软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(7):2071−2091 [doi: 10.13328/j.cnki.jos.005561]

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

深度神经网络训练中梯度不稳定现象研究综述*

陈建廷, 向 阳

(同济大学 电子信息与工程学院,上海 201804)

通讯作者: 向阳, E-mail: shxiangyang@tongji.edu.cn

摘 要: 深度神经网络作为机器学习领域的热门研究方向,在训练中容易出现梯度不稳定现象,是制约其发展的重要因素,控制和避免深度神经网络的梯度不稳定现象是深度神经网络的重要研究内容.分析了梯度不稳定现象的成因和影响,并综述了目前解决梯度不稳定现象的关键技术和主要方法.最后展望了梯度不稳定现象的未来研究方向.

关键词: 深度神经网络;梯度不稳定现象;梯度衰减;梯度爆炸

中图法分类号: TP183

中文引用格式: 陈建廷,向阳.深度神经网络训练中梯度不稳定现象研究综述.软件学报,2018,29(7):2071−2091. http://www.jos.org.cn/1000-9825/5561.htm

英文引用格式: Chen JT, Xiang Y. Survey of unstable gradients in deep neural network training. Ruan Jian Xue Bao/Journal of Software, 2018,29(7):2071−2091 (in Chinese). http://www.jos.org.cn/1000-9825/5561.htm

Survey of Unstable Gradients in Deep Neural Network Training

CHEN Jian-Ting, XIANG Yang

(College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

Abstract: As a popular research direction in the field of machine learning, deep neural networks are prone to the phenomenon of unstable gradients in training, which has become an important element that restricts their development. How to avoid and control unstable gradients is an important research topic of deep neural networks. This paper analyzes the cause and effect of the unstable gradients, and reviews the main models and methods of solving the unstable gradients. Furthermore, the future research trends in the unstable gradients is discussed.

Key words: deep neural network; unstable gradient; vanishing gradient; exploding gradient

深度神经网络作为深度学习领域的重要模型之一,在计算机视觉、语音识别等领域取得巨大突破.由于训练多层神经网络得到最优参数是非确定性多项式困难问题(non-deterministic polynomial hard,简称 NP-hard)[1],所以神经网络的训练过程成为影响最终效果的核心因素.梯度下降(gradient descent)算法作为神经网络的主要训练方法,但在将其应用在深度神经网络时易出现梯度不稳定现象.该现象严重影响了模型的实际效果,导致准确率降低,收敛速度缓慢等,使得深度神经网络模型更加难以训练,后续研究与实践应用也受到阻碍,因此,梯度不稳定现象已成为制约深度神经网络模型发展的关键问题,受到学术界与工业界的高度关注.

学者们通过在神经网络中引入深度特性来提高提取特征的能力,但是梯度不稳定现象随着前馈神经网络

* 基金项目: 国家重点基础研究发展计划(973)(2014CB340404); 国家自然科学基金(71571136); 上海市科委基础研究项目(16JC403000)

Foundation item: National Basic Research Program of China (973) (2014CB340404); National Natural Science Foundation of China (71571136); Project of Science and Technology Commission of Shanghai Municipality (16JC403000)

收稿时间: 2017-09-27; 修改时间: 2017-11-10; 采用时间: 2018-01-10; jos 在线出版时间: 2018-02-08

CNKI 网络优先出版: 2018-02-08 11:56:10, http://kns.cnki.net/kcms/detail/11.2560.TP.20180208.1155.013.html

的层数和循环神经网络时间序列长度的增加也愈发严重,严重影响深度优势的发挥和训练收敛速度.自从 1994 年首次提出梯度衰减[2,3]的不稳定现象以来,学者们提出了一系列改进方法来缓解该问题,大致可分成 3 类:一是改进训练方法,如结合无监督学习的思想设计训练策略;二是优化节点运算,如采用 ReLU、Batch-Normalization 等技巧优化影响梯度传播的关键运算;三是调整网络结构,如以 LSTM 模型代替传统 RNN 模型以及采用捷径连接技巧改变传统迭代形式,这些改进在一定程度上缓解了梯度不稳定现象,使得深度神经网络能够充分训练,发挥其在机器学习领域的优势.

本文详细分析出现梯度不稳定现象的原因,并综述目前缓解梯度不稳定现象的主要模型和方法.本文第 1 节阐述梯度不稳定现象的成因及影响.第 2 节论述缓解梯度不稳定现象的主要方法.第 3 节展望梯度不稳定现象的未来研究方向.第 4 节总结全文.

1 梯度不稳定现象成因及影响

梯度不稳定现象是指采用梯度下降法训练的深度神经网络,在利用反向传播算法(back-propagation algorithm,简称 BP)[4−6]计算各参数梯度的迭代过程中,根据链式求导法则,迭代量需要乘以各中间变量或参数来不断更新,若“乘数”均远远大于“1”,则更新结果随迭代过程迅速增加,发生梯度爆炸的不稳定现象;若“乘数”均远远小于“1”,则该结果将随迭代迅速减小,发生梯度衰减的不稳定现象.若深度神经网络中发生此类梯度不稳定现象,将难以充分发挥深度结构优势,使其深度特性失效,还可能导致前馈神经网络收敛速度缓慢,影响训练效率.本节将通过分析典型前馈神经网络和循环神经网络反向传播的梯度计算过程,总结梯度不稳定现象的成因,并结合具体实验,说明该现象对深度神经网络的影响.

1.1 前馈神经网络梯度计算

(1) 全连接神经网络梯度计算

全连接神经网络(fully connected neural network,简称 FNN)[5,6]是最典型的前馈神经网络,深度全连接神经网络中包含多个全连接隐层,隐层中每个节点与前一层所有节点全连接.m 层的全连接网络结构如图 1 所示,其前馈传播过程可形式化表示为

$$\boldsymbol{y}_l = \boldsymbol{W}_l \boldsymbol{x}_{l-1} + \boldsymbol{b}_l \tag{1}$$

$$\boldsymbol{x}_l = \sigma(\boldsymbol{y}_l) \tag{2}$$

$\boldsymbol{x}_l$,$\boldsymbol{y}_l$ 表示第 l 层的输出向量和中间向量,其中的元素表示该层各节点相应的变量.$\boldsymbol{W}_l$ 和 $\boldsymbol{b}_l$ 为模型参数,$\boldsymbol{W}_l$ 表示第 l–1 层与第 l 层之间的连接权重矩阵,其中,元素所在行数和列数分别对应连接节点在隐层中的序号.$\boldsymbol{b}_l$ 表示第 l 层的偏置向量.函数 σ 表示非线性激活函数.

训练样本的输入数据经过前向传播到输出层与标签比较计算损失值,梯度下降法根据各参数对损失值的梯度进行参数调整.计算各隐层输出值梯度的反向传播迭代过程为

$$\frac{\partial e}{\partial \boldsymbol{x}_{l-1}} = \boldsymbol{W}_l^T \left[\sigma'(\boldsymbol{y}_l) \times \frac{\partial e}{\partial \boldsymbol{x}_l} \right] \tag{3}$$

其中,“×”表示 hadamard 乘积,即同维矩阵(或向量)对应元素相乘;e 表示损失函数值.隐层每个节点导数值的具体计算公式,即公式(3)从元素角度可表示为

$$\left[\frac{\partial e}{\partial \boldsymbol{x}_{l-1}} \right]_i = \sum_j^{n_l} \left[\sigma'(\boldsymbol{y}_l) \right]_j \cdot \left[\frac{\partial e}{\partial \boldsymbol{x}_l} \right]_j \cdot \left[\boldsymbol{W}_l \right]_{i,j} \tag{4}$$

其中,n_l 表示第 l 层的节点总数,i,j 分别表示第 l–1 层和第 l 层节点序号.由隐层梯度计算各隐层权重和偏置梯度的公式为

$$\frac{\partial e}{\partial \boldsymbol{W}_l} = \boldsymbol{x}_{l-1} \left[\sigma'(\boldsymbol{y}_l) \times \frac{\partial e}{\partial \boldsymbol{x}_l} \right] \tag{5}$$

$$\frac{\partial e}{\partial \boldsymbol{b}_l}=\sigma'(\boldsymbol{y}_l)\times\frac{\partial e}{\partial \boldsymbol{x}_l} \tag{6}$$

梯度下降法根据梯度对参数更新,$\boldsymbol{W}_l=\boldsymbol{W}_l-\eta\frac{\partial e}{\partial \boldsymbol{W}_l},\boldsymbol{b}_l=\boldsymbol{b}_l-\eta\frac{\partial e}{\partial \boldsymbol{b}_l}$,其中,$\eta$为学习率.

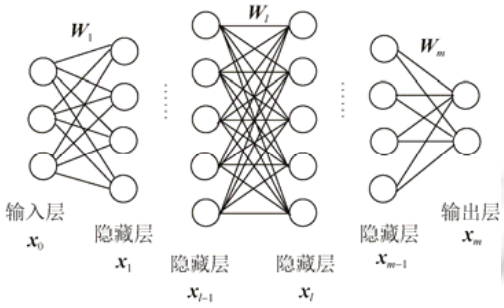


Fig.1 Architecture of fully connected neural network

图 1 全连接神经网络结构

(2) 卷积神经网络梯度计算

卷积神经网络(convolution neural network,简称 CNN)[7,8]作为另一种重要的前馈神经网络,数据仍然是以当前层输出作为下一层输入的形式在多隐层结构中传播,区别在于隐层节点的输入输出不是数而是矩阵,连接前后隐层节点之间的不是权重而是卷积核.其前馈传播过程如图 2 所示.

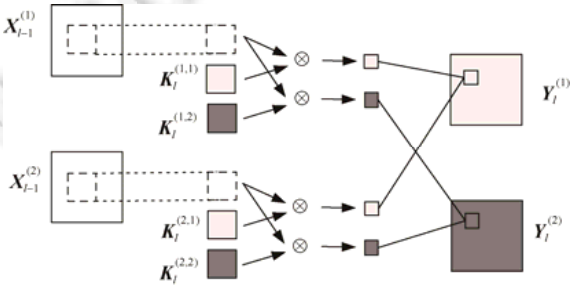


Fig.2 Process of convolution layer feed forward propagation

图 2 卷积层前馈传播过程

可形式化表示为

$$\boldsymbol{Y}_l=\boldsymbol{K}_l\otimes\boldsymbol{X}_{l-1}+\boldsymbol{b}_l \tag{7}$$

$$\boldsymbol{X}_l=\sigma(\boldsymbol{Y}_l) \tag{8}$$

$\boldsymbol{X}_l,\boldsymbol{Y}_l$表示第 l 层的输出变量和中间变量,若第 l 层有 n_l 个节点,则 $\boldsymbol{X}_l,\boldsymbol{Y}_l$由 n_l 个矩阵组成,每个矩阵对应一个节点,即 $\boldsymbol{X}_l=\left\{\boldsymbol{X}_l^{(1)},\boldsymbol{X}_l^{(2)},...,\boldsymbol{X}_l^{(n_l)}\right\},\boldsymbol{Y}_l=\left\{\boldsymbol{Y}_l^{(1)},\boldsymbol{Y}_l^{(2)},...,\boldsymbol{Y}_l^{(n_l)}\right\}$.$\boldsymbol{K}_l$表示卷积核,第 l–1 层的第 m 个节点与第 l 层的第 n 个节点之间由大小为 $k\times k$ 的矩阵 $\boldsymbol{K}_l^{(m,n)}$进行特殊的卷积运算.$\boldsymbol{b}_l$仍表示由 n_l 个元素组成的偏置向量.假定卷积核移动步长为 1,则公式(7)的卷积运算从元素的角度可表示为

$$\left[\boldsymbol{Y}_l^{(n)}\right]_{i,j}=\sum_m^{n_{l-1}}\sum_{u,v}^{k}\left[\boldsymbol{K}_l^{(m,n)}\right]_{u,v}\cdot\left[\boldsymbol{X}_{l-1}^{(m)}\right]_{i+u-1,j+v-1}+\left[\boldsymbol{b}_l\right]_n \tag{9}$$

卷积层的梯度计算仍采用反向传播算法,根据链式求导法,其推导的迭代计算公式为

$$\frac{\partial e}{\partial \boldsymbol{X}_{l-1}}=\boldsymbol{K}_l\odot\left[\sigma'(\boldsymbol{Y}_l)\times\frac{\partial e}{\partial \boldsymbol{X}_l}\right] \tag{10}$$

“⊙”表示特殊的反卷积运算,具体的隐层节点输出矩阵中元素的导数为

$$\left[\frac{\partial e}{\partial \boldsymbol{X}_{l-1}^{(m)}}\right]_{i+u-1,j+v-1}=\sum_{n}^{n_l}\sum_{u,v}^{k}[\sigma'(\boldsymbol{Y}_l^{(n)})]_{i,j}\cdot\left[\frac{\partial e}{\partial \boldsymbol{X}_l^{(n)}}\right]_{i,j}\cdot\left[\boldsymbol{K}_l^{(m,n)}\right]_{u,v} \tag{11}$$

卷积核梯度和偏置向量梯度中的元素值分别为

$$\left[\frac{\partial e}{\partial \boldsymbol{K}_l^{(m,n)}}\right]_{u,v}=\sum_{i,j}^{a,b}\left[\sigma'(\boldsymbol{Y}_l^{(n)})\right]_{i,j}\cdot\left[\frac{\partial e}{\partial \boldsymbol{X}_l^{(n)}}\right]_{i,j}\cdot\left[\boldsymbol{X}_{l-1}^{(m)}\right]_{i+u-1,j+v-1} \tag{12}$$

$$\left[\frac{\partial e}{\partial \boldsymbol{b}_l}\right]_{n}=\sum_{i,j}^{a,b}\left[\sigma'(\boldsymbol{Y}_l^{(n)})\right]_{i,j}\cdot\left[\frac{\partial e}{\partial \boldsymbol{X}_l^{(n)}}\right]_{i,j} \tag{13}$$

其中,a,b 表示 $\boldsymbol{X}_l^{(n)}$矩阵的维度.上述公式(11)~公式(13)即是移动步长为 1 的卷积层的反向梯度计算过程.梯度下降法根据梯度来更新卷积核和偏置向量.

1.2 循环神经网络梯度计算

循环神经网络(recurrent neural network,简称 RNN)[9,10]作为处理序列信息的网络模型,完成一次预测或分类任务包含多个时刻,每一时刻均对应一个输入和输出.循环神经网络的数据流也与前馈神经网络不同,隐层中存在流向自身的反馈闭环实现记忆功能,具体结构如图 3 所示.

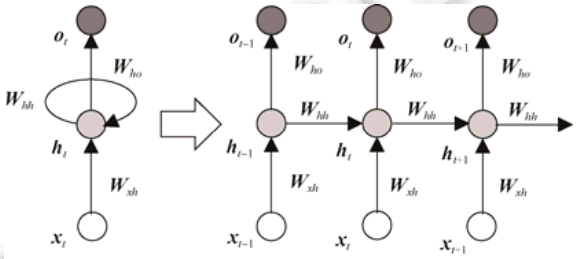


Fig.3 Architecture of recurrent neural network

图 3 循环神经网络结构

相应的计算公式为

$$\boldsymbol{y}_t=\boldsymbol{W}_{xh}\boldsymbol{x}_t+\boldsymbol{W}_{hh}\boldsymbol{h}_{t-1}+\boldsymbol{b}_h \tag{14}$$

$$\boldsymbol{h}_t=\sigma(\boldsymbol{y}_t) \tag{15}$$

$$o_t=\sigma(\boldsymbol{W}_{ho}\boldsymbol{h}_t+\boldsymbol{b}_o) \tag{16}$$

$\boldsymbol{x}_t$、$\boldsymbol{h}_t$、$\boldsymbol{o}_t$ 分别表示输入层、隐层、输出层输出向量,$\boldsymbol{W}_{xh}$、$\boldsymbol{W}_{hh}$、$\boldsymbol{W}_{h.}$分别表示输入层连接隐层、隐层连接隐层、隐层连接输出层的权重矩阵.

由于隐层输出具有随时间序列自我更新的特性,梯度下降采用时间驱动的反向传播算法(back propagation through time,简称 BPTT)[11]计算梯度.假设损失值只与最后时刻输出相关,隐层输出梯度的迭代公式为

$$\frac{\partial e}{\partial \boldsymbol{h}_{t-1}}=\boldsymbol{W}_{hh}^T\left[\sigma'(\boldsymbol{y}_t)\times\frac{\partial e}{\partial \boldsymbol{h}_t}\right] \tag{17}$$

公式与全连接网络的迭代公式(3)类似,区别在于迭代更新的序号由层数变成时刻,迭代所需的权重参数由不同的层间权重变成唯一的 $\boldsymbol{W}_{hh}$.隐层输出梯度中的元素值,即隐层输出节点各时刻的偏导数计算公式为

$$\left[\frac{\partial e}{\partial \boldsymbol{h}_{t-1}}\right]_i=\sum_{j}^{n}\left[\sigma'(\boldsymbol{y}_t)\right]_j\cdot\left[\frac{\partial e}{\partial \boldsymbol{h}_t}\right]_j\cdot\left[\boldsymbol{W}_{hh}\right]_{i,j} \tag{18}$$

n 表示隐层输出元素个数.由各时刻隐层输出梯度,根据多元复合函数链式求导法,其计算连接权重的公式为

$$\frac{\partial e}{\partial \boldsymbol{W}_{xh}}=\sum_{t}^{T}\boldsymbol{x}_t^T\left[\sigma'(\boldsymbol{y}_t)\times\frac{\partial e}{\partial \boldsymbol{h}_t}\right] \tag{19}$$

$$\frac{\partial e}{\partial \boldsymbol{W}_{hh}}=\sum_{t}^{T}\boldsymbol{h}_{t-1}^T\left[\sigma'(\boldsymbol{y}_t)\times\frac{\partial e}{\partial \boldsymbol{h}_t}\right] \tag{20}$$

T 表示输入序列长度,即共存在 T 个隐层输出梯度.梯度下降法根据梯度来更新参数.

1.3 梯度不稳定现象成因

分析全连接神经网络和卷积神经网络两种前馈神经网络以及循环神经网络的梯度反向传播计算过程,公式(3)、公式(10)和公式(17)指出隐层输出梯度可由迭代计算得到,具体元素值即偏导数的计算如公式(4)、公式(11)和公式(18)所示.三者的梯度反向传播过程具有相似性,均可大致分为3步:第1步,隐层节点输出的偏导数与激活函数导数值相乘;第2步,将乘积结果与权重矩阵或卷积核中的元素再次相乘;第3步,根据层间的连接关系对乘积结果求和,结果为前一层节点输出值的偏导数.这3步计算影响梯度迭代的稳定性,本文将逐步分析.

第1步将隐层节点输出的偏导数乘以激活函数导数.以全连接隐层为例,该过程具体为

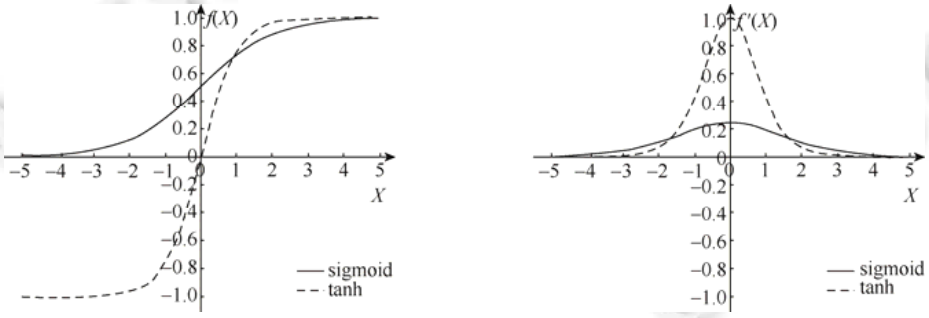
$$f_j=\left[\sigma'(\boldsymbol{y}_l)\right]_j\cdot\left[\frac{\partial e}{\partial \boldsymbol{x}_l}\right]_j \tag{21}$$

在循环神经网络中,则需将公式(21)中的 $\boldsymbol{x}_l$ 和 $\boldsymbol{y}_l$ 替换成 $\boldsymbol{h}_t$ 和 $\boldsymbol{y}_t$;在卷积层中,则需将公式(21)中的 $\boldsymbol{x}_l$ 和 $\boldsymbol{y}_l$ 替换成 $\boldsymbol{X}_l^{(n)}$和 $\boldsymbol{Y}_l^{(n)}$,向量角标替换成矩阵角标.虽然参与运算的变量形式不同,但乘法运算是一致的.激活函数导数作为乘积项,直接影响计算结果.常见的饱和激活函数主要包括 sigmoid 函数和 tanh 函数[12],二者的函数表达式为

$$\text{sigmoid}(x)=\frac{1}{1+\mathrm{e}^{-x}} \tag{22}$$

$$\tanh(x)=\frac{\mathrm{e}^{x}-\mathrm{e}^{-x}}{\mathrm{e}^{x}+\mathrm{e}^{-x}} \tag{23}$$

函数及导数图像如图4所示.sigmoid函数的导数区间为(0,0.25],tanh函数的导数区间为(0,1],且绝大部分区间的导数值接近 0[13],所以,输出梯度中的元素 $\left[\frac{\partial e}{\partial \boldsymbol{x}_l}\right]_j$ 与激活函数导数$[\sigma'(\boldsymbol{y}_l)]_j$ 相乘后的结果 f_j 必然小于输出梯度元素 $\left[\frac{\partial e}{\partial \boldsymbol{x}_l}\right]_j$.



(a) 函数图像 (b) 导数图像

Fig.4 Function & Derivative graph of sigmoid & tanh activation

图4 sigmoid 函数、tanh 函数及导数图像

第2步将隐层输出梯度与激活函数导数相乘结果中的元素与权重矩阵或卷积核中的元素再次相乘.在全连接层中,该过程具体为

$$g_{i,j}=f_j\cdot[\boldsymbol{W}_l]_{i,j} \tag{24}$$

循环神经网络和卷积层的该计算过程需将公式(24)中的权重参数 $\boldsymbol{W}_l$ 替换成相应的 $\boldsymbol{W}_{hh}$ 和 $\boldsymbol{K}_l^{(m,n)}$.但其乘法运算仍保持一致.若权重或卷积核的元素小于1,则乘积结果 $g_{i,j}$ 将小于被乘数 f_j;若权重或卷积核的元素大于1,则乘积结果 $g_{i,j}$ 将大于被乘数 f_j.

第3步将之前乘积的结果求和,得到迭代的最终结果.全连接层中的该过程为

$$\left[\frac{\partial e}{\partial \boldsymbol{x}_{l-1}}\right]_i=\sum_j^{n_l} g_{i,j} \tag{25}$$

循环神经网络的该过程与全连接层类似,其求和项数也等于隐层节点数 n,卷积层在该求和过程中的求和项数则取决于卷积核的大小 k 和输出矩阵的数量 n_l.假设所有乘积结果 $g_{i,j}$ 相同,项数越多,则求和的结果 $\left[\frac{\partial e}{\partial \boldsymbol{x}_{l-1}}\right]_i$ 越大,项数越小,则求和的结果 $\left[\frac{\partial e}{\partial \boldsymbol{x}_{l-1}}\right]_i$ 越小.

由此可见,激活函数导数值、全连接层中的权重值与节点数量和卷积层中的卷积核数值、大小与输出矩阵数量共同决定着相连两层梯度的大小关系.若无法平衡迭代过程中的 3 步运算,每次迭代的梯度均比原来梯度偏大或偏小,随着网络层数或时间序列的增加,迭代次数也随之增加,隐层输出梯度作为迭代结果将逐层(时刻)递增或递减,反向传播后期的隐层输出梯度将极大或极小,发生严重的梯度爆炸或梯度衰减的不稳定现象.

进一步分析影响梯度迭代结果的变量,第 1 步中的激活函数导数值取决于激活函数输入值和函数表达式,而函数输入值又取决于前一层输出和隐层参数;第 2 步中的权重数值和卷积核数值作为隐层参数在初始化后随训练不断变化;第 3 步中求和的各项由第 2 步权重或卷积核参数与第 1 步结果相乘得到,所以隐层参数是影响梯度迭代的关键变量,直接或间接影响相关变量的计算结果.但是,训练过程中的参数是不确定的,这将导致相关变量难以估量,其他因素的设计缺乏指导,难以平衡上述 3 步运算,从而引发梯度不稳定现象.

综合上述分析,在模型设计阶段,激活函数表达式、参数初始值、隐层节点数量、卷积核的大小和输出矩阵数量等作为可人为控制的因素,直接或间接影响梯度反向传播的稳定程度.若不能合理设计这些超参数,将发生严重的梯度不稳定现象.

1.4 梯度不稳定现象影响

1.4.1 导致深度神经网络深度特性失效

深度神经网络的深度特性在前馈神经网络和循环神经网络中的表示形式不同.诸如全连接神经网络和卷积神经网络的前馈神经网络的深度体现在隐层数量上,堆叠的隐层越多,网络深度越深.其深度特性是指利用多隐层结构多次对特征融合重构,从而挖掘数据中更深层的内在规律,达到更高的分类或预测准确率.循环神经网络的深度体现在输入时刻的数量上,序列长度越长,网络深度越深.其深度特性是指通过对从远到近多个时刻的信息的融合提取,挖掘长序列信息之间的内在联系,即长期依赖能力.循环神经网络能有效学习的序列信息越长,长期依赖能力就越强.

反向传播过程中各变量的梯度信息可反映变量变化后网络输出结果随之变化的幅度.由梯度的定义可知,函数变量的梯度大小表示函数在该点的最大变化速率.进一步解释,若某变量梯度极小,当该变量变化时,函数值几乎不变,若变量梯度极大,即使变量在梯度方向上细微变化,函数值也将大幅度改变.因此,当深度神经网络发生梯度不稳定现象时,梯度呈现递减或递增的规律性差异.当梯度不同的变量发生变化时,输出结果将产生不同程度的变化.由第 1.3 节可知,神经网络深度越深,这种差异越明显,对不同类型网络的深度特性产生严重影响,具体解释如下.

(1) 在深度前馈神经网络中发生梯度不稳定现象,隐层输出变量的梯度将逐层递减或递增.① 若发生梯度衰减现象,标号较小的底层输出梯度远小于高层输出梯度,甚至接近于 0.这部分隐层输出结果的差异性对最终输出结果几乎没有影响,底层利用随机参数融合重构的特征对学习几乎不起作用,甚至起负作用,掩盖了数据内部规律.② 若发生梯度爆炸现象,底层输出梯度远大于高层输出梯度,隐层输出细微的差别都会导致最终结果明显的不同,所有特征都被过度强化,挖掘大量次要特征作为分类预测依据,发生过拟合现象.所以在梯度不稳定的前馈神经网络中堆叠大量隐层,往往适得其反,无法充分发挥其深度特性.

(2) 在深度循环神经网络中发生梯度不稳定现象,隐层输出变量的梯度将逐时刻地递减或递增.① 若发生梯度衰减现象,前期时刻隐层输出梯度远小于后期隐层输出梯度,融合了前期输入数据的隐层输出结果对模型输出几乎没有影响,即使此时的隐层输出包含了远距离信息,也无法从中学习到有价值的规律,输出结果更多取

决于后期输入数据.② 若发生梯度爆炸现象,前期时刻隐层输出梯度大于后期隐层输出梯度,模型更侧重于挖掘前期输入数据中的规律,而忽略后期输入数据.循环神经网络因其结构的特殊性极易发生梯度不稳定现象,所以必须限制输入序列长度,难以充分发挥其长期依赖的深度特性.

由于前馈神经网络和循环神经网络在该问题上具有相似性,本文仅进行循环神经网络的验证实验.采用序列长度为 2 的单隐层循环神经网络拟合逻辑“与”运算,具体结构如图 5 所示.激活函数为 sigmoid,隐层输出 h_1 作为最终结果,固定参数 w_{xh}=1,b_h=–2.根据公式(18),令 w_{hh} 的值为 2 和 8,循环神经网络将发生梯度衰减和梯度爆炸现象.设计样本输入和标签为{[(1,1),1],[(0,1),1],[(1,0),1]},输入为(1,1)和(0,1)的样本的前期输入 x_0 不同,而(1,1)和(1,0)样本的后期输入 x_1 不同,根据期望的长期依赖能力和“与”逻辑运算定义,x_0 和 x_1 是等价的,(0,1)和(1,0)的输出结果应该相同.但实际上,因为长期依赖的差异,二者输出必然不同,通过比较二者输出与(1,1)输出之差,验证在梯度不稳定条件下,不同位置的数据影响输出结果的程度不同,实验结果见表 1.

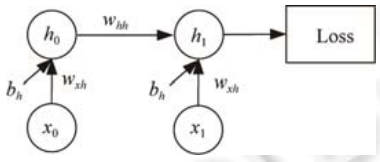


Fig.5 Architecture of recurrent neural network experiment

图 5 循环神经网络实验结构

Table 1 Hidden gradients & loss values from recurrent neural network trained by gradient descent once

表 1 循环神经网络梯度下降 1 次中的隐层梯度和输出

样本输入	W_{hh}=2 梯度衰减			W_{hh}=8 梯度爆炸		
	$\frac{\partial e}{\partial h_0}$	$\frac{\partial e}{\partial h_1}$	输出	$\frac{\partial e}{\partial h_0}$	$\frac{\partial e}{\partial h_1}$	输出
1,1	–0.29	–0.61	0.39	–0.35	–0.24	0.76
0,1	–0.30	–0.68	0.32	–1.02	–0.51	0.49
1,0	–0.25	–0.81	0.19	–0.92	–0.46	0.54

在梯度衰减模型中,x_0 不同的样本(1,1)和(0,1)输出结果相差 0.07,而 x_1 不同的样本(1,1)和(1,0)结果相差 0.20,说明前期输入的 x_0 不同对输出结果几乎没有影响,但 x_1 不同却使得最终结果呈现明显差异,模型侧重学习 x_1 中蕴含的信息;在梯度爆炸模型中,(1,1)和(0,1)的结果差值为 0.27,大于(1,1)和(1,0)的结果差值 0.22,与梯度衰减模型相反,不同的 x_0 导致输出结果的差异更明显,模型更侧重学习 x_0 中蕴含的信息.所以在梯度不稳定条件下,不同时刻的输入数据对最终输出结果的作用程度不同,盲目增加序列长度,并不能提高其长期依赖能力,反而会导致学习效果下降.

1.4.2 导致前馈神经网络收敛速度缓慢

神经网络的收敛目标是得到一组最优模型参数使损失函数值最小,由于模型参数众多且表达式复杂,需利用最优化算法以迭代训练的方式求解.所以收敛过程是指通过训练调整各参数趋近最优点(极值点),使损失函数值逐渐减小到最小值的过程.单位次数训练后损失函数值减小的幅度越大,收敛速度越快.

梯度下降法是一种典型的无约束最优化算法.根据梯度的定义,梯度向量(的模)大小反映梯度方向变化速度的大小.梯度下降法沿负梯度方向调整参数,而调整距离与梯度的大小有关,即:

$$|\Delta\boldsymbol{\omega}|=\eta\cdot\left|\frac{\partial e}{\partial\boldsymbol{\omega}}\right| \tag{26}$$

当学习率一定时,梯度越大,调整的距离越大.假设在调整前后所跨越的局部变量空间中损失函数的梯度固定不变,则函数变化量等于变化率乘以距离,即:

$$\Delta e=|\Delta\boldsymbol{\omega}|\cdot\left|\frac{\partial e}{\partial\boldsymbol{\omega}}\right| \tag{27}$$

所以,当梯度较小时,函数变化速率和调整距离都比较小,损失函数值小幅度变化,收敛速度缓慢.相反,当梯度较大时,函数变化速率和调整距离都比较大,函数值大幅度减小,收敛速度极快.因此,在多数情况下,初始条件

下的变量梯度较大,梯度下降使其快速收敛.当变量接近最优点(极值点)时的梯度较小,微调变量使其逐渐逼近最优解.所以梯度下降法能够有效收敛函数,并被广泛用于线性回归等机器学习模型的训练任务中.

但是,对于极复杂模型的训练,梯度较小或较大的梯度下降法,收敛速度都有可能缓慢[14].复杂模型的损失函数表示式十分复杂,难以估计变量的梯度变化规律,采用梯度下降法训练很可能出现以下情况:若当前值梯度较小,则与之对应的近距离调整所跨越的变量空间较小,由于梯度连续变化,极小局部变量空间中的梯度差距可忽略,所以不会影响其收敛效果,收敛速度依然缓慢;若当前值梯度较大,则与之对应的远距离调整所跨越的变量空间较大,如果梯度发生明显变化,无法预料该范围内的函数值变化趋势,存在沿负梯度方向函数值先减后增的可能,越过该方向上的极小值点,落点处的损失函数值比起点处略微减少,甚至更大,导致收敛速度缓慢,甚至无法收敛.

梯度不稳定现象导致诸如多隐层全连接神经网络和多隐层卷积神经网络的深度前馈神经网络整体参数梯度较小或较大.深度前馈神经网络顶层输出梯度与损失函数值相关.训练初期,随机初始化的参数计算得到的损失值相近,所以顶层输出梯度大小基本固定.又因为在多隐层结构中发生梯度衰减或梯度爆炸现象时,各隐层输出梯度逐层递减或递增,所以在梯度稳定、梯度衰减、梯度爆炸 3 种条件下比较顶层以下的各隐层输出梯度大小的结果为

$$\left|\frac{\partial e}{\partial \boldsymbol{x}_l}\right|_{\text{梯度衰减}} < \left|\frac{\partial e}{\partial \boldsymbol{x}_l}\right|_{\text{梯度稳定}} < \left|\frac{\partial e}{\partial \boldsymbol{x}_l}\right|_{\text{梯度爆炸}} \tag{28}$$

而且越靠近底层,梯度大小的差异越明显.由第 1.1 节中公式(5)、公式(6)、公式(12)和公式(13)可知,各隐层输出梯度结合前向传播输入和激活函数导数计算得到参数梯度,而各隐层输入和激活函数导数的数值范围是相近的,意味着隐层参数梯度大小主要取决于隐层输出梯度大小,同样受梯度不稳定现象影响呈现明显的差异,底层的参数梯度在梯度衰减条件下极小,在梯度爆炸条件下极大.所有参数梯度汇总后的梯度向量的大小呈现同样的规律,即:

$$\left|\frac{\partial e}{\partial \boldsymbol{\omega}}\right|_{\text{梯度衰减}} < \left|\frac{\partial e}{\partial \boldsymbol{\omega}}\right|_{\text{梯度稳定}} < \left|\frac{\partial e}{\partial \boldsymbol{\omega}}\right|_{\text{梯度爆炸}} \tag{29}$$

这种差异性不会随着训练数据的改变而消失,而是长期存在于训练过程中.

结合上述分析,网络参数作为梯度下降变量,由梯度不稳定引发参数梯度整体较小或较大的问题,易导致收敛速度缓慢.在学习率一定的条件下,若发生梯度衰减现象,参数梯度较小,过于精细的参数调整会使模型收敛缓慢,初始损失值减小到最小值需要大量的训练,训练效率低下;若发生梯度爆炸现象,参数梯度较大,大幅度调整容易使得新参数越过该方向上的最优解,导致收敛缓慢甚至无法收敛.如果多次出现该现象,则需重新调整学习率,否则,严重影响模型的训练效果.

本文将结合实验验证上述内容.采用单隐层且每层单节点的全连接网络拟合映射关系,结构如图 6 所示.隐层和输出层的参数包括连接权重 w_1,w_2 和偏置 b_1,b_2,采用 tanh 激活函数,由平方损失函数计算损失值,样本输入和标签为(0,0).由公式(4)可知,w_2 决定输出层和隐层之间的梯度(偏导数)传播的稳定性,固定参数 w_1=1,令 w_2 的值为 0.5、1 和 2,对应梯度衰减、梯度稳定和梯度爆炸 3 种情况,构造 3 组以 b_1 和 b_2 为自变量的损失函数.在保证各组 b_2 方向上偏导数相同的条件下选取初始点.以学习速率为 0.5 的梯度下降法训练 1 次,观察训练前后损失函数值的变化,比较收敛速度.结果如图 7 所示.

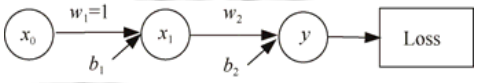
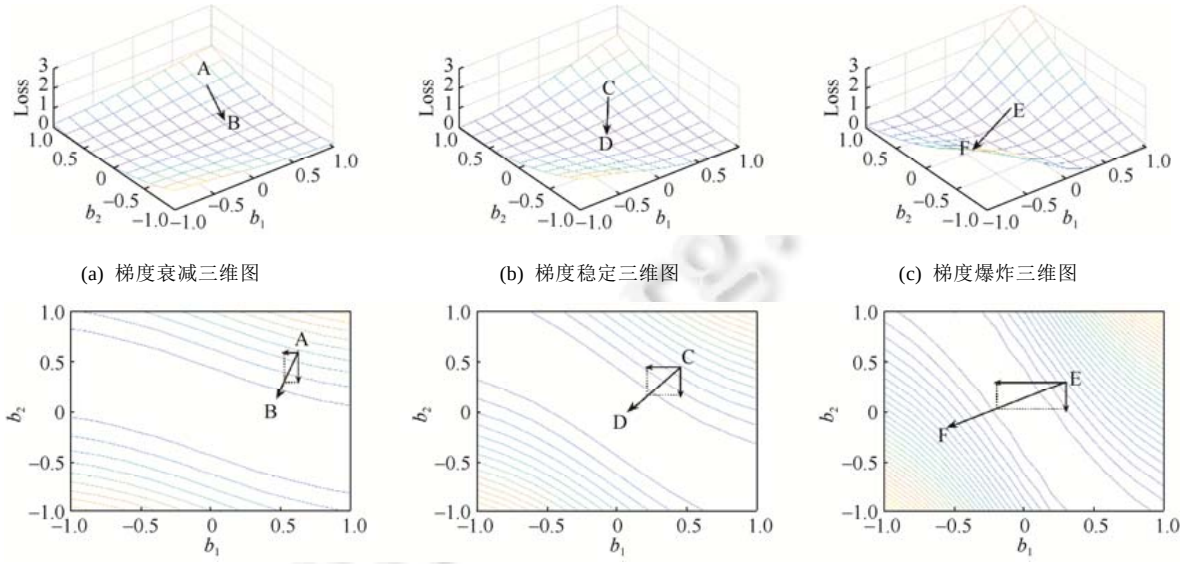


Fig.6 Architecture of feed forward neural network experiment

图 6 前馈神经网络实验结构

由图 7(d)~图 7(f)可知,在 b_2 方向上梯度(偏导数)相同的条件下,梯度衰减模型中 A 点梯度的模小于梯度稳

定模型中 C 点,梯度爆炸模型中 E 点梯度的模大于梯度稳定模型中 C 点.比较训练前后损失值变化情况,在梯度衰减模型中,(b_1,b_2)经过训练从 A 点变为 B 点,损失值减少 0.31.在梯度稳定模型中,(b_1,b_2)从 C 点移动到 D 点,损失值减少 0.38,且 D 点的损失值接近最小值 0.在梯度爆炸模型中,(b_1,b_2)从 E 点移动到 F 点,损失值没有减小,F 点损失值比 E 点大 0.19.观察图 7(c)和图 7(f),损失值不减反增是因为(b_1,b_2)移动越过该方向上的极小值点,落在损失值更大的 F 点.所以在上述条件下,发生梯度衰减和梯度爆炸现象的点梯度下降的收敛速度缓慢甚至无法收敛.



(a) 梯度衰减三维图 (b) 梯度稳定三维图 (c) 梯度爆炸三维图

(d) 梯度衰减等高线图 (e) 梯度稳定等高线图 (f) 梯度爆炸等高线图

Fig.7 Process of feed forward neural network trained by gradient descent

图 7 前馈神经网络梯度下降训练过程

2 梯度不稳定现象的解决策略

梯度不稳定现象严重影响到深度神经网络发挥深度优势和训练收敛效率.因此,有大量研究分析或尝试解决该问题.总的来说,有 3 种改善梯度不稳定问题的策略:一是改进训练方法,针对深层网络设计专门训练策略;二是优化节点运算,针对反向传播算法中影响梯度迭代的关键因素进行优化;三是调整网络结构,通过改变梯度迭代的形式,解决梯度在迭代过程中不稳定的问题.

2.1 改进训练方法

当我们使用反向误差传播算法训练深度神经网络时,由于梯度不稳定导致某些层的参数得不到有效调整,从而无法有针对性地提取样本特征.因此希望能够找到一组可以有效提取样本特征的初始参数,减少梯度不稳定现象对训练过程的影响.Hinton 等人[15]提出预训练(pre-training)参数法,利用无监督特征学习方法,解决梯度不稳定的问题.预训练参数法最初被用于深度置信网络(deep belief network,简称 DBN)[15−17].如图 8 所示.DBN 网络作为生成性网络,由一系列受限波尔兹曼机(restricted Boltzmann machine,简称 RBM)[18,19]单元串联堆叠组成.RBM 是一种对称全连接无向神经网络,由输入特征的可见层和重构特征的隐层构成,利用对比散度(contrastive divergence)[20]准则,以输入特征与重构特征联合概率分布最大为目标进行无监督训练,使得网络以最小代价重构输入特征,降低数据间的依赖性.经过多次融合重构,将原始数据抽象成高度概念化、相互独立的特征.对网络参数预训练后,针对具体的分类任务强化所需特征,使用梯度下降算法对参数进行微调(fine-tuning).

预训练参数法能够缓解梯度不稳定现象的原因在于其降低了参数训练对梯度下降法的依赖性.从本质上

讲,梯度不稳定现象并没有消失,而是降低了它对模型训练结果的影响.利用无监督训练得到的参数具备了相当的提取特征能力,即使底层参数由于梯度衰减现象而无法有针对性地提取特征,仍然能够利用RBM网络的参数对输入特征进行抽象和重构,经过多层网络计算后得到的特征具有抽象性高、概括性强等特点,充分体现了多层网络的深度特性.同时,无监督学习的特征有利于避免过拟合现象的发生.

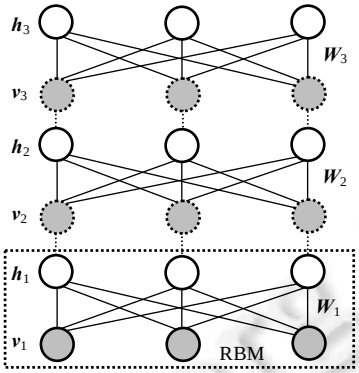


Fig.8 Deep belief network

图 8 深度置信网络

虽然预训练的深度置信网络在某些领域被证明具有良好的效果[21,22],但仍有以下几方面值得思考.

(1) 预训练参数作为无监督学习方法,由于缺少样本标签,针对目标特征的模型选择和设计主要依靠设计者的经验,实际效果难以保证.

(2) 微调需对预训练的参数进一步调整,所以采用高标准的无监督训练对整体训练的意义不大.如何确定预训练目标,平衡预训练与微调二者关系成为又一难点.

(3) 无监督训练所需计算资源十分巨大,特别是针对复杂模型和大规模数据,训练所需时间极长,速度和效率成为制约此类方法发展的关键因素.

有大量工作利用无监督预训练的方法初始化参数,以缓解梯度不稳定现象.例如,Bengio 等人[16]利用自动编码器(auto-encoder)初始化参数,并结合有监督学习进一步调整参数,加强对概率分布不敏感的特征提取;Ranzato 等人提出稀疏对称编码器(sparse encoding symmetric machine,简称 SESM)[23]作为参数无监督学习算法,使得神经网络能够学习稀疏化的特征.

2.2 优化节点运算

由第 1.3 节可知,在反向传播算法计算深度神经网络模型参数梯度的迭代过程中,导致梯度不稳定现象的主要因素包括:隐层节点数量、参数初始值以及激活函数表达式.针对上述因素,学者们进行了相应改进,使反向传播算法的迭代过程更加稳定.

2.2.1 参数初始化策略

神经网络参数因为随训练不断调整,所以参数初始化过程常常被忽略.事实上,参数初始值持续影响着训练过程.学者们从实践中发现多节点隐层的输出和梯度均大致服从均值为 0 的正态分布,因此方差大小可反映数据的大小特征,通过比较各隐层梯度方差,判断反向传播中隐层梯度的稳定程度.假设存在网络参数与输入变量独立分布均值为0,并且激活函数导数值恒等于1的多隐层全连接网络,计算前向传播过程中节点输出与反向传播过程中梯度的方差公式为[24]

$$\mathrm{Var}[\boldsymbol{x}_l]=\mathrm{Var}[\boldsymbol{x}_0]\prod_{i=1}^{l} n_{i-1}\cdot\mathrm{Var}[\boldsymbol{W}_i] \tag{30}$$

$$\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_l}\right]=\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_m}\right]\prod_{i=l+1}^{m} n_i\cdot\mathrm{Var}[\boldsymbol{W}_i] \tag{31}$$

早期学者将参数初始值任意随机设定在 0 的附近[25],增加了训练的难度和不确定性.之后使用均匀分布 $U\left(-\frac{1}{\sqrt{n_l}},\frac{1}{\sqrt{n_l}}\right)$ 初始化网络权重 $\boldsymbol{W}_l$,则 $\boldsymbol{W}_l$ 的方差为 $\mathrm{Var}[\boldsymbol{W}_l]=\frac{1}{3n_l}$,如果每层网络的节点数相等,则参数 $\boldsymbol{W}_l$ 对误差求导结果的方差为

$$\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{W}_l}\right]=\mathrm{Var}\left[\boldsymbol{x}_{l-1}\right]\cdot\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_l}\right]=\frac{\mathrm{Var}\left[\boldsymbol{x}_0\right]\cdot\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_m}\right]}{3^{m-2}} \tag{32}$$

可见每层网络权重矩阵梯度的方差随总层数的增加而呈幂指数减小,导致参数梯度过小,梯度下降更新迟缓.Glorot 等人认为,深层网络权重梯度的方差应与网络层数无关,即满足条件:

$$n_{l-1}\cdot\mathrm{Var}[\boldsymbol{W}_l]=1 \tag{33}$$

$$n_l\cdot\mathrm{Var}[\boldsymbol{W}_l]=1 \tag{34}$$

据此提出了规范化初始化(normalized initialization)方法,也被称为 Xavier 初始化方法[17,24],公式为

$$\boldsymbol{W}_l\sim U\left(-\frac{\sqrt{6}}{\sqrt{n_{l-1}+n_l}},\frac{\sqrt{6}}{\sqrt{n_{l-1}+n_l}}\right) \tag{35}$$

通过 Xavier 方法初始化网络参数,前向传播中每层节点输入数据的概率分布基本相同,反向传递中每层节点梯度值的概率分布也基本相同,缓解了梯度不稳定现象、同时使得权重矩阵梯度方差与网络层数无关,仅与输入样本方差和输出层梯度方差有关,避免了在深层网络中参数梯度趋近于 0 导致的参数更新效果不佳和收敛速度慢的问题.上述推导完全建立在激活函数导数为 1 的假设上,若采用不同的非线性激活函数,该方法的实际效果将与预期效果存在一定差距,并且分类任务越复杂,实际效果越好.针对该现象,He 等人[26]以 ReLU 为激活函数(y=max(0,x)),针对其不对称的特性,得出了具体的方差公式为

$$\mathrm{Var}\left[\boldsymbol{x}_l\right]=\mathrm{Var}\left[\boldsymbol{x}_0\right]\prod_{i=1}^{l}\frac{1}{2}n_{i-1}\cdot\mathrm{Var}\left[\boldsymbol{W}_i\right] \tag{36}$$

$$\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_l}\right]=\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_m}\right]\prod_{i=l+1}^{m}\frac{1}{2}n_i\cdot\mathrm{Var}\left[\boldsymbol{W}_i\right] \tag{37}$$

推导出不同的参数满足条件:

$$n_{l-1}\cdot\mathrm{Var}[\boldsymbol{W}_l]=2 \tag{38}$$

$$n_l\cdot\mathrm{Var}[\boldsymbol{W}_l]=2 \tag{39}$$

并且只需要满足上述其中一个条件即可,分别采用正态分布 $N\left(0,\frac{2}{n_{l-1}}\right)$ 和 $N\left(0,\frac{2}{n}\right)$ 初始化权重矩阵,对应的 $\boldsymbol{W}_l$ 梯度方差为

$$\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{W}_l}\right]=\frac{n_m}{n_l}\mathrm{Var}\left[\boldsymbol{x}_0\right]\cdot\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_m}\right] \tag{40}$$

$$\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{W}_l}\right]=\frac{n_0}{n_l}\mathrm{Var}\left[\boldsymbol{x}_0\right]\cdot\mathrm{Var}\left[\frac{\partial e}{\partial \boldsymbol{x}_m}\right] \tag{41}$$

仍能保证权重矩阵的梯度方差与网络层数无关.实验发现,针对 ReLU 的初始化方法在对应的深层卷积网络训练中确实更好地缓解了梯度衰减现象,有效提高了模型的收敛速度,这些优势在网络参数较少的模型中更加突出.

综上所述,权重参数初始化策略主要采用具有统计规律的均匀分布或正态分布,依据前后两层节点数量控制参数范围,尽可能平衡梯度迭代过程中权重(见公式(24))和节点数目(见公式(25))对稳定性的影响.然而,上述推导均建立在一定的假设条件上,在实际网络训练中,隐层输出梯度与激活函数导数的分布规律难以估计,梯度迭代计算的稳定性仍然难以保证.但是,随着对网络训练中数据分布规律的进一步认识,可提出更具针对性的参数初始化策略,从而平衡影响梯度迭代计算的各方面因素.

2.2.2 改进激活函数

神经网络利用激活函数增加非线性因素,从而提高模型的表达能力.研究发现,激活函数不仅在特征提取中起重要作用,它的导数还决定反向梯度传播的能力[24,27].2001 年生物学家拟合了更精确的脑神经激活模型,发现其具有不对称和稀疏分布等性质[28],为人工神经网络科学家设计激活函数提供了新思路.

目前的研究结果表明,相比于传统饱和非线性函数,不饱和非线性函数更适用于深度神经网络.例如纠正线性单元(rectified linear unit,简称 ReLU)[29],其函数表达式为

$$y=\begin{cases}x, & x\geqslant 0\\ 0, & x<0\end{cases} \tag{42}$$

2000 年 Hahnloser 等人[30]首次使用 max(0,x),作为神经元的激活函数.2010 年 Nair 等人[29]在改进 RBM 隐层单元时正式提出 ReLU.2012 年 Krizhevsky 等人[31]成功地将 ReLU 应用在深层卷积神经网络中,并大幅度提升了图像识别的准确率.Softplus[32]是另一种常见的不饱和激活函数,具体表达式为

$$y=\log(1+\mathrm{e}^{x}) \tag{43}$$

Softplus 的函数图像与 ReLU 类似,可看作 ReLU 的近似光滑表示,如图 9 所示.

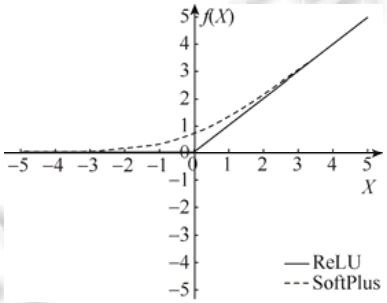


Fig.9 Unsaturated nonlinear activation

图 9 不饱和非线性激活函数

根据第 1.3 节所讨论的激活函数导数对梯度不稳定现象的影响(见公式(21)),这类激活函数与传统激活函数明显不同,不但激活区间宽广,占据了一半的输入空间,而且激活区间导数值等于或约等于 1,提高了反向传播过程中梯度的稳定程度,而饱和区间导数值等于或约等于 0,使得反向传播过程直接忽略了饱和节点,停止传递.此类激活函数有效地缓解了梯度不稳定现象的发生.同时,ReLU 激活函数单边强制置 0 的策略更符合脑神经激活模型单边抑制、宽阔兴奋边界、稀疏分布的特性,有助于从稠密数据中解耦特征,增强特征提取的鲁棒性,减少参数间的依存关系,同时满足非线性的要求.激活区间采用恒等映射,在保证梯度有效传递的同时通过多层复合运算增强了模型的表现力.每层网络中激活的节点根据待学习的特征确定,避免出现参数过少或过多导致的欠拟合或过拟合现象.ReLU 简单的运算有利于提高训练速度.但是,使用 ReLU 的神经单元一旦进入饱和状态,很难被再次激活,对应参数也随之固化.针对该问题,Andrew 等人提出了 leaky ReLU[33],表达式为

$$y=\begin{cases}x, & x\geqslant 0\\ 0.01x, & x<0\end{cases} \tag{44}$$

在激活区间仍采用恒等映射作为激活方式,而在饱和区间以极小非零斜率的线性映射进行反向抑制.该方法虽然牺牲了 ReLU 稀疏梯度的特性,但提高了训练过程的鲁棒性,而且饱和区间的导数值极小,对梯度反向传播几乎没有影响.类似的改进还有 PReLU[26],表达式为

$$y=\begin{cases}x, & x\geqslant 0\\ ax, & x<0\end{cases} \tag{45}$$

其中,a 是随训练动态学习的参数.将每个激活函数的非线性程度作为网络参数,由数据和模型来确定.实验发现,随着网络的前向传播,激活函数的非线性程度逐渐增加,即底层网络趋向于保留更多信息,顶层网络趋向于分辨有用信息.

ReLU 及衍生函数的提出极大地优化了深度神经网络,特别是在梯度稳定性方面,在保证非线性能力的前提下,使一半输入空间的导数恒为 1,有效缓解了梯度迭代中激活函数导数(见公式(21))对梯度稳定的负面影响,确定的导数值进一步方便了学者对神经网络中变量与梯度的分布规律的分析,从而有针对性地优化初始参数与输入数据的数值分布.

除梯度稳定性外,这一系列函数的提出也为未来激活函数的设计思路指明了方向,具体可总结为以下 3 点.

(1) 启发式设计.ReLU 的设计思路来自对人脑神经元激活模型的认识.在学者对神经网络隐层规律和特征含义充分认识之前,利用启发式思维,借鉴其他领域相似模型或函数,改进激活函数设计,并通过实验证明其正确性是非常有效的方法.

(2) 非线性拟合.ReLU 相比于传统的对称激活函数表达式较为简单,但由其构造的深度神经网络能够更好地拟合样本数据与标签之间的复杂映射关系.所以,提高由激活函数提供的非线性能力,丰富网络函数空间,将成为主要研究内容之一.

(3) 简单、易计算.无论是前向传播还是反向传播,ReLU 系列激活函数的计算量相比传统激活函数少很多,这对神经网络的高效训练具有重要意义.

2.2.3 归一化处理

尽管已经提出针对传统节点运算的改进方法,但是由于网络隐层间传输数据或梯度的不确定性,无法统计其分布规律,因此难以进一步优化.针对该问题,Google 公司的 Ioffe 等人借鉴图像处理中白化(whiten)处理的方法,提出了批归一化算法(batch normalization,简称 BN)[34].在传统的隐层计算过程中,对激活函数输入数据进行归一化,强行改变激活函数输入数据的分布规律.具体运算过程为

$$\hat{\boldsymbol{y}}_l=\frac{\boldsymbol{y}_l-\mathrm{E}[\boldsymbol{y}_l]}{\sqrt{\mathrm{Var}[\boldsymbol{y}_l]}} \tag{46}$$

$$\boldsymbol{z}_l=\boldsymbol{\gamma}_l\times\hat{\boldsymbol{y}}_l+\boldsymbol{\beta}_l \tag{47}$$

其中,$\boldsymbol{y}_l$表示第 l 层激活函数的原来输入,对其归一化后得到$\hat{\boldsymbol{y}}_l$.因为归一化运算导致数值偏移,破坏了数据原来表示的特征,所以在每层归一化后引入缩放(scale)和偏移(shift)操作重构特征(见公式(47)),参数 $\boldsymbol{\gamma}_l$ 和 $\boldsymbol{\beta}_l$ 随训练动态学习.结果 $\boldsymbol{z}_l$ 代替 $\boldsymbol{y}_l$ 输入激活函数.

当隐层引入 BN 算法时,偏置向量 $\boldsymbol{b}_l$ 被归一化运算抵消,可不必设置该参数.以全连接层为例,引入 BN 算法后的前馈传播过程为

$$\boldsymbol{z}_l=BN(\boldsymbol{W}_l\boldsymbol{x}_{l-1}) \tag{48}$$

$$\boldsymbol{x}_l=\sigma(\boldsymbol{z}_l) \tag{49}$$

BN 函数表示 BN 算法的计算过程(见公式(46)和公式(47)),同时可证明缩放输入数据对 BN 函数输出结果没有影响,即:

$$BN(W_lx_{l-1})=BN(\alpha\cdot W_lx_{l-1}) \tag{50}$$

α为任意常数,该特性使得隐层更倾向于挖掘数据的分布规律和相关性,而不是数值大小.

相应的隐层反向传播计算也有所改变,具体为

$$\frac{\partial e}{\partial\boldsymbol{x}_{l-1}}=\frac{\partial\mathrm{BN}(\boldsymbol{W}_l\boldsymbol{x}_{l-1})}{\partial\boldsymbol{x}_{l-1}}\left[\sigma'(\boldsymbol{z}_l)\times\frac{\partial e}{\partial\boldsymbol{x}_l}\right] \tag{51}$$

隐层输出梯度先与激活函数导数相乘,再根据 BN 算法的计算求梯度.虽然与传统形式类似,但其梯度则相对稳定.首先,激活函数导数值取决于表达式与输入,经过 BN 算法得到的激活函数输入满足均值为$\boldsymbol{\beta}_l$、标准差为$\boldsymbol{\gamma}_l$的分布,有效避免了因为数据集中在激活区间或抑制区间而导致激活函数导数整体偏大或偏小的问题.另外,乘以 BN 函数对 $\boldsymbol{x}_{l-1}$ 的梯度比乘以传统权重更有利于梯度稳定,因为 BN 函数对 $\boldsymbol{x}_{l-1}$ 求梯度的结果同样不受缩放输入数据的影响,即:

$$\frac{\partial BN(W_lx_{l-1})}{\partial x_{l-1}}=\frac{\partial BN(\alpha\cdot W_lx_{l-1})}{\partial x_{l-1}} \tag{52}$$

权重 $\boldsymbol{W}_l$ 成倍增加或减小,结果仍保持不变.在传统模型中,由第 1.3 节公式(24)对权重影响梯度稳定的分析可知,权重的倍增或倍减会导致梯度倍增或倍减,严重影响了梯度的稳定程度.所以 BN 算法通过优化上述两个方面,很大程度地提高了反向传播过程中的梯度稳定性.

同时,BN 算法还具有利用较大学习率加快收敛速度和有效避免过拟合现象等优势.对于特殊结构的网络,如 CNN、RNN,也已经有适用于相应模型的 BN 算法[34,35].所以,BN 算法因其具有多种优点,现已成为深度神经网络的必需组件,在各种模型中发挥作用.

为了进一步发挥 BN 算法的优势,提高深度神经网络的梯度稳定性,本文认为后续研究可从以下角度展开.

(1) BN 算法反向传播过程(见公式(51))中,BN 函数对 $\boldsymbol{x}_{l-1}$ 梯度的结果 $\frac{\partial \mathrm{BN}(\boldsymbol{W}_l\boldsymbol{x}_{l-1})}{\partial \boldsymbol{x}_{l-1}}$ 代替了传统的权重 $\boldsymbol{W}_l$.虽然该梯度值不受权重大小的影响,而是取决于 $\boldsymbol{W}_l\boldsymbol{x}_{l-1}$ 的分布规律,由于分布规律具有未知性,难以确定梯度的实际范围,但是该梯度大小却直接影响反向传播的梯度稳定性.因此,解决 $\boldsymbol{W}_l\boldsymbol{x}_{l-1}$ 乘积结果分布未知的问题至关重要.在隐层采用 BN 算法时,隐层输出 $\boldsymbol{x}_{l-1}$ 的分布规律可根据激活函数表达式和 BN 算法输出分布规律进行估计,权重 $\boldsymbol{W}_l$ 将成为影响乘积结果分布规律的关键因素,这对参数初始化策略提出了更高的要求.

(2) BN 算法的输出作为激活函数的输入,与激活函数表示式共同作用于激活函数导数值,因此,BN 算法的输出与激活函数表达式密切相关.在传统模型中,由于激活函数的输入分布规律未知,无法有针对性地设计函数表达式.但是,采用 BN 算法后,通过归一化和缩放偏移操作确定了激活函数输入的分布规律,这对函数表达式设计工作具有重要价值,能够有效降低激活函数导数对梯度稳定性的不利影响.

2.3 调整网络结构

当网络层数多到一定程度后,即使运用多种优化技巧也很难保证梯度稳定,此时可以考虑通过调整网络结构,改变传统模型反向传播的迭代计算.这方面的研究可大致分为两类:一是门限策略;二是捷径连接策略.

2.3.1 门限策略

最成功的门限策略模型是长短期记忆模型(long short-term memory,简称 LSTM)[36],作为 RNN 梯度不稳定问题的解决方案,广泛应用于机器翻译、语音识别等领域.LSTM 采用门限策略优化了 RNN 的长期依赖能力[2],即 LSTM 可以从经过长时间传递后的混合信息中提取需要的内容.该方法最早在 1997 年由 Hochreiter 等人[36]提出,引入记忆单元(memory cell)和常数错误传送结构(constant error carousel,简称 CEC)保存和传递信息,使用输入门(input gate)、输出门(output gate)控制输入数据对状态的更新和状态信息的输出.之后,由 Gers 等人[37]在此基础上引入忘记门(forget gate)试图清除无用的历史信息,构成了最常见的 LSTM 模型,其结构如图 10 所示.具体计算过程可形式化表述为

$$\boldsymbol{f}_t=\sigma(\boldsymbol{W}_{xf}\boldsymbol{x}_t+\boldsymbol{W}_{hf}\boldsymbol{h}_{t-1}+\boldsymbol{b}_f) \tag{53}$$

$$\boldsymbol{i}_t=\sigma(\boldsymbol{W}_{xi}\boldsymbol{x}_t+\boldsymbol{W}_{hi}\boldsymbol{h}_{t-1}+\boldsymbol{b}_i) \tag{54}$$

$$\hat{\boldsymbol{C}}_t=\tanh(\boldsymbol{W}_{xC}\boldsymbol{x}_t+\boldsymbol{W}_{hC}\boldsymbol{h}_{t-1}+\boldsymbol{b}_C) \tag{55}$$

$$\boldsymbol{C}_t=\boldsymbol{f}_t\times\boldsymbol{C}_{t-1}+\boldsymbol{i}_t\times\hat{\boldsymbol{C}}_t \tag{56}$$

$$\boldsymbol{o}_t=\sigma(\boldsymbol{W}_{xo}\boldsymbol{x}_t+\boldsymbol{W}_{ho}\boldsymbol{h}_{t-1}+\boldsymbol{b}_o) \tag{57}$$

$$\boldsymbol{h}_t=\boldsymbol{o}_t\times\tanh(\boldsymbol{C}_t) \tag{58}$$

其中,σ表示 sigmoid 函数,x_t 表示 t 时刻的输入,$\boldsymbol{h}_t$ 表示 t 时刻的输出,$\hat{\boldsymbol{C}}_t$ 和 $\boldsymbol{C}_t$ 表示 t 时刻记忆单元待更新与更新后存储的信息,$\boldsymbol{f}_t$、$\boldsymbol{i}_t$、$\boldsymbol{o}_t$ 分别表示忘记门、输入门和输出门的计算结果,1 表示完全保留,0 表示完全舍弃.

LSTM 缓解梯度不稳定的原因在于反向传播中记忆单元梯度的迭代公式与传统表示(见公式(17))不同,具体公式为

$$\frac{\partial e}{\partial \boldsymbol{C}_t}=\tanh'(\boldsymbol{C}_t)\times\left[\boldsymbol{o}_t\times\frac{\partial e}{\partial \boldsymbol{h}_t}\right]+\boldsymbol{f}_{t+1}\times\frac{\partial e}{\partial \boldsymbol{C}_{t+1}} \tag{59}$$

t 时刻记忆单元的梯度表达式由两项组成.第 1 项中的变量包括输出门的值、tanh 函数的导数和输出节点梯度

$\frac{\partial e}{\partial \boldsymbol{h}_t}$, 由于输出节点同时作用于下一状态的多个门和输入节点,输出梯度 $\frac{\partial e}{\partial \boldsymbol{h}_t}$ 受多方面因素影响,所以第 1 项乘积结果具有较强的不确定性.第 2 项为上一时刻的记忆单元梯度值与忘记门的乘积,影响因素较少,结果具有稳定性.结合两者的特性,在简单的迭代计算中加入具有动态化的子项,既满足梯度迭代对差异性的需求,又保证了整体的稳定性.

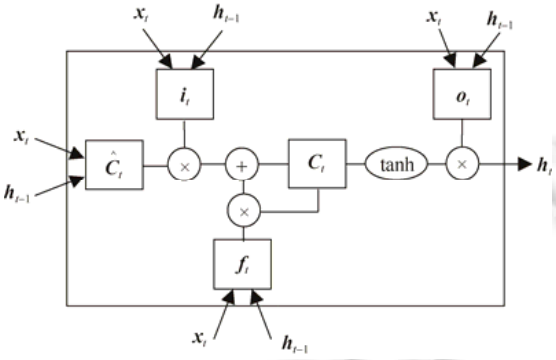


Fig.10 Architecture of long short-term memory unit

图 10 长短期记忆单元结构

LSTM 存在众多变体,Gers 等人为了更精确地利用历史信息,将记忆单元加入到门单元的输入中,提出了窥视孔连接(peephole connections)的方法[38],在丰富门限控制因素的同时增加了梯度传播路径,进一步提高了梯度的稳定性.针对 LSTM 参数过多的问题,Sak 等人[39]采用了投影变换的方法对输出进行线性降维,而 Cho 等人将 LSTM 进行简化,提出 GRU(gated recurrent unit)[40],只设置了重置门(reset gate)和更新门(update gate),实验结果表明,GRU 具有与 LSTM 相同的长期依赖能力.

受到 LSTM 的启发,Srivastava 等人将门限策略迁移到深度前馈神经网络中,提出了 highway network 模型[41,42],该模型利用门限将每层节点的输出分为两部分:其一是前一层传到本层的输入向量 $\boldsymbol{x}_{l-1}$,不进行任何处理直接通过,如同“高速公路”一样;其二是由传统隐层结构拟合的非线性函数 $H(\boldsymbol{x}_{l-1})$的计算结果,具体内容可形式化表述为

$$\boldsymbol{x}_l = \boldsymbol{t}_l \times H(\boldsymbol{x}_{t-1}) + \boldsymbol{c}_l \times \boldsymbol{x}_{l-1} \tag{60}$$

$$\boldsymbol{t}_l = \sigma(\boldsymbol{W}_{tl}\boldsymbol{x}_{l-1} + \boldsymbol{b}_{tl}) \tag{61}$$

$$\boldsymbol{c}_l = \sigma(\boldsymbol{W}_{cl}\boldsymbol{x}_{l-1} + \boldsymbol{b}_{cl}) \tag{62}$$

其中,$\boldsymbol{t}_l$ 和 $\boldsymbol{c}_l$ 作为门限的控制器,由饱和非线性函数产生.与之对应的反向传播表达式为

$$\frac{\partial e}{\partial \boldsymbol{x}_{l-1}} = \frac{\partial \boldsymbol{t}_l}{\partial \boldsymbol{x}_{l-1}}\frac{\partial e}{\partial \boldsymbol{t}_l} + \frac{\partial \boldsymbol{c}_l}{\partial \boldsymbol{x}_{l-1}}\frac{\partial e}{\partial \boldsymbol{c}_l} + \frac{\partial \boldsymbol{x}_l}{\partial \boldsymbol{x}_{l-1}}\frac{\partial e}{\partial \boldsymbol{x}_l} \tag{63}$$

由当前层梯度计算下一层梯度的表达式与 LSTM 类似,由多个项相加组成,在一定程度上降低了梯度不稳定现象发生的可能性.

门限策略虽然在 LSTM 模型和 highway network 模型中发挥着作用,提高了模型的训练效果,但仍有以下几点需要进一步考虑.

(1) 梯度的波动性.门限策略通过改变传统梯度迭代公式,引入更多的动态化因子,每次梯度迭代结果相比于原来梯度,可能更大也可能更小,虽然避免了由于梯度连续增加或减小而导致的梯度不稳定现象,但是梯度传播呈波动形式,这对模型的梯度下降训练具有怎样的影响仍需进一步研究.

(2) 参数的繁重性.每增加一个门,就需要分配相应的权重和偏置,这导致隐层参数成倍增加.特别是在 highway network 模型中,过多的参数将加重内存负担.门限计算也将消耗更多的计算资源,影响训练效率.简化模型、提高效率始终是门限策略待解决的关键问题.

(3) 解释的真实性.门限策略的设计初衷是根据需求对特征进行筛选过滤.但实际上只是对特征的每个维

度上增加比例系数,而比例系数只是由含参线性变换和激活单元产生.神经网络的特征往往隐含在数据内部,特别是在底层网络(前期时刻)中,重构特征往往需要复杂运算,所以门限策略简单的运算除了提高梯度稳定性之外,是否真正能够筛选过滤特征仍需进一步验证.

2.3.2 捷径连接

门限的方法引入了大量的参数,增加了模型的复杂度,并且不能完全避免梯度不稳定现象.在 highway network 中,如果门限值接近于 0,将失去使用门限策略的意义.He 等人采用捷径连接(shortcut connection)的思想,提出了残差网络(residual network,简称 ResNet)[43,44].ResNet 利用非线性网络可以拟合任意函数的特性,使用包含少数隐层的浅层网络拟合自定义的残差函数 $F(\boldsymbol{x})=H(\boldsymbol{x})-\boldsymbol{x}$,残差函数再与恒等映射 $\boldsymbol{x}$ 相加,构成基本的残差单元,实现期望的特征映射关系 $H(\boldsymbol{x})$,如图 11(a)所示.将上述残差单元逐个连接构成深层网络,其前馈传播过程为

$$\boldsymbol{y}_l = F(\boldsymbol{x}_{l-1}) + \boldsymbol{x}_{l-1} \tag{64}$$

$$\boldsymbol{x}_l = \sigma(\boldsymbol{y}_l) \tag{65}$$

在反向传播中,梯度迭代计算表达式为

$$\frac{\partial e}{\partial \boldsymbol{x}_{l-1}} = (1 + F'(\boldsymbol{x}_{l-1})) \times \left[\sigma'(\boldsymbol{y}_l) \times \frac{\partial e}{\partial \boldsymbol{x}_l} \right] \tag{66}$$

(a) 正常残差单元 (b) 预激活残差单元

Fig.11 Architecture of residual unit

图 11 残差单元结构

其中,σ表示 ReLU 激活函数,恒等映射的导数为 1,与传统前馈神经网络的反向梯度迭代运算(见公式(3)和公式(10))相比,$1+F'(\boldsymbol{x}_{l-1})$稳定在“1”附近,降低了权重或卷积核的不确定性导致的梯度不稳定现象,而且在反向传播过程中,恒等映射一直存在,极大程度地缓解了梯度不稳定现象,解决了深层网络比浅层网络更难训练的问题.该团队在 2015 ImageNet 图像识别比赛中利用 152 层的 ResNet 取得最高的图像分类准确率.通过后续研究发现[44],恒等映射相比于定值缩放变换、含参缩放变换、1×1 卷积变换以及 Dropout 更适合作为层间捷径.ReLU 激活函数、BN 算法在数据流中的位置对模型效果有直接影响,将激活函数置于浅层网络中,构成预激活(pre-activation)残差单元,如图 11(b)所示.由该单元组成的残差网络模型训练后的准确度更高.

为了充分发挥捷径连接的优势,有学者尝试将网络中的每一层都连接起来,构成稠密网络(DenseNet)[45],图 12 展示了 4 层稠密卷积网络结构.假设存在 m 层稠密网络,第 l 层有 l 个输入,第 l 层输出连接到后面共 $m-l$ 层,整个网络共有 $\frac{m(m+1)}{2}$ 条连接.层与层全连接的网络使数据信息流途径最大化,丰富了特征提取的内容,可以对之前所有节点的输出进行特征提取和重构.反向传播过程的梯度信息流也更加丰富,缓解了梯度不稳定问题.有学者尝试构建了百层以上的深层稠密网络,并应用在图像识别任务中[45].

捷径连接利用残差单元达到了稳定梯度的目的,但是残差单元提取特征的能力仍需研究.在对传统模型隐层特征函数 $H(x)$没有充分认识的前提下,拟合残差函数 $F(x)$无疑是具有挑战性的.残差单元内部参数和结构设计若仍采用传统方法,能否拟合残差函数,能否有效提取特征,仍需考证.目前有学者对此进行了研究,将残差网络根据连接关系展开,解析展开视图(unraveled view),如图 13 所示.将残差网络分解为不同长度的网络路径,并且证明真正有价值的是长度较短的路径,这与残差网络层数越深效果越好的特点相矛盾[46].这说明,部分残差

单元内部的网络结构作用很小,更多的是依靠捷径连接的恒等映射传递来自底层的特征以及来自高层的梯度,从而实现信息的稳定流动.这违背了残差网络的设计初衷,如何真正拟合残差函数,依靠残差单元提取特征,这不仅需要充分认识特征提取函数,而且需要了解构建网络的函数拟合能力,这些内容均需进一步研究.

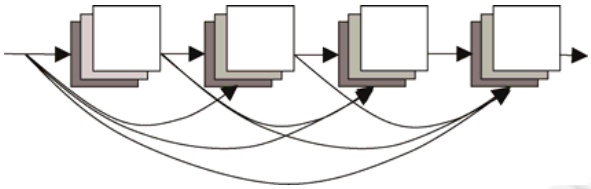


Fig.12 Architecture of DenseNet

图 12 稠密卷积网络结构

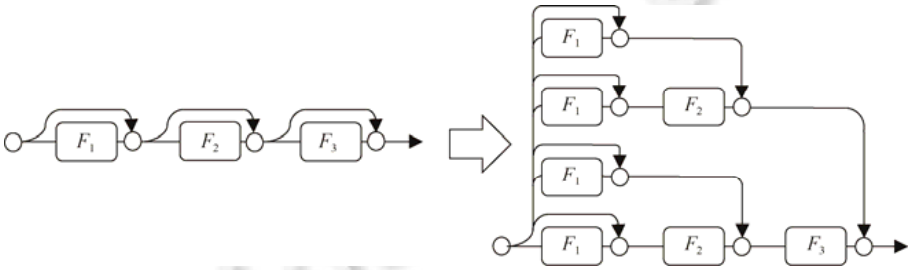


Fig.13 Unraveled view of residual network

图 13 残差网络展开视图

3 梯度不稳定现象未来研究方向

上文依据 3 种策略总结了缓解梯度不稳定现象的研究工作,但梯度不稳定现象依旧存在,导致训练深度神经网络仍然面临着巨大的挑战.根据目前的研究分析,梯度不稳定现象来源于深度神经网络的深层结构,并作用于深度神经网络的梯度下降训练算法,所以未来的研究工作应该从模型设计和训练算法两大方向上展开.

3.1 针对模型设计的研究

(1) 将具有良好梯度稳定特性的模型设计优化思想迁移到更多模型.目前的模型设计优化方法,无论是节点运算或是网络结构,主要思路均是针对某一类模型设计缓解梯度不稳定现象的优化方法,但是这样的方法往往无法直接移植到其他模型,或者移植后无法发挥作用.解决这个问题的思路是:深入理解优化方法的核心思想以及对特定模型有效的原因,再根据其他模型的特点进行调整,从而将优化方法应用到更多模型中,这样做不仅能够检验优化方法的正确性与通用性,还有可能在迁移的过程中产生新的方法或思路,更好地缓解其他模型中的梯度不稳定现象.例如,有学者借鉴前馈神经网络 highway network 和残差网络模型的优秀特性,将其迁移到 LSTM 等循环神经网络结构中,缓解其中的梯度不稳定现象[47−50].所以,在深入理解现有优化方法的基础上,如何将优化思想迁移到更多模型并推陈出新,是未来重要的研究方向之一.

(2) 基于新的梯度稳定性约束条件的模型设计.在参数随机初始化的条件下,训练中各参数的梯度也是不确定的.在本文第 1 节中,论证了在梯度呈现随反向传播逐层递增或递减的不稳定现象时,不利于模型训练.那么梯度应该满足怎样的条件才是稳定且有利于训练的,是否可以通过模型设计使梯度服从这样的约束条件,从而缓解梯度不稳定现象.目前,针对该问题的主要思路是:以分布规律作为梯度稳定性的约束条件,即保证各层梯度的均值和方差是稳定的,并据此设计了相应的参数初始化算法、激活函数等节点运算,目前这方面的研究已经相当完善.最近有学者另辟蹊径,从梯度的 L2 范数入手,结合梯度迭代公式,提出基于酉矩阵的参数结构和激活函数,使得各层梯度的 L2 范数不会随反向传播递增,在一定程度上避免了梯度爆炸现象[51,52].这说明,其他形

式的约束条件也可以指导模型的梯度稳定性设计.因此,从梯度稳定性的约束条件入手,据此设计网络模型,是未来解决梯度不稳定现象的一个重要研究方向.

3.2 针对训练算法的研究

(1) 面向梯度不稳定现象研究新的梯度下降法.当采用梯度下降算法训练深度神经网络时,受梯度不稳定现象的影响,无法有效地更新参数,致使网络训练效果差,说明现有的梯度下降法不适合训练深度神经网络.除了对模型设计优化外,是否可以从训练算法入手,研究新的梯度下降算法,使得在梯度不稳定的条件下,有效训练模型,而且随着不断训练,使各层梯度趋于稳定.目前已有学者提出对梯度采用归一化的方法,强调以梯度方向训练参数,而忽略梯度本身的大小,以定长梯度训练可以避免梯度不稳定现象导致的收敛速度缓慢问题[53,54].而梯度下降算法的关键因素包括梯度和学习率,是否可以通过优化学习率,摆脱梯度不稳定现象的影响.这方面优化方法的提出与证明,不仅需要充分的数学解释,还需要完备的实验验证,这也是一个具有重要价值的研究方向.

(2) 探索无需反向梯度传播的模型训练新算法.深度神经网络训练中发生梯度不稳定现象的原因在于其采用梯度下降法训练,需要以反向传播的方式计算梯度.所以,对训练算法的研究除了改进梯度下降算法之外,尝试颠覆这种传统训练方法,探索其他的训练算法与相应模型,在不需要反向梯度传播的条件下实现模型的训练.深度置信网络的无监督预训练过程即是采用了上述思想[15],在对当前层充分训练后,再开始下一层的训练,不需要从高到低地反向传播梯度,也就避免了梯度不稳定现象的发生.深度置信网络首次挖掘了深度神经网络的巨大潜力,但是由于无监督训练的代价过高,以及针对梯度下降法的不断优化,导致无监督训练逐渐被抛弃.但是该方法为设计无需反向梯度传播的训练算法提供了一个思路,至于更多的无需反向梯度传播的训练算法设计仍需继续研究,包括理论上的收敛性证明以及应用中的模型训练效率.目前,无论是对算法可行性的分析,还是对算法不可行性的证明,相关的结论和研究都比较匮乏,所以这是一个极具挑战性的未来研究方向.

4 结束语

近年来,深度神经网络在机器学习领域展现了优秀的特性,受到了广泛关注.但在其训练过程中发生的梯度不稳定现象,严重影响了模型的学习能力,已经成为制约深度神经网络发展的关键问题.本文针对梯度不稳定现象进行综述:首先,通过分析全连接神经网络、卷积神经网络以及循环神经网络的梯度计算,认为发生梯度不稳定现象的根本原因在于训练中参数的不确定性,并总结了导致梯度不稳定现象的主要因素;然后,通过理论分析与模拟实验,论证了梯度不稳定现象会导致前馈神经网络的多隐层结构失效和收敛速度缓慢,以及循环神经网络的长期依赖问题;之后,以改进训练算法、优化节点运算和调整网络结构这 3 种不同策略,归纳整理了缓解梯度不稳定现象发生的重要方法.结合数学解释,论述了各种方法针对梯度不稳定现象的优化思想;最后,从模型设计与训练算法的角度展望了未来对梯度不稳定现象的研究方向.希望本综述可以让更多的人关注到梯度不稳定现象,并期望与有研究意向的学者共同探索新的研究方向.

References:

[1] Judd S. On the complexity of loading shallow neural networks. Journal of Complexity, 1988,4(3):177–192.

[2] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans. on Neural Networks, 1994,5(2):157–166.

[3] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int'l Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998,6(2):107–116.

[4] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature, 1986,323(6088):533–536.

[5] Liu SG, Zheng CX, Liu MY. Back propagation algorithm in feedforward neural network and its improvement: Progress and prospect. Computer Science, 1996,23(1):76–79 (in Chinese with English abstract).

[6] Lecun Y, Bottou L, Orr GB, Muller KR. Efficient BackProp. Neural Networks Tricks of the Trade, 1998,1524(1):9–50.

[7] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based learning applied to document recognition. Proc. of the IEEE, 1998,86(11): 2278–2324.

[8] Li YD, Hao ZB, Lei H. Survey of convolutional neural network. Journal of Computer Applications, 2016,36(9):2508–2515 (in Chinese with English abstract).

[9] Elman JL. Finding structure in time. Cognitive Science, 1990,14(2):179–211.

[10] Jordan MI. Serial order: A parallel distributed processing approach. Advances in Psychology, 1997,121:471–495.

[11] Williams RJ, Zipser D. Gradient-Based learning algorithms for recurrent networks and their computational complexity. Backpropagation: Theory, Architectures, and Applications, 1995,1:433–486.

[12] Wang XG, Guo YB, Qi ZQ. The effect of activation function on the performance of BP network and its simulation research. Techniques of Automation & Applications, 2002,21(4):15–17 (in Chinese with English abstract).

[13] Huang Y, Duan XS, Sun SY, Lang W. A study of training algorithm in deep neural networks based on sigmoid activation function. Computer Measurement & Control, 2017,25(2):126–129 (in Chinese with English abstract).

[14] Yuan ZR. Artificial Neural Network and Its Application. 5th ed., Beijing: Tsinghua University Press, 1999 (in Chinese).

[15] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Computation, 2006,18(7):1527–1554.

[16] Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. In: Scholkopf B, ed. Proc. of the Int'l Conf. on Neural Information Processing Systems. MIT Press, 2006. 153–160.

[17] Bengio Y. Practical recommendations for gradient-based training of deep architectures. In: Neural Networks: Tricks of the Trade. 2nd ed., Berlin: Springer-Verlag, 2012. 437–478.

[18] Smolensky P. Information processing in dynamical systems: Foundations of harmony theory. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol.1. MIT Press, 1986. 194–281.

[19] Zhang CX, Ji NN, Wang GW. Restricted Boltzmann machines. Chinese Journal of Engineering Mathematics, 2015,32(2):159–173 (in Chinese with English abstract).

[20] Carreira-Perpinan MA, Hinton GE. On contrastive divergence learning. In: Cowell R, ed. Proc. of the 10th Int'l Workshop on Artificial Intelligence and Statistics. Barbados: The Society for Artificial Intelligence and Statistics, 2005. 33–40.

[21] Chen Y. Research on Chinese information extraction based on deep belief nets [Ph.D. Thesis]. Harbin: Harbin Institute of Technology, 2014 (in Chinese with English abstract).

[22] Sun JG, Jiang JY, Meng XF, Li XJ. Application of deep belief nets in spam filtering. Journal of Computer Applications, 2014,34(4): 1122–1125 (in Chinese with English abstract).

[23] Ranzato M, Boureau Y L, Lecun Y. Sparse feature learning for deep belief networks. In: Platt JC, ed. Proc. of the Advances in Neural Information Processing Systems. New York: Curran Associates Inc., 2007. 1185–1192.

[24] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research, 2010,9:249–256.

[25] Thimm G, Fiesler E. Neural network initialization. In: Mira J, ed. Proc. of the Int'l Workshop on Artificial Neural Networks. Berlin: Springer-Verlag, 1995. 535–542.

[26] He KM, Zhang XY, Ren SQ, Sun J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: Proc. of the IEEE Int'l Conf. on Computer Vision (ICCV). IEEE, 2015. 1026–1034.

[27] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Gordon G, ed. Proc. of the Int'l Conf. on Artificial Intelligence and Statistics. 2011. 315–323.

[28] Attwell D, Laughlin SB. An energy budget for signaling in the grey matter of the brain. Journal of Cerebral Blood Flow & Metabolism, 2001,21(10):1133–1145.

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

[30] Hahnloser RHR, Sarpeshkar R, Mahowald MA, Douglas RS, Seung HS. Digital selection and analogue amplification coexist in acortex-inspired silicon circuit. Nature, 2000,405(6789):947–951.

[31] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Pereira F, ed. Proc. of the Advances in Neural Information Processing Systems. New York: Curran Associates, Inc., 2012. 1097–1105.

[32] Dugas C, Bengio Y, Bélisle F. Incorporating second-order functional knowledge for better option pricing. In: Leen TK, ed. Proc. of the Advances in Neural Information Processing Systems. MIT Press, 2001. 472–478.

[33] Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. In: Dasgupta S, ed. Proc. of the Int'l Conf. on Machine Learning. 2013.

[34] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach F, ed. Proc. of the 32nd Int'l Conf. on Machine Learning. 2015. 448–456.

[35] Laurent C, Pereyra G, Brakel P, Ying Z, Bengio Y. Batch normalized recurrent neural networks. In: Proc. of the IEEE Int'l Conf. on Acoustics, Speech and Signal Processing. IEEE, 2016. 2657–2661.

[36] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997,9(8):1735–1780.

[37] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Computation, 2000,12(10): 2451–2471.

[38] Gers FA, Schmidhuber J. Recurrent nets that time and count. In: Proc. of the IEEE-INNS-ENNS Int'l Joint Conf. on Neural Networks. IEEE, 2000. 189–194.

[39] Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Proc. of the 15th Annual Conf. of the Int'l Speech Communication Association. 2014. 338–342.

[40] Cho K, Van MB, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. 2014. 1724–1734.

[41] Srivastava RK, Greff K, Schmidhuber J. Training very deep networks. In: Cortes C, ed. Proc. of the Advances in Neural Information Processing Systems. New York: Curran Associates, Inc., 2015. 2377–2385.

[42] Srivastava RK, Greff K, Schmidhuber J. Highway networks. arXiv: 1505.00387, 2015.

[43] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. IEEE, 2016. 770–778.

[44] He KM, Zhang XY, Ren SQ, Sun J. Identity mappings in deep residual networks. In: Leibe B, ed. Proc. of the 14th European Conf. on Computer Vision. Berlin: Springer-Verlag, 2016. 630–645.

[45] Huang G, Liu Z, Weinberger KQ, *et al*. Densely connected convolutional networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. IEEE, 2017. 2261–2269.

[46] Veit A, Wilber M, Belongie S. Residual networks behave like ensembles of relatively shallow networks. In: Lee DD, ed. Proc. of the Advances in Neural Information Processing Systems. New York: Curran Associates, Inc., 2016. 550–558.

[47] Yao K, Cohn T, Vylomova K, Duh K, Dyer C. Depth-Gated LSTM. arXiv: 1508.03790, 2015.

[48] Prakash A, Hasan SA, Lee K, Datla V, Qadir A, Liu J, Farri O. Neural paraphrase generation with stacked residual LSTM networks. In: Proc. of the 26th Int'l Conf. on Computational Linguistics. 2016. 2923–2934.

[49] Zhang Y, Chen GG, Yu D, Yaco K, Khudanpur S, Glass J. Highway long short-term memory RNNS for distant speech recognition. In: Proc. of the IEEE Conf. on Acoustics, Speech and Signal Processing. IEEE, 2016. 5755–5759.

[50] Kim J, El-Khamy M, Lee J. Residual LSTM: Design of a deep recurrent architecture for distant speech recognition. In: Proc. of the Conf. of the Int'l Speech Communication Association. 2017. 1591–1595.

[51] Arjovsky M, Shah A, Bengio Y. Unitary evolution recurrent neural networks. In: Balcan MF, ed. Proc. of the Int'l Conf. on Machine Learning. 2016. 1120–1128.

[52] Henaff M, Szlam A, Lecun Y. Recurrent orthogonal networks and long-memory tasks. In: Balcan MF, ed. Proc. of the Int'l Conf. on Machine Learning. 2016. 2034–2042.

[53] Hazan E, Levy KY, Shalevshwartz S. Beyond convexity: Stochastic quasi-convex optimization. In: Cortes C, ed. Proc. of the Advances in Neural Information Processing Systems. New York:Curran Associates, Inc., 2015. 1594–1602.

[54] Yu AW, Lin Q, Salakhutdinov R, Carbonell J. Normalized gradient with adaptive stepsize method for deep neural network training. arXiv: 1707.04822, 2017.

附中文参考文献:

[5] 刘曙光,郑崇勋,刘明远.前馈神经网络中的反向传播算法及其改进:进展与展望.计算机科学,1996,23(1):76–79.

[8] 李彦冬,郝宗波,雷航.卷积神经网络研究综述.计算机应用,2016,36(9):2508–2515.

[12] 王雪光,郭艳兵,齐占庆.激活函数对 BP 网络性能的影响及其仿真研究.自动化技术与应用,2002,21(4):15–17.

[13] 黄毅,段修生,孙世宇,郎巍.基于改进 sigmoid 激活函数的深度神经网络训练算法研究.计算机测量与控制,2017,25(2):126–129.

[14] 袁曾任.人工神经元网络及其应用.北京:清华大学出版社,1999.

[19] 张春霞,姬楠楠,王冠伟.受限波尔兹曼机.工程数学学报,2015,32(2):159–173.

[21] 陈宇.基于深度置信网络的中文信息抽取方法[博士学位论文].哈尔滨:哈尔滨工业大学,2014.

[22] 孙劲光,蒋金叶,孟祥福,李秀娟.深度置信网络在垃圾邮件过滤中的应用.计算机应用,2014,34(4):1122–1125.

陈建廷(1995－),男,吉林省吉林市人,硕士生,主要研究领域为数据挖掘.

向阳(1962－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为数据挖掘.