第一部分：简述题

1.1

PCA 降维算法的主要计算步骤是：

(1) 计算样本均值 $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

(2) 计算样本的协方差矩阵 $S = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x})^T$

(3) 对 $D \times D$ 的协方差矩阵，计算特征值分解

(4) 挑选前 K 大特征值对应的特征向量组成 $D \times K$ 的矩阵 U

(5) 使用 U 矩阵对样本降维: $z_i = U^T x_i$

1.2

结构风险最小化，就是使测试误差最小化。而经验风险最小化是使训练误差最小化。VC 维的数值是：对于某模型，不断加大数据量，随机给数据赋与标签，直到分类器不能区分。这时的数据量就是模型 VC 维。VC 维定义了模型的复杂度，模型越复杂，VC 维越大，模型泛化性能越差。

1.3

设对于每一个样本 x_i，都有 y_i，当为正样本时，$y_i = 1$，为负样本时，$y_i = -1$。分类平面为，

$$\mathbf{w}^T\mathbf{x} + b = 0$$

定义 ρ 为分类平面间隔，则对于任意样本有，

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho/2$$

由于参数除以一个常数不影响结果，简化为，

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

作为支持向量的样本点满足分类距离最小，因此可以对上述不等式取等，

$$y_s(\mathbf{w}^T\mathbf{x}_s + b) = 1$$

支持向量机的核心思想为最大化分类间隔，分类间隔为支持向量到分类平面几何距离的 2 倍，计算公式如下，

$$\rho = 2\frac{|\mathbf{w}^T\mathbf{x}_s + b|}{||\mathbf{w}||^2} = \frac{2}{||\mathbf{w}||^2}$$

因此最大化分类间隔转化为最小化 $||\mathbf{w}||^2$ 的问题。同时应该保证样本都分对，所以最终的优化方程为，

$$\min \ \frac{1}{2}||\mathbf{w}||^2$$

$$s.t. \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \ \forall i = 0, 1, \ldots, n$$

1.4

SVM 软间隔的优化目标为，

$$\min \ \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t. \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \ \forall i = 0, 1, \ldots, n$$

$$\xi_i \geq 0, \ \forall i = 0, 1, \ldots, n$$

目标函数的意义为在满足要求的情况下，最大化分类间隔和最小化误差。因为 $hinge(x) = max(1 - x, 0)$，所以可以把上述优化目标改为

$$\min \ \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}hinge(y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

通过 Hinge Loss 的表示方法，对软间隔求解的约束被放到了优化函数中。Hinge Loss 是 0/1 损失函数的上界，而且可以最小化，相当于最小化 0/1 损失。相比较于 0/1 损失，Hinge Loss 对于分类正确的点没有变化，对于分错的点，根据距离体现其错误的程度，而不是像 0/1 损失那样把错分点一视同仁。

1.5

对于原始线性不可分的数据，可以将其投影到高维空间，在高维空间中线性可分。

因此核方法的思想是，寻找合适的核函数，对原始数据做变换，得到线性可分数据。值得注意的是，由于数据往往以 $\mathbf{x}\mathbf{x}^T$ 的形式存在，所以直接对内积进行投影即可。

第二部分：编程题

本次实验从 MNIST 数据集中选择 0，1 类，使用 LIBSVM 工具对其进行分类。具体程序请见 data_process.py、build_feature.py、svm.py 文件。

实现主要分为 3 个步骤，首先使用 data_process.py 脚本，解析数据集并存储 0，1 数据到指定位置。之后使用 build_feature.py 脚本，分别对训练集和数据集图片提取 HOG 特征，并将特征按 LIBSVM 要求的格式存储。最后使用 svm.py 脚本，训练模型，进行测试。

SVM 采用了 radial basis function 核函数，最优参数 C 和 gamma 由 LIBSVM 自带的网格参数寻优脚本确定。

```
[kaifeideMacBook-Air:tools kaifeiwang$ grid.py -log2c 0,8,2 -log2g -1,-5,-2 /User
s/kaifeiwang/Desktop/课程资料/模式识别/作业/assignment6/data/train/feature.txt
/Users/kaifeiwang/Desktop/课程资料/模式识别/作业/assignment6/data/train/feature.
txt
gnuplot executable not found
[local] 4.0 -3.0 99.7394 (best c=16.0, g=0.125, rate=99.7394)
[local] 2.0 -3.0 99.7158 (best c=16.0, g=0.125, rate=99.7394)
[local] 4.0 -1.0 99.7315 (best c=16.0, g=0.125, rate=99.7394)
[local] 2.0 -1.0 99.7473 (best c=4.0, g=0.5, rate=99.7473)
[local] 8.0 -3.0 99.6921 (best c=4.0, g=0.5, rate=99.7473)
[local] 8.0 -1.0 99.7315 (best c=4.0, g=0.5, rate=99.7473)
[local] 0.0 -3.0 99.6368 (best c=4.0, g=0.5, rate=99.7473)
[local] 0.0 -1.0 99.7315 (best c=4.0, g=0.5, rate=99.7473)
[local] 4.0 -5.0 99.7079 (best c=4.0, g=0.5, rate=99.7473)
[local] 2.0 -5.0 99.6447 (best c=4.0, g=0.5, rate=99.7473)
[local] 8.0 -5.0 99.6842 (best c=4.0, g=0.5, rate=99.7473)
[local] 0.0 -5.0 99.5894 (best c=4.0, g=0.5, rate=99.7473)
[local] 6.0 -3.0 99.6921 (best c=4.0, g=0.5, rate=99.7473)
[local] 6.0 -1.0 99.7315 (best c=4.0, g=0.5, rate=99.7473)
[local] 6.0 -5.0 99.7 (best c=4.0, g=0.5, rate=99.7473)
4.0 0.5 99.7473
```

图 1: 网格参数寻优

采用五折交叉验证，从图 1 可以看到最优的 C 为 4，gamma 为 0.5。使用找到的最

优参数训练模型，如图 2 所示，在测试集上的结果准确度达到了 99.95%

```
kaifeideMacBook-Air:assignment6 kaifeiwang$ python svm.py
.*
optimization finished, #iter = 1668
nu = 0.009380
obj = -253.757417, rho = 0.301215
nSV = 478, nBSV = 22
Total nSV = 478
Accuracy = 99.9527% (2114/2115) (classification)
```

图 2: 测试结果示意