**《计算机视觉》课内实践练习一**

——基于SVM和Softmax损失函数的线性分类器

* **作业截止时间**

请于该时间之前提交作业：2023年11月15日，5:00 pm

* **团队合作**

**允许自由组建团队合作完成本次实践练习（团队人数上限3人）。每个团队共同提交一份作业文件**，作业部分成绩按团队整体完成质量评价，线上考评部分成绩按团队个人表现评价。请在Jupyter Notebook(.ipynb)文件对应位置处注明团队各个成员的姓名和学号。

* **考察方式**

作业收齐后，将择期开展线上会议（具体时间另行通知，一般在周末），对各团队作业完成情况进行考评，考评人员（老师或助教）将随机指定程序功能或代码，要求进行演示和讲解。团队的**每位成员**都将被要求独立进行演示和回答问题。

本次实习占总成绩比例为20%；个人实习成绩 = 团队作业评分(10%) + 个人线上考评表现(10%)

线上会议室将设等候室，考评过程中各团队将轮流进入会场，团队每位组员将被要求开启摄像头截屏验证身份，故请各位同学开会前调试好设备，并选择适宜的场地。

* **诚信问题**

严禁不同团队间代码的拷贝或抄袭，经代码查重检测确定为抄袭的作业，实习成绩将为0分。

# 实习要求

1. 完成对指定HelloRS32数据集的读取和抽样展示；
2. 实现SVM和Softmax损失函数的损失值及其解析梯度计算，并将解析梯度与数值梯度进行对比验证；
3. 实现简单的基于SVM和Softmax损失的线性分类器训练，打印训练过程，输出模型在训练数据集上的总体精度，比较两个分类器的训练用时；
4. 将HelloRS32数据集分为训练集(train)、验证集(val)和测试集(test)，并通过调校超参数找到两个分类器的最优超参数组合，然后输出分类器在测试集上的总体精度；
5. 确定最优超参数后，将两个线性分类器的权重可视化为RGB图像进行显示；
6. 从测试集中对每个类别的图像样本进行抽样展示，展示时设法对正确预测和错误预测的样本进行区分，分析与定量评价的精度是否相符。

请仔细阅读以下内容，认真理解其中的提示信息，分步骤依次解决课内实践中的问题，完成代码编写和测试，最后按要求提交作业。

# **任务准备**

环境配置：安装Anaconda，它是Python的一个发布版，包含了最流行的科研、数学、工程和数据分析Python包；完成Jupyter Notebook编程环境的配置和准备。

数据准备：下载本次实践练习资料，包括HelloRS32数据集和作业初始代码。

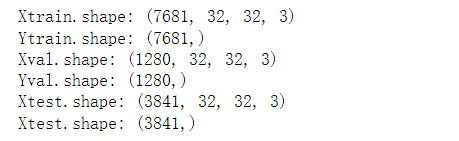
# **HelloRS32数据集图像的读取**

在Jupyter Notebook中加载HelloRS32数据集并完成数据图像的显示，完成load\_data函数关于用于数据集的读取部分。函数visualize\_samples用于显示训练集数据中各个类别的部分图像。

代码示例：



结果示例：



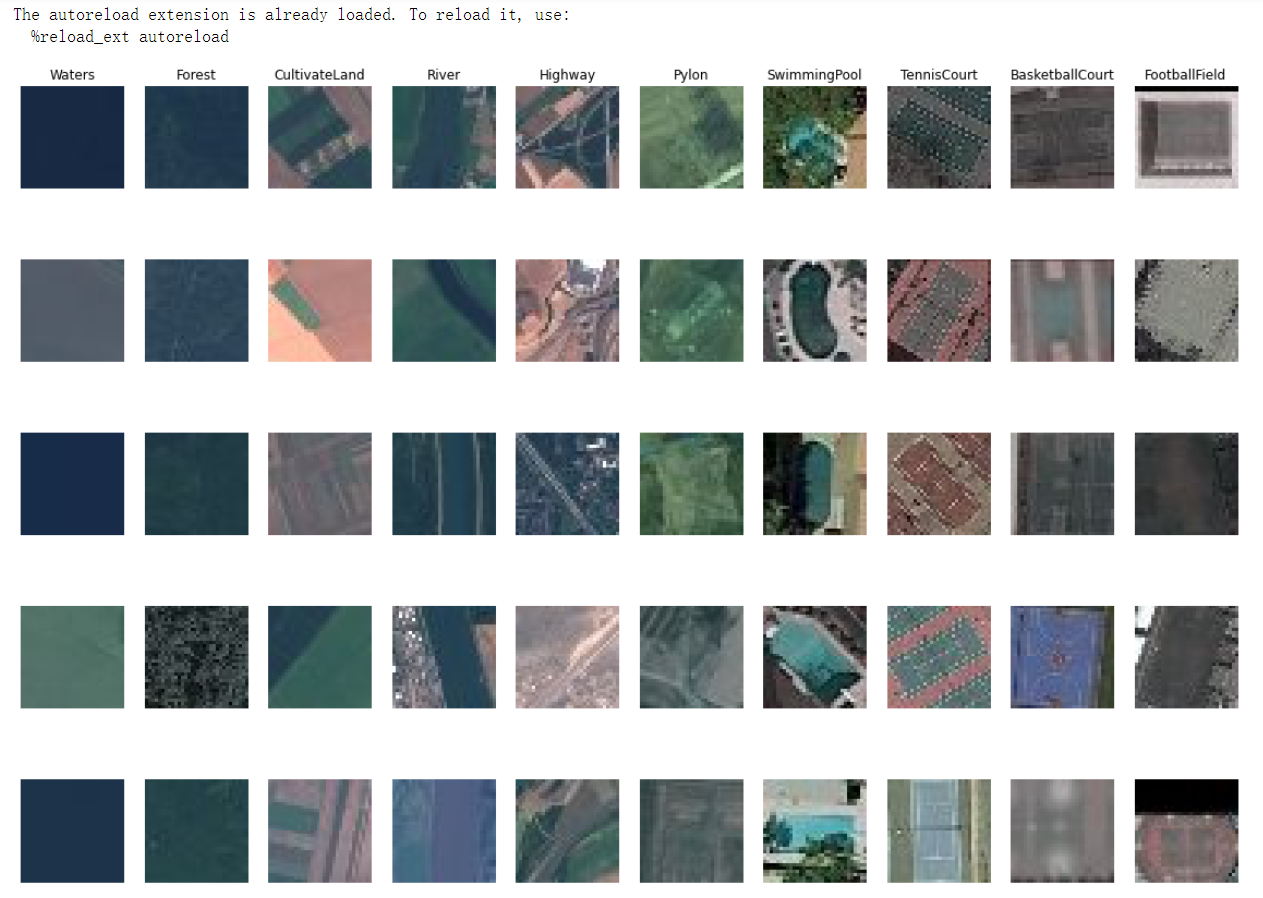
# 数据预处理

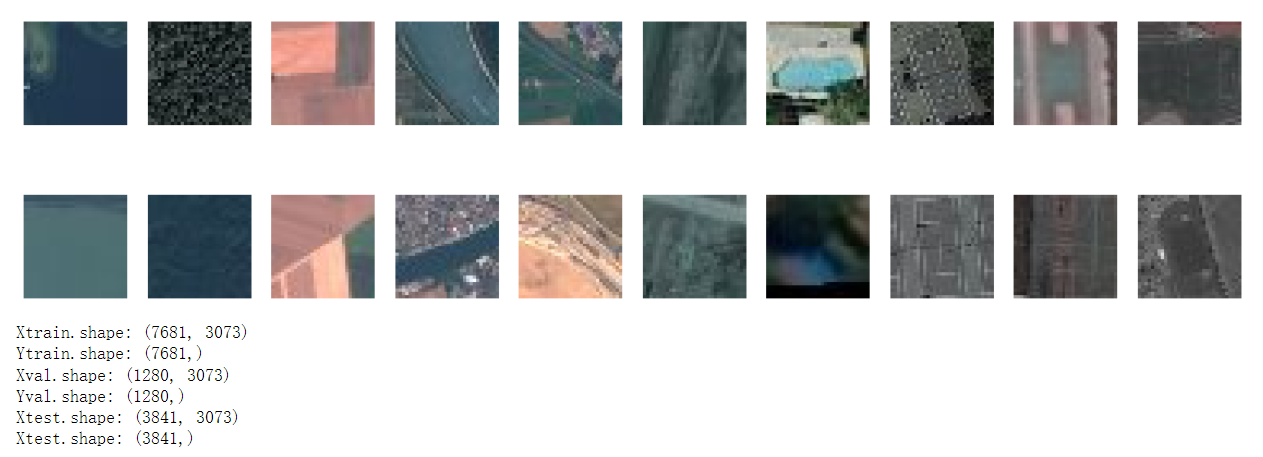
函数data\_ processing用于数据的预处理。使用图像去均值法进行数据预处理，完成数据预处理后分别输出训练集、验证集和测试集的大小。

代码示例：



结果示例：





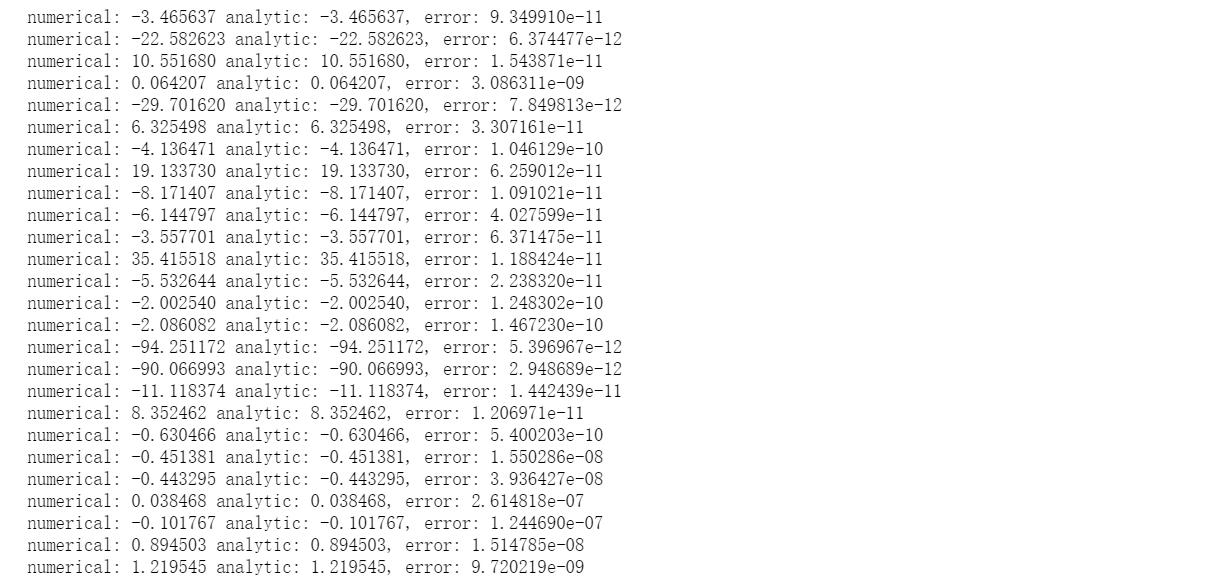
# 比较数值梯度与解析梯度

首先对SVM和Softmax进行损失计算，分别由函数svm\_loss\_naive()和函数softmax\_loss\_naïve()实现，获得损失值后，分别计算SVM与Softmax的数值梯度与解析梯度，同时实现函数svm\_grad\_check()与softmax\_grad\_check()进行梯度验证，对数值梯度和解析梯度进行比较。

代码示例：



结果示例：

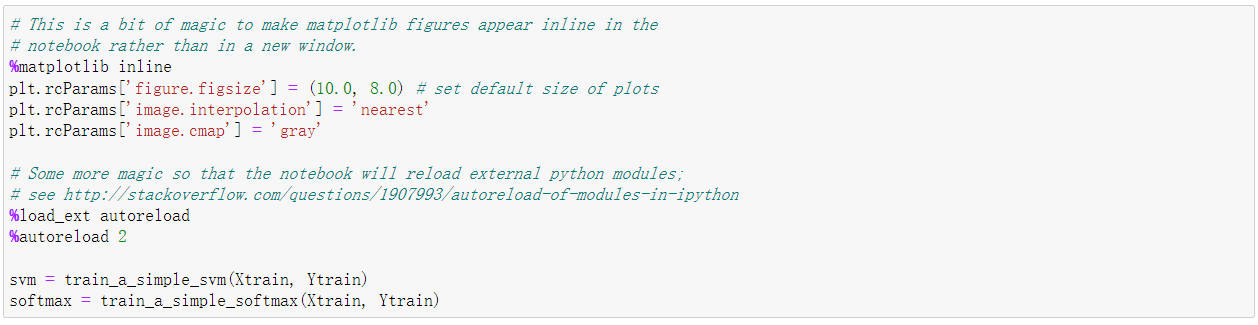


其中numerical为数值梯度值，analytic为解析梯度值，error为误差值。

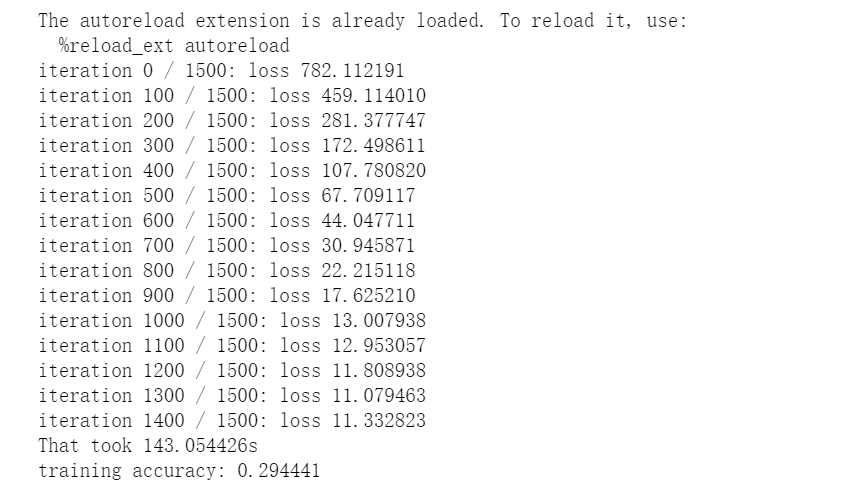
# 训练简单的SVM和Softmax线性分类器，并比较运行时间

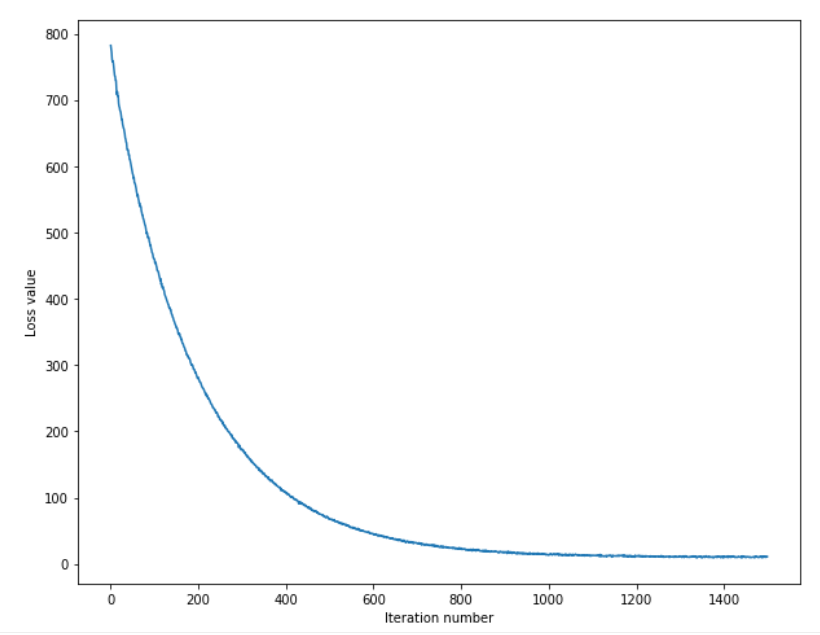
完成一个简单的SVM和Softmax训练，在训练过程中每迭代一定次数保存输出损失函数值并绘制迭代次数与损失函数的曲线图，最后输出训练耗费的总时间与训练精度。训练分别使用函数train\_a\_simple\_svm()和train\_a\_simple\_softmax()实现。

代码示例：



结果示例：



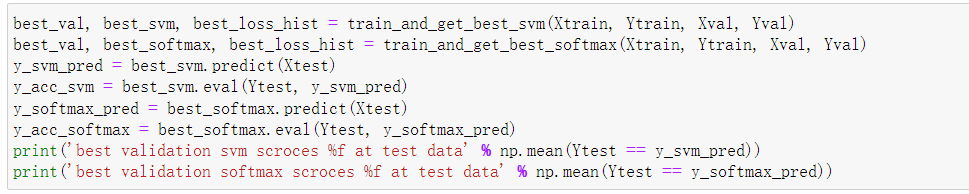


在上例中共迭代1500次，每迭代100次输出损失值并绘制迭代次数与损失函数的曲线图，最终输出训练花费的总时长和训练的精度。

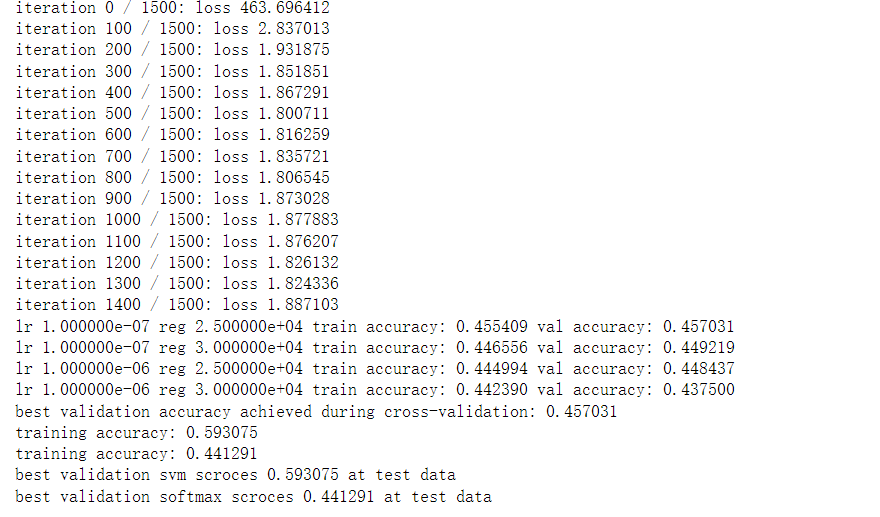
# 训练并找到最佳精度的SVM和Softmax分类器

不断调整超参数（学习率和正则化强度）对SVM和Softmax进行训练，找到具有最佳精度的一组模型参数。训练过程中输出不同参数下损失函数随迭代次数的变化值，训练完成后输出模型参数、训练精度与验证精度，进行比较输出具有最佳精度的SVM和Softmax。示例使用函数train\_and\_get\_best\_svm()和train\_and\_get\_best\_softmax()对模型进行训练，函数best\_svm.predict()和best\_softmax.predict进行测试。

代码示例：



结果示例：

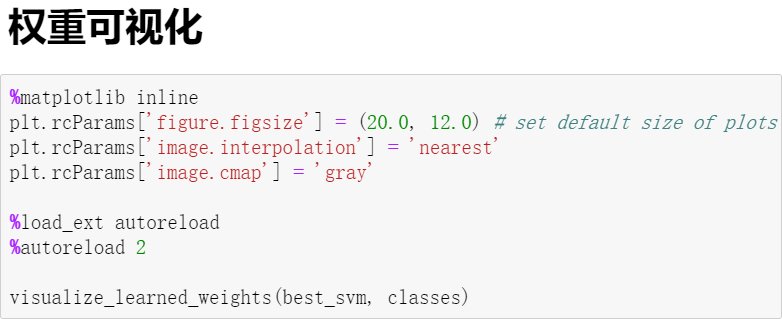


上例中仅选择两组学习率与正则化强度进行超参数调整，比较并选择其中表现最佳的超参数组合，同学们在实习中可进行更多的尝试和调整。

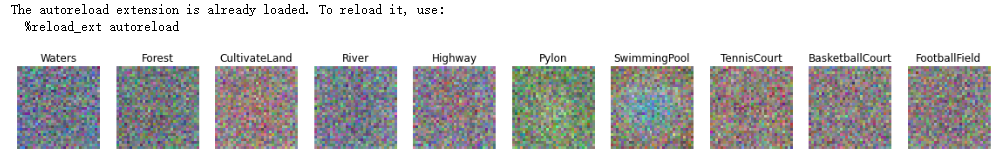
# 权重可视化

使用函数visualize\_learned\_weights()实现权重的可视化，直观地理解分类器到底学习到了什么。

代码示例：



结果示例：



# 作业提交要求

1. 在Jupyter Notebook下运行svm\_softmax\_sgd.ipynb笔记本，首先需在笔记本开头处填写学号和姓名（如下图所示）；完成全部功能和输出相应结果后，将该笔记本内容打印为PDF文件提交，PDF需统一重命名为**队长学号\_队长姓名.pdf**。
2. 实习过程中需不断更新svm\_softmax\_sgd.py中的代码，实习完成后，该源码文件需作为作业附件提交用于检查，文件需统一命名为**队长学号\_队员学号\_队员学号.py**。

