AutoDL Challenge Design and Beta Tests-Towards automatic deep learning

Zhengying Liu, Olivier Bousquet, André Elisseeff, Sergio Escalera, Isabelle Guyon, Julio Jacques, Adrien Pavao, Danny Silver, Lisheng Sun-Hosoya, Sebastien Treguer, et al.

AutoDL Challenge Design and Beta Tests
- Towards automatic deep learning

Zhengying Liu[1,*], **Olivier Bousquet**[2], **André Elisseeff**[2], **Sergio Escalera**[3,7]
Isabelle Guyon[1,3], **Julio Jacques Jr.**[7], **Adrien Pavao**[1],
Danny Silver[6], **Lisheng Sun-Hosoya**[1], **Sebastien Treguer**[5],
Wei-Wei Tu[4], **Jingsong Wang**[4], **Quanming Yao**[4]
[1]U. Paris-Saclay (UPSud/INRIA), France, [2]Google Zürich, Switzerland, [3]ChaLearn, US,
[4]4Paradigm, Beijing, China, [5]La Paillasse, France, [6]U. Acadia, Canada,
[7]U. Oberta de Catalunya, Spain, U. of Barcelona, Spain, and CVC, Spain

Abstract

Following the success of the first AutoML challenges , we designed a new challenge called AutoDL. We target applications such as speech, image, video, and text, for which deep learning (DL) methods have had great success in the past few years. All problems will be multi-label classification problems. We hope to drive the community to work on finding fully automatic ways of designing DL models. Raw data will be provided (no features extracted). The source of the datasets and the type of data will be concealed, but the data structure will be revealed. All datasets will be formatted in a uniform tensor manner, to encourage participants to submit generic algorithms (not necessarily constrained to DL). We will impose restrictions on training time and resources to push the state-of-the-art further. We will provide a large number of pre-formatted public datasets and set up a repository of data exchange to enable meta-learning. In this paper, the challenge protocol and baseline results are presented to seek community feed-back. The challenge is planned for 2019, but the exact schedule is not announced yet.

1 Motivation

Despite recent successes of deep learning [11], designing good neural networks remains difficult and requires practical experience and expertise. Selecting the architecture and hyper-parameters, and training a complex deep network is often a long and frustrating trial-and-error process involving lots of heuristics. The acceleration of the demand for deep learning solutions naturally gives rise to the need for improving the automation of the design of deep learning solutions. Many approaches have been proposed to address this problem. Earlier work used neuro-evolution strategies [14] inspiring more recent methods using *evolutionary algorithms* [1, 17, 19]. However, methods of this kind are known to scale poorly and may over-fit. *Bayesian optimization* methods provide another possibility but they also have scaling issues when the dimension (number of hyper-parameters) is high [20, 9]. Lastly, ideas borrowed from *reinforcement learning* have recently been applied for this problem [24, 2]. But almost all of these methods require huge computational resources (e.g. GPU) accessible only to big companies and laboratories.

The proposed methods vary a lot in terms of capacity and complexity, and may use different settings, resources, data, metrics, etc. Thus, based alone on the performances reported in the literature, it is difficult to form an opinion of the relative merit of these various approaches. To fairly evaluate all these methods and help advance the state of the art in this emerging research area, it is necessary to devise standard benchmarks. This motivates the organization of this AutoDL challenge.

[*]Corresponding author **zhengying.liu@inria.fr**. The authors are in alphabetical order of last name, except the first author.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

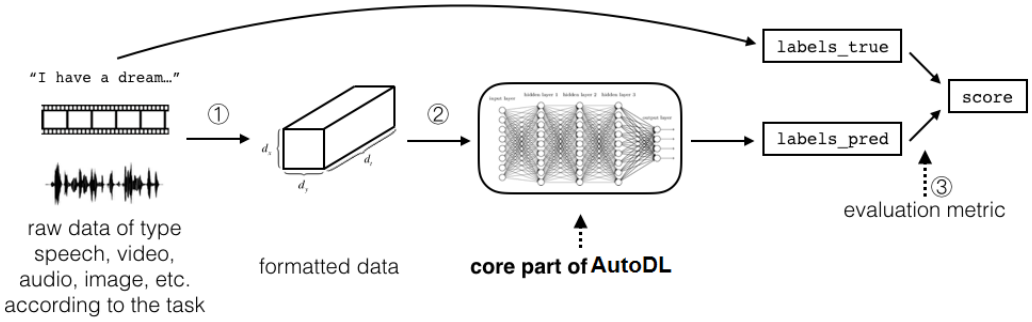

Figure 1: **Workflow of AutoDL challenge tasks.**

2 Challenge Protocol

The main differences between the AutoDL challenge and prior AutoML challenges [6, 7] are:

1. **Raw data:** Data are no longer pre-processed in a uniform feature vector representation; they include all data types representable as spatio-temporal sequences. We will use a generic data format called TFRecords , used by TensorFlow [2]. This format allows us to format any 3D+time data, including text, speech, image, video, etc.
2. **Large scale datasets:** For development, datasets will all be under 2GB (after compression), for practical reasons (See Section 4), however, for final testing, datasets of hundreds of thousands of examples will be used.
3. **Any-time learning:** The metric of evaluation (based on learning curves, see Section 3) will force the participants to provide algorithms, which can be stopped at any time (not known in advance), but the participants will be informed of the maximum time budget.
4. **Fixed resource learning:** The participants will be given limited memory and computational resources to run their code. They will be informed of resources made available to them (number of cores, memory, etc.). Their (compressed) code size will be limited to 300 MB.
5. **Uniform tasks and metrics:** All problems will be multi-label classification problems and be evaluated with the same metric (see Section 3).

One key aspect of this challenge, and other past AutoML challenges [8] we organized, is that it is a **code submission** challenge. The participants will submit code that will be trained and tested on the challenge platform without any human intervention, on datasets they will never see. **Pre-training** will be allowed, provided that the submission size limit is respected. The challenge will be run in two phases:

- **Feed-back phase:** During a **development period** lasting 3 to 4 months, feed-back will be immediately provided on a leaderboard, using **practice datasets** (invisible to the participants, but visible to their submitted code). A large number of **public datasets** ($\simeq 100$), in the same format as the practice datasets, will be made available for method development and to encourage research on **meta-learning**.
- **Final test phase:** After the challenge deadline, the code of the participants will be **blind tested** on some new **final evaluation datasets** that were never disclosed before to the public.

In the AutoDL challenge, we have therefore 3 types of datasets: **public**, **practice**, and **final**. In contrast with some previous AutoML challenges [6, 7] in which the practice data was distributed to the participants (except for the target values of the validation and test sets), in the AutoDL challenge, neither the practice datasets nor the final evaluation datasets will be exposed to the participants directly in any of the challenge phases: their code will be fully blind tested. However, a large number of fully labeled "public" datasets will be provided, and we will set up a **public repository** so participants can exchange among themselves other datasets. A **starting kit** in Python with a TensorFlow interface will be provided. We will give several examples of sample submissions, including some calling scikit-learn and other libraries. It will be possible to call e.g. PyTorch

[2] `https://www.tensorflow.org/`

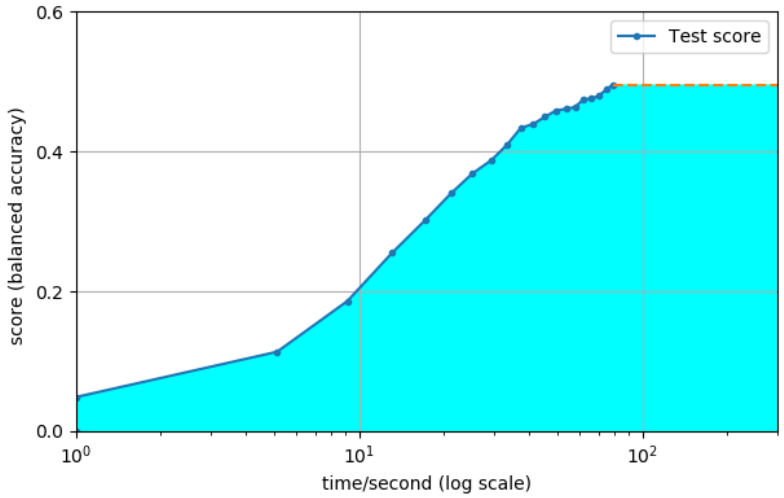

Figure 2: **Example of learning curve.** We modified the Codalab platform so participants can save their results, at any intervals they choose, to incrementally improve their performance, until the time limit is attained. In this way, we can plot their learning curves: performance as a function of time. By evaluating them with the area under the learning curve, we push them to implement any-time learning methods. The x-axis will be on a log scale. This shows an example in which the participant could have avoided saving so frequently towards the end of the curve. The algorithm was stopped before the time limit was reached.

3 Challenge metrics

One novelty in the AutoDL challenge is to enforce any-time learning by scoring participants with the **Area under the Learning Curve** (ALC) (Figure 2). The participants can train in increments of a chosen duration (not necessarily fixed) to progressively improve performance, until the time limit is attained. This allows us to plot their learning curves: "performance" as a function of time. We compute ALC by the trapeze method, as a function of log(time), where "time" is the cumulative time of training and testing. We compute a separate learning curve and ALC for each dataset and rank the participants according to ALC. The overall ranking is made by **averaging ranks** over all datasets.

For each dataset, a "performance" point on the learning curve is the average performance over all the labels of the task. For each label individually, we compute the Balance ACuracy BAC = (1/2) (TPR + TNR). To that end, the participants are asked to provide a normalized score between 0 and 1, analogous to a probability, which we threshold at 0.5. We may use that score to compute various other metrics, including the Area under the ROC curve (AUC). We stress that we do not consider multi-class classification metrics. Each label is scored independently.

4 Datasets and baseline methods

The datasets are still under formatting. However, we alreary formatted ten datasets (5 public datasets and 5 practice datasets) to run **beta-tests**, see Table 1. Between now and NIPS 2018, We will be running a rehearsal of the challenge, in the form of a **hackathon** lasting a few days, open internally at Google. The hackathon will be organized using the CodaLab platform, of which a cloned instance was created running on Google Cloud

In contrast with the real AutoDL challenge (which will take place in 2019), the hackathon will have a **single phase** during which participants will get immediate feed-back on five **practice datasets** when they submit their code, without having access to them (for that reason, we do not give any detail on such datasets here). There will be **no final testing** on separate datasets. But, similarly to the real AutoDL challenge, the participants' submitted code will be trained and tested automatically on the challenge platform, without any human intervention. We set up CodaLab so all five datasets can be processed in parallel.

Table 1: **Hackathon data summary statistics.** We formatted 10 datasets for the hackathon. The public datasets are shown. Similar practice datasets have been formatted, but remain hidden. For the real AutoDL challenge, we plan to format $\simeq 100$ public datasets and set-up a repository of data exchange, in the format of the challenge

#	Dataset	Domain	XY locality	Label number	Sample number train	Sample number test	Tensor dimension row	Tensor dimension col	Tensor dimension time
1	adult	tabular	No	3	39074	9768	1	24	1
2	katze	video	Yes	6	1528	863	120	160	181
3	starcraft	speech	No	10	21115	2567	1	1	16000
4	tweet	text	No	20	11314	7532	1	50	473
5	munster	image	Yes	10	60000	10000	28	28	1

For this hackathon, resources will be limited to **2 hours of CPU time**. Preliminary baseline are found in Table 2. We compare those results with literature reference performances. Work still remains to be done to improve baseline methods or adjust the tasks to make them doable within two hours.

Table 2: **Baseline results.** Benchmark results on public datasets for planned hackathon, obtained by running our starting kit and collected from the literature (*) Performances are in AUC for adult, tweet (20-newsgroup), and munster (MNIST), from the top ranking participants of the AutoML challenge [8, 3]. The others are reported in multi-class accuracy. AUC and BAC are normalized as in the AutoML challenge betwen 0 and 1.

	#	Dataset	Domain	Literature Performance (*)	Literature References	Starting Kit Accuracy	Starting Kit AUC	Starting Kit BAC
Public	1	adult	tabular	0.77	[22, 10, 8, 3]	-	0.72	0.41
	2	katze	video	0.90	[16, 23, 18]	0.54	0.26	0.263
	3	starcraft	speech	0.79	[5]	0.47	0.95	0.42
	4	tweet	text	0.98	[21, 15, 8, 3]	0.61	0.82	0.57
	5	munster	image	0.99	[13, 12, 8, 3]	0.99	0.99	0.99
Private	1	*hidden*	tabular	0.90	*hidden*	0.85	0.83	0.52
	2	*hidden*	video	- never used -	*hidden*	0.68	0.41	0.41
	3	*hidden*	speech	0.88	*hidden*	0.42	0.93	0.36
	4	*hidden*	text	0.88*	*hidden*	0.76	0.66	0.57
	5	*hidden*	image	0.84	*hidden*	0.47	0.96	0.47

We briefly describe the baseline methods employed. For **tabular** data, we applied auto-sklearn for 2h [4]. For **image** data, we create a self-scaling convolutional neural network (SCNN): We repeatedly apply 2D convolution layers (with kernel shape 3×3) followed by 2D max-pooling (of shape 2×2) until the row number or the column number is equal to 1, doubling the number of filters after each iteration (the initial value is 32). Then a dense hidden layer is applied, followed by a Dropout layer of rate 0.5. Finally, we apply a dense output layer, consisting of sigmoid units individually trained by log-loss (no softmax since we have multi-label problems). For **video** data, we only have a preliminary baseline consisting of a fully connected neural network applied to frames averaged over time. For **text** data, text is encoded as a string of numbers (one for each word) then passed to a multi-layer neural network. The first layer performs word embedding. It is followed by two dense layers. For **speech** data, we compute power spectrograms with FFT. Then, we apply a SCNN, similarly to that of image tasks, followed by two dense layers.

5 Conclusion and further work

We presented the design of a new challenge that will stimulate the AutoML community to embrace deep learning and tackle the hard problems of automatic architecture design and hyper-parameter search for models trained directly on raw data. By making available a large number of pre-formatted public datasets, and encouraging data exchange by setting up a repository, we hope to stimulate research in meta-learning. We have run baseline methods and are getting ready to start a full challenge rehearsal to verify the feasibility of the tasks, given the allotted time and computational resources. The results will be reported at NIPS and feed-back from the community will be seeked.

Acknowledgements

This challenge would not be possible without the work of many people, including contributors to the challenge protocol, starting kit, and datasets: Stephane Ayache (AMU, France), Mahsa Behzadi (Google, Switzerland), Baiyu Chen (UC Berkeley, USA) Mehreen Saeed (FAST Nat. U. Lahore, Pakistan), Michele Sebag (U. Paris-Saclay; CNRS, France), Jun Wan (Chinese Academy of Sciences, China), Hugo Jair Escalante (INAOE, Mexico). The challenge is running on the Codalab platform administered by CKCollab LLC with primary developers Eric Carmichael and Tyler Thomas. Google is the primary sponsor of the challenge. Other institutions of the co-organizers provided in-kind contributions.

References

[1] F. Assunção, N. Lourenço, P. Machado, and B. Ribeiro. Evolving the topology of large scale deep neural networks. In *European Conference on Genetic Programming*, pages 19–34. Springer, 2018.

[2] B. Baker, O. Gupta, N. Naik, and R. Raskar. Designing Neural Network Architectures using Reinforcement Learning. *arXiv:1611.02167 [cs]*, Nov. 2016. arXiv: 1611.02167.

[3] Codalab. AutoML challenge platform. the datasets "adult" and "dorothea" were used in round 0. the target values of "adult" were race, race (white v.s. other), and income. *https://competitions.codalab.org/competitions/2321* , 2015.

[4] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2962–2970. Curran Associates, Inc., 2015.

[5] Google Brain. Tensorflow speech recognition challenge. *https://www.kaggle.com/c/tensorflow-speech-recognition-challenge*.

[6] I. Guyon, K. Bennett, G. Cawley, H. J. Escalante, S. Escalera, Tin Kam Ho, N. Macia, B. Ray, M. Saeed, A. Statnikov, and E. Viegas. Design of the 2015 ChaLearn AutoML challenge. pages 1–8. IEEE, July 2015.

[7] I. Guyon, I. Chaabane, H. J. Escalante, S. Escalera, D. Jajetic, J. R. Lloyd, N. Macià, B. Ray, L. Romaszko, M. Sebag, A. Statnikov, S. Treguer, and E. Viegas. A Brief Review of the ChaLearn AutoML Challenge: Any-time Any-dataset Learning Without Human Intervention. In *Workshop on Automatic Machine Learning*, pages 21–30, Dec. 2016.

[8] I. Guyon and et. al. *Analysis of the AutoML Challenge series 2015-2018*. 2018.

[9] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter. Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, volume 54 of *Proceedings of Machine Learning Research*, pages 528–536. PMLR, Apr. 2017.

[10] R. Kohavi. Scaling up the accuracy of naive bayes classifiers: a decision tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207, 1996.

[11] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Y. LeCun, C. Cortes, and C. Burges. MNIST. *http://yann.lecun.com/exdb/mnist/*.

[14] K. O. S. a. R. Miikkulainen. Evolving Neural Networks Through Augmenting Topologies. 2002.

[15] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[16] R. of human actions database. KTH dataset. *http://www.nada.kth.se/cvap/actions/*.

[17] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. Le, and A. Kurakin. Large-Scale Evolution of Image Classifiers. *arXiv:1703.01041 [cs]*, Mar. 2017. arXiv: 1703.01041.

[18] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[19] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *CoRR*, abs/1712.06567, 2017.

[20] K. Swersky, J. Snoek, and R. P. Adams. Freeze-thaw bayesian optimization. *arXiv preprint arXiv:1406.3896*, 2014.

[21] UCI repository. 20-newsgroup dataset. *https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups*.

[22] UCI repository. Adult dataset. *https://archive.ics.uci.edu/ml/datasets/adult*.

[23] W. Yang, Y. Wang, and G. Mori. Human action recognition from a single clip per action. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 482–489. IEEE, 2009.

[24] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.