手写体数字识别实验

1. 实验介绍

手写体数字识别问题，简而言之就是识别出 10 个阿拉伯数字，由于数字的清晰程度或者是个人的写字习惯亦或是其他，往往手写数字的形状，大小，颜色深浅，书写位置会大不一样。因此让计算机学会识别手写体是一项具有挑战的任务。

2. 实验目标

1） 理解基本的图像分类算法流程以及数据驱动的方法（训练/测试）；
2） 理解训练集/验证集/测试集的划分，学会使用验证集进行超参数调整；
3） 了解梯度方向直方图（HOG）计算原理，并计算图像 HOG 特征；
4） 实现并应用多类支持向量机（Multi-class SVM）分类器；
5） 实现 LeNet5 神经网络，基于此进行图像分类；
6） 对比并分析基于 HOG 的分类算法和基于 LeNet5 网络的分类算法之间的差异。

3. 实验原理

1) 梯度方向直方图

HOG（Histogram of Oriented Gradient）通过直方图的形式来描述图像的梯度方向分布特征。假定图像在(x,y)和(R,G,B)上是连续的，记(x,y)坐标处的颜色值为$f(x,y)$。那么f对坐标的梯度可表示为，$\nabla f(x,y)=(\frac{\partial f}{\partial x},\frac{\partial f}{\partial y})$。对于离散的图像来说，梯度可以表示为，

$$\frac{\partial f}{\partial x}=f(x+1,y)-f(x-1,y),\frac{\partial f}{\partial y}=f(x,y+1)-f(x,y-1)。$$

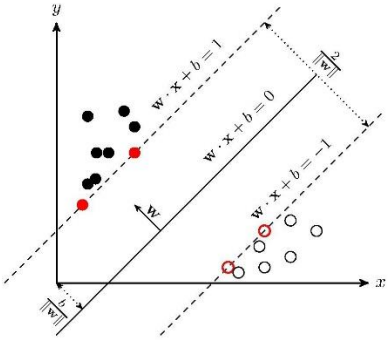
上述式子相当于用模板$[-1,0,1],[-1,0,1]^T$对图像进行卷积，生成x和y方向的梯度值。梯度是一个矢量，有大小和方向，分别表示为$|\nabla f|=\sqrt[2]{\left|\frac{\partial f}{\partial x}\right|^2+\left|\frac{\partial f}{\partial y}\right|^2},\theta=\arctan(\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}})$。通常情况下会对角度加上$\pi/2$，调整范围到$[-\pi,\pi]$。

给定一幅图像，将图像划分为若干网格，每个格子里包含了若干像素。分别统计每个各自的梯度直方图（HOG）。统计 HOG 的时候，按照梯度方向平均分为 9 个范围（bin），每个 bin 的范围是 20°。单独计算每个网格中不同角度的梯度幅度，将落在同一个 bin 的梯度幅度值累加，就得到了一个长度为 9 的直方图。假设一幅的图像，均分为8×8的网格，以2×2个网格作为一组，称为 block。通过滑动窗口的方式，对每个 block 内的 HOG 特征进行归一化处理。最终将所有归一化后的特征合并为一维向量作为图像的 HOG 特征。如果图像大小为80×80，那么图像可划分为10×10个网格，经过滑动窗口得到$(10-1)\times(10-1)$个 block，每个 block 特征为 36 维，最终得到的 HOG 特征大小为$(10-1)\times(10-1)\times36=2916$。

2) 多类支持向量机

支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器。SVM 学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。如下图所示，$\boldsymbol{w}\cdot\boldsymbol{x}+b=0$ 即为分离超平面，对于线性可分的数据集来说，这样的超平面有无穷多个，但是几何间隔最大的分离超平面却是唯一

的。



给定数据集和超平面$\boldsymbol{w}\cdot\boldsymbol{x}+b=0$，计算样本点$(\boldsymbol{x_i},y_i)$到该超平面的距离，记为，

$$\gamma_i = y_i\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\cdot\boldsymbol{x_i}+\frac{b}{\|\boldsymbol{w}\|}\right),$$

所有样本点到超平面的最小距离为，$\gamma = \min\limits_{i=1,2,\dots,N}\gamma_i$。通过最大化该最小间距即可得到目标超平面，该问题可表述为约束最优化问题。

$$\max_{\boldsymbol{w},b}\gamma,$$

$$s.t.\, y_i\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\cdot\boldsymbol{x_i}+\frac{b}{\|\boldsymbol{w}\|}\right)\geq\gamma, i=1,2,\dots,N.$$

基于训练集数据可求解得到目标分类平面参数，

$$\boldsymbol{w}^* = \sum_{i=1,2,\dots,N}\alpha_i^* y_i\boldsymbol{x}_i,$$

$$b^* = y_j - \sum_{i=1,2,\dots,N}\alpha_i^* y_i(\boldsymbol{x}_i\cdot\boldsymbol{x}_j),$$

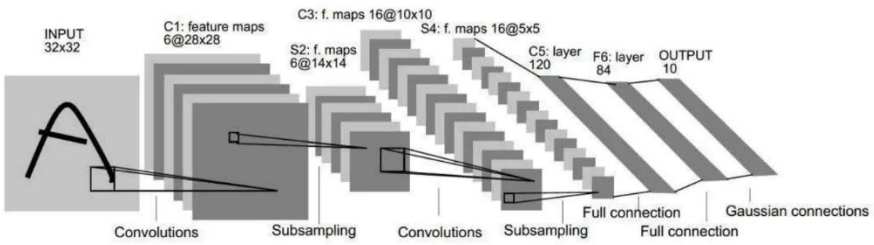
其中α_i为拉格朗日乘子，对于任意训练样本 $(\boldsymbol{x}_i,y_i)$，总有 $\alpha_i=0$ 或者 $y_j(\boldsymbol{w}\cdot\boldsymbol{x}_j+b)=1$。若 $\alpha_i=0$，则该样本不会在最后求解模型参数的式子中出现。若 $\alpha_i>0$，则必有 $y_j(\boldsymbol{w}\cdot\boldsymbol{x}_j+b)=1$，所对应的样本点位于最大间隔边界上，是一个支持向量。这显示出支持向量机的一个重要性质：训练完成后，大部分的训练样本都不需要保留，最终模型仅与支持向量有关。

多类支持向量机原理是将多类分类问题转换为若干个二分类问题，包含成对分类方法和一类对余类分类方法。本实验介绍后者，对于每一类，将其作为正类，其余所有类别作为负类，构造二分类器。如果数据集总共有M个类别，则可以构建$(M-1)$个二分类器。给定新数据进行决策时，计算新数据到M个决策平面的带符号距离，选择最大距离对应的类别作为对新数据的预测结果。

3) 基于神经网络的分类算法

利用 PyTorch 搭建神经网络。PyTorch 是一个开源的深度学习框架，基于 Torch，用于自然语言处理，计算机视觉等应用，其支持 GPU，具备较强的灵活性，支持动态神经网络，易于理解。通过构建神经网络，对手写体数字数据集进行训练，并更新网络参数，使网络具备识别 手写体数字的能力。

本实验采用 1994 年由 YannLeCun 提出的 LeNet5 作为模型。LeNet5 包含 2 个 5x5 的卷积层，2 个 2x2 的池化层和 3 个全连接层，其结构示意图如下，



4. 数据集介绍

本实验采用 MNIST 数据集，训练集包含 6 万张 28×28 像素点的灰度图片和 6 万个对应的 标签 label，测试集包含 1 万张 28×28 像素点的灰度图片和 1 万个对应的 label。

数据集网站：http://yann.lecun.com/exdb/mnist/，网站里面有详细的数据集介绍。

Four files are available on this site:

train-images-idx3-ubyte.gz:	training set images (9912422 bytes)
train-labels-idx1-ubyte.gz:	training set labels (28881 bytes)
t10k-images-idx3-ubyte.gz:	test set images (1648877 bytes)
t10k-labels-idx1-ubyte.gz:	test set labels (4542 bytes)

训练集：train-images-idx3-ubyte.gz（6 万张图像），train-labels-idx1-ubyte.gz（6 万张图像对应的标签）；测试集：t10k-images-idx3-ubyte.gz（1 万张图像），t10k-labels-idx1-ubyte.gz（1 万张图像对应的标签）

5. 实验内容

（1）实现基于 HOG 特征和多类 SVM 的分类算法，使用 MNIST 数据集训练并测试。

（2）基于 pytorch 实现 LeNet5，使用 MNIST 数据集训练并测试网络。

具体要求：

（1） 计算基于 HOG 特征的分类算法在训练集、测试集的准确度，可视化多类 SVM 分类平面

（2） 实现 LeNet5 并画出训练集、测试集 loss 曲线，训练集、测试集准确率曲线

（3） 调整不同学习率，观察并分析对训练 loss 的影响（要有文字和图片说明）

（4） 使用不同的优化器，观察并分析对训练 loss 的影响（要有文字和图片说明）

（5） 对比并分析两种分类算法结果

（6） 使用绘图软件自己制作手写数字（0-9）进行测试

参考论文：http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf