CS231n学习笔记

Assignment Git:

<https://github.com/CS231n-zju/CS231n>

视频地址：

<https://www.youtube.com/watch?v=vT1JzLTH4G4&index=2&list=PLe7764SJVnV10-Nr7e0sBlC9J0LRf4sQo>

课程作业：

<http://cs231n.github.io/>

Syllabus:

<http://cs231n.stanford.edu/syllabus.html>

Note翻译:

<http://www.52ml.net/17723.html>

**Week1**

[Video1. Lecture 1 | Introduction to Convolutional Neural Networks for Visual Recognition 2](#_Toc2112591153)

[Unknown sentences and words： 2](#_Toc1564220493)

[Note 4](#_Toc1185172199)

[Prehistory of computer vision 4](#_Toc1929631022)

[Image Classification 5](#_Toc379236954)

[表示 5](#_Toc1542101240)

[Challenges： 5](#_Toc1680257090)

[The image classification pipeline： 5](#_Toc670285452)

[Nearest Neighbor Classifier 5](#_Toc944323155)

[Example image classification dataset: CIFAR-10 5](#_Toc132581261)

[k - Nearest Neighbor Classifier 6](#_Toc725349747)

[Validation sets for Hyperparameter tuning（用于参数调整的验证集） 6](#_Toc968946720)

[N-fold cross-validation 7](#_Toc317607179)

[NN的优劣 8](#_Toc1380269737)

[Unknown words 8](#_Toc1898839932)

[Assignment1\_Q1: k-Nearest Neighbor classifier 9](#_Toc1904471151)

[Tricks: 9](#_Toc1765164493)

[Linear Classification 10](#_Toc1419090188)

[Parameterized mapping from images to label scores 10](#_Toc1607429092)

[线性分类器： 10](#_Toc1012825772)

[Interpreting a linear classifier 10](#_Toc140916453)

[Loss function 12](#_Toc892334248)

[Multiclass Support Vector Machine loss 12](#_Toc1951627281)

[Softmax classifier 13](#_Toc359135487)

[SVM和softmax的区别 14](#_Toc1221711092)

[Unknown words 14](#_Toc920084376)

[Video2. Lecture 2 | Image Classification 15](#_Toc1501935802)

[Unknown sentences and words： 15](#_Toc1481727494)

[Note: 16](#_Toc603552097)

[Q&A 16](#_Toc1934601116)

[Image Classification: 16](#_Toc196469659)

[Linear Classification: 16](#_Toc568659602)

# Video1. Lecture 1 | Introduction to Convolutional Neural Networks for Visual Recognition

## Unknown sentences and words：

The amount of visual data in our world has really exploded to a ridiculous degree.

There are just a crazy large, massive amount of visual data been produced out there in the world.

Roughly 80 percent of all traffic on the Internet would be video.

So it’s really critical that we develop algorithms that can utilize and understanding this data.

Sometimes we call visual data the dark matter(暗物质) of the internet in an analogy with（类比） dark matter in physics.

annotate 注释 vt.

catalog 登记成册 vt

interdisciplinary 跨学科 adj.

constellation 一群 n.

neuroscience 神经科学

cognitive science 认知科学

overlap 重叠部分

concurrently 同时地

encompassing 包含

seminar 研讨会

syllabus 教学大纲

tag team 两人一组

take over 接手

agenda 议程

chill 寒冷

onset 开始

proactive 积极主动

go after 追逐 predators went after prey 捕食者/被捕食者

intelligent animal 智慧动物

cortex 大脑皮层

manipulate 操纵

Renaissance period of time 文艺复兴期间

obscuration 昏暗

pinhole 小孔

in the mean time 同时

inspired 受到启发

electrode 电极

respond excitedly 兴奋地回应

primary visual cortex 初级视觉皮层

by and large 大体上说

thesis 论文

geometric 几何 algebra 代数

pay tribute to sb. 赞扬sb.

holistic 全面的

idealized 理想化的

intuitive 直观的 intuition 直觉

deconstruct 解构

seminal 意义深远的

cylinder 圆柱

generalized 广义的

elastic 可伸缩的

razor 剃须刀

frankly 坦白地

audacious 大胆地

ambitious 雄心勃勃的

object segmentation 对象分割

made some headway 有一些进展

gain momentum 见效

boosting 助推

There is a lot to admire this work.

computer chips 电脑芯片

near-real-time 近实时的

roll out 推出

occlusion 遮挡

viewpoint 视角

intrinsic 本质的

diagnostic 诊断的

pyramid 金字塔

clue 线索

landscape 风景画

highway 公路

compose 组成

histogram of gradient 梯度直方图

deformable 可变形的

all along 自始至终

It's also motivated by an observation that is rooted in machine learning

bottleneck 瓶颈

benchmarking 基准学习

gigantic 巨大的

stringent 严格的

on par with 达到相当的水平

hover （数值）徘徊在，原义为翱翔

deep dive 深入学习

have a deep dive into convolutional neural network

principle 原理

tremendous capacity 巨大的能力

academia 学术环境

calory 卡路里

image captioning 给图像加解说

tweaking 调整

collaborator 合作者

deploy 部署

coarse 粗糙的

transistor 晶体管

orders magnitude 数量级

crunching 运算

capacity 容量

fancy 设想

pop up 突然出现

tackle 解决

semantic 语义的

perceptional 知觉的

augmented reality 增强现实

intricacy 错综复杂的事物

exposure 暴露

holy grail 圣杯

exemplify 举例说明

conscious 有意识的

respectable 可敬的

compatriot 同胞

by no means 决不

hallucinate 使产生幻觉

psychedelic 迷幻的

render 给予

extenuating 情有可原的

circumstance 环境

prerequisite 先决条件

calculus 微积分

derivative 导数

linear algebra 线性代数

reintroduce 恢复

## Note

### Prehistory of computer vision

Around 540 millions years ago, the first animals developed eyes, the onset of vision started this exploded speciation phase.

2012 CNN首次在ImageNet中表现突出

# Image Classification

## 表示

一张图用三个维度表示：长度、宽度、通道 w\*h\*c个number

每个number ranges from 0 (black) to 255 (white)

## Challenges：

对于人类来说，识别图片是非常简单的，所以在计算机视觉算法的角度上这个挑战很有价值。以下是一些challenges：

* Viewpoint variation角度变化
* Scale variation尺度变化
* Deformation形变Many objects of interest are not rigid bodies（刚体） and can be deformed in extreme ways.
* Occlusion遮挡
* Illumination conditions光照条件
* Background clutter背景干扰The objects of interest may blend into（融入） their environment, making them hard to identify.
* Intra-class variation类内变化The classes of interest can often be relatively broad, such as chair. There are many different types of these objects, each with their own appearance.

## The image classification pipeline：

* Input: training set.
* Learning: training a classifier, or learning a model.
* Evaluation: In the end, we evaluate the quality of the classifier by asking it to predict labels for a new set of images that it has never seen before.

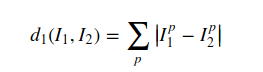
## Nearest Neighbor Classifier

### Example image classification dataset: CIFAR-10

60000张图片：10000张test，50000张train，大小为32\*32 pixel，共10类。

本质是，输入一张test图片，找出训练集里与它的像素差最低的一张训练图（x中的一张图），直接认为其所对应的y就是答案。

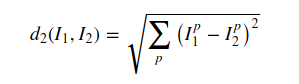
核心实现功能（使用L1距离）：



|  |
| --- |
| nn **=** NearestNeighbor() *# create a Nearest Neighbor classifier class*  nn**.**train(Xtr\_rows, Ytr) *# train the classifier on the training images and labels*  Yte\_predict **=** nn**.**predict(Xte\_rows) *# predict labels on the test images# and now print the classification accuracy, which is the average number# of examples that are correctly predicted (i.e. label matches)*  **print** 'accuracy: %f' **%** ( np**.**mean(Yte\_predict **==** Yte) |

|  |
| --- |
| import numpy as np  **class** **NearestNeighbor**(object):  **def** **\_\_init\_\_**(self):  **pass**  **def** **train**(self, X, y):  """ X is N x D where each row is an example. Y is 1-dimension of size N """  *# the nearest neighbor classifier simply remembers all the training data*  self**.**Xtr **=** X  self**.**ytr **=** y  **def** **predict**(self, X):  """ X is N x D where each row is an example we wish to predict label for """  num\_test **=** X**.**shape[0]  *# lets make sure that the output type matches the input type*  Ypred **=** np**.**zeros(num\_test, dtype **=** self**.**ytr**.**dtype)  *# loop over all test rows*  **for** i **in** xrange(num\_test):  *# find the nearest training image to the i'th test image*  *# using the L1 distance (sum of absolute value differences)*  distances **=** np**.**sum(np**.**abs(self**.**Xtr **-** X[i,:]), axis **=** 1)  min\_index **=** np**.**argmin(distances) *# get the index with smallest distance*  Ypred[i] **=** self**.**ytr[min\_index] *# predict the label of the nearest example*  **return** Ypred |

L2距离（欧氏距离）：



L2得到的结果（35.4%）比L1（ 38.6% ）低一些

The L2 distance prefers many medium disagreements to one big one.

## k - Nearest Neighbor Classifier

本质是查找training data中最相近k张图（NN中为1张图，即k=1时），然后对这k个答案进行投票。

Higher values of **k** have a smoothing effect that makes the classifier more resistant to outliers.

K越高，分类越平滑，因此可以更好地抵抗离群点。

### Validation sets for Hyperparameter tuning（用于参数调整的验证集）

K的值应该如何确定？

不要用测试集去调整参数，容易使得你的模型过拟合。

将training data中的一小部分分割成validation data，这里将5万张train中1千张作为验证集，来寻找最优的k（accuracy最高的k）。

|  |
| --- |
| *# assume we have Xtr\_rows, Ytr, Xte\_rows, Yte as before*  *# recall Xtr\_rows is 50,000 x 3072 matrix*  Xval\_rows **=** Xtr\_rows[:1000, :]  *# take first 1000 for validation*  Yval **=** Ytr[:1000]  Xtr\_rows **=** Xtr\_rows[1000:, :]  *# keep last 49,000 for train*  Ytr **=** Ytr[1000:]  *# find hyperparameters that work best on the validation set*  validation\_accuracies **=** []  **for** k **in** [1, 3, 5, 10, 20, 50, 100]:    *# use a particular value of k and evaluation on validation data*  nn **=** NearestNeighbor()  nn**.**train(Xtr\_rows, Ytr)  *# here we assume a modified NearestNeighbor class that can take a k as input*  Yval\_predict **=** nn**.**predict(Xval\_rows, k **=** k)  acc **=** np**.**mean(Yval\_predict **==** Yval)  **print** 'accuracy: %f' **%** (acc,)  *# keep track of what works on the validation set*  validation\_accuracies**.**append((k, acc)) |

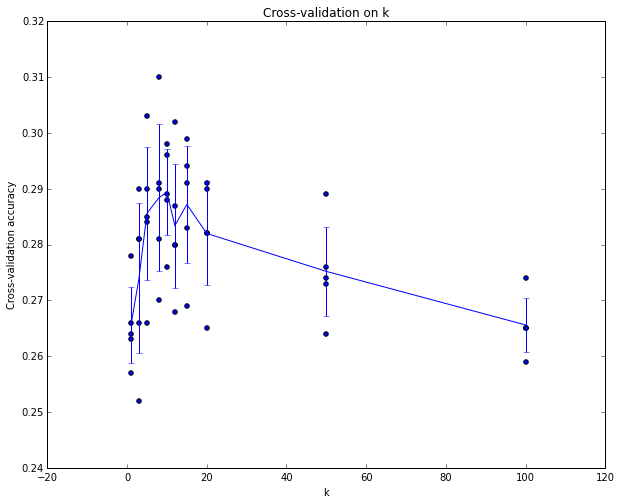
### N-fold cross-validation

N折交叉验证法

**数据集较小的情况下**。可以把training data划分成n份，依次用其中1份做验证集，那么对于每次对某个参数的实验，均可以获得n个accuracy。

以k值选择为例，取均值后，可以做出k与accuracy的折线图，图中发现k取7左右时accuracy最高（虽然我们并没有做k=7的实验）。

***N取值越高，下图中的折线会更加平滑，有利于我们选择出最优的k来。***



不使用交叉验证法的原因：计算代价较高，要计算n次。

**如果要调整的参数较多**，更倾向于50%-90% of the training data for training and rest for validation.

但参数较少时，交叉验证法更加保险。

一般取3-fold, 5-fold or 10-fold cross-validation.

## NN的优劣

训练时间非常少，但是测试时间非常久。

In pracitice, 我们希望反过来，而深度神经网络是符合的。

有一些NN的优化算法，如ANN等，是在训练阶段将图片索引化（kdtree或者用k-means先聚类一下）

低维度下NN是一个很好的选择，但是对于图像分类并不适用。

NN并不能体现图像之间的语义差别，更多的是图像的背景，色彩的差异。

## Unknown words

presumably 大概;可能 adv.

It presumably due to the strong black background.

subtracted 减掉的，除去的 v.

elementwise 分别作用

stretched  拉伸

euclidean distance 欧氏距离

monotonic function 单调函数

preserve 保持

metric 指标 matrices 矩阵的复数

unforgiving  无情的

Induce 诱导

dot products 点积

tweaking hyperparameters 调整参数

deploy 部署

proxy 代理

**generalization** of your classifier  分类器的泛化能力

arbitrarily 任意地

error bars 误差线

Standard deviation 标准差

in favor of  有利于

desirable 可取的

trade off 权衡

counter-intuitive 与正常预期相反的

perceptual 感知的

Inadequate 不足的

embed 嵌入

embark 开始着手做

As a rule of thumb 作为经验法则

# Assignment1\_Q1: k-Nearest Neighbor classifier

Implement NN with L1 distance and get the accuracy of 0.385900,

use 7503.271591 s.

K=1, Got 137 / 500 correct => accuracy: 0.274000

K=2, Got 139 / 500 correct => accuracy: 0.278000

## Tricks:

1.sum(a,axis=0/1)

我们平时用的sum应该是默认的axis=0 就是按照列相加

而当加入axis=1以后就是将一个矩阵的每一行向量相加

1. no loop实现L2距离

将完全平方展开，利用矩阵乘法加速。

Sqrt(sum ( (X1-x2)^2) )

X1^2+ x2 ^2- 2 \* x1 \* x2

X 为test

self.X\_train为train

|  |
| --- |
| M = np.dot(X, self.X\_train.T) te = np.square(X).sum(axis=1) tr = np.square(self.X\_train).sum(axis=1) dists = np.sqrt(-2 \* M + tr + np.matrix(te).T) |

print(M.shape,te.shape,tr.shape,dists.shape)

(500, 5000) (500,) (5000,) (500, 5000)

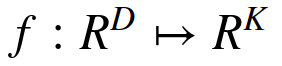
# Linear Classification

We will then cast this as an optimization problem in which we will minimize the loss function with respect to the parameters of the score function.

我们将其转化为一个优化问题，目的是通过优化评分函数的参数最小化损失函数。

## Parameterized mapping from images to label scores

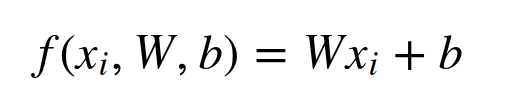
Score function maps the raw image pixels to class scores.



D是图像pixel维度，K是class的数量。

### 线性分类器：

一个最简单的score function：



Xi即输入图像， has all of its pixels flattened out to a single column vector of shape [D x 1]

参数包含：The matrix **W** (of size [K x D]), and the vector **b** (of size [K x 1])

注：b被称作 **bias vector** ，因为它虽然影响评分，但是 without interacting with the actual data .

Advantages：

一旦训练完成，training data可以被discarded，只保留参数即可。

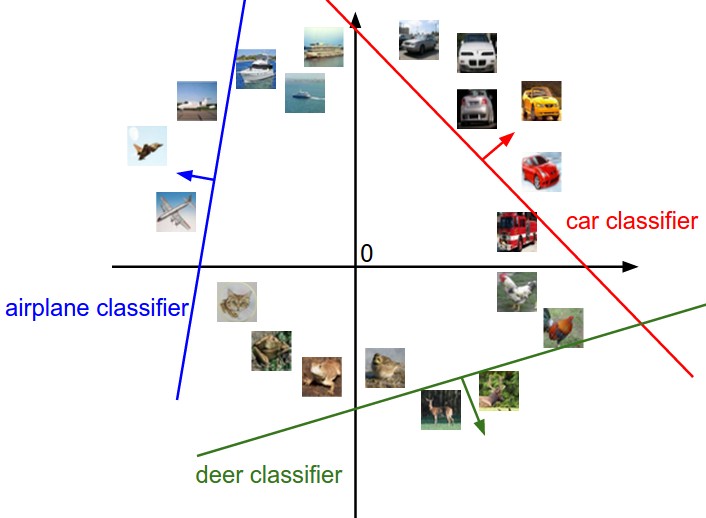
Classifying the test image involves a single matrix multiplication and addition，因此速度比和整个training data作比较快很多（相对于KNN）。

### Interpreting a linear classifier

Weight本质是，对于某一类，比如船，分类器对图像上每个pixel每个通道颜色的like和dislike。对于船，蓝色的pixel得到的score更高（可能与海洋有关）。

**Analogy of images as high-dimensional points**

每个图片可以被认为是一个高维空间中的点。e.g. each image in CIFAR-10 is a point in 3072-dimensional space of 32x32x3 pixels



注：参数的意义

权重w：旋转line

偏置向量b：平移line，如果没有b，xi=0的点将会永远待在零点。

**Interpretation of linear classifiers as template matching.**

W中每一行实际就是一个class的本质体现，每次classfy就是将图像与w的每一行作内积。

和KNN类比：

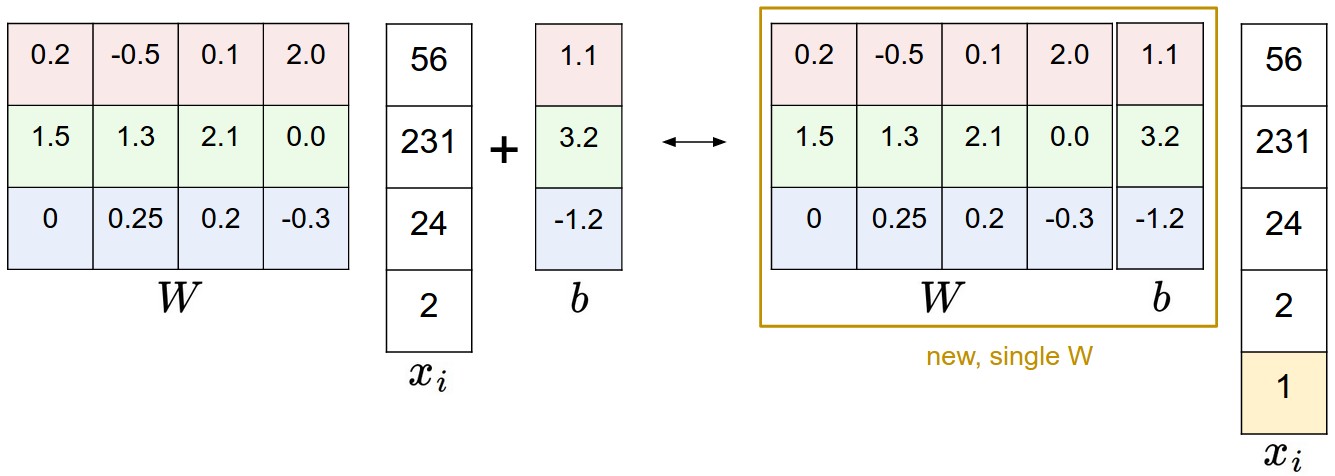
|  |
| --- |
| 我们实际上还是在做KNN，但是并不是把每张test image和traing data中的每一张做匹配，找到最相近的一张图，其label作为结果。  IMG_256  而是我们利用training data训练出10个分类的10张template图，我们把每张test image和这10个template作匹配，找到最相近的template，其label作为结果。  我们用内积代替了L1和L2 |

颜色的分布，也暗示了training data中红色的cat比较多，比如car的template中红色范围更多。

**Bias **trick****

在xi末尾添加一个常数1，把矩阵简化：

这样score function从2017-10-25 11-37-53屏幕截图变成了2017-10-25 11-37-57屏幕截图



**Image data preprocessing**

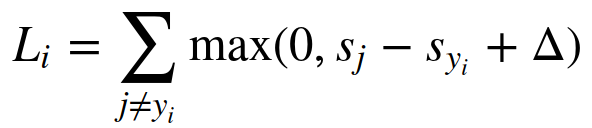
归一化数据，集中数据，去除平均值？

## Loss function

衡量prediction和ground truth的差别。Loss越高，分类器越差。

### Multiclass Support Vector Machine loss

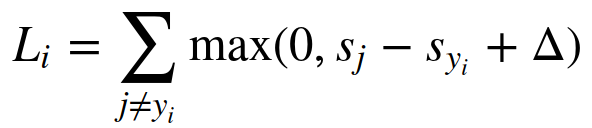
The Multiclass SVM loss for the i-th example is then formalized as follows:



对于第i个例子，对于它被分为3个类的score分别为s1,s2,s3,而ground truth是其应该被放进类1,所以计算s2/s3和s1的差值并加上梯度delta.一些比s1高的分数没有多于delta的,将会取0,而其余则取他们的差值.

In summary, the SVM loss function wants the score of the correct class yi to be larger than the incorrect class scores by at least by Δ (delta).如果不是，则这一项变为0.

鉴于前面对s的取值公式,则L可被写作:



这种max loss,即max(0,−)常被称作hinge loss,有时用他的平方,即max(0,−)2

作为loss,使得惩罚更强烈(二次方而不是线性)!

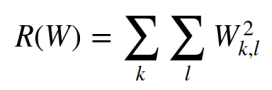
选择平方版本还是标准版本可以通过cross-validation来决定.

If any class has a score inside the red region (or higher), then there will be accumulated loss. Otherwise the loss will be zero.

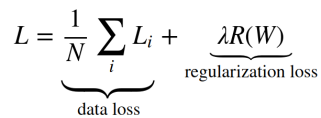


**Regularization**

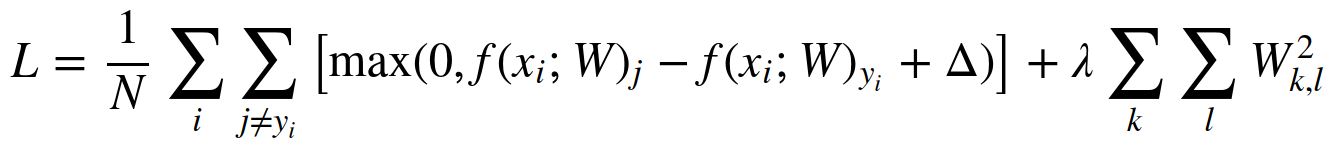
如果有这么一组权重,可以使得所有的example的loss都是0,那么可以得到无数种这样的W,只要简单地乘以一个数就可以.为了避免这种现象,在loss中添加了一个penalty:



这样可以得到最终的loss,其中左边是对于所有examples,Loss的均值.



展开得到(这里N是examples的个数,λ由cross-validation决定)



左边i是每个example,j是每个example里除了ground Truth对应的label外,其他label的hinge loss.

右边k是每个类(例如k=10),l是image flatted之后的维度 (例如l=3072)

λ是正则化的强度.

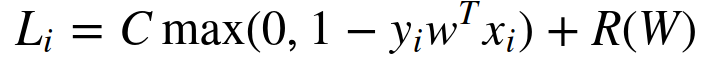
Penalty的一些优势: 趋向于选择更小更弥散的weights,可以less overfitting.同时,因为penalty的存在,we can never achieve loss of exactly 0.0 on all examples(只有w全部是0的时候才会出现).

|  |
| --- |
| **All we have to do now is to come up with a way to find the weights that minimize the loss.** |

**Practical Considerations**

需要注意到,delta的设置并不会很大的影响loss,真正影响的是w的大小,他会直接影响score的大小,以及不同classes的scores之间的差异.delta可以被safely设置成1.

Binary Support Vector Machine的损失函数定义如下,这里C和λ都控制着同样的权衡:data loss和regularization loss,并且这里C∝1/λ.

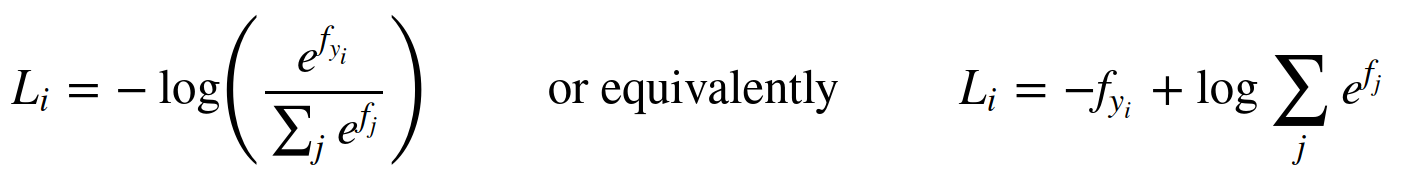


备注部分:在初始行驶中进行最优化,损失函数通常不可微分,但是可以使用次梯度;其他版本的多元SVM:OVA,AVA等等. 使用文中的版本you can construct multiclass datasets where this version can achieve zero data loss, but OVA cannot.

### Softmax classifier

Softmax分类器是二元逻辑斯特回归的多元一般形式.

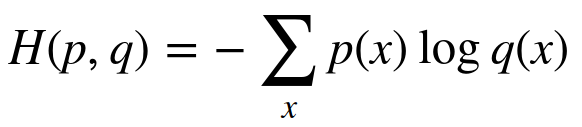
它的loss function与SVM的不同在于,修改了hinge loss部分,变成了cross-entropy loss.



对于每一个example Li, j是分类结果class中的一个,fyi是把i分类为yi的score,这样可以将Li控制在[0,1]之间

**信息论知识**

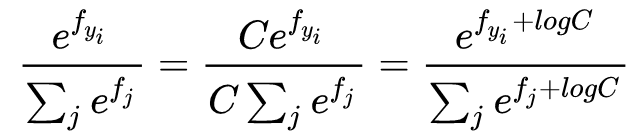
真实分布p和预测分布q的交叉熵被定义为:



我们希望,真实分布和预测分布之间的相对熵最小,即差异最小.也就是希望所有的预测概率分布能和真实概率分布一致.

**Practical issues: Numeric stability**

由于指数级别的数值非常大,除以一个这样的数会导致数值不稳定性.用上下同时乘以C的方式增强数值稳定性.通常取:log*C*=−max*jfj*

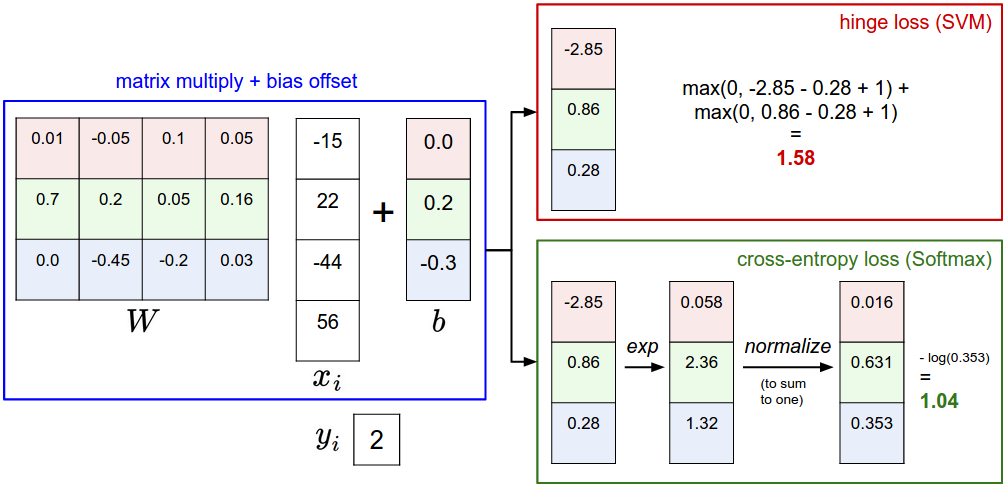


这样使得新的指数必然<=0,这样除号上下的数都必然小于1,不会造成指数爆炸问题.

**名字的误解**

SVM使用hinge loss,而softmax使用交叉熵loss,softmax本质指的是一种函数,他们将数值归一化,因此并没有softmax loss这一说法.

### SVM和softmax的区别



上图以一张图xi为例,由于我们的目标是降低loss(最好是0).

我们可以发现,SVM鼓励的是让ground truth的score比其他类别的score高出一个margin,越高越好,越高,loss越低.

而softmax鼓励的是让-log(正确分类的概率)越低越好,也就是正确分类的概率越高越好.

Softmax也提供了每个分类结果的概率.但是这个概率是加引号的!因为随着λ的惩罚增强,w会降低,因此”概率”会弥散.

对于softmax来说,data loss是永远可以被降低的,但是不会达到0,但是SVM只要达到边界,就认为可以了.

## Unknown words

Inefficient 效率

Cast 铸造，计算

Optimization 优化

concrete example 具体实例

Arguably 可以说

Foreshadowing 伏笔

For the sake of 为了

Monochrome 单色

squashing 挤压

Prototype 原型

with the terminology…… 根据……理论

Cumbersome 笨重的

normalization  归一化

Subtracting 减去

Scale 天平，测量，规范

Anthropomorphise 拟人化

in the sense 在这个意义上

Yield 生产，产量

Abbreviate 缩写

Clamped 夹紧

If this is not the case 如果不是这样的话

Desired 渴望的

Terminology 术语

hinge loss 铰链损失

Quadratically 二次的

quantifies  量化

constraint  约束

Uniformly 一致地

Magnitudes 量

regularization penalty  正则化

Generalization 一般化

Diffuse 弥漫的

negligible 可以忽略不计的

pathological  病态的,不理智的

brush over 刷过(温习一遍)

Tradeoff 权衡

Arbitrarily 任意的

Reciprocal 倒数

Unconstrained 无约束的

Kink 扭结,不可导点

Differentiable 可微分的

Subgradient 次梯度

Scope 范围

Uncalibrated 未校准

Squash 挤入

Distribution 分布

Peaky 憔悴的

Scenarios 情节,情况

# Video2. Lecture 2 | Image Classification

## Unknown sentences and words：

Mechanic 操作n.

Attendance 出席率n.

get lost in the shuffle 在搁置中被遗忘

Antsy 坐立不安的

Cosmetic 装饰的

Seamless 无缝隙的

Coupons 优惠券

Concretely 具体地

Hardwired 天生的

Grid 网格

Distill 蒸馏/提取

Subtle 微妙的

Twitch 抽搐

Muscle 肌肉

Robust 稳健的

Moody 多变的

tail peeking out 尾巴露了出来

couch cusion 沙发垫

Foreground 前景

Simultaneously 同时

Miraculous 神奇的

Incredible 难以置信的

spit out 吐出

convex hull 凸包

Encryption 加密

Enumerate 列举

Explicit 明确的

high-end 高端

Brittle 脆弱的

Scalable 可以升级的

Ingest 摄取

Concise 简洁的

Spurious 污点

Fancier 奇特

cycle through all the different folds

underlying statistical assumption 基本的统计假设

Alleviate 减轻

Perceptual 感性

curse of dimensionality 维度灾难

Non-trival 非平凡

parametric model 参数化模型

scaling offset 缩放偏移

Unravel 解开

blurry windshield 模糊的挡风玻璃

Multimodal 多模态

## Note:

L1距离更多的依赖于坐标系,L2则并不是的

对于每一列的属性有具体含义时,L1是更好的选择

设置参数的时候:不要选择最适合training data的(例如当我们选择k=1的时候,可能最适合training data,但是很明显,k越大,对于test data的分类会更平滑),也不要选择最适合test data的(过拟合test data将使得model不再robust)

# Q&A

## Image Classification:

The L2 distance prefers many medium disagreements to one big one.

## Linear Classification:

**Image data preprocessing.为什么要去掉mean image和归一化?**

Video2里面30分钟:

L1距离更多的依赖于坐标系,L2则并不是的

对于每一列的属性有具体含义时,L1更好

设置参数的时候:不要选择最适合training data的?

40分钟:维度灾难问题