激光与光电子学进展 先进成像

光学神经网络训练算法中超参数对网络性能的影响

曹雯，刘美玉，陆鸣豪，邵晓锋，刘启发，王瑾*
南京邮电大学通信与信息工程学院，江苏 南京 210003

摘要 面向数字图像识别，使用光学器件构建基于快速傅里叶变换（FFT）的光学神经网络（ONN），其中的线性光学处理单元由马赫-曾德尔干涉仪（MZI）实现。这些MZI以网格状布局连接，对通过的光信号进行调制，实现乘法和加法，从而实现对图像的分类识别。针对该ONN对手写数字图像进行识别出现的问题，研究训练算法中的主要超参数即动量系数和学习率对网络性能的影响。首先比较不同学习率下随机梯度下降（SGD）、均方根传递（RMSprop）、适应性矩估计（Adam）和自适应梯度（Adagrad）4种训练算法结合不同非线性函数和不同隐藏层个数后，ONN在识别手写数字图像上的表现。实验结果显示：在学习率从0.5变化到5×10^{-5}、RMSprop训练算法下，具有2个隐藏层、非线性函数为Softplus的FFT型ONN具有最高的识别精确度，达97.4%。此外，着重分析在具有不同动量系数的SGD算法结合不同非线性函数和不同隐藏层个数时ONN对手写数字图像识别的准确率、运行内存和训练时间的影响。进一步，在学习率为0.05和0.005时，比较了SGD、RMSprop训练算法以及各自在引入动量后的网络识别性能。实验结果显示：动量系数为0时，采用SGD算法训练的具有2个隐藏层、非线性函数为Softplus的ONN的识别精度为96%，动量系数为0.9时，ONN的识别精度提高到96.9%；而加入动量的RMSprop算法会导致网络识别准确率不收敛或收敛较慢。

关键词 光学神经网络；马赫-曾德尔干涉仪；训练算法；动量；学习率
中图分类号 TP183 **文献标志码** A **DOI**：10.3788/LOP230535

Influence of Hyperparameters on Performance of Optical Neural Network Training Algorithms

Cao Wen, Liu Meiyu, Lu Minghao, Shao Xiaofeng, Liu Qifa, Wang Jin*
School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, *Jiangsu, China*

Abstract An optical neural network (ONN) based on fast Fourier transform (FFT) is constructed for digital image recognition in optical devices. Herein, ONN uses Mach-Zehnder interferometer (MZI) as its linear optical processing unit. These MZIs are connected in a grid-like layout and modulate the passing optical signals to achieve multiplication and addition. Subsequently, MZIs achieve classification and recognition for images. In this study, the influence of main hyperparameters (e. g. , momentum coefficient and learning rate of the training algorithm) on the performance of ONN in recognizing handwritten digital images is investigated. First, the performance of ONN with four training algorithms in recognizing handwritten digital images under different learning rates is compared. These algorithms connect with different nonlinear functions and different number of hidden layers, namely, stochastic gradient descent (SGD), root mean square prop (RMSprop), adaptive moment estimation (Adam), and adaptive gradient (Adagrad). Additionally, the accuracy, running memory, and training time of ONN with the SGD algorithm connected with different nonlinear functions and different number of hidden layers are analyzed under different momentum coefficients. The recognition performance of ONN with SGD and RMSprop training algorithms is also compared after the introduction of momentum, where the learning rate is 0.05 and 0.005. The experimental results show that when the learning rate changes from 0.5 to 5×10^{-5}, the FFT-typed ONN with the RMSprop training algorithm, two hidden layers, and the nonlinear function of Softplus has the highest recognition accuracy, reaching 97.4%. Furthermore, for the momentum coefficient of 0, the ONN with two hidden layers and the nonlinear function of Softplus trained by the SGD algorithm exhibits the highest recognition accuracy of 96%, when the momentum coefficient increases to 0.9, the accuracy of ONN is improved to 96.9%. However, the

收稿日期：2023-01-30；**修回日期**：2023-02-18；**录用日期**：2023-02-27；**网络首发日期**：2023-03-09
基金项目：国家自然科学基金（61575096）
通信作者：*jinwang@njupt.edu.cn

RMSprop algorithm with momentum leads to nonconvergence or slow convergence of network recognition accuracy.

Key words optical neural network; Mach-Zehnder interferometer; training algorithm; momentum; learning rate

1 引 言

近年来随着机器学习技术的发展，深度神经网络在各种新兴应用中表现出极大的性能提升[1-2]。具有深度分层结构的人工神经网络在解决普遍存在的大规模计算问题方面表现出优秀的能力[3-5]。随着深度学习模型和数据集规模的不断扩大，对电子处理器的计算能力提出了更高的要求。但是，训练最先进的人工智能相关应用程序所需的计算力每3.5个月就翻1倍[6]，复杂的结构和大量的参数在训练和推理过程中消耗了大量的资源，计算本质上受制于处理单元和存储单元之间的数据传输速率[7-9]，因此，对高速、低功耗的神经网络加速器有着迫切的需求。由于光学元件和技术具有超宽带和低功耗的特点，光学系统是潜在的下一代神经网络加速器[10-11]。

得益于光电子器件制造技术的发展和成熟，尤其是集成光电子技术，光学神经网络（ONN）技术取得了突破性进展。2017年，Shen等[12]提出并实验演示了一种使用奇异值分解（SVD）的级联可编程马赫-曾德尔干涉仪（MZI）网络实现的相干光学计算架构，但未解决数据大量传输导致功耗大的问题。2018年，Hughes等[13]深入讨论了通过反向传播和梯度测量训练的光学神经网络，这项工作为有效实现片上训练和优化可重构集成光学平面提供了途径，但是距离实际应用还存在很多问题，如每个移相器处光强测量带来的大量损耗等。2019年，Zhang等[14]提出了一种新的基于神经进化的学习算法来设计和训练神经元，该算法在模式识别和深度强化学习方面具有广阔的应用前景，但神经进化的训练消耗较大。2020年，Zang等[15]提出了一种基于时间拉伸方法的电光神经网络结构，并采用随机梯度下降（SGD）算法训练三层电光神经网络，该网络在手写数字识别任务上的识别准确率为94%，有待提高。

为解决训练中网络性能较低、训练方式单一的问题，本文针对训练中不同算法中的超参数对网络性能的影响进行研究。通过研究算法的学习率、动量对光学神经网络的影响，找出超参数的最佳配置，以提高ONN的数字识别精确度，更大程度上减少能耗和识别时间。本文的主要贡献如下。1）与以往采用单一的训练参数的网络不同，所提网络采用多种训练参数。测试SGD、均方根传递（RMSprop）、适应性矩估计（Adam）和自适应梯度（Adagrad）4种算法在学习率为0.5、0.05、0.005、5×10^{-4}和5×10^{-5}下，含2个隐藏层的ONN在Softplus和ReLU分别作为非线性函数时的性能表现，并与4种算法在不同学习率下，含1个隐藏层的ONN在Softplus和ReLU分别作为非线性函数时的性能表现作对比。此外，在SGD和RMSprop算法中加入动量参数，分析动量参数的设置对训练后ONN识别性能的影响。2）为了验证所提ONN模型的有效性，将优化后的ONN在MNIST数据集上进行验证[16]，与Fang等[17]所述的识别精确度为94.8%的ONN相比，准确率有较大提升，即所述优化方案是有效的。3）以识别精确度、运行内存、训练时间3个指标去评价光学神经网络性能，为在光学中实现算法优化提供实验基础。

2 光学神经网络工作原理与架构

2.1 MZI型矩阵向量乘法原理

神经网络线性层计算本质上为矩阵乘法，表达式为

$$\boldsymbol{y}=\begin{pmatrix}y_0\\y_1\\\vdots\\y_q\end{pmatrix}=\begin{pmatrix}\sum_{m=0}^{p}W_{0m}x_m\\\sum_{m=0}^{p}W_{1m}x_m\\\vdots\\\sum_{m=0}^{p}W_{qm}x_m\end{pmatrix},\tag{1}$$

式中：$\boldsymbol{x}$为输入向量；$\boldsymbol{W}$为神经元权重矩阵；$\boldsymbol{y}$为输出值。光学神经网络的线性层由线性光学处理器组成。线性光学处理器的基本元件是2×2可重构马赫-曾德尔干涉仪。它由2个3 dB耦合器组成（3 dB耦合器可通过2×2多模干涉仪或定向耦合器实现），其中一个耦合器内臂上有一个移相器（相移用θ表示），第二个耦合器输出端有另一个移相器（相移用φ表示）。MZI传输矩阵可表示为

$$U_{\mathrm{MZI}}(\theta,\varphi)=\mathrm{i}\mathrm{e}^{\frac{\mathrm{i}\theta}{2}}\begin{pmatrix}\mathrm{e}^{\mathrm{i}\theta}\sin\frac{\theta}{2} & \cos\frac{\theta}{2}\\\mathrm{e}^{\mathrm{i}\theta}\cos\frac{\theta}{2} & -\sin\frac{\theta}{2}\end{pmatrix}。\tag{2}$$

MZI可以实现二阶特殊酉群中的任何矩阵。这些矩阵的特点是共轭转置矩阵等于其逆矩阵，且行列式等于1。这些移相器是ONN可编程性的核心。通过调节两个移相器的θ和φ，可以调整每个MZI的输出，以实现对矩阵向量乘法的模拟。

通过适当的排列方式相互连接多个MZI，可以构成N输入N输出的结构，该结构可以实现N阶任意酉矩阵，从而通过物理方式实现奇异值分解定理[18]，进而对任意矩阵进行向量矩阵乘法。图1给出了有8个输入端口的MZI的两种连接拓扑图。其中，一个椭圆形表示一个MZI。MZI连接构成线性层的一部分，输入数据通过input端口输入，output端口输出。网格型ONN中MZI只将相邻的波导混合。设N为输入个

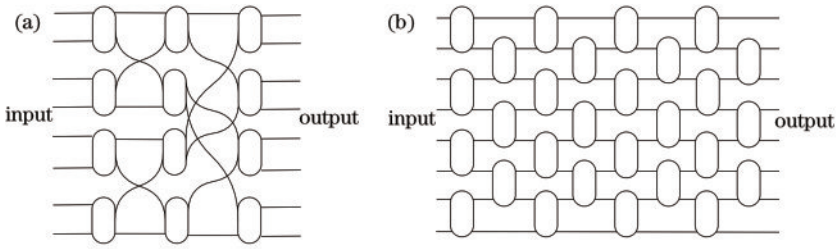

图 1　输入端口数为 8 时，MZI 的两种连接拓扑图。(a) FFT 型 ONN 的连接拓扑图；(b) 网格型 ONN 的连接拓扑图

Fig. 1　Two connection topologies of MZI when the number of input ports is 8. (a) Connection topology of FFT-typed ONN; (b) connection topology of grid-typed ONN

数，在 $P(P\leqslant N)$ 层之后，每个波导最多连接到其附近 $2P$ 个波导，而 FFT 型 ONN 可以连接 2^P 个波导。因此，与网格型 ONN 相比，FFT 型 ONN 可以将酉矩阵的深度从 N 降低到 $\log_2 N$。一般而言，减少 MZI 的数量可以降低网络的整体噪声和损耗。

通过特定的移相器配置可以实现 FFT 型网络结构。将输入表示为 $x_n\in C^N$，则它的傅里叶变换[19]为

$$X_k=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}x_n\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}nk}。\tag{3}$$

根据 FFT 算法，可将式(3)写为

$$X_k=\frac{1}{\sqrt{2}}\left(E_k+\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}nk}O_k\right),\tag{4}$$

$$X_{k+N/2}=\frac{1}{\sqrt{2}}\left(E_k-\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}nk}O_k\right),\tag{5}$$

式中：O_k 和 E_k 分别是 x_n 的奇数和偶数元素的傅里叶变换。将式(4)和式(5)写成矩阵形式，有

$$\begin{pmatrix}X_k\\X_{k+N/2}\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1&\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}nk}\\1&-\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}nk}\end{pmatrix}\begin{pmatrix}E_k\\O_k\end{pmatrix}\equiv U_k\begin{pmatrix}E_k\\O_k\end{pmatrix}。\tag{6}$$

由式(2)可知，$\boldsymbol{U}_k=\boldsymbol{U}_{\mathrm{MZI}}(\theta=\pi/2$ 和 $\varphi=2\pi k/N)$。输入数据通过 FFT 单位乘法器，在第 k 层配置 $\theta=\pi/2$ 和 $\varphi=2\pi k/N$，则可以执行 FFT。

2.2　光学神经网络整体架构

从整体上看，人工神经网络是一个输入向量 $\boldsymbol{x}$ 并返回输出向量 $\boldsymbol{y}$ 的函数，这是利用网络一层一层传递的方式实现的。ONN 流程框架如图 2 所示。需要先对手写数字图像进行数据预处理，将图像的上半部分和下半部分分别作为实部和虚部，即 $28\times28=784$ 维实值输入转换为 $392=784/2$ 维复值矢量。由于 FFT 的输入必须是 2 的 n 次方，输入不足时需补足，补充的其他输入为 0，所以网络输入个数为 512。ONN 由 2 个隐藏层和 1 个输出层组成。每一层由线性部分和非线性部分构成，先对输入向量进行线性矩阵向量乘法运算，再对结果进行非线性函数处理。网络整体为 512 输入 10 输出，线性部分中每个矩阵为 N 输入 M 输出，以 $N\times M$ 来表示。

所搭建的网络中线性乘法使用可配置的 MZI 阵列来实现。利用奇异值分解得到 $\boldsymbol{M}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}$，其中 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 是酉矩阵，$\boldsymbol{V}^{\mathrm{H}}$ 是 $\boldsymbol{V}$ 的 Hermitian 矩阵，$\boldsymbol{\Sigma}$ 是由 $\boldsymbol{M}$ 的特征值组成的对角矩阵[20]。基于奇异值矩阵分解的架构中的 MZI 阵列可以实现任意矩阵乘法而没有基本损失，这些架构也很容易配置和控制。

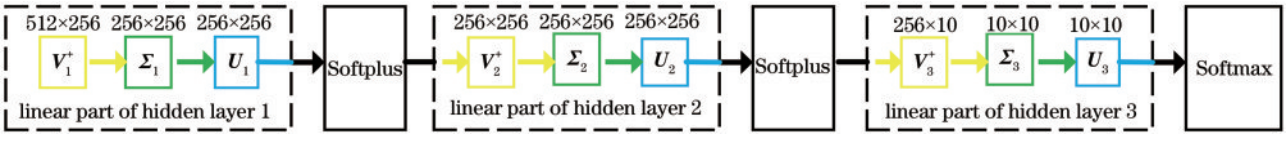

图 2　ONN 流程

Fig. 2　Flowchart of the ONN

由于线性层不可能学习和识别非线性模型，因此神经网络中的非线性是至关重要的。非线性层可以通过可饱和吸收器、石墨烯、硫化物等二维材料的强非线性效应来实现[21]。本文使用 ReLU 和 Softplus 函数模拟光学非线性函数。ReLU 函数定义为

$$R(x)=\max(0,x),\tag{7}$$

该函数的特点为计算速度快，当输入为正时，不存在梯度饱和的问题。作为 ReLU 函数的平滑版，Softplus 函数定义为

$$S(x)=\log(1+\mathrm{e}^x),\tag{8}$$

这个函数可解决在反向传播算法中用于更新隐藏层权值的梯度随着模型深度的增加而趋于消失的问题[22]。

网络的最后一步是 Softmax 函数处理，结果以识别概率的形式表示，数值范围为 [0,1]。Softmax 函数可以表示为

$$f(x_i)=\frac{\mathrm{e}^{x_i}}{\sum_{j=1}^{K}\mathrm{e}^{x_j}},i=1,2,\cdots,K,\tag{9}$$

式中：x_1，x_2，…，x_K 为 Softmax 层的输入值；输出值 $f(x_i)$ 表示样本属于第 i 类的概率[23]。

2.3　ONN前向传播和反向训练流程及主要超参数

2.3.1　计算流程

图3为所提ONN的前向传播和反向训练流程。其中，训练过程通常为最小化神经网络对一组训练示例集的预测误差，训练实例以输入和目标输出的形式表现。所提光学神经网络的训练实际上是在计算机中实现的。

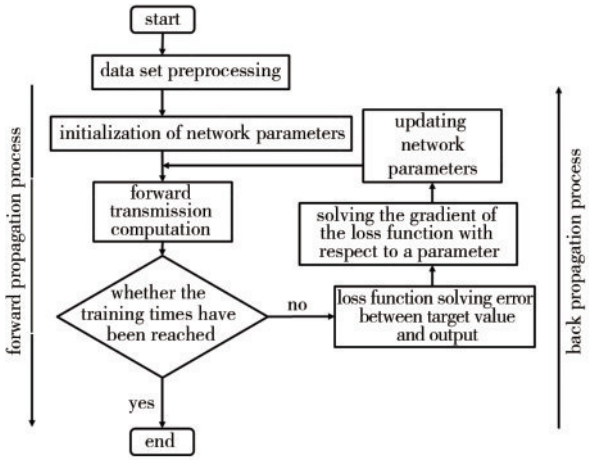

图3　ONN的前向传播和反向传播的流程

Fig. 3　Flowchart of forward propagation and backward propagation of ONN

前馈神经网络通过线性操作和向量的元素非线性函数的交替序列将输入向量映射到输出向量，也称为激活[24]。在神经网络的输出上定义一个损失函数L。通过利用损失函数优化权重的梯度，调整线性操作中涉及的矩阵元素，使L在大量训练示例上最小化。反向传播算法通常用于解析、计算这些梯度，从输出层反向到输入层依次利用链式规则[25]。

2.3.2　学习率和动量

学习率是影响网络是否收敛和收敛速度的重要因素，如果学习率设置过大，即参数更新的幅度过大，可能越过损失值最小点而在附近抖动，导致无法收敛，而如果学习率设置过小，参数更新速度慢，模型收敛速度慢，能耗增加。所以找到一个最佳的学习率将使ONN性能大大提升。文献常将学习率设置在$(10^{-6}, 1)$这个范围，所以本实验在此基础上，对FFT型ONN的学习率进行更进一步的探索。

神经元的权重更新计算公式为

$$w_t = w_{t-1} - \varepsilon \frac{\partial L(w)}{\partial w}, \tag{10}$$

式中：w_t为第t轮更新的参数；$\varepsilon>0$为学习率。

在训练算法中引入动量是一种加速梯度下降的技术，它在迭代过程中在目标值持续减小的方向上累积速度矢量。加入动量后，神经元的权重更新计算公式为

$$\begin{cases} v_t = \alpha v_{t-1} + \varepsilon \dfrac{\partial L(w)}{\partial w} \\ w_t = w_{t-1} - v_t \end{cases}, \tag{11}$$

式中：v_t为当前学习速度；$\alpha\in[0,1]$是动量系数。

2.3.3　反向传播算法

在反向传播中，采用4种算法来实现梯度计算。SGD算法通过执行一次更新来消除批量梯度下降对大型数据集计算时产生的冗余，因此它通常更新参数速度更快，也可以用于在线学习，但是SGD以高方差执行频繁更新，可能导致目标函数剧烈波动。Adagrad是一种基于梯度的优化算法，可以根据参数调整学习速度，对不频繁更新的参数执行较大的更新，对频繁更新的参数进行较小的更新，因此它非常适合处理稀疏数据。RMSprop算法可以自适应学习率，用于解决Adagrad算法学习率急剧下降的问题。Adam算法是计算每个参数的自适应学习率的另一种方法，可以记录过往梯度与当前梯度的平均，使每一次更新的梯度与过往的梯度不会相差太大，使梯度可以保持平滑、稳定的过渡，可以适应不稳定的目标函数。

3　超参数对训练效率的影响

3.1　测试流程

将MNIST数据集中60000张训练集图片（每张图片都有对应的标签值）分成了600个批次进行输入，同时设置学习率为0.05，然后按照图3的步骤进行500次训练。

测试学习率对ONN训练效率的影响：在每组实验中，分别改变SGD、RMSprop、Adam和Adagrad 4种算法的学习率，分析学习率对不同非线性函数、具有不同隐藏层个数的ONN的识别精确度、训练运行内存和训练时间3个方面的影响。

测试动量参数对ONN训练效率的影响：在每组实验中更换嵌入动量的SGD和RMSprop算法的动量系数，分析动量对不同非线性函数、具有不同隐藏层个数的ONN识别MNIST手写数字图像的影响，主要分析识别的精确性、训练过程中运行内存占用量和平均训练时长等方面。

3.2　测试环境

本实验采用的硬件和软件仿真环境如表1所示。

表1　实验平台参数

Table 1　Experimental platform parameters

Operating system	Windows7 64 bit
CPU	Intel(R) Core(TM) i5-5200U CPU @2.20 GHz
GPU	GeForce 920M
Software platform	PyTorch1.1.0

3.3　结果与分析

首先比较了不同学习率(R)下采用SGD、RMSprop、Adam和Adagrad 4种算法训练后的ONN的识别精确度，结果如表2所示，此时使用Softplus为非线性函数。从表2可以看出：采用SGD算法，在学习率为0.05和

表2 不同学习率的4种训练算法下ONN的识别精确度（Softplus作为非线性函数）

Table 2 Accuracy of ONN with four training algorithms under different learning rates (Softplus as the nonlinear function)

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	0.091	0.960	0.941	0.879	0.459
RMSprop	0.101	0.887	0.974	0.959	0.937
Adam	0.101	0.919	0.973	0.960	0.942
Adagrad	0.965	0.960	0.942	0.834	0.246

0.005时，ONN识别精确度达0.940以上，当学习率为0.05时，ONN识别精确度最高，为0.960，当学习率为0.5时，ONN不收敛；采用RMSprop算法，在学习率为0.005时，ONN识别精确度最高，为0.974，在学习率为5×10^{-4}和5×10^{-5}时，ONN识别精确度也达到了0.930以上；采用Adam算法，在4个学习率下，ONN识别精确度都达到了0.910以上，学习率为0.005时，ONN识别准确度是最高的，为0.973；采用Adagrad算法，在学习率为0.5时，ONN识别精确度最高，为0.965，学习率为0.05和0.005时，ONN的识别精确度达到0.942以上；特别地，当学习率为0.005时，采用各算法训练后的ONN的识别精确度均可以达到0.940以上。非线性函数不同对ONN的识别精确度也会有影响。表3给出了4种训练算法下使用ReLU为非线性函数的ONN的识别精确度。对比表2和表3可以看出：使用Softplus为非线性函数的ONN的识别精确度普遍较高；使用ReLU为非线性函数时，4种算法下的ONN最大识别精确度为0.941，小于使用Softplus为非线性函数时4种算法下的ONN的最大识别精确度（0.974）。

表3 不同学习率的4种训练算法下ONN的识别精确度（ReLU作为非线性函数）

Table 3 Accuracy of ONN with four training algorithms under different learning rates (ReLU as the nonlinear function)

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	0.021	0.911	0.853	0.651	0.435
RMSprop	0.109	0.919	0.935	0.901	0.671
Adam	0.069	0.894	0.941	0.712	0.591
Adagrad	0.934	0.910	0.644	0.264	0.132

进一步，探究隐藏层个数对ONN识别精确度的影响。表4为4种训练算法下使用Softplus为非线性函数时不含隐藏层2的ONN的识别精确度。对比表2和表4可以看出：在学习率为0.005时，ONN性能表现均较好；含2个隐藏层时4种算法下ONN最大的识别精确度为0.974，高于只有1个隐藏层时4种算法下ONN的最大识别精确度（0.961）。表5给出了4种训练算法下使用ReLU为非线性函数的不含隐藏层2的

表4 不同学习率的4种训练算法下仅有一个隐藏层的ONN的识别精确度（Softplus作为非线性函数）

Table 4 Accuracy of ONN with four training algorithms without hidden layer 2 under different learning rates (Softplus as the nonlinear function)

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	0.948	0.904	0.689	0.304	0.135
RMSprop	0.202	0.927	0.956	0.907	0.648
Adam	0.852	0.944	0.950	0.882	0.558
Adagrad	0.961	0.939	0.824	0.288	0.079

表5 不同学习率的4种训练算法下仅有一个隐藏层的ONN的识别精确度（ReLU作为非线性函数）

Table 5 Accuracy of ONN with four training algorithms without hidden layer 2 under different learning rates (ReLU as the nonlinear function)

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	0.098	0.905	0.814	0.539	0.290
RMSprop	0.168	0.924	0.933	0.904	0.565
Adam	0.837	0.930	0.926	0.863	0.463
Adagrad	0.923	0.916	0.745	0.218	0.162

ONN的识别精确度。对比表3和表5可以看出：在学习率为0.05时，ONN识别精确度均大于0.894；含2个隐藏层时4种算法下ONN最大的识别精确度为0.941，高于只有1个隐藏层时4种算法下ONN的最大识别精确度（0.933）。

表6和表7分别为隐藏层个数不同的ONN的训练运行内存和训练时间。对于训练运行内存和训练时间两个评价指标，含2个隐藏层的ONN是只有1个隐藏层的ONN的2倍左右。因为减少ONN隐藏层个数后，神经元个数减少，网络规模减小，每次迭代的计算量变少，所以训练内存消耗减小、训练时间变短。

表6 学习率为0.05时4种训练算法下ONN的训练运行内存（Softplus作为非线性函数）

Table 6 Running memory of ONN with four training algorithms when the learning rate is 0.05 (Softplus as the nonlinear function) unit: kB

Algorithm	Two hidden layers	One hidden layer
SGD	668.124	353.919
RMSprop	684.356	333.144
Adam	709.480	330.339
Adagrad	717.371	336.585

此外，不同学习率的4种训练算法下具有2个隐藏层的ONN的训练运行内存基本维持在650～750 kB区间内，含1个隐藏层的ONN的训练运行内存基本维持在330～380 kB区间内，如表8和表9所示。因为每次输入网络的图片是随机选取的，网络对每张图像的

表 7 学习率为 0.05 时 4 种训练算法下 ONN 的训练时间（Softplus 作为非线性函数）

Table 7 Training time of ONN with four training algorithms when the learning rate is 0.05 (Softplus as the nonlinear function) unit: ms

Algorithm	Two hidden layers	One hidden layer
SGD	449.995	183.853
RMSprop	450.083	186.815
Adam	447.009	156.496
Adagrad	436.669	198.566

表 8 不同学习率的 4 种训练算法下 ONN 的训练运行内存

Table 8 Running memory of ONN with four training algorithms under different learning rates unit: kB

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	671.940	688.124	696.836	731.680	736.724
RMSprop	720.472	684.356	656.360	726.204	732.252
Adam	728.012	709.480	712.940	717.196	740.156
Adagrad	709.597	717.371	709.236	723.026	735.144

表 9 不同学习率的 4 种训练算法下不含隐藏层 2 的 ONN 的运行内存

Table 9 Running memory of ONN with four training algorithms without hidden layer 2 under different learning rates unit: kB

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	373.418	353.919	344.307	376.671	350.531
RMSprop	339.749	333.144	367.170	363.365	333.895
Adam	348.659	330.339	369.867	378.350	339.490
Adagrad	359.918	336.585	373.809	366.658	369.934

识别能力也是随机的，与算法之间没有明显的关联。相似地，不同学习率的 4 种训练算法下的 ONN 的训练时间基本维持在 350～460 ms 区间内，不同学习率的 4 种训练算法下含 1 个隐藏层的 ONN 的训练时间基本维持在 140～220 ms 区间内，如表 10 和表 11 所示。这是因为，尽管网络对每张图像的学习时间是不一样的，但单批次输入到网络中的图像是随机的，所以训练时间基本稳定维持在一个区间内。

表 10 不同学习率的 4 种训练算法下的 ONN 的训练时间

Table 10 Training time of ONN with four training algorithms under different learning rates unit: ms

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	453.210	449.995	385.783	366.278	372.436
RMSprop	398.588	450.083	377.086	445.179	388.789
Adam	439.016	447.009	430.769	397.805	423.890
Adagrad	351.312	436.669	424.947	400.590	426.575

表 11 不同学习率的 4 种训练算法下不含隐藏层 2 的 ONN 的训练时间

Table 11 Training time of ONN with four training algorithms without hidden layer 2 under different learning rates unit: ms

Algorithm	R=0.5	R=0.05	R=0.005	R=5×10^{-4}	R=5×10^{-5}
SGD	161.840	183.853	199.215	192.506	158.501
RMSprop	172.940	186.815	193.341	177.395	153.627
Adam	173.191	156.496	155.703	146.066	170.774
Adagrad	165.250	198.566	170.376	185.795	190.148

除了学习率，算法中的动量系数也是影响 ONN 性能的重要因素。首先比较在不同动量系数的 SGD 算法下 ONN 的识别精确度、训练时间和训练运行内存，如表 12 和图 4 所示。在动量系数为 0～0.97 时，

表 12 学习率为 0.05 时不同动量系数的 SGD 算法下的 ONN 的识别精度、训练运行内存、训练时间

Table 12 Accuracy, running memory, and training time of ONN with the SGD algorithm under different momentum coefficients when the learning rate is 0.05

Momentum coefficient	Accuracy	Running memory /kB	Training time /ms
0	0.960	668.124	449.995
0.1	0.962	602.880	380.814
0.5	0.964	614.468	402.444
0.9	0.969	625.308	403.404
0.97	0.943	627.606	394.597
0.98	0.099	645.633	409.608
1.0	0.098	633.696	396.202

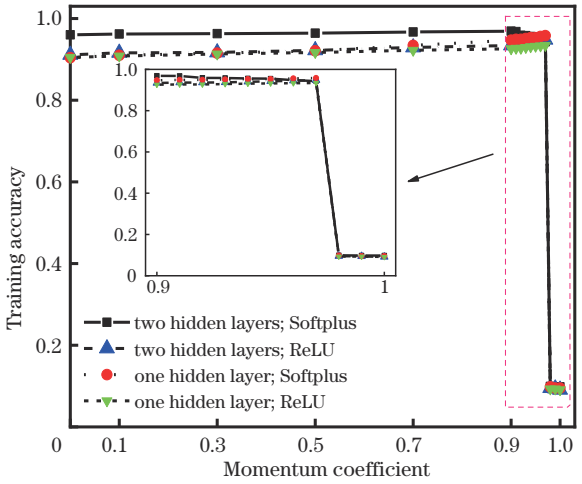

图 4 学习率为 0.05 时不同动量系数的 SGD 算法下具有不同非线性函数和不同隐藏层个数的 ONN 的识别精度

Fig. 4 Accuracy of ONN with the SGD algorithm under different momentum coefficients with different nonlinear functions and different number of hidden layers when the learning rate is 0.05

SGD算法下的ONN识别精确度均在0.930以上；在动量系数为0.9时，含2个隐藏层和Softplus作为非线性函数的ONN的识别精确度达到最高，为0.969；在动量系数为0.97时，含2个隐藏层和ReLU作为非线性函数的ONN与含1个隐藏层和Softplus作为非线性函数的ONN的识别精确度达到最高，分别为0.948和0.958；在动量系数为0.97时，不含隐藏层2和ReLU作为非线性函数的ONN的识别精确度达到最高，为0.934；当动量系数为0.98～1时，因为动量系数过大，网络无法识别图像特征，使得识别精确度突降。

此外，加入动量的SGD算法下ONN的训练运行内存为602～650 kB，均小于未加动量的情况，未加动量下的ONN的相关参数如表12所示。而且，加入动量的SGD算法下ONN的训练时间为380～410 ms，均小于未加动量的情况。

进一步，为了对比ONN在SGD、RMSprop训练算法特别是各自在引入动量后的识别性能，给出在0.05和0.005学习率下两种训练算法及其加入0.9的动量系数时ONN的收敛过程，结果如图5和图6所示。可以看到：在这两个学习率下，采用有或无动量系数的SGD训练算法的ONN的训练过程都收敛；但是，对于RMSprop训练算法，动量的引入对ONN的识别性能影响较大。首先，对于没有加入动量的RMSprop训练算法，当学习率为0.05时，在Epoch（使用训练集全部样本训练网络的次数）为320～360之间，ONN的识别准确度突增，如图5所示。这是由于初始化后神经元的权重较小，且学习率设置过大，经历了比较多的训练次数后ONN的训练过程才慢慢收敛。而当学习率为0.005时，ONN的训练过程很快收敛，如图6所示。其次，对于加入动量的RMSprop训练算法，当学习率为0.05时，ONN学习数据步长太大再加上引入动量会累积之前的梯度，从而导致梯度发生振荡的现象加剧[23]，导致ONN的训练过程并不收敛，如图5所示；当

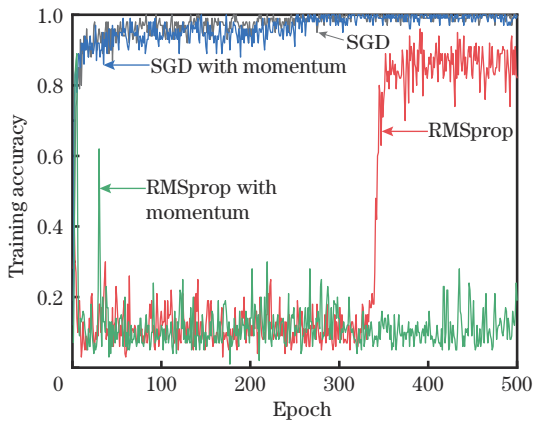

图5　学习率为0.05时不同训练算法下的ONN的识别精确度随训练次数增加的变化

Fig. 5　Variation in the accuracy of ONN with different training algorithms with the epoch when the learning rate is 0.05

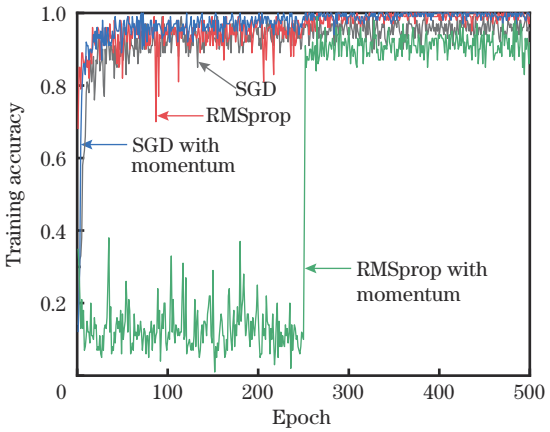

图6　学习率为0.005时不同训练算法下的ONN的识别精确度随训练次数增加的变化

Fig. 6　Variation in the accuracy of ONN with different training algorithms with the epoch when the learning rate is 0.005

学习率减小为0.005时，经历了较长的迭代后ONN的训练过程才收敛，如图6所示。这表明，在RMSprop训练算法中引入动量反而降低了ONN的识别性能。

4　结　　论

使用光学器件构建了基于FFT的ONN，并研究了SGD、RMSprop、Adam和Adagrad 4种训练算法中的主要超参数即动量系数和学习率对具有不同非线性函数、不同隐藏层个数的ONN的手写数字图像识别性能的影响。首先解释了MZI型矩阵向量乘法和所搭建的ONN整体架构，给出了神经元的权重更新计算公式，并分析了学习率和动量系数的作用；然后，分别给出了不同学习率下具有不同非线性函数、不同隐藏层个数的ONN的识别精确度、运行内存和训练时间3个评价指标的模拟实验数值结果。实验结果表明：不同学习率或动量系数对运行内存和训练时间影响不大；用学习率过大的SGD、RMSprop、Adam算法训练网络会导致ONN识别过程不收敛，而用学习率过小的Adagrad算法的ONN的识别过程也不收敛；学习率为0.005时的RMSprop算法表现最好，ONN识别准确率高达97.4%；ReLU作为非线性函数和不含隐藏层2的ONN的识别精确度均不高。此外，着重分析了在不同动量系数的SGD算法下ONN对手写数字图像识别的准确率、运行内存和训练时间，并比较了在引入动量的SGD、RMSprop训练算法下ONN的识别性能。结果显示：SGD算法在有或无动量系数时，训练结果均较好，尤其在动量系数为0.9时，ONN的准确率可达96.9%；与之相反，采用不加入动量系数的RMSprop算法的ONN性能比采用加入动量系数的RMSprop算法的ONN性能要好。因此，在RMSprop训练算法中引入动量反而降低了ONN的识别性能。

参 考 文 献

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

[2] 林梦翔, 黄秀萍, 林志玮, 等. 基于卷积神经网络融合编码与解码特征的降水强度识别[J]. 激光与光电子学进展, 2023, 60(2): 0211003.
Lin M X, Huang X P, Lin Z W, et al. Precipitation intensity recognition was based on a convolution neural network with fused encoded and decoded features[J]. Laser & Optoelectronics Progress, 2023, 60(2): 0211003.

[3] Leydig C, Theis L, Huszár F, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 105-114.

[4] 亢超, 李文祥, 黄岫, 等. 基于深度学习的主动光学校正算法研究[J]. 光学学报, 2021, 41(6): 0611004.
Kang C, Li W X, Huang S, et al. Research on an active optical correction algorithms based on deep learning[J]. Acta Optica Sinica, 2021, 41(6): 0611004.

[5] 王通, 董文德, 沈康, 等. 基于改进的U-Net神经网络的稀疏视角光声图像质量增强方法[J]. 激光与光电子学进展, 2022, 59(6): 0617022.
Wang T, Dong W D, Shen K, et al. Sparse-view photoacoustic image quality enhancement based on a modified U-Net[J]. Laser & Optoelectronics Progress, 2022, 59(6): 0617022.

[6] de Lima T F, Peng H T, Tait A N, et al. Machine learning with neuromorphic photonics[J]. Journal of Lightwave Technology, 2019, 37(5): 1515-1534.

[7] Zhang Q M, Yu H Y, Barbiero M, et al. Artificial neural networks enabled by nanophotonics[J]. Light: Science & Applications, 2019, 8(1): 1-14.

[8] Xu X Y, Tan M X, Corcoran B, et al. 11 TOPS photonic convolutional accelerator for optical neural networks[J]. Nature, 2021, 589(7840): 44-51.

[9] Lin X, Rivenson Y, Yardimci N T, et al. All-optical machine learning using diffractive deep neural networks [J]. Science, 2018, 361(6406): 1004-1008.

[10] Zhao Y H, Wang X, Gao D S, et al. On-chip programmable pulse processor employing a cascaded MZI -MRR structure[J]. Frontiers of Optoelectronics, 2019, 12(2): 148-156.

[11] Wetzstein G, Ozcan A, Gigan S, et al. Inferences in artificial intelligence with deep optics and photonics[J]. Nature, 2020, 588(7836): 39-47.

[12] Shen Y C, Harris N C, Skirlo S, et al. Deep learning with coherent nanophotonic circuits[J]. Nature Photonics, 2017, 11(7): 441-446.

[13] Hughes T W, Momchil M, Yu S, et al. Training of photonic neural networks through in situ backpropagation [J]. Optica, 2018, 5(7): 864-871.

[14] Zhang T, Wang J, Dan Y H, et al. Efficient training and design of a photonic neural networks through neuroevolution[J]. Optics Express, 2019, 27(26): 37150-37163.

[15] Zang Y B, Chen M H, Yang S G, et al. Electro-optical neural networks based on time-stretch method[J]. IEEE Journal of Selected Topics in Quantum Electronics, 2020, 26(1): 7701410.

[16] Alvear-Sandoval R F, Sancho-Gómez J L, Figueiras-Vidal A R. On improving CNN performance: the case of MNIST[J]. Information Fusion, 2019, 52: 106-109.

[17] Fang M Y S, Manipatruni S, Wierzynski C, et al. Design of optical neural networks with component imprecisions[J]. Optics Express, 2019, 27(10): 14009-14029.

[18] Shokraneh F, Nezami M S, Liboiron-Ladouceur O. Theoretical and experimental analysis of a 4×4 reconfigurable MZI-based linear optical processor[J]. Journal of Lightwave Technology, 2020, 38(6): 1258-1267.

[19] Gu J Q, Zhao Z, Feng C H, et al. Toward area-efficient optical neural networks: an FFT-based architecture[C]// 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), January 13-16, 2020, Beijing, China. New York: IEEE Press, 2020: 476-481.

[20] Clements W R, Humphreys P C, Metcalf B J, et al. Optimal design for universal multiport interferometers[J]. Optica, 2016, 3(12): 1460-1465.

[21] Bao Q L, Zhang H, Ni Z H, et al. Monolayer graphene as a saturable absorber in a mode-locked laser[J]. Nano Research, 2011, 4(3): 297-307.

[22] Zheng H, Yang Z L, Liu W J, et al. Improving deep neural networks using softplus units[C]//2015 International Joint Conference on Neural Networks (IJCNN), July 12-17, 2015, Killarney. New York: IEEE Press, 2015.

[23] Zhu D Y, Lu S Y, Wang M Q, et al. Efficient precision-adjustable architecture for softmax function in deep learning[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2020, 67(12): 3382-3386.

[24] Shastri B J, Tait A N, de Lima T F, et al. Photonics for artificial intelligence and neuromorphic computing[J]. Nature Photonics, 2021, 15(2): 102-114.

[25] Okuma N, Sato M. Non-Hermitian topological phenomena: a review[J]. Annual Review of Condensed Matter Physics, 2023, 14: 83-107.