

# 文本复制检测报告单(全文标明引文)

№:BC202005271027391054080957

检测时间:2020-05-27 10:27:39

检测文献: 基于深度学习的图像分类研究

作者: 匡乐

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

CNKI大成编客-原创作品库

时间范围: 1900-01-01至2020-05-27

## 检测结果

总文字复制比: 2.6%

跨语言检测结果: 0%

去除引用文献复制比: 2.6%

去除本人已发表文献复制比: 2.6%

单篇最大文字复制比: 0.9% ( 7078708\_廖泽宇\_基于tensorflow的卷积神经网络实现 )

重复字数: [706] 总段落数: [3]

总字数: [27111] 疑似段落数: [2]

单篇最大重复字数: [257] 前部重合字数: [0]

疑似段落最大重合字数: [413] 后部重合字数: [706]

疑似段落最小重合字数: [293]



指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似自我剽窃 ☐ 一稿多投 ☐ 疑似整体剽窃 ☐ 过度引用 ☐ 重复发表

表格: 0 公式: - 疑似文字的图片: 0 脚注与尾注: 0

4.3% ( 413 ) 基于深度学习的图像分类研究\_第1部分 ( 总9664字 )

3.2% ( 293 ) 基于深度学习的图像分类研究\_第2部分 ( 总9232字 )

0% ( 0 ) 基于深度学习的图像分类研究\_第3部分 ( 总8215字 )

( 注释: 无问题部分 文字复制比部分 引用部分 )

## 1. 基于深度学习的图像分类研究\_第1部分

总字数: 9664

相似文献列表 文字复制比: 4.3%(413) 疑似剽窃观点: (0)

1	7078708_廖泽宇_基于tensorflow的卷积神经网络实现 廖泽宇 - 《高职高专院校联合比对库》 - 2019-05-21	2.7% ( 257 ) 是否引证: 否
2	机动学院-10159010-王京 机动学院 - 《大学生论文联合比对库》 - 2019-06-06	1.4% ( 136 ) 是否引证: 否
3	随机梯度定位算法及其基于Zynq7000的模拟 刘闯 - 《大学生论文联合比对库》 - 2018-05-20	0.6% ( 59 ) 是否引证: 否

## 原文内容

毕业设计说明书

学生姓名: 匡乐学号: 20164484

学院：计算机与信息工程学院  
专业年级：2016级计算机与科学技术  
题目：基于深度学习的图像分类研究  
指导教师：周慧斌讲师  
评阅教师：杨卫民副教授  
2020年5月

## 摘要

深度学习目前被广泛的应用到生活生产当中，对人们的生活方式和生产发展有着非常积极的作用。神经网络的发展还是在21世纪后才有了飞速的发展，这离不开硬件技术的成熟，极大提高了计算机计算能力，使得规模巨大，深层的神经网络训练成为了可能。而计算机视觉方向是被证明实践过运用深度卷积神经网络取得了非常成功的效果，应运而生了很多的新奇的产品，面部识别和自动驾驶都离不开神经网络技术的支持。随着技术的发展，网络的规模和层数变得更深更大，训练难度也加剧，其中卷积神经网络因为有着非常高效的局部连接和参数共享的优势，得到了广泛的应用。

本毕业设计就卷积神经网络最流行的几种网络架构进行模拟研究，分析卷积神经网络各种衍生网络并且运用到计算机视觉方向——图像分类识别，评判形式包括了训练集性能比对，泛化性能比对，以及通过参加kaggle下CIFAR-10 - Object Recognition in Images竞赛评判。

经过实验比对分析，调整网络结构，采用正则化和归一化等手段能够显著的影响模型在数据集中的表现，也能提高其泛化能力。

关键词：深度学习；图像分类；卷积神经网络；CIFAR-10；kaggle竞赛

Title 基于深度学习的图像分类研究

Abstract:

At present, deep learning is widely used in life and production, and it has a very positive effect on people's lifestyle and production development. The development of deep neural networks has only developed rapidly after the 21st century. This is inseparable from the maturity of hardware technology, Greatly improve computer computing power. It makes the scale huge and deep neural network training becomes possible. The direction of computer vision is proved to have achieved very successful results by using deep convolutional neural networks. Afterwards, a lot of novel products were produced. Similarly, facial recognition and autonomous driving have traces of deep neural networks. With the development of technology, the scale and layers of the network have become deeper and larger. The difficulty of training has also increased. Because of the advantages of very efficient local connection and parameter sharing, convolutional neural networks have been widely used.

This graduation project simulates the most popular network architectures of convolutional neural networks. Analyze various derivative networks of convolutional neural networks and apply them to the direction of computer vision, mainly image classification and recognition. The evaluation form includes comparison training set performance and generalization performance. There is also the CIFAR-10-Object Recognition in Images competition held on the kaggle platform.

Through experimental comparison and analysis, adjusting the network structure, and adopting means such as regularization and normalization can significantly affect the performance of the model in the data set, and can also improve its generalization ability.

Keywords: Deep learning; image classification; convolutional neural network; CIFAR-10;Kaggle;

## 目录

摘要.....	1
1 引言 .....	6
1.1 图像识别背景意义 .....	6
1.2 卷积神经网络背景发展 .....	6
2 神经网络基础 .....	7
2.1 前馈神经网络.....	7
2.2 深度神经网络的激活函数.....	7
2.2.1 Sigmoid 函数.....	8
2.2.2 双曲正切函数 tanh .....	9
2.2.3 线性整流函数 ( Rectified Linear Unit ) .....	10
2.3 反向传播算法.....	12
2.3 常见优化算法 .....	14
2.4.1 标准梯度下降算法.....	15
2.4.2 批量梯度下降法(Batch Gradient Descent).....	15
2.4.3 随机梯度下降法(Stochastic Gradient Descent).....	16
2.4.4 动量优化法.....	17
2.4.5 AdaGrad算法.....	17
2.4.6 RMSProp算法.....	18
2.4.7 Adam算法.....	18
3 卷积神经网络相关理论.....	19
3.2 重要影响力网络 .....	19
3.3 卷积神经网络组成 .....	21

3.3.1 卷积层 .....	21
3.3.2 池化层 .....	23
3.3.3 全连接层 .....	23
4 解决过拟合 .....	23
4.1 正则化 .....	23
4.1.1 L1和L2正则化 .....	24
4.1.2 Dropout正则化 .....	24
4.2 Mini-batch .....	26
4.3 归一化 .....	26
4.3.1 Batch Normalization .....	27
4.3.2 Weight Normalization .....	27
4.3.3 Layer Normalization .....	27
5 相关算法及数据集介绍 .....	27
5.1 网络结构 .....	28
5.1.1 VGG .....	28
5.1.1 VGG-16特点 .....	29
5.1.2 块结构 .....	29
5.2 模型修改 .....	30
5.2.1 输入层 .....	30
5.2.2 模型深度 .....	30
5.2.3 Keras快速建立模型 .....	30
5.3 数据增强 .....	31
5.3.1 常用技术 .....	31
6 卷积神经网络应用 .....	33
6.1 图像预处理 .....	34
6.1.1 归一化 .....	34
6.1.2 独热编码 .....	34
6.1.3 选择mini-batch大小 .....	35
6.1.4 数据增强 .....	35
6.2 优化及参数设定 .....	35
6.2.1 学习率 .....	36
6.2.2 训练轮次 .....	36
6.3 实验环境 .....	37
6.4 实验数据分析 .....	37
6.4.1 整体训练过程 .....	38
6.4.2 初步尝试 .....	39
6.4.3 调整学习率 .....	39
6.4.4 使用dropout .....	40
6.4.5 使用图像增强 .....	41
6.4.6 L2正则化解决过拟合问题 .....	42
6.4.7 结果分析 .....	42
结语 .....	44
参考文献 .....	47

## 1 引言

### 1.1 图像识别背景意义

社会和科技的发展，图像载体的信息在人们生活中的占比越来越高。面对大量图像数据，人们有很多的需求，例如对图像中的物体进行分类识别，这能极大的提高人们对有用图像的提取效率。人工的对图像分类耗时耗力，但是，如果利用计算机处理快速数据并且不知疲倦的特点，那么可以极大提高生产效率。图像作为一种非结构性数据，对传统计算机算法具有很大的专业性要求，很多的传统处理方法只能应对有针对性的图像，泛化不强，并且要求技术人员有较为专业的图形分析处理能力。

图像分类研究的是从各种图像中提取有价值的信息，根据信息特点分类。图像分类系统的流程可以包括：原始图像采集和数字化，预处理图像，对特征提取和分类。图像算法体现在除了图像采集和数字化的其他步骤中。

### 1.2 卷积神经网络背景发展

普遍认为卷积神经网络的思想源于生物体中的视觉处理单元。在对生物的研究过程中，发现大脑视觉皮层中的神经元只对视觉范围内的少部分区域的物体有反应，称之为感受野。还发现不同的大脑细胞对不同的图像特点有不一样的敏感度，有的对直线边缘反应很强烈，有的感受野巨大，但对边缘不敏感。受此影响的启发，科学家提出了感知机。感知机采用网络级联结构。这很类似后来的卷积层。

卷积神经网络的有效性证明来自于一个重要的算法，误差反向传播算法（Back Propagation），该算法能够行而有效的训练神经网络，使得神经网络可以拟合复杂的非线性映射。证明了神经网络的理论正确性。

最早提出的实用卷积神经网络1998年的LeNet-5，由Yann LeCun等人发表论文提出，在论文中使用LeNet-5卷积神经网络进行邮件上的手写数字识别任务，在数据集中达到了其他传统机器学习算法未有准确度。

在神经网络发展的历程中，大家一致认为，模型的能力上限很大程度上受到网络深度和规模影响，笼统的说，越大越深的网

络，能提取到的特征越丰富。

卷积神经网络发展初期受到硬件的限制，层次浅，参数少，无法处理高分辨率图像。如今随着算法和硬件技术发展，超过百层，参数达到上亿规模的网络都已经诞生。

## 2 神经网络基础

### 2.1 前馈神经网络

深度前馈网络 ( deep feedforward network )，有很多别名，前馈神经网络，多层感知机等等。从名字可以看出，这种网络含有多个隐藏层，具有一定的深度，网络的结构复杂也标志着理想拟合能力的上限高，但不是越多层次的网络就越好，有关这方面后面会讨论到。前馈的意义是每层神经元，从上一层的输出单方向接受信号，又向下一层单方向输出信号，网络属于无环结构。

前馈神经网络的意义是拟合目标函数。在最简单的线性函数拟合任务中，最终的函数是简单的，只有两个参数需要学习， $w$ 和 $b$ ，在神经网络中的体现就是，具有一个神经元，接收一个输入 $x$ ，神经元具有参数 $w$ 和 $b$ ，初始时两个参数值是随机值，训练的使得 $w$ 和 $b$ 等于真实的值，训练过程就是不断喂入数据，通过梯度下降算法策略的改变 $w$ 和 $b$ 。

经过不断地发展和变化，前馈神经网络的结构如今已经趋于统一，那就是单层输入，单层输出，中间包含了若干的隐藏层，这样的结构类似于生物神经结构中神经元突触连接。经过时间的校验，这样的结构被证明拟合能力强，便于训练。在此基础上，科学家们又研究探讨出了非常多的变体，进一步复杂了网络，从改善网络训练难度，网络训练速度以及网络的信息承载能力。

### 2.2 深度神经网络的激活函数

在没有激活函数的历史中，神经网络缺少实际意义，因为神经网络的参数结构意味值只能拟合类似的函数，不同的网络 $W$ 的数目以及 $W$ 和 $b$ 值不一样，即使经过了多层连接，但终归是线性函数，而实际应用中，非线性情况非常多，例如最简单的异或门逻辑，多复杂的线性函数都无法很好的拟合该函数。而在每一层的最后（除最后的输出层），也就是层与层之间插入激活函数，可以尽可能小代价的条件下，拟合非线性函数。可以从数学角度分析出，多层传递中，只要有一层能够拟合非线性函数，那么整个网络就具有了拟合非线性函数的能力。

简单地说，激活函数就是加入到人工神经网络中的一个函数，目的在于帮助神经网络从数据中学习复杂模式。相比于人类大脑中基于神经元的模型，激活函数是决定向下一个神经元传递何种信息的单元，这也正是激活函数在人工神经网络中的作用。激活函数接收前一个单元输出的信号，并将其转换成某种可以被下一个单元接收的形式。

目前使用的各种非线性激活函数：

#### 2.2.1 Sigmoid 函数函数如下图所示：

##### 图2.1 sigmoid函数图像

sigmoid函数在过往的神经网络中用于隐藏层的输出，值域是 $(0,1)$ 。函数特点是导数值可以由函数值简单表示，因此在训练计算时简单。函数在 $x=0$ 附近的梯度较大，便于梯度下降算法的进行。函数的基本特点如下：

定义域为 $(-\infty, +\infty)$

值域为 $(-1,1)$ ，为有界函数

函数 $f(x)$ 是连续可导函数

函数 $f(x)$ 的导数为 不定积分为 $C$ 为常数

这种激活函数从未在实际模型中使用，实际使用效果不佳，几乎在任何任务中效果都低于常用的ReLU函数。Sigmoid 函数计算量大，在 $x$ 值远离0的时候会导致梯度消失问题，并且梯度值的均值不以零为中心，通常在二进制分类问题中才会使用。

#### 1.1.2 双曲正切函数 tanh

函数如下图所示：

##### 图2.2 tanh函数图像

双曲正切函数的图像如上，它是双曲正弦函数 $\sinh x$ 与双曲余弦函数 $\cosh x$ 的商。在深度学习中， $\tanh x$ 函数使用频率不高。值域范围也是 $(-1, 1)$ ，在图像上和sigmoid函数有些类似，但是0附近的梯度要比sigmoid函数更大，意味值训练时变化更快。

相比于 Sigmoid 函数，它的值域均值关于0中心对称，有些许进步。但是和ReLU函数相比，由于幂运算的问题， $\tanh$ 函数计算量较大，并且和sigmoid函数一样， $x$ 值远离0值时，梯度值几乎为0，无法训练，存在着梯度消失的问题。

#### 1.1.3 线性整流函数 ( Rectified Linear Unit ) 函数如下图所示：

##### 图2.3 ReLU函数图像

该函数又称修正线性单元。

优点就是：

(1)  $x>0$  时，梯度恒为1，不论 $x$ 值多大都能够以较快的速度训练收敛，解决了梯度消失问题；

(2) 增大了网络的稀疏性。在 $x$ 为负数时，梯度为0，但输出值也是0，意味值虽然无法训练，但该神经元对下一层的输出是0，不贡献任何的值。这代表神经元关闭，减少了网络中神经元的数量。这一点和正则化以及dorpout不谋而合。在训练集表现相同的多个网络，稀疏的网络往往泛化性能会更加优秀。

(3) 函数非常简单，运算花费小，对训练速度影响微乎其微。

这是一个使用十分广泛的激活函数，尤其是在卷积神经网络中。它计算简单，不会饱和，不产生梯度消失问题，但不以零为中心。它也存在dying ReLU问题，过多的节点 $x$ 是负数使得节点不参与训练，整个网络可能训练速度收到严重影响，甚至无法拟合到最真实值。这即是一个问题也是一个优点，在适当的时候能够增加稀疏性，但是过多节点的休眠却成了弊病。

#### 1.1.4 Leaky ReLU 和 Parametric ReLU 函数函数如下图所示：

##### 图2.4 Leaky ReLU函数图像

针对ReLU的缺点，学者提出了用改进的Leaky ReLU取代普通的ReLU函数。图像中的 $\alpha$ 设置为了0.01，这是一个超参数，可以在训练时候，根据实际情况手动改。它的意义在于能够使得所有的节点都能够得到训练，在 $\alpha$ 较大时，网络较为稠密，训练速度更快； $\alpha$ 接近0时，激活函数退化为ReLU，网络整体变得稀疏。如果 $\alpha$ 设为1，那么 Leaky ReLU 退化为线性函数

，不在具有激活函数的意义。

因此， $\alpha$ 的值永远不会设置为趋近于1的值。如果我们分别将 $\alpha$ 设为每个神经元的超参数，则得到 Parametric ReLU 或 PReLU。

### 2.3 反向传播算法

图2.5 3层神经网络

反向传播算法 (Back propagation Algorithm, 简称BP算法) 在整个神经网络的发展中具有跨时代的意义，在反向传播算法提出之前，神经网络根本无法训练。它的提出，使得网络参数能够以自动的方式改变，按链反向传播，改变参数。

反向传播过程，需要存储和计算梯度，梯度的计算依赖于梯度下降算法 (Gradient descent)，梯度下降算法帮助误差函数逼近最小值，优秀的梯度下降算法能极大加快学习过程。梯度下降算法可以简单理解为：从山顶下山时，挑选一条梯度最陡的路最快。

反向传播算法需要根据计算输出值与实际值的偏差来更新参数 (相差越大修改的幅度越大)，深度学习称这样的衡量为误差损失函数，计算值和真实值相差越大，误差损失越大，反之，误差趋近于0。其中为预测输出值，为实际值。

误差函数衡量的是训练集所有样本的最终预测值与实际值的差值大小，只和输出层的预测类别有关，但这个预测值与前面几层中的参数有直接或间接关系。比喻的话，输出值是狗，但实际值是猫，误差就很大。通过梯度下降算法优化前面所有层的参数，当输出值是猫的时候，误差值变的很小，函数趋向收敛于最小值。

梯度下降算法能够根据模型当前输出值和真实值的差异，完成自动的计算，是误差损失降到最小。因为计算的过程和前馈过程刚好相反，这个过程称之为反向传播计算，已经成为了神经网络训练不可缺少的一个重要步骤了。先给定所有参数一个基础默认值，这个值通常是很小的正数。参数会随着梯度下降算法进行而改变，慢慢的朝向使误差损失函数值最小值方向改变。整个过程是随着梯度方向进行的，所以称之为梯度下降算法。当梯度无限接近0的时候，或者是产生正负值振荡时，代表误差损失逼近最小值。

如下图，由于梯度表达的是函数在某点变化率最大的方向，通过计算偏导数得到，所以使用梯度下降方式，会极大地加快学习进程。

图2.6 损失函数

在实际操作中，理论上要先检查最后一层中的权重值和偏移量会如何影响结果。将误差函数  $E$  求偏导，就能看出权重值和偏移量对误差函数的影响。可以通过链式求导法则来计算这些偏导数，得出这些参数变化对输出的影响。求导公式如下：和为得到上述表达式中的未知量，将分别对和求偏导：

原式对求偏导对求偏导然后反向计算误差函数关于每一层权重值和偏移量的偏导数，并通过梯度下降法来更新调整后的权重值和偏移量，直到计算回到的第一层为止。

上述流程就是反向传播算法，在神经网络训练中，这个步骤会反复的进行，这是由于各种数据可能存在着噪声，不一定每一步的下降都意味着是正确的，因此，需要大量的进行梯度运算，按照一定的比例去改变梯度，减少单步噪声带来的影响，更多的取均值。反向传播算法的重要性不言而喻，因此在一般的模型构建和训练中要完全的保障这个过程的正确性，如果存在错误，那么整个网络可能都无法拟合到最终的目标值。

### 1.3 常见优化算法

梯度下降算法存在着很多的变体，不同的算法有着自己的优势，没有绝对优秀的算法，需要根据具体的情况选择。下面讨论介绍几种常见的梯度下降算法，其中有非常传统的算法，也有当前研究领域新兴的优秀算法。

这里对待训练的模型做一个规定：假设模型只有一个隐藏层，并且参数为：，为了简便的讨论，将 $b$ 归入 $W$ ，可以理解为 $b$ 也与 $X$ 的某个分量相乘，不过该分量一直是0。模型的误差损失函数是，关于 $W$ 的梯度运算为，定义学习率为 $\alpha$ 。

#### 2.4.1 标准梯度下降算法

在标准梯度下降算法中，更新参数的规则为：其中，表示 $t$ 时刻的模型参数。

该更新规则完全按照当前运算得到梯度方向更新，从优势来看，就是非常直接，永远朝最直接有效的方向改变。值得一提的是，和下面要提到的批量梯度下降算法相比，这里每一个样本更新一次参数。类比实际，标准梯度下降算法，就好像是在下山的时候每次选择最陡峭的路下去，每一步的改变也是朝向最优值最大幅度的前进。

标准梯度下降法主要有两个缺点：

3 训练速度缓慢：因为是每一个样本做一次计算，因此没有充分利用到矩阵计算的优势。而且每次运算用的样本是不同的，可能存在噪声，也就是说当前样本计算得到的最快的下降方向未必是正确的，反复叠加下，就好像是走了很多的冤枉路，每一步贪婪的选择，在整体来看却在盘桓的下降。

4 容易陷入局部最优解：在拟合函数训练时，可能函数存在着局部最优值，如果采用“目光短浅”的标准梯度下降算法，在落入局部最优值时，缺少能够逃出局部的方法。从而导致无法训练得到最优值，模型训练进行不下去。

#### 2.4.2 批量梯度下降法(Batch Gradient Descent)

在实际的训练中，输入值往往以矩阵的形式批量输入。假定批量数据样本数量为 $n$ ，输入样本值矩阵为，真实标签值矩阵为，

第 $i$ 批次样本对应误差损失函数关于参数 $W$ 的梯度为，在此定义下，批量梯度下降法更新参数表达式为：其中，表示 $t$ 时刻的模型参数。

与标准梯度下降算法的表达式相比，很多相似，不同点在于更新值用到的不是单个样本的梯度值，而是批量样本梯度值的和，这样的意义在于，如果存在反常的噪声也会被梯度求和弱化对更新的影响，单个值的误差不再导致梯度下降方向的改变，对训练速度有较大的提升。

批量梯度下降算法类似站在原地尝试的向不同的方向计算梯度，但是并不走过去，而是综合所有的结果，取平均最快的一条路走到下一个点。

对比标准梯度下降算法，批量梯度下降算法有太多的优势，因此更具有使用价值。

#### 2.4.3 随机梯度下降法(Stochastic Gradient Descent)

随机梯度下降算法和标准梯度下降算法很类似，但又结合了批次训练的特点。输入数据依然是数量为 $n$ 的单个批次，从中



随机抽取一个样本，使用随机梯度下降算法更新参数表达式为：

其中，表示从n个计算得到的梯度中随机选择的一个梯度，表示t时刻的模型参数。

随机梯度下降算法每次计算得到一个批次的梯度并且计算误差损失值，这点和批量梯度下降算法相同。但是却只利用了一个批次中的一个梯度值来更新，而不是平均值，这一点又和标准梯度下降算法类似。之所以这样做，是因为批量梯度下降算法在有些情况下，比如噪声较大时或者是训练到即将收敛时，批量梯度的均值会非常的小甚至是消失了，导致最终模型反复的在极大值附近盘桓无法进一步下降。而采用随机采样一个梯度，就拥有很大机会是训练继续下去，进一步达到更小的误差损失。换一个角度来看，这种随机采样的方法再次的和正则化思想不谋而合，抛弃部分数据从而提高泛化随机性。

SGD的计算方法比较简单，计算量很小，使得即使是非常大量的数据依然可以很快的速度训练。在训练初期速度比标准梯度下降算法要快，比批量梯度下降算法要慢，但在训练的末端收敛情况要比批量梯度下降算法更加优秀。

SGD是随机选择梯度，不可避免会引入噪声，这使得参数更新方向并不总是正确。SGD对局部最优值可能需要较长时间的训练才能摆脱出来，这与随机到的梯度值有关系。

2.4.4 动量优化法

动量优化方法是在随机梯度下降法的基础上进行的改变，改进了随机梯度下降算法速度较慢的弊端。常用的是标准动量优化方法Momentum。

Momentum的思想是在SGD的基础上引入一个动量的概念，记录历史下降梯度信息。

从训练集中选取一个尺寸为n的批量样本，对应的标签值分别为，则Momentum更新优化表达式为：

其中，表示t时刻积攒的动量。 $\alpha$ 表示对动量的利用比例，通常取值为0.9。表示t时刻模型参数。

动量的引入主要解决了随机梯度下降算法的两个不足：第一个是随机梯度的方法，动量法可以通过控制 $\alpha$ 的大小来操控随机的程度；第二点是SGD在梯度下降过程中梯度会来回摆动，影响训练速度。关于这点，由于动量的引入，每次更新参数时，受到上次更新参数的影响。就好像是小球向山下滚带上了一定的惯性，小球不容易产生左右的来回滚动。

2.4.5 AdaGrad算法

AdaGrad算法，能够自适应模型的所有参数的学习率，所有的参数都具有自己独立的学习率，而不是简单的让学习率随训练轮次的增多而减小。对误差损失函数贡献较大的参数将会有较大的学习率，反之，则学习率会比较小。

AdaGrad算法优化表达式表示为：

在一个多分类问题中， $i$ 表示第 $i$ 个分类， $t$ 表示第 $t$ 次迭代。设置为一个很小的值，为了避免分母为0的情况。表示t时刻，指定分类 $i$ ，代价函数 $J(\cdot)$ 关于 $W$ 的梯度。

AdaGrad算法对出现频率高的类别标签给予较小并且越来越小的学习率，而频率低的类别数据将会设定为较大的学习率。如果数据集的分布不均或是数据稀疏，AdaGrad能够较好的处理这种情况。

AdaGrad不需要手动设置学习率变化的特点非常的有吸引力。但是算法随着时间的进行会导致学习率越来越小，最后接近0无法继续训练。

2.4.6 RMSProp算法

RMSProp算法改进了AdaGrad的梯度积累变为指数加权的平均值，这样使得即使是在模拟非凸函数时也有着不错的表现。

RMSProp算法的参数更新表达式表示为：表示前 $t$ 次的梯度平方的均值。 $\alpha$ 和AdaGrad中一样是对历史下降梯度的利用比例，是全局初始梯度学习率。是非常小的值，避免分母为0。

RMSProp是对AdaGrad的进一步提高，从表达式分析可得，是分母，由于取加权平均，解决了学习率越发走低的弊病，同时学习率依然能够根据参数值的大小自动适应变化。

RMSProp算法经过实际运用的考验，被证实时一种行而有效的梯度优化算法。广泛的运用在深度学习领域。

2.4.7 Adam算法

Adam算法是对RMSProp算法一种改进。RMSProp算法在训练初期时，二阶矩估计存在过高的问题。Adam引入了偏置修正，改善了该问题。

指 标
疑似剽窃文字表述
1. 2.4.4 动量优化法
动量优化方法是在随机梯度下降法的基础上进行的改变，改进了随机梯度下降算法速度较慢的弊端。常用的是标准动量优化方法Momentum。
2. 训练集中选取一个尺寸为n的批量样本，对应的标签值分别为，则Momentum更新优化表达式为：
其中，表示t时刻积攒的动量。 $\alpha$ 表示对动量的利用比例，通常取值为0.9。表示
3. AdaGrad算法
AdaGrad算法，能够自适应模型的所有参数的学习率，所有的参数
4. 学习率会比较小。
AdaGrad算法优化表达式表示为：
在一个多分类问题中，
5. 避免分母为0。
RMSProp是对AdaGrad的进一步提高，从表达式分析可得，是分母，由于取加权平均，解决了学习率越发走低的弊病，

1	基于卷积神经网络人脸识别研究与实现 任飞凯(导师：邱晓晖) - 《南京邮电大学硕士论文》 - 2019-12-09	1.4% ( 128 ) 是否引证：否
2	5400413_王胜_基于深度神经网络结构的实时场景文字检测算法研究 王胜 - 《高职高专院校联合比对库》 - 2018-05-25	1.2% ( 114 ) 是否引证：否
3	基于深度学习的驾驶行为识别方法研究 吴昊(导师：秦文虎) - 《东南大学硕士论文》 - 2018-04-01	1.2% ( 108 ) 是否引证：否
4	跨天气道路场景语义分割关键技术 王文雪(导师：张洪刚) - 《北京邮电大学硕士论文》 - 2018-03-08	0.9% ( 87 ) 是否引证：否
5	基于CNN的隧道地质超前预报GPR实测数据分类研究及应用 党巾涛(导师：郭元术) - 《长安大学硕士论文》 - 2018-04-02	0.9% ( 85 ) 是否引证：否
6	基于深度学习的图像分类方法及加速训练技术 路星昊(导师：顾华玺;王皓) - 《西安电子科技大学硕士论文》 - 2019-06-01	0.9% ( 85 ) 是否引证：否
7	基于深度学习的图像超分辨率重建 刘世豪(导师：李军) - 《青岛大学硕士论文》 - 2019-06-10	0.9% ( 85 ) 是否引证：否
8	基于CRNN模型的弱标签城市交通工具声识别检测 仇鹏(导师：韦娟;胡军锋) - 《西安电子科技大学硕士论文》 - 2019-06-01	0.9% ( 85 ) 是否引证：否
9	3150104927_徐畅_中文唇语识别模型的构建与优化 徐畅 - 《大学生论文联合比对库》 - 2019-05-28	0.9% ( 85 ) 是否引证：否
10	1441507007_陈越_基于深度学习的流行音乐快乐因子提取研究 陈越 - 《大学生论文联合比对库》 - 2018-05-28	0.7% ( 61 ) 是否引证：否
11	基于深度学习的移动端图像识别研究 朱成诚(导师：孔英会;冯奇) - 《华北电力大学硕士论文》 - 2019-03-01	0.4% ( 35 ) 是否引证：否

原文内容

Adam算法参数更新表达式表示为：其中，和分是一阶动量项以及二阶动量项。和为超参数，其值经常分别取0.9和0.999；，分别为各自的修正值，该值在训练初期影响较大，在后期几乎没有影响，是一个取值很小的数，避免分母为0。

在近几年实际运用中，Adam算法使用的频率可能是最高之一，它的独特优势使得训练速度很快，梯度总是以平稳而又快速的方式下降，计算看似麻烦，但是计算量并不很大，使得运算速度很快。

### 3 卷积神经网络相关理论

#### 1.2 重要影响力网络

在2005年Dave Steinkraus等人探索发现专门的图形处理器（GPU）在机器学习训练方面有着显著的运算速度优势。AlexNet，提出激活函数ReLU，使CNN成为图像分类的主流算法。GoogleLeNet他提出了Inception结构和1x1卷积。其后雨后春笋般出现大量卷积神经网络提升技术，例如：批处理方法，残差学习思想，甚至是让网络去自己学习深度。

#### （1）AlexNet

AlexNet在计算机视觉领域取得了远超其他的分类算法的效果。[AlexNet首次在大规模图像数据集实现了深层卷积神经网络结构](#)，AlexNet网络的兴起也标志着神经网络在大众面前的火爆开端。其创新点是：

AlexNet是当时最大卷积神经网络，具有5个卷积层和3个全连接层，。

实现高效的GPU卷积运算结构，也使得此后GPU成为深度学习的主要工具。

首次运用了多种技巧，其中包括了现今依然常用的：dropout，ReLU激活函数，以及数据增强技术。能够很好的解决深层神经网络常常存在的过拟合问题，网络具有6千万参数，但依然很好的拟合了最终需要的模型。

#### （2）GoogLeNet

GoogLeNet网络于2014年由Christian Szegedy等人提出，首次运用在google举办的图像识别竞赛中，该网络训练得到的模型在比赛中取得了第一名的成绩。GoogLeNet的名字是对经典神经网络LeNet的致敬。GoogLeNet又称为Inception，由于存在多个版本，又分为Inception v1,v2,v3,v4。GoogLeNet的提出带来了不同于主流加大加深网络的思想，而是在单层内使用相同乃至更少参数的结构，提取更为丰富的特征。

GoogLeNet的核心是inception模块，通常的网络单层内只有一种结构，是卷积层，池化层，或是全连接层的其中之一。而在inception v1模块中同时包含了这些内容，包括了1x1卷积，3x3卷积以及5x5卷积，除了三个卷积层，还有一个3x3的最大池化层。这些卷积层和pooling层得到的特征合并在一起作为最终的输出。整个网络由很多的inception模块组成。Inception v1同时创新的使用了1x1卷积，从传统卷积核的意义来看1x1卷积无意义，但是1x1卷积在较小计算量的前提下能够减少计算值的尺寸。对控制网络参数数量起到了积极的作用。并且1x1卷积在不增加感受野的同时，能够进行叠加更多的卷积步骤，不增加运算量就能提取更多的特征值。

#### （3）ResNet

浅层网络逐渐增加层，模型在训练集和测试集上的性能会变好，因为模型复杂度更高了，表达能力更强了，可以对潜在的映射关系拟合得更好。而“退化”指的是，给网络叠加更多的层后，性能却快速下降的情况。按道理，给网络叠加更多层，浅层网络的解空间是包含在深层网络的解空间中的，深层网络的解空间至少存在不差于浅层网络的解，因为只需将增加的层变成恒等映射，其他层的权重原封不动copy浅层网络，就可以获得与浅层网络同样的性能。但是网络训练却不能找到更优的解，说明了优化存在问题的。ResNet调整模型结构，让模型更易于优化。ResNet网络的特色在于残差块。跨越层间的连接直接提升了



网络的整体训练效率，使网络优化得到简化。网络同时大量使用了batch Normalization。对深层的网络的梯度传递起到了优化。

### 1.3 卷积神经网络组成

卷积神经网络是指多层的前馈神经网络，其中包含有若干卷积层，其实不光有卷积层，还有池化层。卷积神经网络经过多年发展演变有了较为固定的层次结构，通常是若干个卷积层，夹杂在其中的若干池化层以及最后的若干层全连接层组成。卷积神经网络在图像识别和语音识别领域有着比其他的机器学习算法更好的效果。卷积神经网络由于广泛使用卷积层的关系，大大减少了参数的数量，在同等深度下，训练难度远远低于其他网络，并且同样是采用反向传播进行梯度下降。

#### 1.2.1 卷积层

卷积层的定义是由一组具有可训练的卷积核的集合。一般的卷积核尺寸是很小的，常见的是3x3。深度和输入数据保持一致。卷积核在输入数据空间维度上滑动计算，由多个卷积核组成的卷积层叠加得到输出。由可视化分析可以发现，浅层卷积核一般会学习到简单特征的识别，例如线条边缘，色块。而深层次的卷积核会对图像中的高级特征有反应，例如面部器官。这与动物神经细胞的功能有些相似。

卷积层相对于全连接有着两方面的优势：

##### (一) 局部连接：

局部连接，就是卷积层的节点只和其前一层的部分节点相连接，只学习局部特征。

图像领域中，传入的数据是图片的像素点，和其他的数据不同，图片像素点的相对位置特征十分重要，相互靠近的像素往往关联性很强，例如一张脸上，组成嘴唇的像素点之间关联性很强，位置不可以随意互换，但是嘴唇和头发的像素点就关联性弱，即使是位置的调动，不会影响到最终的判断。

因此，采用部分神经元接受图像信息，再通过汇合全部的图像信息达到增强图像信息的效果。两层网络之间如果是全连接，那么参数的数量将会是输出数据尺寸和输出尺寸的乘积。在图像识别这样参数众多的领域，过多的全连接会使参数暴涨。过多的参数使网络复杂，泛化能力差，同时还加大训练的难度。局部连接的方法大幅度减少了参数数量，加快学习速率，同时也一定程度上的减少了过拟合的可能。

##### (二) 参数共享

对于权值共享的理解就是，某层网络在处理输入的数据时，会用一个卷积核去扫描这整个数据，卷积核里面的数就叫做权重，整个数据的每个位置是被同样的卷积核扫描得到的，因此说权重是一样的，也就是共享。参数共享体现的依然是：图片的底层特征是与特征在图片中的位置无关的。

卷积层的卷积核是在整个输入数据上滑动计算，这就是参数共享。如果每个位置都有一个独立的卷积核，那么卷积核的数量将会随输入尺寸而增大，参数数量将会是原来的上万倍。而参数共享并不会导致信息丢失，这基于同一特征在图像上任何位置出现都是等价的原则。同层次滤波器相当于一组功能相同但是位置不同的特征提取。

#### 图3.1 按步卷积操作

#### 1.2.2 池化层

池化层的具有重要作用，主要是对卷积层提取到的特征下采样，不丢失特征的同时，降低特征数据的尺寸，这样可以减少存储成本和运算成本。在卷积神经网络中池化层通常位于卷积层之间。常见的池化种类有最大池化（max pool）和平均池化（avg pool）。通常的池化滤波器尺寸为2x2，在输入数据上移动的步长为2。

池化层和卷积层是密切配合关系。从抽象意义来说，卷积层类似特征提取工具，提取到低阶或是高阶特征，而后由池化操作保留到提取的特征，抛弃无用冗余信息。得到的信息保留有重要的特征，同时数据尺寸缩小很多，利于后续操作进行。

#### 1.2.3 全连接层

卷积层提取特征，但是卷积层对特征的相对位置不敏感，这取决于卷积层的参数共享。但高级特征与位置通常息息相关，例如人的面部，不同的人，嘴唇，鼻子，眼睛的相对位置会有差异，分辨不同的人需要用到这些相对位置信息，卷积层无法做到这些，这时就必须要靠全连接层，全连接层有大量的参数冗余，恰好的满足了位置信息的提取运算。

全连接层是神经网络重要组成，传统的理解就是拟合复杂的线性函数。通过激活函数计算，可拟合非线性函数，提高拟合能力。在分类任务中，卷积神经网络非常有效。卷积层和池化层相互配合提取特征，简化运算，映射至特征空间。而全连接层则将特征表示映射到样本标记空间，因此全连接层一般位于神经网络的最后几层。实际中可以由卷积操作替代，转化为卷积核尺寸为1x1的卷积操作。

### 4 解决过拟合

#### 4.1 正则化

在图4.1中展示了神经网络拟合一种示例，如果要叉和圈分开采用方式2或是方式3都可以，方式1拟合不足很多分类是错误的。而方式3也存在着问题，因为样本本身存在噪声，拟合模型不应该被要求在训练样本中达到100%准确度，这对泛化能力没有提升。有时候我们应该采用尽可能简单化的模型，可以看到方式2的曲线要比方式3的曲线更加让人信服，实际情况也往往如此。方式3就称为过拟合。

##### 4.1.1 L1和L2正则化

正则化的出现正是为解决过拟合而存在。过拟合的情况往往是神经元的参数w值过大导致，我们需要的模型是拟合能力不错并且w值较小，这样得到模型比较简洁，拟合能力适中。常见的几种正则化都是在损失函数中加入关于w的L范数，常用L1正则化（式1）和L2正则化（式2），对w比较大的模型进行强烈惩罚，w比较小的模型计算的损失会更小，容易被选中。

L1正则化得到的W参数矩阵是稀疏矩阵，有很多的0，而L2正则化得到的不是稀疏矩阵，W的值很小。图4.1 训练拟合情况

##### 4.1.2 Dropout正则化

神经网络如果规模很大，而训练数据集很少，比较容易发生过拟合，有一种方法是训练多个结构不同的网络进行组合，利用网络的差异来减轻过拟合情况。实际上很少使用这种方法，因为网络训练时间长，运算成本高。通常采用具有相似效果的Dropout，不增加训练难度的前提下，能够显著的抑制网络内部的相互依赖性，提高模型泛化能力。

Dropout的原理非常的形象化，多轮训练时，每次随机按比例使部分的神经元暂时关闭，不参与前向和反向传播计算，那



么暂时关闭的神经元参数就相当于从网络中剔除。不同轮次训练之间，好像是在使用不同的网络结构，起到了和同时使用多种网络结构组合训练的效果，神经元之间的依赖性大大降低。

进行计算时，每个神经元都有相同的概率关闭或是开启，整个网络的结构在训练时处于动态变化中，如图4.2所示。

#### (一) 求平均

采用了Dropout的神经网络使用了求平均的思想，不同轮次之间的网络结构不同，因此梯度下降的方向也有了不同，所有的训练轮次综合起来，单个神经元会被重复的进行参数的变化，这个最后的参数结果是多个网络结构训练的变化值平均而来的结果。虽然用了多种结构的网络，但这样子做不仅没有提高运算量，还减轻了单轮次内需要训练的参数量，但是网络想达到收敛，要比不适用Dropout要多训练很多轮次。

#### (二) 减轻神经元之间相互依赖关系

在不同轮次中，出现的神经元组合不一样，甚至是数量也不一样。神经元如果存在着关联，受到Dropout的影响将会减轻或者是消失。神经元之间的固定关联会增加网络的复杂性，提高过拟合的可能。Dropout产生了微弱的正则化效果。

图4.2 Dropout操作

### 2.2 Mini-batch

在训练神经网络时，一般输入是以矩阵的形式，数据量为第一维度。图像识别领域中数据集数量基本在上万，乃至十万百万。受到内存限制，这样一次输入一个矩阵将会变得极为困难。如果采用另外一个极端，那就是一次喂入一个数据，损失函数难以收敛至全局最优。参数在局部范围内来回震荡，训练效率低下，对内存的利用也不够充分。

这就是Mini-batch诞生的缘由，选择batch-size，也就是一次喂入数据的量，也是在内存利用和梯度下降效率之间的抉择。

#### (一) 增大batch的优点

对GPU而言在处理合适大小的矩阵时效率最高，碎片化的batch不能充分利用到GPU加速运算的优势。并且batch较大时，取得的下降梯度平均值越接近整个样本集的平均值（当batch退化为整个样本集时就完全相等），这样下降的方向更为准确，训练的速度能够明显加快。

#### (二) 盲目增大的坏处

batch的大小不能无脑加大，首先要考虑到GPU的内存和显存位数，过大的batch使存取内存频率增加，反而降低训练效率。并且过大的batch会有批量梯度下降算法的共有毛病，当训练至即将收敛时，由于过于平均，导致参数精度比随机梯度下降算法差，这一点不利于模型的精细化处理。

### 2.3 标准化

深度神经网络具有很多的层。假如数据值的分布是均值为0，方差为1的高斯分布，但经过了层层运算，到了激活层就不再是原来的分布了。分布情况的变化可能影响到整个网络的训练效果。例如，当最后几层网络的激活函数输入值都为负数时，整个网络就好像停止了训练一样，参数不在发生改变。其实网络还没有达到收敛。即便是正数，也可能是非常大的值，使收敛的难度加大，需要训练的时间变长。严重的影响了网络训练效果。

标准化操作就是让网络的数据分布在各个部位处于最佳状态，数据分布在合理范围，降低训练难度。

神经网络中，常用的标准化策略有批标准化 (Batch Normalization)，权重标准化 (Weight Normalization)，层标准化 (Layer Normalization)。

#### 2.2.1 Batch Normalization

神经网络乃至机器学习都有一个默认共识，算法能够经过训练后能够在从未遇到过的样本上优秀工作的条件是：训练样本和测试样本是属于相同分布的。这是从概率统计学的角度上保证算法泛化有效的条件。

Batch Normalization总的来说，就是在所有的隐藏层激活函数输入之前，将输入值的分布固定下来，而不是随着输入批量样本的变化而改变。由于计算分布是需要多组数据的，因此需要配合mini-batch来使用。Batch Normalization的启发来自于输入图像处理白化操作。将这个操作推广至不限于输入层的所有隐藏层。就好像是进行了多次白化处理。能够保证即使是最后一层的输出值依然是保持分布在合理有效的范围内。

也有一种说法，Batch Normalization配合上合适的参数初始化方法，使得损失函数的初始值在一个较小的范围内，而往往这样子的梯度下降比较快，容易收敛至最优值。

#### 2.2.2 Weight Normalization

WN的做法是将权值向量 $w$ 在其欧氏范数和其方向上解耦成了参数向量 $v$ 和参数标量 $g$ 后使用SGD分别优化这两个参数。

一个神经元经过激活函数后的输出为 $y$ ：

$$y = \phi(wx + b)$$

其中的权重可以用方向向量 $v$ 和大小标量 $g$ 来分别表示：经过实验的证明，方向向量和大小标量分开进行训练，能够有效的加快网络的训练速度。相比于Batch Normalization，Weight Normalization的计算量更小。

#### 2.2.3 Layer Normalization

Layer Normalization可以将数据重新分布到激活函数的非饱和区，具有权重/数据伸缩不变性的特点。起到缓解梯度消失/爆炸、加速训练、正则化的效果。Batch Normalization的标准化对象是一个神经元的单批次内全部样本进行标准化，而Layer Normalization是对同一层的所有神经元输入值进行标准化。

一般的均值和方差计算如下：其中 $l$ 代表第 $l$ 层， $H$ 代表本层神经元个数。

### 5 相关算法及数据集介绍

CIFAR-10使用已久，是用于图像识别的经典数据集。几乎在高级深度学习框架中都包含了该数据集的包。和名字相似，数据集共有10个类别的物体彩色图片，统一都是 $32 \times 32 \times 3$ 的尺寸，最后的3代表的是RGB格式，每个像素点都由三种颜色混成，因此具有较高的对比度。物品的类别是：飞机、汽车、鸟类、猫、鹿、狗、蛙类、马、船和卡车 ( airplane、automobile、bird、cat、deer、dog、frog、horse、ship、truck )。其中每种物体类别都含有6000张图片，总计是50000张用作训练，10000张用作测试。

图5.1 CIFAR-10数据集

VGG是由Simonyan和Zisserman在文献《Very Deep Convolutional Networks for Large Scale Image Recognition》中提

出卷积神经网络模型，VGG是 Visual Geometry Group的缩写，表示牛津大学视觉几何组。模型最初是用于参加2014年谷歌公司举办的ImageNet图像分类和定位大赛，该算法模型在分类任务上排名第二，在定位任务上排名第一。

### 1.1 网络结构

#### 5.1.1 VGG

VGG中根据卷积核大小和卷积层数目的不同，可分为A，A-LRN，B，C，D，E共6个配置(ConvNet Configuration)，其中以D,E两种配置较为常用，分别称为VGG16和VGG19。下图5.2给出了VGG的六种结构配置：

#### 图5.2 VGG网络结构

图中绿色部分即指明了VGG16所采用的结构。从图中可以发现，VGG16共包含：13个卷积层，3个全连接层，5个池化层。并且层次分布相当的有规律，其中含有块：若干卷积层->池化层，最后的是三个全连接层。由于池化层没有可训练的参数，因此不计入，而能够训练的卷积层和全连接层加起来是16层网络，所以叫做VGG-16。

#### 1.1.1 VGG-16特点

VGG-16的特点是比较简单，所有的卷积核使用相同的尺寸（3x3）和卷积步长（1），采用same填充，使得每一个卷积层和前一层保持一样的宽高。使用的池化层参数也全都一致，均为2x2。模型是由若干卷积层和池化层堆叠构成，形成了相似的5个块（block）结构。

#### 1.1.2 块结构

VGG-16网络可以划分为5个块，每个块之间由池化层分隔，每个块内包含若干卷积层和一个池化层。同一个块内不同的卷积层包含的卷积核数量是相同的。不同块之间单层卷积核数量逐块翻倍：64、128、256、512、512。而每层尺寸逐块减半：224、112、56、28、14、7。这体现了随着网络加深提取到的高级特征逐渐增多，而冗余信息逐渐减少。

### 1.2 模型修改

#### 1.2.1 输入层

标准的VGG-16网络的输入图像尺寸是224x224。而CIFAR-10所有的图像的的尺寸是32x32。因此输入层需要改变成为32x32x3。

#### 1.2.2 模型深度

VGG-16后两块的参数数量较多，为了加快模型的训练速度，先截取VGG-16的前3块。全连接层不采用庞大的三层结构，而是采用平展层加10分类的softmax分类器。所有的参数初始化使用He Normal。在加入Dropout层，减轻过拟合情况。

#### 1.2.3 Keras快速建立模型

Keras是python语言中的高级深度学习框架。使用Keras能够快速有效的搭建各种复杂的神经网络，并且Keras支持了现今各类主流的训练方法，类似Dropout，标准化，各类优化器等等，能够满足一般的实验要求。通过summary函数查看网络结构得到图5.3。

#### 图5.3 改进VGG-16网络结构

### 1.3 数据增强

在训练的过程中，数据量对模型的最终效果起很大的作用。当数据量较小时，模型在训练集很容易收敛，同时在验证集和测试集上的表现会很差，意味着模型容易过拟合，泛化能力差。

图片的特性是特征在各个位置出现的可能都是一致的，有的特征存在由对称的特点。即使是数据量足够多，很可能关于位置 and 对称方面的信息还是不够多。因此在图像识别领域广泛的使用数据增强技术。

#### 1.3.1 常用技术

##### （一）翻转

可以对图片进行水平和垂直翻转。这对提取有对称结构的信息非常有帮助，例如：物体的朝向。

##### （二）旋转

旋转带来的收益可能和翻转有些类似，但是不是所有的图像都可以旋转，如果图片是正方形就可以。

##### （三）缩放比例

图片的缩放可以是放大也可以是缩小。放大时得到的图像比原来的大，还需要进行裁剪，剪除背景和部分的图像主体，让图像的尺寸能够被模型处理。

##### （四）裁剪

翻转和旋转都不会改变图像的大小，然而裁剪会改变图像的大小。在训练的时候通常采用随机裁剪任意部分，在测试的时候则选择关闭裁剪功能，不裁剪。值得注意的是，在竞赛的时候，如果采用裁剪输入的多个版本然后将结果进行会合，对预测的提升效果会非常明显。

裁剪和缩放的最大差异在于，裁剪只是从原图像中随机裁剪一部分进行训练。该部分会在随后调整补充为一个正常图像大小。

##### （五）移位

移位只涉及沿X或Y方向（或两者）移动图像。

##### （六）颜色增强

在图像检测项目中，有些采集的图片数据与天气或季节变换有关，但由于一些限制条件，无法采集到全年的图片数据，所以需要通过现有的数据，比如只采集到第一季度的图片，对图片进行数据增强。目标特征在颜色上有很明显特征，比如多数为蓝色和红色。因为天气和季节的变换，全年采集到的图片，其特征的颜色深度和亮度应该有所不同。对颜色的数据增强，包括色彩的饱和度、亮度 and 对比度（contrast）等方面。

##### （七）颜色变换

上面的几何变换操作，没有根本改变图像本身的内容，它是选择了图像的一部分或者对像素点进行了重分布。如果改变了图像本身的内容，就属于颜色变换的数据增强了，常见的技术包括加入噪声、局部模糊、颜色变换、擦除、填充等等。严格来说颜色增强也是颜色变换。

加入噪声的数据增强就是在原来的图片基础上，随机叠加一些噪声，最普遍的做法就是加入高斯噪声。更复杂的就是在面

积大小选定，加入位置随机的矩形区域上添加像素制造黑色矩形块，产生一些彩色噪声，以 Coarse Dropout 方法为例，随机的选择图像上任意部分擦除颜色，相当于加入了黑色块。

- (八) 多样本数据增强
- 上述的几种都是单样本数据增强，多样本数据增强方法是利用多个样本来产生新的样本。常见的方法有：SMOTE、mixup、SamplePairing等等。多样本数据增强可以通过多样本合成新的样本，增加原本数量较少多的类别的图像。减少样本分布不平衡情况，能够显著提升模型的泛化能力。
- 类不平衡现象是很常见的数据分布现象，假设利用一组数据集来训练一个分类器网络，如果这组数据集包含了多个类别的数据，但有的数据占比例很小，只有1%，那么，即便是分类器在该类别上的判断全部出错，只要是在其他分类中表现良好，最终的准确度也会非常的高，但是用该分类器到实际生产环境中运用，由于有一整个类别无法分类，效果会非常的差劲。

指 标
疑似剽窃文字表述
1. 局部连接的方法大幅度减少了参数数量，加快学习速率，同时也一定程度上的减少了过拟合的可能。
2. Simonyan和Zisserman在文献《Very Deep Convolutional Networks for Large Scale Image Recognitio

3. 基于深度学习的图像分类研究\_第3部分

总字数：8215

相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)

原文内容

SMOTE方法，运用了插值技术，将全部的样本映射至样本空间，欧式距离较为相近的两点之间随机取出一一点作为新的样本值，将新的样本值也加入到样本空间，重复这个过程将会产生足够数量的新样本。

Sample Pairing需要运用其他的基础数据增强技术，两个相同标签的样本采用同样的数据增强方法处理后得到的值求平均，得到新的样本值。而标签就是原来样本的标签。

6 卷积神经网络应用

接下来介绍本设计的实验内容，将前面所述各种理论知识综合运用。构建一个修改版本的VGG-16网络。目的是通过在CIFAR-10数据集上的精心训练，能够在训练集和验证集上达到较高的通过率。最后提交到Kaggle平台相关竞赛评判系统中，查看测试集的通过率以及在平台全球竞赛排行榜的成绩。

2.1 图像预处理

CIFAR-10数据集是使用非常广泛的、很规整的数据集。无需图片尺寸进行处理，也没有干扰很大的噪声。但是为了能够达到更优秀的训练效果，需要进行输入集归一化，对标签集进行独热编码，批量读入数据集以提高GPU并行处理性能，最后是对图片的增强，详情可见5.3章节。

2.1.1 归一化

CIFAR-10图片加载后为numpy的array格式，shape为[1,32,32,3]，数值范围为(0,255)，输入值最大到达255。输入值大小本身不具备任何的信息，重要的是输入值的分布情况，另外一点，输入值过大，可能影响到最终损失值也过大，不利于损失函数的快速收敛。基于以上两点原因，对输入数据的进行归一化处理。在numpy中由于运算符的重载使得这步非常的简单。只需要像处理标量一样的去处理矩阵。

2.1.2 独热编码

在分类预测任务中，最后的输出值不会在具体分类的名字，只需要用不同的数字来代替即可，这对二分类而言是非常合适的，但在十分类中，如果用0到9的输出值代表十个类别，演变为了一个回归任务，对最后的准确度有很大的影响，因为存在中间值的情况会很多。常见多分类预测任务的最后一层网络的激活函数是softmax激活函数。它输出值是一个矩阵，尺寸为分类的数量。输出矩阵的每一个数代表着是对应分类的可能性，因此值一定是在(0,1)之间，取最大值即是我们的预测输出值。基于该方法，我们需要预先处理标签值。例如，某图像的标签值是2，那么独热编码之后的值就是[0,0,1,0,0,0,0,0,0,0]。

在tensorflow中工具有专门的方法来转换处理，来批量处理矩阵，效率远远超过循环处理，方法名：to\_categorical。

2.1.3 选择mini-batch大小

由于是使用的GPU训练，每次采用批量输入数据将会充分利用到GPU矩阵运算带来的加速效果。而GPU的内存大小是有限的，不可能一次将全部的批量数据全部投入，因此batch-size的选择是对训练速度和GPU承载能力的一种权衡选择。

根据LeCun的[6] Dominic Masters, Carlo Luschi,Revisiting Small Batch Training for Deep Neural Networks论文中的研究，batch-size最好的实验表现一般在2~32之间。文中提到一个关键思想，mini-batch的一个重要效果是，每次没有使用全部的数据来训练，梯度的变化并不是全量数据，人为的带来的噪声，而就是这噪声，有利于使梯度下降算法走出局部最优解。

因此，实验中选择mini-batch的策略为：在训练初期为了提高单轮次的效率，设置为32。随着训练到接近收敛时，设置batch-size为2甚至是1。为的是让损失函数值进一步的减小。

2.1.4 数据增强

对于数据增强方面，实验中采用了对比方法。先不使用数据增强进行训练，当模型趋于收敛时，使用数据增强，观察记录数据增强为最后的通过率带来的增益大小。

增强的方法采用了：垂直翻转，水平翻转，旋转。在tensorflow中存在专门的ImageDataGenerator，简化这整个过程。在训练的初期不建议直接打开数据增强，因为减少影响因素能有利于我们在训练过程中调优，找到问题的关键点。

2.2 优化及参数设定

训练一个网络，有很多的超参数待调试，每一个超参数对训练效果有着不同的影响，过往的训练经验或者是其他人的论文中的超参数，不一定就是最适合自己的模型，超参数的选择需要反复的尝试和考量。



### 2.2.1 学习率

学习率是几乎所有梯度下降算法都存在的参数，有的可以自动学习，但即使是自动学习也需要手动设置初始值，因此学习率需要精心设置，并且多次尝试。较小的学习率，意味着参数变化慢，变化的粒度更细。模型训练的末轮时，较小的学习率能保障尽可能的去拟合更准确的参数。较小学习率也会导致学习速度缓慢，在训练初期，根本没必要让参数细粒度的去变化，大概率的可以确定不会在这个过程中收敛，尤其是被困在局部最小值的时候，可能需要很多轮次才能出来。而较大的学习率虽然步长较大，但在训练的末轮容易在较大的范围内产生来回振荡，而不能收敛至更小的值。

根据其他论文中的训练经验，初期采用大学习率加快训练的进度，在末期则使用较小的学习率以搭配前面提到的小batch-size不错过任何一个梯度下降的机会，虽然花费的时间可能比较长，但能在训练的末端精度上有着显著提高。

基于以上的观念，优化器方面选择Adam优化器和SGD优化器对比实验，Adam优化器的优点是利用了历史学习动量，使得梯度下降始终朝着大方向前进，而不是反复的变动，浪费了学习的效率，搭配较大的学习率，非常适合学习的初期。SGD优化器采用随机梯度下降，在下降方向上有着很多的变化性，其优点是能够有着随机动量来突破局部最优值的陷阱，结合小学习率使用，能够在末轮学习当中，发挥出最优秀的效果。

### 2.2.2 训练轮次

训练轮次的选择对最终的模型的收敛效果并不是决定性作用，换言之，如果机器的性能足够好，训练轮次当然是越大越好。但考虑到机器的性能并不是那么的优秀，再加上有一种训练的思想叫做早停，我设置的训练轮次非常的小。早停也就是当模型训练的过程中，在训练集的通过率会逐渐增大，但是在测试集上并不是单调上升，而是一种曲折上升，甚至在末端可能会转折有所下降，这是因为出现了过拟合现象，模型学到了训练集的错误特征，而解决的方法是在模型尚未发生过拟合或是过拟合不严重的时候停止训练。

我机器的性能不是很好，在训练的过程中，程序有时发生卡顿，导致整个训练结果的丢失，这非常的影响训练的效果。我采用了灵活的设计，每次模型30轮次，不论效果，都停止下来，将训练结束的时候的模型参数保存，开启下次30轮训练时，将上次保存好的参数载入到模型中，继续训练，在我的多次尝试后，发现效果非常不错，因为部分超参数的改变，甚至是网络的局部改变，都不会影响到上次模型保存参数的价值，训练的成果得以继承。借此方法，能够在训练效果不佳的模型上进行改进，并且充分的利用了前面的训练，避免了每次改变参数和网络结构后的重新训练，大大节约了训练的时间。

### 2.3 实验环境

系统：windows 10；

编程语言：python；

网络框架：tensorflow、keras；

GPU：GTX 1650 super 4G；

内存：16GB；

由于GPU的内存只有4G，在开启数据增强的时候内存不够用，启用了tensorflow自带的动态内存分配，这会使得部分的数据分配在普通内存上，而不是全部位于GPU，稍微的降低了学习效率。

在这个配置下，不开启数据增强的条件下，单论次训练用时约20秒。开启后，单论次的训练用时约为1分钟。

在较差的学习环境下，也有很多的加快学习效率方法，最重要的前提是保障训练效果不造成明显的影响。例如提高学习率，调整batch至GPU的最合适大小。

### 2.4 实验数据分析

在多阶段的训练过程中，每一段学习过程都做了数据的记录，主要是训练集上的预测准确率（accuracy，简记作acc）、损失值（loss），以及验证集上的预测准确度、损失值。所有的数据使用tensorboard动态记录生成，图表的绘制使用了matplotlib，因为要对阶段性数据分析，因此不能直接使用tensorboard的生成图表。

#### 2.4.1 整体训练过程

整个网络训练按照逐步调试原则进行，先尝试性搭建网络，根据训练效果，调整网络和超参数。不断迭代这个过程，使网络最终达到预期的优秀表现。下面的图是整体训练效果如下：

图6.1 训练集acc图图6.2训练集loss图

图6.3验证集acc图图6.4验证集loss图

从图中可以看到，网络在数据集验证集中的变化不是一成不变的，这是由于在多次训练过程中不断优化网络的结果。

所使用的网络最终在训练集上的accuracy达到：97.36%，在验证集上的accuracy达到：87.08%。在Kaggle平台上的评分达到0.87110，由于比赛已经结束，所以榜单排名不会改变，对比排名是第37名。

图6.5 kaggle得分

#### 2.4.2 初步尝试

图6.6 初步训练集acc图图6.7 初步验证集acc图

在搭建网络之初，对整个训练的情况并不了解，使用优化器是SGD，学习率为0.0001。

开始训练后，不论是训练集还是验证集的acc，都随着时间而增长，说明了网络结构和算法是基本正确的，优化起到了正向的作用。

由于这是一个非常小的学习率，导致的结果是训练缓慢，在训练轮次达到140轮，训练集上的acc率才只有50%，而验证集上的性能大大超过了训练集，达到了60%。因此，初步判断的结论是，学习率过小。需要调整学习率。

#### 2.4.3 调整学习率

图6.8 调整学习率训练集acc图图6.9 调整学习率验证集acc图

基于首次训练总结的问题和解决方法，分三次调整学习率，每次调整都是增大学习率，直到学习的速度达到满意的效果。由于学习率的特性，每次调整的值选择为[0.001,0.01,0.1]。

由图[14]和图[15]分析得到学习率的调整大大加速了整个学习的进度，训练集上的acc率快速的达到了80%，在训练轮次达到300的时候，训练速度也没有表现出有下降的趋势，说明了在训练集上，网络依然有着很好的梯度下降方向。

新的问题是，验证集中的acc率非常差，一直维持在0.62附近波动，随着训练进行，没有明显的上升。说明模型发生了过



拟合现象。一般而言，解决过拟合问题，可以采用多种方法，代价最为低廉的就是dropout。改变网络结果，在平展层上加入dropout。

#### 2.4.4 使用dropout

图6.10 dropout训练集acc图图6.11 dropout验证集acc图

在使用了dropout技术后，继续训练，网络在训练初期，acc率稍有下降，这是因为dropout有着轻微的正则效果，导致模型在开始时失去了部分参数参与，预测的值也和原来有不少的差别，降低了acc率。

随着训练进行，模型在训练集上达到了完全的拟合，acc率达到近100%，验证集上的acc率也有了显著的提升达到72%，提升了近10%。说明了dropout有一定的效果。72%的准确度依然不能让我们满意，这依然是严重的过拟合，因此需要用其他的技术。

由于数据集的限制，无法得到更多的数据。在这种情况下，图像识别领域广泛的采用图像增强技术。通过图像的平移、翻转、颜色增强等方法，变相的扩充我们的数据集。因此下一轮训练使用dropout与图像增强配合使用，希望能够有所提高网络的泛化能力。

#### 2.4.5 使用图像增强

图6.12图像增强训练集acc图图6.13图像增强验证集acc图

使用了图像增强技术后，数据集变相扩充了好几倍。训练的初期，模型在训练集上的acc率大幅度的减小，这意味着先前的模型的泛化能力非常的差，不能在很好的在数据增强产生的数据上有良好的表现，这与我们先前的预判是相符合的。

随着训练进行，训练集上的acc率回到了近97%，接近收敛，在局部很小的范围内波动。足以证明整个模型又一次的训练至最优值。验证集上的acc率收敛到87.5%，提升了近15%，这是相当大的提升。充分说明了数据集的大小对网络训练最终的效果起着至关重要的作用。87.5%的训练效果在十分类任务当中，已经是不错的准确率。

但分析训练集和验证集的acc率差距，可以看出依然存在着过拟合，有充足的理由怀疑模型在训练集上拟合过度了，可以通过适当的方式降低其在训练集的准确度，以便于提升网络在其它测试集的期望准确度。

过拟合的产生往往是网络的复杂度超过了需要，而网络中层参数越大，拟合的函数模型就越复杂。因此降低过拟合，可以对所有的卷积层进行L2正则化，预期的效果是：训练集上acc率有所下降。验证集上acc率保持不变或者是有所提升。

#### 2.4.6 L2正则化解决过拟合问题

图6.14正则化训练集acc图图6.15正则化验证集acc图

L2正则化对较大的参数进行强烈惩罚，而惩罚的力度可以通过超参数 $\lambda$ 来调控，我一共尝试了三种参数值，每一次都在减小参数值，这是因为过于强烈的惩罚，会降低模型在验证集上的准确度，过犹不及。

多次训练过程中，对L2正则化参数 $\lambda$ 的选择进行了多种尝试，尝试的值包括了[0.1,0.01,0.001]。分析图[20]和图[21]，L2正则化对训练的结果有着巨大影响，这主要是因为L2正则化抑制着所有卷积层参数的增大，减少了网络的复杂度。网络复杂度降低也就意味着，拟合能力有所下降。调节 $\lambda$ 参数，也是在平衡拟合能力的过程。

经过尝试，在0.001的情况下，测试集的准确度有了明显的下降，并且随训练轮次加大，有更多的下降，而在验证集上下降的幅度非常下，不超过1%。这符合我们的预期，由于轮次加多，模型在验证集上的性能也有所下降，所以需要使用早停，在测试集准确度和验证集准确度之间做一个权衡，选择最适合的状态。

#### 2.4.7 结果分析

最后的训练效果是不错的。总共花费的训练时间大概是接近一天，这中间包括了调优时的暂停时间。模型的参数调整还有这很多进步空间，该网络的结构也不一定就是最优秀的网络。但是整个训练过程是可重复的。模型训练总是在拟合能力上左右权衡。有一个理论是说，不论是精心调制的超高层次模型，还是直接瞎猜，其实本质是没区别的。而真正有区别的是作用目的。也就是说，没有放之四海而皆准的网络，没有最佳的网络，只有适合局部应用领域的网络。

上述训练过程充分展示了，如何提高模型训练速度，模型拟合能力，以及发生过拟合时的调整方法。整体的学习训练调整思路是如此的，下图[22]kaggle比赛排行榜：

图6.16 kaggle排行榜

对比最优秀的10个网络，可以看到在测试集上准确度达到惊人的95%，如果有人看过CIFAR-10数据集，会惊讶的觉得，这已经超越了人类的能力。因为有的数据非常的模糊，即便是人类来判断，也无法识别究竟是什么。

深度学习的一个误区就是模型在测试集上越准确越好，意味着泛化能力越强。这是错误的，由于训练者为了提高测试集上的准确度，做出了非常多的尝试，导致了测试集退化，变成了调试的训练集。当更换测试集或者是投入到实际生产当中时，就会发现，泛化能力远不如原来的测试集。这也是当前研究领域的一大失败，采用了非常多种的提升手段，却发现模型还不如没使用之前。

结语

#### (1) 总结

图像识别是计算机视觉研究的基础，也是如今人工智能的一个重要的成熟的领域。过去的十几年里，卷积神经网络在这方面大放异彩，成为了主流的技术。卷积神经网络其中蕴含的哺乳动物视觉细胞功能，以及局部连接、参数共享的思想为其实现了处理视觉信号网络的基本功能。卷积神经网络有着优秀的扩展性，在不具备图像处理专业知识的情况下，依然能够以超越人类的分类精度工作。

本设计中，先概括性的介绍了神经网络的背景历史和发展前景，还有一些创新性的网络结构。着重分析了经典卷积神经网络的拓扑结构。分析了多种现如今广泛使用的优化技术，包括了：正则化、dropout、mini-batch、归一化。

对现有的VGG-16卷积神经网络进行了改进，专门性的用于CIFAR-10数据集，详尽的展示了如何在一个并不知道特征的数据集上进行学习的全过程。充分的利用了上述的各类调试技巧。

分析了在训练过程中遇到的各种情况，例如欠拟合、过拟合问题，训练速度问题，以及在数据集大小不变的前提下如果解决模型的泛化问题。最后经过近800轮次的训练，得到一个验证集得到87%左右的模型。

#### (2) 展望

本设计最终的模型取得了不错的效果，但是还是有很大的提升空间和研究价值。作为人工智能基础，不仅仅是卷积神经网络

络，整个深度学习领域都在日新月异的发展中，由于研究时间和研究水平的限制本文有很多的不足。在不断的深入学习后，总结可以发现，VGG-16模型在如今众多的神经网络当中已经是历史产品，其网络结构有一定的深度，但优化能力不足。需要对比设计多种网络结构，以此检验各种网络优劣得失。

本文在卷积神经网络的解释上有待提高，目前，深层卷积神经网络内部包含的信息量巨大，缺乏高效的解读方法。当前的研究领域对这方面有着不少探索，例如对卷积神经网络卷积核的可视化。目前尚未有突破性的进展，并不能泛化的解释所有的网络。有句话叫做知其然，还得知其所以然。卷积神经网络并不是什么玄学神学，靠的不是碰运气，如果能够更多的了解这一利器，深度学习可能会有突破性的进展。而不是盲目的加大网络，加深网络。

目前神经网络的规模越来越大，训练的成本高昂。有不少的研究有针对性的在寻找移动端和低功耗设备上可运行的卷积神经网络。这一研究方向也不容小觑，在全民手机化的时代，如果能够将卷积神经网络更加广泛的利用到人们生活的各个角落，可以大大提高深度学习的影响力。是一项影响世界的重要改变。

致谢

致谢

深度学习对开始的我是一个全新的，陌生的方向，很多看起来高大上的名词。神经网络在媒体上的火爆热度也是带来了不少好奇。经过了系统的学习和实际运用，我了解到了神经网络背后的理论科学是非常的严谨和奇妙的。神经网络充分展现了多学科之间交叉带来的化学反应。概率论，计算机，图像学，甚至是仿生学等等，看似神秘的哺乳动物行为，竟然能够通过计算机来模拟达到。也无愧于当前不论是研究领域还是社交媒体都在火热的探讨。

毕业设计能够顺利的完成，得益于牢固的基础知识铺垫。正如庖丁解牛，知其全貌后，对任何部分的调控都是得心应手。除了知识的积累，还有的就是坚持不懈的探究精神，整个设计的难度是属于中等难度，并不高。各类的参考资料很多，在学习和研究过程中，我充分的利用到了前人的优秀经验知识，少走了很多的弯路。不断地打磨毕设的过程，也是一个在打磨自己能力的过程。校园和社会存在着不小的差距。不论在哪，发现问题和解决问题，都是计算机科学的重要能力。

毕业设计的最后，非常感谢提供了很多帮助的周慧斌老师。周老师在整个过程中提供了很多他的独到意见，包括了论文准备初期的学习和进度安排，这一方面为我们小组的毕设带来了很多的信心，充分相信自己能够完成任务；周老师在专业知识上也指点了不少，一阵见血的提出了论文存在的不少问题，并且为我们指导了解决的方法；老师也为今后的职场方向提供不少建议。周老师对工作负责的态度和专业知识的严谨精神，是我今后工作学习的榜样。再次向周老师表示真挚敬意。

大学生涯在紧张和忙碌的过程中度过了最后时刻，虽然远在四方，但和同学之间还是有不少的互相帮助。在遇到困难时，我们互相帮助，探讨解决。毕业设计不光是一场知识能力的校验，也是综合能力的考量。

在最后非常衷心感谢学校的培养，在学校的四年时光，已经把学校当成了一个家，学校方面为我们提供了很多的照顾，也尽力创造最好的学习生活环境。即将离开学校虽然有很多不舍，但大学得到专业的知识和能力，相信我们走出校园后也会能够发挥出自己的价值所在。感谢一切在校园中互相帮助鼓舞的同学、老师，祝今后一切顺利。

参考文献

[1] Matthew D. Zeiler and Rob Fergus ( 2014 ) : Visualizing and Understanding Convolutional Networks. In David Fleet, Tomas Pajdla, Bernt Schiele, & Tinne Tuytelaars, eds. Computer Vision – ECCV 2014. Lecture Notes in Computer Science. Springer International Publishing, 818 – 833.

[2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner ( 1998 ) : Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 11 (November 1998), 2278 – 2324.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton ( 2012 ) : ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger, eds. Advances in Neural Information Processing Systems 25. Curran Associates, Inc., 1097 – 1105.

[4] Visual Object Classes Challenge 2012 VO(2012).

[5] 陈希孺,2009.概率论与数理统计[M].中国科学技术大学出版社.

[6] Dominic Masters, Carlo Luschi,Revisiting Small Batch Training for Deep Neural Networks, arXiv:1804.07612v1.

[7]BoydS,Vandenberghel,2018.Introductiontoappliedlinearalgebra: vectors,matrices,andleast squares[M/OL]. Cambridgeuniversitypress. <http://vmls-book.stanford.edu/>.

[8] Cover T M, Thomas J A, 2006. Elements of information theory[M/OL]. 2nd edition. Wiley. <http://www.elementsofinformationtheory.com/>.

[9] MacKay D J C, 2003. Information theory, inference, and learning algorithms[M]. Cambridge University Press.

[10] NocedalJ,WrightSJ,2006. Numericaloptimization[M]. 2ndedition. Springer.

[11] Rasmussen C E, 2003. Gaussian processes in machine learning[C/OL]//Bousquet O, von Luxburg U, Rätsch G. Lecture Notes in Computer Science: volume 3176 Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures. Springer: 63-71. [https://doi.org/10.1007/978-3-540-28650-9\\_4](https://doi.org/10.1007/978-3-540-28650-9_4).

[12] Strang G, 2014. Differential equations and linear algebra[M/OL]. Wellesley-Cambridge Press. <http://math.mit.edu/dela>.

[13] StrangG,2016. Introductiontolinearalgebra[M/OL]. 5thedition. Wellesley-CambridgePress. <http://math.mit.edu/linearalgebra>.

[14] 陈希孺,2009.数理统计学教程[M].中国科学技术大学出版社.

[15] ThomasGB,WeirMD,HassJ,etal.,2005. Thomas'calculus[M]. Addison-Wesley.

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例

2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

- 3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例
- 4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比
- 5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的
- 6.红色文字表示文字复制部分;绿色文字表示引用部分
- 7.本报告单仅对您所选择比对资源范围内检测结果负责



 [amlc@cnki.net](mailto:amlc@cnki.net)

 <http://check.cnki.net/>

 <http://e.weibo.com/u/3194559873/>

CNKI科研诚信管理系统研究中心