

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。深度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有着深度神经网络的痕迹。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](#_Toc40811825)

[1 引言 6](#_Toc40811826)

[1.1 图像识别背景意义 6](#_Toc40811827)

[1.2 卷积神经网络背景发展 6](#_Toc40811828)

[2 神经网络基础 7](#_Toc40811829)

[2.1 前馈神经网络 7](#_Toc40811830)

[2.2 深度神经网络的激活函数 8](#_Toc40811831)

[2.2.1 Sigmoid 函数 8](#_Toc40811832)

[2.2.2 双曲正切函数 tanh 9](#_Toc40811833)

[9](#_Toc40811834)

[2.2.3 线性整流函数（Rectified Linear Unit, ReLU） 10](#_Toc40811835)

[2.3 反向传播算法 11](#_Toc40811836)

[2.4 常见优化算法 14](#_Toc40811837)

[2.4.1 标准梯度下降算法 14](#_Toc40811838)

[2.4.2 批量梯度下降法(Batch Gradient Descent) 14](#_Toc40811839)

[2.4.3 随机梯度下降法(Stochastic Gradient Descent) 15](#_Toc40811840)

[2.4.4 动量优化法 16](#_Toc40811841)

[2.4.5 AdaGrad算法 16](#_Toc40811842)

[2.4.6 RMSProp算法 17](#_Toc40811843)

[2.4.7 AdaDelta算法 18](#_Toc40811844)

[2.4.8 Adam算法 18](#_Toc40811845)

[3 卷积神经网络相关理论 19](#_Toc40811846)

[3.2 重要影响力网络 19](#_Toc40811847)

[3.3 卷积神经网络组成 20](#_Toc40811848)

[3.3.1 卷积层 21](#_Toc40811849)

[3.3.2 池化层 22](#_Toc40811850)

[3.3.3 全连接层 23](#_Toc40811851)

[4 解决过拟合 23](#_Toc40811852)

[4.1正则化 23](#_Toc40811853)

[4.1.1 L1和L2正则化 23](#_Toc40811854)

[4.1.2 Dropout正则化 24](#_Toc40811855)

[4.2 Mini-batch 25](#_Toc40811856)

[4.3 归一化 26](#_Toc40811857)

[4.3.1 Batch Normalization 26](#_Toc40811858)

[4.3.2 Weight Normalization 27](#_Toc40811859)

[4.3.3 Layer Normalization 27](#_Toc40811860)

[5 相关算法及数据集介绍 27](#_Toc40811861)

[5.1 网络结构 28](#_Toc40811862)

[5.1.1 VGG 28](#_Toc40811863)

[5.1.1 VGG-16特点 29](#_Toc40811864)

[5.1.2 块结构 29](#_Toc40811865)

[5.2 模型修改 30](#_Toc40811866)

[5.2.1 输入层 30](#_Toc40811867)

[5.2.2 模型深度 30](#_Toc40811868)

[5.2.3 Keras快速建立模型 30](#_Toc40811869)

[5.3 数据增强 31](#_Toc40811870)

[5.3.1 常用技术 31](#_Toc40811871)

[6 卷积神经网络应用 33](#_Toc40811872)

[6.1 图像预处理 34](#_Toc40811873)

[6.1.1 归一化 34](#_Toc40811874)

[6.1.2 独热编码 34](#_Toc40811875)

[6.1.3 选择mini-batch大小 35](#_Toc40811876)

[6.1.4 数据增强 35](#_Toc40811877)

[6.2 优化及参数设定 35](#_Toc40811878)

[6.2.1 学习率 36](#_Toc40811879)

[6.2.2 训练轮次 36](#_Toc40811880)

[6.3 实验环境 37](#_Toc40811881)

[6.4 实验数据分析 37](#_Toc40811882)

[6.4.1 整体训练过程 38](#_Toc40811883)

[6.4.2 初步尝试 39](#_Toc40811884)

[6.4.3 调整学习率 39](#_Toc40811885)

[6.4.4 使用dropout 40](#_Toc40811886)

[6.4.5 使用图像增强 41](#_Toc40811887)

[6.4.6 L2正则化解决过拟合问题 42](#_Toc40811888)

[6.4.7 结果分析 42](#_Toc40811889)

[结语 44](#_Toc40811890)

[参考文献 47](#_Toc40811891)

# 引言

## 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。

## 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。

卷积神经网络的流行离不开一个重要的算法，误差反向传播算法（Back Propagation），该算法使得神经网络的训练成为了可能。证明了神经网络的理论正确性。

历史上的第一个实用卷积神经网络是LeNet-5由Yann LeCun等人提出，该卷积神经网络提出用于手写数字识别，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。

# 神经网络基础

## 2.1 前馈神经网络

深度前馈网络（deep feedforward network），也叫做前馈神经网络（feedforward neural network）或者是多层感知机（multilayer perceptron，MLP）。对深度前馈网络，从感知机的角度可能更加容易。感知机又被称为最简单的神经网络，多层感知机就是在感知机的基础上设计添加了更多层。

前馈网络的目标是模拟近似一个函数，简单来是说对于目标函数，网络学习的最终目标是训练输出一个函数，并且尽可能去接近。举个例子，一个分类器会将输入映射到一个类别，前馈网络会定义一个映射，然后通过学习到参数的值，使得新的映射能够得到最正确的函数映射。

深度前馈网络，是由输入层、输出层、隐藏层和各层之间的连接构成，其中隐藏层会根据模型的大小以及复杂程度设计成数量任意的多层，各层之间的连接就是实际表示特征的权重。“深度”是指一个神经网络模型通常是由非常多层组成的，设计的层数越多，深度越大，得到模型也越复杂。对于前馈网络中的“前馈”，因为网络中的数据流向是单向的， 只会按照输入层->隐藏层->输出层的顺序流动，即数据流经当前一层处理后，只会作为下一层的输入流入出，而不会对上一层有任何影响和反馈。

## 2.2 深度神经网络的激活函数

激活函数是神经网络中一个至关重要的概念，决定了某个神经元是否被激活，判断该神经元获得的信息是否有用，并决定该保留还是该去掉此神经元。

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：

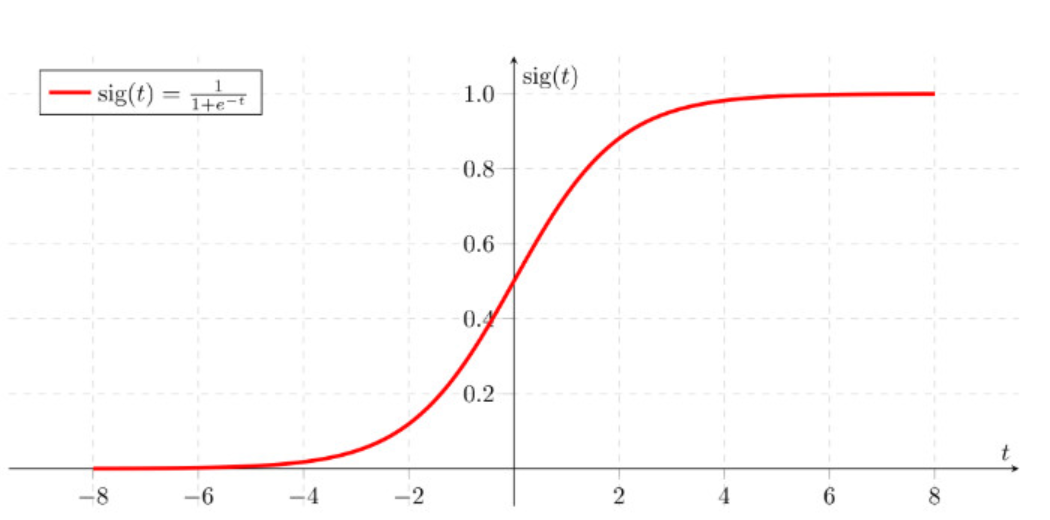


图2.1 sigmoid函数图像

sigmoid函数又叫做Logistic函数，常用于隐层神经元的输出，取值范围是(0,1)，它可以将一个任意实数映射到(0,1)的区间，常常用来做二分类。在特征相差复杂或是相差比较小时效果比较好。Sigmoid函数f(x)具有如下基本性质：

* 定义域为(-∞,+∞)
* 值域为(-1,1), 为有界函数
* 函数f(x)在定义域内为连续和光滑函数
* 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，仅由于历史原因罗列在此。Sigmoid 函数计算量大，会导致梯度消失问题且不以零为中心，通常在二进制分类问题中才会使用。

### 双曲正切函数 tanh

### 

函数如下图所示：

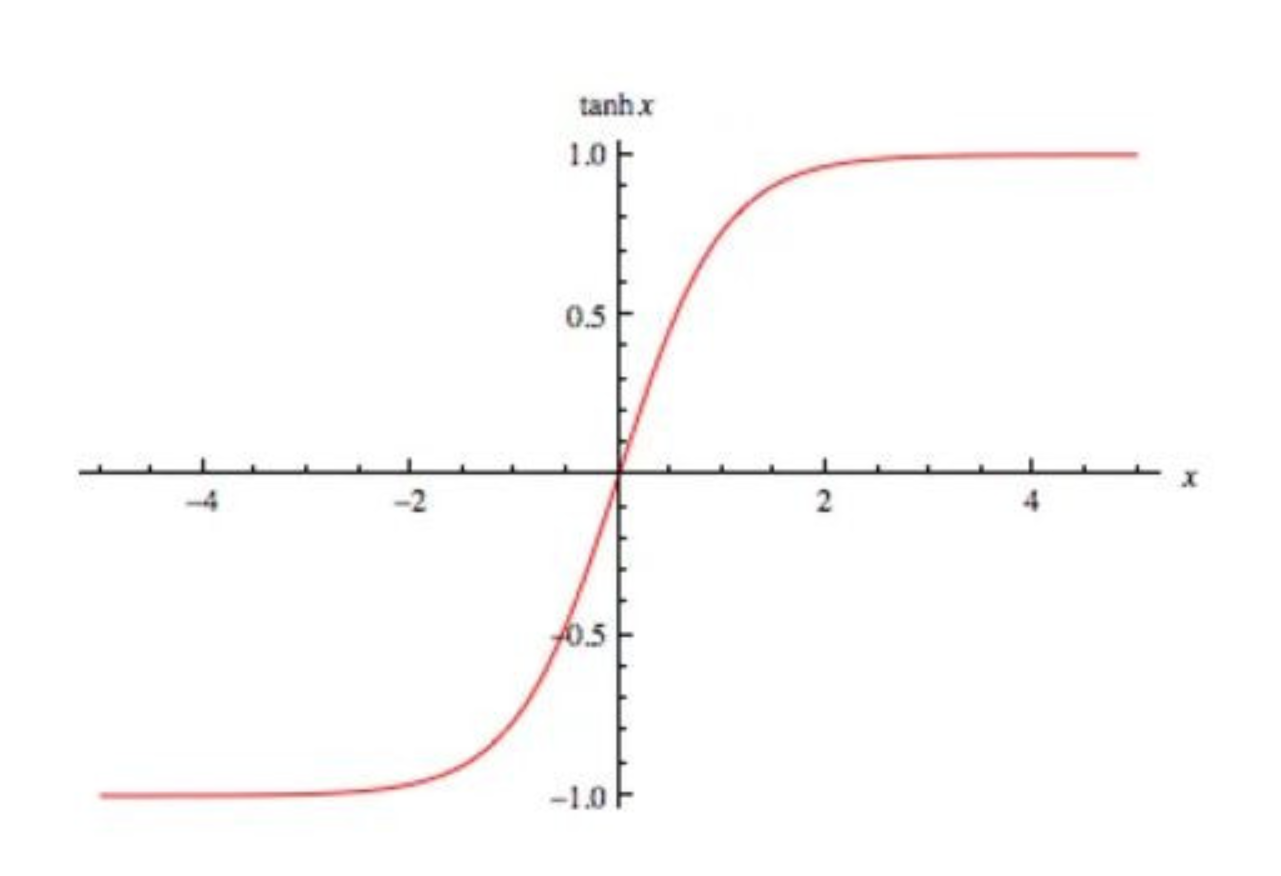


图2.2 tanh函数图像

双曲正切函数是双曲函数的一种，它是双曲正弦函数与双曲余弦函数的商。在人工智能领域，函数是一种常见的激活函数，取值范围为（-1，1），在特征相差较大时的效果会比较好，在循环过程中会能够不断地扩大特征效果。

相比于 Sigmoid 函数，它仅仅解决了以零为中心这个问题。而对比ReLU函数，由于幂运算的问题，tanh函数计算量比较大，收敛速度缓慢。同时它也存在着梯度消失的问题。

### 线性整流函数（Rectified Linear Unit, ReLU）

函数如下图所示：

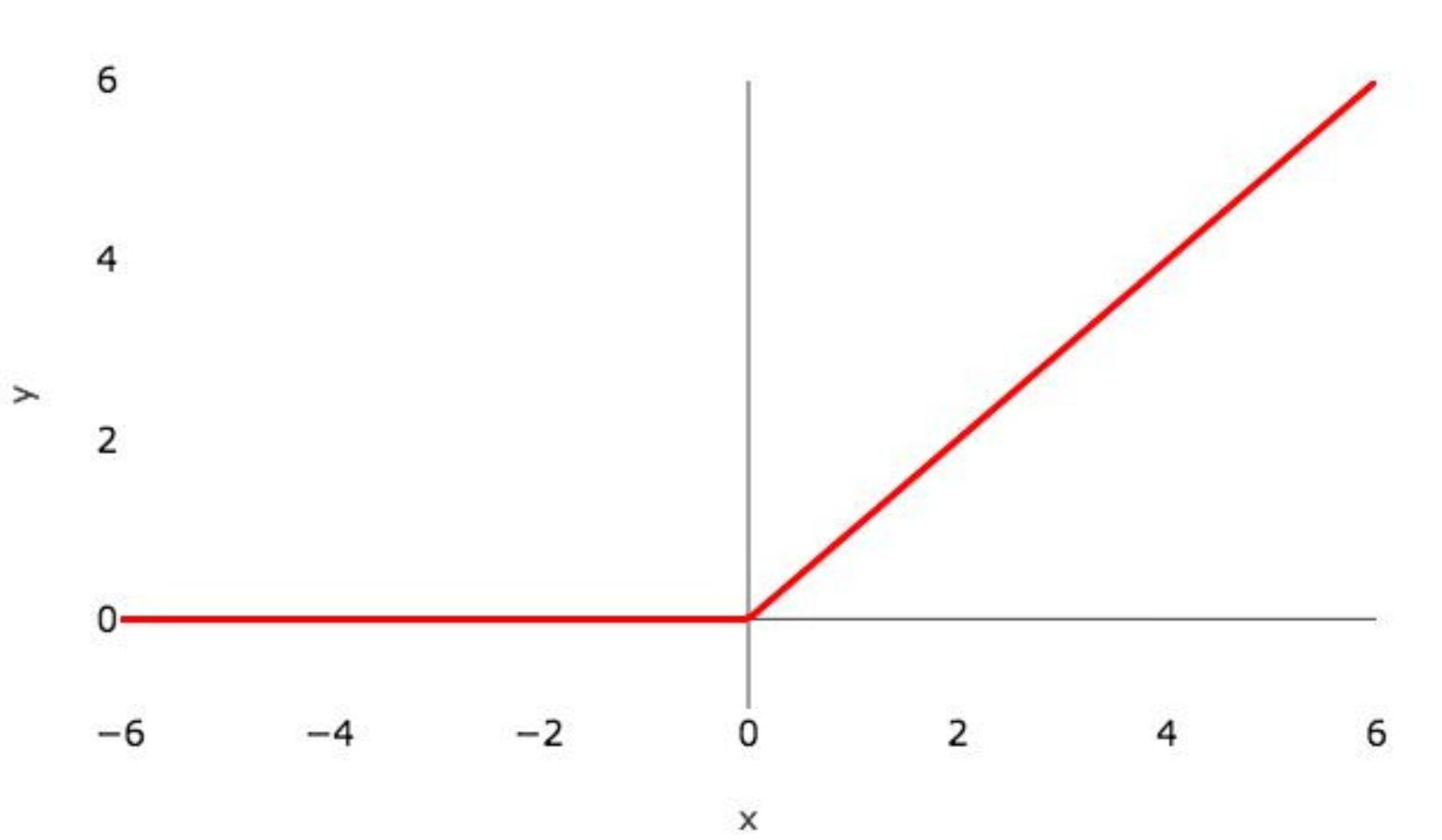


图2.3 ReLU函数图像

该函数又称修正线性单元。

优点就是：

(1) x>0 时，梯度恒为1，无梯度耗散问题，收敛快；

(2)增大了网络的稀疏性。当x<0时，该层的输出为0，训练完成后为0的神经元越多，稀疏性越大，提取出来的特征就越具有代表性，泛化能力越强。即得到同样的效果，真正起作用的神经元越少，网络的泛化性能越好；

(3)运算量很小,训练时间少。

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，即当输入的是负数时，输出都会变成零，这样就导致一些节点完全死掉，神经元也不会进行学习。

* + 1. Leaky ReLU 和 Parametric ReLU 函数

函数如下图所示：

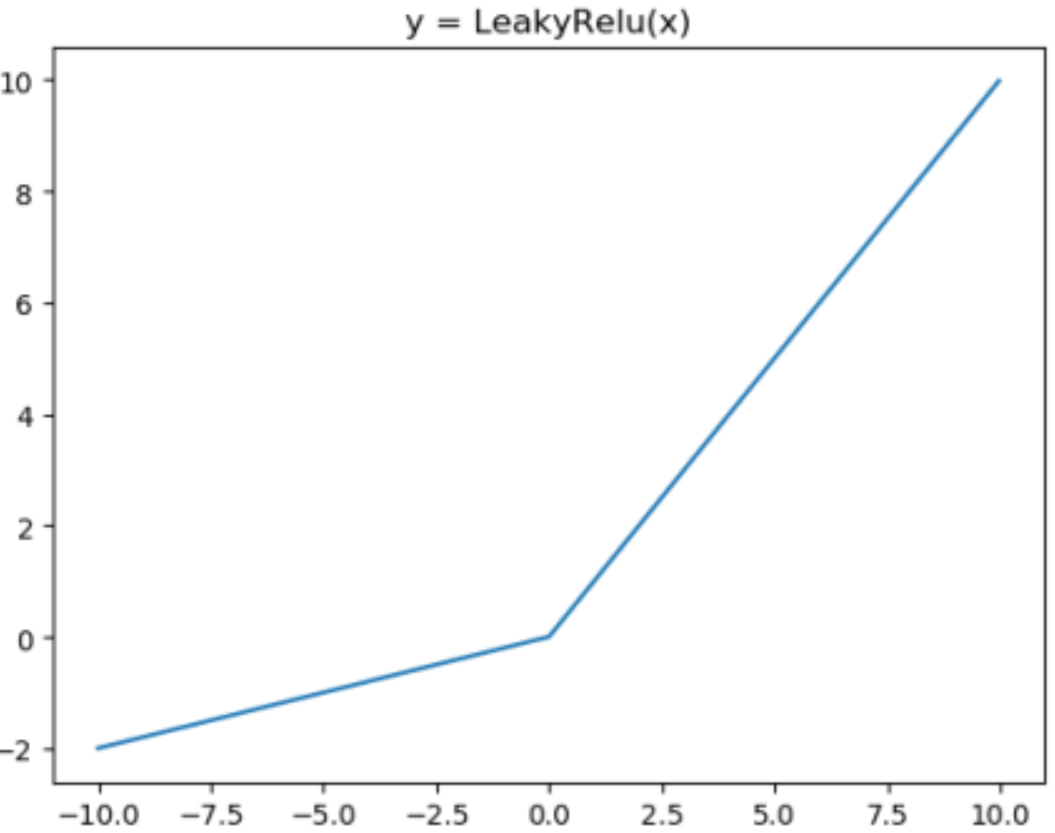


图2.4 Leaky ReLU函数图像

这里α是一个通常设置为 0.01 的超参数。显然，Leaky ReLU 在一定程度上解决了dying ReLU这一问题。请注意，如果我们将α设为 1，那么 Leaky ReLU 将成为线性函数 f（x）=x，并且没有任何用处。

因此，α的值永远不会设置为趋近于 1 的值。如果我们分别将α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

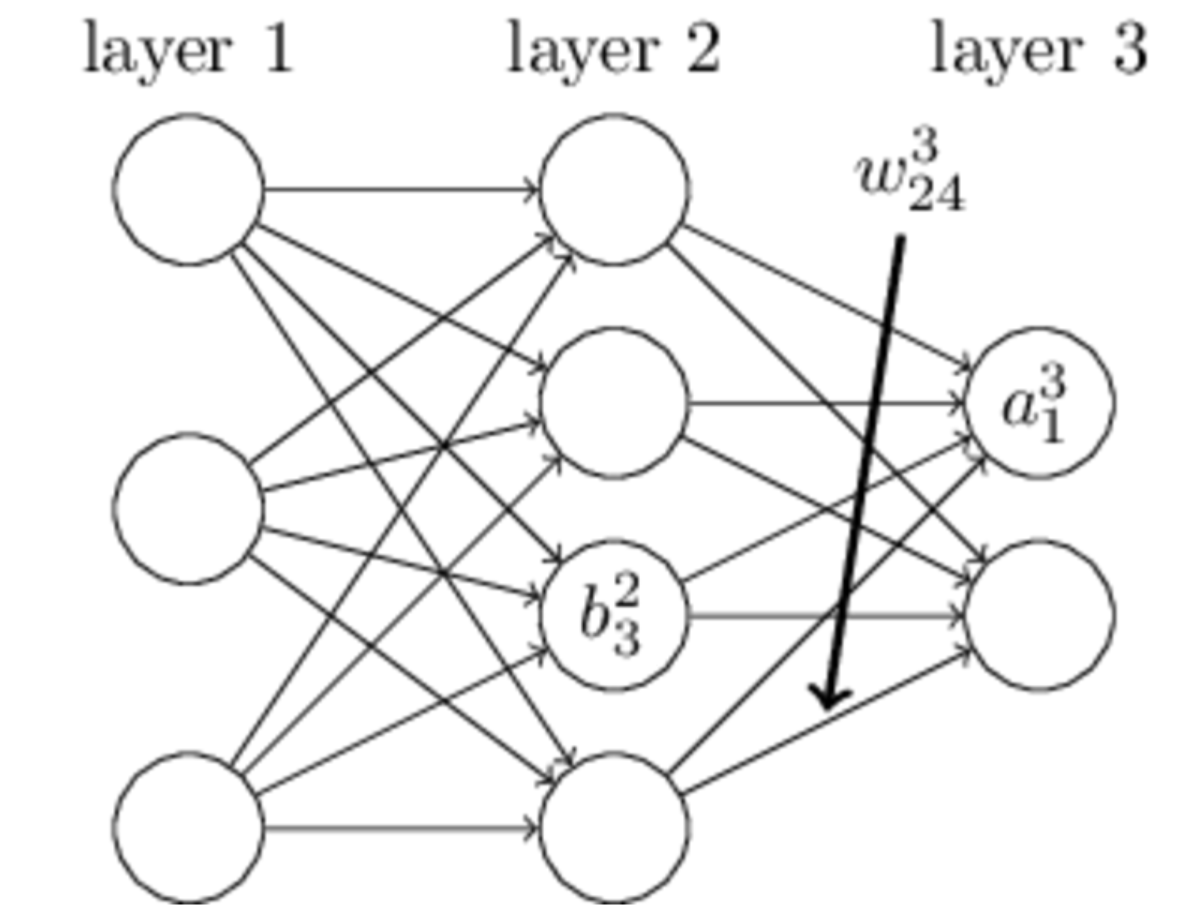


图2.5 3层神经网络

反向传播算法（Back propagation Algorithm，简称BP算法）是神经网络训练的重要思想基础。

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。

因为需要不断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），所以需要用误差函数（Error function，也称损失函数，loss function）来衡量训练集所有样本最终预测值与实际值的误差大小。

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。

梯度下降算法是其中一种使误差函数最小化的算法，大部分深度学习模型都是采用梯度下降算法来进行优化训练。给定一组函数参数，梯度下降从一组初始参数值开始，迭代移向一组使损失函数最小化的参数值。这种迭代最小化是使用微积分实现的，在梯度的负方向上采取渐变更改。使用梯度下降的典型例子是线性回归。随着模型迭代，损失函数逐渐收敛到最小值。

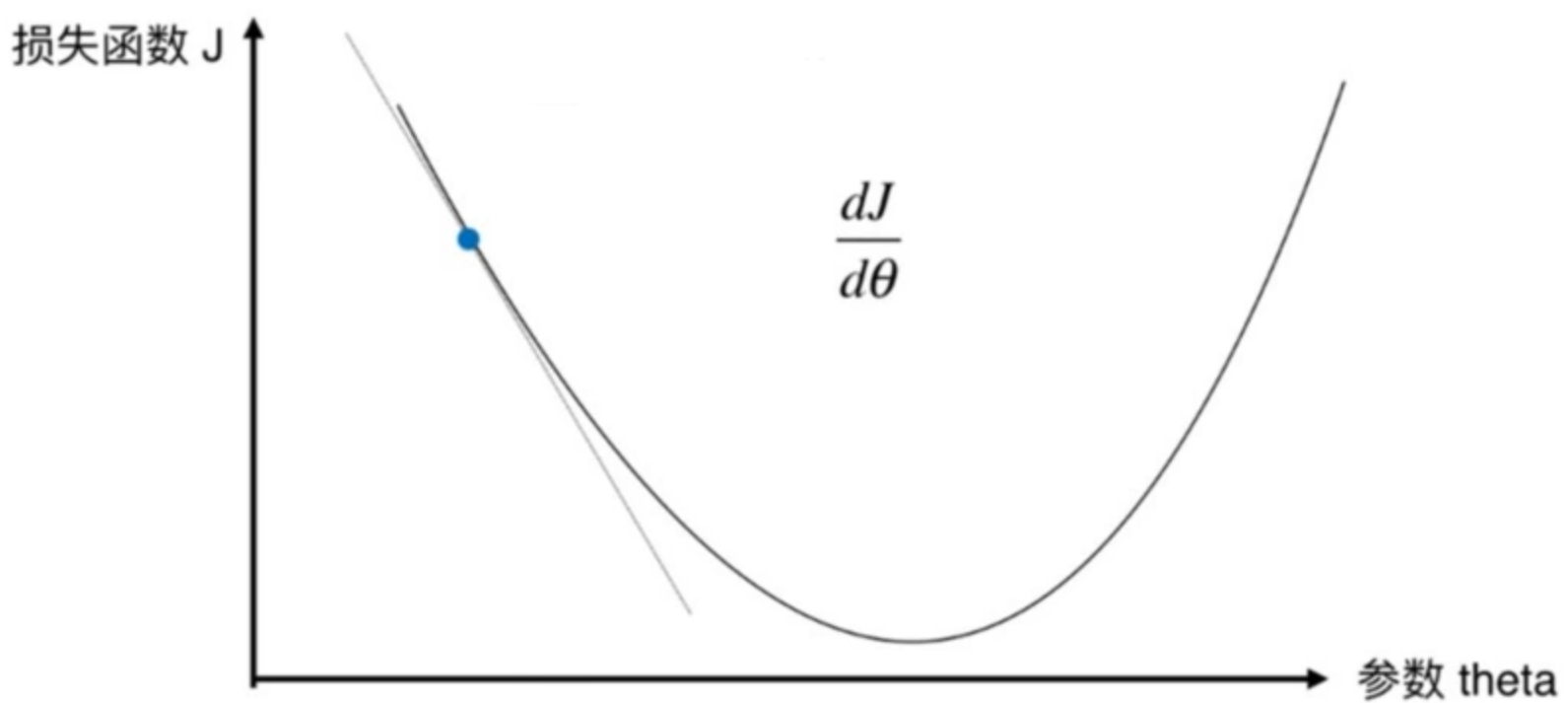
由于梯度表达的是函数在某点变化率最大的方向，通过计算偏导数得到，所以使用梯度下降方式，会极大地加快学习进程。

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，直到出错的最初层为止。

这个过程就是反向传播算法，又称 BP 算法，它将输出层的误差反向逐层传播，通过计算偏导数来更新网络参数使得误差函数最小化，从而让网络输出得出符合预期的输出。

## 2.4 常见优化算法

### 2.4.1 标准梯度下降算法

假定要学习训练的模型的参数为W，代价函数为，则代价函数关于模型参数的偏导数也就是相关梯度为，学习率为，则使用梯度下降法更新参数为：

其中，表示t时刻的模型参数。

从表达式来看，模型参数的更新调整，与代价函数关于模型参数的梯度有关，即沿着梯度的方向不断减小模型参数，从而最小化代价函数。

基本策略可以理解为“在有限视距内寻找最快路径下山”，因此每走一步，参考当前位置最陡的方向(即梯度)进而迈出下一步。

标准梯度下降法主要有两个缺点:

1. 训练速度慢：每走一步都要要计算调整下一步的方向，下山的速度变慢。在应用于大型数据集中，每输入一个样本都要更新一次参数，且每次迭代都要遍历所有的样本。会使得训练过程及其缓慢，需要花费很长时间才能得到收敛解。
2. 容易陷入局部最优解：由于是在有限视距内寻找下山的方向。当陷入平坦的洼地，会误以为到达了山地的最低点，从而不会继续往下走。所谓的局部最优解就是鞍点。落入鞍点，梯度为0，使得模型参数不在继续更新。

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

假设批量训练样本总数为n，每次输入和输出的样本分别为，，模型参数为W，代价函数为J(W)，每输入一个样本i代价函数关于W的梯度为，学习率为ηt，则使用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。

从表达式来看，模型参数的调整更新与全部输入样本的代价函数的和（即批量/全局误差）有关。即每次权值调整发生在批量样本输入之后，而不是每输入一个样本就更新一次模型参数。这样就会大大加快训练速度。

基本策略可以理解为，在下山之前掌握了附近的地势情况，选择总体平均梯度最小的方向下山。

批量梯度下降法比标准梯度下降法训练时间短，且每次下降的方向都很正确。

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

对比批量梯度下降法，假设从一批训练样本n中随机选取一个样本。模型参数为W，代价函数为J(W)，梯度为，学习率为，则使用随机梯度下降法更新参数表达式为：

其中，表示随机选择的一个梯度方向，表示t时刻的模型参数。

，这里虽然引入了随机性和噪声，但期望仍然等于正确的梯度下降。

基本策略可以理解为随机梯度下降像是一个盲人下山，不用走一步计算一次梯度，但是他总能下到山底，只不过过程会显得弯弯扭扭。

虽然SGD需要走很多步的样子，但是对梯度的要求很低（计算梯度快）。而对于引入噪声，大量的理论和实践工作证明，只要噪声不是特别大，SGD都能很好地收敛。应用大型数据集时，训练速度很快。比如每次从百万数据样本中，取几百个数据点，算一个SGD梯度，更新一下模型参数。相比于标准梯度下降法的遍历全部样本，每输入一个样本更新一次参数，要快得多。

SGD在随机选择梯度的同时会引入噪声，使得权值更新的方向不一定正确。此外，SGD也没能单独克服局部最优解的问题。

### 2.4.4 动量优化法

动量优化方法是在梯度下降法的基础上进行的改变，具有加速梯度下降的作用。一般是标准动量优化方法Momentum。

使用动量(Momentum)的随机梯度下降法(SGD)，主要思想是引入一个历史梯度信息动量来加速SGD。

训练集中取一个大小为n的小批量样本，对应的真实值分别为，则Momentum优化表达式为：

其中，表示t时刻积攒的加速度。表示动力的大小，一般取值为0.9。表示t时刻模型参数。

动量主要解决SGD的两个问题：一是随机梯度的方法（引入的噪声）；二是Hessian矩阵病态问题（可以理解为SGD在收敛过程中和正确梯度相比来回摆动比较大的问题）。解决策略为：由于当前权值的改变会受到上一次权值改变的影响，类似于小球向下滚动的时候带上了惯性。这样可以加快小球向下滚动的速度。

### 2.4.5 AdaGrad算法

自适应学习率优化算法针对于机器学习模型的学习率，传统的优化算法要么将学习率设置为常数要么根据训练次数调节学习率。极大忽视了学习率其他变化的可能性。然而，学习率对模型的性能有着显著的影响，因此需要采取一些策略来想办法更新学习率，从而提高训练速度。

目前的自适应学习率优化算法主要有：AdaGrad算法，RMSProp算法，Adam算法以及AdaDelta算法。

AdaGrad算法，独立地适应所有模型参数的学习率，缩放每个参数反比于其所有梯度历史平均值总和的平方根。具有代价函数最大梯度的参数相应地有个快速下降的学习率，而具有小梯度的参数在学习率上有相对较小的下降。

AdaGrad算法优化策略一般可以表示为：

假定一个多分类问题，i表示第i个分类，t表示第t迭代同时也表示分类i累计出现的次数。η0 表示初始的学习率取值一般为0.01，ϵ是一个取值很小的数（一般为1e-8）为了避免分母为0。Wt 表示t时刻即第t迭代模型的参数， 表示t时刻，指定分类i，代价函数J(⋅) 关于W的梯度。

从表达式可以看出，对出现比较多的类别数据，AdaGrad给予越来越小的学习率，而对于比较少的类别数据，会给予较大的学习率。因此AdaGrad适用于数据稀疏或者分布不平衡的数据集。

AdaGrad 的主要优势在于不需要人为的调节学习率，它可以自动调节；缺点在于，随着迭代次数增多，学习率会越来越小，最终会趋近于0。

### 2.4.6 RMSProp算法

RMSProp算法修改了AdaGrad的梯度积累为指数加权的移动平均，使得其在非凸设定下效果更好。

RMSProp算法的一般策略可以表示为：

其中，表示t时刻即第t迭代模型的参数， 表示t次迭代代价函数关于W的梯度大小，表示前t次的梯度平方的均值。α 表示动力（通常设置为0.9），η0表示全局初始学习率。ϵ是一个取值很小的数（一般为1e-8）为了避免分母为0。

RMSProp借鉴了AdaGrad的思想，观察表达式，分母为。由于取了个加权平均，避免了学习率越来越低的问题，而且能自适应地调节学习率。

RMSProp算法在经验上已经被证明是一种有效且实用的深度神经网络优化算法。目前它是深度学习从业者经常采用的优化方法之一。

### 2.4.7 AdaDelta算法

首先，Adam中动量直接并入了梯度一阶矩（指数加权）的估计。其次，相比于缺少修正因子导致二阶矩估计可能在训练初期具有很高偏置的RMSProp，Adam包括偏置修正，修正从原点初始化的一阶矩（动量项）和（非中心的）二阶矩估计。

AdaDelta算法策略可以表示为：

其中，和分别为一阶动量项和二阶动量项。β1,β2 为动力值大小通常分别取0.9和0.999；分别为各自的修正值。表示t时刻即第t迭代模型的参数， 表示t次迭代代价函数关于W的梯度大小；ϵ是一个取值很小的数（一般为1e-8）为了避免分母为0。

### 2.4.8 Adam算法

Adam中动量直接并入了梯度一阶矩估计。其次，相比于缺少修正因子导致二阶矩估计可能在训练初期具有很高偏置的RMSProp，Adam包括偏置修正，修正从原点初始化的一阶矩（动量项）和（非中心的）二阶矩估计。

Adam算法策略表示为：

其中，和分别为一阶动量项和二阶动量项。和为超参数，大小通常分别取0.9和0.999；，分别为各自的修正值。表示t时刻即第t迭代模型的参数，表示t次迭代代价函数关于W的梯度大小；ϵ是一个取值很小的数（一般为1e-8）避免分母为0。

# 3 卷积神经网络相关理论

## 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：

* 训练出当前最大规模的卷积神经网络，此前LeNet-5网络仅为3个卷积层及1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。
* 通过众多的skills（dropout、RELU、Data Augmentation）解决深层神经网络的过拟合问题，使得该网络在60 million参数的情况下依旧能很好收敛。这些方法现在已经成为了CNN不可或缺的一部分。

1. GoogLeNet

GoogLeNet是2014年Christian Szegedy提出的一种全新的神经网络结构，在此之前的AlexNet、VGG等结构都是通过增加网络的深度，来获得更好的训练效果，但是层数的增加有时候会带来很多负作用，例如梯度消失、梯度爆炸等。GoogLeNet的提出则从另一种不同的角度来提升训练的效果：它能更高效的利用计算资源，在相同的计算量下能提取到更丰富的特征，从而帮助提升训练结果。

GoogLeNet的核心是inception模块，inception模块包含几种不同大小的卷积，即1x1卷积，3x3卷积和5x5卷积，还包括一个3x3的max pooling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，也是下一个模块的输入。整个网络由很多的inception模块组成。inception结构的主要贡献有两个：一是使用1x1的卷积来进行升降维；二是在多个尺寸上同时进行卷积再聚合。

1x1卷积能够在相同尺寸的感受野上叠加更多的卷积操作，能够提取到更加丰富的特征，而且使用1x1卷积还起到了降低维度的效果，减少了计算复杂度。

1. 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。

## 卷积神经网络组成

卷积神经网络是前馈神经网络的一种，卷积神经网络主要由卷积层，池化层，全连接层构成。运用于图像处理领域的卷积神经网络通常有固定的层次结构，由多个卷积层和末尾的全连接层组成，在部分或者是全部的卷积层后有池化层，最后的池化层连接一个平展层。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校算法更好的效果。卷积神经网络一般采用反向传播算法进行训练，对比其他的神经网络，卷积神经网络需要训练的参数数量较少，是当前广泛使用的一种网络。

### 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：

1. 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。

在计算机图像领域中，图像中的某一块区域，不同像素之间的相关性与像素之间的距离同样有关，一般的经验是距离较近的像素间相关性较强，距离较远的像素则相关性就较弱。通俗话来说，狗的图片，不会因为狗所在图片的位置，而影响到这是一张狗的图片，组成狗的像素和背景像素之间的想关心较弱。

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。这样的连接也很容易产生过拟合，参数浪费。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。

1. 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。

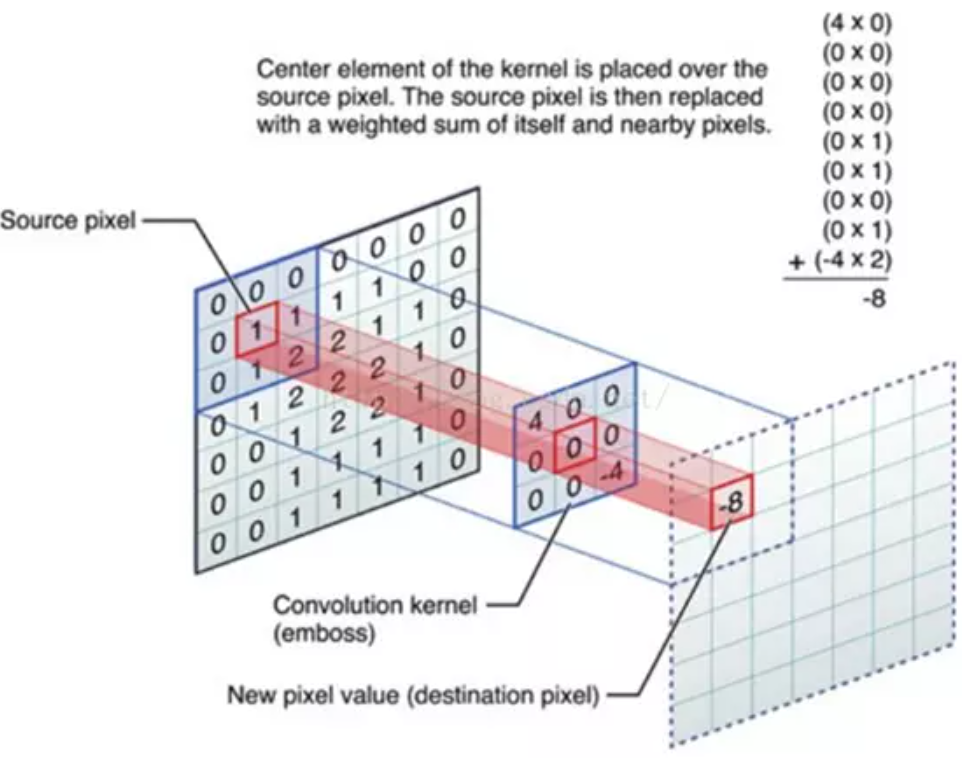


图3.1 按步卷积操作

### 池化层

池化层的作用主要是对特征图进行向下采样，降低输入数据的空间尺寸。以此减少空间冗余和计算成本。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。

### 全连接层

高级特征一般是与位置有关的，比如一张人脸图片，眼睛和嘴位置不同，那么在处理到高层特征时，不同位置就需要用不同的神经网络权重，位置信息变得十分重要，这时候卷积层就不能胜任了，就需要用全连接层。

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。在卷积神经网络中，一般用作分类器使用。卷积层和池化层将原始数据映射到隐形特征空间，而全连接层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。

# 解决过拟合

## 4.1正则化

在图2中展示了神经网络拟合一种示例，如果要将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。

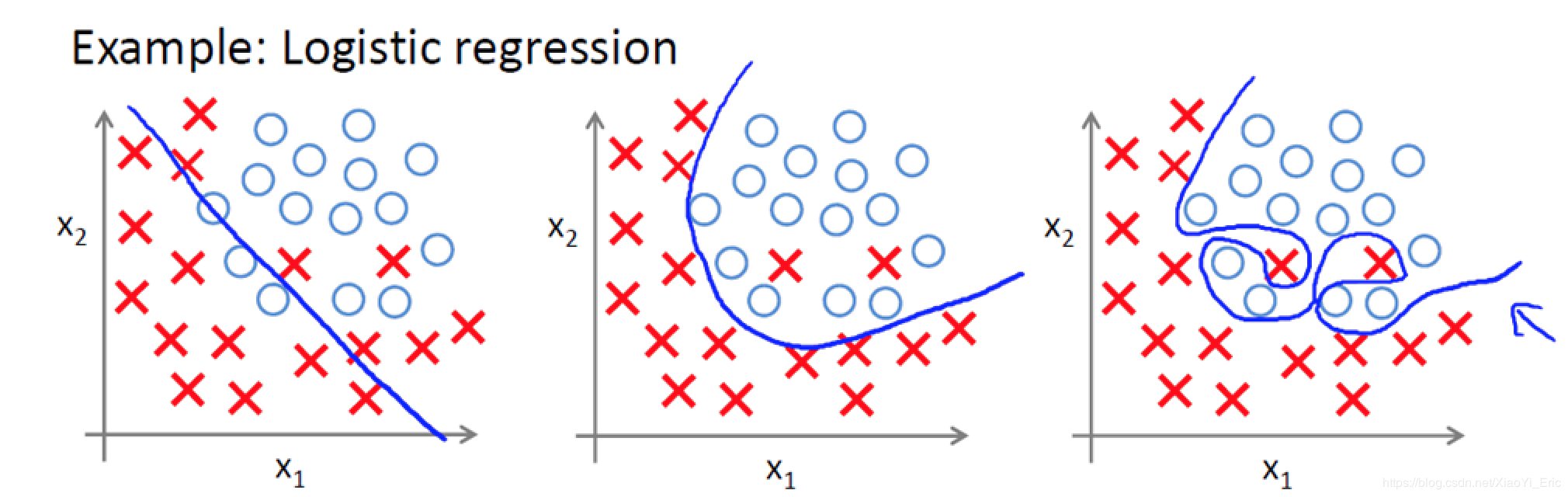


图4.1 训练拟合情况

### 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，在训练集很少时通常通过模型集成的方法，也就是训练多个模型组合，然而多个模型一起训练费时费力。而Dropout的出现解决了这个问题。

Dropout可以作为训练深度神经网络的一种trick供选择。在每个训练批次中，通过忽略一半的特征检测器（让一半的隐层节点值为0），可以明显地减少过拟合现象。这种方式可以减少特征检测器（隐层节点）间的相互作用，检测器相互作用是指某些检测器依赖其他检测器才能发挥作用。

我们在前向传播的时候，让某个神经元的激活值以一定的概率p停止工作，这样可以使模型泛化性更强，因为它不会太依赖某些局部的特征，如图3所示。

1. 求平均

标准的模型即没有dropout，我们用相同的训练数据去训练5个不同的神经网络，一般会得到5个不同的结果，此时我们可以采用 “5个结果取均值”或者“多数取胜的投票策略”去决定最终结果。dropout掉不同的隐藏神经元就类似在训练不同的网络，随机删掉一半隐藏神经元导致网络结构已经不同，整个dropout过程就相当于对很多个不同的神经网络取平均。

1. 减少神经元之间复杂的共适应关系

因为dropout程序导致两个神经元不一定每次都在一个dropout网络中出现。这样权值的更新不再依赖于有固定关系的隐含节点的共同作用，阻止了某些特征仅仅在其它特定特征下才有效果的情况 。迫使网络去学习更加鲁棒的特征。

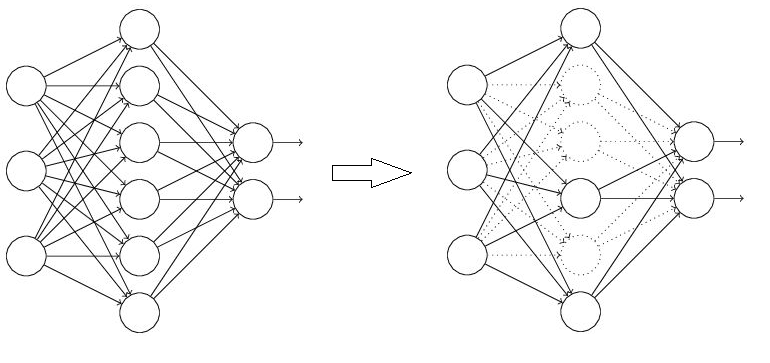


图4.2 Dropout操作

## 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。

1. 增大batch的优点

内存利用率提高了，大矩阵乘法的并行化效率提高。跑完一次 epoch（全数据集）所需的迭代次数减少，对于相同数据量的处理速度进一步加快。在一定范围内，一般来说 batch-size 越大，其确定的下降方向越准，引起训练震荡越小。

1. 盲目增大的坏处

内存利用率提高了，但是内存容量可能不够。跑完一次 epoch（全数据集）所需的迭代次数减少，要想达到相同的精度，其所花费的时间大大增加了，从而对参数的修正也就显得更加缓慢。batch-size 增大到一定程度，其确定的下降方向已经基本不再变化，这样的梯度下降方式不够灵活，缺乏动量。

## 归一化

深度神经网络涉及到很多层的叠加，而每一层的参数更新会导致上层的输入数据分布发生变化，通过层层叠加，高层的输入分布变化会非常剧烈。虽然神经网络的各层的输入信号分布不同，但最终“指向”的样本标记是不变的，即边缘概率不同而条件概率一致。

为了降低分布变化的影响，可使用归一化策略Normalization，把数据分布映射到一个确定的区间。

神经网络中，常用的归一化策略有BN(Batch Normalization)， WN(Weight Normalization)， LN(Layer Normalization)。

### Batch Normalization

统计机器学习中有一个经典的假设：训练数据和测试数据是满足相同分布的。这是通过训练数据获得的模型能够在测试集上获得好的效果的一个基本保障。

Batch Normalization可以减少神经网络中的covariate shift。covariate shift是指训练集的样本数据和目标样本集分布不一致时，训练得到的模型无法很好的泛化。简单来说就是当你训练好了一个函数，输入的分布变了，这个函数就无法处理了。Batch Normalization标准化输入，从而输入信息的均值和方差，以此消除covariate shift加速网络收敛。

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。

### Weight Normalization

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

Weight Normalization通过重写深度学习网络的权重W的方式来加速深度学习网络参数收敛，没有引入mini-batch的依赖。

### Layer Normalization

LN可以将数据分布拉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对一个神经元的batch所有样本进行标准化，layer normalization对一个样本同一层所有神经元进行标准化。

# 相关算法及数据集介绍

CIFAR-10是一个彩色图像数据集。CIFAR-10 是由Hinton 的学生Alex Krizhevsky和Ilya Sutskever整理的一个用于识别普适物体的小型数据集。一共包含10个类别的RGB 彩色图片：飞机（airplane）、汽车（automobile）、鸟类（bird）、猫（cat）、鹿（deer）、狗（dog）、蛙类（frog）、马（horse）、船（ship）和卡车（truck）。每个图片的尺寸为32 × 32 ，每个类别有6000个图像，数据集中一共有50000张训练图片和10000 张测试图片。

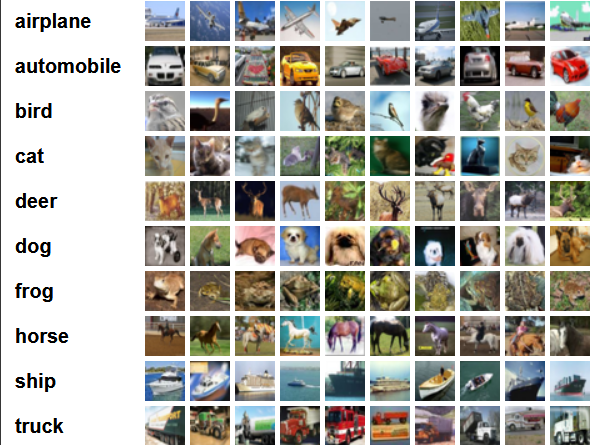


图5.1 CIFAR-10数据集

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，其名称来源于作者所在的牛津大学视觉几何组(Visual Geometry Group)的缩写。该模型参加2014年的 ImageNet图像分类与定位挑战赛，取得了优异成绩：在分类任务上排名第二，在定位任务上排名第一。

## 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：

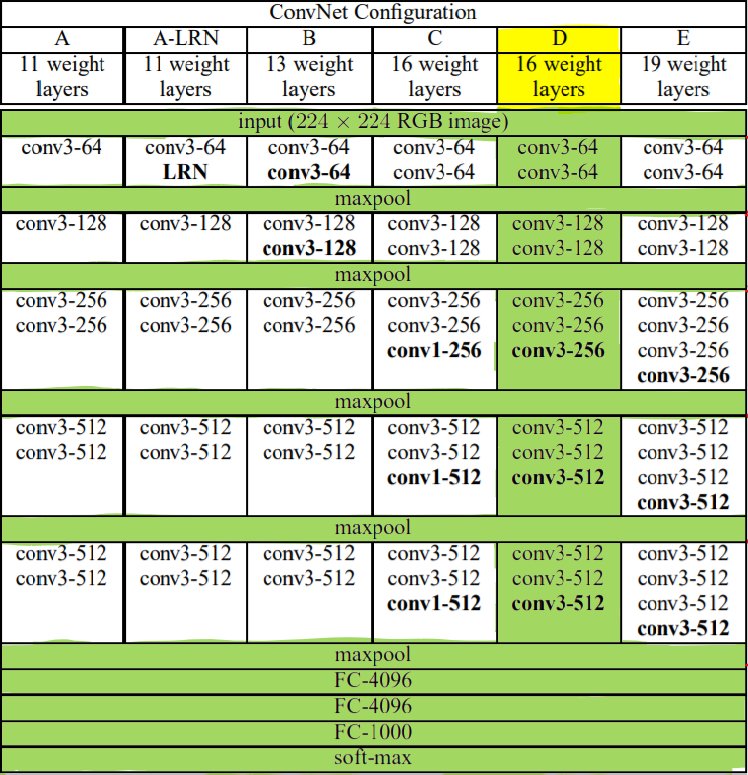


图5.2 VGG网络结构

图中绿色部分即指明了VGG16所采用的结构。针对VGG16进行具体分析发现，VGG16共包含：13个卷积层（Convolutional Layer），3个全连接层（Fully connected Layer），5个池化层（Pool layer）。其中，卷积层和全连接层具有权重系数，因此也被称为权重层，总数目为13+3=16，这即是VGG16中16的来源。

### 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。

### 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。同一个块内卷积层包含的卷积核数量相同。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。

## 模型修改

### 输入层

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

### 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。

### Keras快速建立模型

Keras是python语言中的高级深度学习框架。Keras 的开发重点是支持快速的实验。能够以最小的时延把你的想法转换为实验结果，是做好研究的关键。通过summary函数查看网络结构得到图6。

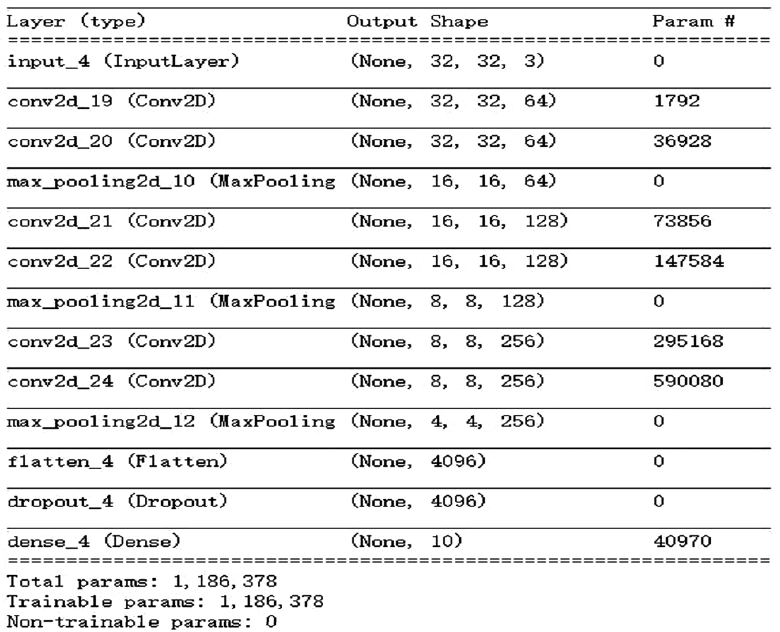


图5.3 改进VGG-16网络结构

## 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。数据不足很容易导致模型过拟合，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，很可能关于位置和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。

### 常用技术

1. 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。

1. 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。

1. 缩放比例

图像可以向外或向内缩放。向外缩放时，最终图像尺寸将大于原始图像尺寸。大多数图像框架从新图像中剪切出一个部分，其大小等于原始图像。

1. 裁剪

翻转和旋转不会改变图像的大小，裁剪会改变图像的大小。在训练的时候会采用随机裁剪的方法，在测试的时候选择随机裁剪中间部分或者不裁剪。值得注意的是，在一些竞赛中进行模型测试，一般都是裁剪输入的多个版本然后将结果进行会合，对预测的提升效果非常明显。

与缩放不同，裁剪只是从原始图像中随机抽样一个部分。然后，将此部分的大小调整为原始图像大小。这种方法通常称为随机裁剪。

1. 移位

移位只涉及沿X或Y方向（或两者）移动图像。

1. 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、亮度和对比度（contrast）等方面。

1. 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，可以对图片上随机选取一块区域并擦除其图像信息。

1. 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、SamplePairing、mixup。SMOTE 全程是 Synthetic Minority Over-sampling Technique 方法，它通过人工合成新样本来处理样本不平衡问题，从而提升分类器性能。

类不平衡现象是很常见的数据分布现象，它是指数据集中各类别数量差距不小。如果样本类别之间相差较大，则会影响分类器的分类效果。假设小样本数据数量非常少，例如仅占总体的 1%，则即使小样本全部识别错误，在loss最小化策略下的分类器识别准确率仍然能够达到 99%，但由于并没有学习小样本的特征，实际生产环境中分类效果就会很差。SMOTE方法，通过插值技术，将每个样本对应到特征空间中的某一点，根据样本不平衡比例确定好一个采样倍率 N。对每一个小样本类样本 (x,y)，按欧氏距离找出 K 个最近邻样本，从中随机选取一个样本点，在特征空间中样本点与最近邻样本点的连线段上随机选取一点作为新样本点。重复上述步骤直到小样本数据的数量达到合适量即可。

Sample Pairing 方法的原理非常简单，从训练集中随机抽取两张图片分别经过基础数据增强操作 (如随机翻转等) 处理后经像素以取平均值的形式叠加合成一个新的样本，标签为原样本标签中的一种。

# 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。

## 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。但是为了能够到达更优秀的训练效果，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。

### 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。

### 独热编码

在分类预测任务中，最后的输出值不会是具体分类的名字，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中工具类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。

### 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点。

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。

### 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。

## 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模型，超参数的选择需要反复的尝试和考量。

### 学习率

学习率参数是梯度下降算法的重要超参数，学习率越低，损失函数的变化速度就越慢。使用小的学习率可以确保我们的模型不错过任何的局部最小值，但是也意味收敛花费的时间变多，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优化器比对实验，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。

### 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，换而言之，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。

## 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。

## 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。

### 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：

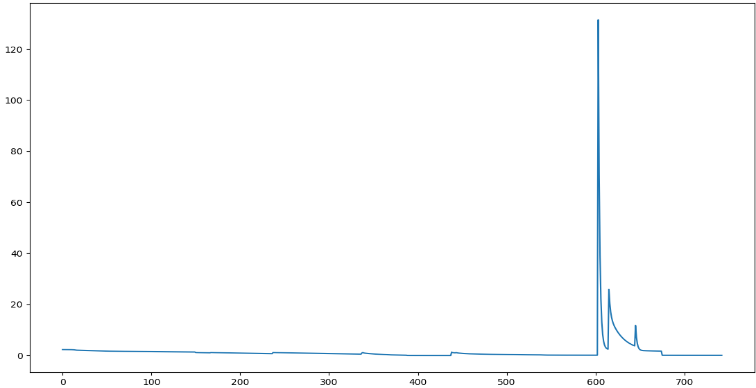
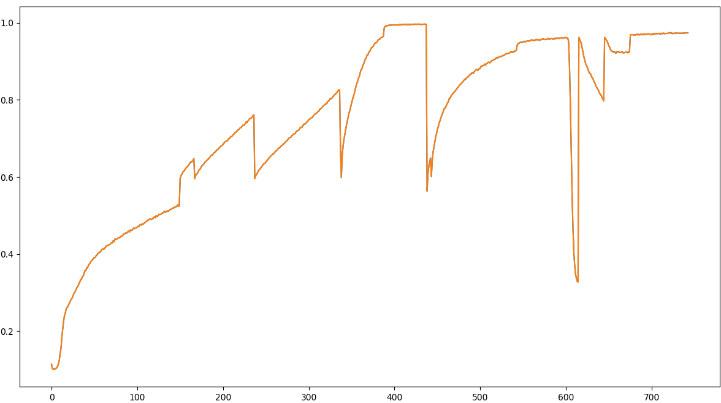


图6.1 训练集acc图

图6.2训练集loss图

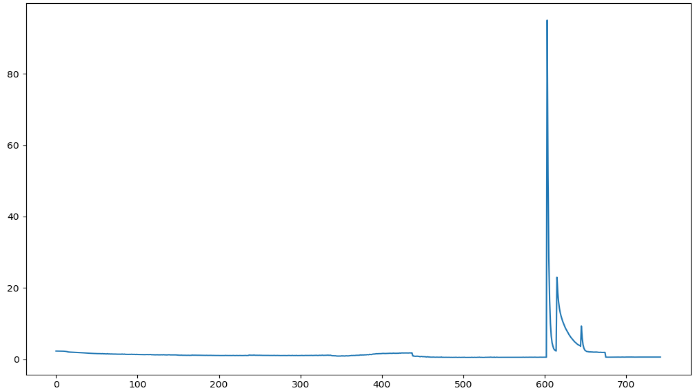
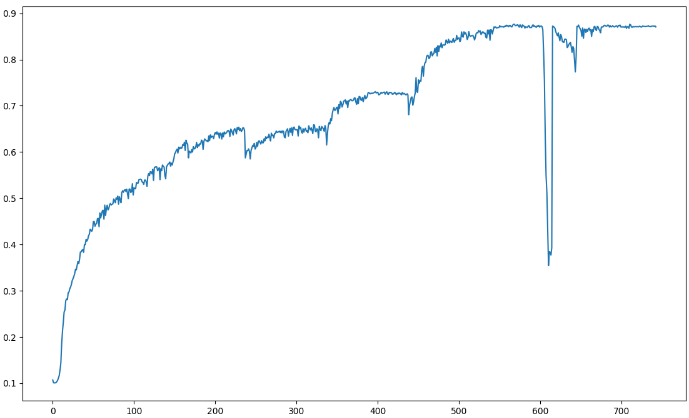


图6.3验证集acc图

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平台的上的评分达到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。

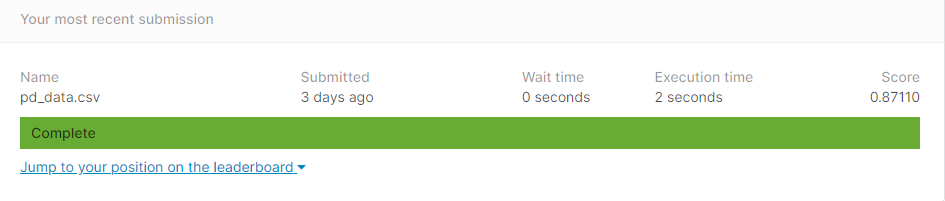


图6.5 kaggle得分

### 初步尝试

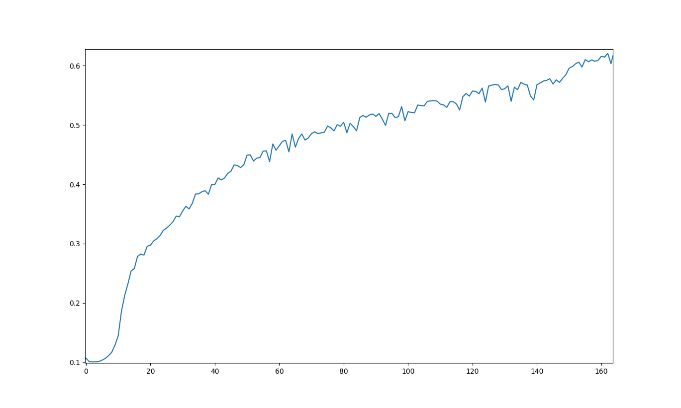
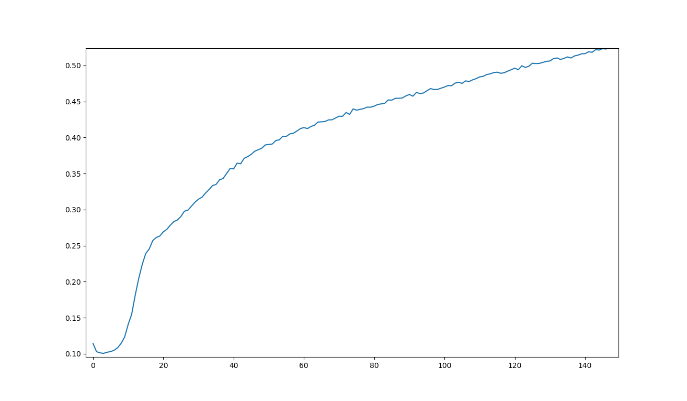


图6.6 初步训练集acc图

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了超过了训练集，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。

### 调整学习率

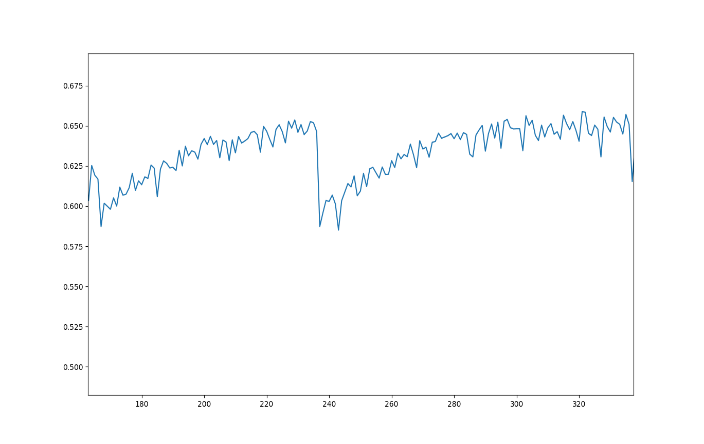
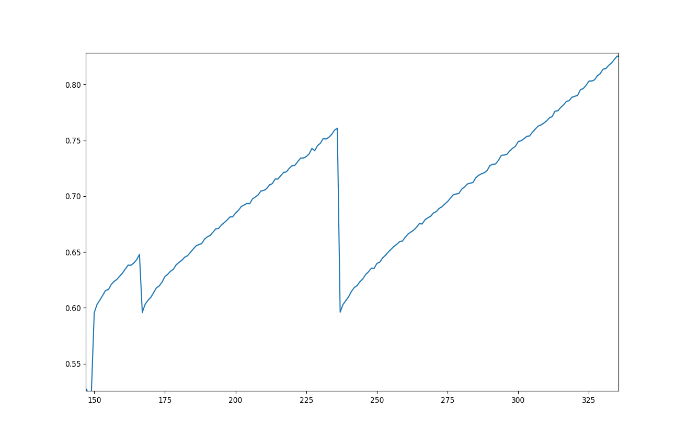


图6.8 调整学习率训练集acc图

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的整个学习的进度，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。

### 使用dropout

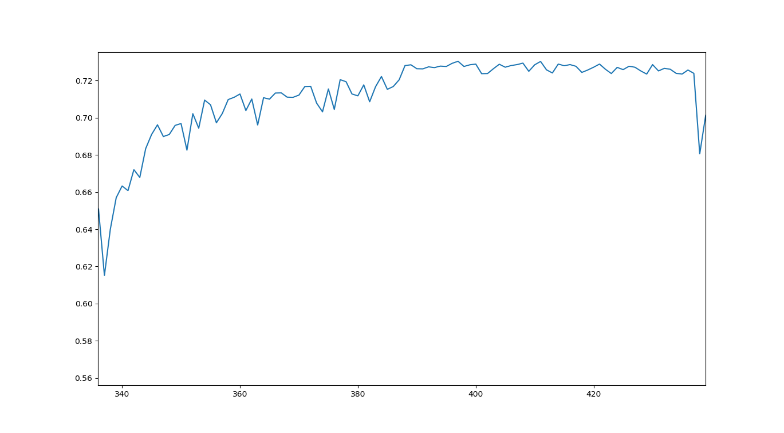
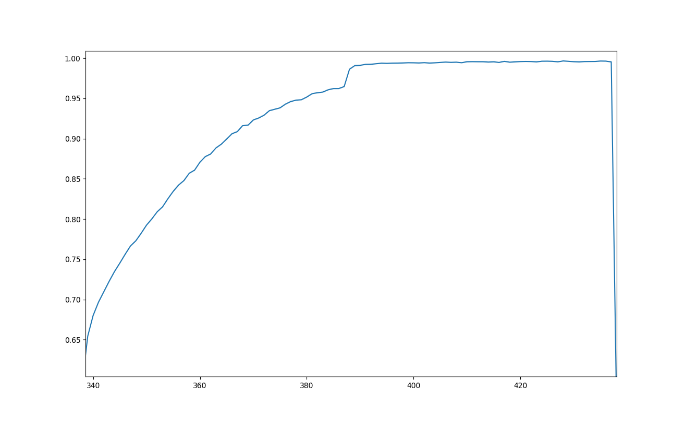


图6.10 dropout训练集acc图

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，提高网络的泛化能力。

### 使用图像增强

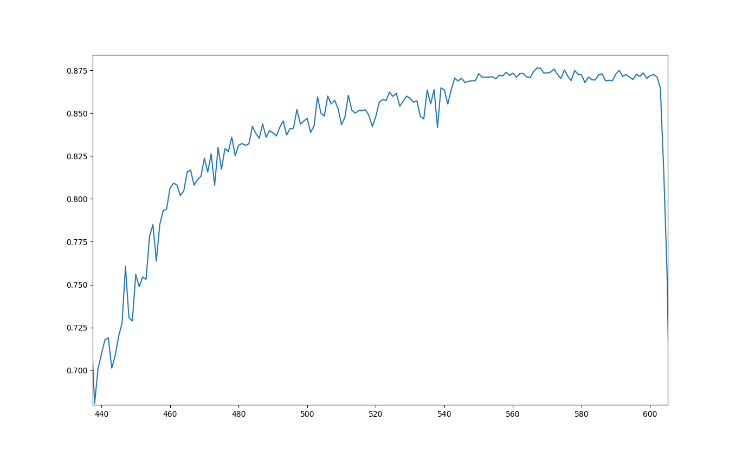
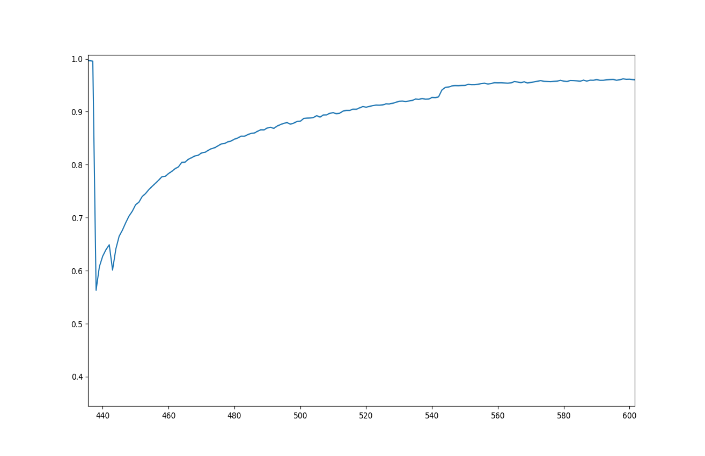


图6.12图像增强训练集acc图

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。

### L2正则化解决过拟合问题

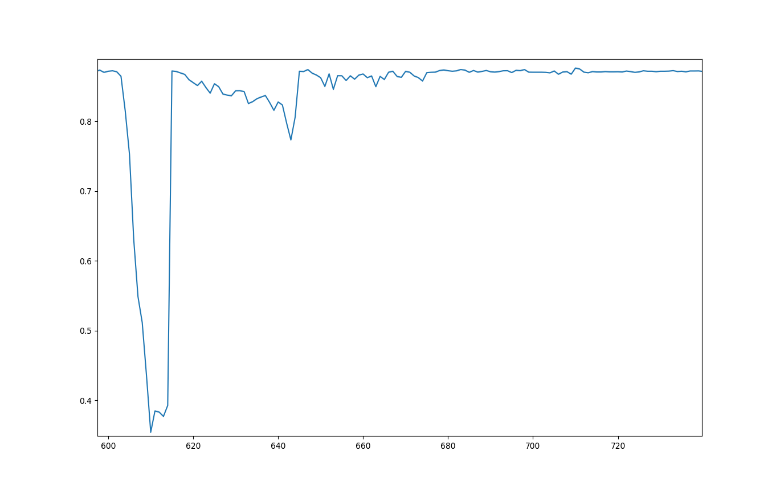
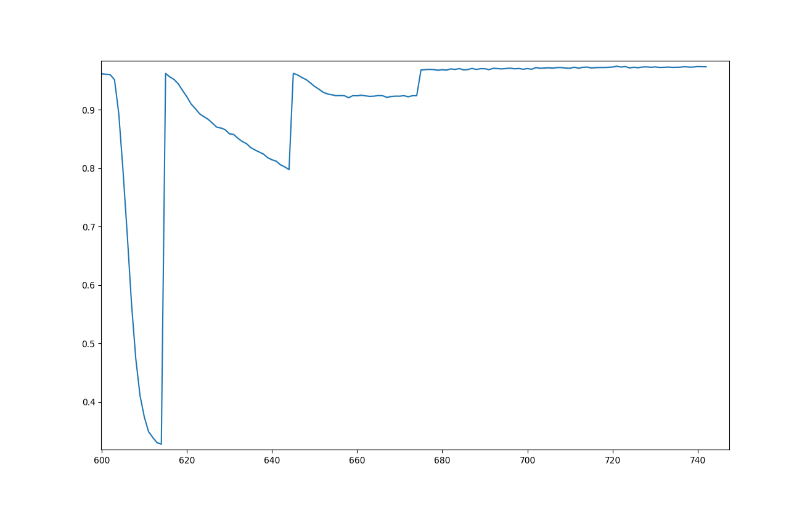


图6.14正则化训练集acc图

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λ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λ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，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λ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。

### 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：

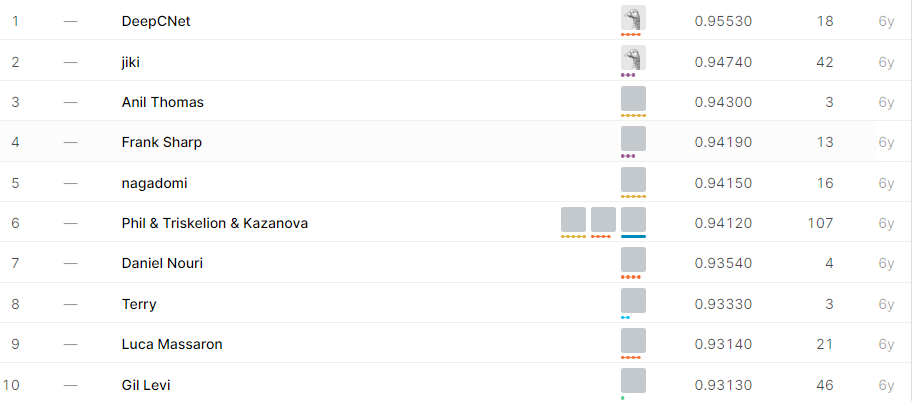


图6.16 kaggle排行榜

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。着重的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。

1. 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，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会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。

在最后非常衷心感激学校的培养，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 UniversityPress.

[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.

[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.