

**课 程 报 告**

**题目：基于卷积神经网络技术进**

**行 MNIST图像数据的分类**

**院（系）： 计算机科学与技术学院**

**课程名称： 人工智能导论（专选课）**

**学 号： U201814772**

**姓 名： 杨 开 源**

**任课老师： 魏巍 辜希武**

**报告日期： 2019 年 6 月 29 日**

**目录**

[1 绪论 3](#_Toc12820069)

[1.1课题背景与意义 3](#_Toc12820070)

[1.2国内外研究现状 4](#_Toc12820071)

[2 卷积神经网络技术的学习 6](#_Toc12820072)

[2.1从C语言到Python 6](#_Toc12820073)

[2.2关于MNIST数据集 6](#_Toc12820074)

[2.3关于TensorFlow框架系统 7](#_Toc12820075)

[2.3.1 TensorFlow简介 7](#_Toc12820076)

[2.3.2 TensorFlow核心——计算图 7](#_Toc12820077)

[2.3.4 张量（Tensor）与会话（Session） 8](#_Toc12820078)

[2.3.5 常量与变量 8](#_Toc12820079)

[2.4卷积神经网络的学习 8](#_Toc12820080)

[2.4.1 CNN算法与BP算法对比 8](#_Toc12820081)

[2.4.2 特征提取层（卷积层）—— C层（Convolution layer） 9](#_Toc12820082)

[2.4.3 特征映射层（下采样层）—— S层（Subsampling layer） 10](#_Toc12820083)

[2.4.4 全连接层（fully connected layer） 11](#_Toc12820084)

[2.4.5 卷积神经网络优缺点 11](#_Toc12820085)

[3 CNN技术的代码实现 12](#_Toc12820086)

[3.1 导入MNIST数据集 12](#_Toc12820087)

[3.2 构建图模型 12](#_Toc12820088)

[3.3 训练模型与优化器 14](#_Toc12820089)

[3.4 评估我们的模型 14](#_Toc12820090)

[3.4.1 流程分析 14](#_Toc12820091)

[3.4.2 实战代码 14](#_Toc12820092)

[3.4.3 正确率计算结果 15](#_Toc12820093)

[4 总结、展望与致谢 16](#_Toc12820094)

[4.1 个人总结 16](#_Toc12820095)

[4.2 未来展望 16](#_Toc12820096)

[4.2.1 神经网络发展的展望 16](#_Toc12820097)

[4.2.2 个人学习发展的展望 17](#_Toc12820098)

[4.3 致谢 17](#_Toc12820099)

[5 参考文献 18](#_Toc12820100)

# 1 绪论

## 1.1课题背景与意义

人工智能：用机器模拟人的意识和思维，自从图灵在《机器能思考吗》一文中提出后，就带有一丝神秘与诱人的气息。“图灵测试”也成为媒体争相报道的焦点。计算机很容易处理一些可以用形式化的规律来描述的问题，而人工智能面临的困难是那些很难用形式化符号描述的任务，如：语音识别、图像识别，这貌似对人类不算什么难题。

但是，现实中的许多事物极其复杂，人们很难深入看清本质，很难知道哪些特征是重要的，甚至不知何为真正的特征。研究生物神经网络带给我们的启示就是可以让机器自主发掘隐藏在知识中的规律，而不只是简单把知识灌输给计算机。人类从原始数据中提取高度抽象的特征是极其困难的，而计算机通过模拟人脑就可以用较简单的模型来表示复杂概念，解决了特征提取这一关键问题。机器学习就是利用这些已有数据，通过统计学的方法得出模型，再利用模型预测结果。深度学习是深层次神经网络，机器学习的一种实现方法。如今，深度学习逐渐发展成以人工神经网络为核心的算法体系。

人工神经网络算法体系中有两个很有名的算法：BP (Back Propagation) 神经网络和卷积神经网络，它们分别掀起了神经网络的第二、三次兴起（我们会在下一小节提到神经网络的发展历史）。本文主要是对卷积神经网络，即CNN（Convolutional Neural Networks） 算法进行学习讨论。也会涉及到BP算法和KNN（k-Nearest Neighbor) 算法的理论。

卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经网络。具有表征学习能力能力，能够按其阶层结构对输入信息进行平移不变分类，因此也被称为“平移不变人工神经网络。主要应用于计算机视觉，自然语言处理领域。卷积神经网络的出现是受到了生物处理过程的启发，因为神经元之间的连接模式类似于动物的视觉皮层组织。如图1-1所示，卷积神经网络架构与常规人工神经网络架构非常相似，特别是在网络的最后一层，即全连接。此外，还注意到卷积神经网络能够接受多个特征图作为输入，而不是向量。

卷积神经网络的创始人是着名的计算机科学家Yann Lecun，他是第一个通过卷积神经网络在MNIST数据集上解决手写数字问题的人。本文的内容为基于卷积神经网络技术进行 MNIST图像数据的分类

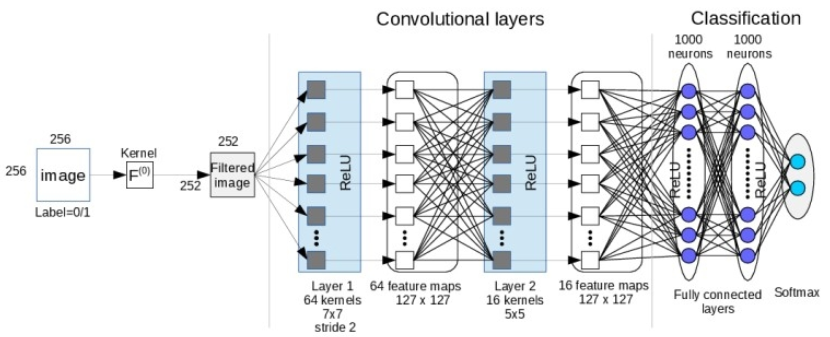


图1-1 卷积神经网络架构

## 1.2国内外研究现状

如今，人工智能语音助手已经走入寻常百姓家：微软的Cortana、苹果公司的Siri、小米科技的小爱同学，对人类的生活产生了翻天覆地的影响。一款几千元的移动设备，也能成为训练人工智能语音助手的工具。更多的人工智能算法已经运用到你我的生活细节之中：语音输入、电商与新闻的智能推荐……无人驾驶在技术层面逐步完善，特斯拉、奔驰、谷歌和百度等国内外大型公司都在探索这一领域。医疗教育养老领域的应用也在逐步扩展。人工智能的发展在经历又一个春天。

可是神经网络的发展并非一帆风顺，而是经历了“三起二落”的历史，更让我们对这些坚持研究神经网络的科学家们肃然起敬。1943年，心理学家McCulloch和数学家Pitts参考了生物神经元的结构，发表了抽象的神经元模型 MP。1958年，人们把两层神经元首尾相接，组成单层神经网络，称做感知机，其成了首个可以学习的人工神经网络。引发了神经网络研究的第一次兴起。11年后，权威学者Minsky用数学公式证明了只有单层神经网络的感知机无法对异或逻辑进行分类，要想解决异或可分问题，需要把单层神经网络扩展到两层或者以上。然而在那个年代计算机的运算能力，是无法支撑这种运算量的。第一次寒冬就此来领。时隔近二十年，神经网络的研究第二次兴起：反向传播方法的提出，有效解决了两层神经网络的算力问题。1995年，支持向量机诞生，一举击败神经网络，成为当时人工智能领域的主流算法，神经网络进入第二个冬季。2006年，深层次神经网络出现，2012年，卷积神经网络在图像识别领域中的惊人表现，又引发了神经网络研究的再一次兴起。

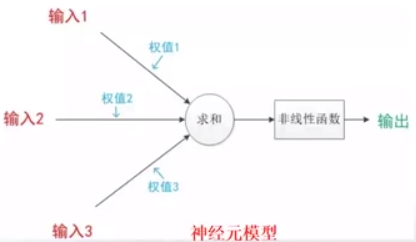
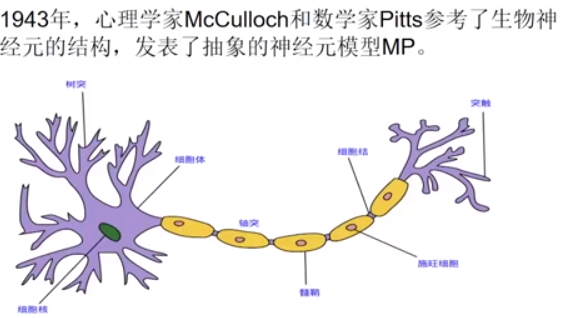


图1-2 神经元与神经元模型 MP

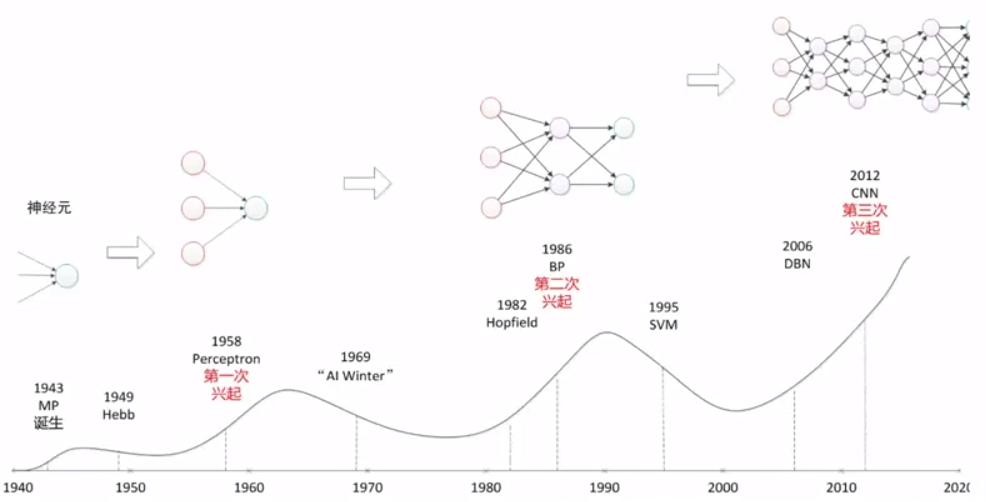


图1-3 神经网络发展的“三起二落”

如今，对卷积神经网络的研究日益完善，CNN在不同方面取得的进步：比如，层的设计，活跃函数、损失函数。除了从CNN的各个方面回顾其进展，我还介绍了CNN在计算机视觉任务上的应用，其中包括图像分类、物体检测、物体追踪、姿态估计、文本检测、视觉显著检测、动作识别和场景标签。但是，仍然还有很多工作值得进一步研究。首先，鉴于最近的CNN变得越来越深，它们也需要大规模的数据库和巨大的计算能力，来展开训练。人为搜集标签数据库要求大量的人力劳动。所以，研究人员都渴望能开发出无监督式的CNN学习方式。

# 2 卷积神经网络技术的学习

## 2.1从C语言到Python

Python是一种面向对象的动态类型语言，最初被设计用于编写自动化脚本(shell)，随着版本的不断更新和语言新功能的添加，越来越多被用于独立的、大型项目的开发。我们在上学期已经学习过面向过程编程的C语言，自己也在空闲时间学习过C++的语法和面向对象的基础。Python的学习较为快速，Python基础语法在列表上更加简便、切片操作十分有趣，字典的引入也提高了开发速度。Python语言在网络爬虫、数据分析领域也有很多的运用，在近年来热度不断提升，需要进一步的学习。在人工智能领域，Python“胶水语言”的作用和开源的本质，让其成为人工智能领域的活跃语言。

## 2.2关于MNIST数据集

MNIST数据集：MNIST数据集是NIST提供的更大集合的子集。数字已经过尺寸标准化，并以固定尺寸的图像为中心。MNIST是在机器学习领域中的一个经典问题。它包含7万张黑底白字手写数字图片，其中55000张为训练集，5000张为验证集，10000 张为测试集。每张图片大小为 28\*28 像素，图片中纯黑色像素值为 0，纯白色像素值为 1。数据集的标签是长度为 10的一维数组，数组中每个元素索引号表示对应数字出现的概率。

在将MNIST数据集作为输入喂入神经网络时，需先将数据集中每张图片变为长度784一维数组，将该数组作为神经网络输入特征喂入神经网络。例如： 一张数字手写体图片变成长度为 784的一维数组[0.0.0.0.0.231 0.235 0.459……0.219 0.0.0.0.]输入神经网络。该图片对应的标签为[0.0.0.0.0.0.1.0. 0.0]，标签中索引号为6的元素为 1，表示是数字6出现的概率为100%，则该图片对应的识别结果是6。

## 2.3关于TensorFlow框架系统

### 2.3.1 TensorFlow简介

TensorFlow是一个采用数据流图（data flow graphs），用于数值计算的开源软件库。节点（Nodes）在图中表示数学操作，图中的线（edges）则表示在节点间相互联系的多维数据数组，即张量（tensor）。它灵活的架构让你可以在多种平台上展开计算，例如台式计算机中的一个或多个CPU（或GPU），服务器，移动设备等等。TensorFlow 最初由Google大脑小组（隶属于Google机器智能研究机构）的研究员和工程师们开发出来，用于机器学习和深度神经网络方面的研究，但这个系统的通用性使其也可广泛用于其他计算领域。

### 2.3.2 TensorFlow核心——计算图

TensorFlow是一种计算图模型，即用图的形式来表示运算过程的一种模型。TensorFlow程序一般分为图的构建和图的执行两个阶段。图的构建阶段也称为图的定义阶段，该过程会在图模型中定义所需的运算，每次运算的的结果以及原始的输入数据都可称为一个节点。图模型的构建阶段：只在图中定义所需要的运算，而没有去执行运算。我们可以用图2-1来表示。

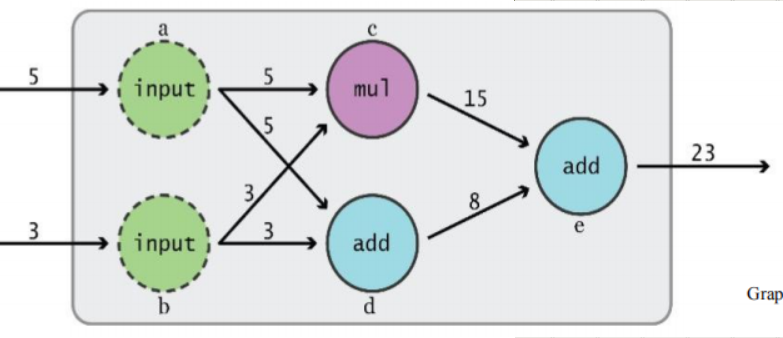


图2-1 图的构建阶段

第二个阶段为图的执行阶段，也就是在会话中执行图模型中定义好的运算。此外，我们还可以利用CPU或GPU等计算资源分布式执行图的运算过程。一般我们无需显示的指定计算资源，TensorFlow可以自动地进行识别，如果检测到我们的GPU环境，会优先的利用GPU环境执行我们的程序。

图模型的优点：

1. 节约系统开销，提高资源的利用率，可以更加高效的进行运算。
2. 这种结构有利于我们提取中间某些节点的结果，方便以后利用中间的节点去进行其它运算。
3. 这种结构对分布式运算更加友好，运算的过程可以分配给多个CPU或是GPU同时进行，提高运算效率。
4. 图模型把运算分解成了很多个子环节，所以这种结构也让我们的求导变得更加方便。

### 2.3.4 张量（Tensor）与会话（Session）

Tensor（张量）是TensorFlow中最重要的数据结构，用来表示TensorFlow程序中的所有数据。我们可以把Tensor理解成N维矩阵（N维数组）。其中零维张量表示的是一个标量，也就是一个数；一维张量表示的是一个向量，也可以看作是一个一维数组；二维张量表示的是一个矩阵；同理，N维张量也就是N维矩阵。

### 2.3.5 常量与变量

TensorFlow中对常量的初始化，不管是对数值、向量还是对矩阵的初始化，都是通过调用constant（）函数实现的。因为constant（）函数在TensorFlow中的使用非常频繁，经常被用于构建图模型中常量的定义。变量variable（）也是在TensorFlow中经常会被用到的函数。变量的作用是保存和更新参数。执行图模型时，一定要对变量进行初始化，经过初始化后的变量才能拿来使用。变量的使用包括创建、初始化、保存、加载等操作。

## 2.4卷积神经网络的学习

### 2.4.1 CNN算法与BP算法对比

传统神经网络中，采用的是BP的方式进行，即采用迭代的算法来训练整个网络，随机设定初值，计算当前网络的输出，然后根据当前输出和期望输出 之间的差去改变前面各层的参数，直到收敛。对于一个Deep Network（7层以上），残差传播到最前面的层已经变得太小，出现所谓的梯度扩散。加之支持向量机的诞生，神经网络的第二次寒冬来临。

CNN(Convolutional Neural Networks)是一个多层的神经网络，每层由多个二维平面组成，而每个平面由多个独立神经元组成。 C层为特征提取层（卷积层） S层是特征映射层（下采样层）CNN中的每一个C层都紧跟着一个S层最后是全连接层。

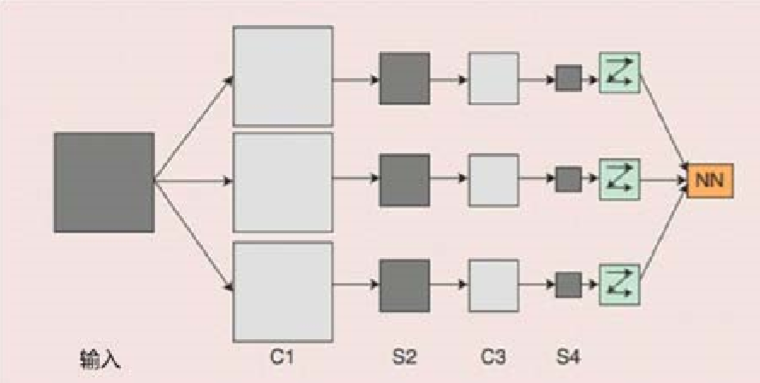


图2-2 CNN基本框架流程

### 2.4.2 特征提取层（卷积层）—— C层（Convolution layer）

卷积核（滤波器）是可以用来提取特征的。图像和卷积核卷积，就可以得到特征值。卷积核放在神经网络里，就代表对应的权重（weight)卷积核和图像进行点乘（dot product), 就代表卷积核里的权重单独对相应位置的Pixel进行作用

卷积神经网络中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法最佳化得到的。卷积运算的目的是提取输入的不同特征，第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网路能从低级特征中迭代提取更复杂的特征。

对图像用一个卷积核进行卷积运算，实际上是一个滤波的过程。每个卷积核都是一种特征提取方式，就像是一个筛子，将图像中符合条件的部分筛选出来。

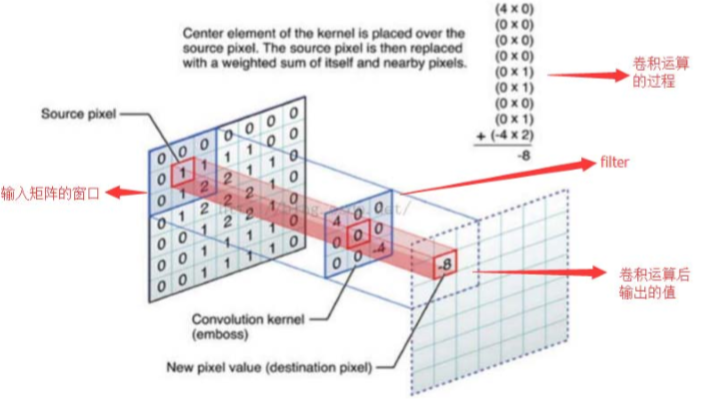
****

图2-3 卷积层的提取过程

### 2.4.3 特征映射层（下采样层）—— S层（Subsampling layer）

池化（Pooling）称为欠采样或下采样。主要用于特征降维，压缩数据和参数的数量，减小过拟合，同时提高模型的容错性。主要有：最大池化和平均池化。

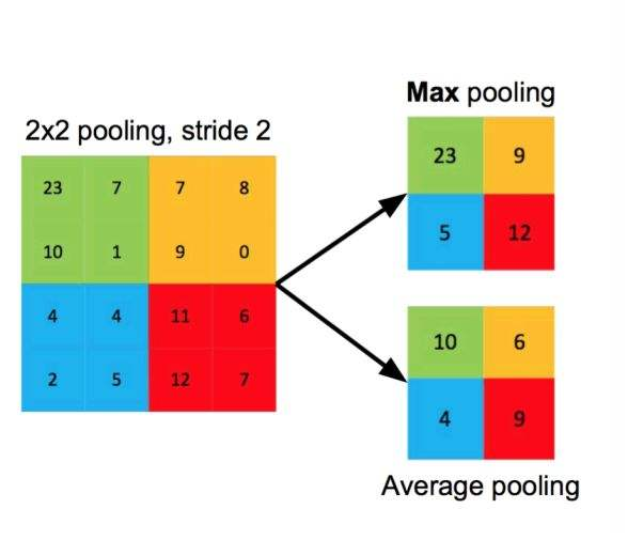


图2-4 最大池化与平均池化

池化后特征不变，图像压缩时去掉的信息只是一些无关紧要的信息，而留下的信息则是具有尺度不变性的特征，是最能表达图像的特征。

### 2.4.4 全连接层（fully connected layer）

连接所有的特征，将输出值送给分类器（如 softmax分类器）。接所有的特征，将输出值送给分类器（如 softmax分类器）如果说卷积层、池化层和激活函数层等操作是将原始数据映射到隐层特征空间的话，全连接层则起到将学到的“分布式特征表示”映射到样本标记空间的作用。

### 2.4.5 卷积神经网络优缺点

CNN的生物学基础支持不足，没有记忆功能。神经网络并不是每个神经元关注的是一个特征，而一组神经元关注一个特征。实现比较复杂，训练所需时间比较久。不是单一算法,不同的任务需要单独训练。

CNN算法的优点在于解决传统算法中权值太多的问题。卷积神经网络关键技术是多卷积核与权值共享技术。前者在每个卷积核都会将图像生成为另一幅特征映射图。为了使特征提取更充分，我们可以添加多个卷积核（滤波器）以提 取不同的特征。对于后者，例如局部连接中，每个神经元都对应25个参数，一共10000 个神经元，如果这10000个神经元的25个参数都是相等的，那么参数数目就变为25了。把这25个参数对应的卷积操作，看成是特征提取的方式，与图像区域的位置无关。卷积神经网络中相同的卷积核的权值和偏置值是一样的。同一种卷积核按照某种顺序对图像进行卷积操作，卷积后得到的所有神经元都是使用同一个卷积核区卷积图像的，都是共享连接参数。因此，权值共享减少了卷积神经网络的参数数量。

# 3 CNN技术的代码实现

注：由于有时代码文的文字排版不宜调整，且层次不易分清，故部分采用图片。

## 3.1 导入MNIST数据集

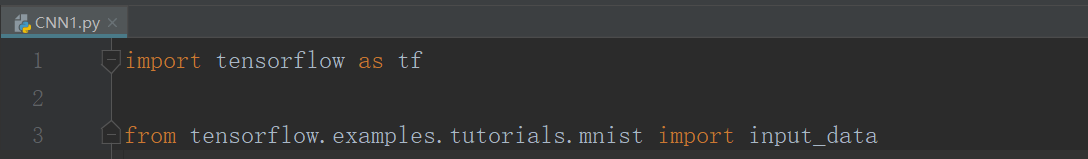


图3-1 下载导入数据集

## 3.2 构建图模型

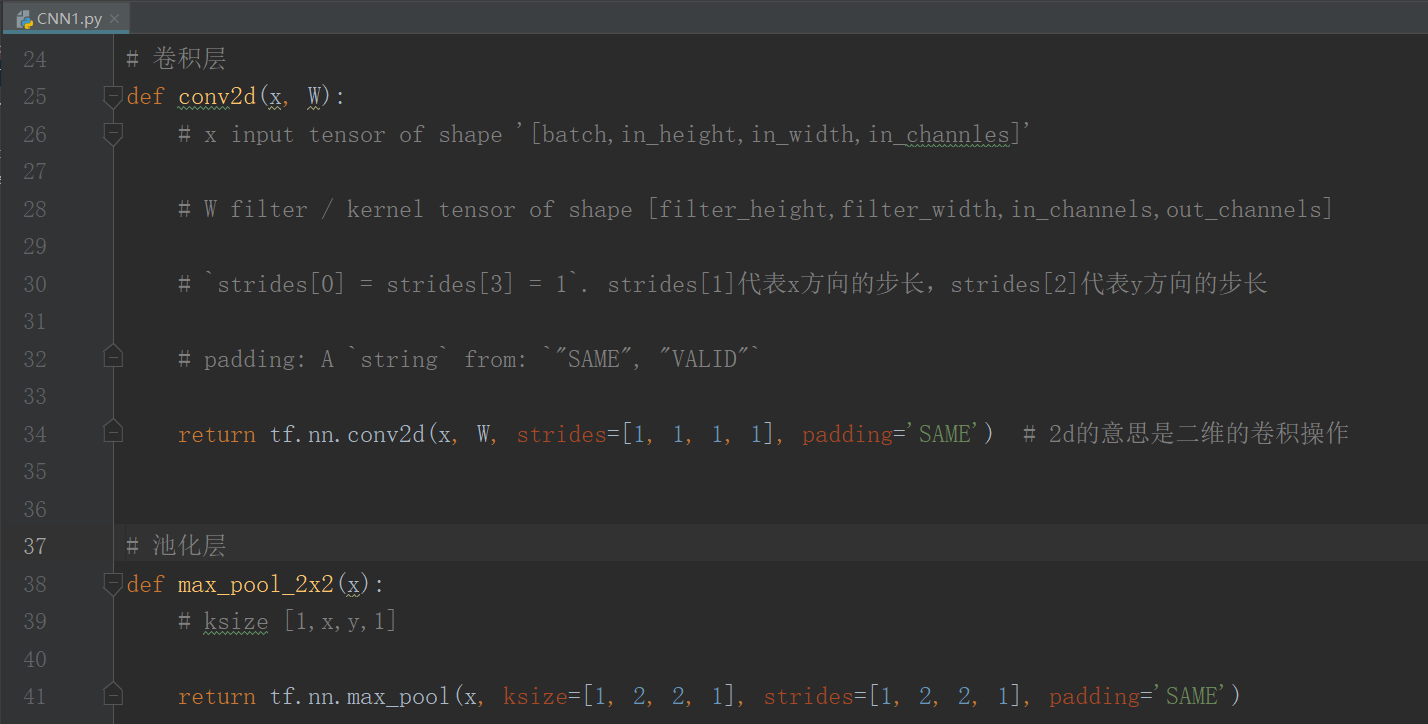


图3-2 卷积层与池化层

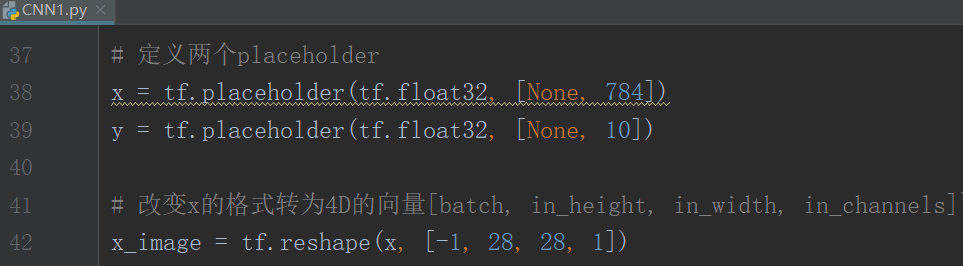


图3-3 创建可交互的操作单元

两次卷积层操作和两次全连接层输出

28\*28的图片第一次卷积后还是28\*28（数组变小了，但是图像大小不变）第一次池化后变为14\*14。第二次卷积后为14\*14（卷积不会改变平面的大小），第二次池化后变为了7\*7进过上面操作后得到64张7\*7的平面。



图3-4 两次卷积操作

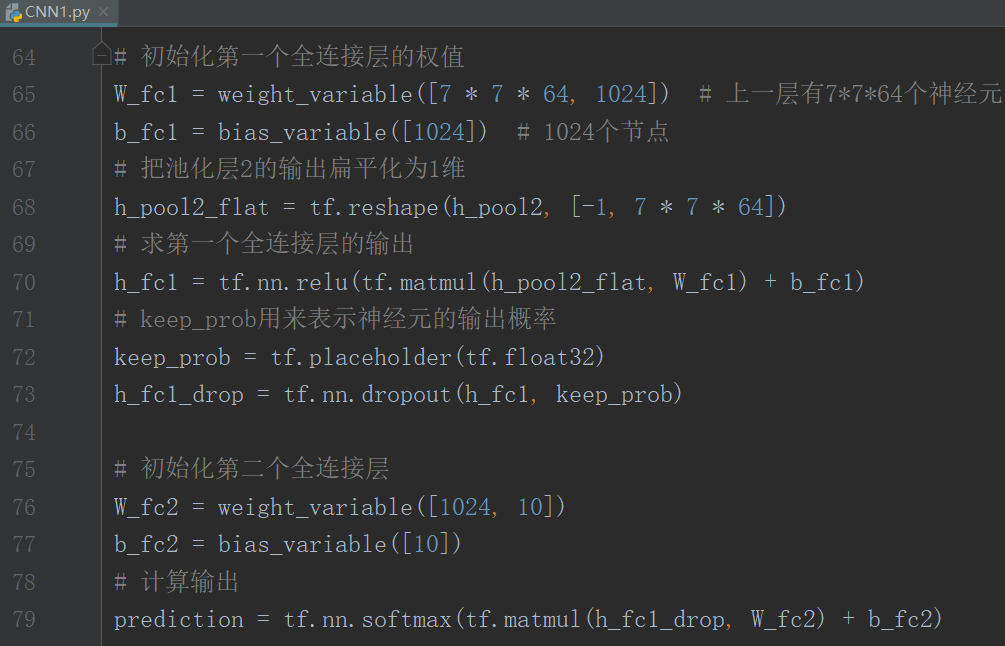


图3-5 两次全连接层的输出

## 3.3 训练模型与优化器

为了训练我们的模型，我们首先需要定义一个指标来评估这个模型是好的。其实，在机器学习，我们通常定义指标来表示一个模型是坏的，这个指标称为成本或损失。一个非常常见的，非常漂亮的成本函数是“交叉熵”。交叉熵产生于信息论里面的信息压缩编码技术，但是它后来演变成为从博弈论到机器学习等其他领域里的重要技术手段。

Adam(adaptive moment estimation)是利用梯度的一阶矩估计和二阶矩估计动态调整每个参数的学习率。优点是每一次迭代学习率都有一个明确的范围,使得参数变化很平稳。同时它也是实际学习中最常用的算法，基本可以将正确率提高到98%以上。

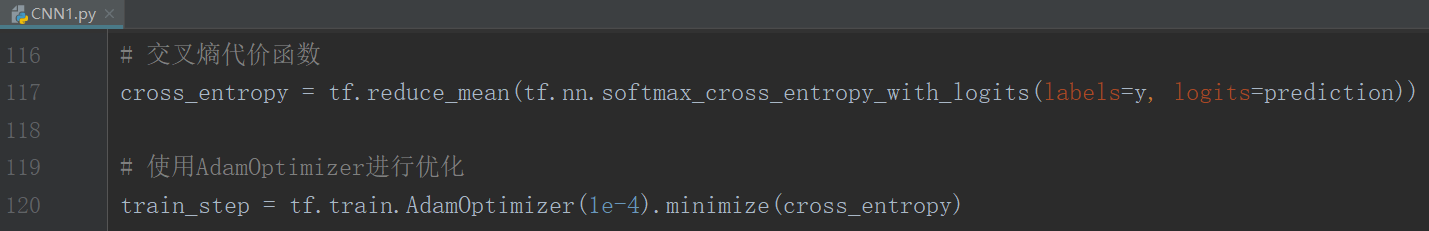


图3-6 交叉熵代价函数与Adam优化

## 3.4 评估我们的模型

### 3.4.1 流程分析

（1）找出那些预测正确的标签。tf.argmax 是一个非常有用的函数，它能给出某个tensor对象在某一维上的其数据最大值所在的索引值。由于标签向量是由0,1组成，因此最大值1所在的索引位置就是类别标签。

（2）得到一组布尔值后，为了确定正确预测项的比例，我们可以把布尔值转换成浮点数，然后取平均值。

（3）算所学习到的模型在测试数据集上面的正确率。

### 3.4.2 实战代码

# 结果存放在一个布尔列表中  
correct\_prediction = tf.equal(tf.argmax(prediction, 1), tf.argmax(y, 1))

# argmax返回一维张量中最大的值所在的位置

# 求准确率  
accuracy = tf.reduce\_mean(tf.cast(correct\_prediction, tf.float32))  
  
with tf.Session() as sess:  
 sess.run(tf.global\_variables\_initializer())  
 for epoch in range(21):  
 for batch in range(n\_batch):  
 batch\_xs, batch\_ys = mnist.train.next\_batch(batch\_size)  
 sess.run(train\_step, feed\_dict={x: batch\_xs, y: batch\_ys, keep\_prob: 0.7})  
 acc = sess.run(accuracy, feed\_dict={x: mnist.test.images, y: mnist.test.labels, keep\_prob: 1.0})  
 print("Iter " + str(epoch) + ", Testing Accuracy= " + str(acc))

### 3.4.3 正确率计算结果

计算的正确度保持在95%以上，最终稳定在99%出头的位置上。

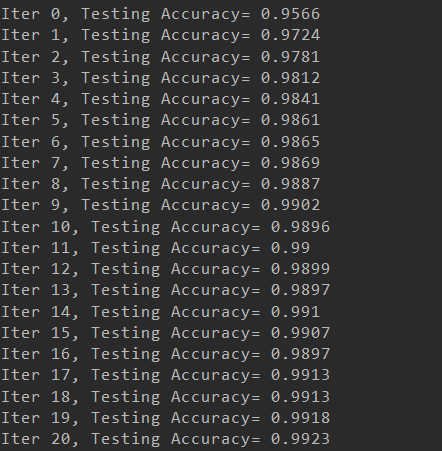


图3-7 正确率结果

# 4 总结、展望与致谢

## 4.1 个人总结

这次学习工作的成果大致有一下几点：  
（1）学习运用了python语言和数据分析与展示的库。利用C语言的基础类比学 习快速上手。发觉了python语言的特点与乐趣，特别是网络爬虫与机器学习方向的应用，打开了学习的又一扇大门。

（2）对神经网络的历史和两大算法（BP算法与CNN算法）有了了解和认识，认识到了人工智能学科发展的客观规律。科研的道路不是一帆风顺，人类总在从“正确”走向“错误”的路途中，逐渐靠近真理。

（3）课堂上辜老师在机器学习、深度学习上通俗易懂的讲解让我对原本高深的知识充满了兴趣，也看到了自己的不足：数学要是主动多学一点，英语能力再强一点，就能多接触好多更优秀更新的资源。能在大一结束的时候更好地思考自己的优缺点和今后的路，可能是影响最大的收获了。

（4）TensorFlow的学习使用。安装学习使用都来源于中国大学MOOC和TensorFlow中文社区的学习指导。开源框架的使用让我看到了程序领域开源的重要性。本来准备类似所学的慕课中在Linux环境下进行，但是个人水平不够。暑假也需要加强这一著名开源操作系统的使用。

## 4.2 未来展望

### 4.2.1 神经网络发展的展望

神经网络的发展思路让我们获益匪浅——生命科学是21世纪研究的重点学科之一，这也会对人工智能学科的发展带来巨大突破。而计算机科学和人工智能学科的发展也同样会有助于生命科学的研究。各学科的发展越来越需要借鉴对方的思维和研究方式。正如华科校友北大朱怀球教授在我的高中做讲座时所说：“薛定谔的那本《生命是什么》启发了我从物理学转向生物的研究，并且借助物理学研究的思维。”今后的人工智能学科的突破，可能也来自于非信息专业的启发。这也要求我们计算机专业的学生保持敏感度，重视其他自然学科的发展与成果，而不是身边很多同学认为的 “计算机是热门行业，其他专业都不值得学习”。这不是严谨的治学态度。

“三起两落”的历史也告诉我们，一种流行算法在不断的优化后，必然会遇到瓶颈。而打败他的是更为先进的算法实现。如今卷积神经网络流行了七八年的时间，参考历史，下一个寒冬、下一个新的神经网络算法还会远吗？

### 4.2.2 个人学习发展的展望

这次学习经历首先让我明白了自己数学英语基础的贫瘠，大一一年较为简单的考试不能很好地反应今后的要求。大二一年的重点还是要放在这两门科目上。第二是真正的进入计算机学科的学习状态。这次专业选修课我接触到python，了解到这一门语言还有很多很多好玩的用法，可以暑假一点点接触。我慢慢意识到，我们专业的自学过程常常是在学习一个东西的过程中发现了一个又一个新的有趣的“技能点”，然后开枝散叶，真正绘制一幅自己的“技能树”。人工智能本身的学习也是精彩的，我有机会和在北大信科的同学联系，发现他们也开了这样一门《人工智能引论课程》，但他们两三个人一个小组，实现了很多我们上课提到过的实例，如“电影推荐”、“对联自动生成”、“藏头诗”等等这样对比发现导论课，依然可以沉下心进一步学习很多东西。大二一年我希望自己学习的重点在算法和人工智能方向，从这一门课开始，一步一步走下去。

## 4.3 致谢

这次课程的学习十分顺利。首先感谢两位老师。魏老师的开头介绍让我对人工智能学科有了一个较为全面正确的认识，结束时的自然语言处理与语音处理更是令我印象深刻。辜老师机器学习、人工神经网络与深度学习的讲解十分易于理解，让我在数学储备不足的情况下也听懂了。其次感谢我的同学魏强，在我的anaconda安装tensorflow后仍显示No module named ‘tensorflow.examples’后积极帮我寻找原因，不过最后我直接选择了重新开一个tensorflow环境用Pycharm进行。再次感谢我的老师和同学，你们的教学、帮助我感激在心。希望今后自己技术变得强一点，也能去帮助别人共同进步。

# 5 参考文献

1. 中国大学MOOC《人工智能实践：TensorFlow笔记》北京大学 曹健
2. 中国大学MOOC 《Python语言程序设计》《Python数据分析与展示》《Python机器学习应用》北京理工大学 嵩天
3. 《机器学习》南京大学 周志华
4. 《TensorFlow实战》 黄文坚
5. TensorFlow中文社区 <http://www.tensorfly.cn/>