AutoKeras

Jan Kruszewski, Damian Skowroński, Tomasz Krupiński
Warsaw University of Technology

June 2, 2022

Abstract

placeholder

1 Introduction

Machine learning has developed significantly in recent years. This has resulted in a demand for specialists in this field. Hence the development of a derivative of machine learning - automatic machine learning (AutoML). It is a solution to the problem of time-consuming model building and the shortage of machine learning specialists.

In this article we will discuss the AutoKeras package which is one of the AutoML tools. To explore its functionality we will use datasets describing binary classification problems. Based on the analysis of the results we will see whether the use of our package brings satisfactory results.

2 References

- "An Open Source AutoML Benchmark" by Gijsbers - AutoML benchamrk on different AutoML packages. In our benchmark we use same datasets.
- article - article on the above benchmark.
- AutoKeras website - site containing informations used in description of the framework.

2.1 Descripion of the framework

In this paragraph, we will briefly describe the AutoKeras framework.

2.2 Authors

AutoKeras is an AutoML framework based on the Keras package, developed by a group of people associated with Texas AM University. The main idea of the developers was to create a package that would give a chance to use machine learning to people even without much experience in this field. The package has an extensive and clear documentation.

2.3 Division of AutoKearas into modules

The framework is divided into 4 main modules.

- The Searcher module, is responsible for searching the neural architecture. It uses Bayesian optimisation and a Gaussian process. The search algorithms use the CPU.
- Neural networks are trained in the Trainer module.
- Graph is a module that processes computational graphs that Searcher controls to morph the network.
- The last module is Storage. It is responsible for saving the final models. Due to their size they are not stored in RAM.

2.4 What distinguishes AutoKeras?

A couple of benefits set AutoKeras apart from other AutoML frameworks. There is no need to use cloud services, which are not free and not affordable for anyone who wants to use machine learning. AutoKeras also ensures data security and privacy. It is also worth mentioning that AutoKeras focuses on deep learning problems unlike frameworks like SMAC, TPOT, Auto-WEKA or Auto-Sklearn which focus on shallow models.

2.5 API

The inteface of AutoKeras has been created in a similar way to Sklearna. It can be built in 3 lines of code using the constructor and the fit and predict methods. It is worth noting that it has two levels 'task-level' for users with less knowledge of the system and 'search-level' for advanced users who control preprocessing and neural network architecture themselves. In general, AutoKeras offers us the following models: ImageClassifier, ImageRegressor, TextClassifier, TextRegressor, StructuredDataClassifier, Structured DataRegressor and AutoModel. The framework has also many user-friendly features.

2.6 Preprocessing

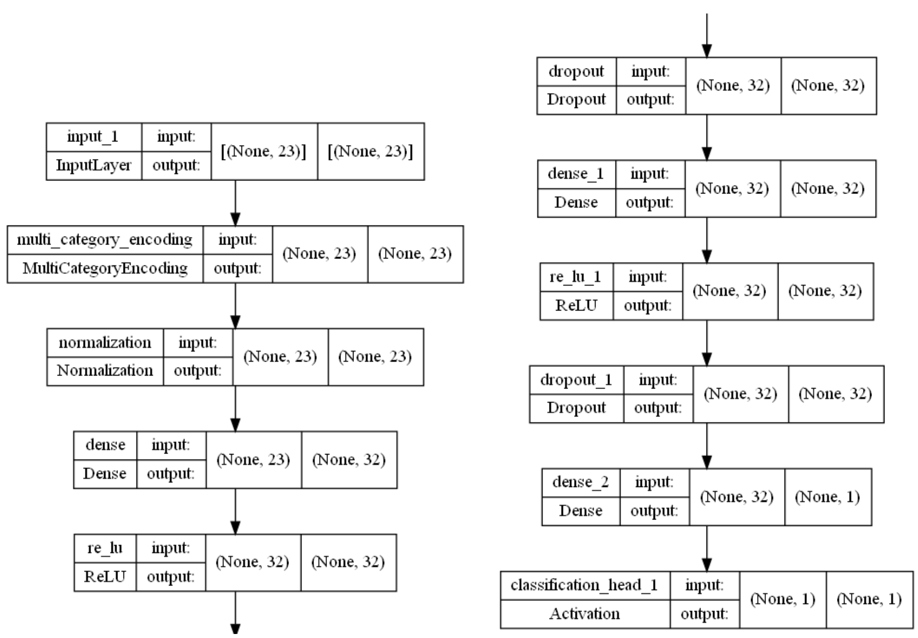
Package does the necessary preprocessing for us before training. Hence, our function does not have any additional code snippets devoted to this activity. AutoKeras performs for us the basic vectorisation, data cleaning and normalisation, which is an important step for neural networks.

2.7 Use of resources

AutoKeras is designed to take full advantage of the CPU, GPU and RAM, where are stored only currently important information. The rest of the data is stored in other places such as the hard drive.

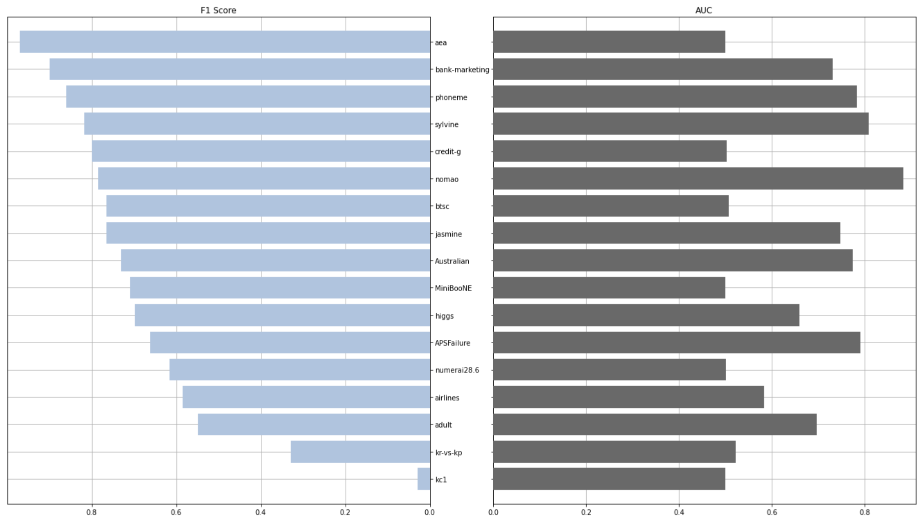
3 Results of the evaluation

3.1 Example model



The example model, whose diagram is shown above, was selected after twenty searches for the best model. The model starts with basic layers such as the input layer, the encoding layer and the normalisation layer. Next are mainly dense layers, with a linear function and others with a ReLU function. As it was a classification task, it ends with a classifier layer. Thus, it is not a very complex architecture. Nevertheless, this architecture fulfils its purpose.

3.2 Result of benchmark



We tested our solution on seventeen collections available on the Internet, used in already mentioned Open Source AutoML Benchmark. We presented two metrics on the above graphic, namely F1 and AUC. As you can see, F1 is not correlated with AUC at all. For the vast majority of collections, F1 and AUC gave better results than random models. The average F1 score is 0.72.

3.3 Result of benchmark – group competition

Team	AUC	set	framework
KTR	0.7912	valid	FLAML
Gakubu	0.7908	valid	AutoGluon
Gakubu	0.7852	valid	Own implementation
WTF	0.7835	valid	AutoPytorch
Moja grupa	0.7815	valid	AutoPytorch
Moja grupa	0.7808	valid	Own implementation
Moja grupa	0.7789	valid	Autosklearn
WTF	0.7768	valid	Own implementation
KTR	0.7617	valid	Own implementation
Tojada	0.7469	valid	AutoKeras

Also, each autoML group ran a test on one particular set to be able to compare the quality of the pipelines made to create predictions from tabular data. It is worth mentioning that not every implementation created neural networks. One could therefore intuitively conclude that neural networks would have the best results, obviously compensating for the training time compared to, for example, regular algorithms. However, it turned out that AutoKeras achieved the worst results of all the other frameworks, indicating that not everything can be done with neural networks.