软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(7):2157–2168 [doi: 10.13328/j.cnki.jos.005822]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种简单的共享式多层梯度补给方法*

杜 飞[1], 杨 云[1,2,3], 胡媛媛[1], 曹丽娟[1]

[1](云南大学 国家示范性软件学院,云南 昆明 650504)

[2](昆明市数据科学与智能计算重点实验室,云南 昆明 650504)

[3](云南省高校数据科学与智能计算重点实验室,云南 昆明 650504)

通讯作者: 杨云, E-mail: yangyun@ynu.edu.cn

摘 要: 深度学习通过多层特征提取方式,可以将原始复杂数据自动表征为高级抽象特征,该模型具有很强的建模能力,普遍应用于图像识别语音识别、自然语言处理等高复杂问题中.但深度学习由于网络层数深、参数规模庞大,训练时常常会产生梯度消失、陷入局部最优解、过度拟合等现象.借鉴集成学习的思想,提出一个新颖的深度共享集成网络,该网络通过在深度学习各隐藏层引出多个独立输出层的联合训练的方式,在网络的各层注入梯度,从而对低层隐藏层进行梯度补给,从而降低深度学习中的梯度消失现象,并通过集成多输出层的方式使得整个网络拥有更强的泛化性能.

关键词: 深度学习;集成学习;堆叠泛化;梯度消失;梯度注入

中图法分类号: TP182

中文引用格式: 杜飞,杨云,胡媛媛,曹丽娟.一种简单的共享式多层梯度补给方法.软件学报,2020,31(7):2157–2168. http://www.jos.org.cn/1000-9825/5822.htm

英文引用格式: Du F, Yang Y, Hu YY, Cao LJ. Easy way for multilayer gradient supplies. Ruan Jian Xue Bao/Journal of Software, 2020,31(7):2157–2168 (in Chinese). http://www.jos.org.cn/1000-9825/5822.htm

Easy Way for Multilayer Gradient Supplies

DU Fei[1], YANG Yun[1,2,3], HU Yuan-Yuan[1], CAO Li-Juan[1]

[1](National Pilot School of Software, Yunnan University, Kunming 650504, China)

[2](Kunming Key Laboratory of Data Science and Intelligent Computing, Kunming 650504, China)

[3](Yunnan Provincial University Key Laboratory of Data Science and Intelligent Computing, Kunming 650504, China)

Abstract: Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These have dramatically improved the state-of-the-art methods in speech recognition, visual object recognition, natural language processing, and many other domains. However, due to the large number of layers and large parameter scales, deep learning often results in gradient vanishing, falling into local optimal solution, overfitting, and so on. By using ensemble learning methods, this study proposes a novel deep sharing ensemble network. Through joint training many independent output layers in each hidden layer and injecting gradients, this network can reduce the gradient vanishing phenomenon, and through ensemble multi-output, it can get a better generalization performance.

Key words: deep learning; ensemble learning; stacked generalization; vanishing gradients; gradients injection

深度学习(deep learning)[1,2]是指通过多层特征提取方式将原始复杂数据表征为高级抽象特征的一类方法,

* 基金项目: 国家自然科学基金(61663046, 61876166); 云南省应用基础研究计划(2016FB104); 云南省中青年学术技术带头人后备人才项目(2017HB005); 云南省创新团队项目(2017HC012); 云南省高校重点实验室建设计划

Foundation item: National Natural Science Foundation of China (61663046, 61876166); Yunnan Applied Fundamental Research Project (2016FB104); Yunnan Provincial Young Academic and Technical Leaders Reserve Talents (2017HB005); Yunnan Provincial Innovation Team (2017HC012); Yunnan Provincial University Key Laboratory Construction Plan Fund

收稿时间: 2017-11-07; 修改时间: 2018-03-11, 2018-08-18; 采用时间: 2018-11-16

该类方法构造的模型对数据具有很强的抽象能力,可以自动地从数据中学习到各种复杂的抽象特征.目前,深度学习在语音识别[3]、计算机视觉[4−6]、自然语言处理[7]等问题上都取得了突破性进展.通过堆叠深度学习的层数[8−10],深度学习的泛化性能获得了显著的提升,但在网络层数加深的同时也会产生梯度消失或爆炸(vanishing or exploding gradients)[11−13]问题,这使得较深的深度学习模型常常会陷入比较糟糕的局部最优或鞍点(saddle point)[14].针对梯度消失的问题,最著名的方式是使用贪心逐层预训练[8]的方式将网络权重初始化到一个较优值,然后再使用反向传播(backpropagation)[15]算法进行训练.

该方法引领了深度学习的复苏,但堆叠的网络层数一般到达 6、7 层便很难再继续堆叠,并且逐层预训练所花费的训练周期代价也非常高昂.之后,随着可训练数据量以及计算性能的提升,研究人员使用非饱和性神经元[16−18]以及中间层归一化权重[19]的方式极大地提升了神经网络的层数.但网络层数也并不是越多越好,当层数逐步增加时,便会产生网络退化问题(degradation problem)[4],此时,网络层数的增加反而会降低泛化性能,而调试出一个最佳的网络层数也需要大量的经验技巧以及训练周期.

本文提出一种新颖的深度共享集成网络.该方法允许神经网络在各隐藏层新生长出多个独立的输出层,通过共享特征提取层(隐藏层),集成各输出层的方式对原始数据进行表征学习.在训练阶段,各输出层对输入数据进行独立的预测,而隐藏层的权重将使用叠加梯度的方式进行训练,每一隐藏层权重的梯度将来自于各个独立输出层的反向传播梯度累加.在测试阶段,网络各输出层使用集成的方式进行输出.通过实验,结果表明,深度共享集成网络可以普遍提高神经网络的泛化性能,并且可以大大缓解深层神经网络的梯度消失问题.该方法不仅可以降低研究人员对于网络层数的经验技巧,并且相比于传统集成多个神经网络的方式,还极大地降低了训练成本.

本文第 1 节描述该想法的研究动机.第 2 节描述之前的相关工作.第 3 节正式介绍深度共享集成网络.第 4 节通过实验对比深度共享集成网络在各项任务中的泛化性能.第 5 节进一步探讨一些深度共享集成网络未来的研究想法.

1 研究动机

深度共享集成网络[20]的研究动机来源于深度学习与集成学习(ensemble learning)[21]相融合.其中,集成学习具有非常好的泛化性能,且广泛应用于聚类[22,23]、半监督学习[24]、特征混合[25,26]等领域.深度学习和集成学习两者都可以看作是联结主义模型,从训练过程看,深度学习使用 BP 算法自顶向下的训练模型,而集成学习则选择的是自底向上的构建模型.但在测试阶段,二者都从输入数据开始前向传播执行网络模型.在传统研究中,两种模型的研究是相互独立的,研究人员通常将神经网络当作是集成学习中的基学习器(base learner)来训练网络,然后训练多个基神经网络,但该过程非常耗费资源并且训练周期非常缓慢,因此很难在实际中得到应用.深度学习模型具有很强的表征能力,随着层数的堆叠,网络的表征能力也在提高,但正如没有免费的午餐理论(no free lunch theorems)[27]指出的那样,我们不可能只纯粹地提高网络层数就能提高模型的泛化能力,针对特定的数据,也应该有其适合的网络层数.

深度学习的执行过程也可以看作是特征逐层抽象的过程,从底层的低级特征到顶层的高级抽象特征,原始特征被逐层地表示为更有利于特定的任务形式.因此,从迁移学习(transfer learning)[28]的观点来看,在浅层网络学习到的特征(浅层隐藏单元),也应该有助于将网络扩展到深层时使用,并且从多任务学习(multitask learning)[29]的观点来看,在深度学习的各隐藏层同时加以学习也会更有利于训练出更具泛化性能的抽象特征.

深度共享集成网络也可以看作是共享深度学习隐藏层的集成学习模型,该模型使用多层同时学习的模式,试图从中间层直接注入梯度,以此缓解深层神经网络的梯度消失问题,并使用集成学习与多任务学习的思想训练深度学习模型,与传统方法中集成多个神经网络训练相比,该模型采用共享隐藏层、集成多个输出层的训练方式其训练速度更快,并且内存消耗更小.

2 相关工作

深度共享集成网络可看作是由 Wolpert 提出的堆叠泛化(stacked generalization)[30]的一种扩展形式,该方法是使用采样学习的方式训练多个元模型(meta-model),然后将这些元模型的输出作为下一层元模型的输入进行同样的采样学习,经过多层学习后,堆叠出深层的网络结构.如果堆叠的层数取值恰当,那么这种方法可显著地提升元模型的泛化性能[31,32].由于该方法可任意地与多种元模型结合,因此可以十分便利地提升元模型性能,比如与支持向量机(support vector machines,简称 SVMs)[33]相结合的堆叠支持向量机(stacked SVMs)[34,35],与回归模型结合的堆叠回归(stacked regression)[36],与 Boosting 算法相结合的 DeepBoost[37],而最近提出的深度森林(deep forest)[38]则属于将随机森林堆叠起来的深层网络结构.特别地,堆叠泛化与非监督学习算法中的自动编码器(autoencoder)[39]和受限玻尔兹曼机(restricted Boltzmann machines,简称 RBMs)[40]相结合的堆叠自动编码器(stacked autoencoder,简称 SAEs)[41]以及深度置信网络[42],成为了早期深度学习复兴的重要网络结构.

堆叠泛化最严重的问题在于,网络的每一层一旦堆叠训练出来就不再改变.这种贪心策略很容易将整个网络陷入局部最优值,如果网络的某一层被引入了噪声,则其后的所有层都将受到影响,因此传统的堆叠泛化很难扩展到深层.而堆叠泛化在深度学习的语境中,也被称为逐层监督式预训练[8],该训练方式在逐层训练之后会使用 BP 算法进行调优(fine-tuning),通过梯度的反向传播可以将网络进一步地加以优化.但是,由于 BP 算法本身存在的梯度消失问题,当网络堆叠到较深的层数时,底层的网络权重依然没有办法通过 BP 算法进行梯度修正.在 2014 年提出的 GoogLeNet[6]中,研究人员通过在中间层引入两个输出层来缓解深层神经网络的梯度消失问题.

与堆叠泛化或逐层监督预训练不同,深度共享集成网络并不是一层层堆叠起来的,而是与传统神经网络相同预先设置为深层结构,这样不仅可以提升训练效率还能避免逐层贪心策略所陷入的局部最优解问题.在逐层训练过程中,通常会将底层的输出层去除,仅保留最上层的输出层作为最终的输出,而在深度共享集成网络中,所有的输出层都将得到保留,我们充分利用各层的输出层进行集成,通过实验,结果表明,集成所有输出层结果,往往要好于单个输出层的测试结果.

与本文网络结构相似的还有 2014 年提出的深度监督网络[43],该网络同样使用中间层梯度注入的方式显著提升了卷积神经网络在图像分类问题中的精确度,并且深度监督网络也可看作是本文网络结构在输出层取 1 时的一个特例.但与本文的兴趣点不同,我们主要探讨将该方法作为一个深度学习与集成学习相结合的模型框架,可以方便地嵌套在任意神经网络中.在深度监督网络中,训练过程引入的中间层输出层会在训练结束后删除,而我们的实验结果表明,保留各中间层输出层进行投票输出往往拥有更佳的泛化性能.

3 模型描述

本节首先介绍深度共享集成网络的模型结构,然后再详细地介绍如何训练深度共享集成网络.

3.1 深度共享集成网络模型结构

如图 1(b)所示,与图 1(a)所示的传统神经网络相比,深度共享集成网络的每一层都有多个独立的输出层.数据由输入层经过各级隐藏层进行特征提取,然后再交由各输出层进行分类或回归,最后再将各层的输出结果进行集成,融合成最终结果进行输出.假设 L 表示神经网络的隐藏层数量 $l\in\{1,2,...,L\}$, $a^{(l)}$表示第 l 层的输入向量($a^{(0)}=x$), $w_i^{(l+1)}$ 表示 l 层连接到 l+1 层第 i 神经元的权重向量, $b_i^{(l+1)}$ 表示 l 层连接到 l+1 层第 i 神经元的偏置项,$f(x)$表示任意激活函数,例如:$f(x)=1/(1+\exp(-x))$,则传统神经网络的第 l 层到第 l+1 层的传播如公式(1)和公式(2)所示:

$$s_i^{(l+1)}=w_i^{(l+1)}a^l+b_i^{(l+1)} \tag{1}$$

$$a^{(l+1)}=f(s^{(l+1)}) \tag{2}$$

假设 $\theta_i^{(l+1)}$ 表示第 l 层到 l+1 层第 i 输出层的权重矩阵, $c_i^{(l+1)}$ 表示第 l 层到 l+1 层第 i 输出层的偏置项, $out_i^{(l+1)}$ 表示 l+1 层第 i 个独立输出的输出向量,如果我们使用 softmax 激活函数进行输出,那么深度共享集成网络每一层的输出就如公式(3)和公式(4)所示:

$$o_i^{(l+1)} = \theta_i^{(l+1)} a^{(l+1)} + c_i^{(l+1)} \tag{3}$$

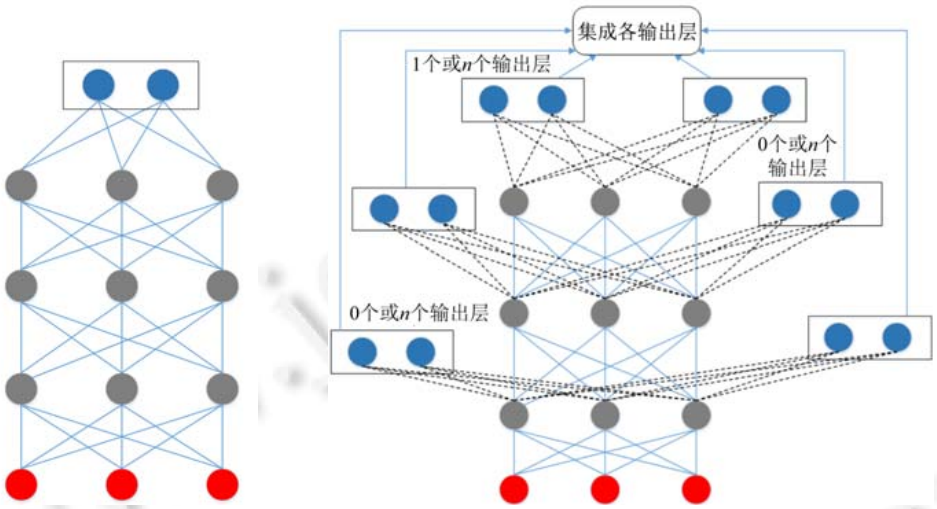
$$out_i^{(l+1)} = softmax(o_i^{(l+1)}) \tag{4}$$

在最终的集成输出时,我们可以采用加权投票的方式进行集成输出.假设需要完成一个 N 分类任务,$N\in\{1,\ldots,n\}$,$\beta_{l,i}$ 表示第 l 层第 i 个输出层投票权重,$z_{l,i}^{(n)}$ 表示第 l 层第 i 个输出层第 n 个输出分类,则深度共享集成网络的集成输出就如公式(5)和公式(6)所示:

$$vote_n = \sum_{l=1}^{L}\sum_{i=1}^{I} \beta_{l,i} z_{l,i}^{(n)} \tag{5}$$

$$ensemble = \max(vote_1, vote_2, \ldots, vote_n) \tag{6}$$

在常用设置中,我们通常将投票权重β设置为 1,也就是使用平权投票的方式进行集成,但是,如果各输出层的验证精度差异较大,也可以设置一个与各层验证精度有关的投票函数进行加权投票.



(a) 标准神经网络 (b) 深度共享集成网络

Fig.1 Diagram of comparison between multilayer shared ensemble network and standard neural network

图 1 深度共享集成网络与标准神经网络结构对比示意图

3.2 模型训练

深度共享集成网络的训练方式与传统神经网络相同,都是使用 BP 算法进行梯度反向传播修改模型权重.但是,由于每个输出层都是独立的,当一条数据传入到深度共享集成网络中时,会有多个输出层进行反向梯度回馈,因此,该网络的共享隐藏层需要进行多层梯度叠加来计算总梯度损失.以二分类任务为例,假设有 m 条数据,上标 i 表示第 i 条数据,其取值范围为 $i\in[1,m]$,$x^{(i)}$表示第 i 条输入数据,$y^{(i)}$表示第 i 条数据的真实分类,$g(x)$表示深度学习的输出,那么深度学习常用的交叉熵损失函数如公式(7)所示:

$$J(w,\theta) = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)}\log(g(x^{(i)},w,\theta)) + (1-y^{(i)})\log(1-g(x^{(i)},w,\theta))) \tag{7}$$

假设$\beta_{l,j}$表示第 l 层第 j 个输出层投票权重,$J_{l,j}(w,\theta)$ 表示第 l 层第 j 个输出层的损失函数,那么深度共享集成网络的损失函数如公式(8)所示:

$$loss_{ensemble} = \sum_{l=1}^{L}\sum_{j=1}^{I} \beta_{l,j} J_{l,j}(w,\theta) \tag{8}$$

假设 $\nabla w^{(l)}$ 表示第 l 层的权重梯度,下标 i 表示第 i 层网络,下标 j 表示第 j 个输出层,由公式(8)可以推出,$\nabla w^{(l)}$ 的计算公式就如公式(9)所示,该公式表明,当前层权重梯度等于当前层之上各输出层的加权代价函数梯度之和.

$$\nabla w^{(l)}=\sum_{i=0}^{L-l}\sum_{j=1}^{I}\beta_{l+i,j}\frac{\partial J_{l+i,j}(w,\theta)}{\partial w^{(l)}} \tag{9}$$

为了方便理解,我们还可以将 $\nabla w^{(l)}$ 表示为当前输出层反向梯度与上层隐藏层反向梯度的形式,假设 $\delta^{(l)}$ 表示第 l 隐藏层反馈梯度,$a^{(l)}$ 表示第 l 层的输入.那么第 l 隐藏层的权重梯度就如公式(10)所示:

$$\nabla w^{(l)}=\delta^{(l)}a^{(l)} \tag{10}$$

假设 $\delta_i^{(l)}$ 表示第 l 层第 i 单元的梯度, $w_{j,i}^{(l)}$ 表示第 l 层第 i 单元连接到第 l+1 层第 j 单元的连接权重, $a_i^{(l)}$ 表示第 l 层第 i 单元的输入,$f'(x)$表示激活函数的导数,s_{l+1} 表示第 l+1 层的神经元个数,那么第 l 层第 i 单元的梯度就如公式(11)所示:

$$\delta_i^{(l)}=\left(\sum_{j=1}^{s_{l+1}}w_{j,i}^{(l)}\delta_j^{(l+1)}\right)f'(a_i^{(l)})+\sum_{j=1}^{I}\beta_{l,j}\frac{\partial J_{l,j}(w,\theta)}{\partial a_i^{(l)}} \tag{11}$$

完整的深度共享集成网络训练过程如算法 1 所示.

算法 1. 随机梯度下降算法用于训练深度共享集成网络.

输入:给定数据集 $X=\{x^{(1)},x^{(2)},...,x^{(n)}\}$,数据集标记 $Y=\{y^{(1)},y^{(2)},...,y^{(n)}\}$,学习率$\alpha$;

输出:输出层权重β,训练迭代次数 iters,网络层数 L,每一层的输出层个数 J.

执行深度共享集成网络前向传播.

1: for i=1:*iters*:

2: 随机采样 m 条训练数据

3: $a^{(0)}\leftarrow x$

4: for l=1:L:

5: 执行每一隐藏层输出: $a^{(l+1)}\leftarrow f(a^{(l)}w^{(l)})$

6: for j=1:J:

7: 执行每一输出层输出: $o_j^{(l+1)}\leftarrow a^{(l+1)}\theta_j^{(l+1)}$

执行多层共享网络反向传播.

8: 计算最上层输出层残差:

$$\delta^{(L)}\leftarrow\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{J}\beta_{L,j}\frac{\partial(y^{(i)}\log(g(o^{(i)}))+(1-y^{(i)})\log(1-g(o^{(i)})))}{\partial o_j^{(l)}}.$$

9: 计算最上层第 j 输出层权重梯度:

$$\nabla\theta_j^{L}\leftarrow\frac{1}{m}\sum_{i=1}^{m}\beta_{L,j}\frac{\partial(y^{(i)}\log(g(o^{(i)}))+(1-y^{(i)})\log(1-g(o^{(i)})))}{\partial\theta_i^{(L)}}.$$

10: 更新最上层第 j 输出层权重: $\theta_j^{(L)}\leftarrow\theta_j^{(L)}+\alpha\nabla\theta_j^{(L)}$

11: for i=1:L−1:

12: $l\leftarrow L-i$

13: for j=1:J:

14: 计算第 l 层第 j 输出层梯度:

$$\nabla o_j^{(l)}\leftarrow\frac{1}{m}\sum_{i=1}^{m}\beta_{L,j}\frac{\partial(y^{(i)}\log(g(o^{(i)}))+(1-y^{(i)})\log(1-g(o^{(i)})))}{\partial o_j^{(l)}}$$

15: 计算第 l 层第 j 输出层权重梯度:

$$\nabla\theta_j^{(l)}\leftarrow\frac{1}{m}\sum_{i=1}^{m}\beta_{L,j}\frac{\partial(y^{(i)}\log(g(o^{(i)}))+(1-y^{(i)})\log(1-g(o^{(i)})))}{\partial\theta_i^{(l)}}$$

16: 更新第 l 层第 j 输出层权重: $\theta_j^{(l)}\leftarrow\theta_j^{(l)}+\alpha\nabla\theta_j^{(l)}$

17: 计算第 l 隐藏层残差: $\delta^{(l)}=\left(\sum_{j=1}^{s_{l+1}} w_j^{(l)}\delta^{(l+1)}\right) f'(a^{(l)})+\sum_{j=1}^{J}\nabla o_j^{(l)}$

18: 计算第 l 隐藏层权重梯度: $\nabla w^{(l)}=\delta^{(l)}a^{(l)}$

19: 更新第 l 隐藏层权重: $w^{(l)}\leftarrow w^{(l)}+\alpha\nabla w^{(l)}$

4 实验比较

在本节中,我们使用深度共享集成方法构建了深度前馈神经网络、深度卷积网络以及深度循环网络,分别命名为深度共享集成前馈神经网络、深度共享卷积集成网络、深度共享循环集成网络,并比较了各自基础网络在各项任务中的性能.通过实验比较我们发现,深度共享集成方法可以普遍提高基础网络的泛化性能.以下是我们所比较的实验数据集.

- MNIST[44]:标准的手写数字图像数据集;
- CIFAR-10[45]:标准的小型自然图像数据集;
- GTZAN[46]:用于音乐分类的音频数据集;
- sEMG[47]:用于手势识别时序传感器数据集;
- IMDB[48]:用于情感分类的电影评论英文文本数据集;
- UCI 数据集[49]:LETTER、ADULT、YEAST 低维数据集.

如表 1 所示,我们使用不同领域、不同维度数据集进行对比实验,并统一将数据集划分为训练集与测试集两个部分.

Table 1 Data set partition

表 1 实验数据集划分

数据集	数据应用领域	数据维度	训练数据集	测试数据集
MNIST	数字图像识别	784(28×28 灰度图)	60 000	10 000
CIFAR-10	小型自然图像识别	3 072(32×32×3 彩色图)	50 000	10 000
GTZAN	语音识别	3 840(1280×3)	700	300
sEMG	手势识别	3 000	1 440	360
IMDB	文本分类	500(截断为 500 长度)	2 500	2 500
LETTER	低维分类	16	16 000	4 000
ADULT	低维分类	14	32 561	16 281
YEAST	低维分类	8	1 038	446

在实验中,除 MNIST 数据集中的逐层对比实验,网络结构都采取按层衰减根号 2 倍的金字塔结构,使用的卷积核为 3×3,卷积方式为网络尺寸不变的 same 卷积操作;我们使用跨步为 2 的卷积操作替换池化方法进行下采样,每一层网络都使用批量归一化算法进行网络解耦,并使用 leaky ReLU 作为激活函数进行非线性激活;在共享集成中,每两层接入 5 个全连接层,并使用 softmax 进行输出.在训练时,每一层网络权重的方差初始化为 0.5,学习率初始化为 0.5,最低学习率为 0.001,使用学习率指数衰减的方式进行训练,并使用 adam 梯度下降算法进行权重更新.表 2 给出测试 CIFAR-10 数据集的网络结构.

Table 2 Architecture on test CIFAR-10

表 2 测试 CIFAR-10 的网络结构

输入	CNN 32×32×3	CNN+sharing ensemble 32×32×3	DNN 3 072	DNN+sharing ensemble 3 072
L1	same 卷积:3×3,64 stride 1,BN,leaky ReLU 输出维度:32×32×64	same 卷积:3×3,64 stride 1,BN,leaky ReLU 输出维度:32×32×64	全连接层:3072×1024 BN,leaky ReLU 输出维度:1 024	全连接层:3072×1024 BN,leaky ReLU 输出维度:1 024
L2	same 卷积:3×3,64 stride 2,BN,leaky ReLU 输出维度:16×16×64	same 卷积:3×3,64 stride 2,BN,leaky ReLU 输出维度:16×16×64	全连接层:1024×512 BN,leaky ReLU 输出维度:512	全连接层:1024×512 BN,leaky ReLU 输出维度:512
		5 个全连接层:16384×10×5 5 个 softmax 层		5 个全连接层:512×10 5 个 softmax 层

Table 2 Architecture on test CIFAR-10 (Continued)
表 2 测试 CIFAR-10 的网络结构(续)

输入	CNN+sharing ensemble 32×32×3	DNN 32×32×3	DNN+sharing ensemble 3 072	CNN 3 072
L3	same 卷积:3×3,64 stride 1,BN,leaky ReLU 输出维度:16×16×64	same 卷积:3×3,64 stride 1,BN,leaky ReLU 输出维度:16×16×64	全连接层:512×512 BN,leaky ReLU 输出维度:512	全连接层:512×512 BN,leaky ReLU 输出维度:512
L4	same 卷积:3×3,128 stride 2,BN,leaky ReLU 输出维度:8×8×128	same 卷积:3×3,128 stride 2,BN,leaky ReLU 输出维度:8×8×128	全连接层:512×512 BN,leaky ReLU 输出维度:512	全连接层:512×512 BN,leaky ReLU 输出维度:512
		5 个全连接层:8192×10×5 5 个 softmax 层		5 个全连接层:512×10 5 个 softmax 层
L5	same 卷积:3×3,128 stride 1,BN,leaky ReLU 输出维度:8×8×128	same 卷积:3×3,128 stride 1,BN,leaky ReLU 输出维度:8×8×128	全连接层:512×256 BN,leaky ReLU 输出维度:256	全连接层:512×256 BN,leaky ReLU 输出维度:256
L6	same 卷积:3×3,256 stride 2,BN,leaky ReLU 输出维度:4×4×256	same 卷积:3×3,256 stride 2,BN,leaky ReLU 输出维度:4×4×256	全连接层:256×256 BN,leaky ReLU 输出维度:256	全连接:256×256 BN,leaky ReLU 输出维度:256
		5 个全连接层:4096×10×5 5 个 softmax 层		5 个全连接层:256×10 5 个 softmax 层
L7	same 卷积:3×3,256 stride 1,BN,leaky ReLU 输出维度:4×4×256	same 卷积:3×3,256 stride 1,BN,leaky ReLU 输出维度:4×4×256	全连接层:256×128 BN,leaky ReLU 输出维度:128	全连接:256×128 BN,leaky ReLU 输出维度:512
L8	same 卷积:3×3,256 stride 2,BN,leaky ReLU 输出维度:2×2×256	same 卷积:3×3,256 stride 2,BN,leaky ReLU 输出维度:2×2×256	全连接层:128×128 BN,leaky ReLU 输出维度:128	全连接层:128×128 BN,leaky ReLU 输出维度:128
		5 个全连接层:1024×10 5 个 softmax 层		全连接层:128×10 5 个 softmax 层
输出	全连接层:1024×10 softmax 层输出	对 20 个 softmax 层进行平权投票输出	全连接层:128×10 softmax 层输出	对 20 个 softmax 层进行平权投票输出

4.1 深度共享集成网络降低梯度消失

当深度学习网络层数逐渐加深时,其网络便会出现梯度消失现象.特别是当使用易饱和性神经元(如:tanh、sigmoid)时,深度学习就会变得非常难以训练,通常需要非常小心地初始化参数范围,并对数据进行归一化处理.为了检验深度共享集成网络的防止梯度消失性能,我们使用 MNIST 数据集对深度卷积神经网络与深度共享卷积集成网络做逐层对比实验.如图 2 所示.

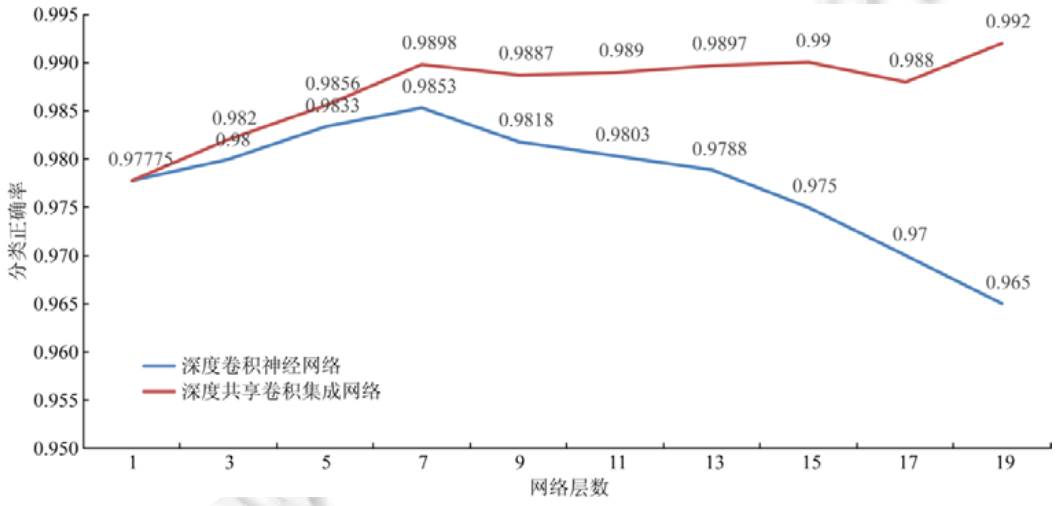


Fig.2 Layer by layer comparison experiment of deep CNN and deep sharing ensemble CNN on MNIST
图 2 深度卷积神经网络与深度共享卷积集成网络在 MNIST 数据集上的逐层比较实验

在实验中,我们每一层都只使用 3×3×64 的卷积核进行 same 卷积操作,为了保证网络不缩减,我们没有使用池化操作,并且卷积的跨步为 1,在深度神经网络中每两层卷积层后接入一个全连接层并使用 softmax 函数进行输出,在深度共享集成网络中,每两层卷积层后接入 5 个全连接层并各自使用 softmax 函数输出,最后再使用平权投票的方式进行最后输出,为模拟梯度消失现象,我们选择了 tanh 作为神经网络的激活函数,并每两层比较网络的泛化性能.实验中我们使用的数据批量尺寸为 100,梯度下降算法为 adam,起始学习率为 0.1,最低学习率为 0.001,使用学习率指数衰减的方式进行训练.通过实验比较,如图 2 所示,我们发现,深度共享卷积集成网络在网络层数较深时,可以有效地缓解梯度消失情况,即使网络层数不断增加,也不会损坏其泛化性能.

4.2 CIFAR-10图像识别实验比较

CIFAR-10 数据集是一套深度学习领域测试图像识别的基准数据集,该数据集总共包含 60 000 张 32×32×3 的 10 分类小型自然图片.如表 3 所示,我们分别比较了使用神经网络与卷积神经网络配合深度共享集成方法构建起来的网络性能.标准的神经网络的测试正确率为 44.50%,而加上深度共享集成之后的正确率为 47.35%.在卷积网络的测试中,我们选取了 AlexNet 作为基础网络,其正确率可达 83%,而通过在中间层接入多个独立输出层之后,其网络性能可以提升到 84.23%.同时,为了方便比较,我们也加入其他一些算法作为比较参考.

Table 3 Compare the test accuracy of CIFAR-10 dataset (%)

表 3 比较 CIFAR-10 数据集测试正确率(%)

ResNet	93.57[4]
gcForest(gbdt)	69.00[38]
gcForest(5grains)	63.37[38]
Deep Belief Net	62.20[38]
gcForest(default)	61.78[38]
Random Forest	50.17[38]
Logistic Regression	37.32[38]
SVM (linear kernel)	16.32[38]
DNN	44.50
DNN+sharing ensemble	**47.35**
Deep CNN(AlexNet)	83.00
Deep CNN+sharing ensemble	**84.23**

4.3 GTZAN音乐分类实验比较

GTZAN 数据集包含了 100 首 10 种音乐风格音频数据,每首音频的时长为 30s.我们将该数据划分 700 首作为训练数据,300 首作为测试数据.在实验中,我们使用 MFCC 特征来表示 30s 的音频数据,因此将每首 30s 的原始音频转换为了 1280×32 的特征矩阵.如表 4 所示,我们构建了深度共享集成前馈网络、深度共享卷积集成网络及其对照网络进行实验,在实验中,深度神经网络以及深度卷积网络的测试正确率分别为 60.33%和 63.00%,而使用了深度共享集成方法后的模型正确率分别可以达到 61.20%和 64.33%.

Table 4 Compare the test accuracy of GTZAN dataset (%)

表 4 比较 GTZAN 数据测试正确率(%)

gcForest	65.67[38]
Random Forest	50.33[38]
Logistic Regression	50.00[38]
SVM (rbf kernel)	18.33[38]
DNN	60.33
DNN+sharing ensemble	**61.20**
Deep CNN	63.00
Deep CNN+sharing ensemble	**64.33**

4.4 sEMG手势识别实验比较

sEMG 数据集包含了 1 800 条 6 种手势移动数据,每条数据的特征为 3 000 维.我们将其划分为 1 440 条作为训练数据集,360 条作为测试数据集.使用深度共享集成前馈网络、深度共享集成 LSTM 及其对照网络模型进行实验,在 LSTM 的构建实验中,将原始 3 000 维的数据重塑为 6×00 的数据,其中,6 作为训练长度,300 作为每一

时间片段输入到 LSTM 中的输入维度.如表 5 所示,通过实验比较,深度神经网络以及深度 LSTM 的测试正确率分别在 42.23%以及 47.33%,而加入深度共享集成之后的正确率分别提升到 44.52%和 48.00%.但同时也发现,深度共享集成方法对于 LSTM 性能的提升所起作用十分微小,并且在有些情况下,反而会损坏 LSTM 的性能.

Table 5 Compare the test accuracy of sEMG dataset (%)

表 5 比较 sEMG 数据测试正确率(%)

gcForest	71.30[38]
Random Forest	29.62[38]
SVM (rbf kernel)	29.62[38]
Logistic Regression	23.33[38]
DNN	42.23
DNN+sharing ensemble	**44.52**
Deep LSTM	47.33
Deep LSTM+sharing ensemble	**48.00**

4.5 IMDB文本情感分类实验比较

IMDB 数据集总共包含了 50 000 份英文影评进行情感分类,其中,25 000 份作为训练数据,25 000 份作为测试数据.在实验中,将每份文本长度截断为 500,文本长度不足时进行补零处理,由于原始文本被表示为 tf-idf 特征,为此我们统一将 tf-idf 特征转换为 100 维的词向量进行实验,在卷积网络的实验中,我们使用 1 维卷积构建模型.如表 6 所示,分别构造深度共享集成前馈网络、深度共享卷积网络、深度共享 LSTM 以及各自对应的网络模型进行比较.通过实验比较,深度神经网络、深度卷积网络以及深度 LSTM 的测试正确率分别在 88.23%、91.35%以及 93.34%,而加入深度共享集成之后的正确率分别提升到 89.56%、92.00%以及 93.78%.

Table 6 Compare the test accuracy of IMDB dataset (%)

表 6 比较 IMDB 数据测试正确率(%)

gcForest	89.16[38]
Logistic Regression	88.62[38]
SVM (linear kernel)	87.56[38]
Random Forest	85.32[38]
DNN	88.23
DNN+sharing ensemble	**89.56**
Deep CNN	91.35
Deep CNN+sharing ensemble	**92.00**
Deep LSTM	93.34
Deep LSTM+sharing ensemble	**93.78**

4.6 UCI低维时间序列分类实验比较

我们同样也使用深度共享集成网络在一些 UCI 低维数据集上进行实验对比.选取了 LETTER、ADULT、YEAST 这 3 套数据集进行实验.其中,LETTER 数据集包含了 16 个特征,16 000 条作为训练数据,4 000 条作为测试数据;ADULT 包含了 14 个特征,32 561 条作为训练数据,16 281 条作为测试数据;YEAST 包含 8 个特征,1 038 作为训练数据,446 作为测试数据.由于这 3 套数据集的特征数量较少,不太需要类似 CNN、LSTM 这类复杂的网络结构,因此我们仅构造深度共享集成前馈网络进行比较.如表 7 所示,在 LETTER 数据集中,深度神经网络的测试正确率为 95.70%,而深度共享集成前馈网络的测试正确率为 96.88%;在 ADULT 数据集中,深度神经网络的正确率为 85.25%,而深度共享集成前馈网络的测试正确率为 85.70%;在 YEAST 数据集中,深度神经网络的正确率仅为 55.60%,而深度共享集成前馈网络的测试正确率为 58.00%.

Table 7 Compare the test accuracy of UCI low dimensional dataset (%)

表 7 比较 UCI 低维数据测试正确率(%)

	LETTER	ADULT	YEAST
gcForest	97.40[38]	86.40[38]	63.45[38]
Random Forest	96.50[38]	85.49[38]	61.66[38]
DNN	95.70	85.25	55.60
DNN+sharing ensemble	**96.88**	**85.70**	**58.00**

5 结 论

深度共享集成网络是一种非常灵活的集成学习框架,相比于传统的集成学习算法,通过共享隐藏层集成的方式,深度集成网络可以极大地减少训练参数的数量,节约训练周期,并且还可以轻松地将该算法扩展为深度共享集成卷积网络以及深度共享集成循环网络等常用的深度学习模型.在元模型不变的情况下,深度共享集成方法可以普遍提高模型 1%~3%的泛化性能.

但深度共享集成方法也带来了额外的问题,即增加了深度学习的超参数种类,哪些层应该引入输出层,每一层需要设置多少个输出层都没有固定的配置,需要根据具体任务经验性地进行超参数配置.同时,在实验中,我们发现深度共享集成网络对于循环神经网络的性能提升所起作用很有限,特别是当循环网络的循环周期较长时,深度共享集成方法对网络的性能提升所起作用非常微小,有时甚至还会损坏网络的性能,这也是在未来工作中需要研究的地方.深度共享集成方法的泛化性能还非常依赖于基础网络的结构,如在处理图像识别任务时,深度共享集成神经网络与深度共享集成卷积网络的性能差异非常巨大,而该差距严重地受到基础网络的影响,因此,如何构建更具通用性的深度共享集成架构依然是未来工作中需要研究的地方.

References:

[1] Bengio Y. Learning deep architectures for AI. Foundations & Trends® in Machine Learning, 2009,2(1):1–127.

[2] Lecun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015,521(7553):436.

[3] Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In: Proc. of the Automatic Speech Recognition and Understanding. IEEE, 2016. 167–174.

[4] He K, Zhang X, Ren S, *et al*. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016,770–778.

[5] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012,1097-1105.

[6] Szegedy C, Liu W, Jia YQ, Sermanet P, Reed SE, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proc. of the CVPR. 2015. 1–9.

[7] Cui Y, Wang S, Li J. LSTM neural reordering feature for statistical machine translation. Computer Science, 2015,(2).

[8] Bengio Y, Lamblin P, Popovici D, *et al*. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 2007,153-160.

[9] Erhan D, Manzagol PA, Bengio Y, *et al*. The difficulty of training deep architectures and the effect of unsupervised pre-training. Immunology of Fungal Infections, 2009,5:153–160.

[10] Mesnil G, Dauphin Y, Glorot X, *et al*. Unsupervised and transfer learning challenge: A deep learning approach. Workshop on Unsupervised & Transfer Learning, 2012,7:1–15.

[11] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans. on Neural Networks, 2002,5(2):157–166.

[12] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998,6(02):107-116.

[13] Pascanu R, Mikolov T, Bengio Y. Understanding the exploding gradient problem. Arxiv Preprint Arxiv, 2012.

[14] Dauphin YN, Pascanu R, Gulcehre C, *et a*l. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. Advances in neural information processing systems, 2014,2933-2941.

[15] Lecun Y, Boser B, Denker JS, *et al*. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1989,1(4):541–551.

[16] Goodfellow IJ, Warde-Farley D, Mirza M, *et al*. Maxout networks. Computer Science, 2013,1319–1327.

[17] He K, Zhang X, Ren S, *et al*. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. Proceedings of the IEEE international conference on computer vision, 2015,1026–1034.

[18] Nair V, Hinton G E. Rectified linear units improve restricted Boltzmann machines. In: Proc. of the Int'l Conf. on Machine Learning. Omnipress, 2010. 807–814.

[19] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. of the Int'l Conf. on Machine Learning. 2015,448–456.

[20] Du F. Research on deep sharing ensemble networks [MS. Thesis]. Kunming: Yunnan University, 2018 (in Chinese with English abstract).

[21] Dietterich, Thomas G. Ensemble methods in machine learning. In: Proc. of the Int'l Workshgp on Multiple Classifier Systems. 2000,1857(1):1–15.

[22] Yang Y, Chen K. Time series clustering via RPCL network ensemble with different representations. IEEE Trans. on Systems Man & Cybernetics Part C, 2011,41(2):190–199.

[23] Yang Y, Chen K. Temporal data clustering via weighted clustering ensemble with different representations. IEEE Trans. on Knowledge & Data Engineering, 2010,23(2):307–320.

[24] Yang Y, Liu X. A Robust Semi-supervised Learning Approach Via Mixture of Label Information. Elsevier Science Inc., 2015.

[25] Yang Y, Jiang J. Hybrid sampling-based clustering ensemble with global and local constitutions. IEEE Trans. on Neural Networks & Learning Systems, 2017,27(5):952–965.

[26] Yang Y, Jiang J. HMM-based hybrid meta-clustering ensemble for temporal data. Knowledge-based Systems, 2014,56(C):299–310.

[27] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation, 1997,1(1):67–82.

[28] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans. on Knowledge & Data Engineering, 2010,22(10):1345–1359.

[29] Caruana, Richard A. Multitask connectionist learning. In Proceedings of the 1993 Connectionist Models Summer School, 1993,372–379.

[30] Wolpert DH. Stacked Generalization. Springer US, 2011.

[31] Kai MT, Witten IH. Stacked generalization: when does it work. In: Proc. of the 15th Int'l Joint Conf. on Artifical Intelligence. Morgan Kaufmann Publishers Inc., 1997,866–871.

[32] Kai MT, Witten IH. Issues in Stacked Generalization. AI Access Foundation, 1999.

[33] Burges, Christopher JC. A tutorial on support Vector Machines for Pattern Recognition. Data mining and knowledge discovery, 1998, 2(2): 121-167.

[34] Chen J, Wang C, Wang R. Using stacked generalization to combine SVMs in magnitude and shape feature spaces for classification of hyperspectral data. IEEE Trans. on Geoscience & Remote Sensing, 2009,47(7):2193–2205.

[35] Ness SR, Theocharis A, Tzanetakis G, *et a*l. Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs. In: Proc. of the Int'l Conf. on Multimedia 2009. DBLP, 2009. 705–708.

[36] Breiman L. Stacked regressions. Machine Learning, 1996, 24(1):49–64.

[37] Kuznetsov V, Mohri M, Syed U. Multi-class deep boosting. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. MIT Press, 2014. 2501–2509.

[38] Zhou Z H, Feng J. Deep forest. arXiv preprint arXiv:1702.08835, 2017.

[39] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science, 2006,313(5786):504–507.

[40] Hinton GE. A practical guide to training restricted boltzmann machines. Momentum, 2012,9(1):599–619.

[41] Vincent P, Larochelle H, Lajoie I, *et al*. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 2010,11(12):3371–3408.

[42] Ranzato MA, Boureau YL, Lecun Y. Sparse feature learning for deep belief networks. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2007. 1185–1192.

[43] Lee C Y, Xie S, Gallagher P, *et al*. Deeply-supervised nets. Artificial intelligence and statistics, 2015,562-570.

[44] Lécun Y, Bottou L, Bengio Y, *et al*. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998,86(11): 2278–2324.

[45] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Citeseer, 2009.

[46] zanetakis G, Cook P. Musical genre classification of audio signals. IEEE Trans. on Speech & Audio Processing, 2002,10(5): 293–302.

[47] Sapsanis C, Georgoulas G, Tzes A, *et al*. Improving EMG based classification of basic hand movements using EMD. In: Proc. of the Engineering in Medicine and Biology Society. IEEE, 2013,5754–5757.

[48] Maas AL, Daly RE, Pham PT, *et al*. Learning word vectors for sentiment analysis. In: Proc. of the Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics. 2011,142–150.

[49] Bache K, Lichman M. UCI machine learning repository. University of California, School of Information and Computer Science, 2013, http://archive.ics.uci.edu/ml

附中文参考文献:

[20] 杜飞.深度共享集成网络的研究[硕士学位论文].昆明:云南大学,2018.

杜飞(1991－),男,硕士,主要研究领域为深度学习,集成学习.

胡媛媛(1995－),女,硕士生,CCF 学生会员,主要研究领域为深度学习.

杨云(1981－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为机器学习,数据挖掘,模式识别,时间数据处理与分析.

曹丽娟(1994－),女,硕士生,主要研究领域为深度学习.