Batch-attention:深度学习中一种新的协调过拟合与欠拟合的方法

胡涵清,李政勋,吴竹南

北京信息科技大学经济管理学院,北京 100192

摘要 在深度学习网络训练的过程中,现有大多数提升模型效果的方法都集中在网络上,要提升模型的效果与准确率,就须关注数据的特性。提出了一种新的深度学习模型训练框架Batch-attention,从数据层面出发,改变了原有训练方式,经实验证明可以协调深度学习模型的过拟合与欠拟合。通过在Cifar10与Cifar100数据集上分别采用Resnet34、Transformer和efficientnet-b7进行实验对比,证明了采用Batch-attention的模型相对于基准模型,在测试集上的准确率与F1-score均有一定提升。在测试实验中,进一步分析了Batch-attention的作用机制。

关键词 深度学习;过拟合;注意力机制;有监督学习;机器学习

随着深度学习热潮的到来,人们对神经网络模型的关注度越来越高,神经网络模型在不断的迭代过程中变得越来越复杂,参数也变得越来越多,但因此很容易导致模型过拟合或欠拟合[1],即在损失下降的同时,准确率却不再提升,由此衍生出了一大批解决过拟合或欠拟合的方法,如Dropout[2]、L1和L2正则化[3]等。但这些方法几乎都是从模型优化的角度出发,鲜有方法可以注意到数据的特性[4]。本研究从数据而不是模型优化角度出发,提出了一种与众不同的模型训练方法——Batch-attention(批注意力机制),通过协调某一部分拟合得不好数据的训练次数来提升模型总体拟合效果,从而提升模型的测试效果。同时,在测试实验中Batch-attention还表现出了较好的可解释性与合理性。

1 相关研究

从根源上讲,深度学习与统计学密不可分,也可以说深度学习是数据的科学[5]。针对深度学习问题,近年来,研究人员对模型越发重视,不断去探索

收稿日期:2022-07-18;修回日期:2023-02-07

基金项目:北京信息科技大学"促进同校分类发展——经管学院专业学位点与研究生教育改革"项目

作者简介:胡涵清,副教授,研究方向为大数据分析与挖掘、机器学习算法等,电子信箱:hanqinghu@bistu.edu.cn

引用格式:胡涵清, 李政勋, 吴竹南. Batch-attention: 深度学习中一种新的协调过拟合与欠拟合的方法[J]. 科技导报, 2023, 41(13): 100-108;
doi: 10.3981/j.issn.1000-7857.2023.13.010

新的模型机制如Resnet[6]、VIT[7]等。但与此同时,带来的另一个问题就是深度学习领域的研究者可能会越来越忽视数据的重要性。而Batch-attention就强调了数据在模型训练过程中的重要性。

目前,在机器学习与深度学习领域,数据的重要性尤为显著。无论是集成学习,还是特征工程无一不是在强调数据的重要性。Bagging[8]作为一种常用的集成学习方法,在训练的时候采用Bootstrap采样策略有放回的抽取训练集,来划分训练跟测试数据,降低模型方差。类似的还有K折交叉验证方法[9],通过把整个数据集分为K份,然后取其中1份做测试集,其余K-1份数据作为训练集。二者其实都是在采取不同的数据训练策略。在经典集成学习算法Adaboost中就采用了对训练数据给予权重,然后通过训练得到多个弱分类器组成强分类器的训练方法[10]。Batch-attention与Adaboost具有异曲同工之处,二者都是针对数据进行差异化训练,区别是对于数据的具体处理方式方法不同。

另外,在深度学习领域,衍生出了一些增强数据多样性的方法。数据增广是从数据预处理的角度改善模型效果的一种有效方法[4]。在模型开始训练之前,对图片数据剪裁、缩放等,增大数据集的多样性与随机性,从而提高模型的泛化性能。还有的研究者通过对抗神经网络(GAN)对原始数据进行生成,从而增强数据集的数量与多样性[11]。

Batch-attention在原理解释上类似注意力机制[12],注意力机制的作用机理是对一组数据中的那些更重要的特征给予更多的关注,而Batch-attention则是让模型更关注一个批次中那些表现不好的数据,对表现不好的批次数据分配更多的训练权重。因为对这些数据的识别会对模型的准确率提升起到至关重要的作用。

总体来说,Batch-attention思想是建立在原有深度学习框架的小批次训练和损失函数优化基础上的,近年也有很多在批次训练与损失函数上对模型训练框架进行优化改进的研究。例如在数据的处理上,Batch Normalization[13]、Layer Normalization[14]等通过对数据进行归一化来降低过拟合,改善模型。在Batch大小的选择上,Byrd等[15]针对大规模机器学习问题,提出了使用不同样本大小的批量优化方法。在损失收敛速度方面,Li等[16]对小批量数据随机梯度下降(SGD)的收敛速度进行了改进。在分布式算法方面,Dekel等[17]提出将串行梯度的在线预测算法转换为分布式算法的方法。另外,还有很多围绕Batch和损失进行优化的研究,这些研究涵盖了很多方面,然而却很少有针对数据的独特性进行数据差异化训练的研究,而Batch-attention弥补了这方面的空白。

2　模型描述

人类在学习的过程中,会因学习内容与个人学习能力的差异,有不同的学习速度与学习效果,对于那些不易学的资料,就需要注入更多精力。与人类的学习过程类似,深度学习网络在训练过程中也应该对那些不容易学好的数据,给予更多的精力进行学习。

目前通用的深度学习训练方法[18]是首先把数据划分为训练数据和测试数据,然后再把训练数据和测试数据分批次训练,通过损失函数,利用反向传播更新梯度。在这个过程中通常会存在一些过拟合或者欠拟合的问题[19],这样的问题并不是某个模型的特例,而是在整个深度学习领域普遍存在的[20]。如何降低训练误差和缩小训练误差与测试误差的差距是决定深度学习模型效果的重要因素。

Batch-attention从优化模型训练架构的角度出发,对每个Batch的数据进行差异化训练,将训练过程中的损失作为筛选指标,通过给予不同Batch不同的训练次数,有差异地对同一个训练轮次(epoch)里面不同Batch的数据进行训练,而不是在一个epoch里把所有Batch的数据无偏地遍历一遍,从而取得更好的训练效果。

在深度学习模型进行训练以后,由于每一轮训练都是无差别地遍历每一个Batch,所得到的模型结果不一定是均衡的。举一个简单的例子,猫狗分类[21]。如图1所示,这里有3张狗的图片分别是狗A、狗B、狗C,有3张猫的图片分别是猫A、猫B、猫C,如果把这些图片输入进网络里,机器可能会从

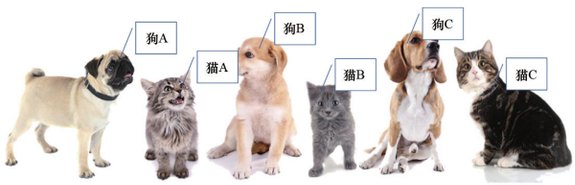


图1 猫狗分类举例

某个角度,有可能是体型、有可能是颜色,对特征进行提取[22],从而对图片进行分类。

假设在训练时,把这6张图片分为3个批次:Batch1(狗A、猫A)、Batch2(狗B、猫B)、Batch3(狗C、猫C)然后送入模型,训练10轮后取得了可能大于80%的精度,然后进行测试,当把狗A、猫A输进去进行测试时,模型可以根据特征进行分类。当把狗B、猫B输进去的时候,模型也可以很好地进行辨认。但当把狗C、猫C输进去的时候,由于初始化参数或是数据特征的原因,在训练过程中,模型在Batch3上并没有达到预期的训练效果,模型可能会觉得猫C的体型跟狗的差不多,所以把猫C辨认成狗,产生了误差。观测者发现模型的分类效果不好,就开始不断重复训练。那么对于当前的模型来说,模型对于Batch1、Batch2的拟合效果已经足够好了,继续训练很可能会导致过拟合,从而降低模型的泛化性。而对于Batch3来说,模型又是不够好的,需要继续拟合(图2)。

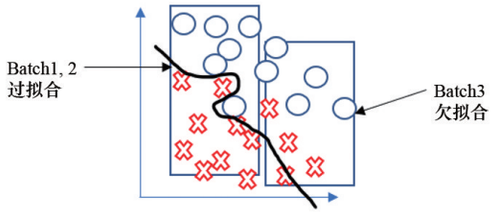


图2 部分数据过拟合与部分欠拟合状态

所以训练出来的模型对于整个数据集来说并不是单纯的过拟合与欠拟合,而是部分数据过拟合、部分数据欠拟合的状态,这时就需要让模型对Batch1、Batch2和Batch3给予不同程度的训练。Batch-attention的作用机制就是在每一轮训练结束时,将训练集的批次平均损失作为下一个epoch中检测不同Batch的拟合效果的指标,如果这个批次的数据拟合效果不好的话,就选择继续对这个Batch的数据进行一些训练,而继续训练的次数,取决于Batch的损失与上一轮训练批次平均损失的差值,相差得越大,需要额外给予的注意力即训练权重也就越大[23]。最终,使模型对各个Batch的数据都可以取得比较好的效果,以在数据拟合上达到图3的效果,加强模型在整体数据集上的泛化能力与拟合优度。

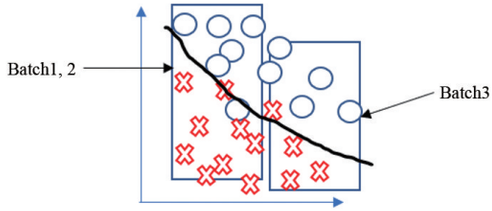


图3 各个Batch之间的数据拟合较好

Batch-attention并没有选择把那些拟合得不好的数据批次单独提取出来重复训练,是因为过多地训练这些拟合得不好的Batch会降低模型对其他类型数据的拟合效果,因此选择在对一个epoch的数据进行总体遍历的基础上,有选择地对那些拟合不好的Batch进行重复训练。

3 模型架构

Batch-attention作用于模型训练的每个批次训练上,并通过对数据给予不同程度的训练,来优化模型效果。具体算法流程如下。

Epoch1:

1)将数据集划分为N个Batch,输入模型进行训练,得到预测值y_i。

2)对预测值y_i与真实值y_i'计算交叉熵损失l_i,对l_i进行累加,记为El(Epoch loss),通过损失更新权重,迭代模型。

$$l_i=-\sum y_i'\log(y_i)$$

$$E_l=\sum l_i$$

Epoch2:

1)取一个Batch的数据,输入模型进行预测。

2)计算Batch的损失l_i,进行一次权重更新。

3)把上一个epoch平均损失的K倍即$K\times(El/$

N)作为阈值,与计算出的损失l_i进行比较。K是超参数,决定了需要给予额外训练数据的比例。

4) 若$l_i>K\times(El/N)$,则说明预测值与真实值偏差较大,需要对这组数据给予更多的训练,重复2~4的步骤,直到$l_i\leqq K\times(El/N)$。反之,若$l_i<K\times(El/N)$,则说明预测值与真实值偏差小于均值,不需要对这组数据给予额外的训练,切换到下一个Batch的数据即可。

Epoch3:

同Epoch2。

……

总之,首先进行第1个epoch的训练,对每个Batch进行遍历,记录每个Batch的损失。然后开始第2轮的训练,还是逐一提取每个Batch的数据,但当再次计算损失并通过梯度下降更新参数时,把$K\times(El/N)$作为是否需要给予额外的attention的阈值,也就是每一个Batch是否拟合良好的指标。通常意义上来讲,随着不断拟合,模型在数据集上的整体损失值会越来越小,拟合的效果也会越来越好。但在观测过程中会发现,总有一些数据的损失会远超上一个epoch的损失均值,从而提高本轮损失均值,这意味着模型对这些数据的拟合效果远没有其他数据好,相对于数据集整体而言,要更加关注这些拟合效果不好的数据,它们才是提升准确率的关键。

4　实验分析

为了验证Batch-attention在不同模型与数据集上的效果,在Cifar10和Cifar100数据集[24]上分别采用Conv(Resnet34)[6]、Transformer(TNT)[25]模型架构与Conv(efficientnet-b7)[16]模型架构进行对比实验。

在CIFAR100上改用efficientnet-b7,而不是沿用之前的模型是因为后者在原参数条件下,在CIFAR100上的准确率较低,不具有参考性,超过一半的数据处于欠拟合状态,所以在使用了Batch-attention后,准确率反而会更低。因此在CIFAR100上改用了efficientnet-b7进行实验。

4.1　训练过程

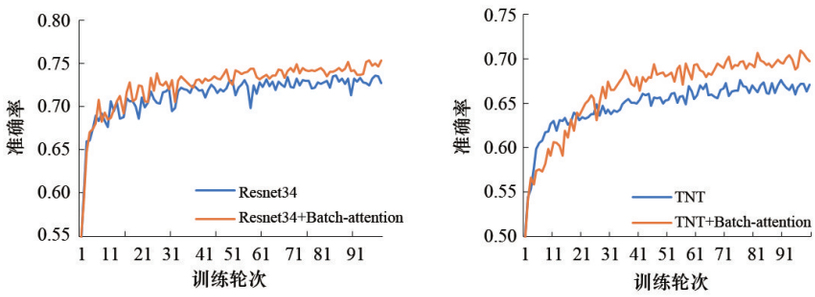
实验在Nvidia 3070 GPU上通过pytorch[26]进行,所有模型都采用交叉熵损失函数[27],GPU采用Nvidia 3070。由于GPU的限制,参数Batchsize的大小设置为32,学习率为0.0001,epoch数为100,优化器为adam[28]。

同时,在进行了多轮反复实验后,发现K值过大会导致Batch-attention效果不显著,产生过拟合,过小会导致模型对数据失去针对性,一般建议选取K值为1.1~1.5,CIFAR10数据集实验在K值为1.2时进行,CIFAR100数据集实验在K值为1.5时进行。

4.2　CIFAR10测试结果分析

TNT与Resnet的原模型测试结果和应用Batch-attention后的模型测试结果在CIFAR10上的表现如图4与表1所示。

通过测试结果,可以看出,无论是Resnet架构还是TNT架构的模型,在用了Batch-attention之后,要比原基准模型的准确率高2%~3%。同时Batch-attention模型的F1-score[29]也同样高出2%~3%,这说明无论是在精确率还是在召回率[30]上,



(a) Resnet模型　(b) TNT模型

图4　2种模型在CIFAR10上的准确率

表1 Res与TNT在Batch-attention下的Test loss和F1-score参数对比

网络名称/参数	最小值(Test loss)	最大值(F1-score)
Resnet34	0.925596	0.737586
Resnet34+Batch-attention	0.856234	0.753712
TNT	1.062034	0.675982
TNT+Batch-attention	1.073162	0.709335

Batch-attention都比原模型框架具有更明显的优势。

另外,通过对比测试集损失的最小值,可以看出,二者的测试集损失很接近。在TNT模型上,Batch-attention的损失甚至要更高一点。损失是衡量预测值与真实值的接近程度[31],实验的结果表现出原基准模型和Batch-attention的测试集损失没有太大变化,但准确率却有显著提升,印证了上文的理论基础。

Batch-attention由于对那些训练集损失较大的数据进行了额外的训练,所以平均每个epoch要比原来的训练时间长20%左右。即便如此,在取同样的训练时间点时,Batch-attention的准确率仍然要比基线模型好,所以Batch-attention还有助于加快模型收敛速度。

如图5所示,从Train loss与准确率的散点连线图也可以看出,在相同的Train loss下,Batch-attention有更高的准确率,这说明Batch-attention也在一定程度上缓和了过拟合。

4.3 CIFAR100测试结果分析

efficientnet-b7原模型(Imagenet预训练)和应用Batch-attention后的模型在CIFAR100上的测试结果表现如图6所示。

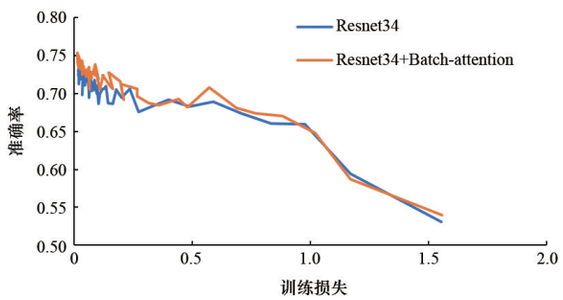


图5 训练损失与准确率散点连线

实验结果表明,总体来讲Batch-attention可以起到一定的加快收敛速度的作用,但是根据任务的类别与模型的差异会有不同效果。

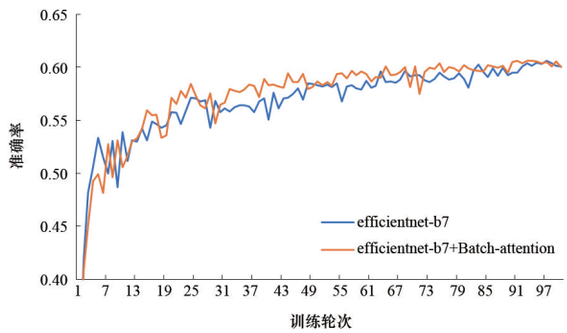


图6 efficientnet-b7与efficientnet-b7+Batch-attention准确率

对比来看,Batch-attention在CIFAR100上的效果不如CIFAR10好,但是还是提高了一定的模型准确率,加快了收敛速度。分析主要原因可能是分类类别变多导致。因为Batch-attention的作用机制是针对那些拟合不好类别的数据进行额外训练,那么当分类的类别变多、模型收敛到最后时,各个类别的识别误差可能没有什么太大的偏差,或者每个Batch数据的差异性不明显,就会导致Batch-attention没有需要特别关注的数据,模型对各类别数据的拟合效果差不多,所以导致Batch-attention的效果变弱。

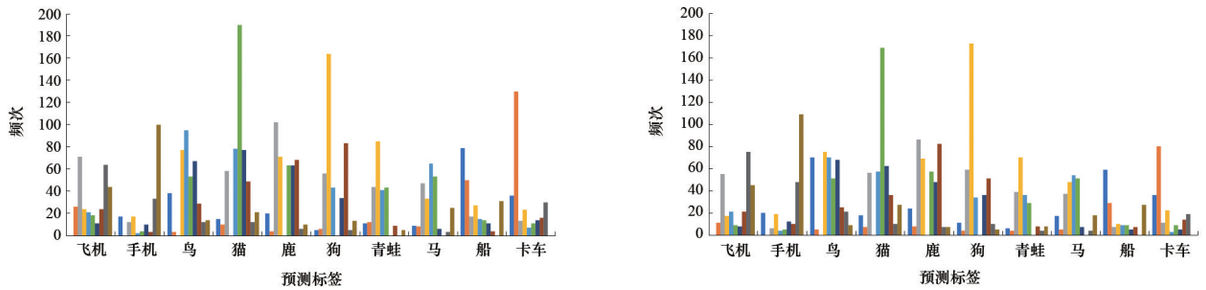
4.4 CIFAR10推理结果分析

为了进一步验证Batch-attention的作用机制,下面对Transformer(TNT)模型与Batch-attention在测试集上的表现进行进一步分析,为更清晰地看出测试结果在不同预测类别上的差异性,这里采用数据类别比较少的CIFAR10数据集进行解释说明。分别选用TNT与TNT+Batch-attention中准确率最高的0.675982和0.709335所对应的权重进行测试,

对模型的误判进行记录。结果如图7与表2所示。

通过三维数据图可以更形象地展示模型误判样本与真实样本之间的关系。如图7所示,实验收集了所有预测标签与真实标签不相等的数据即误判数据。

由误判数据可以看出,在基线模型TNT上,误判次数最多的数据是狗跟猫2类图片,模型把猫误判为狗的频次为190次,模型把狗误判为猫的频次为164次,也就是模型更容易把猫误判为狗。在例中介绍过,由于这2个物种在特征上的相似性,所以对于模型来说,要区分这2类图片相对于其他图片而言,具有更大的难度。而在船这一类图片的判别上误差相对较小,说明此时模型对数据集中各个类别图片的拟合能力是不均衡的。所以根据短板效应,应该优先降低猫类别的误差率。



(a) TNT　(b) TNT+Batch-attention

图7　两种模型的数据误判图

表2　TNT与TNT+Batch-attention测试错误数据频次与方差

类别	预测标签										方差
	飞机	手机	鸟	猫	鹿	狗	青蛙	马	船	卡车	
TNT	230	249	420	521	367	449	293	285	165	263	11144.36
TNT+B	261	153	356	503	288	389	251	254	198	255	9163.96

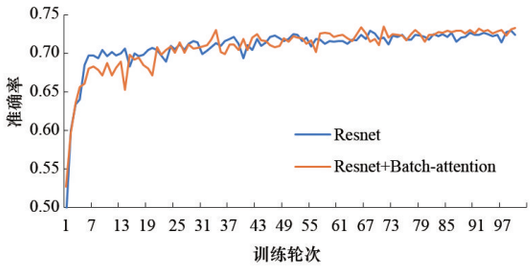
而在添加了Batch-attention后,模型把猫误判为狗的频次为169次,把狗误判为猫的频次为173次。可以看出,Batch-attention大大降低了把猫误判为狗的频次,因为经过Batch-attention,模型对这类图片的学习更多了,这类图片承载了更多的信息。

在表2中,对测试过程进行了整体分析,对模型预测各个类别的误判频次进行了统计,可以看出TNT+Batch-attention的误判频数方差对比原模型较小,这说明Batch-attention起到了平衡预测类别的作用,即原模型可能对类别A的预测效果偏好,对类别B的预测效果不好,在添加了Batch-attention后,可以提升模型对原来预测效果不好的那些数据(类别B)的敏感性,模型对各类别图片判别能力的不均衡性显著减小了,且由于平衡了模型的短板,模型整体准确率也得到了提升。因此,实验的结果有力支撑了之前在模型描述阶段阐述的观点。

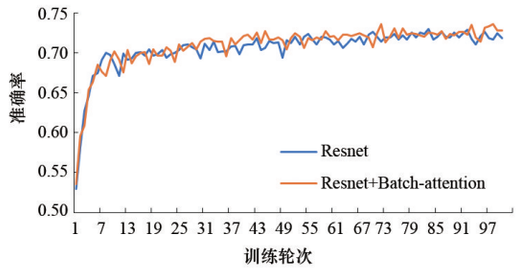
4.5　模型鲁棒性探究

由于Batch-attention更加注重数据的特性,所以当数据存在误差时,模型效果可能出现偏差,为了更好地分析模型的适用范围与鲁棒性,下面对有噪声情况下的模型效果进行实验分析。实验采用Resnet34网络,在CIFAR10数据集上进行训练。共进行了3次对比实验,由于数据批次为32,所以在训练数据集上每个批次随机抽取1、3、6个即3%、10%、20%的数据,对其标签进行随机初始化。测试集准确率如图8所示。

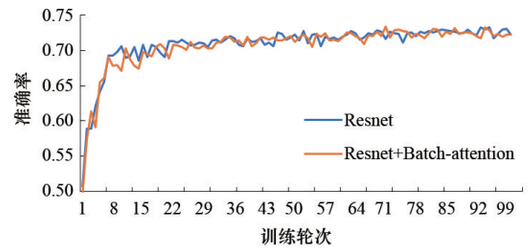
从实验对比结果可以看出,在对数据集分别添加了3%、10%与20%的数据噪声后,Batch-attention的作用效果相对之前有所下降,虽然随着噪声的增加,与基准模型的差距逐渐缩小,但依然平均高于基准模型,因此Batch-attention还具有一定的鲁棒性。



(a) 数据噪声3%



(b) 数据噪声10%



(c) 数据噪声20%

图8 Batch-attention在3%、10%、20%数据噪声下的作用效果

5 结论

Batch-attention的核心思想是强调数据的独特性,针对不同类型的数据应该采取不同的训练方式。在Batch-attention的训练过程中,前半部分看不出模型的太多差别,待基线模型基本收敛后,才能看到改进方式在不断提升,这是因为对于刚开始时未收敛的模型来说,损失下降得比较快,因为模型整体处于欠拟合状态,所以每一轮训练都会使模型的损失大幅下降,几乎没有数据的损失会大于阈值。然而在基线模型基本收敛之后,Batch-attention开始起作用,由于模型基本收敛,所以模型的损失基本不再下降,或者下降得非常缓慢,这时大于阈值的Batch就会逐渐增多,Batch-attention的效果也就逐渐凸显。Batch-attention的另一个缺点是由于对拟合不好的数据批次进行了重复训练,所以会增加一定的训练时间,平均比原模型多15%左右的训练时间,代表约有15%的数据接受了重复训练,因为K值是重复训练数据批次的阈值,所以这个额外的训练时间是跟K值成反比的。在实验过程中也考虑到Batch-attention可能是因为单纯地增加了训练次数,所以模型的效果更好了,但是实验结果验证了Batch-attention的作用远不止于此,即便将Batch-attention的训练时间前推15%,其测试效果在同等训练时间下也要普遍比原模型好很多。

同时在实验过程中发现,Batch-attention存在着一定的适用条件,只适用于原模型准确率较好的情况下,如果基线模型准确率较低的话,那么此时数据集中拟合得不好的数据量占比较大,增加这些数据的训练次数会较大地降低拟合好的数据的效果,从而使模型效果下降。

未来可以对Batch-attention从以下几个方面进行改进。

1)目前的Batch-attention在对训练数据进行判别后,进行的操作是重复训练,但这样可能会造成过拟合,可以尝试在对原始图片进行数据增广后,再传入模型进行训练,可能会降低模型的过拟合,提高泛化能力。

2)Batch-attention对拟合得不好数据的评判标准与处理方法只是其中比较易于实现的一种,相关研究可以Batch-attention的思想为路线,从数据特性的评判标准与数据处理方法入手,对方法进行改进以及创新。

3)Batch-attention训练框架基于误差函数与反向传播进行训练调整,所以也可以适当拓展到其他有监督学习领域。

参考文献(References)

[1] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

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

[3] Nowlan S J, Hinton G E. Simplifying neural networks by soft weight-sharing[J]. Neural Computation, 1992, 4(4): 473-493.

[4] Shorten C, Khoshgoftaar T M. A survey on image data augmentation for deep learning[J]. Journal of Big Data, 2019, 6(60): 1-48.

[5] Tenenbaum J B, Kemp C, Griffiths T L, et al. How to grow a mind: Statistics, structure, and abstraction[J]. Science, 2011, 331(6022): 1279-1285.

[6] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770-778.

[7] Han K, Wang Y H, Chen H T, et al. A Survey on vision transformer[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(1): 87-110.

[8] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123-140.

[9] Wong T T. Performance evaluation of classification algorithms by *k*-fold and leave-one-out cross validation[J]. Pattern Recognition, 2015, 48(9): 2839-2846.

[10] Athitsos V, Alon J, Sclaroff S, et al. BoostMap: An embedding method for efficient nearest neighbor retrieval [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(1): 89-104.

[11] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in Neural Information Processing Systems 27. La Jolla: Neural Information Processing Systems, 2014: 2672-2680.

[12] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems 30. La Jolla: Neural Information Processing Systems, 2017: 6000-6010.

[13] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift [J]. International Conference on Machine Learning, 2015, 37(1): 448-456.

[14] Takagi S, Yoshida Y, Okada M. Impact of layer normalization on single-layer perceptron: Statistical mechanical analysis[J]. Journal of the Physical Society of Japan, 2019, 88(7): 074003.

[15] Byrd R H, Chin G M, Nocedal J, et al. Sample size selection in optimization methods for machine learning[J]. Mathematical Programming, 2012, 134(1): 127-155.

[16] Li M, Zhang T, Chen Y Q, et al. Efficient mini-batch training for stochastic optimization[C]//Proceedings of the 20th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2014: 661-670.

[17] Dekel O, Gilad-Bachrach R, Shamir O, et al. Optimal distributed online prediction using mini-batches[J]. Journal of Machine Learning Research, 2012, 13: 165-202.

[18] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python[J]. Journal of Machine Learning Research, 2011, 12: 2825-2830.

[19] Cai Z, Vasconcelos N. Cascade R-CNN: High quality object detection and instance segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(5): 1483-1498.

[20] Dietterich T G. Ensemble methods in machine learning [M]//Multiple Classifier Systems. Berlin: Springer-Verlag, 2000: 1-15.

[21] Panigrahi S, Nanda A, Swamkar T. Deep learning approach for image classification[C]//2nd International Conference on Data Science and Business Analytics. Piscataway, NJ: IEEE, 2018: 511-516.

[22] Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10000 classes[C]//2014 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2014: 1891-1898.

[23] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: A statistical view of boosting[J]. The Annals of Statistics, 2000, 28: 337-407.

[24] Pang Y, Sun M, Jiang X, et al. Convolution in convolution for network in network[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(5): 1587-1597.

[25] Hong D F, Han Z, Yao J, et al. Spectral former: Rethinking hyperspectral image classification with transformers [J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 5518615.

[26] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library[C]// Advances in Neural Information Processing Systems 32. La Jolla: Neural Information Processing Systems, 2019: 8026-8037.

[27] Zhang Z, Sabuncu M R. Generalized cross entropy loss for training deep neural networks with noisy labels[C]// Advances in Neural Information Processing Systems 31. La Jolla: Neural Information Processing Systems, 2018: 8792-8802.

[28] Liu Z Q, Cao Y W, Wang Y Z, et al. Computer vision-

based concrete crack detection using U-net fully convolutional networks[J]. Automation in Construction, 2019, 104: 129-139.

[29] Guo S, Wang K, Kang H, et al. BTS-DSN: Deeply supervised neural network with short connections for retinal vessel segmentation[J]. International Journal of Medical Informatics, 2019, 126: 105-113.

[30] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(10): 1615-1630.

[31] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318-327.

Batch-attention: A method for reconciling overfitting and underfitting in deep learning

HU Hanqing, LI Zhengxun, WU Zhunan

School of Economics and Management, Beijing Information Science & Technology University, Beijing 100192, China

Abstract In the process of deep learning network training, most existing methods aim to improve the model effect focus on the network. However, to improve the effect and accuracy of the model it is necessary to pay attention to the characteristics of the data. In this paper, batch-attention, a new training framework for deep learning model, is proposed, which changes the original training method from the data level. It is shown that the method can coordinate overfitting and underfitting of the deep learning model. Experimental comparisons using Resnet34, TNT and efficientnet-b7 on Cifar10 and Cifar100 data sets respectively prove that the batch-attention model has improved both accuracy and F1-score in the test set compared with the benchmark model. In addition, the mechanism of batch-attention is further analyzed in the follow-up experiment.

Keywords deep learning; overfitting; attention mechanism; supervised learning; machine learning

（责任编辑 王志敏）