近期工作汇总

阅读CrossEntropySampling方法及复现论文效果

motivation

这篇文章探索了如何在测试集中进行子集选择，**挑选出的子集能够最真实的反应模型在实际测试集的准确率**。这篇文章利用了分层抽样能够降低方差的思想，利用模型倒数第二层**hidden state**作为分层指标进行抽样，选择出的子集能够在保持准确率的前提下减少方差。

method

优化方法：random walking

$$\min_{T\subset S,|T|=n} \overline{CE}(T) = -\frac{\sum_{i=1}^{m}\sum_{z_i=1}^{K} P_S^{e_i}(z_i)\log P_T^{e_i}(z_i)}{m},$$

where

$$P_S^{e_i}(z_i) = \frac{|\{\boldsymbol{x}\in S \mid f_{e_i}(\boldsymbol{x}) = z_i\}|}{|S|}.$$

Algorithm 1 Test Input Selection

Input: Original unlabled test set S, DNN $\mathfrak{M}$, the budget n for labeling inputs.

Output: Selected test set T ($|T| = n$) for labeling.

1: Selecting randomly p examples as the initial test set T.
2: **while** $|T| < n$ **do**
3: Randomly select ℓ groups of examples, $Q_1, \ldots, Q_\ell$. Each group contains $\min(q, n - |T|)$ examples.
4: Choose the group that minimizes the cross entropy, i.e.,

$$Q^* = \min_{Q_i} \overline{CE}(T \cup Q_i), i = 1, \ldots, \ell. \quad (11)$$

5: $T \leftarrow T \cup Q^*$.
6: **end while**

复现论文效果

阅读完论文后，我先跑通源代码中的实验，并复现了部分实验的结果（在MNIST数据集上），验证算法的正确性。

No.	Train Set	Model	Operational Test Set	Actual Acc. (%)	E-Value CES/SRS
1	MNIST	LeNet-1	MNIST	93.1	0.588
2		LeNet-4		96.8	0.655
3		LeNet-5		98.7	0.708
4	Mutant1[a]			79.5	0.499
5	Mutant2[a]			77.3	0.380
6	Mutant3[a]			79.1	0.478

MNIST	LeNet-1	MNIST	94.86%	0.577
	LeNet-4		96.79%	0.619
	LeNet-5		98.68%	0.7237
Mutant1			79.53%	0.443
Mutant2			77.27%	0.6358
Mutant3			79.14%	0.548

阅读Deepcore框架并在Deepcore框架下实现CES

我借鉴CES的源代码，将部分方法重新在Deepcore框架下重新实现，并在MNIST和CIFAR10,CIFAR100数据集上重新进行了测试，并得到了部分实验结果。

我对比了random sampling和cross entropy sampling两种方法在三个数据集上的表现，每次sampling选取测试集fraction=0.1的数据，并独立重复采样15次，得到均值和方差的结果如下图所示。

datasets	model	crossentropy sampling fraction=0.1															mean	variance	true acc
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
MNIST	LeNet-1	0.94726	0.958208955	0.940298507	0.94428	0.94726	0.94627	0.95423	0.95124	0.95423	0.95025	0.95522	0.95025	0.95124	0.95323	0.94826	0.950116086	2.16213E-05	0.9486
	LeNet-4	0.96915	0.978109453	0.964179104	0.96816	0.97413	0.96418	0.96716	0.96816	0.97114	0.97214	0.96617	0.97114	0.97214	0.9791	0.97015	0.970348259	1.95469E-05	0.9679
cifar10	ResNet	0.73433	0.721393049	0.75422883	0.73632	0.74129	0.73632	0.7602	0.73433	0.72139	0.78209	0.75522	0.74129	0.74428	0.77612	0.75423	0.746202318	0.000289955	0.74940002
cifar100	ResNet	0.39502	0.403980106	0.404975116	0.39104	0.41095	0.39403	0.39104	0.41592	0.39104	0.42289	0.42189	0.40398	0.40896	0.41493	0.39602	0.404444442	0.000117075	0.41080001
datasets	model	random sampling fraction=0.1															mean	variance	true acc
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
MNIST	LeNet-1	0.947	0.939999998	0.948000014	0.947	0.956	0.948	0.95	0.95	0.96	0.939	0.944	0.948	0.948	0.941	0.935	0.946733336	4.09237E-05	0.9486
	LeNet-4	0.97	0.967000008	0.970000029	0.975	0.967	0.96	0.968	0.978	0.968	0.963	0.962	0.966	0.962	0.962	0.965	0.966866672	2.4981E-05	0.9679
cifar10	ResNet	0.734	0.758000016	0.736000001	0.733	0.777	0.736	0.75	0.736	0.765	0.728	0.744	0.773	0.739	0.737	0.749	0.746333337	0.000219289	0.74940002
cifar100	ResNet	0.394	0.393999994	0.409000009	0.395	0.422	0.423	0.417	0.402	0.393	0.423	0.382	0.418	0.402	0.407	0.425	0.407066671	0.000175929	0.41080001

之后为了验证采样数据测试的模型准确率能够反映真实准确率，我还对其进行了假设检验，选取原假设H_0：模型采样后测试的准确率的均值是true acc.采用显著性水平$\alpha = 0.05$，以正态分布作为独立重复实验的噪声与true acc的偏差，假设检验的检验量与t值如下所示。

datasets	model	true acc	检验量	t_14(0.025)
MNIST	LeNet-1	0.9486	1.262781522	2.145
	LeNet-4	0.9679	2.144687386	
cifar10	ResNet	0.7494	0.727307535	
cifar100	ResNet	0.4108	2.274926691	

将CES方法由TensorFlow模型迁移到pytorch模型

我把结果给学长看过之后学长觉得cifar10和cifar100两个数据集采用的模型准确率偏低，让我换两个模型训练后重新实验。

我在网上搜索相关的模型代码之后发现TensorFlow的模型相比于pytorch来说少很多，而且模型的准确率都更低，达不到学长要求的准确率。所以我只能使用pytorch模型，但是CES源代码中的相关方法都是在TensorFlow的框架下实现的，无法直接迁移到pytorch模型，因此我**根据pytorch框架的相关方法对原方法进行了重新实现**。

实验结果

datasets	model	crossentropy sampling fraction=0.1															mean	variance	true acc
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
MNIST	LeNet-1	0.94726	0.958208955	0.940298507	0.94428	0.94726	0.94627	0.95423	0.95124	0.95423	0.95025	0.95522	0.950248756	0.951243781	0.953233831	0.94826	0.950116086	2.16213E-05	0.9486
	LeNet-4	0.96915	0.978109453	0.964179104	0.96816	0.97413	0.96418	0.96716	0.96816	0.97114	0.97214	0.96617	0.971144279	0.972139303	0.979104478	0.97015	0.970348259	1.95469E-05	0.9679
cifar10	ResNet	0.95423	0.954228856	0.949253731	0.95721	0.94726	0.95025	0.95224	0.94726	0.95224	0.95821	0.94129	0.948258706	0.953342895	0.956423945	0.9548	0.951767143	1.92191E-05	0.9543
cifar100	efficien	0.89796	0.8475	0.879396985	0.85606	0.86967	0.90176	0.85859	0.86364	0.88413	0.84051	0.86902	0.882352941	0.889724311	0.861111111	0.88917	0.872705899	0.000313277	0.8697
datasets	model	random sampling fraction=0.1															mean	variance	true acc
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
MNIST	LeNet-1	0.947	0.939999998	0.948000014	0.947	0.956	0.948	0.95	0.95	0.96	0.939	0.944	0.948000014	0.948000014	0.940999985	0.935	0.946733336	4.09237E-05	0.9486
	LeNet-4	0.97	0.967000008	0.970000029	0.975	0.967	0.96	0.968	0.978	0.968	0.963	0.962	0.966000021	0.962000012	0.962000012	0.965	0.966866672	2.4981E-05	0.9679
cifar10	ResNet	0.954	0.957	0.951	0.953	0.953	0.953	0.96	0.948	0.954	0.951	0.968	0.955	0.938	0.955	0.954	0.9536	3.69067E-05	0.9543
cifar100	ResNet	0.871	0.865	0.858	0.873	0.846	0.88	0.862	0.882	0.866	0.86	0.867	0.862	0.881	0.868	0.875	0.867733333	8.83289E-05	0.8697

datasets	model	true acc	检验量	t_14(0.025)
MNIST	LeNet-1	0.9486	1.262781522	2.145
	LeNet-4	0.9679	2.144687386	
cifar10	ResNet	0.9543	2.237635512	
cifar100	efficien	0.8697	0.657741941	

下一步工作

1.确定最佳的超参数p,q,l，兼顾效率与准确率。

2.目前算法效率不能满足实际需求，需要提升效率