Approach. Spring Publishing Company, Berlin.
<https://doi.org/10.1007/978-3-540-76829-6>

- [30] Fehrer, R. and Feuerriegel, S. (2015) Improving Decision Analytics with Deep Learning: The Case of Financial Disclosures.
- [31] Pomerol, J.-Ch. (1997) Artificial Intelligence and Human Decision Making. *European Journal of Operational Research*, **99**, 3-25. [https://doi.org/10.1016/S0377-2217\(96\)00378-5](https://doi.org/10.1016/S0377-2217(96)00378-5)
- [32] Phillips-Wren, G. and Jain, L. (2006) Artificial Intelligence for Decision Making. In: Gabrys, B., Howlett, R.J. and Jain, L.C., Eds., *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science, Vol. 4252, Springer, Berlin, Heidelberg, 531-536. https://doi.org/10.1007/11893004_69
- [33] Torra, V., et al. (2012) Modeling Decisions for Artificial Intelligence. *Proceedings of MDAI: International Conference on Modeling Decisions for Artificial Intelligence*, Barcelona, 20-22 November 2013.
- [34] Stalidis, G., Karapistolis, D. and Vafeiadis, A. (2015) Marketing Decision Support Using Artificial Intelligence and Knowledge Modeling: Application to Tourist Destination Management. *Procedia—Social and Behavioral Sciences*, **175**, 106-113. <https://doi.org/10.1016/j.sbspro.2015.01.1180>
- [35] Howard, A.G. (2013) Some Improvements on Deep Convolutional Neural Network Based Image Classification. CoRR, abs/1312.5402.
- [36] Bengio, Y., et al. (2007) Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, **19**, 153-160.
- [37] Algorithm, G. (2001) *Encyclopedia of Mathematics*. Springer Science + Business Media B.V./Kluwer Academic Publishers, Berlin.
- [38] Curme, C., Preis, T., Stanley, H.E. and Moat, H.S. (2014) Quantifying the Semantics of Search Behavior before Stock Market Moves. *Proceedings of the National Academy of Sciences*, **111**, 11600-11605. <https://doi.org/10.1073/pnas.1324054111>
- [39] Zhang, M., Chen, G. and Wei, Q. (2015) Discovering Consumers' Purchase Intentions Based on Mobile Search Behaviors. *Advances in Intelligent Systems and Computing*, **400**, 15-28.
- [40] Vieira, A. (2015) Predicting Online User Behaviour Using Deep Learning Algorithms.