

观点 | 小心训练模型，数据少也可以玩转深度学习

2017年06月11日 15:15:21 机器之心

0

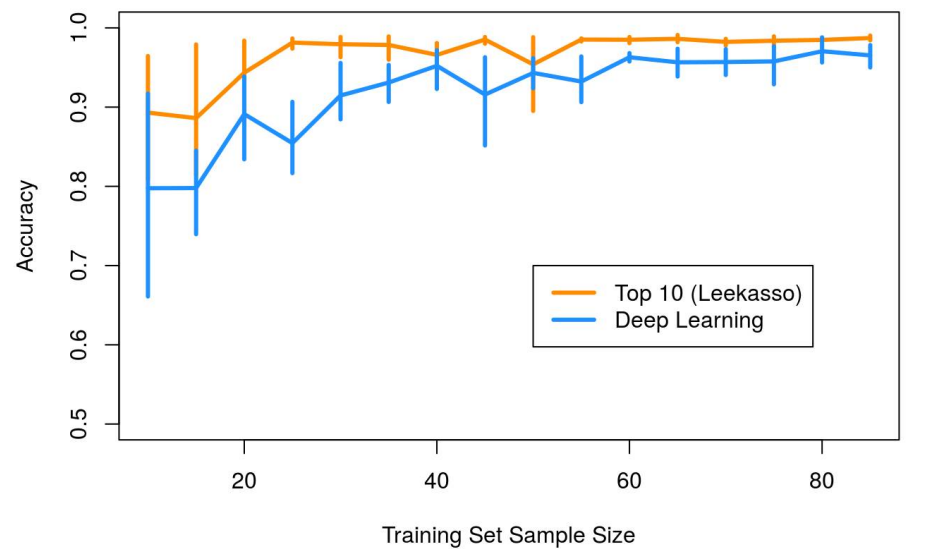
选自Github

作者：Andrew L. Beam

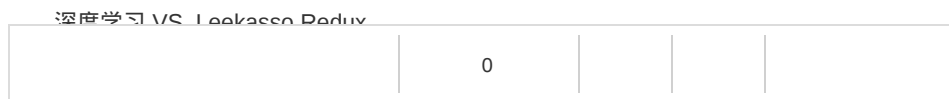
机器之心编译


最近，Jeff Leek 在 Simply Stats 上发表了一篇题为「如果你的数据量不够大就不要使用深度学习」（Don't use deep learning your data isn't that big）的文章（链接见文末），认为只有获得了谷歌、Facebook 这样规模的数据才有资格做深度学习。对于这点 Andrew L. Beam（本文作者）并不反对，他认为这使我们清楚地意识到深度学习并不是一种万能的灵药；但是，虽然 Beam 同意其核心观点，但是其还有很多不明确或不清晰的地方，并且 Beam 认为只要小心地训练模型，就能在小数据设置中使用深度学习。机器之心对该文进行了编译，原文链接请见文末。

Jeff Leek 采用两种方法基于 MNIST 数据集对手写字体进行分类。他对比了五层神经网络（激活函数使用的是 hyperbolic tangent）的系统 and Leekasso，Leekasso 仅仅使用了带最小边际 p-value 的 10 块像素。他惊讶地表明，在使用少量样本时，Leekasso 要比神经网络性能更加出色。



难道如果你的样本量小于 100，就因为模型会过拟合并且会得出较差的性能而不能使用深度学习？可能情况就是如此，深度学习模型十分复杂，并且有许多训练的技巧，我总感觉缺乏模型收敛性/复杂度训练也许才是性能较差的原因，而不是过拟合。



机器之心

专业的人工智能媒体与产业服务平台。

热文排行

- 日榜周榜月榜
- 国务院领导说出了大实话：房地产泡沫或..
 - 为什么越是有钱人，越要贷款买房？
 - 30万亿资金正在撤出楼市找韭菜，城市家..
 - 20家银行停止房贷！为拿到贷款，房奴要..
 - 比去年更惨！刚需今年买房要过三道关！
 - 10年前中国房价最高的县城，现在变成
 - 监管层发动组合拳 6月行情风向已变？
 - 旅美台湾人上海归来后说：台湾下一代自..
 - 15股中期业绩暴增 机构新猎物曝光
 - 嘲讽开发商：购房还有刚需？别做梦了韭..



首先第一件事就是建立一个使用该数据集的深度学习模型，也就是现代版的多层感知机（MLP）和卷积神经网络（CNN）。如果 Leek 的文章是正确的话，那么当只有少量样本时，这些模型应该会产生严重的过拟合。

我们构建了一个激活函数为 RELU 的简单 MLP 和一个像 VGG 那样的卷积模型，然后我们比较它们和 Leekasso 性能的差异。

所有的代码都可下载：https://github.com/beamandrew/deep_learning_works/blob/master/mnist.py

多层感知机模型是非常标准的：

```
def get_mlp(n_classes):
    model = Sequential()
    model.add(Dense(128, activation='relu', input_shape=(784,)))
    model.add(Dropout(0.5))
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(n_classes, activation='softmax'))
    model.compile(optimizer='Adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model
```

CNN 模型也和以前的十分相似：

```
def get_cnn(n_classes):
    model = Sequential()
    model.add(Convolution2D(32, 3, 3, activation='relu', input_shape=(28, 28, 1)))
    model.add(Convolution2D(32, 3, 3, activation='relu'))
    model.add(MaxPooling2D())
    model.add(Convolution2D(64, 3, 3, activation='relu'))
    model.add(Convolution2D(64, 3, 3, activation='relu'))
    model.add(MaxPooling2D())
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(n_classes, activation='softmax'))
    model.compile(optimizer='Adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model
```

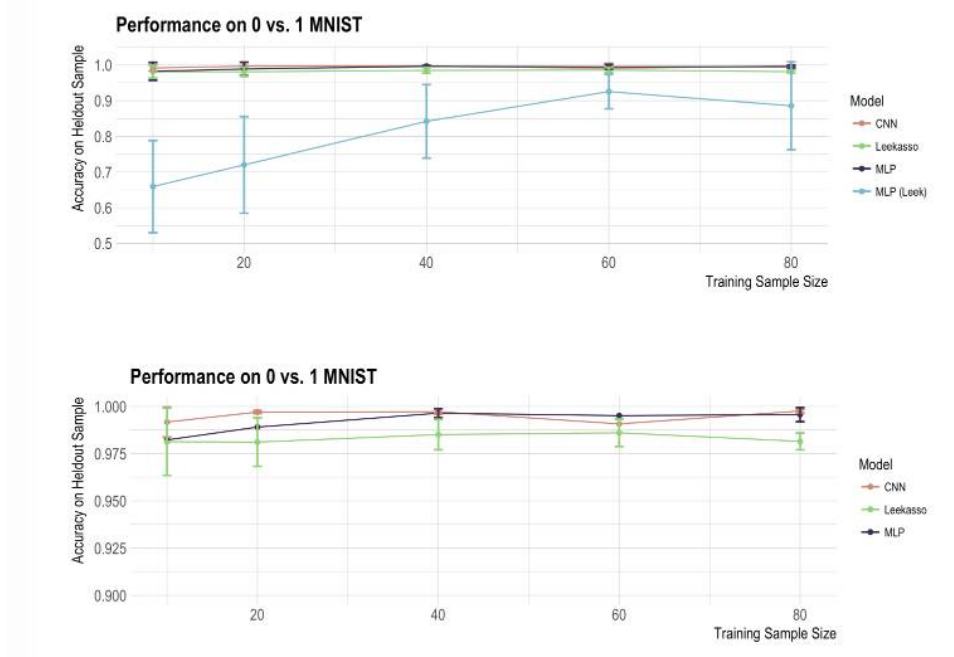
作为参考，MLP 大约有 12 万个参数，而 CNN 大约有 20 万个参数。根据原文的假设，当我们有这么多的参数和少量样本时，模型好像真的会出错。

我们尽可能地靠近原始分析，我们使用了 5 层交叉验证（5-fold cross validation），但使用了标准 MNIST 测试集进行评估（大约有 2000 张测试样本）。我们将测试集分为两部分，上半部分用于评估训练过程的收敛，而下半部分用于测量样本预测的准确度。我们甚至没有调整这些模型，对于大多数超参数，仅仅只是使用合理的默认值。

我们尽可能地重新构建了原文中 Leekasso 和 MLP 的 Python 版本。代码可以在此处下载：https://github.com/beamandrew/deep_learning_works/blob/master/mnist.py

以下是每个模型的样本精度：

	0			
--	---	--	--	--



这两个模型的精度和原来的分析有很大的不同，原始分析中对小样本使用 MLP 仍然有很差的效果，但我们的神经网络在各种样本大小的情况下都可以达到非常完美的精度。

为什么会这样？

众所周知，深度学习模型的训练往往对细节要求极高，而知道如何「调参」是一件非常重要的技能。许多超参数的调整是非常具体的问题（特别是关于 SGD 的超参数），而错误地调参会导致整个模型的性能大幅度下降。如果你在构建深度学习模型，那么就一定要记住：模型的细节是十分重要的，你需要当心任何看起来像深度学习那样的黑箱模型。

下面我对原文模型出现问题的猜测：

激活函数是十分重要的，而 tanh 神经网络又难以训练。这也就是为什么激活函数已经大量转而使用类似「RELU」这样的函数。

确保随机梯度下降是收敛的。在原始比较中，模型只训练了 20 个 epoch，这可能是不够的。因为当 $n=10$ 个样本时，20 个 epochs 仅仅只有 $20 \times 10=200$ 次的梯度迭代更新。而遍历全部的 MNIST 数据集大概相当于 6 万次梯度更新，并且更常见的是遍历数百到数千次（大约百万次梯度更新）。如果我们仅仅执行 200 次梯度更新，那么我们需要比较大的学习率，否则模型就不会收敛。h2o.deeplearning() 的默认学习率是 0.005，这对于少量的更新次数来说太小了。而我们使用的模型需要训练 200 个 epoch，并且在前 50 次 epoch 中，我们能看到样本精度有很大的一个提高。因此我猜测模型不收敛可以解释两者样本精度的巨大差别。

经常检查超参数的默认值。Keras 之所以这么优秀，是因为其默认参数值通常反映了当前的最佳训练，但同时我们也需要确保选择的参数符合我们的问题。

不同的框架可能得出很不一样的结果。我尝试使用原 R 代码去观察能不能得到相似的结果。然而，我并不能使用 h2o.deeplearning() 函数得出一个优异的结果。我猜测可能是和其使用的优化过程有关，其好像使用的是弹性均值 SGD 以计算多个结点而加速训练。我不知道当你仅有少量样本数据时会不会出现故障，但我认为可能性是很大的。

幸好，RStudio 那些人太好了，他们刚刚发布了 Keras 的 R 接口：<https://rstudio.github.io/keras/> 这样我就可以完全用 R 语言重建我的 Python 代码了。我们之前使用 MLP 用 R 实

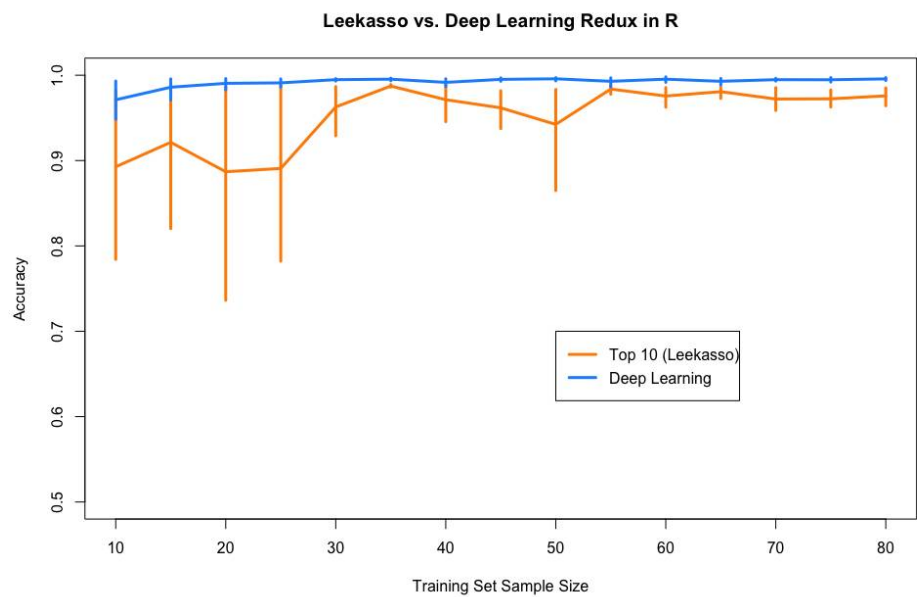
	0			
--	---	--	--	--

```
model <- keras_model_sequential()
model %>%
  layer_dense(units = 128, input_shape = c(784), activation = 'relu') %>%
  layer_dropout(0.5) %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dropout(0.5) %>%
  layer_dense(units = 1, activation = 'sigmoid') %>%
  compile(
    optimizer = optimizer_adam(lr=0.001),
    loss = 'binary_crossentropy',
    metrics = c('accuracy')
  )

model %>% fit(as.matrix(training0[, -1]), as.matrix(training0[, 1]),
  verbose=0, epochs=200, batch_size=1)

score <- model %>% evaluate(as.matrix(testing[, -1]), as.matrix(testing[, 1]), batch_size=128, verbose=0)
deep[b,i] <- score[[2]]
```

我将这个放进了 Jeff 的 R 代码中，并重新生成了原来的图表。我对 Leekasso 进行了一点修改。原来的代码使用了 lm()（即线性回归），我觉得很奇怪，所以我切换成了 glm()（即 logistic 回归）。新的图表如下所示：



深度学习真是厉害了！一个类似的现象可能能够解释 Leekasso 的 Python 和 R 版本之间的不同。Python 版本的 logistic 回归使用了 liblinear 作为其解算器，我认为这比 R 默认的解算器更加可靠一点。这可能会有影响，因为 Leekasso 选择的变量是高度共线性的（collinear）。

这个问题太简单了，以致于不能说明什么有意义的东西。我重新运行了 Leekasso，但仅使用了最好的预测器，其结果几乎完全等同于全 Leekasso。实际上，我确定我可以做出一个不使用数据的且具有高准确度的分类器。只需要取其中心像素，如果是黑色，则预测 1，否则就预测 0，正如 David Robinson 指出的那样：

		0			
--	--	---	--	--	--



David 还指出，大多数数字对（pairs of numbers）都可以由单个像素进行分类。所以，这个问题很可能不能给我们带来任何关于「真实」小数据场景的见解，我们应当对其结论保持适当的怀疑。

关于深度学习为什么有效的误解

最终，我想要重新回到 Jeff 在文中所提出的观点，尤其是这个声明：

问题在于：实际上仅有少数几个企业有足够数据去做深度学习，[...] 但是我经常思考的是，在更简单的模型上使用深度学习的主要优势是如果你有大量数据就可以拟合大量的参数。

这篇文章，尤其是最后一部分，在我看来并不完整。很多人似乎把深度学习看成一个巨大的黑箱，有大量可以学习任何函数的参数，只要你有足够的数据。神经网络当然是极其灵活的，这种灵活性正是其成功原因的一部分，但不是全部，不是吗？

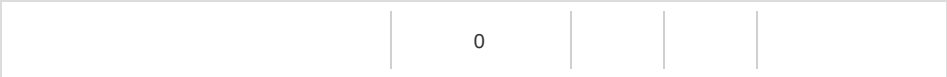
毕竟，这种超级灵活的模型在机器学习和统计学中有着 70 多年的发展历史。我并不认为神经网络是先验（priori）的，我也不认为比同等复杂度的其他算法更灵活。

下面是我对其成功所作的原因总结：

在偏差/方差折衷中一切都是一个练习。更明白地讲，我认为 Jeff 真正在做的辩驳是关于模型复杂度和偏差/方差折衷。如果你没有很多数据，很可能训练一个简单模型（高偏差/低方差）要比复杂模型（低偏差/高方差）效果更好。客观来讲，在大多数情况下这是一个好建议，然而...

神经网络有很多技术来防范过拟合。神经网络有很多参数，按照 Jeff 的观点如果我们没有足够的数据去可靠地评估这些参数值，将会导致高方差。我们清楚地意识到了这个问题，并且开发了很多降低方差的技术。比如 dropout 结合随机梯度下降导致了一个像 bagging 一样糟糕的处理，但是这是发生在网络参数上，而不是输入变量。方差降低技术（比如 dropout）以其他模型难以复制的方式被加进了训练程序。这使得你可以真正训练大模型，即使没有太多数据。

深度学习允许你轻易地把问题的具体约束直接整合进模型以降低方差。这是我想说明的最重要的一点，也是我们以前经常忽视的一点。由于其模块化，神经网络使你可以真正整合，极大降低模型方差的强约束（先验）。最好的一个实例是卷积神经网络。在 CNN 中，我们实际上把图像的属性编码进模型本身。例如，当我们指定一个大小为 3x3 的过滤器时，实际上是在直接告诉网络本地连接的像素的小集群将包含有用的信息。此外，我们还可以把图像的平移和旋转不变性直接编码进模型。所有这些都将模型偏差至图像属性，以极大地降低方差，提升预



你并不需要拥有谷歌量级的数据。以上所述意味着即使人均 100 到 1000 个样本也能从深度学习 中受益。通过所有这些技术，我们可以改善方差问题，而且依然可以从其灵活性中受益。你甚至可以通过迁移学习来创建其他工作。

总结一下，我认为上述原因很好地解释了为什么深度学习在实践中奏效，打破了深度学习需要大量参数和数据的假设。最后，本文并不是想说 Jeff 的观点错了，而是旨在提供一个不同的新视角，为读者带来启发。



原文链接：[http://beamandrew.github.io/deeplearning/2017/06/04/deep_learning_works.h](http://beamandrew.github.io/deeplearning/2017/06/04/deep_learning_works.html)
tml

Jeff Leek 在 Simply Stats 的文章链接：[https://simplystatistics.org/2017/05/31/deeplearn](https://simplystatistics.org/2017/05/31/deeplearning-vs-leekasso/)
ing-vs-leekasso/)

本文为机器之心编译，转载请联系本公众号获得授权。

加入机器之心（全职记者/实习生）：hr@jiqizhixin.com

投稿或寻求报道：editor@jiqizhixin.com

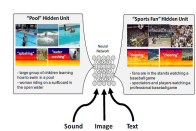
广告商务合作：bd@jiqizhixin.com

点击阅读原文，查看机器之心官网↓↓↓

■

作者历史文章

学界 | 让机器耳濡目染：MIT提出跨模态机器学习模型



选自arXiv机器之心编译作者：YusufAytar等人参与：李泽南不变性表示（invariantrepresentation）是视觉、听觉和语言模型的核心，它[详细]

2017年 06月11日 15:15

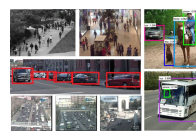
资源 | PyTorch第一版中文文档发布



机器之心报道参与：黄小天近日，使用GPU和CPU优化的深度学习张量库PyTorch上线了其第一版中文文档，内容涵盖介绍、说明、Package参考、torchvi[详细]

2017年 06月10日 13:15

资源 | 神经网络目标计数概述：通过Faster R-CNN实现当前最佳的目标计



选自SoftwareMill机器之心编译作者：KrzysztofGrajek参与：黄小天在机器学习中，精确地计数给定图像或视频帧中的目标实例是很困难的一个问题。[详细]

	0			
--	---	--	--	--

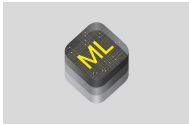
学界 | Facebook 新研究：大批量SGD准确训练ImageNet仅需1小时



选自arXiv机器之心编译参与：蒋思源由于近来互联网数据越来越大，深度学习模型越来越复杂，执行训练的时间也越来越长。因此近日Facebook提出了一种将批量大小提[详细]

2017年 06月09日 11:05

教程 | 如何使用Swift在iOS 11中加入原生机器学习视觉模型



选自Hackernoon机器之心编译作者：AlexWulff参与：侯韵楚、李泽南随着WWDC大会上iOS11的发布，苹果终于推出了原生机器学习和机器视觉框架，由[详细]

2017年 06月09日 11:05

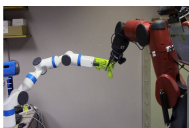
重磅 | 波士顿动力被软银收购，「被丰田收购」传言告破



机器之心报道机器之心编辑部Alphabet（谷歌）想要甩手波士顿动力（BostonDynamics）的传言已经持续了很长时间，而接手者基本上已经被认为是丰田了，[详细]

2017年 06月09日 11:05

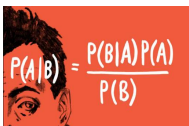
业界 | CMU和谷歌联手研制左右互搏的对抗性机器人



选自IEEESpectrum机器之心编译作者：EvanAckerman参与：蒋思源、SmithCMU和谷歌研究者正在使用基于博弈论和深度学习的对抗性训练策略来提[详细]

2017年 06月09日 11:05

从贝叶斯角度，看深度学习的属性和改进方法



选自arXiv.org机器之心编译参与：蒋思源、吴攀深度学习是一种高效的非线性高维数据处理方法，它可以更自然地解释为一种工程或算法，而本论文希望从贝叶斯的角度将[详细]

2017年 06月08日 12:15

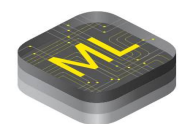
课程 | 来自硅谷的人工智能工程师直通车：打造工业级AI工程师！



人工智能是目前发展最快的领域之一。根据职业发展平台Paysa的专业预测，人工智能在未来5年，将发展成为一个价值160亿美元的市场。仅2017年，美国公司就计划在[详细]

2017年 06月08日 12:15

资源 | 用苹果Core ML实现谷歌移动端神经网络MobileNet



选自GitHub机器之心编译作者：MatthijsHolleman参与：李泽南6月5日开幕的WWDC2017开发者大会上，苹果正式推出了一系列新的面向开发者的[详细]

2017年 06月08日 12:15

[关于头条](#) | [如何入驻](#) | [发稿平台](#) | [奖励机制](#) | [版权声明](#)
[用户协议](#) | [帮助中心](#) © 1996-2015 SINA Corporation, All Rights Reserved

	0			
--	---	--	--	--