

# 基于反向随机投影的神经网络改进算法

许邓艳<sup>1</sup>, 卢民荣<sup>2,3</sup>

(1.福建工程学院 应用技术学院, 福建 福州 350118;

2.福建江夏学院 会计学院, 福建 福州 350108;

3.福建省社科研究基地财务与会计研究中心, 福建 福州 350108)

**摘要:** 提出基于反向随机局部投影的神经网络效率改进算法, 通过降低深度学习中的网络规模, 重点解决了从“局部连接”到“全连接”和随机节点抽取时输入端节点信息丢失的问题, 从而提升网络的效率。在算法中设置缩减参数, 提升了算法的可伸展性, 以适用于不同数据集的学习。通过数据集 ISO-LET 进行实验, 结果表明, 基于反向随机局部投影的神经网络效率改进算法的准确率、效率分别平均提升了 3.48% 和 105.21%; 在迭代 20 次的实验中进行缩减参数调节实验, 当参数设置为 1.4 时其准确率则优于传统全连接网络 2.61%, 效率提升了 272.78%, 具有明显的优势。

**关键词:** 反向随机投影; 局部投影; 神经网络; 局部连接

中图分类号: TP391.9

文献标志码: A

文章编号: 1672-4348(2020)04-0358-07

## An improved neural network algorithm based on reverse random projection

XU Dengyan<sup>1</sup>, LU Minrong<sup>2,3</sup>

(1.School of Applied Technology, Fujian University of Technology, Fuzhou 350118, China;

2. School of Accounting, Fujian Jiangxia University, Fuzhou 350108, China;

3. Finance and Accounting Research Centre of Fujian Social Science Research Base, Fuzhou 350108, China)

**Abstract:** A neural network efficiency improvement algorithm based on reverse random local projection was proposed. By reducing the network size in deep learning, the problem of information loss of input nodes caused by “local connection” to “full connection” and random node extraction was solved, thus improving the efficiency of the network. The reduction parameters were set in the algorithm to improve the extensibility of the algorithm, so that it can be applied to the learning of different data sets. Then, experiments were conducted, using data set ISOLET. Results indicated that the accuracy and efficiency of the algorithm based on reverse random local projection were increased by 3.48% and 105.21% respectively. When echos were set as 20 the reduction parameter adjustment experiment was carried out, and when the parameter was set as 1.4, the accuracy was improved by 2.61% compared with the traditional fully-connected network, and the efficiency was improved by 272.78%, which showed obvious advantages.

**Keywords:** reverse random projection; local projection; neural network; local connection

近年来,神经网络的算法、模型及研究内容相当广泛,反映出多学科交叉技术领域的特点。以人工智能的应用为代表,如专家系统、机器人等应用系统,已经具备机器学习的特性<sup>[1-2]</sup>。基于神

神经网络算法的深度学习,能够让机器自我学习以及模式识别,相关学习算法不仅受学者关注,也吸引了 IBM、Google、Baidu 等 IT 公司投入大量应用研发。随着机器学习、深度学习的发展,学习网络

规模越来越大、复杂度越来越高,学习效率问题成为算法最关键的也是最难的研究热点<sup>[3-4]</sup>。从1986年提出BP神经网络(back propagation, BP)至今,BP仍是目前研究和应用最广泛的模型<sup>[5]</sup>,受网络的中间层数、各层的神经元个数的影响,全连接神经网络规模通常都比较大,因此网络的学习速度慢和容易陷入局部极小值<sup>[6-7]</sup>。以卷积神经网络(convolutional neural networks, CNN)为典型代表的局部连接网络,通过池化、卷积可以大幅降低网络的规模,有效提升训练效率,然而卷积感受野过大就会影响感知效率,另外随着网络深度的增加,不断加深抽象层级也会影响计算效率<sup>[8]</sup>。循环神经网络(recurrent neural network, RNN,也称递归神经网络)和在RNN基础上延伸出来的长短期记忆人工神经网络(long-short term memory, LSTM),主要是通过全连接网络中的数据集局部随机训练,在效率和准确率上比BP有明显优势。另一些学者是通过适当的低秩因子来移除深度神经网络卷积核中的冗余参数来降低网络规模,也有基于参数共享和 Huffman 编码来压缩网络的规模,从而实现比较高的压缩率<sup>[9-11]</sup>。近年,运用深度神经网络连接权重稀疏和对滤波器进行谱聚类后修剪等大幅提升了网络的效率<sup>[7,12]</sup>。综上,CNN网络通过数据特征卷积(再卷积)的局部网络逐层缩小后,效率大幅提升,而RNN和LSTM则是训练数据的局部化(随机性)缩小训练规模,以及基于这些模型提出的快速学习由于不具有图灵完备性而导致学习能力也低于全连接网络,考虑了效率则损失了准确率。因此,我们有必要从“局部连接”回到“全连接”的神经网络进行连接优化<sup>[13]</sup>,从而在保证准确率的情况下提升效率。

针对神经网络的效率问题,本研究提出一种基于反向随机局部投影的神经网络效率改进算法,引入连接缩小参数降低除倒数第2层至输出层外的其它网络规模,同时运用缩减集合解决传统基于随机取数存在一定信息丢失的概率问题,倒数第2层至输出层仍然保留“全连接”,从而实现从“局部连接”回到“全连接”神经网络的连接优化。最后通过26字母发音数据集 ISOLET 不同参数、不同网络层数的测试,来论证在保证不降低准确率的情况下,基于反向随机局部投影的网络在语音识别上的效率有较大提升。

## 1 模型

### 1.1 BP全连接网络原理

以典型的3层神经网络为例,网络实现主要分为前向网络和反向网络,神经网络的节点与权重的工作原理基本雷同,具体步骤如下<sup>[5-7]</sup>:

第一步:各层节点准备,设置网络的迭代次数。

准备输入层信息  $X(x_1, x_2, x_3, \dots, x_n)$ ,  $n$  表示输入的节点数(实验中数据集的维度大小),输出层信息  $Y(y_1, y_2, y_3, \dots, y_m)$ ,  $m$  表示输出的节点数(实验中数据集的分类数),由于网络层数、神经元个数的选择没有相应的理论指导<sup>[15-16]</sup>,  $h$  表示中间神经元个数,一般用  $h = [\sqrt{m \times n}]$  表示,  $[\ ]$  表示直接取整。

第二步:初始化各层全连接权重,第一次需要初始化,后面根据误差反向传播进行更新。

第1至2层:0到1之间随机生成  $n \times h$  权重矩阵,用  $W_1$  表示;同理生成第2至3层权重  $h \times m$ ,用  $W_2$  表示。

第三步:第1层到第2层前向信息传输。

首先,把第1层的输入转为第1层的输出即  $O1 = X$ ,再与  $W_1$  计算获得第2层的输入(公式1),然后对第2层的输入进行激活,生成第2层的输出(公式2)。

$$X'_j = \sum_{i=1}^n (O1_i \times W1_{i,j}) \quad (1)$$

式中,  $O1_i$  表示第1层的第  $i$  个节点(神经元)的输出,  $X'_j$  表示第2层的第  $j$  个节点的输入。

$$O2 = \sigma(X'_j) \quad (2)$$

式中,  $\sigma$  表示激活函数,常见的激活函数有 sigmoid、tanh、Relu 等函数,实验中采用  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ 。

第四步:重复第二至第三步(对于多层神经网络,重复至输出层即结束前向网络),将矩阵计算换成  $W2$ ,输出激活后为  $O3$ ,然后判断网络的输出  $O3$  和实际输出的  $Y$  的误差。

第五步:计算总误差,并进行反向传播。

首先把网络的输出  $O3$  和实际输出的  $Y$  进行相差比较,获得总误差  $E$ ,计算如公式3示。

$$E3 = Y - O3 = \sum_{i=1}^m (Y_i - O3_i) \quad (3)$$

然后进行反向传播第 3 层(也叫输出层、最终层、结果层等,示例为 3 层网络)到第 2 层,由于矩阵运算的原因,需要对第 2 至 3 层权重  $h \times m$ ,  $W_2$  进行转置即  $W_2^T = W(m \times h)$ , 进行计算误差平摊即每个节点的误差  $\Delta E$ , 如公式 4 示。

$$\Delta E_2 = \frac{\partial E_3}{\partial W_2^T} = \partial \cdot O_3(1 - O_3) \quad (4)$$

式中,  $\Delta E_2$  表示第 2 层节点输出的平均误差,  $\partial$  为学习率, 不同应用则网络设定的权值不一样, 因为设置过高会使梯度下降时找不到最优解, 设置过低又容易让网络效率变低。

第六步:更新权重,这也是网络的核心。

根据上述公式(3)、(4)推导,把平均误差计算出来分摊至各个连接的权重进行调节,如公式 5。

$$\Delta W_2 = \partial \times E_3 \times \sigma(O_3) \times (1 - \sigma(O_3)) \cdot O_3^T \quad (5)$$

式中,  $\Delta W_2$  更新后的第 2 层权重,  $\partial$  为学习率,  $\sigma$  表示激活函数,  $O_3^T$  表示  $O_3$  的转置。

第七步:重复第五至第六步(对于多层神经网络,重复至输入层即结束反向网络),计算误差和权重矩阵的更新,即完成  $W_1$  的更新。

第八步:继续前向网络、反向网络即第二到第七步,直至整个网络迭代结束。

## 1.2 局部网络模型

权重是神经网络的核心部分<sup>[13]</sup>,其计算量也是最大的部分,前向网络中需要同神经元进行矩阵相乘运算,在反向网络中需要转置后与误差进行矩阵相乘运算,再根据学习率修正权重矩阵的值,因此,我们可以在现行网络结构和工作原理不变的情况下,用降低权重矩阵思路优化局部连接,提升网络的效率(本方法也称为局部网络模型),另一方面,为了让输出层与输入层有较为直接连接则让第 2 至 3 层网络连接(最后一层的网络连接)仍然使用全连接方式,具体改进后的网络示意图如图 1。

在图 1 中,左图虚线为不连接部分,原第 1 到 2 层全连接权重矩阵规模为  $n \times h$  即  $n \times [\sqrt{m \times n}]$ ,拟用连接缩小参数 *reduce* 将连接数缩小,虚线即为缩小部分。右图为缩小后的连接情况示例,数字 1、2、3 表示输出次数即权重连接次数,连接匹配也根据缩小后的连接进行相应的变化,该变化需要解决的难点有两个:(1)如何保

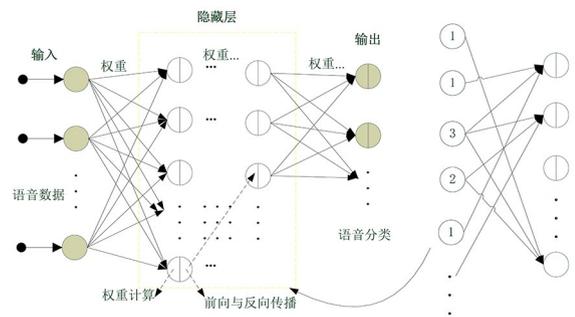


图 1 改进连接后的神经网络结构

Fig.1 Improved neural network structure

证前面节点都要有输出;(2)如何保证前面节点的输出不得过于集中某几个节点,即如何平均分布。在前期的研究中,随机取数存在一定信息丢失的概率问题,为解决连接匹配两个难点,算法在初始权重过程中需对各层设置输入集合和缩减集合,因此时间效率在初始化网络上不如全连接网络。然而影响网络的效率主要是学习过程的前向网络和反向网络权重计算,因此局部投影缩小规模在效率上有较大提升。

## 1.3 算法实现

基于改进的权重连接主要是通过输出端(相对上一层网络的输入,图 1 中的右侧数字节点)反向局部投影到输入端(相对下一层网络的输出,图 1 中的右侧非数字节点),为方便表达,仍然使用上述经典的 3 层网络,变量的定义同上述说明,具体平均分布投影过程如下:

第一步:根据缩小参数计算连接数公式(6)和输入端节点平均输出公式(7)。

$$W_{\text{count}} = \text{ceil}\left(\frac{n}{\text{reduce}}\right) \times h = \text{ceil}\left(\frac{n}{\text{reduce}}\right) \times [\sqrt{m \times n}] \quad (6)$$

式中,  $W_{\text{count}}$  表示连接规模缩小后的权重连接数,  $\text{ceil}$  表示向上取整,即如果有小数点则直接进位,用  $O_{\text{ceil}} = \text{ceil}\left(\frac{n}{\text{reduce}}\right)$  表示局部连接数。

$$O_{\text{min}} = \left\lfloor \frac{W_{\text{count}}}{n} \right\rfloor \quad (7)$$

式中,  $\lfloor \cdot \rfloor$  表示取整,如果有小数直接去除,  $O_{\text{min}}$  表示输入端节点平均最少输出。而  $O_{\text{min}} + 1$  则为平均最多输出  $O_{\text{max}}$ 。

第二步:反向随机局部投影连接,反向投影具

体做法:

(1)从输出端第一节点开始,从输入端节点1至 $n$ (输入端集合用 $I$ 表示)中随机取局部连接数 $O_{ceil}$ ,被取出的节点设置该节点随机权重 $W(O_{ceil} \times h)$ 并记入输入端节点号;

(2)对已经反向投影的局部连接节点进行记数 $O_c$ (从1开始),为保证所有输入节点信息都能被传递到输出端,在输入端集合 $I$ 中随机去除 $O_c$ 中记数最高的至缩减集合 $R$ 中。

(3)改进随机局部连接算法,部分研究表明随机局部连接算法确实有效,但传统基于概率的随机取数存在一定信息丢失的概率问题<sup>[13,16-17]</sup>。改进方法为如果 $I$ 的数量小于 $O_{ceil}$ ,则从 $R$ 中再还原至少 $I$ 的数量和 $O_{ceil}$ 数量差,还原规则为 $O_c$ 最小优先,如已经达到输入端节点最多输出( $O_{avg} + 1$ )则不再被还原,具体 $I$ 、 $R$ 、 $O_{ceil}$ 等集合的出入栈结构如图2示。

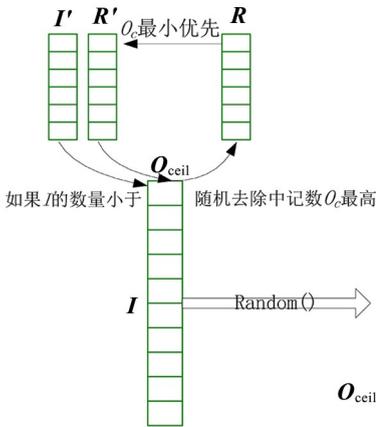


图2 随机局部连接各集合的出入栈结构

Fig.2 Structure of random local connections

(4)继续输出端下一节点的投影,重复(1)-(3),直至输出端节点结束。

第三步:如果为多层网络,重复第一、二步,直到倒数第2层。

第四步:从“局部连接”再到全连接。

在倒数第2层到最后1层,采用全连接输出,不要再使用局部投影,以免前期输入到最终输出产生丢失。

在3层网络中,第2至3层仍为全连接网络,因此算法主要以网络中第2层(输出端)到第1层(输入端)反向随机局部投影连接算法进行设计,具体实现的过程如下:

Inputs: input\_nodes: 输入端,  
hidden\_nodes: 输出端,  
reduce: 连接缩小参数.

Outputs: weights: 局部投影的权重.

```

n, h ← len ( input_nodes ), len ( hidden_nodes ) #输入与输出端节点数
oceil, omin, omax ← ceil ( n/reduce ) ,
int(oceil * h/n) , omin + 1 #节点连接数
oc, weights ← [ ( 0 * 2 ) for i in range
(h) ] #初始化节点取数和权重矩阵
I, T, R ← [ ( i * 1 ) for i in range(h) ] ,
[ ], [ ] #初始化输入端、临时和缩减集合
for i in range(hidden_nodes) : #输出端
节点迭代
    I ← I-R #排除缩减集合
    for j in range(oceil) : #随机取数,个数为 oceil 则停止
        t ← random.randint ( I )
        T.add ( t ) , I.pop ( t ) #临时入栈和出栈
        oc [ t ] ← oc [ t ] + 1 #计数累加
        weights.append ( [ t , random ( 0 , 1 ) ] ) #
        随机权重
    oc ← sort ( oc , 1 ) #按计数排序
    R ← random ( I , oc , random.randint
( I ) ) #随机去除
    if ( len ( I ) < oceil ) : #防止去除过多
        I ← oc [ i ] <= omaxΔ random ( R , oc ,
random.randint ( R ) ) : pass #超限则不再被
        还原
    return weights

```

## 2 实验及分析

实验硬件为联想 Thinkserver RD350 服务器,操作系统为 Windows Server 2008 R2。在软件配置上需要 Python3.7 及相应的运行包安装,开发环境使用 MyEclipse 2017 CI 开发工具。在实验中,采用 BP 网络和本算法针对同一标准数据集进行比较。

### 2.1 参数设置及数据集

为了方便实验对比,采用标准的数据集 ISOLET(训练集和测试集各 26 组数据,表 2)进行

不同侧面的准确率、效率对比分析,以论证反向随机局部投影的改进情况。基于 BP 神经网络的局部连接网络参数(主要设置了缩减参数)配置如表 1 示。

表 1 局部连接网络模型的参数配置

Tab.1 Parameter configuration of local connection network model

序号	参数	值	说明
1	输入层	617	全维度读取数据 617
2	输出层	26	26 个字母分类
3	隐藏层数	127	$\sqrt{617 \times 26}$
4	激活函数	tanh	
5	学习率	0.01	可调整更小
6	迭代次数	20(6, 10, 50, 100)	不同迭代比较准确率
7	缩减参数	1.8(1.4, 1.6, 2.0, 2.5, 3.0)	不同的网络规模

表 2 ISOLET 数据集

Tab.2 ISOLET data set

NO	类型	样本数	文件
1	训练集	6 238	isolet1+2+3+4.data
2	测试集	1 559	isolet5.data

注: Fanty, M., Cole, R.等提供,对口语音母进行分类主要用于语音字母识别和神经信息处理系统 <https://github.com/sethdhaney/isolet-SciKit-ANN>。为便于效果比较,采用 3 次训练的结果取平均值,以下数据均为该数据集的 26 组数据平均值。

## 2.2 模型参数变化策略

反向随机局部投影的神经网络规模随着缩减参数的增加而增多,其模型效率提升,但准确率也会下降,反之也成立,当缩减参数为 1.0 时则模型和全连接网络一致。基于神经网络的深度学习模型中,随着迭代次数增加(也称“学习深入”),网络训练结果准确率逐步提高,但时间消耗也增加(也称“学习效率”)<sup>[18]</sup>,实验以准确率能达到 90%左右效果为基准,取迭代 20 次,分别调整缩减参数为:1.4、1.6、1.8、2.0、2.5、3.0 进行实验,准确率、效率实验效果如图 3、图 4 所示。

在图 3 中,由于全连接网络是没有缩减参数调节,因此其准确率是一条固定的直线,在实验的数据集中缩减参数在 1.8、2.0 时基本和全网络接近,而设置为 1.4、1.6 时则优于传统全连接网络。不过缩减参数设置较大,如取 3.0 时其准确率下降较多,因此,ISOLET 数据集不宜设置过高的缩减参数。

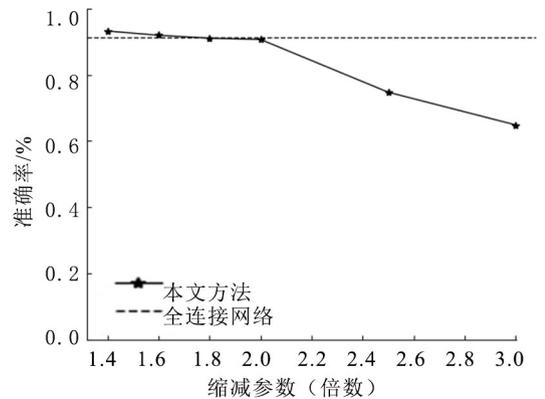
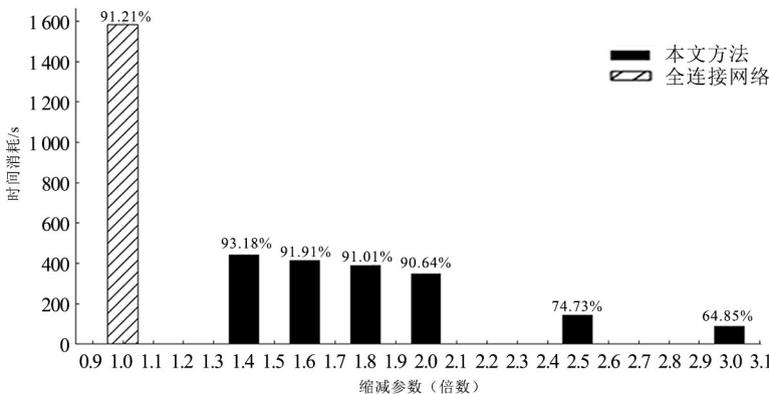


图 3 准确率调节比较(迭代 20 次)

Fig.3 Comparison of accuracy adjustment (echos = 20)



注:准确率显示在时间柱上.%

图 4 效率调节效果(迭代 20 次)

Fig.4 Efficiency regulation effect (echos = 20)

图 4 为不同缩减参数的效率比较,为方便比较将全连接网络时间柱放在 1.0 处(BP 全连接网络无此调节参数),从图中可以看出全连接网络时间消耗很大,而且准确率也没有明显优势。当反向随机局部投影的神经网络的缩减参数设置为 1.4 时准确率则优于传统全连接网络 2.61%,效率上却大幅提升了 272.78%;设置为 1.6、1.8 接近全连接网络,而效率更进一步提升。

### 2.3 模型准确率与效率评价

为了方便不同网络规模的定性比较,以下各实验的对比分析采用准确率接近全网络的缩减参数,定为 1.8。在实验中,以测试集的分类准确率来比较模型的准确率,使用两种模型不同迭代的曲线图进行对比,如图 5 所示,以数据集训练和测试时间的总和进行评价模型的效率,如图 6 所示。

在图 5 中,反向随机局部投影的算法在迭代次数比较低的情况下,学习效果不全面,因此准确率不理想。随着学习的深入(迭代的增加)学习效果明显提升,在迭代 20 次的时候已经处于比较稳定的效果和接近全网络模型。

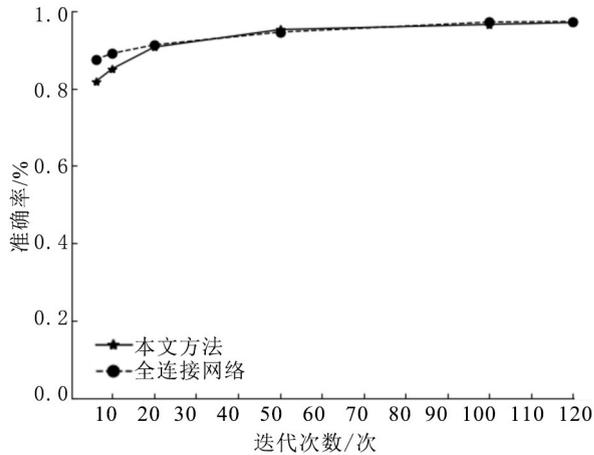


图 5 准确率比较(不同迭代)

Fig.5 Comparison of accuracies (different echos)

在图 6 中,在相同的迭代的情况下效率,可看出传统的全连接网络效率低下,在迭代 20 次的时候,两种网络的准确率基本一致的情况下,本文方法时间消耗却能节省 353.8%。在时间消耗上,本文方法迭代 100 次比全连接网络迭代 20 次还低,准确率从 91.2%(全连接网络迭代 100 次)提升到 96.4%(本方法迭代 20 次)。随着迭代的增加,本文方法所体现出来的效率优势越来越明显,时间

差距也越来越大。

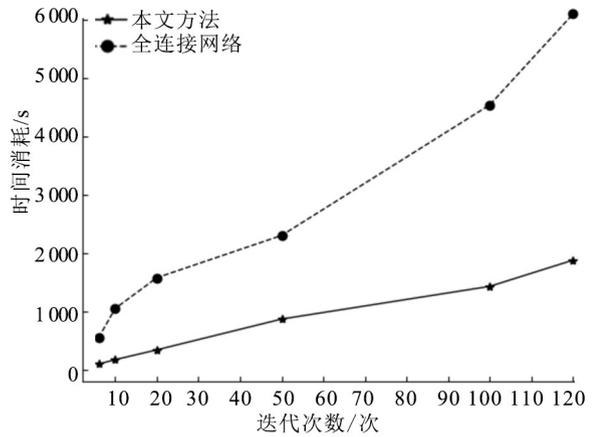


图 6 效率比较(不同迭代)

Fig.6 Efficiency comparison (different echos)

实验表明,迭代 20 次以上时,本方法和全连接网络准确率已经基本接近,而本方法的效率却大幅提升,在同等时间消耗情况下远优于传统全连接网络训练准确率。以局部网络迭代 20 次为例,其准确率已经达到 90% 以上,超过了全连接网络的迭代 6、10 次的效果,时间却节约了一半,以迭代 20、50 次为例对性能提升详细比较如表 3 所示。

表 3 不同迭代的性能比较

Tab.3 Comparison of performance in terms of different echos

迭代次数		准确率提升/	效率提升/
本方法	全网络连接	%	%
20	6	3.60	58.74
20	10	1.80	202.87
50	10	6.89	20.52
50	20	4.33	80.62
50	50	0.69	163.28
平均		3.4	105.21

从表 3 可见,反向随机局部投影算法的平均准确率虽然只提升 3.48%,但平均效率提升却十分明显,具有明显的效率优势,而且在保证准确率的同时,效率提升有效。另外,本算法可以根据不同数据集进行参数调优,进一步体现了算法的伸

缩性。

### 3 结论

1) 本研究提出的反向随机局部投影, 可以按缩减参数调节网络规模, 随着缩减参数的增加而增多, 其模型效率提升, 但准确率也会下降, 反之也成立, 当缩减参数为 1.0 时则模型和全连接网络一致。

2) 随机局部连接算法是基于 BP 神经网络模

型中的全连接网络基础上, 在改进效率的同时保证了准确率。实验中, 算法充分体现了其性能的优越性, 准确率、效率分别平均提升了 3.48% 和 105.21%, 缩减参数同时提升了模型的可适应性。

3) 模型与数据集可以在训练中寻找缩减参数最佳值, 但局部网络普遍存在信息丢失问题, 与全连接网络的准确率仍然有细微差距, 后期研究还需要通过对数据集进行降维、去噪等预处理, 从而更进一步提升算法的准确率。

### 参考文献:

- [1] 刘建伟, 王园方, 罗雄麟. 深度记忆网络研究进展[EB/OL]. 计算机学报, 2020;1-52[2020-08-11]. <http://kns.cnki.net/kcms/detail/11.1826.TP.20200114.1543.004.html>.
- [2] HONAVAR V, UHR L. Artificial intelligence and neural networks-steps toward principled integration[J]. Science, 1994, 9(3): 545-548.
- [3] CHEN Y, EMER J, SZE V. Using dataflow to optimize energy efficiency of deep neural network accelerators[J]. IEEE Micro, 2017, 37(3): 12-21.
- [4] 朱虎明, 李佩, 焦李成, 等. 深度神经网络并行化研究综述[J]. 计算机学报, 2018, 41(8): 1861-1881.
- [5] DING S, SU C, YU J. An optimizing BP neural network algorithm based on genetic algorithm[J]. Artificial Intelligence Review, 2011, 36(2): 153-162.
- [6] 张禹, 李东升, 董小野, 等. 基于改进 BP 神经网络面向 STEP-NC 2.5D 制造特征的智能宏观工艺规划[J]. 机械工程学报, 2020, 56(1): 148-156.
- [7] ZHAO Y, ZHOU D, YAN H. An improved retrieval method of atmospheric parameter profiles based on the BP neural network[J]. Atmospheric Research, 2018, 213: 389-397.
- [8] 马力, 王永雄. 基于稀疏化双线性卷积神经网络的细粒度图像分类[J]. 模式识别与人工智能, 2019, 32(4): 336-344.
- [9] 韩敏, 穆云峰. 基于 RBFLN 网络的改进 RBF 神经网络学习算法[J]. 系统工程学报, 2008, 23(6): 764-768.
- [10] DENTON E, ZAREMBA W, BRUNA J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//NIPS. MIT Press, 2014: 1269-1277.
- [11] LI X, PENG L, YAO X, et al. Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation[J]. Environmental Pollution, 2017, 231: 997-1004.
- [12] ROY D, PANDA P, ROY K. Tree-CNN: a hierarchical deep convolutional neural network for incremental learning[J]. Neural Networks, 2020, 121: 148-160.
- [13] 王帅, 王维莹, 陈师哲, 等. 基于全局和局部信息的视频记忆度预测[J]. 软件学报, 2020, 31(7): 1969-1979.
- [14] 陈铬亮, 徐佳. 基于单词分类的归一化神经网络语言模型研究[J]. 福建工程学院学报, 2016, 14(4): 382-385.
- [15] RAHELI B, AALAMI M, EL-SHAFIE A, et al. Uncertainty assessment of the multilayer perceptron (MLP) neural network model with implementation of the novel hybrid MLP-FFA method for prediction of biochemical oxygen demand and dissolved oxygen: a case study of Langat River[J]. Environmental Earth Sciences, 2017, 76(14): 1-16.
- [16] ARJMANDZADEH Z, NAZEMI A, SAFI M. Solving multiobjective random interval programming problems by a capable neural network framework[J]. Applied Intelligence, 2019, 49(4): 1566-1579.
- [17] 冶忠林, 赵海兴, 张科, 等. 基于邻节点和关系模型优化的网络表示学习[J]. 计算机研究与发展, 2019, 56(12): 2562-2577.
- [18] 黄德根, 张云霞, 林红梅, 等. 基于规则推理网络的分类模型[J]. 软件学报, 2020, 31(4): 1063-1078.