尺度因子正则化 BN 算法

刘向阳* 汪 琦
(河海大学 江苏 南京 211100)

摘 要 针对进一步提升深度神经网络训练的收敛速度问题,借鉴批规范化(Batch Normalization,BN)算法的特点,提出尺度因子正则化 BN 算法。通过对 BN 层中的可学习尺度因子 γ 施加 L2 正则化,使得 γ 得到衰减,进而参数的梯度上界降低,优化空间更加平滑。基于 VGG16 Net 与 AlexNet,在 cifar10、cifar100 及裂缝图像数据集上进行该算法与 BN 算法的图像分类对比实验,结果表明该算法不仅提高了网络训练的收敛速度,而且在相同训练次数下提高了准确率。

关键词 批规范化 尺度因子 L2 正则化 图像分类

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.06.036

BN ALGORITHM OF SCALE FACTOR REGULARIZATION

Liu Xiangyang* Wang Qi
(*Hohai University*, *Nanjing 211100*, *Jiangsu*, *China*)

Abstract In order to further improve the convergence speed of deep neural network training, using the characteristics of batch normalization (BN) algorithm as reference, a BN algorithm of scale factor regularization is proposed. By applying L2 regularization to the learnable scale factor γ in the BN layer, γ was attenuated, the gradient upper bound of the parameter was reduced, and the optimization space was smoother. Based on VGG16 Net and AlexNet, the image classification comparison experiments between this algorithm and the BN algorithm were carried out on the cifar10, cifar100 and crack image datasets. The results show that the proposed algorithm not only improves the convergence speed of network training, but also improves the accuracy rate at the same training times.

Keywords Batch normalization Scale factor L2 regularization Image classification

0 引 言

在过去的十几年里,深度学习的预测能力和识别精度在持续提高[1]。目前深度神经网络(DNNs)越来越广泛地应用于各个领域的实际问题中,在计算机视觉、语音识别、机器翻译、游戏、多模型任务等领域取得了引人注目的成功[2]。然而,由于实际问题的复杂性,需要使用层数较深的网络进行训练,我们需要寻找合适的参数初始化方法以及最佳的学习率来帮助网络收敛。一方面,当底层网络中的参数发生微小的改变,随着网络层数的增加,这些微小的变化经过网络层中的线性变换与非线性激活函数的作用而被放大,尤其在训练具有饱和非线性的模型时,容易导致梯度消失或梯度爆炸[3];另一方面,由于上一层的输出作为当前层的输入,参数的变化会导致每一层的输入分布发生改变。因此,深度神经网络的训练过程难以收敛或者收敛速度很慢。

Ioffe 等[3]于 2015 年提出了批规范化(Batch Normalization,BN),使得网络层的输入具有零均值和单位方差的分布,加快网络训练过程的收敛速度[4]。BN 作为一种技术被广泛地使用到深度网络模型中,因为它有效地提高了网络训练过程的收敛速度,对于超参数的选择更加具有鲁棒性,网络学习的过程变得更加稳定。另外,BN 缓解了梯度消失的问题,允许网络使用饱和性的激活函数(如 sigmoid、tanh 等)。BN 的另

收稿日期:2020-12-18。国家自然科学基金项目(61001139);云南省重大科技专项计划项目(202002AE090010)。**刘向阳**,副教授,主研领域:深度学习,数据分析,机器学习。**汪琦**,硕士生。

一个优点是其正则化作用,由于在训练过程中,样本的均值与方差是在小批量上更新的,与总体样本的统计量存在偏差,因此 BN 会引入一定量的噪声,其作用类似于 Dropout[5],防止过拟合(Overfitting)问题的发生,甚至在使用 BN 的前提下,可以默认缺省 Dropout。继 BN 提出之后,2016 年 Salimans 等[6]提出权重规范化(Weight Normalization,WN),对于产生某一通道特征的所有权重进行规范化,Ba 等[7]提出层规范化(Layer Normalization,LN),针对单个样本,对每一层内的神经元进行规范化;2017 年 Huang 等[8]提出实例规范化(Instance Normalization,IN),对每个特征图进行规范化;2018 年 Wu 等[9]提出组规范化(Group Normalization,GN),按通道方向分为几个组,对每个组进行规范化;2019 年 Luo 等[10]提出自适配规范化(Switchable Normalization),将 BN、LN、IN 结合,网络自适应地选择一个合适的规范化操作。另外,已有部分工作开展了对 BN 中参数的正则化研究。文献[9-11]尝试将权重衰减应用到尺度因子 $\boldsymbol{\gamma}$ 与偏置向量 $\boldsymbol{\beta}$ 上;Cecilia 等[12]认为其只有在具有特定连接属性的网络架构(如 ResNets)以及含有过拟合问题的任务中才有效。对于尺度因子 $\boldsymbol{\gamma}$ 与偏置向量 $\boldsymbol{\beta}$ 的影响分析,以上工作仅从实验角度进行了尝试与分析。

MIT 团队于 2018 年提出 BN 的成功与内部协方差偏移无关,而是因为它使损失函数的解空间更加平滑,并提供相关理论证明[13]。BN 限定了损失函数对参数梯度的上界,而该上界与 BN 中可训练的尺度因子 $\boldsymbol{\gamma}$ 直接相关,$\boldsymbol{\gamma}^2$ 平方越小,梯度上界越小,损失函数更具有平滑性,即梯度下降法的训练过程更加平稳高效。本文受该思想的启发,在尺度因子 $\boldsymbol{\gamma}$ 上施加 L2 正则化,通过反向传播,$\boldsymbol{\gamma}$ 被衰减,进而梯度上界减小,优化空间更平滑。基于 VGG16 Net 与 AlexNet,在全连接层分别添加 BN 算法与本文改进的 BN 算法,在数据集上进行图像分类对比实验;实验结果表明,本文的方法相较于 BN 算法,进一步提高了训练的收敛速度,在相同的训练次数前提下,在测试集上表现出更高的分类准确率。

1 标准 BN 算法

1.1 随机梯度下降法

大多数深度学习算法都涉及到优化问题,即最小化或者最大化目标函数。随机梯度下降法(SGD)是训练深度神经网络中经典且有效的优化器算法,随着迭代次数的增加,通过反向传播目标函数对参数的梯度,更新各个参数,使得目标函数达到最小:

$$\theta = \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} l(x_i,\theta) \tag{1}$$

式中:$x_i(i=1,2,\cdots,N)$ 代表整个数据集的输入;θ 是待优化参数的全体集合,目标函数为损失函数。训练目标是使网络对 N 个输入样本的预测值与真实值之差的平均值达到最小。在深度学习中,损失函数是相当复杂的,存在很多平坦区域的鞍点以及局部极小点,优化算法无法到达全局最小点,因此找到的解是近似最优的。

在进行随机梯度下降时,通常使用小批量来代替全部样本进行训练迭代,这具有显著优势,譬如节约计算机内存。此时,对于每次迭代网络的输入变为 x_i $(i=1,2,\cdots,m)$,m 为批量大小。$\frac{1}{m}\sum_{i=1}^{m}\frac{\partial l(x_i,\theta)}{\partial\theta}$ 是损失函数在该批量上对参数的平均梯度,更新参数为:

$$\theta = \theta - \alpha\frac{1}{m}\sum_{i=1}^{m}\frac{\partial l(x_i,\theta)}{\partial\theta} \tag{2}$$

式中:α 代表学习率。虽然随机梯度下降法在小批量上训练简单有效,但是对调参要求高,否则容易陷入非线性饱和区域,导致梯度消失,反向传播的梯度为零,训练无法进行,难以收敛;或底层网络的微小变化随着网络层数增加而被放大,导致梯度爆炸。

1.2 BN 算法公式化

假设一个数据集含有 d 个样本,即 $x=\{x^{(1)},x^{(2)},\cdots,x^{(k)},\cdots,x^{(d)}\}$,对于每个样本,需要将其进行规范化:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{Var[x^{(k)}]} \tag{3}$$

式中:$E[x^{(k)}]$ 和 $Var[x^{(k)}]$ 是全部训练样本的均值与方差。对于包含 m 个样本的小批量,用其均值 μ_B 和方差 $\sigma_B^2+\varepsilon$(ε 是为了防止分母为零而设置的微小量,下标 B 是小批量的集合)来估计 $E[x^{(k)}]$ 和 $Var[x^{(k)}]$,同理有:

$$x_i^{(k)} = \frac{x_i^{(k)} - \mu_B}{\sqrt{\sigma_B^2+\varepsilon}} \tag{4}$$

均值 μ_B 和方差 σ_B^2 的计算公式为:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{5}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu_B)^2 \tag{6}$$

为了恢复数据本身的表达能力,BN 引入两个可学习的参数 $\boldsymbol{\gamma}^{(k)}$ 和 $\boldsymbol{\beta}^{(k)}$,对规范化后的数据进行线性

变换:

$$y^{(k)} = \boldsymbol{\gamma}^{(k)}\hat{x}^{(k)} + \boldsymbol{\beta}^{(k)} \tag{7}$$

式中:$\boldsymbol{\gamma}^{(k)}$为尺度因子,对规范化后的数据进行缩放。当 $\boldsymbol{\gamma}^{(k)} = \sigma_B, \boldsymbol{\beta}^{(k)} = \mu_B$时,式(7)实现等价变换,能将数据恢复到未进行规范化操作之前,在一定程度上保证了输入数据的表达能力。式(1)－式(7)均为向量化的计算,遵循广播原则。

2　尺度因子正则化 BN 算法

MIT 研究人员[13]从平滑性这个角度解释了 BN 带来的效果,首先引入 Lipschitz 连续的概念,对于函数 $f(x)$:

$$\forall x, \forall y, |f(x) - f(y)| \leqslant L\|x - y\|_2 \tag{8}$$

则称函数 $f(x)$ 是 Lipschitz 连续的,其中 L 是 Lipschitz 系数,也可以理解为梯度的上界。

2.1　BN 的平滑性定理

MIT 人员证明了 BN 具有平滑性效果,主要有以下定理。

定理 1　权重空间的 Lipschitz 性[13]。加了 BN 的网络及未加 BN 的网络,损失函数分别记作 $\hat{L}$ 与 L,假设在一个具有 m 个样本的批量上进行训练,令 $g_j = \max\limits_{\|X\| \leqslant \lambda} \|\nabla_W L\|^2$, $\hat{g}_j = \max\limits_{\|X\| \leqslant \lambda} \|\nabla_W \hat{L}\|^2$,则有:

$$\hat{g}_j \leqslant \frac{\boldsymbol{\gamma}^2}{\sigma_j^2}(g_j^2 - m\mu_{g_j}^2 - \lambda^2\langle \nabla_{y_j} L, \hat{y}_j\rangle^2) \tag{9}$$

定理 1 表明,加入 BN 的网络的损失函数对权重的梯度更具有 Lipschitz 连续性,使损失函数更加平滑,梯度更具预测性,因此提升了网络的性能,而$\frac{\boldsymbol{\gamma}^2}{\sigma^2}$决定梯度上界,该值越小,代表损失函数越平滑。

随着网络的训练,方差 σ_j^2 会趋向变大[13],因此本文需要达到两个目标:1) 控制网络中 BN 层的尺度因子 $\boldsymbol{\gamma}$ 衰减;2) 验证尺度因子 $\boldsymbol{\gamma}$ 衰减后,梯度上界更小,损失函数更加平滑,网络训练的收敛速度最终得到提升。

2.2　尺度因子正则化方法

为了使 BN 层中尺度因子 $\boldsymbol{\gamma}$ 变小,联想到权重衰减的思想,将 L2 正则化[14-17]应用到权重上,使权重衰减到更小的值,在一定程度上减少模型过拟合的问题。同理,为了使 $\boldsymbol{\gamma}$ 得到一定程度的衰减,将 L2 正则化运用到 $\boldsymbol{\gamma}$ 上,则有:

$$l(x_i, \boldsymbol{\gamma}) = l(x_i, \boldsymbol{\gamma})_0 + \frac{\lambda}{2n}\sum_{\gamma}\boldsymbol{\gamma}^2 \tag{10}$$

式中:$l(x_i, \boldsymbol{\gamma})_0$是初始的损失函数;$\frac{\lambda}{2n}\sum\limits_{\gamma}\boldsymbol{\gamma}^2$ 为 L2 正则化项;n 为训练样本大小;λ 是正则化系数,通过 λ 来调控衰减的程度。

当 $\boldsymbol{\gamma}$ 未使用 L2 正则化时,基于小批量 m 的随机梯度下降过程中,参数 $\boldsymbol{\gamma}$ 更新为:

$$\boldsymbol{\gamma} = \boldsymbol{\gamma} - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\partial l(x_i, \boldsymbol{\gamma})_0}{\partial \boldsymbol{\gamma}} \tag{11}$$

当 $\boldsymbol{\gamma}$ 使用 L2 正则化时,由于:

$$\frac{\partial l(x_i, \boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} = \frac{\partial l(x_i, \boldsymbol{\gamma})_0}{\partial \boldsymbol{\gamma}} + \frac{\lambda}{n}\boldsymbol{\gamma} \tag{12}$$

$$\frac{1}{m}\sum_{i=1}^{m}\frac{\partial l(x_i, \boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial l(x_i, \boldsymbol{\gamma})_0}{\partial \boldsymbol{\gamma}} + \frac{\lambda}{n}\boldsymbol{\gamma} \tag{13}$$

此时,参数 $\boldsymbol{\gamma}$ 更新为:

$$\boldsymbol{\gamma} = \left(1 - \frac{\alpha\lambda}{n}\right)\boldsymbol{\gamma} - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\partial l(x_i, \boldsymbol{\gamma})_0}{\partial \boldsymbol{\gamma}} \tag{14}$$

式中:α 为学习率,对比式(11)和式(14),$1 - \frac{\alpha\lambda}{n} < 1$,可以发现,对 $\boldsymbol{\gamma}$ 加入 L2 正则化后,使每次更新的值达到更小。

2.3　算法流程

尺度因子正则化 BN 算法的反向传播过程,需要更新的参数为$\{\boldsymbol{\gamma}, \boldsymbol{\beta}, \mu, \sigma\}$。算法 1 为整个算法的流程。其中尺度因子 $\boldsymbol{\gamma}$ 在训练阶段按照式(14)进行更新,得到的 $\boldsymbol{\gamma}$ 值直接运用到测试阶段。

算法 1　尺度因子正则化 BN 算法

训练在小批量 $B = \{x_1, x_2, \cdots, x_m\}$ 上进行,测试在 $B_{\text{inf}} = \{x_1^{\text{inf}}, x_2^{\text{inf}}, \cdots, x_m^{\text{inf}}\}$ 上进行。理论上训练时$\{\mu, \sigma\}$的无偏估计用于测试过程,实际操作中采用移动平均法更新均值与方差来替代无偏估计,假设移动平均的动量为 θ,学习率为 α。

1. 前向传播:

$\mu_B = \frac{1}{m}\sum\limits_{i=1}^{m}x_i$　　//计算均值

$\sigma_B^2 = \frac{1}{m}\sum\limits_{i=1}^{m}(x_i - \mu_B)^2$　　//计算方差

$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$　　//规范化

$y_i = \boldsymbol{\gamma}\hat{x}_i + \boldsymbol{\beta}$　　//重构变换

2. 反向传播:

$\mu = \theta\mu + (1 - \theta)\mu_B$　　//移动平均法更新均值

$\sigma^2 = \theta\sigma^2 + (1 - \theta)\sigma_B^2$　　//移动平均法更新方差

$\boldsymbol{\gamma} = \left(1 - \frac{\alpha\lambda}{n}\right)\boldsymbol{\gamma} - \alpha\frac{\partial L}{\partial \boldsymbol{\gamma}}$　　//更新 $\boldsymbol{\gamma}$

$\boldsymbol{\beta} = \boldsymbol{\beta} - \alpha\frac{\partial L}{\partial \boldsymbol{\beta}}$　　//更新 $\boldsymbol{\beta}$

3. 测试阶段

$$y_i^{\text{inf}} = \frac{\boldsymbol{\gamma}}{\sqrt{\sigma^2 + \varepsilon}}x_i^{\text{inf}} + \left(\boldsymbol{\beta} - \frac{\boldsymbol{\gamma}\mu}{\sqrt{\sigma^2 + \varepsilon}}\right)$$

3 实验及分析

本文选取 VGG16 Net[18]和 AlexNet[19]作为基准网络,对标准数据集 cifar10、cifar100 进行对比实验。VGG16 Net 由 13 层卷积与 3 层全连接层构成。AlexNet 由 5 层卷积层与 3 层全连接层构成。cifar10 与 cifar100 由 50 000 幅训练图像和 10 000 幅测试图像组成,样本图像分辨率为 32×32。其中,cifar10 包含 10 个类别,cifar100 包含 100 个类别。

3.1 实验设置

环境设置:在具有 64 GB 内存的 Windows 10 系统,GeForce RTX 2080 GPU 环境下借助 TensorFlow[20]框架进行实验。

对比实验设置:将 BN 与尺度因子正则化 BN 放在 VGG16 Net 与 AlexNet 的三层全连接层后,其中加入尺度因子正则化 BN 方法称为实验组,加入 BN 方法称为对照组。为了避免其他的技术对实验对比效果的影响,网络中除了加入 BN 与尺度因子正则化 BN,未使用任何其他优化网络的技巧;为消除随机性对实验对比效果的影响,使实验可复现,在实验组与对照组中,均给随机初始化的参数固定了相同的随机种子。

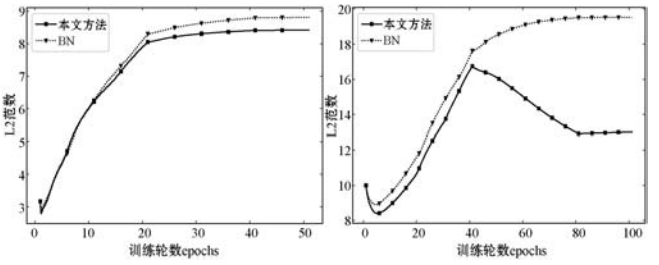
网络设置:实验选取的优化器算法为 momentum,动量值设置为默认的 0.9;隐藏层激活函数选取线性整流函数(ReLU)[21],输出层选用 Softmax 分类器[22]得到分类结果;batch size 选取 128,初始化权重从截断高斯分布里采样。在 VGG16 Net 中,对于 cifar10 数据集,实验总共训练 50 个 epoch,实验组中 γ 的 L2 正则化系数为 0.000 3,实验组与对照组均采用学习率衰减的方式,初始学习率设置为 0.01,在第 20 个 epoch 处衰减为 0.001,第 40 个 epoch 处衰减为 0.000 1。对于 cifar100 数据集,实验总共训练 100 个 epoch,实验组的正则化系数在 1 ~ 40 epoch 设置为 0.000 3,41 ~ 80 epoch 设置为 0.003,81 ~ 100 epoch 设置为 0.000 3,初始学习率设置为 0.01,第 40 个 epoch 处衰减为 0.001,第 80 个 epoch 处衰减为 0.000 1;在 AlexNet 中,学习率均设置为 0.01,对于数据集 cifar10、cifar100,L2 正则化系数分别设置为 0.000 95 与 0.000 25。

3.2 实验中间结果分析

以 VGG16 Net 为例,通过实验验证两点:1) 加入 L2 正则化后尺度因子 γ 是否减小;2) 加入 L2 正则化后梯度上界是否减小。

3.2.1 尺度因子衰减

为了验证尺度因子 γ 是否衰减,对两种方法在最后一层全连接层后的 γ 值随着训练的变化趋势可视化,由于 γ 是向量,故将每个批次上 γ 的 L2 范数作为实验数值进行画图展示,如图 1 所示。



(a) cifar10 (b) cifar100

图 1 不同数据集尺度因子的 L2 范数对比

可以看出,加入 L2 正则化后,γ 数值衰减到更小的值。对于 cifar100 数据集,尺度因子正则化 BN 方法在第 40 个 epoch 处突然下降,其原因是:在第 40 个 epoch 处,学习率由 0.01 减小为 0.001。观察数据的趋势可知,在原 BN 算法中 γ 值有上升的趋势,此时正则化系数从 0.000 3 增加到 0.003,γ 值下降的趋势变大,大于上升的趋势,故数值下降。在第 80 个 epoch 处,正则化系数下降为 0.000 3,学习率下降为 0.000 1,上升趋势与下降趋势基本平衡,因此参数逐渐趋向收敛。

3.2.2 梯度上界减小

为了验证梯度上界是否变小,对于两个数据集,计算三个全连接层权重 W_fc1,W_fc2,W_fc3 在各个训练阶段梯度的 L2 范数,两种方法在不同数据集上权重梯度 L2 范数的均值与标准差对比如表 1 所示。可以看出,加入尺度因子正则化后,梯度 L2 范数的均值与标准差更小,即梯度上界整体更小,梯度波动更小。

表 1 权重梯度 L2 范数的均值与标准差对比

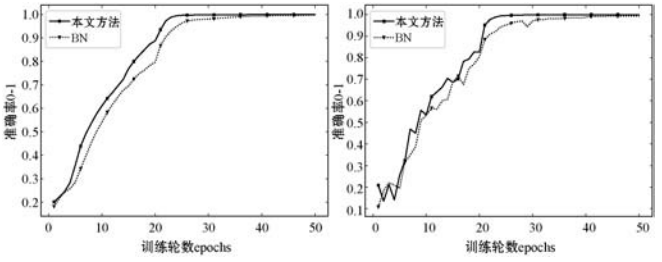
数据集	全连接层	梯度 L2 范数的均值		梯度 L2 范数的标准差	
		BN	本文方法	BN	本文方法
cifar10	W_fc1	0.200 3	**0.193 6**	0.580 8	**0.548 9**
	W_fc2	0.236 5	**0.228 6**	0.238 1	**0.236 6**
	W_fc3	0.198 4	**0.191 7**	0.200 0	**0.196 5**
cifar100	W_fc1	0.531 3	**0.338 4**	1.074 0	**1.009 3**
	W_fc2	0.722 5	**0.478 3**	0.685 3	**0.292 5**
	W_fc3	0.769 4	**0.508 8**	0.667 5	**0.289 1**

3.3 实验结果及分析

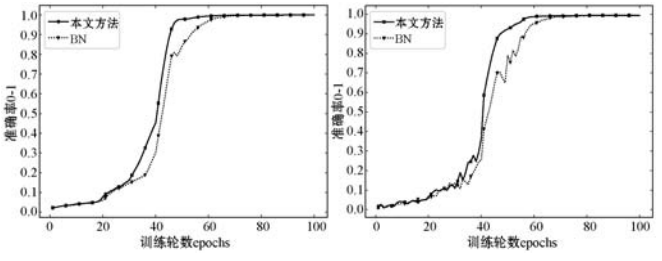
3.3.1 实验结果

绘制训练集与测试集准确率随着训练轮数变化的曲线图,VGG16 Net 实验对比效果如图 2 所示,AlexNet 的实验对比效果如图 3 所示。图中每轮训练对应的训练集与测试集的准确率值分别是对当前训练轮次(epoch)下训练集的 390 个批次与测试集的 78 个批次

的准确率取平均值所得。VGG16 Net 与 AlexNet 在两个数据集上部分训练次数的训练集与测试集准确率见图 4 – 图 5。两种方法在不同网络与数据集上训练收敛的时间对比如表 2 所示。

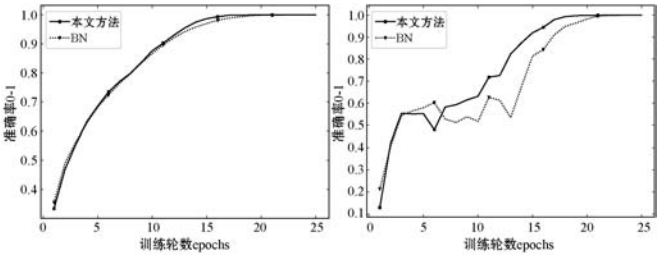


(a) cifar10 训练集　　(b) cifar10 测试集

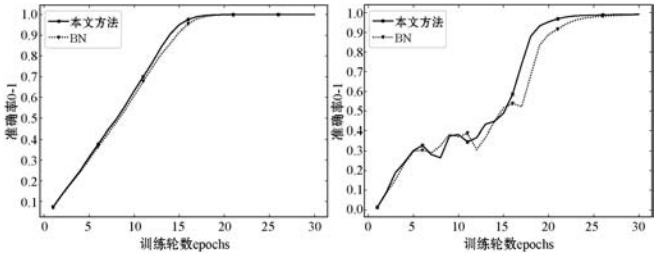


(c) cifar100 训练集　　(d) cifar100 测试集

图 2　VGG16 Net 在 cifar10&cifar100 上的训练与测试准确率对比



(a) cifar10 训练集　　(b) cifar10 测试集



(c) cifar100 训练集　　(d) cifar100 测试集

图 3　AlexNet 在 cifar10&cifar100 上的训练与测试准确率对比

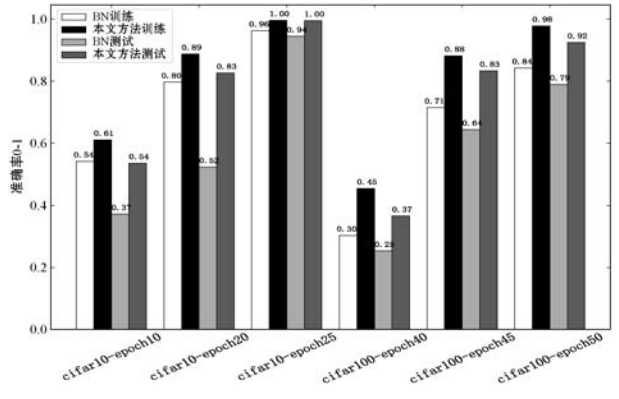


图 4　VGG16 Net 在 cifar10&cifar100 部分训练次数训练集与测试集准确率对比

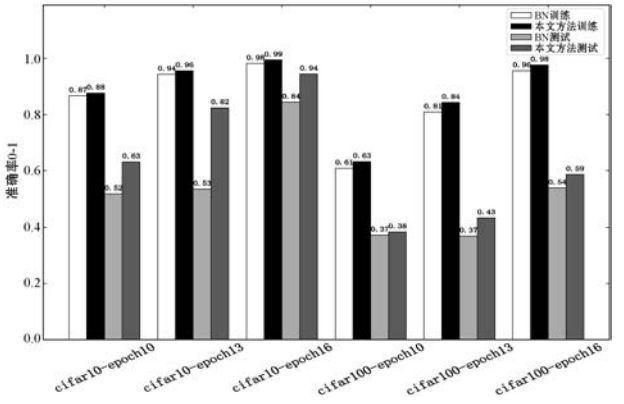


图 5　AlexNet 在 cifar10&cifar100 部分训练次数训练集与测试集准确率对比

表 2　两种方法在不同网络与数据集上达到收敛的时间对比

单位:s

网络	数据集	传统 BN 算法	尺度因子正则化 BN 算法
VGG16 Net	cifar10	719.30	**463.20**
	cifar100	1312.40	**1 138.70**
AlexNet	cifar10	92.70	**82.40**
	cifar100	97.85	**92.70**

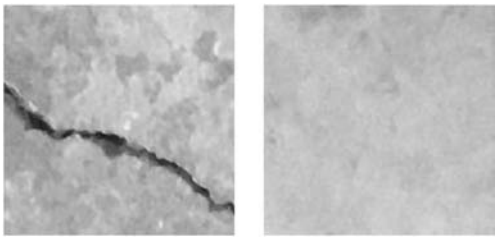
3.3.2　实验分析

图 2 与图 3 表明,无论是 VGG16 Net 还是 AlexNet,本文方法较 BN 算法,在训练时加快了收敛速度;图 4 与图 5 表明,对于同样的训练轮数,本文方法在训练集与测试集上均能达到更高的分类准确率。表 2 表明本文方法使网络收敛更快。该方法相较于传统 BN 算法,性能提升的效果对收敛更慢的深层网络 VGG16 Net 更加显著。

尺度正则化 BN 算法得到改进的主要原因是:γ 衰减后,使得权重的梯度上界更小、更稳定,梯度更具有预测性,即损失函数的 Lipschitz 系数更小,使损失函数的解空间更平滑。在深度学习的背景下,如果函数满足 Lipschitz 连续,Lipschitz 连续函数的变化速度以 Lipschitz 系数为界,加入尺度因子正则化 BN 方法后,该界变得更小。在梯度下降的优化算法下,输入的微小变化随着网络层数的增大将使输出也产生微小的变化,这对于凸优化问题提供了一些保证。因此限制损失函数满足 Lipschitz 连续性,提升了网络的收敛速度。

3.4　裂缝分类

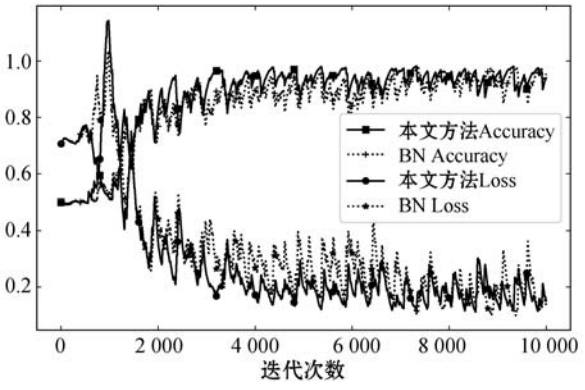
以上实验均从标准的数据集上进行分析,为了进一步验证本文改进算法的有效性,选取了真实的混凝土裂缝数据集[23],其中包含 20 000 幅有裂缝的正样本图像和 20 000 幅无裂缝的负样本图像,数据集中每个图像都是 227 ×227 分辨率的 RGB 图像。图 6 为正负样本示例图。



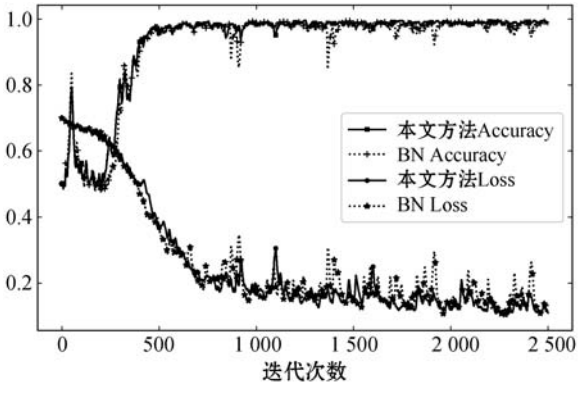
(a) 正样本(裂缝) (b) 负样本(无裂缝)

图 6 混凝土数据集正负样本示例

基于 VGG16 Net 与 AlexNet 网络,训练过程中,每个迭代批次的测试集准确率与损失值的变化趋势对比如图 7 所示。其中 batch size 设置为 64,学习率为 0.000 1,VGG16 Net 的 L2 正则化系数为 0.000 2,AlexNet 的 L2 正则化系数为 0.000 3。



(a) VGG16 Net



(b) AlexNet

图 7 本文方法与 BN 算法在测试集上的效果对比

可以发现,对于裂缝的二分类问题,当 BN 算法在测试集上仍波动明显时,本文方法已趋向稳定,进一步验证了尺度因子正则化 BN 算法相较于标准的 BN 算法,加快了收敛的速度。

4 结 语

本文提出尺度因子正则化 BN 算法,基于 VGG16 Net 与 AlexNet,对两个网络分别添加 BN 算法和尺度因子正则化 BN 算法,并在标准数据集 cifar10、cifar100 及真实裂缝图像数据集进行图像分类对比实验。实验结果表明,本文提出的尺度因子正则化的 BN 算法相较于 BN 算法,提高了训练的收敛速度,并且在相同的训练次数下,该方法提升了图像分类问题的分类准确率。目前,本文算法在图像分类任务上具有显著效果,未来尝试将该算法运用于目标检测任务。

参 考 文 献

[1] 张军阳,王慧丽,郭阳,等. 深度学习相关研究综述[J]. 计算机应用研究,2018,35(7):1921 - 1928.

[2] LeCun Y, Bengio Y, Hinton G E. Deep learning[J]. Nature,2015,521(7553):436 - 444.

[3] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//32nd International Conference on Machine Learning,2015:448 - 456.

[4] 刘建伟,赵会丹,罗雄麟,等. 深度学习批归一化及其相关算法研究进展[J]. 自动化学报,2020,46(6):1090 - 1120.

[5] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research,2014,15(1):1929 - 1958.

[6] Salimans T, Kingma D P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks[C]//30th International Conference on Neural Information Processing Systems,2016:901 - 909.

[7] Ba J L, Kiros J R, Hinton G E. Layer normalization[EB]. arXiv:1607.06450,2016.

[8] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization[C]//IEEE International Conference on Computer Vision,2017:1510 - 1519.

[9] Wu Y X, He K M. Group normalization[J]. International Journal of Computer Vision,2019,128(3):742 - 755.

[10] Luo P, Ren J M, Peng Z L, et al. Differentiable learning-to-normalize via switchable normalization[C]//International Conference on Learning Representations,2019.

[11] He T, Zhang Z, Zhang H, et al. Bag of tricks for image classification with convolutional neural networks[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2019:558 - 567.

[12] Cecilia S, Michael J D. Four things everyone should know to improve batch normalization[EB]. arXiv:1906.03548,2020.

[13] Santurkar S, Tsipras D, Ilyas A, et al. How does batch normalization help optimization? [C]//32nd International Conference on Neural Information Processing Systems, 2018:2483 - 2493.

[14] Zhang L, Yang M, Feng X C. Sparse representation or collaborative representation: Which helps face recognition? [C]//IEEE International Conference on Computer Vision,

2011:471 – 478.
[15] Xu Y, Zhong Z F, Yang J, et al. A new discriminative sparse representation method for robust face recognition via l_2 regularization[J]. IEEE Transaction on Neural Networks and Learning Systems,2017,28(10):2233 – 2242.
[16] Wang J, Yang J C, Yu K, et al. Locality-constrained linear coding for image classification[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010:3360 – 3367.
[17] 刘建伟,崔立鹏,刘泽宇,等. 正则化稀疏模型[J]. 计算机学报,2015,38(7):1307 – 1325.
[18] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[C]//International Conference on Learning Representations,2015.
[19] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM,2017,60(6):84 – 90.
[20] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems [EB]. arXiv:1603.04467,2016.
[21] Toch L. Phone recognition with deep sparse rectifier neural networks[C]//IEEE International Conference on Acoustics, Speech, and Signal,2013:6985 – 6989.
[22] 屈永康,冀俊忠,梁佩鹏,等. 基于正则化 Softmax 回归的全脑功能性磁共振成像数据特征选择框架[J]. 模式识别与人工智能,2016,29(7):641 – 649.
[23] Zgenel F, Sorgu A G. Performance comparison of pretrained convolutional neural networks on crack detection in buildings [C]//35th International Symposium on Automation and Robotics in Construction,2018:693 – 700.

(上接第 242 页)

的整体性能下降了 0.3 百分点。删除了字符表示,BNER 的整体性能下降了 0.99 百分点。由此,表明了预训练语言模型与特征抽取在工业设备故障实体识别中的重要性。此外,实验结果显示了学习单词的字符表示对实体识别的益处,这也表明字符级别的表示能够改善对领域特定词汇的理解和表达。

6 结　语

随着工业领域数字化和智能化的发展,快速准确地识别设备故障信息变得至关重要。本文提出一种边界感知实体识别模型,该模型通过先准确定位实体跨距,后对跨距进行类别分类的方式识别文本中实体。此外,本文构建面向工业设备故障的实体识别数据集,填补了该领域高质量标注数据的空缺。经过实验验证,该模型在工业设备故障实体识别任务中表现出色,对未来的数据分析和知识图谱构建具有重要学术和实践价值。

参 考 文 献

[1] 张阳,刘瑾. 基于字符增强的工业设备故障命名实体识别[J/OL]. 电子科技. [2024 – 01 – 16]. https://doi.org/10.16180/j.cnki.issn1007-7820.2024.10.007.
[2] 黄子麒,胡建鹏. 基于语义感知的工业制造领域知识抽取方法[J/OL]. 计算机工程与应用. [2024 – 01 – 16]. http://kns.cnki.net/kcms/detail/11.2127.TP.20230601.1624.004.html.
[3] Shen D, Zhang J, Zhou G, et al. Effective adaptation of a hidden Markov model-based named entity recognizer for biomedical domain[C]//ACL 2003 Workshop on Natural Language Processing in Biomedicine,2003.
[4] Ju M, Miwa M, Ananiadou S. A neural layered model for nested named entity recognition[C]//2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,2018.
[5] Sohrab M G, Miwa M. Deep exhaustive model for nested named entity recognition[C]//2018 Conference on Empirical Methods in Natural Language Processing,2018.
[6] Etzioni O, Cafarella M, Downey D, et al. Unsupervised named-entity extraction from the Web: An experimental study[J]. Artificial Intelligence,2005,165(1):91 – 134.
[7] 王红,李浩飞,邸帅. 民航突发事件实体识别方法研究[J]. 计算机应用与软件,2020,37(3):166 – 172.
[8] Zhao J, Liu C, Liang J, et al. A novel cascade instruction tuning method for biomedical NER[C]//IEEE International Conference on Acoustics,Speech and Signal Processing,2024.
[9] Li J, Sun A, Han J, et al. A survey on deep learning for named entity recognition[J]. IEEE Transactions on Knowledge and Data Engineering,2022,34(1):50 – 70.
[10] 王春雨. 基于 CRF 的农业命名实体识别研究[D]. 保定:河北农业大学,2014.
[11] Qin Y, Shen G W, Zhao W B, et al. A network security entity recognition method based on feature template and CNN-BiLSTM-CRF[J]. Frontiers of Information Technology & Electronic Engineering,2019,20:872 – 884.
[12] 王腾科,朱广丽,李瀚臣,等. 基于字词融合和多头注意力的专利实体识别[J]. 计算机工程与设计,2023,44(12):3778 – 3783.
[13] Zhao J, Li Z, Xiao Y, et al. HTMapper: Bidirectional Head-Tail mapping for nested named entity recognition[C]//32nd ACM International Conference on Information and Knowledge Management,2023.
[14] 党小超,刘涧,董晓辉,等. 面向不平衡数据的机械设备故障命名实体识别[J/OL]. 计算机工程. [2024 – 01 – 16]. https://doi.org/10.19678/j.issn.1000-3428.0068078.