论文笔记：Image Dataset Classification Difficulty Estimation

原创 Zerg_Wang 于 2020-04-26 17:46:09 发布 321 收藏 1 编辑 版权

分类专栏： Machine Learning 文章标签： dataset deep learning 深度学习

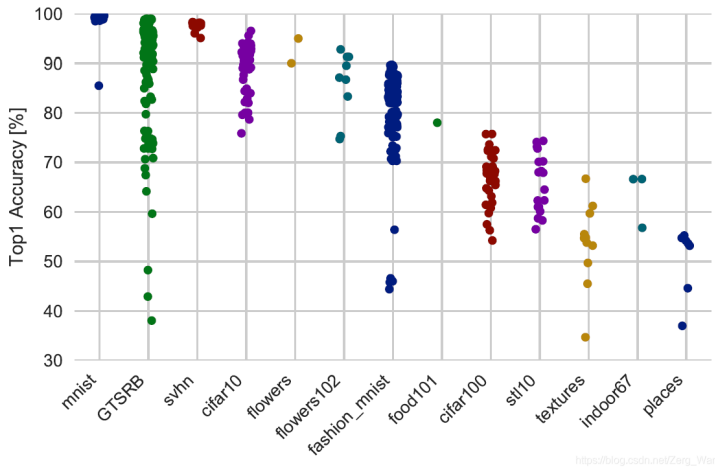
仅个人理解，欢迎指正，

论文题目

Efficient Image Dataset Classification Difficulty Estimation for Predicting Deep-Learning Accuracy

地址：https://arxiv.org/pdf/1803.09588v1.pdf

背景



上图来自论文，展示了13个数据集在不同算法下所达到的精度，每个点代表一种算法，这些算法包括普通的CNN，也有非深度学习类的算法，如SVM和随机森林等。在作者看来，完善的数据集（如Cifar、mnist）或者一些比赛所使用的数据集（如GTRSB），会有较多的算法应用及结果。但对于新的数据集，或因各种原因遭到"冷遇"的数据集，被各类算法验证的情况就会少很多。例如图中各类 flower 分类数据集。而对于新数据集，例如 fashion_mnist，即使数据集规模、样本尺寸、类别数量都与 mnist 相似，在使用与 mnist 同样的配置的算法进行实验后，得到的结果仍然是不够理想的。（结果分布不均，且普遍低于 mnist ），若要在 fashion_mnist 达到较好精度，针对该数据集的单独优化必不可少。
然而，针对新数据集进行特定的算法和网络来寻找最好的训练配置与超参数（如网格搜索、随机搜索、贝叶斯优化等方法），一来需要定义搜索空间，二来搜索代价高，效率低下。对此，作者提出了三种针对分类数据集，简单有效地评估分类任务难度的方法。作者认为，该方法给分类任务的"评分"，可以作为神经架构搜索和超参数优化任务的先验，从而缩小搜索空间，进而达到提高整个分类任务效率，并优化最终结果的目的。

判断数据集的分类难度，主要思想还是判断数据集各个类别之间的相似度，同类样本之间越相似，异类样本之间差异越大，则分类难度越低，分类精度越高，以下的三种算法，基本是基于这个思想的。

符号介绍

在介绍这三种算法前，先明确各类符号：

数据集：$D = (X_{train}; y_{train}; X_{test}; y_{test})$

训练样本：$X_{train} \in R^{n_{train} * d}$

测试样本：$X_{test} \in R^{n_{test} * d}$

训练样本的标签：$y_{train} \in [1; C]^{n_{train}}$

测试样本的标签：$y_{test} \in [1; C]^{n_{test}}$

数据集类别量，单样本数据维度：C, d

训练样本量，测试样本量：n_{train}, n_{test}

模型（包括网络拓扑、超参数以及相关增强操作等）：M

模型M在数据集D上的深度学习实验及该实验的 Top-1 结果：$(D; M), Top-1(D; M)$

数据集得分：$r(D)$

算法一：Silhouette Score

该算法注重比较数据集中同类样本间的相似度以及不同样本之间的区分度。

输入样本：i

与 i 同一类别的所有样本与 i 的平均欧式距离：$a(i)$

与 i 最相近类别的所有样本与 i 的平均欧式距离：$b(i)$

样本 i 的 Silhouette Score 为：

$$s(i) := \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i). \end{cases}$$

某一类别的 Silhouette Score，为该类中所有样本的 Silhouette Score 的平均值。而整个数据集的 Silhouette Score 分数，为该数据集中所有样本 Silhouette Score 的平均值。

Silhouette Score 范围在 (-1, 1)之间，分数越高，表示数据集同类样本越相似，异类样本差异越大。（聚类效果越好）

该算法的时间复杂度：$O(\bar{d}n^2)$，其中$\bar{d}$为计算两个样本之间欧氏距离的时间，n为样本数量。

由上式可知，该算法的效率，主要取决于$\bar{d}$，如果所有样本以原尺寸进行 Silhouette Score 的计算，一方面$\bar{d}$的计算开销无疑是巨大的；另一方面，在原尺寸下，样本噪音会对结果产生较大影响，难以准确衡量样本的相似性。对此，作者提出了几种优化方案：

TABLE I
PIPELINES USED TO COMPUTE THE SILHOUETTE SCORE ON DATASETS

Score	Transformation	$\bar{d}$	Distance	Speedup
S_1	None	d	MSE	31.3×
S_2	None	d	DSSIM	1.0 (Ref)
S_3	Resize image	8^2	MSE	48.4×
S_4	Resize image	8^2	DSSIM	1.3×
S_5	PCA	10	MSE	72.8×
S_6	Autoencoder	1000	MSE	6.4×

S1：MSE（均方误差）即上文提到的相似性计算算法。

S2：SSIM（The Structural Similarity Index，结构相似性指数），用于判断两图像的相似性，结果为一个0到1之间的数，越大表示图像越相似，当值为1时，两图像相同。而DSSIM为结构差异性指数。

$$DSSIM(x, y) = \frac{1 - SSIM(x, y)}{2}$$

S3、S4分别为S1、S2中样本resize到 8 * 8 大小所进行的计算。

S5：利用主成分分析法（Principal Component Analysis，PCA）将样本降维至10。

S6：作者采用了一个Resnet-50 网络作为编码器，该网络权重为在ImageNet 上预训练的网络权重。样本进入网络，在应用非线性层之前，通过最后一个全连接层，从而转换成维度为1000的图像特征向量。

算法二：K-means Clustering

普通深度学习训练的时间复杂度：$O(c(M)n_{train}e))$，其中e Blank Equation代表epoch，$c(M)$取决于网络，越复杂的网络，该值越大。

由上式可知，深度学习时间复杂度随着 n 线性增长，但 Silhouette Scores算法时间复杂度的增长却是n平方级别的。在数据量较大的情况下，Silhouette Scores算法效率显然太低。

对此，作者提出使用 K-means Clustering 的方法（对于所有输入样本，通过该算法进行分类，获得标签，然后与样本真实标签对比），并通过以下指标进行评估：

completeness（完整性）：对于某一类别，原本属于该类别的样本都被分到旗下，则满足完整性要求。该指标类似于召回率。

homogeneity（同质性）：对于某一类别，分到该旗下的样本都是正确的，则满足同质性要求。类似于正确率。

v-measure：同质性和完整性的加权平均。

adjusted mutual information（调整互信息）与adjusted rand index（ 调整兰德系数）：均用于衡量聚类情况，取值范围[-1, 1]，值越大表示分类情况越好。

标签的对应

除了以上指标，作者还提出了自己的指标，希望通过标签对应情况来判断数据集的类别区分度。

通过K-means Clustering 算法后，预测得到的类别数和实际的类别数是一致的（它们一一对应），对于我们而言自然是知道它们的对应关系，但K-means分类器可能不知道，对此，通过某些算法来对应这些标签，之后得到的结果可以用于衡量分类器对于该数据集各类别区分度的“理解程度”，进而判断数据集类别区分度。

那如何对应呢？朴素的想法是通过排列组合，然后选择预测的标签分布与所对应的“实际”分布差距最小的一种（在这种排列下，混淆矩阵的迹是最大的，即矩阵主对角线上元素之和是最大的），但考虑到计算开销，作者提出了“贪心”的策略：

对于实际类别C1，其下的样本可能被预测为K0、K1……我们选择其中最多样本被预测为的那一类，则该类别与C1对应。以此类推。如果出现无法一一对应的情况，作者设置了一个上限参数，值为7。如果无法一一对应的标签数量在7以内，则对于这部分标签进行排列组合的穷举（最大尝试次数为 7! = 5040），对于多出来的标签，则默认采用初始的排列。

该方法简单高效，且在一定程度上保证了预测的准确性的下限。

算法三：Probe Nets

或者，也可以采用一个预定义的神经网络，通过在数据集上的训练得到的精度作为数据集的“得分”。作者称该网络为“Probe Net”。为了提高该算法的适用性及效率，Probe Net需满足：（1）适用于任何图像分类数据集；（2）训练速度足够快；（3）经过若干epoch后便停止训练，即使此时还未完全拟合。

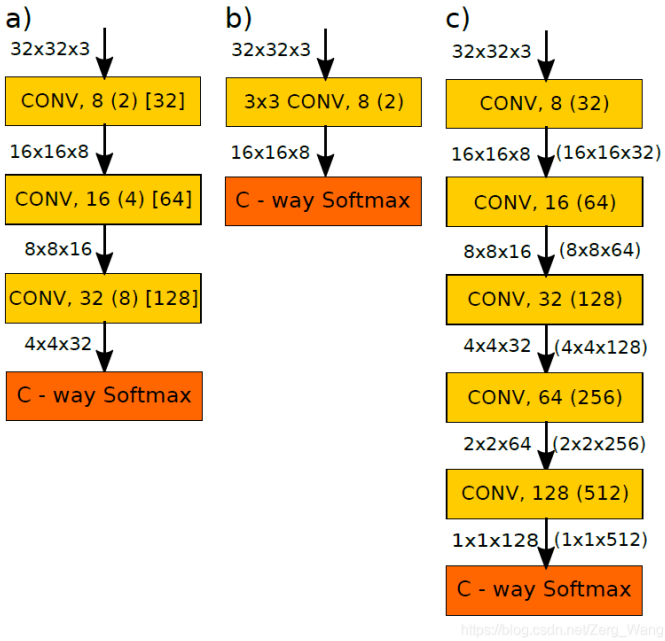
对于Probe Net的结构，作者建议建造两种不同的结构：（1）静态网络，其内部拓扑结构是固定的；（2）动态网络，会根据数据集的类别数而缩放其拓扑结构。

静态网络

静态Probe Net分为三种，regular、shallow、deep。其中regular型的网络有三个卷积层，每层后接批量归一化层、2*2的max pooling层以及ReLU。第一个卷积层采用8个卷积核，之后每层卷积核数量翻倍。（对于regular型网络，还有2个变种，narrow型和wide型，前者第一层的卷积核数为2而后者的第一层卷积核数为32）

shallow和deep与regular类似，前者比regular少两个卷积层，后者比regular多两个卷积层。

随着每层翻倍的卷积核数量，经过最后一层卷积核后，张量形状与根据类别数确定的Softmax层不符，此外，deep和shallow版本的网络参数量也有很大的不同，对此，作者提出了标准化版的deep和shallow网络，标准化后的网络输出层的参数与regular网络一致。



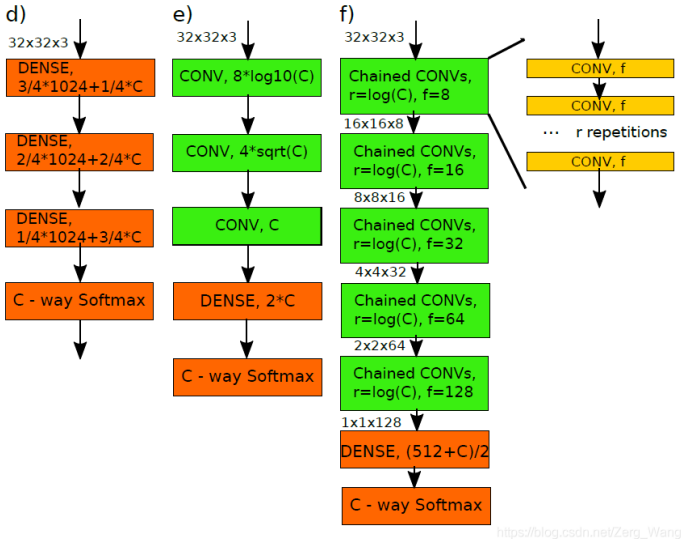
上图展示了静态Probe Net的结构，图a为regular网络及其narrow（网络参数在小括号中）和wide的变种（参数在方括号中），图b、c分别为shallow和deep网络，括号内为normalized版的网络参数。

动态网络

图d展示的网络为MLP（多层感知器），其中间的隐藏层会根据类别数C而调整。

图e展示的网络，其卷积层中的滤波器（filter）的数量会随着C调整，从而导致不同的filter深度。

图f展示的网络，会根据C调整卷积次数。



操作次数与网络参数量

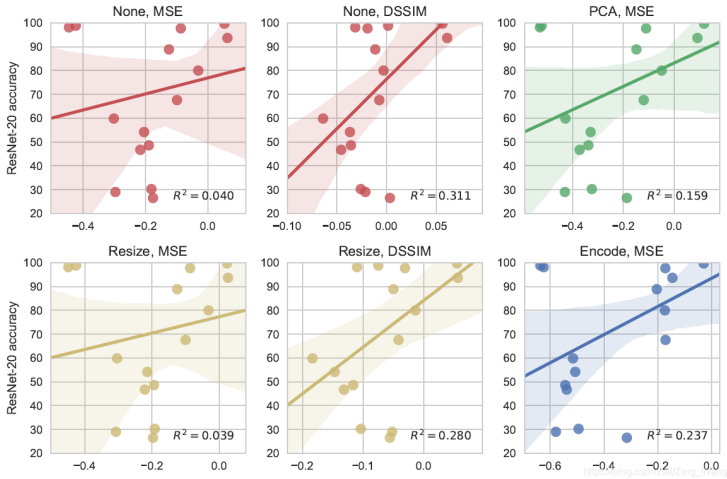
TABLE II
OPERATION COUNT AND NUMBER OF PARAMETERS OF PROPOSED PROBE NETS

Probe Net	$C = 10$		$C = 100$	
	OPs	Weights	OPs	Weights
Regular	0.81M	11K	0.86M	57.5K
Narrow	0.09M	2K	0.10M	13K
Wide	10.34M	114K	10.52M	299K
Shallow	0.24M	21K	0.42M	205K
Shallow norm.	0.06M	5K	0.10M	51K
Deep	1.40M	100K	1.41M	112K
Deep norm.	19.76M	1576K	19.81M	1622K
MLPs	2.90M	2908K	3.10M	3107K
Kernel depth	0.53M	6K	4.56M	384K
Length	1.41M	118K	4.39M	338K
ResNet-20	40.55M	271K	40.56M	277K

实验结果

使用相同条件对15个数据集进行训练，通过分析精度与通过上述方法得到的分数之间的关系，可知上述评分算法的情况。

Silhouette Score算法

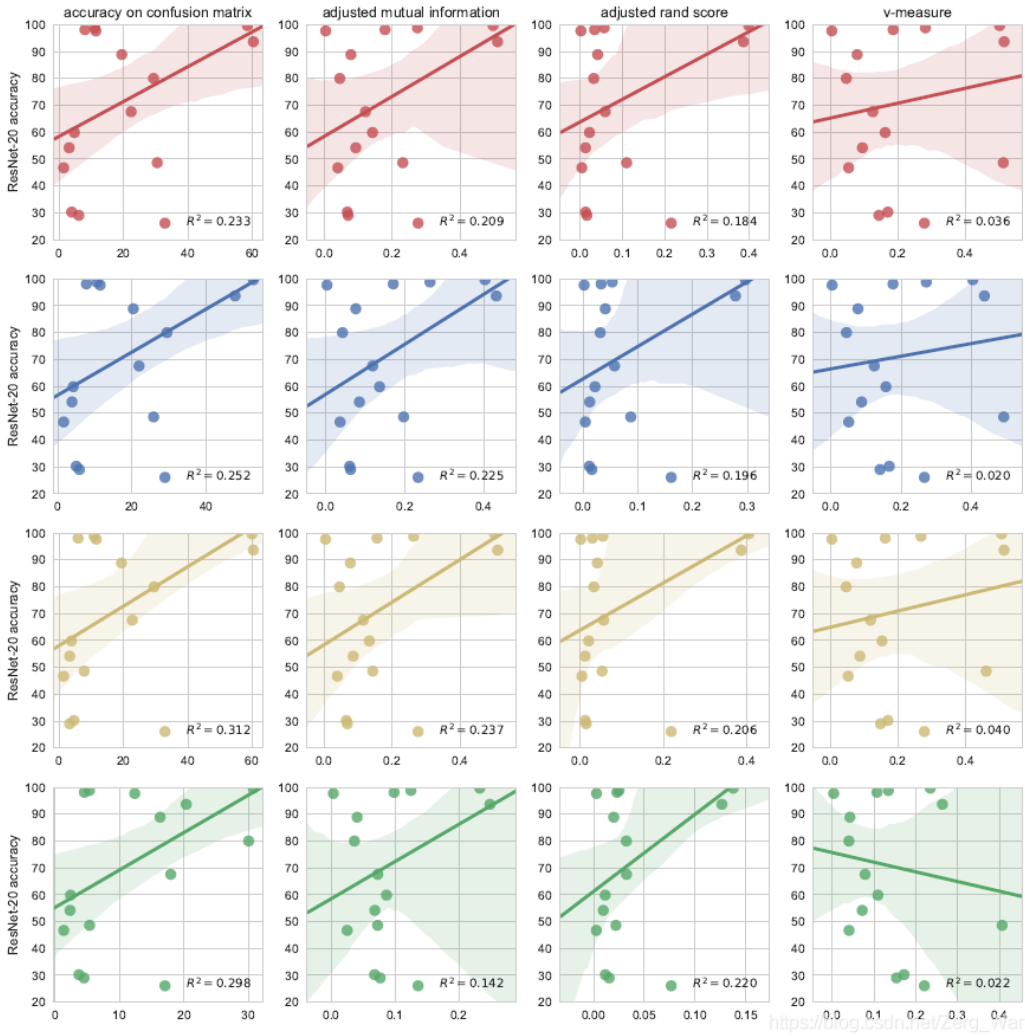


X轴表示数据集的“评分”，Y轴表示数据集对应的训练精度。R^2为回归系数，值越大，数据点分布越趋向于线性，评分算法越好。

由图可知，采用DSSIM的Silhouette Score算法要比采用MSE的要好，作者解释，这是由于DSSIM更多地保留了图像的空间信息。

同理，在原图像尺寸下计算Silhouette Score的算法要比resize后的好。

K-means 聚类算法

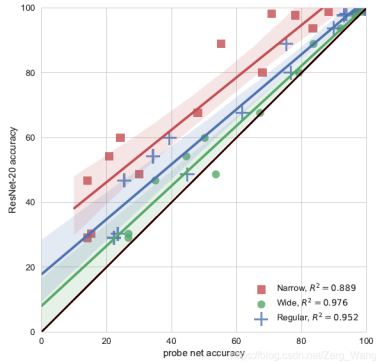


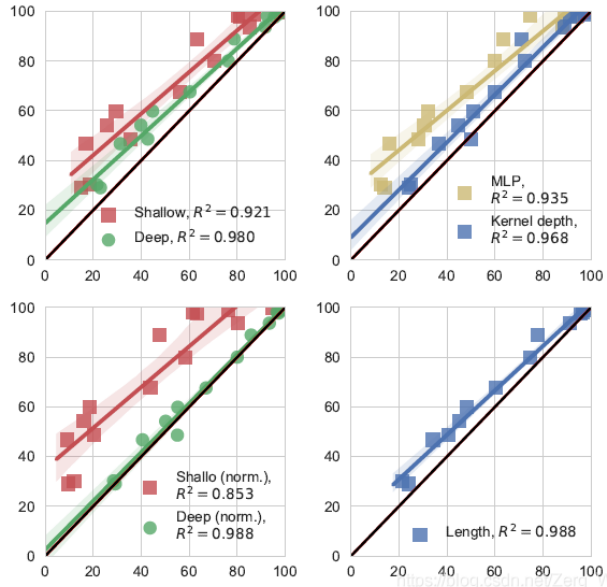
每张子图的X、Y轴与前面所述一致，子图中，每列代表采用不同的衡量标准。（分别为混淆矩阵精度、调整互信息、调整兰德系数、v-measure。由于 v-measure 是 completeness 和 homogeneity 的加权平均，且三者高度相关，所以仅采用 v-measure）

子图中，每行表示采用不同的图像处理，分别为none、resize、 PCA以及encode。由图可知，与none相比，encode和resize对回归系数影响不明显，PCA则有较为明显的提升。

值得一提的是，不采用以及采用PCA的K-means 聚类算法在每张图像上的平均处理速度，要比Silhouette Score算法分别快上5.2倍和50.5倍。

Probe Nets 算法



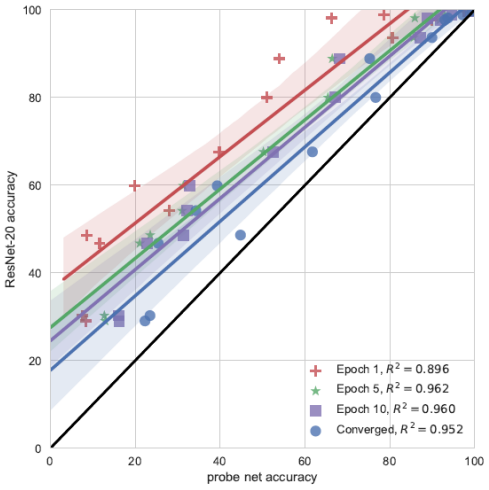


X轴为Probe Net的精度，Y轴为实验统一网络及参数训练下的精度。

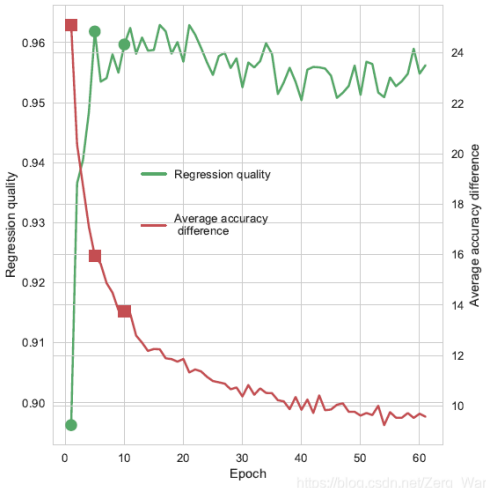
通过上图可知，Probe Net算法的好坏，与网络参数量有较大关系，越多的参数可带来越好的算法表现，但这也会带来更大的计算成本。

Probe Net算法质量与epoch的关系

Probe Net算法在相关性上表现突出，但该算法若要有实际上的意义，其计算成本应该要比正常训练网络低若干个数量级。由于Probe Net 仅需对数据集进行“评分”，而不需要达到完全拟合。因此，为了进一步降低计算成本，作者认为在训练的epoch上可以减少：



作者采用了regular 版本的Probe Net，可见，仅需进行5个epoch，算法便能很好地对数据集进行评估。



绿色线为回归系数R^2。

结论

三种评估数据集分类难度的算法中，Silhouette Score各方面都不行，pass。K-means 聚类算法比Silhouette Score快很多，但算法质量（数据集评分与相同网络下训练精度的线性相关性）和Silhouette Score差不多。而Probe Net 算法在线性相关性方面表现突出（R^2在0.9左右），远超前两种算法，可以较好地评估数据集分类难度，且计算成本较低，评估所需时间是正常训练分类数据集的27倍。

参考资料

https://blog.csdn.net/zhanglianhai555/article/details/104801318