Fate_fjh的博客

卷积神经网络CNN（2）—— BN(Batch Normalization) 原理与使

标签： cnn 神经网络 BN

2016-11-28 11:56 10917人阅读 评论(11) 收藏 举报

分类：

深度学习（8）

目录(?) [+]

前言

Batch Normalization是由google提出的一种训练优化方法。参考论文：Batch Normalization Accelerating Deep Network Training by Reducing Internal Covariate Shift
个人觉得BN层的作用是加快网络学习速率，论文中提及其它的优点都是这个优点的副产品。
网上对BN解释详细的不多，大多从原理上解释，没有说出实际使用的过程，这里从what, why, how三个角度去解释BN。

What is BN

Normalization是数据标准化（归一化，规范化），Batch 可以理解为批量，加起来就是批量标准化。
先说Batch是怎么确定的。在CNN中，Batch就是训练网络所设定的图片数量batch_size。

Normalization过程，引用论文中的解释：

关闭

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

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

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

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

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

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

输入：输入数据x1..xm（这些数据是准备进入激活函数的数据）
计算过程中可以看到，
1.求数据均值；
2.求数据方差；
3.数据进行标准化（个人认为称作正态化也可以）
4.训练参数γ，β
5.输出y通过γ与β的线性变换得到原来的数值
在训练的正向传播中，不会改变当前输出，只记录下γ与β。

在反向传播的时候，根据求得的γ与β通过链式求导方式，求出学习速率以至改变权值

$$\frac{\partial \ell}{\partial \widehat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma$$

$$\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2}(\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}$$

$$\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} = \left(\sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}\right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_{\mathcal{B}})}{m}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

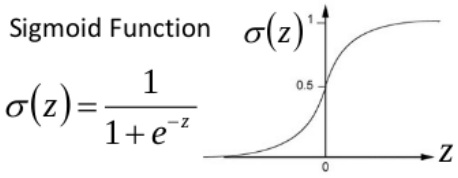
$$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i} \cdot \widehat{x}_i$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i}$$

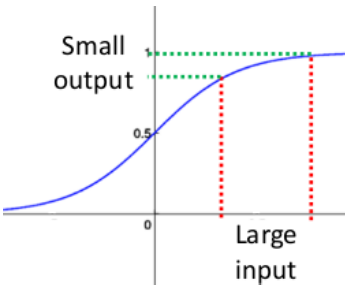
Why is BN

关闭

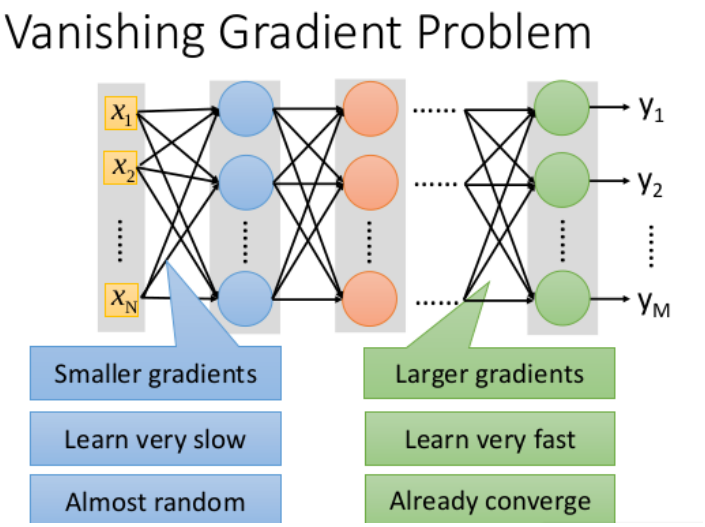
解决的问题是梯度消失与梯度爆炸。
关于*梯度消失*，以sigmoid函数为例子，sigmoid函数使得输出在[0,1]之间。



事实上x到了一定大小，经过sigmoid函数的输出范围就很小了，参考下图



如果输入很大，其对应的斜率就很小，我们知道，其斜率（梯度）在反向传播中是权值学习速率。所以就会出现如下的问题，



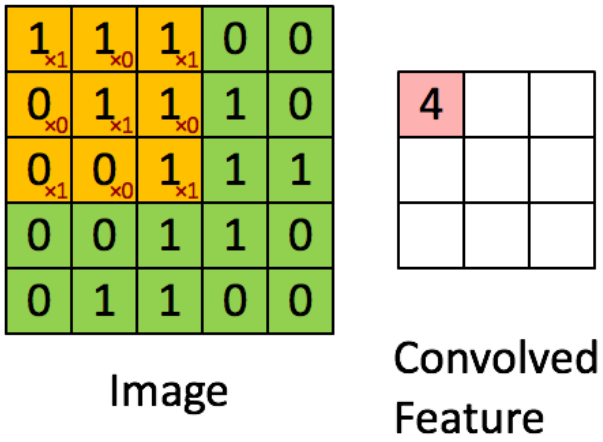
在深度网络中，如果网络的激活输出很大，其梯度就很小，学习速率就很慢。假设每层学习梯度都小于最大值0.25，网络有n层，因为链式求导的原因，第一层的梯度小于0.25的n次方，所以学习速率就慢，对于最后一层只需对自身求导1次，梯度就大，学习速率就快。
这会造成的影响是在一个很大的深度网络中，浅层基本不学习，权值变化小，后面几层一直在学习，结果就是，后面几层基本可以表示整个网络，失去了深度的意义。

关闭

关于*梯度爆炸*，根据链式求导法，
第一层偏移量的梯度=激活层斜率1x权值1x激活层斜率2x...激活层斜率(n-1)x权值(n-1)x激活层斜率n
假如激活层斜率均为最大值0.25，所有层的权值为100，这样梯度就会指数增加。

How to use BN

先解释一下对于图片卷积是如何使用BN层。



这是文章卷积神经网络CNN（1）中5x5的图片通过valid卷积得到的3x3特征图（...）。将特征图里的值，作为BN的输入，也就是这9个数值通过BN计算并保存γ与β，通过γ与β使得输出与输入不变。假设输入的batch_size为m，那就有m*9个数值，计算这m*9个数据的γ与β并保存。正向传播过程如上述，对于反向传播就是根据求得的γ与β计算梯度。

这里需要着重说明2个细节：

1.网络训练中以batch_size为最小单位不断迭代，很显然，新的batch_size进入网络，机会有新的γ与β，因此，在BN层中，有*总图片数/batch_size*组γ与β被保存下来。

2.图像卷积的过程中，通常是使用多个卷积核，得到多张特征图，对于多个的卷积核需要保存多个的γ与β。

结合论文中给出的使用过程进行解释

关闭

Input: Network N with trainable parameters Θ;
subset of activations $\{x^{(k)}\}_{k=1}^{K}$
Output: Batch-normalized network for inference, $N_{\text{BN}}^{\text{inf}}$
1: $N_{\text{BN}}^{\text{tr}} \leftarrow N$ // Training BN network
2: **for** $k = 1 \ldots K$ **do**
3: Add transformation $y^{(k)} = \text{BN}_{\gamma^{(k)},\beta^{(k)}}(x^{(k)})$ to $N_{\text{BN}}^{\text{tr}}$ (Alg. 1)
4: Modify each layer in $N_{\text{BN}}^{\text{tr}}$ with input $x^{(k)}$ to take $y^{(k)}$ instead
5: **end for**
6: Train $N_{\text{BN}}^{\text{tr}}$ to optimize the parameters $\Theta \cup \{\gamma^{(k)}, \beta^{(k)}\}_{k=1}^{K}$
7: $N_{\text{BN}}^{\text{inf}} \leftarrow N_{\text{BN}}^{\text{tr}}$ // Inference BN network with frozen // parameters
8: **for** $k = 1 \ldots K$ **do**
9: // For clarity, $x \equiv x^{(k)}, \gamma \equiv \gamma^{(k)}, \mu_{\mathcal{B}} \equiv \mu_{\mathcal{B}}^{(k)}$, etc.
10: Process multiple training mini-batches $\mathcal{B}$, each of size m, and average over them:

$$\text{E}[x] \leftarrow \text{E}_{\mathcal{B}}[\mu_{\mathcal{B}}]$$
$$\text{Var}[x] \leftarrow \frac{m}{m-1}\text{E}_{\mathcal{B}}[\sigma_{\mathcal{B}}^2]$$

11: In $N_{\text{BN}}^{\text{inf}}$, replace the transform $y = \text{BN}_{\gamma,\beta}(x)$ with $y = \frac{\gamma}{\sqrt{\text{Var}[x]+\epsilon}} \cdot x + \left(\beta - \frac{\gamma\,\text{E}[x]}{\sqrt{\text{Var}[x]+\epsilon}}\right)$
12: **end for**

Algorithm 2: Training a Batch-Normalized Network

输入：待进入激活函数的变量
输出：
1.对于K维的输入，假设每一维包含m个变量，所以需要K个循环。每个循环中按照上面所介绍的方法计算γ与β。这里的K维，在卷积网络中可以看作是卷积核个数，如网络中第n层有64个卷积核，就需要计算64次。
需要注意，在正向传播时，会使用γ与β使得BN层输出与输入一样。
2.在反向传播时利用γ与β求得梯度从而改变训练权值（变量）。
3.通过不断迭代直到训练结束，求得关于不同层的γ与β。如网络有n个BN层，每层根据batch_size决定有多少个变量，设定为m，这里的mini-batcherB指的是*特征图大小*batch_size*，即*m=特征图大小*batch_size*，因此，对于batch_size为1，这里的m就是每层特征图的大小。
4.不断遍历训练集中的图片，取出每个batch_size中的γ
除以图片数量得到平均直，并对其做无偏估计直作为每一层的E[x]与Var[x]。
5.在预测的正向传播时，对测试数据求取γ与β，并使用该层的E[x]与Var[x]，通过图中11:所表示的公式计算BN层输出。
注意，在预测时，BN层的输出已经被改变，所以BN层在预测的作用体现在此处

至此，BN层的原理与使用过程就解释完毕，给出的解释都是本人觉得值得注意或这不容易了解的部分，如有錯漏，请指正。

顶 踩
10 0

- 上一篇 Ubuntu 16 ORB_SLAM2使用KinectV2在ROS上运行总结
- 下一篇 卷积神经网络CNN（3）—— FCN(Fully Convolutional Networks)要点解释

相关文章推荐

- cs231n学习笔记-激活函数-BN-参数优化
- MySQL在微信支付下的高可用运营--莫晓东
- BN层 LN层 WN层作用介绍
- 容器技术在58同城的实践--姚远
- 卷积神经网络CNN（4）—— SegNet
- SDCC 2017之容器技术实战线上峰会
- Batch Normalization 神经网络加速算法
- SDCC 2017之数据库技术实战线上峰会
- 深度学习之卷积神经网络CNN及ten
- 腾讯云容器服务架构实现介绍--董晓
- 深度学习（二十九）Batch Normaliz
- 微博热点事件背后的数据库运维心得--张冬进
- 解读Batch Normalization
- 关于Batch Normalization在Caffe中的使用
- [深度学习] Batch Normalization算法介绍
- CNN和RNN中如何引入BatchNorm

查看评论

东西北 6楼 2017-11-27 10:56发表

博主应该理解错了，推荐看这个http://blog.csdn.net/myarrow/article/details/51848285

Fate_fjh Re: 2017-12-25 15:54发表

回复东西北：请指出错误地方，我看了那文章，感觉没太多区别

东西北 Re: 2017-12-26 15:50发表

回复Fate_fjh：在训练的正向传播中，只有当γ=标准差，β=均值的时候才不会改变输出，说明BN有容纳原始数据模型的能力。但一般情况下输出是会改变的，将BN层的输出做为激活输入

hang_ning 5楼 2017-09-30 11:23发表

看完之后有一个问题想问，就是训练时确定了 bn的参数gama，测试的时候是不是一定要给与测试时相同batch_size的数据？ 如果改变测试的batch，或者直接一个一个测是不是会出问题

Fate_fjh Re: 2017-10-11 15:40发表

回复hang_ning：测试数据进入网络需要配合记录下来的gamma与beta做一次无偏估计，不是将测试数据一个个重新计算

angleboy8 4楼 2017-07-24 11:20发表

"需要注意，在正向传播时，会使用γ与β使得BN层输出与输入一样。"

关闭

请问输出与输入一样，指的是数值一样还是...，应该不是数值一样，
应该是数值范围一样吧。我的理解。

Fate_fjh Re: 2017-07-24 18:53发表

回复angleboy8：在正向传播过程中不会改变feature map的权重，简单可以理解为训练时的正向传播有无BN的输出是一样的

feynman233 3楼 2017-07-22 16:39发表

个人感觉博主对gama和贝塔的理解有误。1.正向传播的时候如果只是记录下这两个值，而不用这两个值去改变输出的话，那与不用bn有什么区别呢，况且，这样还怎么反向传播。2.测试的时候需要的是x的均值和方差啊，要伽马和贝塔的均值干啥，这论文里不是说的很清楚吗，这一点博主搞错了吧

Fate_fjh Re: 2017-07-25 0

回复feynman233：1.按BN论文里说的，训练时确实只记录下每个batch所对应的均值与方差，但是在运行框架例如caffe中，训练时通过加权平均（滑动系数）的方式对参数进行归一化，与论文有差异。2.测试的时候是通过所用的batch记录下来的均值、方差求取新的gama与beta，同时使用均值与方差的无偏估计，测试时进入BN层就是通过新的gama与beta调整输入参数，然后输出。 希望对你理解有帮助，欢迎相互交流。

qq_35554109 2楼 2017-07-07 09:44发表

必须赞一个

qq_27114609 1楼 2017-05-16 19:51发表

很棒，谢谢博主，解决了我很多问题，比之前很多人写的要更清楚！