**深度学习知识点**

一 大华

1.1奥卡姆剃刀原则及什么时候是必要的

应用到机器学习任务中，可以通过减小模型的复杂度来降低过拟合的风险，即模型在能够较好拟合训练集（经验风险）的前提下，尽量减小模型的复杂度（结构风险）。

1.2如何解决过拟合问题

* 深度学习

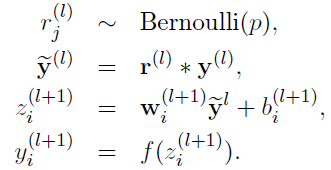
一 Early stopping：在训练的过程中，记录到目前为止最好的validation accuracy，当连续10次Epoch（或者更多次）没达到最佳accuracy时，则可以认为accuracy不再提高了。此时便可以停止迭代了（Early Stopping）

二 数据集扩增：我们就可以通过图像平移、翻转、缩放、切割将数据库成倍扩充

三 正则化：L1 L2

四 参数共享 卷积层就是其中权值共享的方式,一个卷积核通过在图像上滑动从而实现共享参数,大幅度减少参数的个数,用卷积的形式是合理的,因为对于一副猫的图片来说,右移一个像素同样还是猫,其具有局部的特征.这是一种很好的缓解过拟合现象的方法

五 dropout **通过函数产生0~1的概率**，如果值为0.4 则去除40%。前向传播和反向只对保留的激活函数进行。**每一轮的dropout不一样**。**预测时x要乘上p**。



六 Inception系列的思想，辅助节点，用较小的权重加入到分类结果

七 BN 归一化

均值方差只基于当前的minibatch，不基于全部训练数据。

服从标准正态分布，不是均值为0，方差为1的高斯分布

对于输入维度为  [N, H, W, C]  的 Tensor, Batch Normalization 在哪个(些)维度上做归一化（N,C）

1.3 梯度爆炸的解决方法

**激活函数** 比如使用sigmoid，梯度消失就会很明显了，其梯度是不可能超过0.25的。Tanh也不超过1。改用ReLU

**预训练加微调** 其基本思想是每次训练一层隐节点，训练时将上一层隐节点的输出作为输入，而本层隐节点的输出作为下一层隐节点的输入，此过程就是逐层“预训练”（pre-training）；在预训练完成后，再对整个网络进行“微调”（fine-tunning）

**梯度剪切、正则** 设定阈值和正则化抑制

**BN**

**Resnet的残差结构**，保证梯度不会消失

**LSTM**的结构

1.4 目标检测的损失函数

1.5 Adaboost在人脸检测上应用特别好，知道原因么

先选择不同大小的矩形框去框出图片的一些特征，然后利用积分运算计算出像素点的特征值

1.6训练和测试阶段不一样的操作有哪些

在训练阶段，对应用了dropout的层，每个神经元以keep\_prob的概率保留（或以1-keep\_prob的概率关闭），然后在测试阶段，不执行dropout，也就是所有神经元都不关闭，但是对训练阶段应用了dropout的层上的神经元，其输出激活值要乘以keep\_prob。

AlexNet论文发表于2012年，而现在主流的方法是Inverted dropout，和传统的dropout方法有两点不同：

在训练阶段，对执行了dropout操作的层，其输出激活值要除以keep\_prob

测试阶段则不执行任何操作，既不执行dropout，也不用对神经元的输出乘keep\_prob。

二 云从科技

2.1数据不平衡会产生什么问题，如何解决数据不平衡问题

2.2 模型压缩

裁剪：

* 反向传播有些参数不怎么更新，通过设置top-k排除一些权值。
* 统计filter的权重之和或者大于0的个数，排除一些不重要的filter
* 在剔除不重要的参数之后，通过一个retrain的过程来恢复模型的性能，这样就可以在保证模型性能的情况下，最大程度的压缩模型参数及运算量。

核稀疏化：

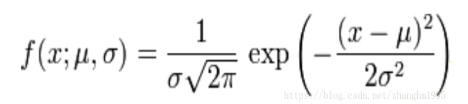
在训练过程中，对权重的更新进行诱导，使其更加稀疏，对于稀疏矩阵，可以使用更加紧致的存储方式，如CSC，但是使用稀疏矩阵操作在硬件平台上运算效率不高，容易受到带宽的影响，因此加速并不明显。

模型设计：设计更加精简的网络拓扑结构

2.3 focal loss解决什么问题

陌陌

3.1高斯函数的公式

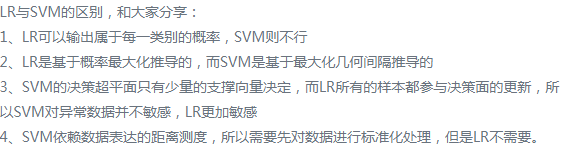


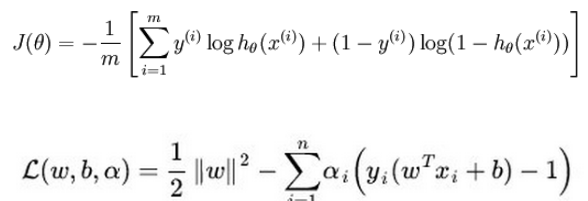
3.2直方图均衡化

直方图均衡化通过调整图像的直方图来增强图像的对比度，经常使用在医学图像分析中

计算PDF（概率密度），利用均衡化公式重新赋值像素值

3.3 SVM与LR区别

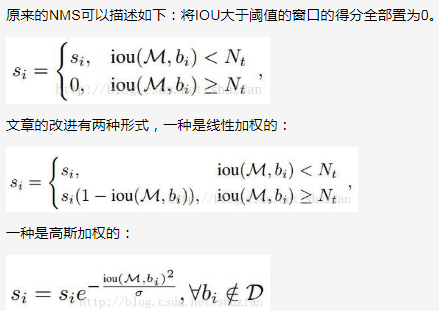




腾讯

4.1 softnms原理，比nms好吗

用了线性加权和高斯加权

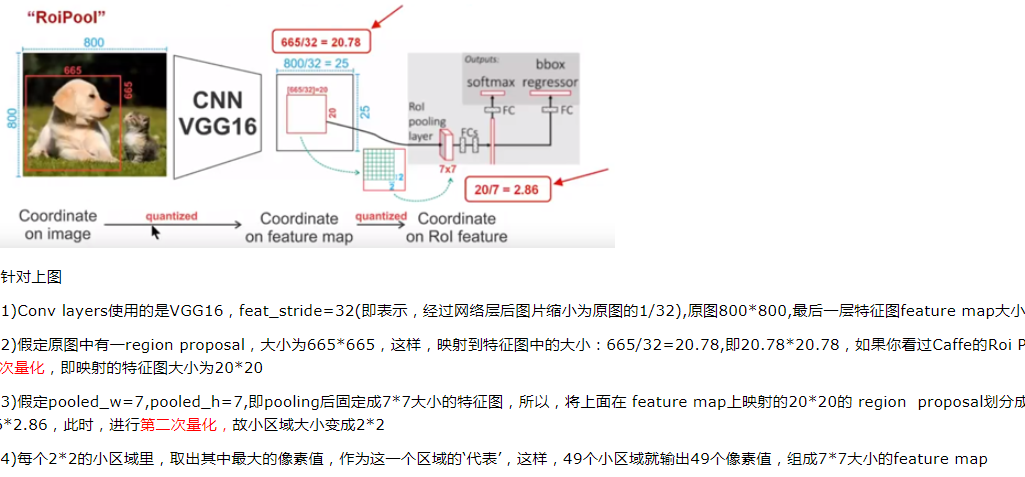


M为当前得分最高框， 为待处理框， 和M的IOU越大，得分就下降的越厉害。

4.2 roi align和roi pooling

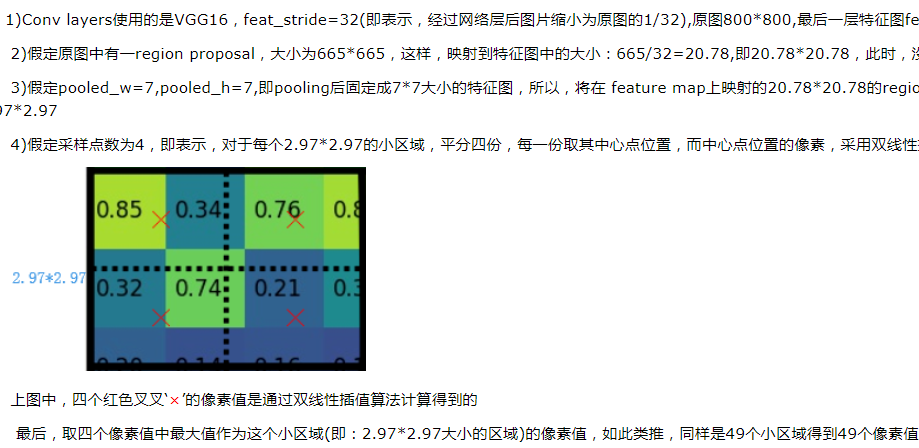
对于检测图片中大目标物体时，两种方案的差别不大，而如果是图片中有较多小目标物体需要检测，则优先选择RoiAlign，更精准些

<https://www.cnblogs.com/wangyong/p/8523814.html>



roi pooling经过两次量化，即将浮点数取整，原本在特征图上映射的20\*20大小的region proposal，偏差成大小为14\*14的，这样的像素偏差势必会对后层的回归定位产生影响。第四步比较重要。

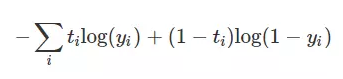
ROIAlign



海康

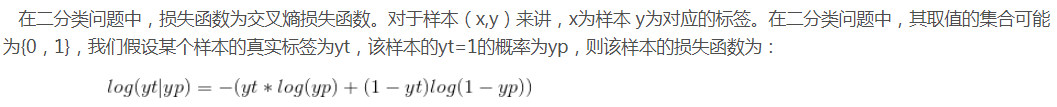
1语义分割损失函数

分类交叉熵caterical\_crossentropy

 以sigmoid作为输出

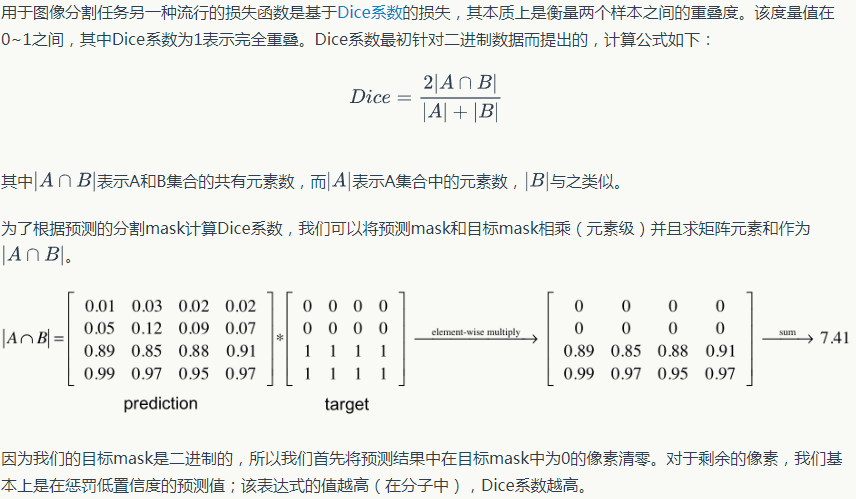
以softmax输出为-ti\*log(yi)

二分类





Dice loss



新浪

二叉树 dfs bfs 队列和栈 （四种方法）

排序算法

深度学习基础知识总结

1. 样本不均衡

正负样本不均衡（正负样本比例达到1:1000）和难易样本不均衡（简单样本主导loss）．一般在目标检测任务框架中，保持正负样本的比例为1:3（经验值）

数据扩充。这是最简单的解决办法，通过数据扩充是检测的若干类物体达到均衡。

种类惩罚。有时候，数据扩充并不能完全解决所有的问题。例如在一张图片中，树和草往往同时出现，这时无论怎么进行数据扩充，草这一类的数量还是远多于树的。这时可以采取种类惩罚的方法，即在loss计算的时候，人为提高数量较少类的loss的权重，从而提高这一类对损失的贡献，从而实现数据均衡的目的

R-CNN类检测器可以通过two-stage级联和启发式采用来解决class imbalance问题，proposal stage包括：Selective Search、EdgeBoxes、DeepMask、RPN，可以迅速候选区域数目降低过滤大量background样本；在第二个分类stage启发式采样，可以采取的策略有：固定前景和背景的比例1:3或者采用OHEM在线困难样本挖掘，可以用来维持前景和背景样本可操作性平衡

one stage检测器需要处理更大的候选位置集，虽然也应用了同样的启发式采样，但是效率低下因为在训练过程中很容易受到简单背景样本的支配；这种低效率问题是目标检测的典型问题，对此典型的解决方法是bootstrapping、HEM

focal loss是一个能够动态缩放的cross entropy loss，当正确类别的置信度提高时缩放因子衰减为0，缩放因子可以自动降低easy例子在训练期间贡献loss的权重，使得模型注重hard例子

Bootstrapping 再抽样训练，可以将每一级分类错误的样本继续添加进下一层进行训练

hard negative mining

困难负样本是指哪些容易被网络预测为正样本的proposal，即假阳性(false positive)，如roi里有二分之一个目标时，虽然它仍是负样本，却容易被判断为正样本，这块roi即为hard negative，训练hard negative对提升网络的分类性能具有极大帮助，因为它相当于一个错题集

如何判断它为困难负样本呢？也很简单，我们先用初始样本集(即第一帧随机选择的正负样本)去训练网络，再用训练好的网络去预测负样本集中剩余的负样本，选择其中得分最高，即最容易被判断为正样本的负样本为困难样本，加入负样本集中，重新训练网络，循环往复，然后我们会发现:咦！我们的网络的分类性能越来越强了！假阳性负样本与正样本间也越来越相似了！(因为此时只有这些妖怪区域能迷惑我们的分类器了)。

OHEM（online hard example miniing）

对训练过程中的困难样本自动选择，其核心思想是根据输入样本的损失进行筛选，筛选出困难样本（即对分类和检测影响较大的样本），然后将筛选得到的这些样本应用在随机梯度下降中训练。在实际操作中是将原来的一个ROI Network扩充为两个ROI Network，这两个ROI Network共享参数。其中前面一个ROI Network只有前向操作，主要用于计算损失；后面一个ROI Network包括前向和后向操作，以hard example作为输入，计算损失并回传梯度。该算法在目标检测框架中被大量使用，如Fast RCNN。在计算损失时，先通过对loss排序，选出loss最大的example进行训练，这就是保证金仅仅学习hard examples。但是这个方法存在一个不足，就是把所有的easy example都去除掉了，对于easy positive example它也给过滤了

Focal Loss（Retinanet）

在标准交叉熵损失基础上修改得到的。这个损失函数可以通过减少易分类样本的权重，使得模型在训练时更专注于难分类的样本



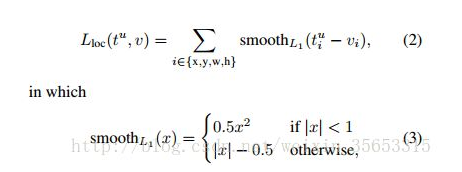
paper中单独做了一个实验，就是直接在CE上加权重，得到的结果是alpha=0.75的时候效果最好，也就是说，正样本的权重为0.75，负样本的权重为0.25，正样本的权重大于负样本，因为本身就是正样本个数远少于负样本。加了gama次方后，alpha取0.25的时候效果最好，也就是说，正样本的权重为0.25，负样本的权重为0.75，这个时候反而负样本的权重在增加，按道理来说，负样本个数这么多，应该占loss主导，这说明gama次方已经把负样本整体的loss衰减到需要加权重的地步。paper中alpha取0.25，gama取2效果最好。

FasterRCNN

对RPN输出的区域推荐进行过滤，过滤出高分数的推荐框（1k-2k个），这个过程将大量easy negative过滤掉，这解决了起那么第二个问题。

然后，还对正负例的比例进行调整，用IOU进行pos与neg的比例，如设置为1：3。这样就防止了negative过多的情况(同时防止了easy negative和hard negative)。解决了前面第一个问题，减小negative对loss的过于贡献。

目标检测为什么使用SmoothL1损失

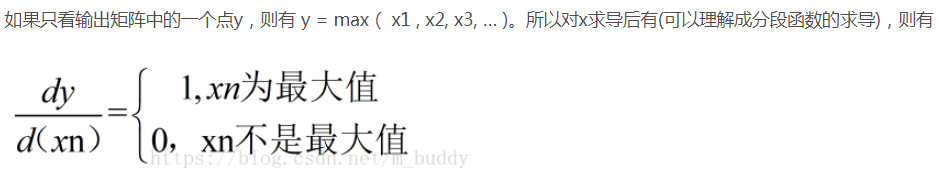


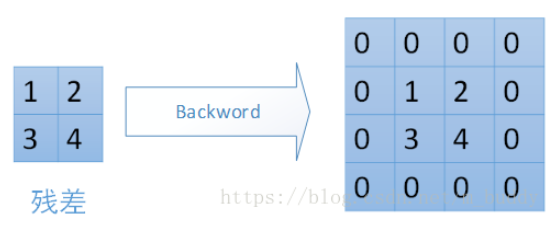
从上面的导数可以看出，L2 Loss的梯度包含 (f(x) - Y)，当预测值 f(x) 与目标值 Y 相差很大时，容易产生梯度爆炸，而L1 Loss的梯度为常数，通过使用Smooth L1 Loss，在预测值与目标值相差较大时，由L2 Loss转为L1 Loss可以防止梯度爆炸。

如上面的公式所示，smooth L1 loss在 |x| >1的部分采用了 L1 loss，当预测值和目标值差值很大时, 原先L2梯度里的 (f(x) - Y) 被替换成了 ±1,，这样就避免了梯度爆炸, 也就是它更加健壮。其实就是在预测值和真实值的差距小于1时，采用L2 loss，大于1时采用L1 loss。减0.5是为了让loss连续，当x为1时两段都是0.5。

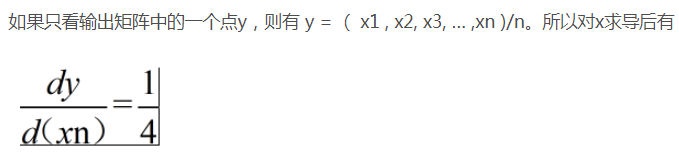
池化层反向传播

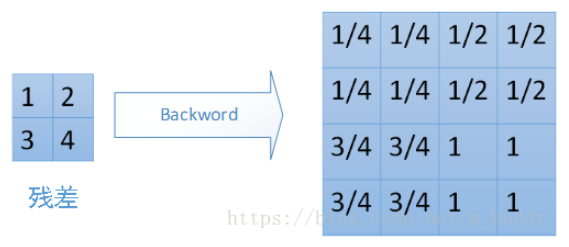
1最大池化





2平均池化





YOLO系列总结

评价指标

NMS

struct boxes

{

int x;

int y;

int w;

int h;

float score;

};

float iou(boxes a1, boxes a2)

{

int left = max(a1.x-a1.w, a2.x-a2.w);

int right = max(a1.y-a1.h, a2.y-a2.h);

int up = min(a1.x+a1.w, a2.x+a2.w);

int down = min(a1.y+a1.h, a2.y+a2.h);

float res = (right - left) \* (up - down);

float iou = res / (4\*a1.w \* a1.h + 4\*a2.w \* a2.h - res);

return iou;

}

void nms(vector<boxes>& array, vector<boxes>& result,float maxval)

{

sort(array.begin(), array.end(), [](const boxes &a1,const boxes &a2) { return a1.score > a2.score; });

vector<bool> del(array.size(), false);

for (int i = 0; i < array.size(); ++i)

{

for (int j = i+1; j < array.size(); ++j)

{

if (iou(array[i], array[j]) > maxval)del[j] = true;

}

}

for (int i = 0; i < array.size();++i) {

if (del[i]) continue;

else result.push\_back(array[i]);

}

return;

}

图像传统算法

**卷积**

1滤波器的大小应该是奇数

2滤波器矩阵所有的元素之和应该要等于1

3如果滤波器矩阵所有元素之和大于1，那么滤波后的图像就会比原图像更亮，反之，如果小于1，那么得到的图像就会变暗。如果和为0，图像不会变黑，但也会非常暗。

4对于滤波后的结构，可能会出现负数或者大于255的数值。对这种情况，我们将他们直接截断到0和255之间即可。对于负数，也可以取绝对值。

卷积核： 锐化 边缘 模糊 浮雕（有方向角）运动模糊

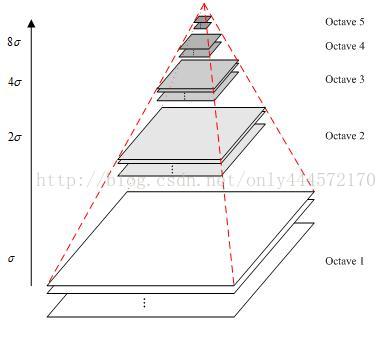
**特征处理方法**：

**颜色直方图**

**SIFT**

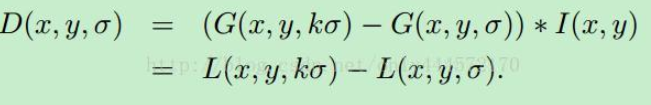
1. 尺度空间的建立

用一幅图像逐渐生成一组高斯模糊后的图像，然后将原始图像的尺寸缩小一半，再逐渐生成下一组模糊后的图像，以此类推,生成不同尺度空间的图



b 生成DOG

为了有效地在尺度空间检测到稳定的关键点位置，引入the difference-of-Gaussian function



c 在DOG空间寻找关键点

为了找到关键点，在DOG空间中对每个点，将其与当前层图像中的8点邻域和上下两层的9点邻域比较（共26个点），当其比这26个点都大或者都小的时候，确定为当前层的一个特征点。并记录下当前尺度与位置。

**HOG**（方向梯度直方图(Histogram of Oriented Gradient)）

1）灰度化（将图像看做一个x,y,z（灰度）的三维图像）；

2）采用Gamma校正法对输入图像进行颜色空间的标准化（归一化）；目的是调节图像的对比度，降低图像局部的阴影和光照变化所造成的影响，同时可以抑制噪音的干扰；

3）计算图像每个像素的梯度（包括大小和方向）；主要是为了捕获轮廓信息，同时进一步弱化光照的干扰。

4）将图像划分成小cells（例如6\*6或者8\*8像素/cell）；

5）统计每个cell的梯度直方图（不同梯度的个数），即可形成每个cell的descriptor；

6）将每几个cell组成一个block（例如33个cell/block），一个block内所有cell的特征descriptor串联起来便得到该block的HOG特征descriptor。

7）将图像image内的所有block的HOG特征descriptor串联起来就可以得到该image（你要检测的目标）的HOG特征descriptor了。这个就是最终的可供分类使用的特征向量了。

滤波就是高斯模糊

矩阵论

矩阵正定性是如何判断的（默认是二次型）

判断正定：1. 特征值全部都大于0，则矩阵正定  2.或者顺序主子式均大于0，则正定。

用处：判断函数是否可以收敛到局部或者全局最优。若Hessian正定,则函数的二阶偏导恒大于0。

矩阵为n×n的实对称矩阵A不同特征值的特征向量之间正交

矩阵求逆

LU追赶法，A-1=A\*/|A| A\*为伴随矩阵

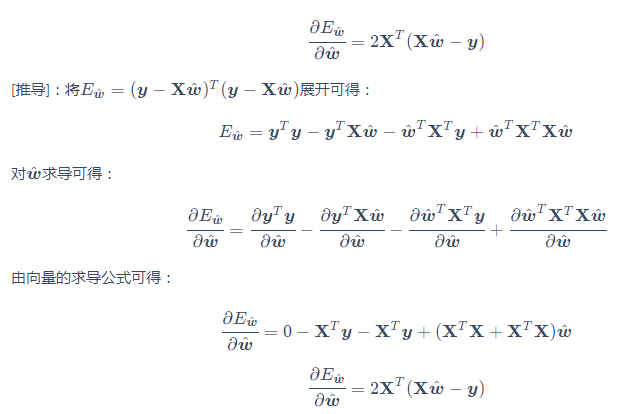
|A11 A21 An1|

|A1n A2n Ann|

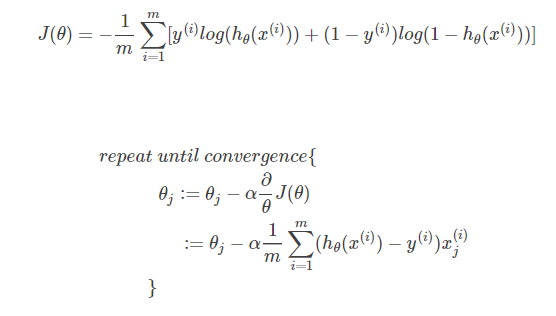
特征向量之和为迹

特征向量之积为det（行列式值）

线性模型



逻辑回归



梯度下降法只用在机器学习上，而神经网络的梯度训练方法不是叫梯度下降，而是叫前向传播，反向传播，更新权值。

