第三讲 损失函数和优化

2019年4月22日 12:50

1.线性分类II

1.1损失函数

1.1.1 multiclass SVM loss

上节课讲到了线性分类的算法，未讲到如何求解最优化的参数W，本节课讲解在求参数W的过程通过定义损失函数(lossfunction)来求解到最优化的解。

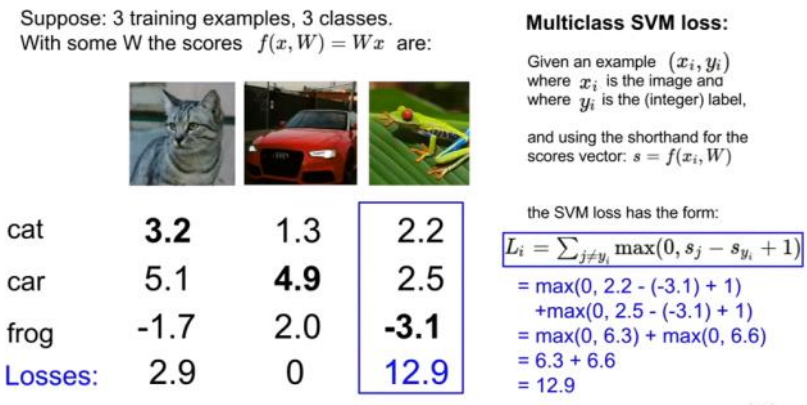


图3.1.1 多分类SVM的损失函数定义

计算方法：当前图片标签是cat，所以losses为

Loss1 = max((car - cat +1),0) + max((frog - cat + 1),0)

Loss2 = ...

Loss3 = ...

Loss = (Loss1+Loss2+Loss3)/3

tip：svm 初始化参数，在训练刚开始第一次迭代的时候各个分类的预测值是差不多相等的，所损失函数的值应该为类别数减去1，如果不是这样那么意味着你的代码可能有bug。

？我知道这样选择lossfunction好，但为什么好，好在哪里，可解释性？

Tip:损失量化，可计算，可优化

1.1.2 正则化

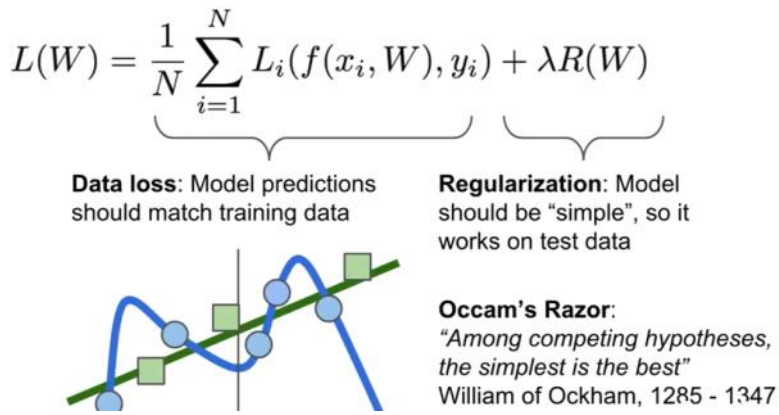


图3.1.2 欧卡姆剃刀与正则项

有时候训练一个模型不能强求拟合数据集，否则可能会出现不适用新数据的情况。（overfitting？）

正则化：

正则化策略主要的目的是限制学习算法的能力，根据惩罚项的不同可分为：L0范数惩罚、L1范数惩罚（参数稀疏性惩罚）、L2范数惩罚（权重衰减惩罚）。

正则化惩罚项比较：

http://www.imooc.com/article/23915?block_id=tuijian_wz

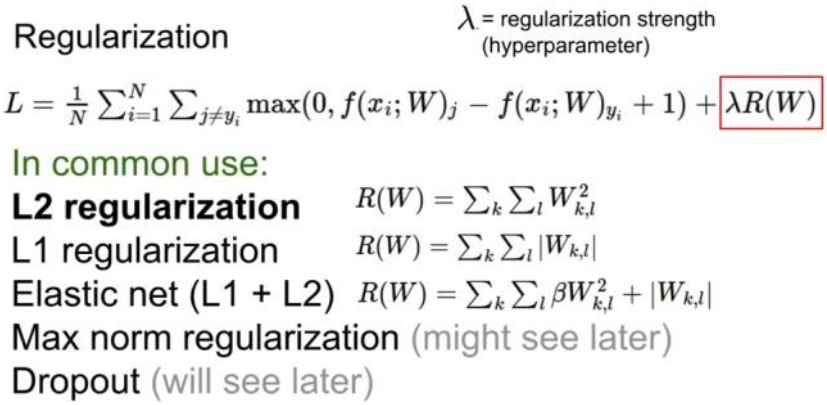


图3.1.3 常用正则项

1.1.3 softmax loss

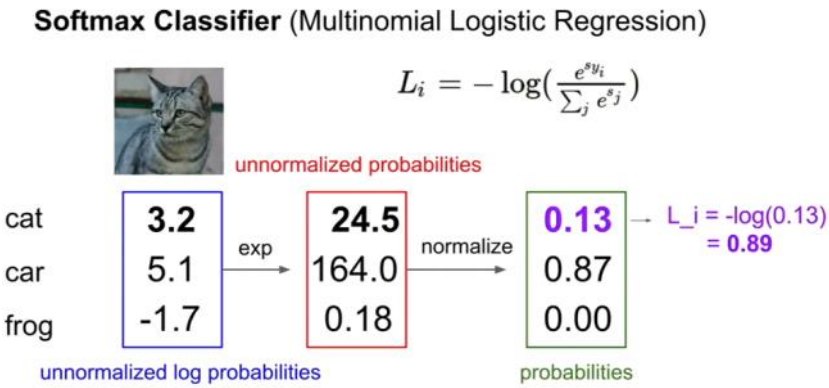


图3.1.4 逻辑斯蒂回归做分类

Softmax loss过程：假设y为图片的得分值(像素点✖参数矩阵＋偏移量)，首先取e的y次幂，然后做归一化，最后得到当前loss值为对归一化后的正确分类项上的得分取对数的负值。

从这可以看到，如果归一化之后的值越接近1(其他分类的得分和越接近0)，loss值越接近0，那么对于该分类任务剩下的就是如何优化减小loss值。

1.2 线性分类总结

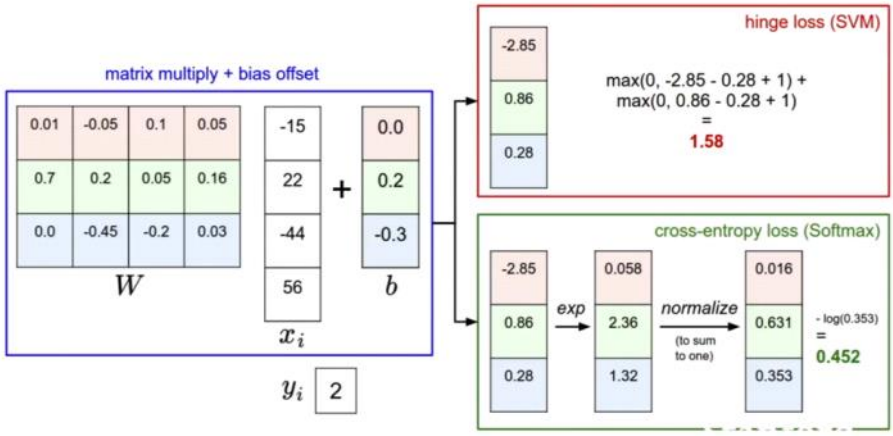


图3.1.5 线性分类整体过程

1.参数矩阵、偏移量赋初始值，并与图像的像素矩阵相乘得到每幅图像的得分。

2.将得分代入损失函数计算损失值，这个损失值表明了当前分类器坏的程度。

比较svm和softmax两个损失函数优缺点在哪，使用场景

一些个人观点：svm对分类正确的单个图不做过多处理，而softmax好像在无穷无尽的优化，正则化之后的正确分类得分几乎不可能达到1，这表明算法应该还有优化空间

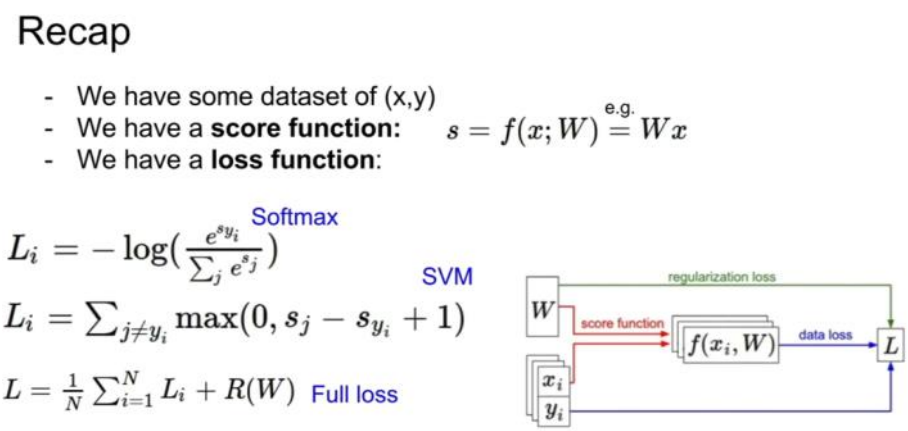


图3.1.6 线性分类小结

1.3优化问题

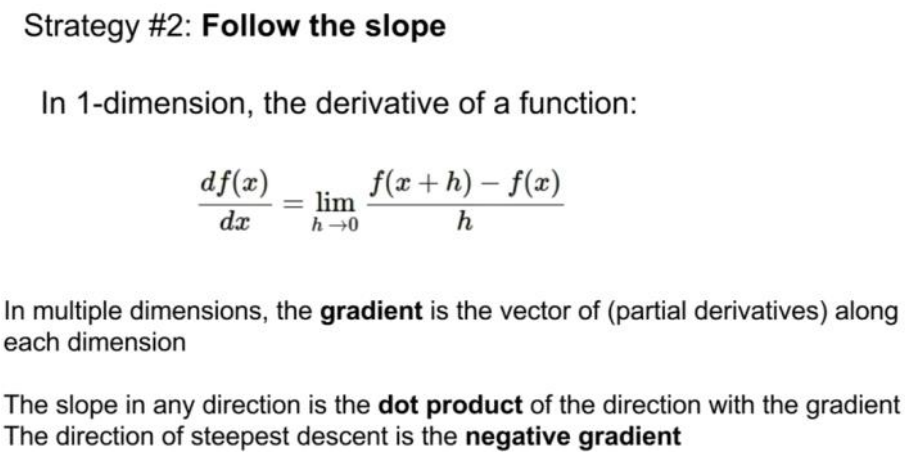


图3.1.7 梯度下降求最优解

损失函数自变量是参数矩阵，根据梯度来决定每次参数矩阵增加或减少一个值则会使函数值变大或者变小，增加一个值，迭代更替参数，直到损失函数值达到最优。

2.进阶模型表示与图像特征优化

2.1 图像特征

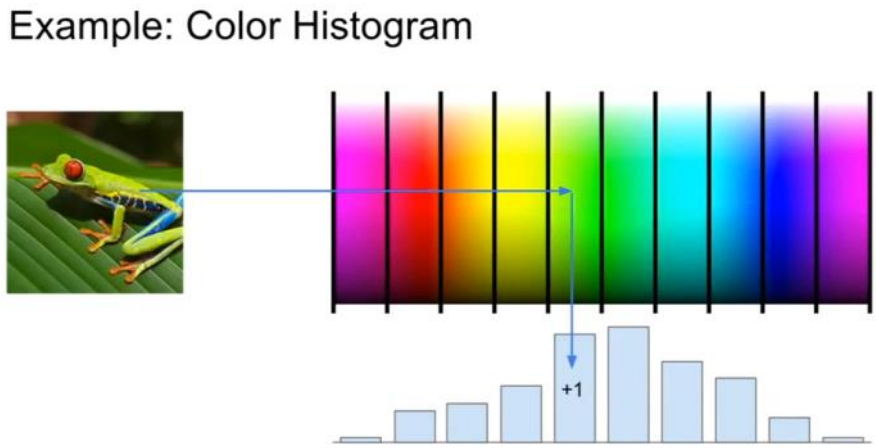


图3.2.1 颜色直方图

在之前使用线性分类器模型时，我们将图像的像素值作为输入，但这么做好像不太好，本节讲述了在将源图像数据输入模型之前先做一些图像特征提取之类的工程。比如这幅青蛙图，首相将连续的颜色空间离散化，将一张图片上的每个像素值对应的颜色做统计，符合每个色块就给每个色块值+1，最后将统计值作为数据输入模型。

特征工程?

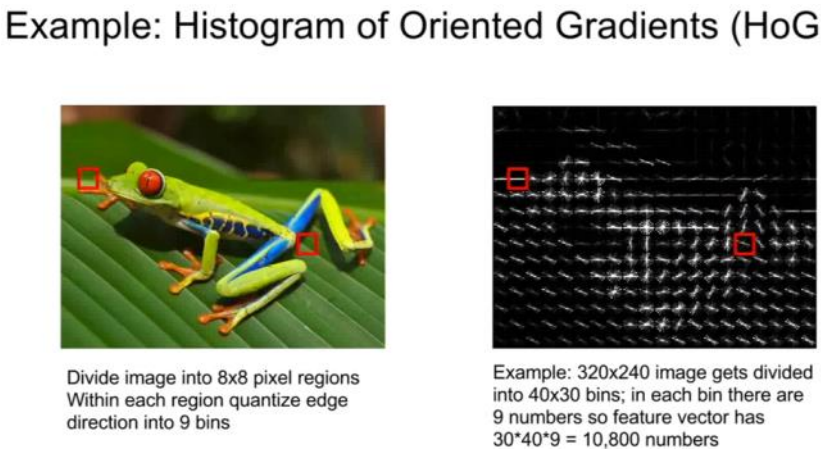
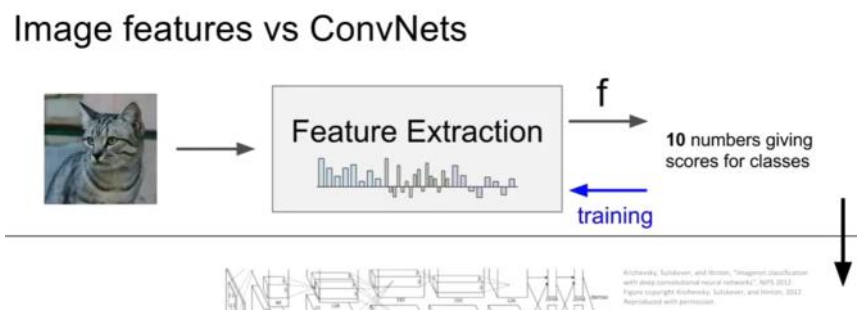


图3.2.2 方向梯度直方图

除了hog特征提取方法外常用的特征提取方法还有LBP特征、HAAR特征



Tips:
(Windows环境)
Python pip安装jupyter遇

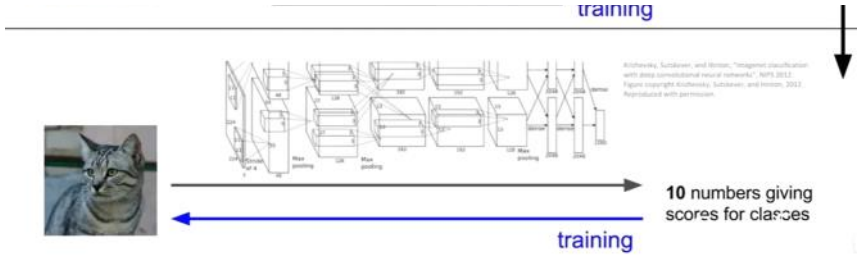


图3.2.3 特征提取与卷积神经网络

其实在卷积神经网络中也有图像特征提取，不过这部分由模型本身来完成。

在knn算法中一次加载5个batch加载数据集出现memery error。

(Windows环境)

Python pip安装jupyter遇到：Command "python setup.py egg_info" failed with error code 1 in ...

首先执行 `pip install --upgrade setuptools`命令再安装。