

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

有趣的机器学习



Python基础 ▼



机器学习 ▼



数据处理 ▼



其他 ▼



切换到 优酷 视频

(Chrome无法播放优酷? 网址框输入"chrome://plugins/", 勾选允许 Flash Player. 实在不行? 请 [点击这里](#))

晚上十点开始的副业收入 - 一份特别收入，每晚十点准时开始

每周都有至少一笔收入自动打入你的账户，无需工作一天，睡觉时都在赚钱 转到 cwziyouren.com

广告



« 上一个

下一个 »

批标准化 (Batch Normalization)

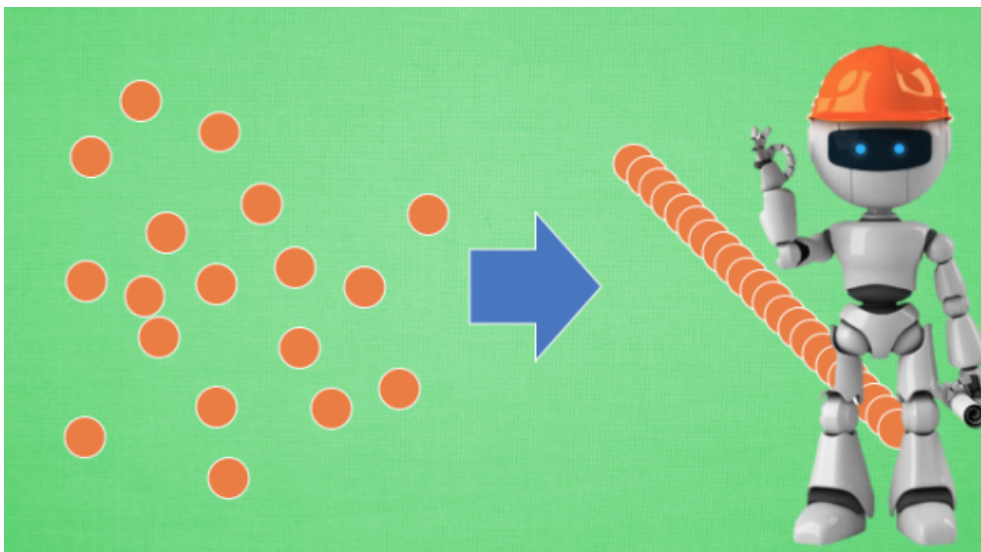
作者: Morvan 编辑: Morvan

- 学习资料:
 - [Tensorflow 使用 Batch normalization](#)

今天我们会来聊聊批标准化 Batch Normalization.

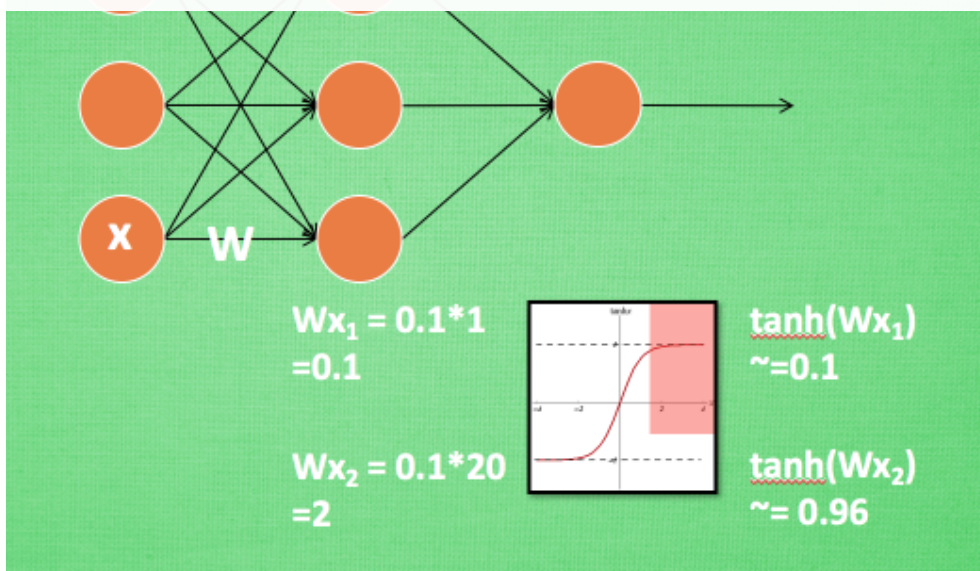
注: 本文不会涉及太多数学推导. 大家可以在很多其他地方找到优秀的数学推导文章.

普通数据标准化

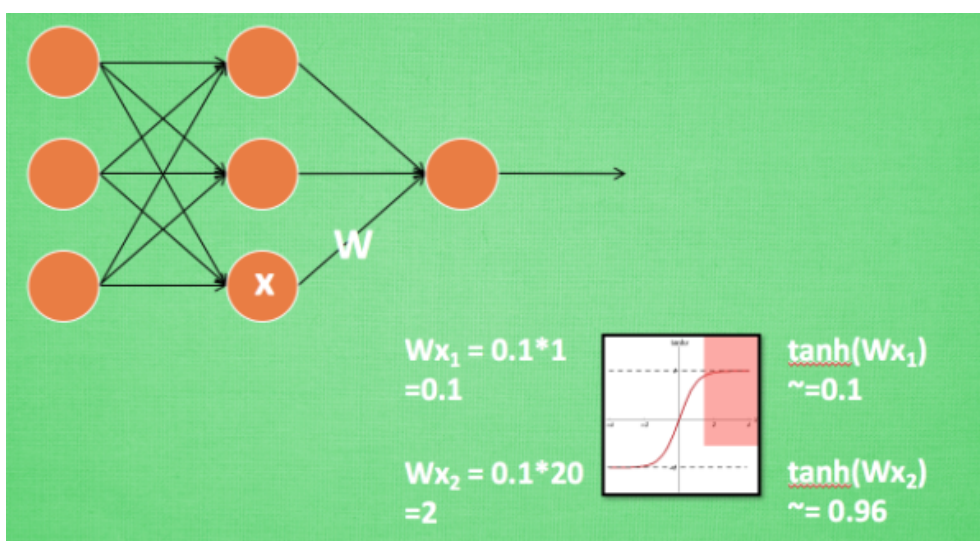


Batch Normalization, 批标准化, 和普通的数据标准化类似, 是将分散的数据统一的一种做法, 也是优化神经网络的一种方法. 在之前 Normalization 的简介视频中我们一提到, 具有统一规格的数据, 能让机器学习更容易学习到数据之中的规律.

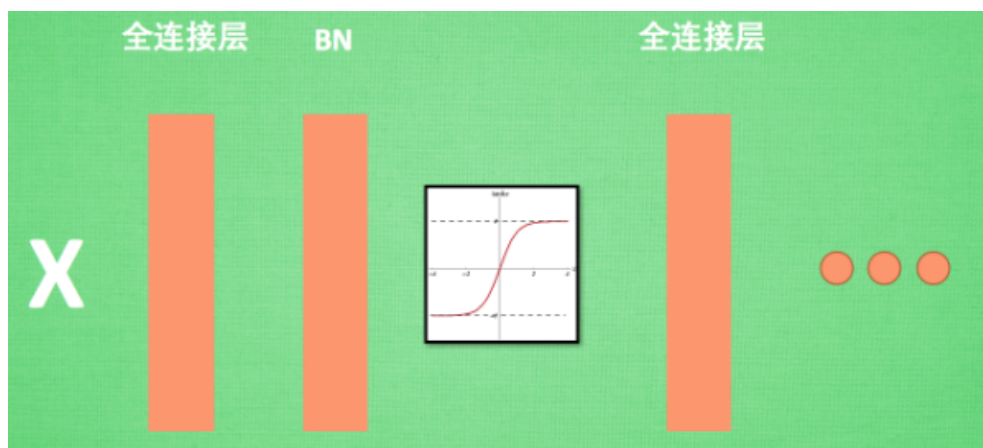
每层都做标准化



在神经网络中, 数据分布对训练会产生影响. 比如某个神经元 x 的值为1, 某个 Weights 的初始值为 0.1, 这样后一层神经元计算结果就是 $Wx = 0.1$; 又或者 $x = 20$, 这样 Wx 的结果就为 2. 现在还不能看出什么问题, 但是, 当我们加上一层激励函数, 激活这个 Wx 值的时候, 问题就来了. 如果使用像 \tanh 的激励函数, Wx 的激活值就变成了 ~ 0.1 和 ~ 1 , 接近于 1 的部分已经处在了激励函数的饱和阶段, 也就是如果 x 无论再怎么扩大, \tanh 激励函数输出值也还是接近 1. 换句话说, 神经网络在初始阶段已经不对那些比较大的 x 特征范围敏感了. 这样很糟糕, 想象我轻轻拍自己的感觉和重重打自己的感觉居然没什么差别, 这就证明我的感官系统失效了. 当然我们是可以用之前提到的对数据做 normalization 预处理, 使得输入的 x 变化范围不会太大, 让输入值经过激励函数的敏感部分. 但刚刚这个不敏感问题不仅仅发生在神经网络的输入层, 而且在隐藏层中也经常会发生.



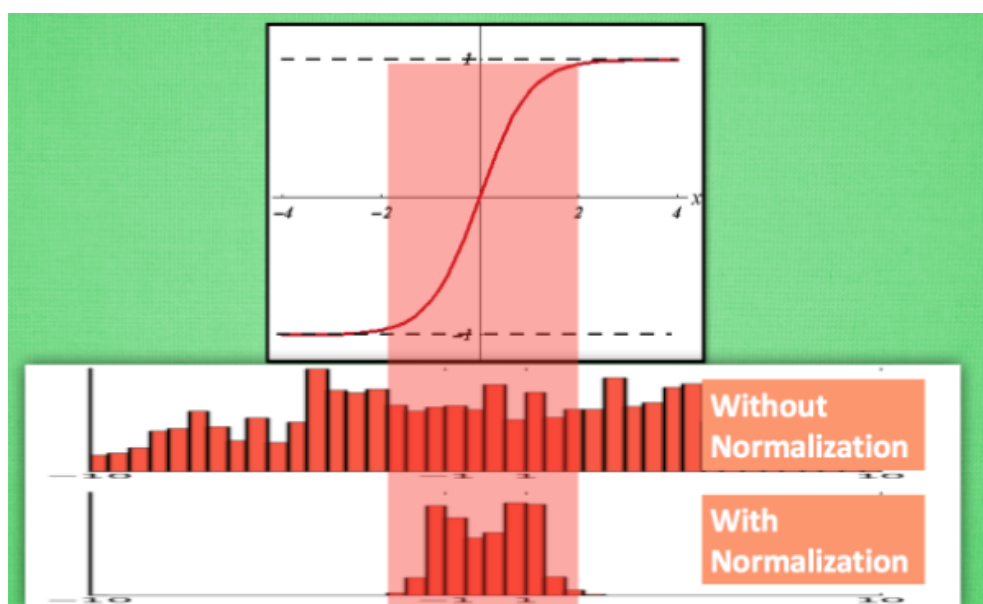
只是时候 x 换到了隐藏层当中, 我们能不能对隐藏层的输入结果进行像之前那样的 normalization 处理呢? 答案是可以的, 因为大牛们发明了一种技术, 叫做 batch normalization, 正是处理这种情况.



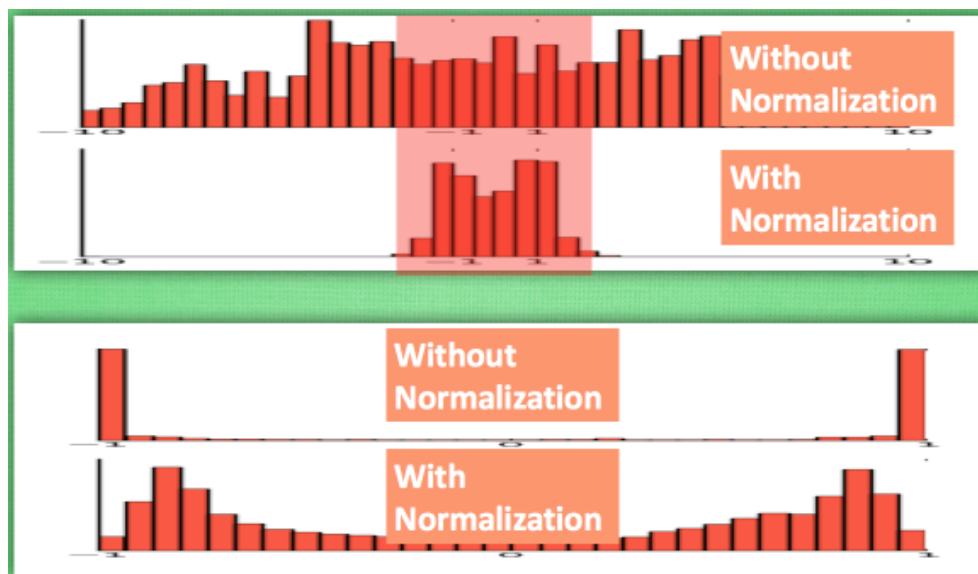
Batch normalization 的 batch 是批数据, 把数据分成小批小批进行 stochastic gradient descent. 而且在每批数据进行前向传递 forward propagation 的时候, 对每一层都进行 normalization 的处理,

BN 效果

Batch normalization 也可以被看做一个层面. 在一层层的添加神经网络的时候, 我们先有数据 X , 再添加全连接层, 全连接层的计算结果会经过 激励函数 成为下一层的输入, 接着重复之前的操作. Batch Normalization (BN) 就被添加在每一个全连接和激励函数之间.



之前说过, 计算结果在进入激励函数前的值很重要, 如果我们不单单看一个值, 我们可以说, 计算结果值的分布对于激励函数很重要. 对于数据值大多分布在这个区间的数据, 才能进行更有效的传递. 对比这两个在激活之前的值的分布. 上者没有进行



没有 normalize 的数据使用 tanh 激活以后, 激活值大部分都分布到了饱和阶段, 也就是大部分的激活值不是-1, 就是1, 而 normalize 以后, 大部分的激活值在每个分布区间都还有存在. 再将这个激活后的分布传递到下一层神经网络进行后续计算, 每个区间都有分布的这一种对于神经网络就会更加有价值. Batch normalization 不仅仅 normalize 了一下数据, 他还进行了反 normalize 的手续. 为什么要这样呢?

BN 算法

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1 \dots x_m\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

标准化工序

反标准化工序

我们引入一些 batch normalization 的公式. 这三步就是我们在刚刚一直说的 normalization 工序, 但是公式的后面还有一个反向操作, 将 normalize 后的数据再扩展和平移. 原来这是为了让神经网络自己去学着使用和修改这个扩展参数 gamma, 和平移参数 β , 这样神经网络就能自己慢慢琢磨出前面的 normalization 操作到底有没有起到优化的作用, 如果没有起到作用, 我就使用 gamma 和 belt 来抵消一些 normalization 的操作.

莫烦PYTHON

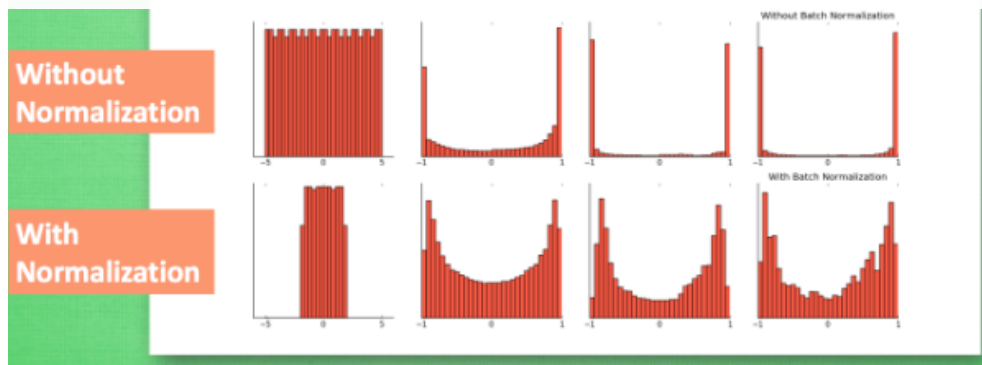
教程 ▼

关于我

赞助

大家说

15 篇



最后我们来看看一张神经网络训练到最后, 代表了每层输出值的结果的分布图. 这样我们就能一眼看出 Batch normalization 的功效啦. 让每一层的值在有效的范围内传递下去.

如果你觉得这篇文章或视频对你的学习很有帮助, 请你也分享它, 让它能再次帮助到更多的需要学习的人.

莫烦没有正式的经济来源, 如果你也想支持 **莫烦Python** 并看到更好的教学内容, 请拉倒屏幕最下方, 赞助他一点点, 作为鼓励他继续开源的动力.

[<< 上一个](#)[下一个 >>](#)

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说



撰写评论

使用社交网站账户登录

或使用来必力便捷评论

邮件

写评论

总评论数 5

按时间正序



一条行走在沙漠中的鱼 2017年3月21日 · 已分享的SNS(1)

感谢您的回答，训练出来的参数是在caffemodel里吗？
那里面不是提出来的参数是weight和bias吗、刚接触不太懂，望指教！

0

0 0



一条行走在沙漠中的鱼 2017年3月20日 · 已分享的SNS(1)

莫大神，BN那四个公式里的参数应该去哪里找？
我最近在编写BN函数的MATLAB版本，有点不太会写，望大神指点。谢谢！

1

0 0



莫烦Python 2017年3月20日

@一条行走在沙漠中的鱼 里面的参数都是 train 出来的,不用手动调

0

0 0



NO NICKNAME 2017年3月8日

莫大神，batch normalization能减少overfitting吗？

1

0 0



莫烦Python 2017年3月8日

@NO NICKNAME 好像是有一定的 regularization 的作用, 就像 dropout 那样,
因为它没那么强烈的 depend on 一个固定的结构.

0

0 0

来必力是？ | 询问

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

支持 让教学变得更优秀

点我 赞助 莫烦

关注我的动向:

[Youtube频道](#) [优酷频道](#) [Github](#) [微博](#)

Email: morvanzhou@hotmail.com

© 2016 morvanzhou.github.io. All Rights Reserved