AUTOMATED MACHINE LEARNING OVERVIEW

Roman BUDJAČ[1], Marcel NIKMON[1], Peter SCHREIBER[1],
Barbora ZAHRADNÍKOVÁ[1], Dagmar JANÁČOVÁ[2]

[1]SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA,
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA,
INSTITUTE OF APPLIED INFORMATICS, AUTOMATION AND MECHATRONICS,
ULICA JÁNA BOTTU Č. 25, 917 24 TRNAVA, SLOVAK REPUBLIC
e-mail: roman.budjac@stuba.sk, marcel.nikmon@stuba.sk,
zahradnikova.barbora@gmail.com, peter.schreiber@stuba.sk

[2]TOMAS BATA UNIVERSITY OF ZLIN, FACULTY OF APPLIED INFORMATICS,
DEPARTMENT OF AUTOMATION AND CONTROL ENGINEERING,
NAD STRÁNĚMI 4511, 760 05 ZLÍN, CZECH REPUBLIC
e-mail: janacova@utb.cz

Abstract

This paper aims at deeper exploration of the new field named auto-machine learning, as it shows promising results in specific machine learning tasks e.g. image classification. The following article is about to summarize the most successful approaches now available in the A.I. community. The automated machine learning method is very briefly described here, but the concept of automated task solving seems to be very promising, since it can significantly reduce expertise level of a person developing the machine learning model. We used Auto-Keras to find the best architecture on several datasets, and demonstrated several automated machine learning features, as well as discussed the issue deeper.

Key words

INTRODUCTION

In the modern digital industry, it is a well-known fact that predictive data analytics has a powerful and positive impact on many areas, such as predictive maintenance, product failure detection, and other fields in revealing a hidden patterns, forecasting energy consumption, predicting machine failures, predicting future outcomes, predicting trends in data collected in the production level of automation and many more. In order to achieve a high level of accuracy, machine learning modeling is expected to be used in the production where higher degree of

automation is implemented. Designing a well performed machine learning data model presumes large data collections, pre-processing, building machine learning model and deploying the final model [1]. The requirement of collecting a huge amount of data from every part of production process results in the importance of the topics such as Big data and Business intelligence in Industry 4.0.

GENERAL IDEA BEHIND THE AUTOMATION MACHINE LEARNING TASK

Manual approach to machine learning assumes the whole model development pipeline to be made by a human machine learning expert. Automated machine learning approach is about to minimize human participation in building models instead of developing one's own model architecture, and in tuning the parameters aiming at the best results for the solved problem [2]. This change in perception can open the door for potential customers without previous programming experience, and for utilizing the ML (Machine Learning) potential to support their own business. Big technological companies bring their own products to distribute automated machine learning algorithms, for example, Azure automated machine learning, Google Cloud AutoML. These services promise to create high-performance models with limited machine learning expertise and State-of-the-art performance [3]. On the other hand, there are free or open source solutions also with high performance, e.g. Auto-SKLEARN, AutoWeka [5] and AutoKeras, Auto Pytorch, Teapot, and more. Although all these toolkits promising, all of options listed above are limited when applied and either work only with certain structure, or they are suitable for only certain types of machine problems, e.g. image classification, natural language processing tasks and speech recognition [6]. In general, they all share the same goal: to find an effective learning algorithm and to optimize parameters of this model to the values that would achieve the best convergence in problem solving [5]. General auto ML algorithm is shown in Figure 1.
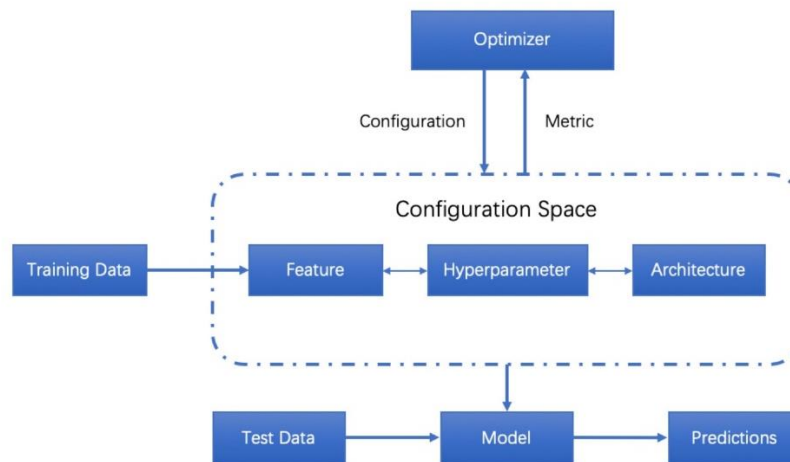

Figure 1 *Auto-machine learning algorithm [17]*

AUTO-MACHINE LEARNING RECENT WORK – FRAMEWORKS

In the recent years, a number of concepts have been presented to improve previous solutions and tools for the implementation of automatic machine learning. In the next section, we will explore some of the existing open source solutions, briefly characterize them, present their approach to finding a solution to the automated machine learning tasks and the benefits they bring.

Optimizing hyperparameters

The problem of the automated machine learning in the past used to be solved by optimizing hyperparameters. This solution assumed that we have a chosen algorithm A with parameter α located in the area Θ where the target corresponds to finding the value of α parameter found in Θ and minimizing the function $l(\alpha) = L(A \alpha)$ [7]. Another option that proved to be effective was the method where instead of one single parameter, many model hyperparameters were selected synchronously, using the gradient of the model selection [8]. More advanced approach is represented by a Bayesian method of optimizing hyperparameters; it is based on the Gaussian process operating correctly on small numerical hyperparameters , SMAC based on random forests and Tree Parzen Estimator, which are suitable for optimizing the high-dimensional hyperparameters in deep neural networks [9]. The aim of the Bayesian optimization is to compose a probabilistic model and then use it to decide how the next function shall be evaluated. The principle is to apply the information gathered from previous estimation, which enables locating the minimum of non-convex complicated function to optimize hyperparameters [10].

Neural Architecture Search

The main idea behind a neural architecture search is to automatically design a neural network architecture as best as possible without any intervention of machine learning experts. Searching for a well-performed architecture could be power- and time-consuming, whereas many recent papers utilized only a limited number of hyperparameters (number of layers, nodes, activation function and optimizing algorithm). Mobeen et al. [11] present used functional modules which can be aggregated to a larger complex network [11].

Mostly in academic papers, two types of optimization algorithm for searching the problem space are dominantly present. Firstly, applying the reinforcement learning able to generate convolutional neural network architecture from scratch by using the gradient method is often applied. In this method, the accuracy is awarded and then used to compute the gradient to update the controller. This controller is autoregressive. That means, it sets the new hyperparameters at the same time [12]. Another approach is not based on starting with "guessing" and composing a network from scratch. Instead, it can use one of the pre-trained network architecture to achieve the top result. Alternatively, genetic algorithms might be applied. Genetic algorithms are heuristically searching strategies used to solve optimization problems inspired by evolutionary theory. In this case, the optimal neural network structure is found by scoring each one, over the value of network accuracy hyperparameter. The top rank received by the network is processed to set up new generation until maximum generations are achieved. Generated network architecture trained by algorithm is called a child network [12].
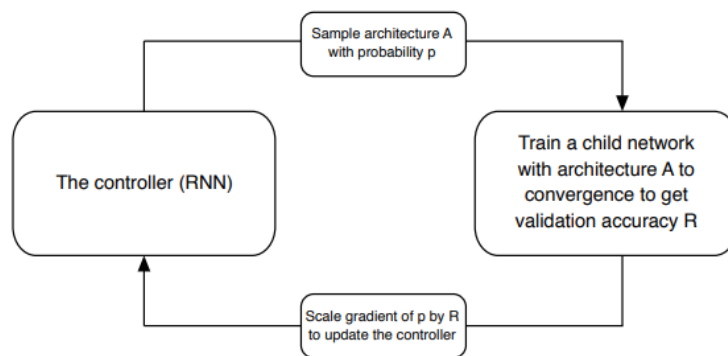

Figure 2 Training child network [17]

109

Implementation toolkits

In this section, we present the existing tools implementing certain sort of automated machine learning. These are open source and free to use, we decided not to present commercial products although they are holding powerful environments to develop automated machine learning, but are not widely available.

AutoKeras

It is a software library developed by DATALab utilized to create an accessible deep neural network with the minimal expertise knowledge [14]. To develop the software, Keras library and Python programming language are used. Autokeras works only with Python 3 and only with version 3.6. None of the other versions (newer or later released) are supported. Using Auto-Keras locally on its own machine instead of need to configure Docker and Kubernetes in the cloud is its main benefit. In future release, wider search space to recurrent neural network, solving image segmentation tasks and object detection tasks are planned [15]. As we can see in Figure 2, the searcher is built by the state-of-the-art Bayesian Optimizer and Gaussian Process.
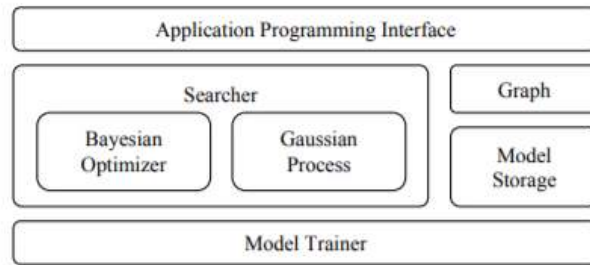

Figure 3 *Layers of Auto-Keras module [13]*

Autoweka

Autoweka is an open source package using the auto-machine learning implemented in a user-friendly interface (GUI) extended by 10 meta methods (AdaBoost, Additive Regression etc.), base learners (Naive Bayes, Random Forest, Linear Regression, Gaussian process etc.), and a hyperparameter selector [5]. User selects learning algorithm and sets the hyperparameters based on own intuition. AutoWeka contains several modes to work: GUI, command line, Experimenter and Knowledge flow. One of the main advantages is that AutoWeka is very well documented software. AutoWeka is extremely useful to quickly compare differences between various learning algorithms applied on dataset.

METHODOLOGY OF EXPERIMENTS AND RESULTS

For the experiment, we used several tools to apply automated machine learning. Firstly, we used Keras with Python 3.6 and Auto-Keras library. For the training set, we preferred public datasets. We chose the MNIST dataset containing 60 thousand handwritten numerals in a training set and 10 thousand for the test set. The dataset is divided into 10 classes. Each image is grayscale 28x28 pixels. Secondly, we applied auto-machine learning on Fashion MNIST containing 60 thousand clothes images in a training set and 10 thousand for the test set. Each image is grayscale 28x28 pixels in 10 classes. These two datasets were trained on GPU Nvidia1060. On MNIST dataset using Autokeras, we reached accuracy 96.7%, fashion MNIST reached 92.1%. CIFAR-10 was trained on Nvidia K80 in the cloud service, and training time was 12 hours with accuracy of 91.2%.

Table 1 Auto machine learning process - results				
	MNIST	**FMNIST**	**CIFAR10**	
Auto-Keras	96.7%	92.1%	91.2%	

DISCUSSION

In this paper, we made an overview of automated machine learning, presented a basic idea, and explained the reasons why it is essential for the artificial intelligence community. After experimenting with Auto-Keras, the results were very promising. This success has consequences in the form of the computer power consumption and time limit. Time limit is the time representing how long Auto-Keras searched the space; default is set to 24 hours. Depending on these two variables, we can achieve different accuracy on the dataset. In the case of CIFAR-10, we can expect the accuracy of above the 95 % if increasing the searching time to 24 or more hours. The whole process of constructing architecture, running a network and optimizing the solution may take a lot of time. If the search space is robust with many parameters, this fact also affects the training time significantly. Currently, this is the biggest weakness of the automated machine learning.

CONCLUSION

In this article, we introduced automated machine learning. The topic appeared a few years ago and has been growing very fast since then. The idea to automate optimizing hyperparameters in a neural network model or even construct a deep neural network by scratching, training, testing and optimizing the ML solution, as well as changing the architecture, can achieve a new degree of automation in artificial intelligence in the future.

References

[1] J.MCHUGH, P. E. CUDDIHY, J. W. WILLIAMS, K. S. AGGOUR, V. S. KUMAR, V. MULWAD. 2017. "Integrated access to big data polystores through a knowledge-driven framework," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, pp. 1494-1503. ISBN 978-1-5386-2715-0

[2] https://cloud.google.com/solutions/hpc/, High performance computing, [Online]. [Accessed: 09-2019] Available at: https://cloud.google.com/solutions/hpc/

[3] https://cloud.google.com/, Cloud AutoML [Online]. [Accessed: 09-2019] Available at: https://cloud.google.com/automl/docs/

[4] TUGGENER, L., AMIRIAN, M., ROMBACH, K., LÖRWALD, S., VARLET, A., WESTERMANN, CH., STADELMANN, T. 2019. Automated Machine Learning in Practice:

State of the Art and Recent Results. In: *6th Swiss Conference on Data Science (SDS)*: Bern, Switzerland, pp. 31-36. ISBN 978-1-7281-3105-4.

[5] THORNTON, C., HUTTER, F., HOOS, H. H., LEYTON-BROWN, K. 2013. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. In: *19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*: Chicago, Illinois, USA, pp. 847-855. ISBN 978-1-4503-2174-7

[6] FEURER, M., KLEIN, A., EGGENSPERGER, K., SPRINGENBERG, J., BLUM, M., HUTTER, F. 2015. Efficient and Robust Automated Machine Learning,. In: *Advances in Neural Information Processing Systems 28*, pp. 2962–2970.

[7] DOMHAN, T., SPRINGENBERG, J. T., HUTTER, F. 2015. Speeding up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves.

In: *24th International Conference on Artificial Intelligence*: Buenos Aires, Argentina, pp. 3460-3468. ISBN: 978-1-57735-738-4

[8] Y. BENGIO. 2000. Continuous optimization of hyper-parameters. In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, Italy, pp. 305-310 vol.1. ISBN 0-7695-0619-4

[9] BERGSTRA, J. S., BARDENET, R., BENGIO, Y. & KÉGL, B. 2011. Algorithms for Hyper-Parameter Optimization. In: *Advances in Neural Information Processing Systems 24* (eds. Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. & Weinberger, K. Q.) 2546–2554 .Curran Associates, Inc., 2011. ISBN: 978-1-61839-599-3

[10] SUN, L. et al. Automatic Neural Network Search Method for Open Set Recognition. 2019. FUJITSU R & D CENTER , Beijing , China Fujitsu Laboratories Ltd ., Kawasaki , Japan. *2019 IEEE Int. Conf. Image Process.* 4090–4094. ISBN 978-1-5386-6250-2

[11] M. AHMAD, M. ABDULLAH AND D. HAN. 2019. A Novel Encoding Scheme for Complex Neural Architecture Search. In: *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, JeJu, Korea (South), pp. 1-4. ISBN 978-1-7281-3271-6

[12] JAAFRA, Y., LUC LAURENT, J., DERUYVER, A. & SABER NACEUR, M. 2019. Reinforcement learning for neural architecture search: A review. *Image Vis. Comput.* Image and Vision Computing, **89**, pp. 57-66. ISSN 0262-8856

[13] ZOPH, B. & LE, Q. V. 2016. Neural Architecture Search with Reinforcement Learning. 1–16.

[14] B. ZOPH, V. VASUDEVAN, J. SHLENS AND Q. V. LE. 2018. Learning Transferable Architectures for Scalable Image Recognition. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, pp. 8697-8710. ISBN 978-1-5386-6420-9

[14] HAIFENG JIN, QINGQUAN SONG, AND XIA HU. 2019. Auto-keras: An efficient neural architecture search system. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.

[15] KOTTHOFF, L., THORNTON, C., HOOS, H. H., HUTTER, F. & LEYTON-BROWN, K. 2017. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *J. Mach. Learn. Res.* **18**, 1–5.

[16] THORNTON, C., HUTTER, F., HOOS, H. H. & LEYTON-BROWN, K. 2013. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In: *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* Part F128815, 847–855.

[17] https://github.com/hibayesian/awesome-automl-papers, What is AutoML? [Online]. [Accessed: 09-2019] Available at: https://github.com/hibayesian/awesome-automl-papers