

厦门大学MAC实验室联合深度赋智在AutoML竞赛获亚军

在近期的ECML PKDD 2019中举办的AutoML类型竞赛AutoCV2中，[厦门大学MAC实验室](#)联合深度赋智取得了feedback阶段第一名，blind test单项第一，总成绩第二名的佳绩

比赛介绍

AutoCV2比赛属于NeurIPS AutoDL系列比赛，由Google，第四范式等在ECML PKDD 2019中举办。比赛的任务是视频/图像的多分类，参赛选手仅需要提交代码，代码会在后台的评测系统上自动进行训练，训练数据集对选手完全未知，需要选手的算法能够完全适应不同的数据集，并且在最短的时间内取得尽可能高的测试精度。

比赛共分为两个阶段，第一个阶段是feedback，参赛选手的代码在后台运行结束后会给出相应的学习曲线的反馈，用于用户调试自己的算法，第二个阶段是blind test，将选手最后提交的代码在另外5个完全未知的数据集进行测试，并且给出最终的结果。

比赛的任务为视频/图像的多分类，并要求每个数据集上的运行时间不超过20分钟。评测指标为每个数据集的学习曲线下的面积（ALC），具体计算方式如下：

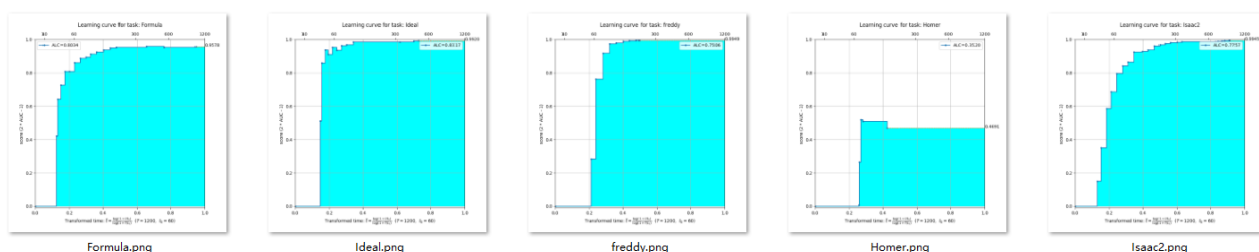
- 在每一个时刻 t ，计算 $s(t) = \text{Norm AUC} = 2 * \text{AUC} - 1$ ， $s(t)$ 是对时间 t 的函数
- 将时间归一化到 $[0,1]$ 区间内，其中的 $T = 1200s$ 为上限， t_0 是参考时间，默认为60s

$$\tilde{t}(t) = \frac{\log(1 + t/t_0)}{\log(1 + T/t_0)}$$

- 然后将 $s(t)$ 在归一化后的时间上进行积分得到ALC

$$\begin{aligned} ALC &= \int_0^1 s(t) d\tilde{t}(t) \\ &= \int_0^T s(t) \tilde{t}'(t) dt \\ &= \frac{1}{\log(1 + T/t_0)} \int_0^T \frac{s(t)}{t + t_0} dt \end{aligned}$$

最终我们在feedback阶段的学习曲线如下图



在计算完所有数据集的ALC之后，取所有队伍的平均排名作为选手的最终分数，我们最终取得了feedback第一，blind test单项第一，总成绩第二名的成绩

All datasets

Dataset 1

Dataset 2

Dataset 3

Dataset 4

Dataset 5

Download CSV

#	User	Entries	Date of Last Entry	<Rank> ▲	Dataset 1 ▲	Dataset 2 ▲	Dataset 3 ▲	Dataset 4 ▲	Dataset 5 ▲	Compute Time	Detailed Results
1	tanglang	60	08/17/19	2.6000	0.8317(1)	0.7506(4)	0.3520(3)	0.7757(1)	0.8034(4)	0:51:20	Learning Curve
2	kakaobrain	15	08/19/19	3.6000	0.8229(2)	0.7516(3)	0.3843(2)	0.7064(5)	0.7661(6)	0:42:55	Learning Curve
3	team_zhaw	16	08/19/19	4.4000	0.6808(3)	0.7907(1)	0.2957(6)	0.6131(7)	0.7762(5)	1:38:04	Learning Curve
4	kvr	7	08/18/19	6.0000	0.3350(10)	0.7431(5)	0.1932(11)	0.7648(2)	0.8072(2)	1:01:59	Learning Curve
5	Letrain	19	08/18/19	6.2000	0.4785(8)	0.7530(2)	0.3402(4)	0.5746(10)	0.7384(7)	1:38:19	Learning Curve
6	automi_freiburg	36	08/19/19	6.4000	0.4794(7)	0.7343(7)	0.5087(1)	0.5928(8)	0.6281(9)	0:55:53	Learning Curve
7	myelinio	1	07/25/19	6.6000	0.0612(12)	0.7316(8)	0.2296(9)	0.7624(3)	0.8084(1)	0:59:40	Learning Curve
8	ether	6	08/12/19	7.2000	0.0143(13)	0.7385(6)	0.2074(10)	0.7580(4)	0.8063(3)	0:59:58	Learning Curve
9	upwind_flys	8	08/19/19	7.8000	0.6747(4)	0.4116(11)	0.3240(5)	0.5837(9)	0.5820(10)	1:17:39	Learning Curve
10	OsbornArchibald	5	07/18/19	8.6000	0.5945(5)	0.6458(9)	0.2914(7)	0.5309(11)	0.5265(11)	1:43:19	Learning Curve

AutoCV2 Final Phase Results																
User	submissions	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	<Rank>	Max 1	Max 2	Max 3	Max 4	Max 5	<Max Rank>	Final Max Rank	Final Rank	
kakaobrain	6271 6385 6505 6655 6661 6667 6745 6751 6757	0.6277±0.0628 (8)	0.9048±0.0517 (5)	0.4076±0.0139 (8)	0.4640±0.0443 (2)	0.2091±0.0122 (3)	5.2	0.6963 (8)	0.9276 (1)	0.4206 (7)	0.5067 (2)	0.2217 (3)	3.8	1	1	
tanglang	6619 6625 6631 6817 6823 6829 6835 6841 6847	0.6231±0.0449 (9)	0.8406±0.0461 (11)	0.4527±0.0270 (4)	0.3688±0.0260 (6)	0.2363±0.0130 (1)	6.2	0.6635 (8)	0.8772 (10)	0.4734 (3)	0.4105 (5)	0.2507 (1)	5.4	2	2	
kvr	6283 6397 6517 6709 6715 6721 6799 6805 6811	0.6835±0.0299 (2)	0.9115±0.0150 (3)	0.4658±0.0083 (2)	-0.0417±0.0060 (16)	0.1627±0.0120 (8)	6.2	0.7174 (2)	0.9226 (2)	0.4778 (2)	-0.0289 (17)	0.1810 (7)		6	4	2
DX10808	6373 6493 6607 6691 6697 6703 6781 6787 6793	0.6499±0.0268 (6)	0.8673±0.0094 (8)	0.3560±0.0158 (11)	0.3702±0.0235 (5)	0.2223±0.0159 (2)	6.4	0.6763 (7)	0.8800 (9)	0.3690 (11)	0.4056 (6)	0.2403 (2)		7	6	4
ether	6307 6427 6541 6673 6679 6685 6763 6769 6775	0.6756±0.0301 (5)	0.9086±0.0065 (4)	0.4700±0.0065 (1)	-0.0430±0.0088 (18)	0.1711±0.0153 (6)	6.8	0.7043 (4)	0.9181 (5)	0.4781 (1)	-0.0255 (15)	0.1979 (4)	5.8	3	5	
Hana.Inst.Tech	6379 6499 6613	0.6815±0.0180 (3)	0.9194±0.0018 (1)	0.4640±0.0067 (3)	-0.0550±0.0082 (19)	0.1523±0.0087 (9)	7	0.7005 (5)	0.9213 (3)	0.4688 (5)	-0.0463 (19)	0.1623 (9)	8.2	8	6	
myelinio	6301 6421 6535 6637 6643 6727 6733 6739 6853	0.6774±0.0370 (4)	0.8829±0.0657 (8)	0.4491±0.0255 (5)	-0.0420±0.0082 (17)	0.1724±0.0114 (5)	7.4	0.7114 (3)	0.9207 (4)	0.4702 (4)	-0.0276 (16)	0.1853 (6)	6.6	5	7	
Letrain	6289 6403 6523	0.4684±0.0020 (14)	0.8406±0.0075 (10)	0.4030±0.0028 (9)	0.3972±0.0222 (3)	0.1865±0.0021 (4)	8	0.4707 (14)	0.8489 (12)	0.4047 (10)	0.4147 (4)	0.1878 (5)		9	9	8
team_zhaw	6277 6391 6511	0.5418±0.0340 (10)	0.8355±0.0915 (12)	0.4110±0.0072 (7)	0.3970±0.0298 (4)	0.1677±0.0052 (7)	8	0.5776 (10)	0.9006 (7)	0.4166 (8)	0.4178 (3)	0.1734 (8)	7.2	7	8	
automi_freiburg	6295 6415 6529	0.1836±0.0022 (20)	0.9138±0.0021 (2)	0.4009±0.0079 (10)	0.5169±0.0404 (1)	0.1031±0.0070 (14)	9.4	0.1856 (20)	0.9158 (6)	0.4066 (9)	0.5494 (1)	0.1111 (13)	9.8	10	10	
accheng	6331 6451 6565	0.6912±0.0366 (1)	0.7757±0.0486 (14)	0.4455±0.0080 (6)	-0.0075±0.0019 (14)	0.0543±0.0128 (17)	10.4	0.7331 (1)	0.8098 (13)	0.4547 (6)	-0.0053 (14)	0.0648 (17)	10.2	11	11	
mmadadi	6361 6481 6595	0.6376±0.0149 (7)	0.8540±0.0040 (9)	0.2131±0.0039 (14)	0.2712±0.0276 (8)	0.0907±0.0052 (15)	10.6	0.6548 (9)	0.8582 (11)	0.2176 (14)	0.2970 (8)	0.0958 (15)	11.4	13	12	
upwind_flys	6313 6433 6547	0.5220±0.0459 (11)	0.7933±0.0030 (13)	0.3523±0.0040 (12)	0.2862±0.0196 (7)	0.1358±0.0179 (11)	10.8	0.5630 (11)	0.7967 (14)	0.3567 (12)	0.3036 (7)	0.1519 (11)		11	12	13

算法简介

我们采用在ImageNet pretrain的ResNet18作为预测模型，为了加速训练过程，我们采用FP16（半精度浮点数）进行训练和测试，算法的核心在于如何在未知数据集的情况下制定最优的数据前处理策略，训练策略，验证策略，数据增强策略以及如何处理各种训练异常情况。

数据前处理策略

训练数据集分为图像/视频两种，为了加速视频数据的处理，我们直接将视频数据在时间维度上求平均，用二维卷积来进行处理，相比三维卷积可以在精度损失很小的情况下获取极大的加速。为了减小整体的FLOPS，我们对于所有的输入图像进行resize，并设置最大的resize尺寸为64。为了加速IO，我们选择在数据集较小的时候直接将整个数据集预先加载进显存，这样可以大幅减小IO时间并且提升训练时的gpu利用率。

训练策略

学习率的调整策略在训练过程中是至关重要的，然而对于未知的数据集我们需要更加灵活的训练策略来根据训练情况及时作出反应。我们这里使用了warm-up，并根据训练loss来及时对学习率进行衰减。

验证策略

在这个比赛中，选择何时进行测试是很重要的，如果测试太频繁则会减小训练的时间，影响最后的精度，测试太少则会取得较小的ALC，因此需要谨慎的选择进行测试的时间。而是否应该进行测试取决于是否在验证集上有较好的性能，如果每一次训练后都在验证集上进行验证则会浪费很多时间，这里我们设置了一定的限制条件来决定是否在这次训练结束之后进行验证，包括训练集精度是否足够/训练时间是否过长/是否进行了连续验证等。

数据增强策略

在数据增强中我们选择使用fast autoaugmentation，通过random search的方式来寻找最适合当前数据集的数据增强方式，并且仅在后期训练精度已经足够高的时候才进行数据增强。

异常情况处理

对于一些训练中的异常情况，比如训练发散/过拟合等，我们都设置了相应的handler来处理这些异常情况，可以保证面对不同的数据集不会有明显的短板。