

CS231n第一周内容导学

R.....

(江小河)

每周时间安排

周一	理论课时
周二	理论课时
周三	休息
周四	理论课时
周五	编程作业
周六	编程作业
周日	编程作业（解答）

第一周课程安排——绪论、机器学习

周一	绪论
周二	KNN
周三	休息
周四	线性分类器
周五	损失函数（部分）
周六	编程作业-KNN
周日	编程作业-SVM（可延后）

课程1-主要内容

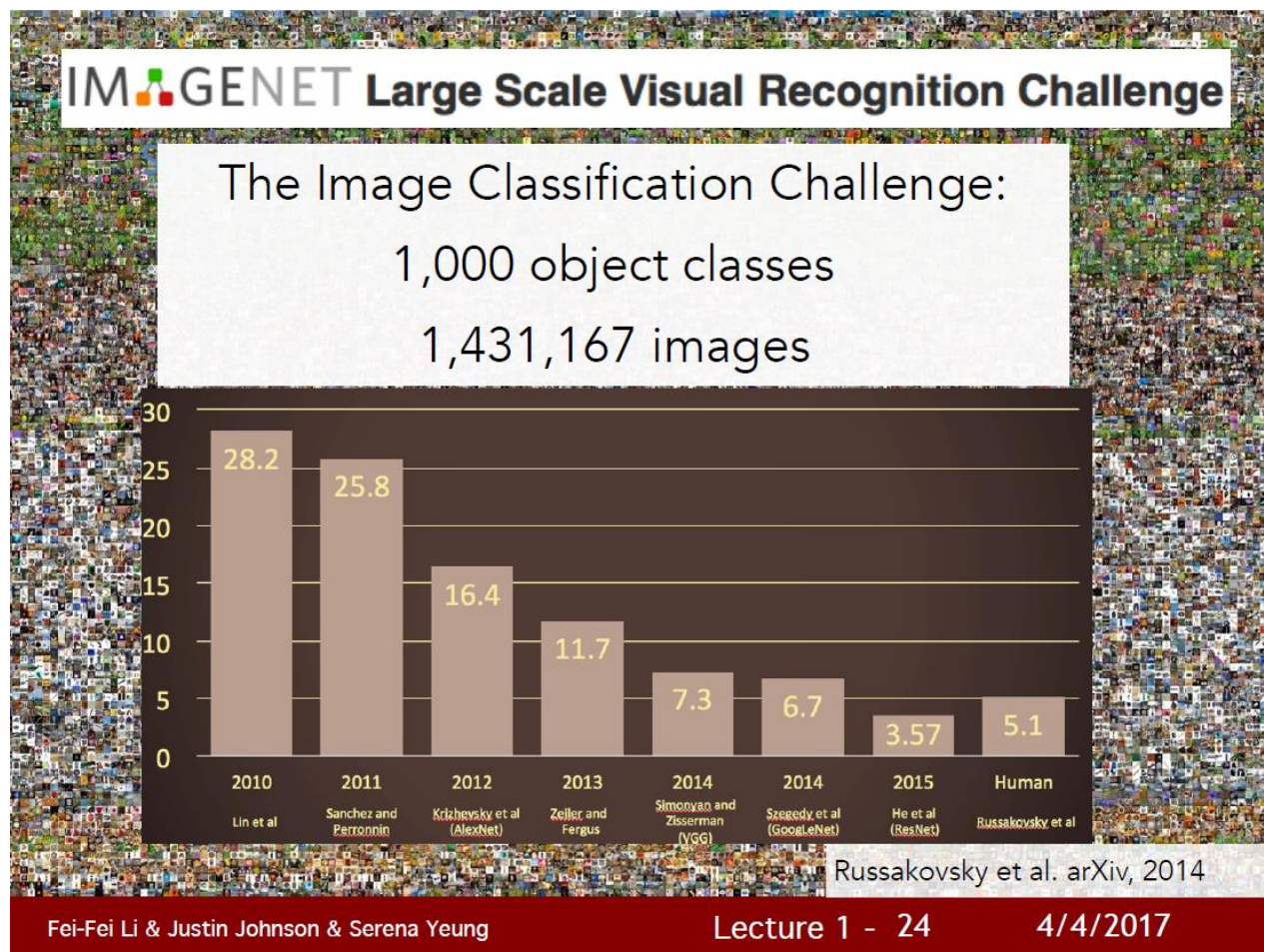
- 对应云课堂课程（第一讲）：
 - <https://study.163.com/course/courseLearn.htm?courseId=1004697005#/learn/video?lessonId=1049395813&courseId=1004697005>
- 对应Youtube课程：
 - <https://www.youtube.com/watch?v=vT1JzLTH4G4&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv>
- 主要内容：
 - 计算机视觉简史
 - 课程概述

Today's agenda

- A brief history of computer vision
- CS231n overview

课程1-CV发展史

- 首先理解概念，什么是计算机视觉？
——让计算机理解图像内容
- 经典算法：
 - 很简短的介绍，但其实是非常有学习意义的研究成果，对新算法的研究依然有很高的借鉴价值
- 深度学习方法：
 - ImageNet比赛，可以说是深度学习在CV领域的先驱
 - 注意这里及之后每年的深度模型的发展趋势，层数变化



课程1-CV发展史

- 计算机视觉任务的应用：
 - 图像分类
 - 目标检测
 - 行为识别
 - 图像描述
 -



Outside border images, clockwise, starting from top left:

Image by Pop Culture Geek is licensed under CC BY 2.0; changes made

Image by the US Government is in the public domain

Image by the US Government is in the public domain

Image by Glogger is licensed under CC BY-SA 3.0; changes made

Image by Sylenius is licensed under CC BY 3.0; changes made

Image by US Government is in the public domain

Inside four images, clockwise, starting from top left:

Image is CC0 1.0 public domain

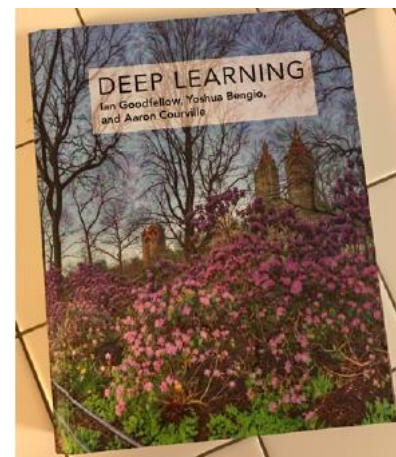
Image by Tucania is licensed under CC BY-SA 3.0; changes made

Page by Intuitive Surgical, Inc. is licensed under CC BY-SA 3.0; changes made

Image by Oyundari Zorigtbaatar is licensed under CC BY-SA 4.0

课程1-课程后勤

- 这部分内容基本上就是.....
 - 假装自己坐在Stanford的教室里.....
 - 假装自己要交作业.....
 - 好好学习一下英语听力
- 一些值得注意的点：
 - 关于花书：书本开源，书很好，就是厚.....
 - 关于课程基础：
 - 编程：Python
 - 数学：微积分和线性代数
 - 机器学习：有基础最好，没有也OK



课程2-图像分类流程

- 主要内容：
 - 图像分类流程
 - K近邻分类器
 - 线性分类器
- 课程开始会讲述一些发生在过去的行政安排
 - 关注一下Assignment1中需要提交的内容，是我们即将学习的内容
 - 提供了一个简易的numpy教程：
 - Python Numpy Tutorial
 - <http://cs231n.github.io/python-numpy-tutorial/>

课程2-K近邻分类器

- 注意本节课程的“分类流程”主要是通过KNN分类器讲解的
- 主要内容：
 - KNN图像分类流程（原理）
 - 训练阶段和预测阶段
 - **Cifar10数据集**
 - 概念 __类__张图像
 - 下载
 - 测试
 - 实验*（科学研究的方法）
 - 交叉验证
 - 缺点
 - 线性分类器

```
def train(images, labels):  
    # Machine learning!  
    return model
```

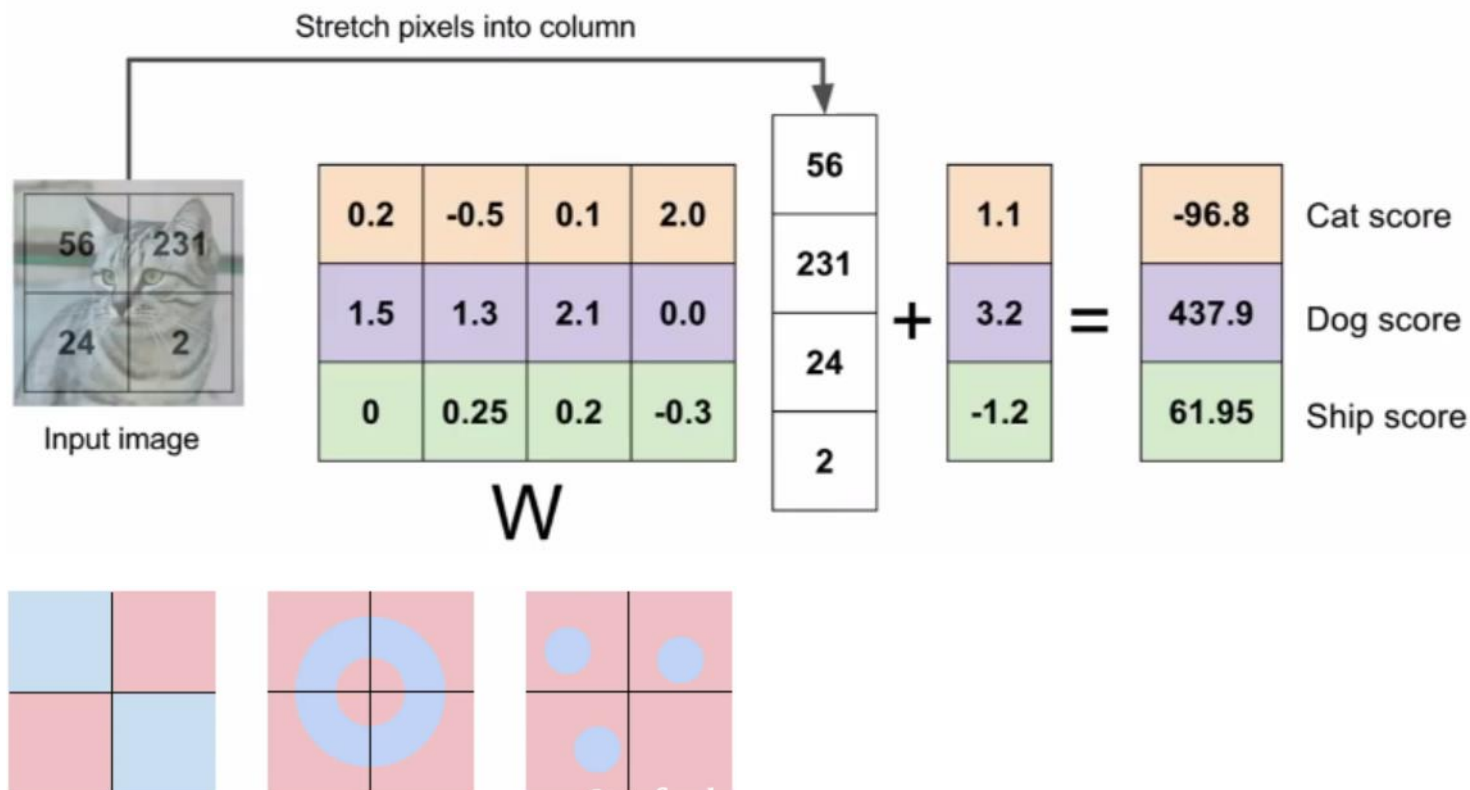
```
def predict(model, test_images):  
    # Use model to predict labels  
    return test_labels
```

课程2-K近邻分类器

- 注意本节课程的“分类流程”主要是通过KNN分类器讲解的

- 主要内容:

- KNN图像分类器
- 线性分类器
 - 什么是线性分类器?
 - 特征提取
 - 计算过程
 - 如何得到线性分类器?
 - 损失函数
 - 反相传播
 - 结果分析
 - 图像分类模板*
 - 高维决策空间
 - 线性分类器局限



课程3-损失函数及优化方法

- 主要内容:

- 内容回顾: 前4分40秒

- 损失函数

- 判断 W (模型) 的好坏

- 多分类SVM loss

- 正则化

- Softmax

- Softmax与multiclass SVM loss对比

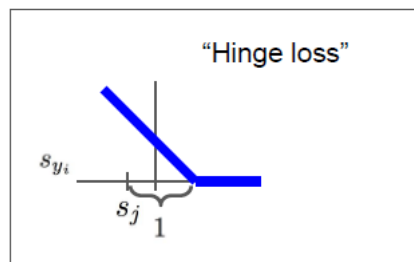
- 优化器

- 随机选择

- 梯度下降

- 计算微积分

Multiclass SVM loss:



$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$
$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

课程2-K近邻分类器-练习

- 下载数据集：
 - Linux系统下，使用assignment1\cs231n\datasets下的get_datasets.sh脚本下载Cifar10并解压



get_datasets.sh









- Win下通常缺乏所需命令行工具，分析脚本内容

课程2-K近邻分类器-练习

- 下载数据集:















```
# Get CIFAR10
wget http://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
tar -xzf cifar-10-python.tar.gz
rm cifar-10-python.tar.gz
```

- 从<http://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>下载数据集，放在assignment1\cs231n\datasets目录下，解压，注意win下的解压工具会多创建一层文件夹（路径）

 batches.meta	2009/3/31 12:45	META 文件	1 KB
 data_batch_1	2009/3/31 12:32	文件	30,309 KB
 data_batch_2	2009/3/31 12:32	文件	30,308 KB
 data_batch_3	2009/3/31 12:32	文件	30,309 KB
 data_batch_4	2009/3/31 12:32	文件	30,309 KB
 data_batch_5	2009/3/31 12:32	文件	30,309 KB
 readme.html	2009/6/5 4:47	HTML 文档	1 KB
 test_batch	2009/3/31 12:32	文件	30,309 KB

课程2-K近邻分类器-练习

- 打开我们的第一次作业文件
 - 环境配置教程请咨询班主任

	.ipynb_checkpoints	2019/5/28 22:20	文件夹
	cs231n	2019/5/28 22:18	文件夹
	.gitignore	2017/4/7 16:37	文本文档
	collectSubmission.sh	2017/4/7 16:37	SH 文件
	features.ipynb	2017/4/7 16:37	IPYNB 文件
	frameworkpython	2017/4/7 16:37	文件
	knn.ipynb	2019/5/28 23:04	IPYNB 文件
	README.md	2017/4/7 16:37	MD 文件
	requirements.txt	2017/4/7 16:37	文本文档
	setup_googlecloud.sh	2017/4/7 16:37	SH 文件
	softmax.ipynb	2017/4/7 16:37	IPYNB 文件
	start_ipython_osx.sh	2017/4/7 16:37	SH 文件
	svm.ipynb	2017/4/7 16:37	IPYNB 文件
	two_layer_net.ipynb	2017/4/7 16:37	IPYNB 文件

- 对于win:
 - 修改数据集路径

```
In [2]: # Load the raw CIFAR-10 data.
# cifar10_dir = 'cs231n/datasets/cifar-10-batches-py'
import os
cifar10_dir = os.path.join("cs231n", "datasets", "cifar-10-batches-py")
X_train, y_train, X_test, y_test = load_CIFAR10(cifar10_dir)

# As a sanity check, we print out the size of the training and test data.
print('Training data shape: ', X_train.shape)
print('Training labels shape: ', y_train.shape)
print('Test data shape: ', X_test.shape)
print('Test labels shape: ', y_test.shape)
```

Training data shape: (50000, 32, 32, 3)
Training labels shape: (50000,)
Test data shape: (10000, 32, 32, 3)
Test labels shape: (10000,)

课程2-K近邻分类器-练习

- 没有依赖包的情况下进行安装补充

```
-----  
ModuleNotFoundError                                Traceback (most recent call last)  
<ipython-input-8-833c0369b4cf> in <module>()  
----> 1 from cs231n.classifiers import KNearestNeighbor  
      2  
      3 # Create a kNN classifier instance.  
      4 # Remember that training a kNN classifier is a noop:  
      5 # the Classifier simply remembers the data and does not train  
  
H:\江小河的教学视频\2019.05.27 My tuition of CS231n2017\2019.05.231n\classifiers\__init__.py in <module>()  
----> 1 from cs231n.classifiers.k_nearest_neighbor import *  
      2 from cs231n.classifiers.linear_classifier import *  
  
H:\江小河的教学视频\2019.05.27 My tuition of CS231n2017\2019.05.231n\classifiers\k_nearest_neighbor.py in <module>()  
      1 import numpy as np  
----> 2 from past.builtins import xrange  
      3  
      4  
      5 class KNearestNeighbor(object):  
  
ModuleNotFoundError: No module named 'past'
```

- 注意安装包的名称

[anaconda python no module named 'past'的解决方法- Joanna-In ...](https://www.cnblogs.com/studylyn/p/7517481.html)

<https://www.cnblogs.com/studylyn/p/7517481.html> ▼

2017年9月13日 - 如上图所示, 错误就是: No module named 'past'. 解决办法不是下载'past'包, 而是下载'future'包: 我是安装了anaconda集成环境, python的单独 ...

插播新闻

```
(tf1.9) C:\Users\jianghe>conda install future
Solving environment: failed

CondaHTTPError: HTTP 404 NOT FOUND for url <https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/noarch/repodata.json>
Elapsed: 00:00.245629

The remote server could not find the noarch directory for the
requested channel with url: https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/noarch/repodata.json

As of conda 4.3, a valid channel must contain a `noarch/repodata.json` and
associated `noarch/repodata.json.bz2` file, even if `noarch/repodata.json` is
empty. please request that the channel administrator create
`noarch/repodata.json` and associated `noarch/repodata.json.bz2` files.
$ mkdir noarch
$ echo '{}' > noarch/repodata.json
$ bzip2 -k noarch/repodata.json

You will need to adjust your conda configuration to proceed.
Use `conda config --show channels` to view your configuration's current state.
Further configuration help can be found at <https://conda.io/docs/config.html>.
```

Anaconda 源使用帮助— USTC Mirror Help 文档

<https://mirrors.ustc.edu.cn/help/anaconda.html> ▼

警告. 由于合规性, **Anaconda** 源目前已经无限期停止服务。 ... Miniconda 安装包可以在 <https://mirrors.ustc.edu.cn/anaconda/miniconda/> 下载。 **添加**USTC仓库镜像: .

📰 镜像站新闻

关于停止Anaconda镜像服务的通知

📅 2019-04-16 👤 TUNA Staff

根据 Anaconda [软件源](#)上的说明, Anaconda 和 Miniconda 是 Anaconda, Inc. 的商标, 任何未经授权的公开镜像都是不允许的。去年我们曾尝试与公司有关人员联系, 但未能取得授权。

在没有上游授权的情况下, 我们无法保证镜像的合法性与服务质量。因此我们决定, 在取得授权之前无限期停止 Anaconda 镜像服务。即日起, 我们将停止 Anaconda 的更新并隐藏镜像入口链接。一个月后, 彻底关闭 Anaconda 镜像的文件下载。请现有用户尽快切换至官方下载地址, 以免影响正常使用。

感谢您的理解与支持!

Anaconda 清华源关闭后, 怎么办?



Neo lei

671 人赞同了该文章



课程2-K近邻分类器-练习

- 有些情况需要修改引入包的顺序

```
In [2]: # Run some setup code for this notebook.
from __future__ import print_function

import random
import numpy as np
from cs231n.data_utils import load_CIFAR10
import matplotlib.pyplot as plt
```

- 运行到这里没有问题就可以继续作业内容了。
- 作业的解答助教会在周末进行讲解

```
In [14]: # Let's compare how fast the implementations are
def time_function(f, *args):
    """
    Call a function f with args and return the time (in seconds) that it took to execute.
    """
    import time
    tic = time.time()
    f(*args)
    toc = time.time()
    return toc - tic

two_loop_time = time_function(classifier.compute_distances_two_loops, X_test)
print('Two loop version took %f seconds' % two_loop_time)

one_loop_time = time_function(classifier.compute_distances_one_loop, X_test)
print('One loop version took %f seconds' % one_loop_time)

no_loop_time = time_function(classifier.compute_distances_no_loops, X_test)
print('No loop version took %f seconds' % no_loop_time)

# you should see significantly faster performance with the fully vectorized implementation

Two loop version took 0.066047 seconds
One loop version took 0.000000 seconds
No loop version took 0.001001 seconds
```