批标准化如何帮助神经网络：实验

刘子源[1)]

[1)](清华大学 电子工程系, 北京 100084)

摘 要 批标准化是一种自适应的重参数化方法，通过控制神经网络内部激励的均值及方差，以解决训练深层模型的困难。批标准化一经提出便取得了巨大的成功，现在已经是各种神经网络的标配。本文 MNIST 上对两个简单神经网络进行了消融实验，实验结果验证了 BN 的有效性。在表示能力不足的神经网络上加入 BN 层确实能够帮助网络提升收敛速度，并且达到更高的准确率。

关键词 神经网络；优化；批标准化；深度学习

How Does Batch Normalization Help Neural Network: Experiments

Ziyuan Liu[1)]

[1)](Department of Electrical Engineering, Tsinghua University, Beijing 100084, China)

Abstract Batch standardization is an adaptive reparameterization method, which solves the difficulty of training deep models by controlling the mean and variance of the internal activation of the neural network. Batch standardization has achieved great success as soon as it was proposed, and it is now the default configuration of various neural networks. In this paper, an ablation experiment was performed on two simple neural networks on MNIST, and the experimental results verified the effectiveness of BN. Adding a BN layer to a neural network with insufficient representation ability can indeed help the network to increase the convergence speed and achieve a higher accuracy rate.

Key words Neural Network; Optimization; Batch Normalization; Deep Learning

1 背景

1.1 训练神经网络为 NPC 问题

最简单的神经网络是由单个节点组成的感知机（perceptron），其优化问题是凸的。意味着所有的局部极值点都是全局极值点，并且有大量且有效的优化算法来求解凸优化问题。但是当神经网络中节点变多或者层数变深时，其优化问题已经被证明是 NP-hard。

考虑下面的问题：“给定一个神经网络和一些训练样本，是否存在一个网络的权重集合，使得神经网络可以对所有的样本输出正确的结果?”。Judd 在 1988 年[1] 证明了这个问题是 NP-complete 的，同时 Judd 也证明了即使只要求神经网络正确输出 2/3 的训练样本，这个问题同样是 NP-complete 的。这意味着，在最坏情形下，近似地训练一个神经网络在本质上都是十分困难的。而后在 1993 年，Blum 和 Rivest 也证明了训练[1]一个两层三节点的神经网络是 NP-complete 的。

1.2 批标准化

批标准化（下文称为 BatchNrom）并不是一个优化算法，而是一个自适应的重参数化的方法，试图解决训练非常深模型的困难[2]。在神经网络的层面，BatchNorm 通过在神经网络的两个层（layer）之间添加一个额外的标准化操作来控制数据的均值和方差，以稳定输入的分布，从而减少网络内部的“Covariate Shift”（通常指样本分布在不同数据集中不一致的现象，后文简称 ICS）。

BatchNorm 计算过程可以写为

$$y = g(\hat{h}), \hat{h} = \gamma \frac{h - \mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}} + \beta \; and \; h = \boldsymbol{w}^T \boldsymbol{x}$$

其中，y 是一个神经元的输出，$g(\cdot)$ 表示某个激活函数，h 和 $\hat{h}$ 分别是 BatchNrom 前和后的隐藏值（即激活函数的输入值），$\boldsymbol{x}$ 和 $\boldsymbol{w}$ 分别是网络某一层的

刘子源，性别男，2000 年生，博士学位在读，学生，主要研究领域为深度学习 E-mail: liuziyua22@mails.tsinghua.edu.cn.
第 1 作者手机号码：13001911233，E-mail: liuziyua22@mails.tsinghua.edu.cn

[1]确切来说，需将此问题转换为其判定问题，即：“对于两层三个节点的网络，是否存在权重和阈值使其可以对给定的训练样本输出正确的结果”

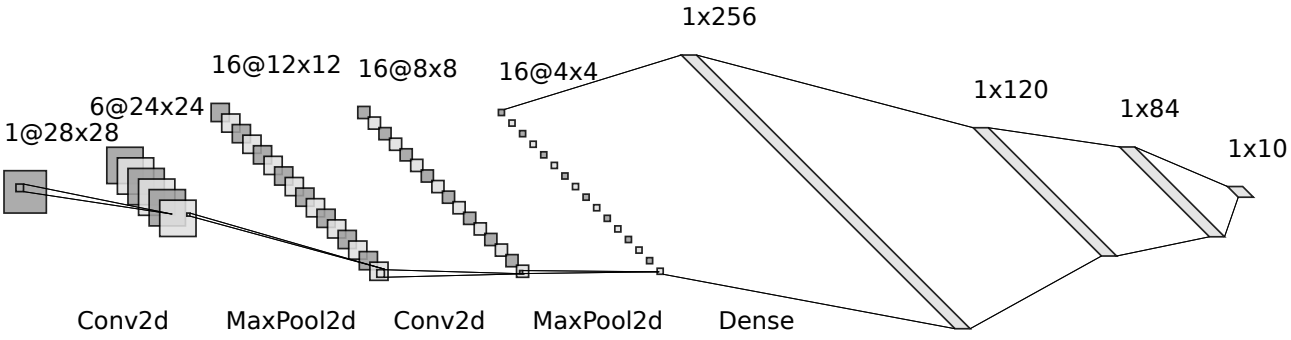

图 1 LeNet 结构图

输入和网络参数。在 BatchNrom 中，$\mu_{\mathcal{B}}$ 和 $\sigma_{\mathcal{B}}$ 分别是 h 的均值和标准差，他们通过一个小批量样本来针对每个神经元独立估计。γ 和 β 分别是缩放和平移的参数。其算法如下面的算法流程图所示。

算法 1 BatchNorm

输入： $h_1, \cdots, h_m$ *based on a miniBatch* $\mathcal{B} = \{x_{1\cdots m}\}, \gamma, \beta$
输出： $\{y_i = BN_{\gamma,\beta}(h_i)\}$
$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} h_i$ // 小批量均值
$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(h_i - \mu_{\mathcal{B}})^2$ // 小批量方差
$\hat{h}_i \leftarrow \frac{h_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}}$ // 标准化
$y_i \leftarrow \gamma\hat{h}_i + \beta \equiv BN_{\gamma,\beta}(h_i)$ // 缩放和平移

在 BatchNorm 方法中，最后的缩放和平移操作是为了确保 BatchNorm 变换具有表示一个恒等变换的能力，从而恢复原网络的表示能力。当 $\gamma^{(k)} = \sqrt{Var[h^{(k)}]}$ 和 $\beta^{(k)} = E[h^{(k)}]$ 时，我们可以恢复出原始的激活值（activation），其中 k 代表隐藏层的第 k 个维度。

2 实验

为了验证批标准化（Batch Normalization[3]）是否能够改进神经网络的训练过程，本文用一些经典的图像分类问题（MNIST[4]）来进行实验。

2.1 求解的优化问题

图片分类问题(image classification)本质上可以归为机器学习中的有监督学习(supervised learning)。在有监督学习中，优化问题是需要找到一个最优的函数 $f(x)$（更确切地说，找到其中最优的参数选择），使得训练样本的损失函数最小，

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} L(y^i, f(x^i, \theta))$$

其中，N 是训练样本的个数，θ 是函数中的参数，x^i 为第 i 个训练样本的特征向量（在图片分类问题中为图片的像素向量，大小为 $H \times W$，H 为图片的高度，W 为图片的宽度），y^i 是对应的样本标签，L 是损失函数。

2.2 算法框架

在 MNIST 数据集上，本文分别采用了多层感知机（MLP）和卷积神经网络（CNN）作为基线算法，并在网络中加入批标准化层（BatchNorm Layer）来对比验证批标准化是否有效。

对于多层感知机，本文选取了和 IOFFEIoffe et al.[3] 文中同样的网络结构。即：输入为 28×28 的图片，之后有三个全连接的隐藏层。其中每层的神经元个数均为 100，并且用 Sigmoid 函数增加非线性，即 $y = g(Wu + b), g(x) = \frac{1}{1+exp(-x)}$。最后一个隐藏层再连接一个 10 个神经元的全连接层（MNIST 有 10 个类别）。

对于卷积神经网络，本文选择了较为简单的 LeNet[4]（图1）网络结构作为基础。即：输入为 28×28 的图片，之后有两个卷积层和两个池化层（Max-Pooling layer），再之后是三个全连接层（网络中各层参数如图1所示）。原本的 LeNet 中采用 Sigmoid 非线性映射。在 LeNet 基础上，本文还采用了 ReLU 非线性的 LeNet 网络（记为 LeNetR，即 LeNet+ReLU），LeNetR 除池化层外，每一层均采用 ReLU[5] 非线性映射。

添加 BN 后的网络。回顾一下，对于全连接层 $z = g(Wu + b)$，其中 W, b 是模型可学习的参数，$g(\cdot)$ 为非线性激活函数（如 sigmoid 或 ReLU）。BN 层添加在非线性函数之前，由于对 $Wu + b$ 进行了标准化，其中的偏置 b 可以忽略，即添加了 BN 后的全连接层表示为 $z = g(\boldsymbol{BN}(Wu))$。对于卷积神经网络，其卷积层的参数可以表示成 $N \times C \times H \times W$ 大小的特征图（feature maps），其中 N 为批量样本大小（batch size），C 表示通道数（特征图个数），

(H, W) 为特征图的尺寸。此时，BN 层对于每个特征图只计算一个均值和方差，并在整个特征图上共享此参数。

2.3 实验分析

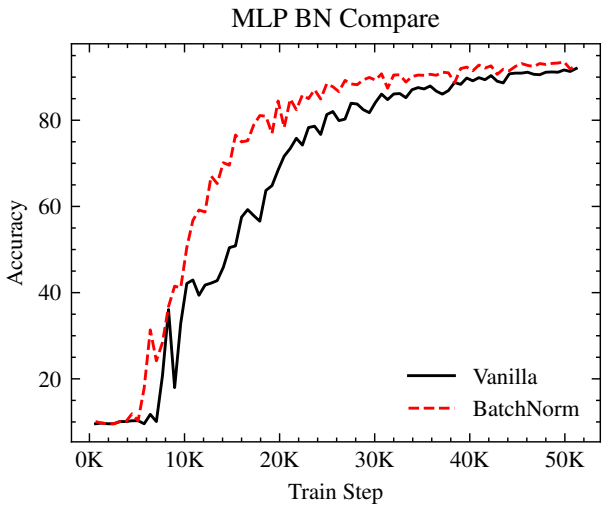

图 2　多层感知机对比结果。图中横轴为训练的迭代次数，纵轴为神经网络模型在测试集上的准确度。黑色实线表示未添加 BN 层的标准多层感知机器的训练曲线（同 2.2 节所描述），红色虚线为在每个全连接层中都添加 BN 层后网的训练曲线。

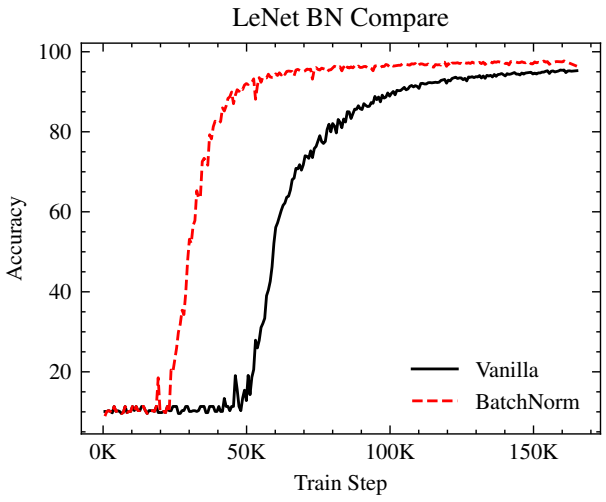

图 3　卷积神经网络（LeNet）对比结果。图中横轴为训练的迭代次数，纵轴为神经网络模型在测试集上的准确度。黑色实线表示未添加 BN 层的标准 LeNet 网络的准确度曲线（同 2.2 节所描述），红色虚线为在每个卷积层和全连接层中添加 BN 层后网络的训练曲线。

实现细节。本文选择 pytorch[6] 作为实验代码框架。在具体训练中，采用交叉熵损失（Cross entropy loss），并选择 Adadelta 优化器[7] 对多层感知机和卷积神经网络进行优化，学习率（learning rate）设置为 1。另外，还采取等间隔下降学习率的学习率调整方法（gamma 设置为 0.7，即 $lr^{(i+1)} = 0.7 \times lr^{(i)}$）。

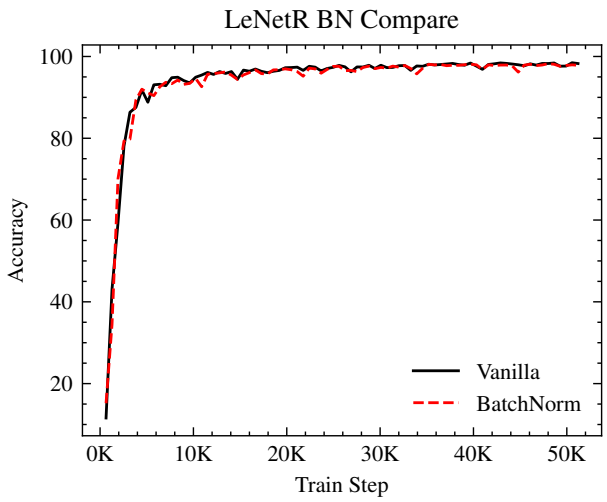

图 4　卷积神经网络（LeNetR）对比结果。图中横轴为训练的迭代次数，纵轴为神经网络模型在测试集上的准确度。黑色实线表示未添加 BN 层的标准 LeNetR 网络的准确度曲线（同 2.2 节所描述），红色虚线为在每个卷积层和全连接层中添加 BN 层后网络的训练曲线。

实验结果。从图2和图3的结果中可以看出，添加了 BN 层的神经网络能够以更快的速度收敛，提前达到稳态。并且在相同训练次数下，添加 BN 层的网络准确率总是更高。这些都验证了 BN 层能够加速神经网络的收敛速度，并提升神经网络的性能。但是，LeNetR（图4）的实验结果并没有体现出 BN 的有效性。事实上，在 LeNetR 网络中是否添加 BN 层对模型的收敛速度和准确率并没有什么改变。究其原因，可能是 ReLU 对网络的改进“掩盖”了 BN 对网络的帮助。从图4可以看出将 Sigmoid 改为 ReLU 后的 LeNet 网络在 4K 的训练次数后就基本达到了稳态（相比标准 LeNet 为 100K），添加 BN 与否并无较大差别。换句话说，LeNetR 网络的表示能力比标准的 LeNet 要强，在 MNIST 数据集上添加 BN 与否都足以迅速地学习到最优的参数。

3 小结

本文为了验证 BN 对神经网络的帮助，在手写数字分类 MNIST 数据集上对简单的多层感知机和卷积神经网络进行了消融实验。实验结果验证了 BN 的有效性，在表示能力不足的神经网络上加入 BN 层确实能够帮助网络提升收敛速度，并且达到更高的准确率。

参考文献

[1] JUDD J S. Neural network design and the complexity of learning[M]. [S.l.]: MIT press, 1990.

[2] GOODFELLOW I, BENGIO Y, COURVILLE A, et al. Deep learning: volume 1[M]. [S.l.]: MIT press Cambridge, 2016.

[3] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International conference on machine learning. [S.l.]: PMLR, 2015: 448-456.

[4] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[5] NAIR V, HINTON G E. Rectified linear units improve restricted boltzmann machines[C]//Icml. [S.l.: s.n.], 2010.

[6] PASZKE A, GROSS S, MASSA F, et al. Pytorch: An imperative style, high-performance deep learning library[M/OL]//WALLACH H, LAROCHELLE H, BEYGELZIMER A, et al. Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 2019: 8024-8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[7] ZEILER M D. Adadelta: An adaptive learning rate method[Z]. [S.l.: s.n.], 2012.

[8] SHIMODAIRA H. Improving predictive inference under covariate shift by weighting the log-likelihood function[J]. Journal of statistical planning and inference, 2000, 90(2):227-244.

[9] RUSSAKOVSKY O, DENG J, SU H, et al. Imagenet large scale visual recognition challenge[J]. International journal of computer vision, 2015, 115(3):211-252.