论文技术报告

杨津

July 20, 2019

1 论文基本介绍

在阅读了高飞老师发布的论文后，首先谈谈自己对这篇论文的理解顺便进行内容梳理。本篇论文的观点是现在越来越多的人在进行模型复杂化以及深化的过程中忽略了特征提取的重要性，本次论文将利用对特征的优化提取以及选择，例如特征的大小，数量以及提取特征的步长等方面进行工作，最后把优化过的特征放入简单的模型当中进行实验。实验结果表明，优化过的特征在简单的模型也能体现不错的分类效果。

2 实现过程

对于本次论文复现的过程中，我主要从以下几个方面进行代码操作，其中分为数据的归一化，训练集随机取块，本次实验的切块大小为6*6，利用Kmeans算法对所有切块进行聚类，得到n个聚类中心，接着对训练样本进行标准化切块，用标准化的切块进行kmeans特征重组,接着池化减少维度，得到每个图像的优化特征表示。将每个图像重组后的特征向量进行训练，最后对测试样本进行相同的特征优化，再进行分类预测。

2.1 数据归一化

标准化是一个比较简单的部分。实际操作就是对每个特征x进行减去该特征的均值，除以方差即可。具体代码实现如下：

```
# 标准化处理
def norm(x):
    x = x.reshape(x.shape[0], -1)
    x -= np.mean(x, axis=0) # 减去均值
    x /= (np.std(x, axis=0, ddof=1) + 1e-7) # 除以方差
    x = x.reshape((x.shape[0],28,28))
    return x
```

2.2 随机切块

随机切块的目的是获得每张图片的更多重叠特征，在进行kmeans聚类后对于特征表示有更好的效果。具体代码如下：

```
# 对x进行切片，对每个x随机取n个6*6的小矩阵，重新组合成x(n*x.shape[0],6,6)大小的数组
```

```
def patch_random(x,n):
    x_random=np.zeros((n*x.shape[0],6,6))
    for i in range(x.shape[0]):
        for j in range(n):
            num=np.random.randint(0,23) # 随机在（0，23）取值当作切片的左上角
            x_random[n*i+j]=x[i,num:num+6,num:num+6]
    return x_random
```

2.3 标准化切块

标准化切块的目的是为了让数据有相同的特征表示，以便于训练和分类预测

```
# 对输入样本进行定步长，定规律，同数量的切片操作，为后面组成新特征做准备
def patch_norm(x,n):
    stride=4 # 步长取4
    x_norm=np.zeros((n*x.shape[0],6,6))
    for i in range(x.shape[0]):
        row=-3 # 从（1，1）开始取
        col=1
        for j in range(n):
            row+=stride
            if row>23:
                row=1
                col+=stride
            x_norm[n*i+j]=x[i,row:row+6,col:col+6]
    return x_norm
```

2.4 kmeans特征变换

我们先通过聚类得到n维的k中心，根据我们标准化后的图片切块对n个k进行欧式距离计算，在最近的i维上记1，其余n-1维都为0（one-hot编码），由此我们将本来每个切片6*6的特征转化为了1*n。具体代码如下：

```
# x为n*36大小的矩阵，其中n为图片数目*切块数目
def cluster(x,k):
    x=x.reshape(x.shape[0],-1)
    cluster=np.zeros((x.shape[0],k.shape[0]))
    for i in range(x.shape[0]):
        mindist=1111111
        index=0
        for j in range(k.shape[0]):
            dist=dists(x[i],k[j]) # 计算欧式距离
            if dist<mindist: # 找到最近的k
                mindist=dist
                index=j
        cluster[i][index]=1 # 对每个切片在k.shape[0]维中距离最近的k维上标记为1，
```

其余都为0

```
    return cluster
```

2.5 池化

池化目的是为了降维。最后用优化的特征向量表示图像。

```
def new_feature(x,n,k):
    x_new=np.zeros((int(x.shape[0]/n),k.shape[0]*4))
    s = int(np.sqrt(n))
    m = int(np.sqrt(n) / 2)
    for i in range(int(x.shape[0]/n)): # 遍历每张图片
        temp = np.zeros((4, k.shape[0])) # 池化矩阵
        for j in range(2): # 遍历temp
            for h in range(m): # 组成池化块的列
                for l in range(m): # 组成池化块的行
                    temp[j]+=x[i*n+h+l*s+j*m]
                    temp[j+2]+=x[i*n+h+l*s+j*m+int(n/2)]
        x_new[i]=temp.reshape(1,-1)
    return x_new
```

最后便是我们的训练以及分类预测过程，用优化的特征向量表示训练集，放入svm分类器当中进行训练，训练完后，用优化过的特征向量表示测试集，输入到分类器中进行分类预测。本次实验最高效果可以打到90%左右。经过对论文内容的部分复现可以看出，特征的优化相比于模型的建立也一样很重要，即验证了本篇论文的论述点。