清华提出LogME，无需微调就能衡量预训练模型的下游任务表现！

夕小瑶的卖萌屋 2021-08-06 16:58

以下文章来源于THUML，作者游凯超

THUML
清华大学软件学院机器学习实验室，专注于迁移学习、深度学习、知识学习等基础理...

文 | 游凯超
源 | THUML

引言

在深度学习时代，神经网络的参数量越来越大，从头开始训练(train from scratch)的成本也越来越大。幸运的是，在计算机视觉、自然语言处理等人工智能应用的主要领域，人们能够采用迁移学习的**预训练-微调**范式来有效降低训练成本。迁移学习使得深度神经网络以预训练模型的形式走进千家万户，不用上千块TPU，我们也能够使用BERT、EfficientNet等大型模型。

如今，对于深度学习框架来说，丰富的预训练模型库已经是标配了(例如TensorFlow Hub, Torchvision Models)。在一些研究领域(比如2020年非常热门的自监督学习)，研究成果最终也是以预训练模型的方式呈现给社区。在深度学习社区里，一些热门领域已经积累了成百上千个预训练模型。

面对众多预训练模型，我们在进行迁移时，该用哪一个好呢？这个重要问题很少有人研究，因此人们目前只好使用一些简单粗暴的办法:

- 使用常见的预训练模型(例如ResNet50)
- 使用预训练指标(例如ImageNet准确率)高的模型

如果想要准确地选择最好的预训练模型，我们需要把每一个候选模型都做一遍微调。因为微调涉及到模型训练，时间至少几个小时起步。有些预训练模型的微调还需要进行超参数搜索，想要决定一个预训练模型的迁移效果就需要将近50个小时！

针对这一问题，我们进行了深入探究，提出了一种名为LogME的方法。它能极大地加速预训练模型选择的过程，将衡量单个预训练模型的时间**从50个小时减少到一分钟，疯狂提速三千倍**！目前该论文已被ICML2021接受。

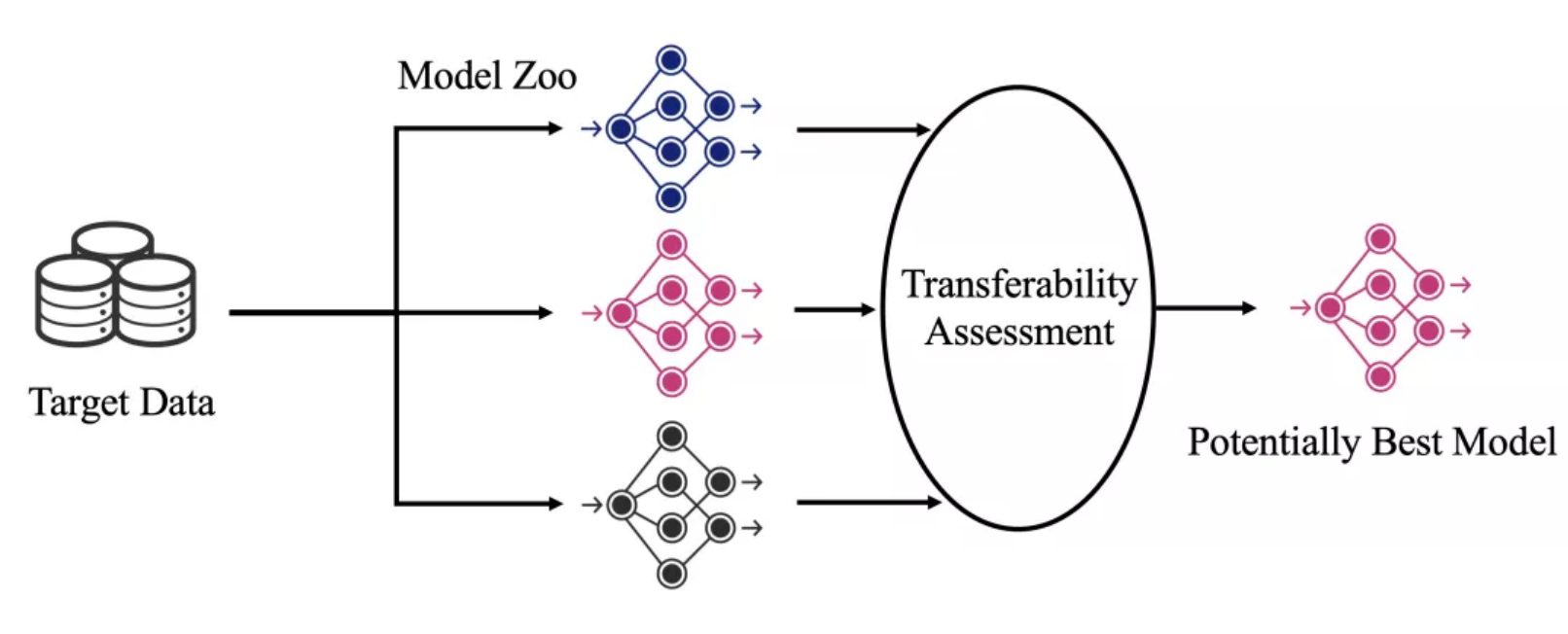
论文标题：
LogME: Practical Assessment of Pre-trained Models for Transfer Learning

论文链接：
https://arxiv.org/abs/2102.11005

GitHub链接：
https://github.com/thuml/LogME

问题描述

预训练模型选择问题，就是针对用户给定的数据集，从预训练模型库中选择一个最适合的预训练模型用于迁移学习。其流程可以简单概括为下图，核心就是要对每一个预训练模型进行迁移性评估(Transferability Assessment)，简单来说就是为每个模型打分，然后选择出打分最高的预训练模型。



▲预训练模型选择问题

最直接的打分方法，就是将预训练模型在给定数据集上进行调参、微调，将最终的准确率或者其它衡量指标作为预训练模型的分数。我们将这种方法称为ground-truth方法，它的选择效果无疑是最好的，总是能选出最合适的预训练模型。然而，它的时间开销太大(每个预训练模型需要50小时)，因此无法实用。

一个好的打分标准，需要在保持与ground-truth打分的高度相关性的同时，尽可能降低时间开销，才能满足实际使用的要求。除了ground-truth方法之外，目前还有两种打分方法(LEEP和NCE)，但是它们的使用范围非常有限，只能用于有监督预训练模型迁移到分类任务的场景，如下表所示，而我们提出的**LogME则能够胜任几乎所有常见的场景**，覆盖了视觉、NLP、分类、回归、有监督预训练模型、无监督预训练模型等方向。

Modality	Pre-train	Target	LEEP	NCE	LogME
vision	classification	classification	✓	✓	✓
	classification	regression	✗	✗	✓
	contrastive	classification	✗	✗	✓
	contrastive	regression	✗	✗	✓
language	LM	classification	✗	✗	✓

▲应用场景比较，LogME能胜任几乎所有常见场景

LogME方法

LogME的优越性能来自于以下三个方面：(1)无须梯度计算;(2)无须超参数调优;(3)算法实现优化。下面围绕这三个方面对LogME进行具体介绍。

为了加速预训练模型选择，我们仅**将预训练模型视作特征提取器**，避免更新预训练模型。这样，只需要将预训练模型在给定数据集上前向传播一遍，就可以得到特征$\{f_i=\phi(x_i)\}_{i=1}^n$和标注$\{y_i\}_{i=1}^n$。于是，这个问题就转化成了如何衡量特征和标注之间的关系，也就是说，这些特征能够多大程度上用于预测这些标注。

为此，我们采用一般性的统计方法，用概率密度$p(y \mid F)$来衡量特征与标注的关系。考虑到微调一般就是在预训练模型的特征提取层之上再加一个线性层，所以我们用一个线性层来建模特征与标注的关系。

说到这里，很多人会想到，一种直观的方法是通过Logistic Regression或者Linear Regression得到最优权重w^*，然后使用似然函数$p(y \mid F, w^*)$作为打分标准。但是这样容易导致过拟合问题，而且这些方法也有很多超参数需要选择，这使得它们的时间开销很大且效果不好。

我们选用的是统计学中的证据(evidence,也叫marginalized likelihood)来衡量特征与标注的关系。它不使用某个特定的w^*的值，而是使用w的分布来得到边缘化似然的值$p(y|F)=\int p(w)p(y|F,w)\mathrm{d}w$。它相当于取遍了所有可能的$w$值，能够更加准确地反映特征与标注的关系，不会有过拟合的问题。其中，$p(w)$与$p(y \mid F)$分别由超参数α和β决定，但是它们**不需要grid search**，可以通过最大化evidence来直接求解。于是，我们就得到了对数最大证据(Log Maximum Evidence，缩写LogME)标准来作为预训练模型选择的依据。具体数学细节不在这里赘述，感兴趣的读者可以阅读底部的论文。算法的具体细节在下图中给出了。注意，虽然LogME计算过程中将预训练模型ϕ视作特征提取器，但是**LogME可以用于衡量ϕ被用于迁移学习(微调)的性能**。

```
Algorithm 1 LogME
 1: Input: Pre-trained model $\phi$
           Target dataset $\mathcal{D}=\{(x_i,y_i)\}_{i=1}^n$
 2: Output: logarithm of maximum evidence (LogME)
 3: Extract features using pre-trained model $\phi$:
           $F\in\mathbb{R}^{n\times D}, f_i=\phi(x_i), Y\in\mathbb{R}^{n\times K}$
 4: Compute SVD $F^TF=V\mathrm{diag}\{\sigma\}V^T$
 5: for $k=1$ to $K$ do
 6:     Let $y=Y^{(k)}\in\mathbb{R}^n$, initialize $\alpha=1,\beta=1$
 7:     while $\alpha,\beta$ not converge do
 8:         Compute $\gamma=\sum_{i=1}^D\frac{\beta\sigma_i}{\alpha+\beta\sigma_i}, \Lambda=\mathrm{diag}\{(\alpha+\beta\sigma)\}$
 9:         Naïve: $A=\alpha I+\beta F^TF, m=\beta A^{-1}F^Ty$
10:         Optimized: $m=\beta(V(\Lambda^{-1}(V^T(F^Ty))))$
11:         Update $\alpha\leftarrow\frac{\gamma}{m^Tm}, \beta\leftarrow\frac{n-\gamma}{||Fm-y||_2^2}$
12:     end while
13:     Compute $\mathcal{L}_k=\frac{1}{n}\mathcal{L}(\alpha,\beta)$ using Eq. 2
14: end for
15: Return LogME $\frac{1}{K}\sum_{k=1}^K\mathcal{L}_k$
```

▲LogME算法具体流程

值得一提的是，LogME算法涉及到很多矩阵分解、求逆、相乘操作，因此一不小心就容易使得算法的复杂度很高(例如上图第9行，粗糙的实现方式)。我们在深入研究该算法后发现，很多矩阵运算的开销可以通过巧妙的计算优化手段大大降低，因此将计算流程优化为上图第10行，**整体的计算复杂度降低了一个阶**，从四次方降低为三次方(见下表)，使得该算法在数秒内就能处理常见情况。

	Complexity per for-loop	Overall complexity
naïve	$\mathcal{O}(D^3+nD^2)$	$\mathcal{O}(KD^3+nKD^2)$
optimized	$\mathcal{O}(D^2+nD)$	$\mathcal{O}(KD^2+nKD+D^3+nD^2)$

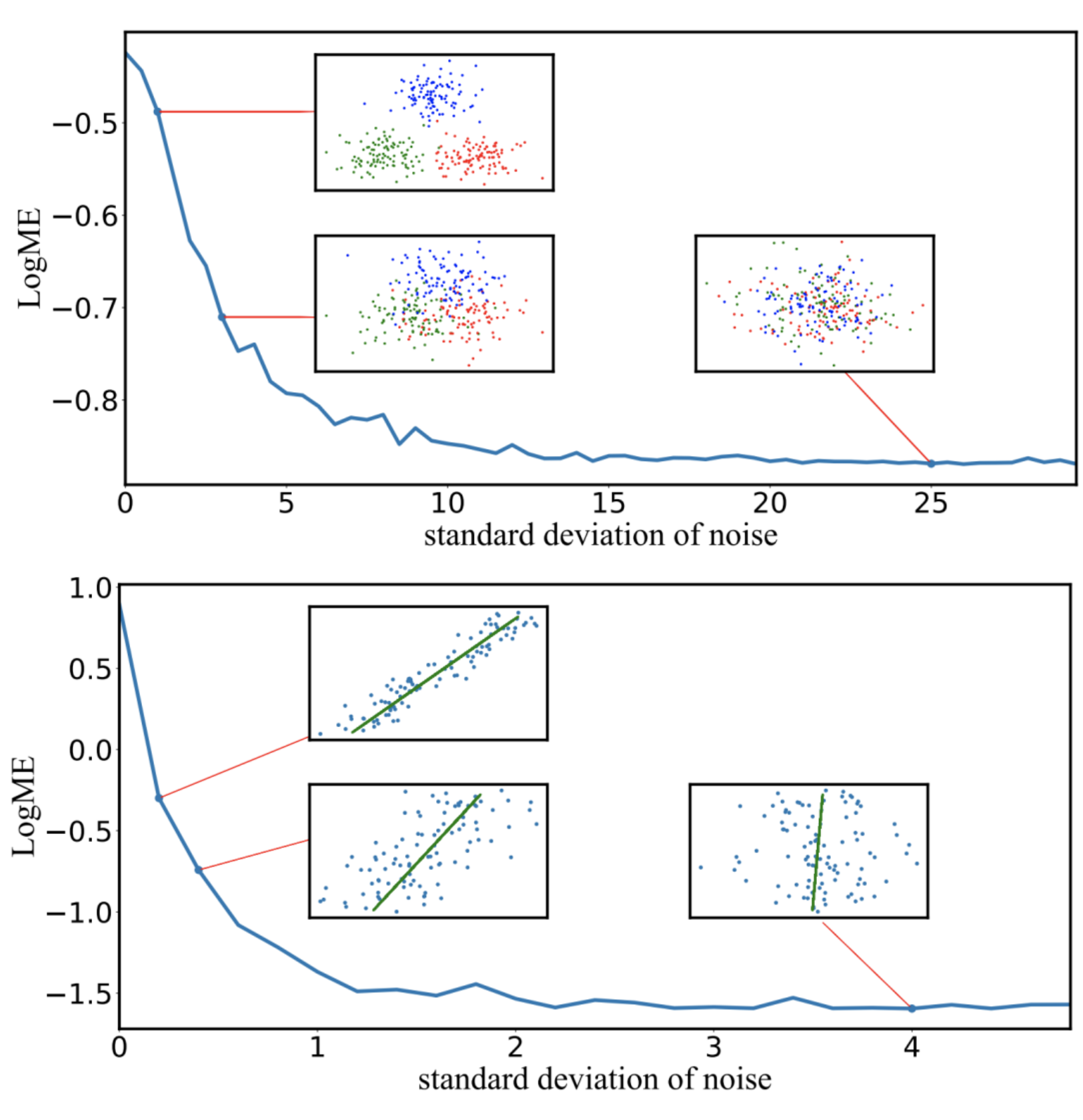
▲优化前后复杂度对比，n是数据量，D是特征维度，K是类别数目

实验

在实验部分，我们用合成数据、真实数据等多种方式方式，测试了LogME在**17个数据集**、**14个预训练模型**上的效果，LogME在这么多数据集、预训练模型上都表现得很好，展现了它优异的性能。

合成数据

首先让我们看看，LogME给出的打分标准与人的主观感觉是否一致。我们为分类问题和回归问题分别设计了一个toy实验，使用生成数据来测量LogME的值。从下图中可以看出，不管是分类任务还是回归任务，**当特征质量越来越差时，LogME的值也越来越低**，说明LogME可以很好地衡量特征与标注的关系，从而作为预训练模型选择的标准。

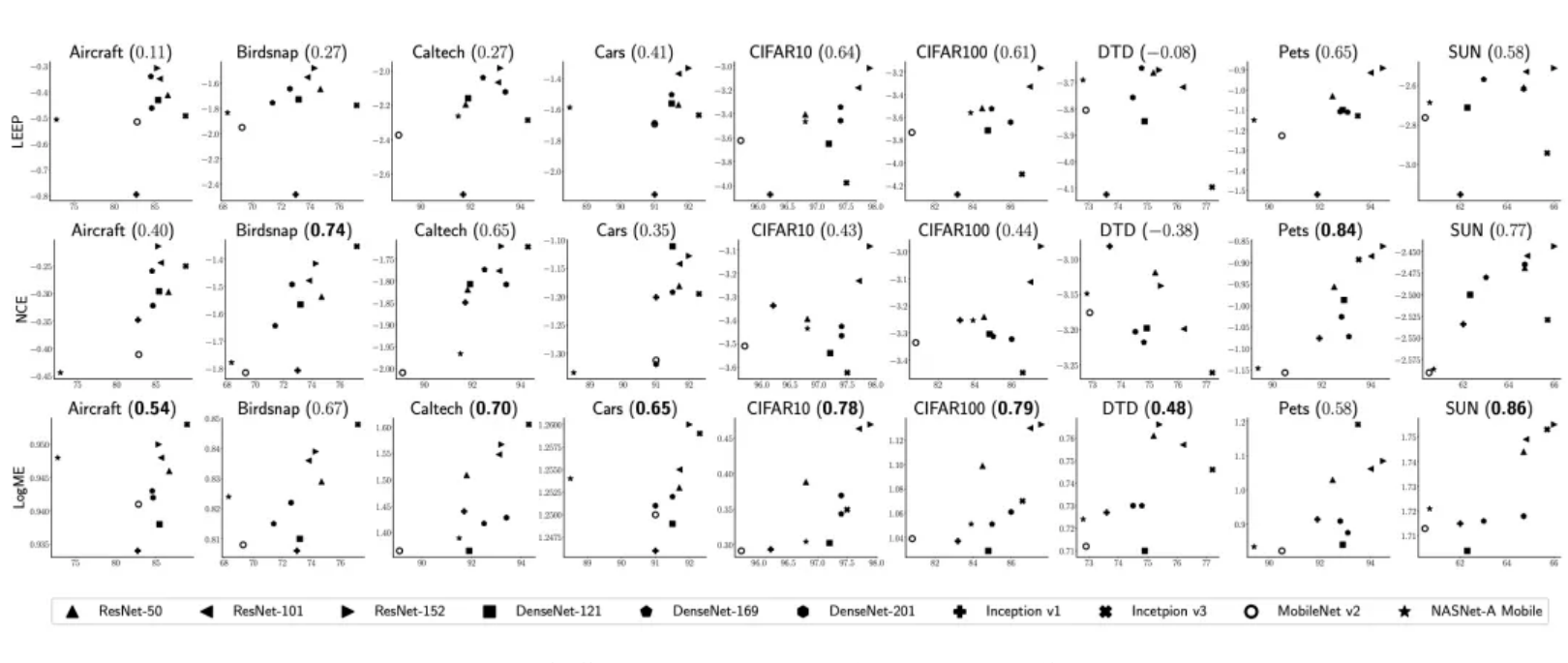


▲特征质量越来越差时，LogME也越来越低。

接下来，我们用LogME来进行预训练模型选择。我们使用若干个常用预训练模型，通过耗时的微调过程得到它们的迁移性指标，然后衡量LogME与迁移性指标的相关性。相关性指标τ_w为加权肯达尔系数，它的取值范围是$[-1,1]$。相关系数为τ_w意味着如果LogME认为预训练模型ϕ_1比ϕ_2好，那么确实ϕ_1比ϕ_2好的概率是$\frac{\tau_w+1}{2}$。也就是说，τ_w越大越好。

有监督预训练模型迁移到分类数据集

我们将10个常用预训练模型迁移到9个常见分类数据集中，发现LogME与微调准确率有很高的相关性(见下图)，显著优于之前的LEEP和NCE方法。在这几个数据集中，LogME的相关系数τ_w至少有0.5，大部分情况下有0.7或者0.8，也就意味着使用LogME进行预训练模型选择的准确率高达85%或者90%。

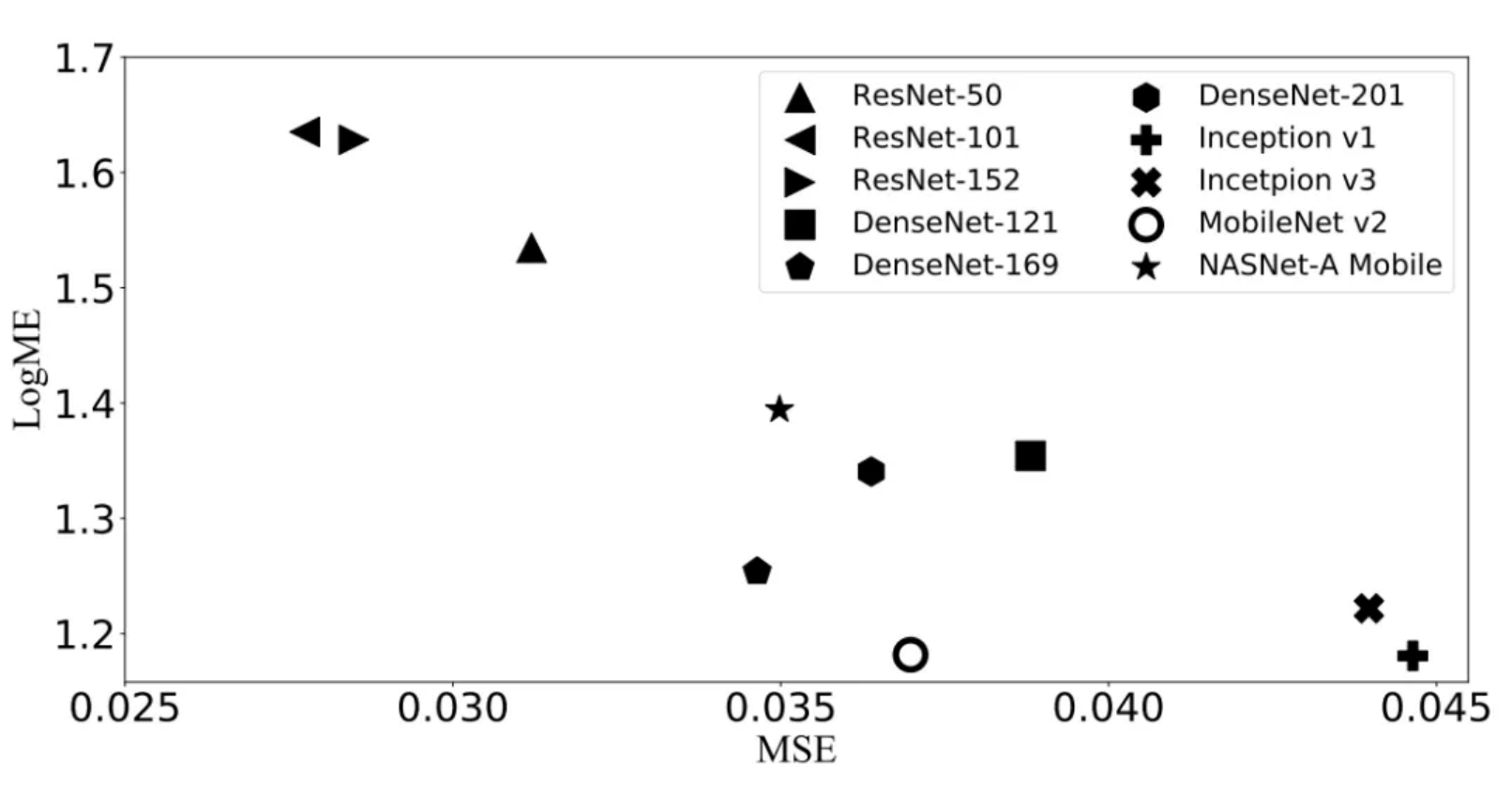


▲有监督预训练模型迁移到分类数据集

值得注意的是，之前的LEEP和NCE方法只能用于这一种场景。接下来的实验(回归任务、无监督预训练模型、NLP模型及任务)，只有LogME能处理。

有监督预训练模型迁移到回归数据集

我们也做了回归任务相关的实验，可以看到LogME与MSE有明显的负相关性，而MSE是越低越好，LogME是越大越好，结果符合预期。



▲有监督预训练模型迁移到回归数据集

无监督预训练模型

在2020年，视觉领域的重要进展之一就是无监督预训练模型。因此我们也尝试了使用LogME来判断无监督预训练模型的质量。从下图的结果来看，不论是分类任务(Aircraft)还是回归任务(dSprites)，LogME都能准确衡量无监督预训练模型的质量。

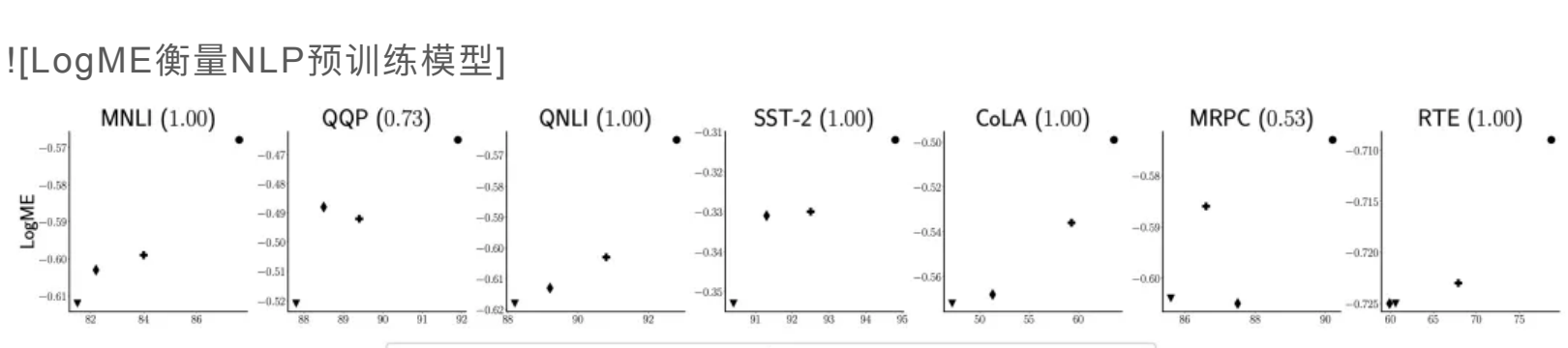
Pre-trained Network	Aircraft		dSprites	
	Accuracy (%)	LogME	MSE	LogME
MoCo V1	81.68	0.93	0.069	1.52
MoCo V2	84.16	0.94	0.047	1.64
MoCo 800	86.99	0.95	0.050	1.58
	τ_w: 1.0		τ_w: 1.0	

▲使用LogME来衡量无监督预训练模型

自然语言处理任务

LogME并不局限于视觉模型与任务，我们还测试了它对NLP预训练模型的评价能力。可以看到，在五个任务上，LogME完美地预测了四个预训练模型的表现的相对大小，在另外两个任务上的表现也不错。

![LogME衡量NLP预训练模型]



时间加速

LogME方法不仅效果好，更难得的是它所需要的时间非常短，可以**快速评价预训练模型**。如果将直接微调的时间作为基准，LogME只需要0.31‰的时间(注意不是百分号，是千分号)，也就是说加速了3000倍！而之前的方法如LEEP和NCE，虽然耗时更少，但是效果很差，适用范围也很有限，完全不如我们的LogME方法。

	Wall-clock time (second)	Proportion
fine-tune (upper bound)	$(1.61 \pm 0.06) \times 10^5$	1000‰
extract feature (lower bound)	37.3 ± 0.6	0.23‰
LEEP (Nguyen et al., 2020)	37.3 ± 0.6	0.23‰
NCE (Tran et al., 2019)	37.5 ± 0.6	0.23‰
LogME (naïve implementation)	839.8 ± 5.6	5.22‰
LogME (optimized)	50.4 ± 0.7	0.31‰

▲各种方法耗时比较，LogME加速3000倍

值得注意的是，像LogME这种根据概率公式计算的方法，一般效果更好，但是耗时也更高。事实上，如果我们采用简单粗暴的实现，评估一个模型就需要八百多秒。正是有了精心优化的版本，我们才能够**既有概率方法的优越效果，又有简单高效的实现**。

展望

因为它的准确、快速、实用性，我们相信LogME除了能够用作预训练模型选择之外，还能够在其它领域有所作为。例如，在无监督预训练中，评估一次预训练模型就需要在整个ImageNet数据集上进行linear protocol evaluation，整个过程需要几个小时。若采用LogME，则只需要一分钟不到，因此可以在训练过程中将LogME作为early stopping的准则。