

# 4. Batch Normalization

## 1. 引入

### 1. 深度学习之 Batch Normalization 的作用

神经网络在训练的时候随着网络层数的加深, 激活函数的输入值的整体分布逐渐往激活函数的取值区间上下限靠近, 从而在反向传播时低层的神经网络的梯度消失。(输入值其本不变, 极为0)

而 Batch Normalization 的作用是通过规范化手段, 将越来越偏的分布拉回到标准化的分布, 使得激活函数的输入值落在激活函数对输入比较敏感的区域, 从而使梯度变大, 加快学习收敛速度, 避免梯度消失的问题

## 2. 原理

BN (Batch Normalization) 是 Google 提出的用于解决深度网络梯度消失和梯度爆炸的问题, 可以起到一定的正则化作用。我们来说一下它的原理:

BN 操作共分为四步:

1) 计算均值:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$$

2) 计算数据方差:

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

3) 规范化:

$$x_i^* = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

4) 尺度变换与偏移:

$$y_i = \gamma \times x_i^* + \beta = \text{BN}_{\gamma, \beta}(x_i)$$

$m$ : mini-batch 中的数据个数

(线性变换)

(非线性)

前3步就是对批规范化，使得结果（各个维度）的均值为0，方差为1。

最后一步目标：让我们的网络可以学习恢复出原始网络所要学习的特征分布

### 1.1BN的gamma labada意义

对网络某一层A的输出数据做归一化，然后送入网络下一层B，这样是会影响本层网络A所学习到的特征的。于是BN最后的“尺度变换和偏移”操作。

引入了这个可学习重构参数 $\gamma$ 、 $\beta$ ，让我们的网络可以学习恢复出原始网络所要学习的特征分布。

Date: \ / \ Page: \

经过此操作，网络每一层的输出小值被“拉大”，大值被“缩小”，所以就有避免了梯度消失和梯度爆炸。

即是一个可学习的，有参数 $(\gamma, \beta)$ 的网络层。

### 3.BN训练和测试有什么不同

训练时，均值和方差针对一个Batch。

测试时，均值和方差针对整个数据集而言。因此，在训练过程中除了正常的前向传播和反向求导之外，我们还要记录每一个Batch的均值和方差，以便训练完成之后按照下式计算整体的均值和方差。

测试模型中，对于均值来说直接计算所有batch  $\mu_B$  值的平均值；然后对于标准偏差采用每个batch  $\sigma_B^2$  的无偏估计（无偏估计是用样本统计量来估计总体参数时的一种无偏推断）。

$$E[x] \leftarrow E_B[\mu_B]$$
$$\text{Var}[x] \leftarrow \frac{m}{m-1} E_B[\sigma_B^2]$$

最后测试阶段，BN的使用公式就是：

$$y = \frac{\gamma}{\sqrt{\text{Var}[x] + \epsilon}} \cdot x + \left( \beta - \frac{\gamma E[x]}{\sqrt{\text{Var}[x] + \epsilon}} \right)$$

## 4.LN与BN

LN: Layer Normalization, LN是“横”着来的, 对一个样本, 经过同一层的所有神经元做归一化。

LN中同层神经元输入拥有相同的均值和方差, 不同的输入样本有不同的均值和方差;

BN: Batch Normalization, BN是“竖”着来的, 经过一个神经网络的所有样本做归一化, 所以与batch size有关系。

BN中则针对不同神经元输入计算均值和方差, 同一个batch中的输入拥有相同的均值和方差。

$Z = WX$

LN是对W的一个限制, 需要它进行归一化处理

BN是对X的一个限制, 让其进行归一化处理

二者提出的目的都是为了加快模型收敛, 减少训练时间。

