

样本生而不等——聊聊那些对训练数据加权的方法



Naiyan ...

深度学习 (Deep Learning)、机器学习、人工智能 话题的优秀回答者

✎ 编辑推荐

446 人赞了该文章

现有大部分机器学习或者深度学习的研究工作大多着眼于模型或应用，而忽略对数据本身的研究。今天给大家介绍的几个文章就关注于在机器学习中如何通过对训练集的选择和加权取得更好的测试性能。

在开始之前，先和大家简单回顾一下我个人觉得相关的几方面工作。其实远在深度学习时代之前，**根据loss对样本加权**的工作就已经有很多。神奇的是，其实在一条线上有着截然相反的想法的研究：第一类工作的想法是如果一个样本训练得不够好，也就是loss高的话，那么说明现在的模型没有很好fit到这样的数据，所以应该对这样的样本给予更高的权重。这一类工作就对应到经典的 Hard Negative (Example) Mining，近期的工作如Focal Loss也是这个思想。另一类工作的想法是学习需要循序渐进，应该先学习简单的样本，逐渐加大难度，最终如果仍然Loss很大的样本，那么认为这些样本可能是Outlier，强行fit这些样本反而可能会使泛化性能下降。这一类中对应的是Curriculum Learning或者Self-Paced Learning类型的工作。**本质上，这两个极端对应的是对训练数据本身分布的不同假设。**第一类方法认为那些fit不好的样本恰恰是模型应当着重去学习的，第二类方法认为那些fit不上的样本则很可能是训练的label有误。

所以，一个很有趣的问题是：**我们应该何时在这两种极端之间选择？在这两个极端之间是不是会有更好的权衡？**这个问题乍看上去没什么简单的办法，今天要介绍的文章就是引入了一个新的信息源——一个无偏的验证集来解决这个问题。有了这样额外的信息源之后，这个问题就变成了**如何对每个样本加权，使得验证集上的loss下降**。一个naive的办法自然是用leave one out，去掉每个样本训练一个model，但是这个cost会非常地大，实际上是不可行的。所以核心就在于如何对model进行近似，用尽量低的代价尽量准确地获得这样的信息。

在[1]中，作者使用了一个统计学中经典工具Influence Function。作者首先从一个twice-differentiable的strictly convex函数出发一步步拓展结论。基本思路是考虑如果我们增加eps某一个样本的weight，会对model的参数有怎样的影响：

$$\begin{aligned}\hat{\theta}_{\epsilon,z} &\stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta) + \epsilon L(z, \theta). \\ \mathcal{I}_{\text{up, params}}(z) &\stackrel{\text{def}}{=} \left. \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}), \\ H_{\hat{\theta}} &\stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \nabla_{\theta}^2 L(z_i, \hat{\theta})\end{aligned}$$

知乎 @Naiyan Wang

其中H在这里是二阶Hessian矩阵。如果有熟悉优化的朋友可以看出来这个形式其实和Newton法很像。实际推导也没有用到很深奥的数学知识，有兴趣的读者可以参照下文章中的附录。我们更进一步可以使用链式法则得出对z加大eps的weight后对于某个测试样本 z_{test} 的loss变化：

$$\begin{aligned}\mathcal{I}_{\text{up, loss}}(z, z_{\text{test}}) &\stackrel{\text{def}}{=} \left. \frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon,z})}{d\epsilon} \right|_{\epsilon=0} \\ &= \nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^{\top} \left. \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon} \right|_{\epsilon=0} \\ &= -\nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^{\top} H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}).\end{aligned}$$

知乎 @Naiyan Wang

这个结论其实很有指导意义，告诉了我们在一个训练好的model上，如何不重新训练就能评估一个样本对某个测试样本的重要性。然而想直接使用这个办法还有最后一个障碍就是Hessian矩阵的计算，对于CNN这样参数量巨大的模型来说，想要完全准确计算代价依旧很高。所以作者又提出了两种近似Hessian矩阵的方法，分别是使用Conjugate Gradient和Stochastic Approximation。

由于这不是文章的重点，所

▲ 赞同 446 ▼

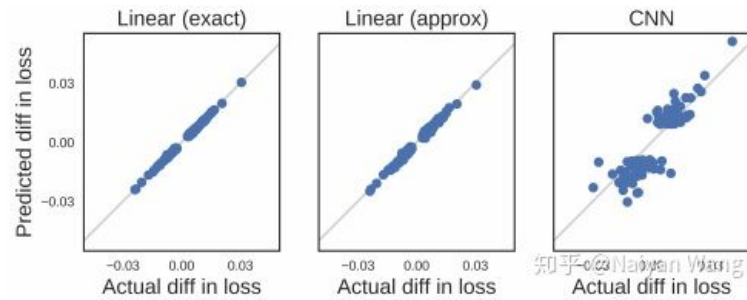
● 29 条评论

➤ 分享

★ 收藏

...

作者在文中还将这样一个结论推广，一方面分别给出了在非凸函数、非收敛的情形以及不可微分的loss下的分析，此方法或此方法的变种都能很有效地预测在验证集上测试性能的变化。另一方面还类似地推导了对某一个训练样本本身扰动带来的变化，这个结论自然地adversarial sample这个问题联系了起来。为了验证这样近似的有效性，作者还去和leave one out的exact influence做了比较，可以看到，在线性模型下这个模型近似的效果基本和GT一致，在CNN这种非凸情形下，虽然肯定不如线性模型下的效果，但是和GT仍然保持了高度相关性。



有了上面的结论，一个自然的想法便是，找出那些对于降低验证集loss没有帮助的样本，把他们从训练集中排除，从而提升model的性能。[2]正就是做了这样的工作，作者们也给这个工作起了一个很有意思的名字——Data Dropout。具体做法是首先使用全部数据得到一个初始模型，然后在这样的初始模型上计算每个样本的influence，去掉那些对降低验证集loss的样本后，使用新的训练集再次训练得到最终的模型。作者也分别在不同规模(CIFAR, ImageNet)和不同应用(Classification, Denoising)中证明了这样做的有效性。

Method	CIFAR-10	CIFAR-100	SVHN
All-CNN [14]	7.25	33.71	-
All-CNN++	5.09	30.67	-
ResNet-110 (reported by [27])	6.41	27.22	-
ResNet-152 (reported by [27])	-	-	2.01
ResNet-110++	4.53	24.98	-
ResNet-152++	-	-	1.64
DenseNet-40 [6]	5.24	24.42	1.79
DenseNet++	3.62	22.51	1.47

	without Algorithm	with Algorithm
ResNet-18 [5]	27.88	24.26
ResNet-34 [5]	25.03	21.97

此方法筛掉的样本数其实也并不多，作者在下表中也报告去除掉的“不好”样本的个数，这几个数据集看来基本是在1%到3%左右，但是带来的性能提升却是显著的。

Method	CIFAR-10	CIFAR-100	SVHN
All-CNN [14]	1365	1419	-
ResNet-110 [5]	1220	1283	-
ResNet-152 [5]	-	-	24261
DenseNet-40 [6]	1076	1149	19433

Method	ImageNet
ResNet-18 [5]	14655
ResNet-34 [5]	13142

另一个相关的工作[3]也是同样的出发点，但是更好地利用了Deep Learning中已有操作，使用meta-learning的办法去学习样本的weight。首先这个问题可以写成一个two-level交替优化的目标函数：

$$\theta^*(w) = \arg \min_{\theta} \sum_{i=1}^N w_i f_i(\theta),$$

$$w^* = \arg \min_{w, w \geq 0} \frac{1}{M} \sum_{i=1}^M f_i^v(\theta^*(w)).$$

其中 f_i 为原始的training loss, f_i^v 为validation loss。但是很显然，直接优化这样一个目标需要在两步之间不停交替迭代，所以作者提出了对下标 i 一个目标函数使用一步gradient descent的近似，同时和上

$$f_{i,\epsilon}(\theta) = \epsilon_i f_i(\theta),$$

$$\hat{\theta}_{t+1}(\epsilon) = \theta_t - \alpha \nabla \sum_{i=1}^n f_{i,\epsilon}(\theta) \Big|_{\theta=\theta_t}.$$

$$\epsilon_t^* = \arg \min_{\epsilon} \frac{1}{M} \sum_{i=1}^M f_i^v(\theta_{t+1}(\epsilon)).$$

更进一步，对于eps的优化同样可以进行一步gradient descent的近似，如果在eps=0附近展开的话，可以得到：

$$u_{i,t} = -\eta \frac{\partial}{\partial \epsilon_{i,t}} \frac{1}{m} \sum_{j=1}^m f_j^v(\theta_{t+1}(\epsilon)) \Big|_{\epsilon_{i,t}=0},$$

$$\tilde{w}_{i,t} = \max(u_{i,t}, 0).$$

为了保证训练时每个batch的effective learning rate一致，作者还对每个batch下的weight做了normalize：

$$w_{i,t} = \frac{\tilde{w}_{i,t}}{(\sum_j \tilde{w}_{j,t}) + \phi(\sum_j \tilde{w}_{j,t})},$$

使用这样的sample weight，对于model的参数theta再次进行一次gradient计算即可完成对于此batch的更新。

文中使用MLP为例，给出了一个导出的weight示例：

$$\frac{\partial}{\partial \epsilon_{i,t}} \mathbb{E} \left[f^v(\theta_{t+1}(\epsilon)) \Big|_{\epsilon_{i,t}=0} \right]$$

$$\propto -\frac{1}{m} \sum_{j=1}^m \frac{\partial f_j^v(\theta)}{\partial \theta} \Big|_{\theta=\theta_t}^\top \frac{\partial f_i(\theta)}{\partial \theta} \Big|_{\theta=\theta_t}$$

$$= -\frac{1}{m} \sum_{j=1}^m \sum_{l=1}^L (\tilde{z}_{j,l-1}^v)^\top \tilde{z}_{i,l-1} (g_{j,l}^v)^\top g_{i,l}.$$

其中 $\tilde{z}_{i,l}$ 代表的是第i个样本第l层的feature，上标v代表的是validation set。同样， $g_{i,l}$ 代表的是第i个样本在第l层收到的gradient。这个结果有着十分直观的含义：**当一个样本和validation set中的样本feature接近，且gradient方向接近的时候，那么我们会增加这个样本的weight**。换句话说，如果一个样本和validation set中的样本接近，且训练的目标一致，那么我们就应该更好地fit这个样本，因为它能直接帮助validation set降低loss。

作者基于这样的方法，还证明了reweighted training在mild condition下的收敛性和传统的SGD算法一致，且可以收敛到validation loss的一个critical point。

在实验部分，作者分别在class imbalanced和noisy label的情况下测试了这个算法。都分别证明了其有效性，但是比较遗憾的是没有和前面提到的[1]进行比较，实验使用的数据集规模也都比较小。

其实还有一些没有覆盖到的paper[4]，我个人觉得没有这两篇有代表性所有就不展开了。总结一下，我觉得这是一个和模型与应用同等重要的问题，其实自己也曾经思索过一段时间但是没有很好的想法。这几个工作提供了一个很不错的思路，即引入一个新的无偏验证集来提供更多的信息。然而这个验证集在实际应用中是否会引入一些overfitting的风险其实还有待更多应用的验证。希望这个方向后续有更多exciting的工作出现。

[1] Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." *ICML* (2017).

[2] Wang, Tianyang, Jun Huan, and Bo Li. "Data dropout: Optimizing training data for convolutional neural networks." *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018.

[3] Ren, Mengye, et al. "Learning to reweight examples for robust deep learning." *ICML* (2018).

[4] Fan, Yang, et al. "Learning What Data to Learn." *arXiv preprint arXiv:1702.08635* (2017).

发布于 2018-12-28

[机器学习](#) [深度学习 \(Deep Learning\)](#)

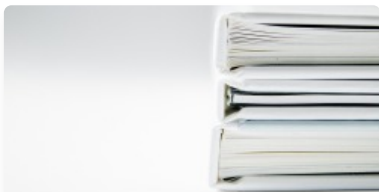
文章被以下专栏收录



Winsty的技术碎碎念

进入专栏

推荐阅读



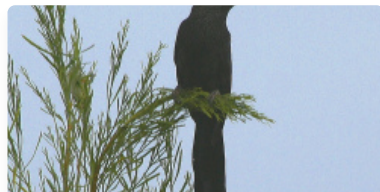
机器学习中样本不平衡的处理方法

谭子健 发表于网站数据分...

不要太纠结小样本中的正确率

这两天做深度学习文本情绪分析（经典的深度学习入门项目），发现这么一个问题。比如常见词汇的杂音处理，作为一个菜B，我马上想到了以下两种方法：1、超高频词语削掉（比如空格这种高频词语...

子楠 发表于机器学习笔...



深度学习与少样本学习

复制粘贴工... 发表于ML学习笔...

29 条评论

切换为时间排序

写下你的评论...



Lyken

5 天前

Percy ICML那篇naiyan你们有在实践中试过吗？

👍 赞



Naiyan Wang (作者) 回复 Lyken

5 天前

没，回头有空可以试试吧

👍 赞



知乎用户 回复 Naiyan Wang (作者)

5 天前

一回头就是好几年了，嘿嘿嘿

👍 1



信息门下走狗

5 天前

我能说第一种方法看上去好像momentum map么？也可以看作从欧氏空间变到了信息几何的黎曼空间处理问题。

👍 赞



Naiyan Wang (作者) 回复 信息门下走狗

5 天前

呃 真的不懂你说的。。。也许可以从别的角度给出解释

👍 赞



信息门下走狗 |

▲ 赞同 446 ▼

💬 29 条评论

➦ 分享

★ 收藏

...

通过对目标函数的扰动计算系统参数的修正，参数 θ 可以看作一个李代数上的向量，作用在图象上对数据进行变换，代价函数定义在数据上，扰动诱导在数据空间的切向量变化，然后通过momentum map回到李代数上，完成对参数的修正。Hessian矩阵在数据上取期望实际定义了一个黎曼度量，这个东西实际是在计算网络复杂度，通过调整权重达到降低网络复杂度。

👍 赞

展开其他 1 条回复

 信息门下走狗 5 天前

直观上，肯定是靠近分界面的样本更重要需要增加权重，而位于类中心部分的样本可以降低权重。另外，也许直接对数据权重进行随机扰动就可以起到类似dropout的约束作用。

👍 赞

 王明朗 回复 信息门下走狗 4 天前

如果是回归问题，怎样直观的去想哪？

👍 赞

 信息门下走狗 回复 王明朗 4 天前

对分类问题，加权是在判决函数变化剧烈的点上增加权重，如果照此理解，对于回归问题，是不是就是在回归函数导数大的位置对应的训练数据点会被加权，而回归函数平缓的地方权重降低呢？

👍 赞

展开其他 1 条回复

 酱油妹 5 天前


感谢分享，学习了。

👍 赞

 张馨宇 5 天前

感觉还是想办法多弄点无偏数据最靠谱

👍 赞

 杜晨壮 回复 张馨宇 5 天前

高级数据标注师hhh

👍 1

 信息门下走狗 回复 张馨宇 5 天前

是，权重这个东西作用不太大，因为其目标是对分界面附近的点进行加权，而实际上由于对抗样本的存在，几乎所有点都在分界面附近，大家的不同就是离开分界面的距离略有差别，以及周围临近非界面的数目有差别，但是在数据缺乏的情况下，只是调整权重作用应该不大。

👍 1

展开其他 2 条回复

 信息门下走狗 5 天前

这个权重选择的目标是什么？似乎是使得在最优的权重上，对权重的扰动不会导致对系统参数的扰动，就是达到系统参数的稳定点。所以，调整权重是在寻找最稳定的网络，寻找对数据扰动最鲁棒的网络结构。但是在没有增加有效训练数据的前提下，似乎对于系统性能的提高不会有太大帮助，毕竟这个调整还是在局部进行的，并不能从根本上改变系统映射。

👍 赞

 Zhang Rui 5 天前

Sample complexity 和 PAC learning已经很成熟了啊；70年代就建立了啊；怎么能说“忽略对数据本身的研究”？

👍 赞

- 

信息门下走狗 回复 Zhang Rui

5 天前
- AI领域重新发明的轮子太多了，连自控中的adjoint calculation都被重新发明了一次，还被当作重大进展呢。
-  赞
- 

Naiyan Wang (作者) 回复 Zhang Rui

5 天前
- 在context中理解这句话吧😏 非要准确说，就是忽略如何对数据本身操作提升性能
-  赞
- 展开其他 2 条回复
- 

信息门下走狗

5 天前
- 仔细想一下，这个东西叫data dropout还真是有点道理。普通dropout是在一个固定数据下，达到对系统参数扰动下代价函数稳定，从而对数据权重的导数为0，这里的方法是在一个固定参数的网络上，对数据权重进行扰动时代价函数稳定，对网络参数的导数为0。
-  2
- 

Ruixin Zhang

5 天前
- 最近正好遇到类似的问题，雪中送炭[赞]
-  赞
- 

满分大王

4 天前
- 好干货啊!!!
-  赞
- 

2prime

4 天前
- 张志华老师今年iclr有篇投稿也做了类似的事情感觉DRO那篇 我发给作者看看
-  赞
- 

right jack

10 小时前
- [2]的工作宏观去看，感觉就是用验证集对训练集的样本做Attention。。。tracking中16年Martin的工作 “Adaptive Decontamination of the Training Set: A Unified Formulation for Discriminative Visual Tracking” 好像就是对在线样本做这样的事情。相应地，工作[2]中的验证集在这里就是新一帧的目标跟踪结果
-  1
- 

Chenyang Huang

9 小时前
- 我其实更关心有偏数据集的reweight
-  赞