获得授权翻译斯坦福CS231n课程笔记系列

获得授权翻译斯坦福CS231n课程笔记系列

CS231n课程笔记翻译：Python Numpy教程

CS231n 课程笔记翻译：Python Numpy教程
numpy for Matlab users

CS231n课程笔记翻译：图像分类笔记（上）

CS231n 课程笔记翻译：图像分类笔记（上）

⋆ 图像分类

DEF: 图像分类

所谓图像分类问题，就是已有固定的分类标签集合，然后对于输入的图像，从分类标签集合中找出一个分类标签，最后把分类标签分配给该输入图像。

困难和挑战： 视角变化(Viewpoint variation)、大小变化(Scale variation)、形变(Deformatio
遮挡(Occlusion)、光照条件(Illumination conditions)、背景干扰(Background clutter)、类内差异(Intra-class variation)

DEF: 数据驱动方法

给计算机很多数据，然后实现学习算法，让计算机学习到每个类的外形。这种方法，就是数据驱动方法。

数据驱动方法的第一步就是收集已经做好分类标注的图片来作为训练集。

图像分类流程： 输入(输入训练集) $\Longrightarrow$学习(训练分类哭或者学习模型)$\Longrightarrow$评价(评价分类器)

⋆ Nearest Neighbor分类器

图像分类数据集： CIFAR-10

DEF: Nearest Neighbor算法

Nearest Neightbor算法将会拿差测试图片和训练集中每一张图片比较，然后将它认为最相似的那个训练集图片的标签赋给这张测试图片。

- **L1距离**

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

- **L2距离**

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

EX: Nearest Neighbor for CIFAR-10

基于CIFAR-10数据的Kaggle算法竞赛排行榜：CIFAR-10 - Object Recognition in Images

★ K-Nearest Neighbor分类器

DEF: K-Nearest Neighbor分类器

其思想：与其只找最相近的那1个图片的标签，不如找最相似的k个图片的标签，然后让他们针对测试图片进行投票，最后把票数最高的标签作为对测试图片的预测。

CS231n课程笔记翻译：图像分类笔记（下）

CS231n 课程笔记翻译：图像分类笔记（下）

★ 用于超参数调优的验证集

DEF: 超参数(hyperparameter)

k-NN分类器需要设定k值，那么选择哪个k值最合适的呢？对于不同的距离函数，比如L1范数和L2范数等，那么选哪个好？还有不少选择需要考虑到(比如：点积)。所有这些选择，被称为**超参数(hyperparameter)**。

Remark: 决不能使用测试集来进行调优。

测试数据集只使用一次，即在训练完成后评价最终的模型时使用。

当你在设计机器学习算法的时候，应该把测试集看做非常珍贵的资源，不到最后一步，绝不使用它。如果你使用测试集来调优，而且算法看起来效果不错，那么真正的危险在于：算法实际部署后，性能可能会远低于预期。这种情况，称之为算法对测试集**过拟合**。

从训练集中取出一部分数据用来调优，我们称之为**验证集（validation set）**。验证集其实就是作为假的测试集来调优。

把训练集分成训练集和验证集。使用验证集来对所有超参数调优。最后只在测试集上跑一次并报告结果。

当训练集数量较小时（因此验证集的数量更小），使用**交叉验证**的方法。

★ Nearest Neighbor分类器的优劣

实现简单，训练时间短，测试时间长。但在实际应用中，更关注测试效率。

CS231n课程笔记翻译：线性分类笔记（上）

CS231n 课程笔记翻译：线性分类笔记（上）

★ 线性分类

评分函数(score function)：原始图像数据到类别分值的映射。

损失函数(loss function)：用来量化预测分类标签的得分与真实标签之间一致性的。

⋆ 从图像到标签分值的参数化映射

线性分类器

$$f(x_i, W, b) = Wx_i + b$$

⋆ 理解线性分类器

将图像看做高维度的点

将线性分类器看做模板匹配

偏差和权重的合并技巧

$$f(x_i, W, b) = Wx_i + b \Longrightarrow f(x_i, W) = Wx_i$$

图像数据预处理：零均值的中心化

CS231n课程笔记翻译：线性分类笔记（中）

CS231n 课程笔记翻译：线性分类笔记（中）

⋆ 损失函数Loss function

DEF: 损失函数(Loss Function)

使用**损失函数(L oss Function)**(有时也叫**代价函数Cost Function**或**目标函数Objective**)来衡量我们对结果的不满意程度。直观地讲，当评分函数输出结果与真实结果之间差异越大，损失函数输出越大，反之越小。

⋆ 多类支持向量机损失Multiclass Support Vector Machine Loss

DEF: 多类支持向量机（SVM）损失函数

SVM的损失函数想要SVM在正确分类上的得分始终比不正确分类上的得分高出一个边界值Δ。

针对第i个数据的多类SVM的损失函数定义如下：

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta)$$

简而言之，SVM的损失函数想要正确分类类别y_i的分数比不正确类别分数高，而且至少要高Δ。如果不满足这点，就开始计算损失值。

对于线性评分函数($f(x_i, W) = Wx_i$)，将损失函数的公式改写如下：

$$L_i = \sum_{j \neq y_i} \max(0, w_j^T x_i - w_{y_i}^T x_i + \Delta)$$

其中w_j是权重W的第j行，被变形为列向量。