**数图期末报告**

**唐文萱**

**16342159**

**目录**

[**一、问题背景** 2](#_Toc46142671)

[**二、算法介绍** 2](#_Toc46142672)

[**2.1传统分类方法介绍** 3](#_Toc46142673)

[**2.2 随机森林** 4](#_Toc46142674)

[**2.3卷积神经网络** 5](#_Toc46142675)

[**2.3.1算法背景** 5](#_Toc46142676)

[**2.3.2 CNN结构介绍** 6](#_Toc46142677)

[**三、模型构建** 9](#_Toc46142678)

[**3.1数据处理** 9](#_Toc46142679)

[**3.2传统分类算法训练** 10](#_Toc46142680)

[**3.2.1 KNN分类器** 11](#_Toc46142681)

[**3.2.2 Logistic Regression分类器** 11](#_Toc46142682)

[**3.2.3 高斯核SVM分类器** 12](#_Toc46142683)

[**3.2.4分类器效果比较** 13](#_Toc46142684)

[**3.3随机森林训练** 14](#_Toc46142685)

[**3.4卷积神经网络** 15](#_Toc46142686)

[**四、结论** 16](#_Toc46142687)

[**4.1对于传统的分类器** 16](#_Toc46142688)

[**4.2对于随机森林和卷积神经网络** 17](#_Toc46142689)

[**五、附录** 17](#_Toc46142690)

[**5.1数据来源** 17](#_Toc46142691)

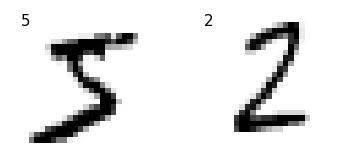
[**5.2代码** 17](#_Toc46142692)

**一、问题背景**

近年来，手写数字识别是计算机视觉和模式识别中一个广受关注的问题，引起了越来越多学者的兴趣。本文将对这个问题做出一些简单的研究。

在本报告中，我们将使用MNIST数据集，它是一个用于测试图像分类算法的经典数据集，包括了70000张规格较小的手写数字图片，由美国高中生和美国人口调查局的职员手写而成。MNIST中的每张图片都是 28×28 像素（即共有784个特征），并且每个像素的值介于 0~255之间。

我们先看看数据集中的一些图片。为了读取一张图片，需要取出MINST中的某一行数据（784维），然后将其变换为28×28的矩阵，最后使用Matplotlib库的imshow函数展示出来。



以上是我们选取的两张手写数字图片，左上角是它们的真实标签。接下来我们将使用一些算法来训练计算机，以尽可能地做到准确识别这些数字。为了研究方便，我们将原来的多分类问题简化为二分类问题，即只判断手写数字是否为5。选择5的原因是5与其它某些数字在笔画上接近，这将加大算法正确分类的难度。另外，我们采用欠抽样的方式，即只随机抽取6000个标签非5的样本。这样做是为了防止数据的不平衡，并且减少训练负担。

**二、算法介绍**

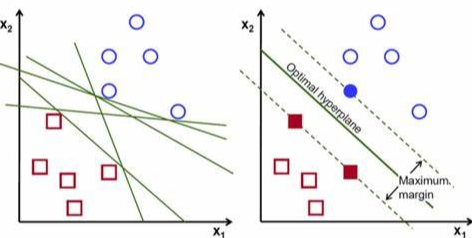
这一部分我们对后文将要使用的分类模型进行介绍。

**2.1传统分类方法介绍**

有一些传统的机器学习分类方法可以用来识别这些数字，在本文中将先用它们来进行分类工作。

KNN，又称K-近邻分类法，是一个非参数方法，也是最简单的分类方法之一。它计算距离每个点最近的K个近邻点，利用这些近邻点的类别来进行投票，以此决定原点的类别。过程中可以使用距离来生成近邻点的权重系数，这样可以让模型更稳健。

SVM，又称支持向量机，是一种广泛应用的分类算法，其原理是试图生成一个尽量宽的间隔带，以此来分割不同类别。核SVM是其中的一种特殊类型，它使用核（Kernel）技术对变量进行处理。核技术的大致思想是用每个样本与特定地标的相似度来替代原特征，例如高斯核即是使用高斯径向基函数计算相似度来变换原特征。SVM算法的优点是只收到一部分样本（准确来讲是只受支持向量）的影响，因此对比线性分类器对于极端数据的抗性更强。

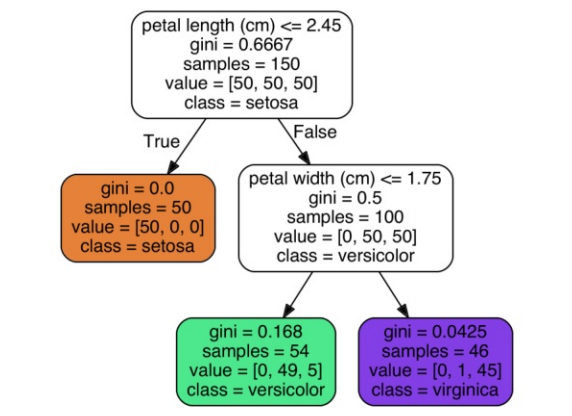


SVM（右）和普通线性分类器

Logistic Regression，又称逻辑回归，是一种广义的线性回归分析。其利用极大似然方法求参，以此来适用标签为哑变量的回归问题。逻辑回归由于参数可解释性强而受到广泛的欢迎，是最古老的几种分类学习方法之一。

**2.2 随机森林**

随机森林是最为经典的集成学习方法，其原理利用了一种简单的非参数分类器——决策树。树模型的流程是：每步均遍历所有变量的所有分割阈值，找出一个使分割后不纯度下降最大的分割方法，一直迭代至每个节点纯度为100%。简单地说，决策树是一种每次均寻求局部最优的非参数分类方法。



决策树示意图

相对地，随机森林通过随机抽取数据和特征，生成大量决策树，最后所有树进行投票来得到最终的类别。随机森林既具有决策树对数据分布没有要求和拟合能力强的优点，又克服了树模型较容易过拟合的缺点，是一种兼顾拟合和泛化能力的模型。

随机森林模型存在着许多超参数，下面我们对其做一些简单的介绍：最大深度，可以理解为叶子节点和根节点的最大容许距离；继续分割最小样本数，只有当节点的样本数大于这个数目，分割才将继续；单个节点最小样本数，每次分割得到的子节点样本数都必须大于这个数目；每次随机抽取样本数和每次随机抽取变量数，决定了每次抽取样本和变量的数目。以上超参数大部分都是正则项参数，即限制了模型的拟合以防止过拟合。

**2.3卷积神经网络**

**2.3.1算法背景**

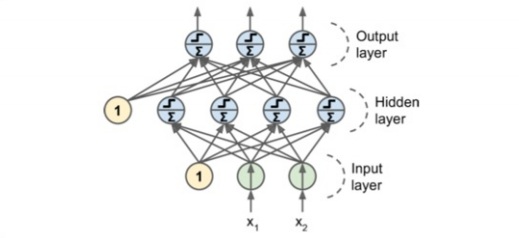
卷积神经网络（CNN）起源于两位医学研究者在1958年及之后数年进行的一系列实验。两位学者在实验中对视觉皮层的结构提供了重要的见解。具体来说，他们发现视觉皮层中的许多神经元有一个小小的局部感受野。这意味着这些神经元只对位于视野中的部分区域的视觉刺激有反应。不同神经元的感受野可能重叠，并且将一起铺满整个视野。此外，他们还发现一些神经元只对水平方向的图像做出反应，而另一些神经元则只对垂直方向的图像做出反应（也就是说两个神经元可以具有相同的感受野，但却分别针对不同方向的图像）。学者们还注意到一些神经元具有较大的感受野，并且它们能对较复杂的模式——例如较低层神经元的输出组合做出反应。

这些研究结果均指向一个自然的结论：高级别的神经元的输入是基于相邻低级神经元的输出的。这种强大的结构已经被证明了能够在一个极大的视野内检测任何区域的各种复杂图案（某些动物强大的视觉能力也侧面证明了这一点）。 在这些对视觉皮层的研究的启发下，一种相关的网络结构于1980年出现，并逐渐演变为现在的卷积神经网络。CNN研究中的一个重要里程碑是四位科学家于1998年发表的一篇论文，该论文介绍了CNN的一个经典架构——LeNet-5（后来广泛用于识别手写支票号码）。

接下来将介绍卷积神经网络最为重要的几个概念——卷积层、卷积核和池化层。

**2.3.2 CNN结构介绍**

在探讨CNN之前，有必要先对基础的网络结构做一点了解。网络结构是多层结构，每层都有若干个神经元，每个神经元都有输入和输出。对每一个神经元来说，输入是这样得到的：低一层的每个神经元输出一个线性组合，然后将得到的值代入一个激活函数得到最终的输入。该神经元将再通过同样的方式影响下一层每一神经元的输入。

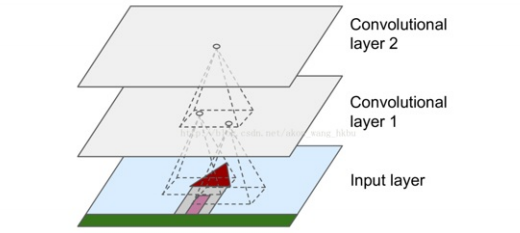


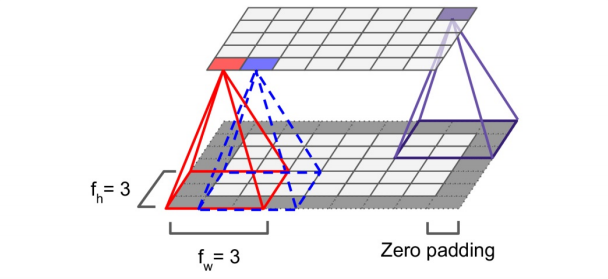
激活函数的选择事实上是一个重要的问题，但此处不再深入探讨，本报告将默认使用Relu（Relu(x)=max(0，x)）作为激活函数。此外，大多数深度网络的训练方法为反向传播，其原理是：对于每个样本，首先进行预测并计算误差（前向），然后反向遍历每个层来衡量每个连接的误差贡献，最后调整连接器权值（梯度下降步长）以减少误差。

现在我们可以来探讨CNN的具体结构了。

**2.3.2.1卷积层**

卷积层是CNN最重要的组成部分。在卷积层中，不同于普通的网络结构，神经元不是连接到底层中的每一个像素（神经元），而是仅仅连接到一个小矩形（类似生物视觉中的感受野）内的像素，这些像素的线性组合（通过激活函数）决定了高层的神经元输入。这种架构让网络可以将当前的低级特征组装成下一隐藏层中的高级特征。

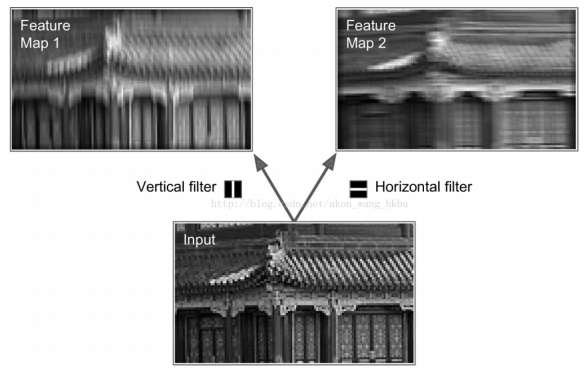




上图中的fh和fw分别代表局部感受野的高度和宽度，相邻感受野之间的距离称为步幅（stride，图中为1）。另外，为了使同一层神经元的感受野保持一致，通常会在低层次的神经元外围添加一些恒定为0的像素点，称为零填充（Zero Padding），这将使得高层次的边缘神经元也能连接至一个完整的感受野（如图中红色神经元所示）。

**2.3.2.2卷积核**

对于一个卷积层，fh、fw或者步幅等都是事先确定的，在训练中我们主要的目标是寻求一个最佳的映射权重矩阵，也叫卷积核。下图展示了最常见的两种卷积核——水平方向卷积核和垂直方向卷积核：



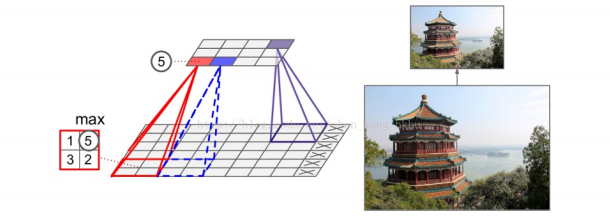
以水平卷积核为例，这是一个充满0的稀疏矩阵，除了中央水平线上的元素是1（当然，训练中它们将会被归一化）。

到目前为止，我们所展示的卷积网络针对的均为简单的薄二维层（例如本文中的数据显然是只有一层像素的黑白图像），显然可以轻易地将模型推广至多层图像结构上（如RGB彩色图像）。

**2.3.2.3池化层**

从上面的描述可以看到，卷积层的计算量是非常大的：以MINST中的数据为例，78×78的黑白图像通过5×5的卷积核映射到一个78×78的高层级中，总共需要78×78×(5×5+1)= 158184个权重参数，考虑到无论是输入或输出我们都只使用了一个薄二维层，这个数字已经十分惊人了。因此，我们有必要引入另外的结构来对图像数据进行压缩处理，这就是池化层。

池化层的操作过程较为简单，它的结构几乎与卷积层一致，唯一不同的是它没有核。换而言之，池化层并不将底层神经元做线性组合，而是直接取视野内的最大值（或者其它聚合值，如平均值）。



显然这是一个极具“破坏性”的操作，它将使得输入的信息几乎减少了75%，但模型的计算负担也将因此大大减轻。

由于卷积神经网络的具体架构（如卷积层和池化层的数量和相对位置）是一个较为艰深的问题，本报告后续的模型将从简单、计算量小的角度出发去建立模型。

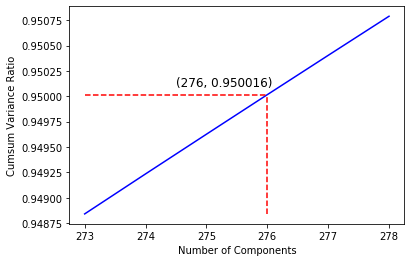
**三、模型构建**

**3.1数据处理**

在进行分类工作之前，有必要先对数据做出一些处理。首先，MINST数据集没有缺失值，所以无需进行缺失值的处理。然后，我们将数据集以8：2的比例分割成训练集和验证集，注意分割时要保持不同标签数据的比例大致不变。完成分割后，下述数据处理和模型训练将只在训练集上进行。

接下来我们需要对数据进行降维处理，原因如下：首先，每一个样本有784维，也就是有784个特征，如果不进行降维处理将使得大多数算法的训练速度过低；其次，不降维也会令算法不得不直接面对一个较为稀疏的数据矩阵，这极有可能降低模型的准确性。

我们选用PCA技术来进行降维操作，其原理是每次找出现有变量的一个线性组合（主成分），使得该组合方差最大。每个组合的方差与总方差的比例称方差贡献率，观察下图：



可以看到前276个主成分的累计方差贡献率恰好超过95%，因此我们决定只选取前276个主成分来构成训练集。这样，我们将数据从784维度降至276维度。然后我们利用已得到的结果对验证集做相同的变换。

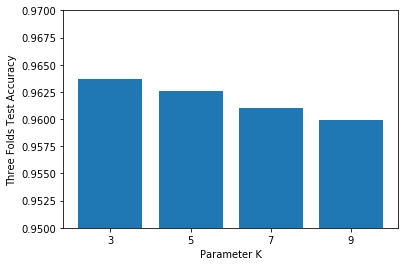
现在我们可以开始训练模型了。

**3.2传统分类算法训练**

我们将逐一使用2.1提到的算法，借助sklearn机器学习库来对数据进行训练。

**3.2.1 KNN分类器**

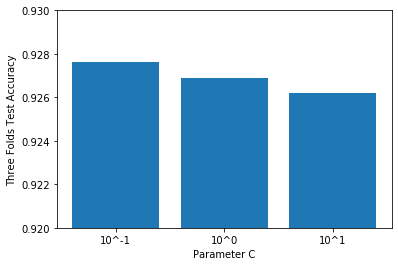
对于KNN分类器，最重要的步骤显然是决定参数近邻数目K的取值。一般来说K的取值在3～10之内是比较合理的。我们将使用网格搜索的技术，利用交叉验证所得到的准确率来决定K的取值。下图展示了使用不同K值得到的三折交叉验证准确率：



因此，我们最终得到的KNN模型的参数配置为：近邻参数K=3，权重系数为欧式距离的倒数。

**3.2.2 Logistic Regression分类器**

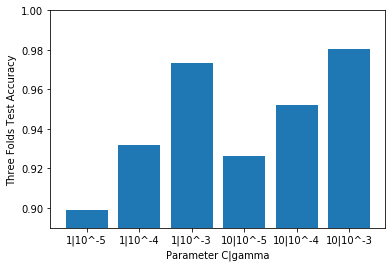
对于逻辑回归，我们同样需要关注超参数的设置，比如惩罚项形式（l1或l2）和惩罚项系数C。类似于KNN，我们使用交叉验证所得到的准确率来决定参数的取值。下图展示了使用不同的参数得到的三折交叉验证准确率：



经过权衡，我们最终得到的逻辑回归模型的参数配置为：惩罚项为l2，惩罚项系数为0.1。

**3.2.3 高斯核SVM分类器**

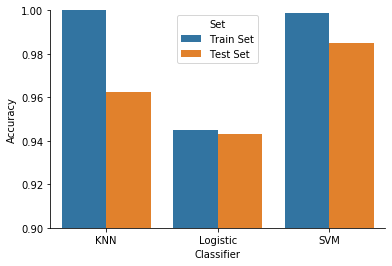
对于支持向量机分类器，我们需要关注的参数有用于计算相似度的高斯径向基函数的参数gamma，以及惩罚项系数C。同样地，我们使用交叉验证所得到的准确率来决定参数组合的取值。下图展示了使用不同的参数组合得到的三折交叉验证准确率：



最后我们决定使用gamma为10^-3，惩罚项系数为10的高斯核SVM分类器。

**3.2.4分类器效果比较**

下图分别展示了上述三种分类器在训练集、验证集上的分类效果：

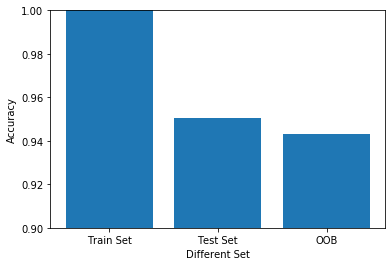


可以看到，三种分类器均有不错的效果。在训练集上，KNN和SVM的拟合程度接近100%，逻辑回归也有94%左右的表现；而在验证集上，SVM的准确率达到98%，展现了强大的泛化能力，KNN和逻辑回归则分别是96%和94%左右。

**3.3随机森林训练**

我们将继续借助sklearn机器学习库来建立随机森林模型对数据进行训练。

鉴于随机森林众多的超参数，对各种参数组合做详尽的网格搜索变得不现实（需要花费大量时间），因此我们将直接使用sklearn库的默认参数来进行模型的训练。事实上，由于随机森林自身具有的强大过拟合抗性，正则项参数的设置将不会对最终的分类效果产生较大影响。另外，我们将增加oob-score这项指标来衡量分类效果。oob-score的原理是：随机森林中的每棵决策树都使用了一个随机抽取的样本集，因此对于每个样本，我们可以选择那些未使用该样本训练的树组成森林来对其进行预测，以这种方式得到的预测结果的准确率就称为oob-score。以下为随机森林模型在训练集、验证集、oob上的准确率一览：

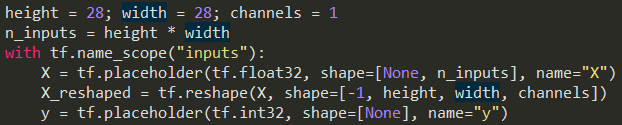


可以看到随机森林分类器在训练集的拟合程度也接近100%，同时验证集的预测准确率和oob-score分别为95%和94%左右。

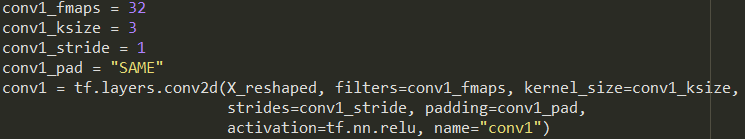
**3.4卷积神经网络**

为了实现卷积网络分类，首先我们需要明确整个网络的架构。我们用tensorflow深度学习库依次搭建以下层级：

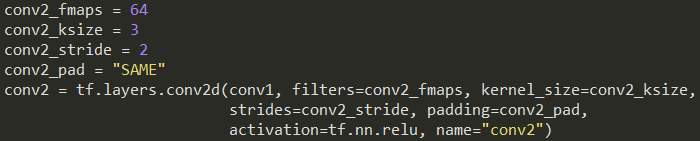
输入层：28×28的单层图像。



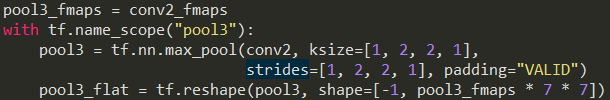
卷积层一：32个神经元，3×3的感受野，步幅为2，并使用零填充。



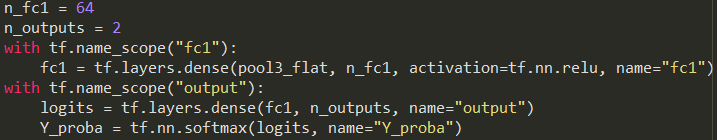
卷积层二：64个神经元，3×3的感受野，步幅为2，并使用零填充。



池化层：64个神经元，2×2的感受野，步幅为2，不使用零填充。

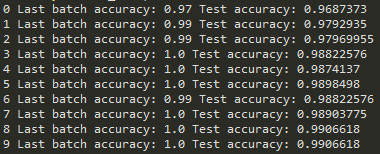


全连接层和输出层：先将池化层结果通过一个全连接层，最后在输出层输出一个二类预测。



事实上本报告中的模型还做了其它设定，比如2.3.2提到的激活函数默认选用Relu；在训练过程中的梯度下降法使用了小批量梯度下降和Adam优化技术；输出层使用了交叉熵（代码中的softmax函数部分）来将数值转化为类别概率。这些参数的设置是所有深度网络的共享参数，换言之它们对于理解卷积网络的架构显得不那么重要，我们将不对其做深入解释（事实上我也不是特别了解其中某些部分的原理）。此外需要注意的是，由于卷积模型本身包含了数据压缩的步骤，我们无需再事先对数据做降维处理，因此训练和验证用的数据需要重新加载至降维前的状态。

让我们查看模型训练的结果。下图展示的是梯度下降每一批次迭代时，模型在当前批次数据（可以理解为当下训练集）和验证集上的准确率。



可以看到最终的结果是：卷积网络分类器在验证集上的准确性达到了99%以上，这是一个非常不错的结果。

**四、结论**

**4.1对于传统的分类器**

从3.1的结果上来看，报告选取的三个传统分类器KNN、逻辑回归、高斯核SVM都取得了不错的结果，其中表现最佳的高斯核SVM在验证集上的分类准确率甚至接近99%。事实上，传统分类器在数据上的良好表现是我没有想到的。

另外，对比三个分类器的表现可以发现结果符合它们的优劣预期。逻辑回归长于解释性，然而相对SVM容易受到极端数据的干扰，因此在本场景下适用性稍显不足。此外KNN和核SVM的表现也说明了在本场景下非线性分类的表现要强于线性分类。

**4.2对于随机森林和卷积神经网络**

参照3.2的结果来看，随机森林展现出了它的几大优点：拟合能力强的同时保证了一定的泛化能力，并且在数据集较大的情况下展现出了不错的训练和验证性能（对比KNN和SVM的低速训练和预测）。对于非工业化的场景，随机森林高准确率和强解释性的特点让其脱颖而出。

参照3.3卷积神经网络的训练结果来看，卷积分类器是表现最好的一个分类器，在训练集和验证集上都有上佳表现，展现了CNN算法在图像方面的一定优势。但其缺点也十分明显：没有解释性、训练速度慢需要加速技术，重型模型不适用于非工业场景；由于对MINST分类问题的简化（简化为二分类），卷积网络相对于传统机器学习方法的准确性优势也不是很明显。此外，由于机器性能限制，网络模型中的众多超参数无法进行调优，这必然会导致模型结果无法达到最佳。

**五、附录**

**5.1数据来源**

本报告所使用的数据来源：Python-Sklearn库中自带的MINST数据集。

**5.2代码**

所有代码见文章附件。

py代码运行环境为Python3.7，sklearn版本为0.23.1，tensorflow版本为2.1.0。