学习笔记

# --------------------------I 数学部分----------------------------------

# 概率论基本公式

## 1.1 加法公式

### （1）X是离散型随机变量

两个事件：



三个事件：



推广到一般形式，设A由构成：



上式中的n指的是括号中元素的个数，而不是A的下标。

### （2）X是连续性变量





## **1.2 乘法定理**



可以理解为：P(A)表示A发生的概率，P（B|A）代表在A发生的情况下B发生的概率，相乘代表取交集，也即A，B同时发生的概率。

推广到3个事件的乘法定理：



## 1.3条件概率

在事件A发生的情况下，B发生的概率：



## 1.4 全概率模型

设实验E地样本呢空间为S，事件Bi，i = 1，2，3…n 是样本空间地一个划分，满足 BiBj = 空，B1 U B2 U …Bn = S。如果A是E地事件，则有：



## 1.5 贝叶斯公式

设实验E地样本呢空间为S，事件Bi，i = 1，2，3…n 是样本空间地一个划分，满足 BiBj = 空，B1 U B2 U …Bn = S。如果A是E地事件，则Bi导致A地发生地概率为：



从左到右依次为：Bi的后验概率，最大似然估计，Bi的先验概率，下面为全概率公式，通过一个积分来计算，但基本上算不出来。

# 2 概率中的常用计算

## 2.1 期望

反应的是随机变量的平均值。

（1）离散型随机变量的数学期望：



（2）连续型随机变量的数学期望：



（3）期望的计算

E(C) = C

E(CX) = C·E(X)

E(X+Y) = E(X)+E(Y)

当X，Y独立时：E(XY) = E(X) E(Y)

## 2.2 方差

（1）反映的是随机变量与与其均值的离散程度，X-E(X)]表示的是样本与均值之间的差，但是插值有正有负，可能会相互抵消，所以加上平方，因为取得是样本与均值的平均方差，所以取期望。

D(X) = E[X-E(X)]2

（2）均方差（标准差，欧氏距离）：



（3）方差的计算

把方差E[X-E(X)]2看作一个函数g(X),计算方差就是计算g（X）的期望。

对于离散型随机变量：



对于连续型随机变量：



（4）方差的性质

D(C) = 0

D(CX) = C2X

D(X+C) = D(X)

设X，Y是两个随机变量，则有：



若X，Y独立，则有：



## 2.3 协方差

协方差反映的是量随机变量之间的相互关系，协方差越大，两随机变量之间相互的影响越大，反之越小。



相关系数：



COV(X,Y)>0，两随机变量正相关，即变化方向相同；小于0则变化方向相反；等于0则不相关。

# 3 有偏估计与无偏估计

若估计量的数学期望为被估计量的的真实值，那么乘该估计量是无偏的，反之是有偏的。

设θ’为θ的估计量（也就是一个关于θ的表达式），如果E(θ’) = θ，则称θ’是θ的无偏估计量，只要不相等就是有偏估计量。

**有偏估计与无偏估计反映的是估计量的准确性。**

对于高斯分布：

是无偏估计

是有偏估计

是无偏估计

# 4 先验概率与后验概率

<https://zhuanlan.zhihu.com/p/26464206>

# 5 极大似然估计（MLE）

设D = {x1，x2，x3…xn}为一组数据集，我们不知其服从何种分布，也不知其具体参数。

极大似然估计使用的前提：这组数据能代表整体的规律。

D服从某种分布Q~N（θ）

极大似然估计的思想就是通过数据集预测分布Q中的参数θ，这个θ使得D中的事件发生的可能性最大（拟合训练集）。



通过取对数，求导，求得使上述函数取最大值的θ。

# 6 常见距离

## 6.1 欧氏距离

就是最常见的求两点距离的公式。**欧氏距离是尺度相关的**，同样的数据不同的量纲计算出来的记过可能不一样。



## 6.2 马氏距离

马氏距离是尺度无关的，它不受量纲的影响。



其中是X的各项特征的协方差矩阵。

## 6.3 曼哈顿距离

就是数值之差的绝对值



# 7 满秩分解

求解矩阵A满秩分解的步骤：

（1）求矩阵A的Hermite标准型H（行最简型矩阵）

（2）设H中非零元所在的列为i1，i2，…in，则列满秩矩阵F=[A1, A2,…,An]

行满秩矩阵G就为H的非零行。

（3）A的满秩分解为 A=FG。

# 8 特征值分解

特征值分解的对象是方阵。其实就是就是矩阵对角化。

对于方阵A,特征值分解的步骤为：

（1）求出A的特征值，以及对应的特征向量，

设特征值对角矩阵为Σ，特征向量矩阵为P。

（2）则A的对角化矩阵为，同样A的特征值分解为：



# 9 奇异值分解

奇异值分解适用于所有形状的矩阵。

对于m x n维的矩阵A，其奇异值分解为：

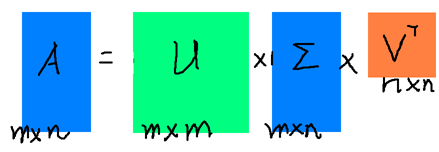


其中，U是一个m\*m的方阵（里面的向量是正交的，U里面的向量称为左奇异向量）；

Σ是一个m\*n的矩阵（除了对角线的元素都是0，对角线上的元素称为奇异值）；

VT是一个n\*n的矩阵，里面的向量也是正交的，V里面的向量称为右奇异向量），

从图片来反映几个相乘的矩阵的大小可得下面的图片



**求解m\*n维矩阵A的奇异值分解的步骤：**

（1）求解，这里的vi 就是右奇异向量，将vi组合成V并求其转置VT。

（2）计算，，这里的就是特征值，将由大至小排列成一个对角矩阵，就是上面的Σ；ui就是左奇异向量，将ui组合成U。

（3）则A的奇异值分解维：



奇异值σ跟特征值类似，在矩阵Σ中也是从大到小排列，而且σ的减少特别的快，在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上了。也就是说，我们也可以用前r大的奇异值来近似描述矩阵，这里定义一下部分奇异值分解：



**实对称矩阵的奇异值分解就等于其特征值分解。**

# 10 范数

在数学上，范数包括向量范数和矩阵范数，**向量范数表征向量空间中向量的大小，矩阵范数表征矩阵引起变化的大小**。一种非严密的解释就是，对应向量范数，向量空间中的向量都是有大小的，这个大小如何度量，就是用范数来度量的，不同的范数都可以来度量这个大小，就好比米和尺都可以来度量远近一样；对于矩阵范数，学过线性代数，我们知道，通过运算，可以将向量X变化为B，矩阵范数就是来度量这个变化大小的。

<https://blog.csdn.net/lz867422770/article/details/80013600>

## 10.1 向量范数

对于向量X = [x1, x2, …, xn]

（1）L1范数：



L1也被称为曼哈顿距离，最小绝对误差。

（2）L2范数



（3）L3范数



## 10.2 矩阵范数

对于矩阵A∈Rmxn

（1）L1范数



列和范数，分别计算所有列的元素的绝对值之和，取出最大值。

（2）L2范数

，λ1为ATA最大的特征值，L2范数也称为谱范数。

（3）L3范数



行和范数，分别计算所有行的元素的绝对值之和，取出最大值。

（4）F范数



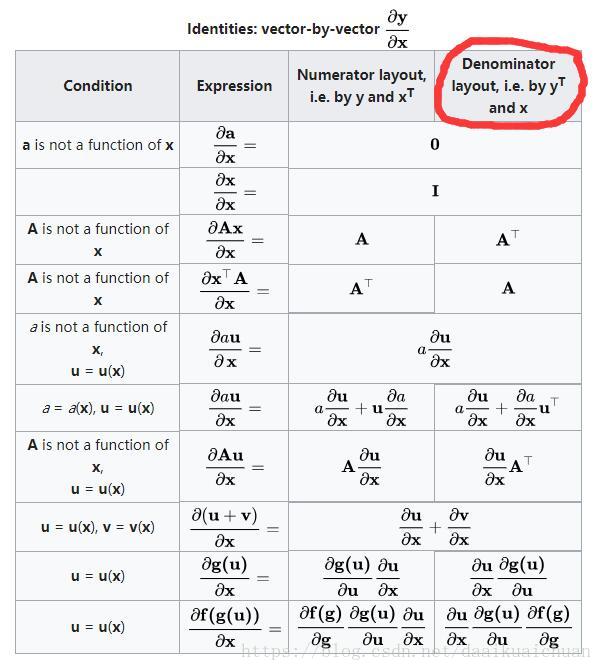
# 11 矩阵求导

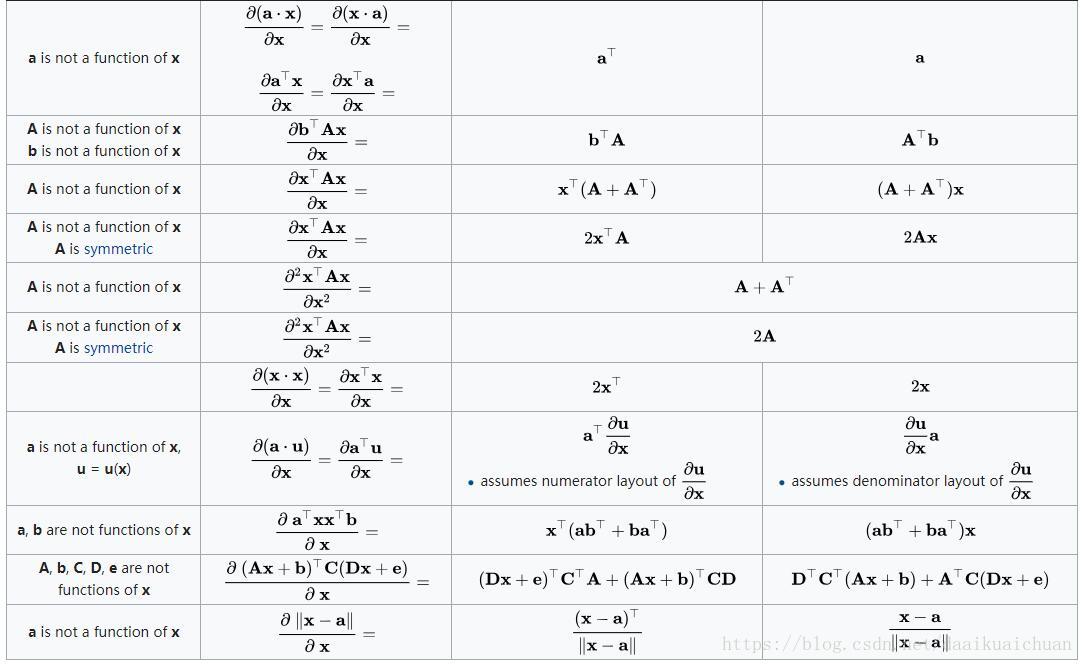
与向量或矩阵相关的求导中，本质其实就是因变量的每一个元素对自变量的每一个元素都进行求导，然后再通过某种方式进行排列。

主要的排列方式有分子布局与分母布局，分子布局意为最后的结果与分子维度为准，分母布局意为最后的结果以分母的维度为准。两种布局之间差一个转置。

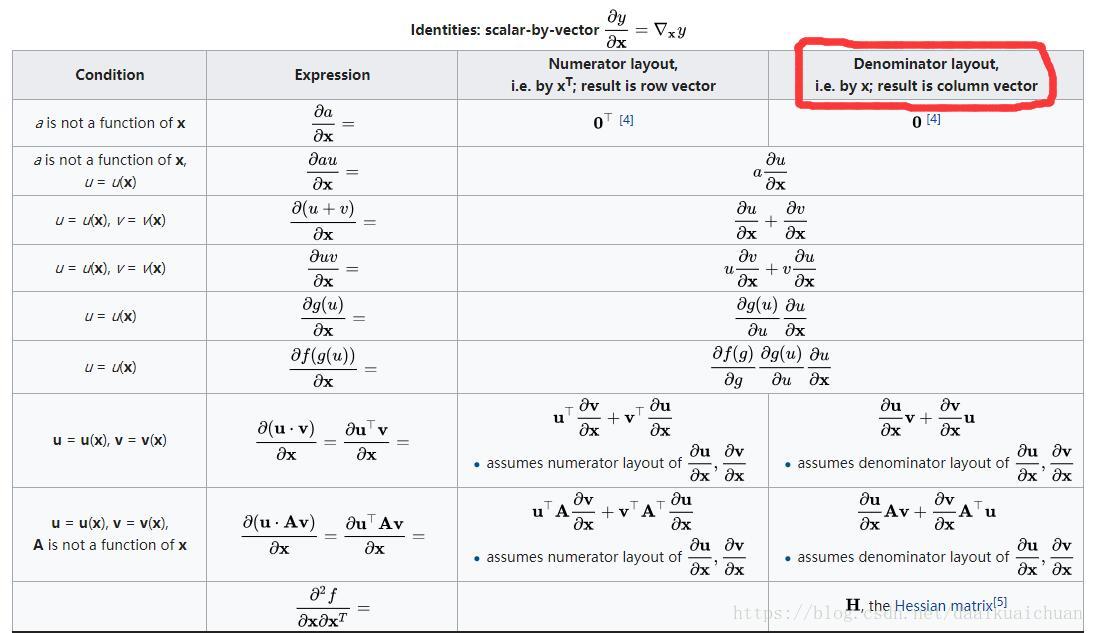
**通常情况下，向量对标量求导采用分子布局，标量对向量求导采用分母布局，向量对向量求导采用分母布局。**

## 11.1 向量对向量求导（向量与矩阵都可充当分子或分母）

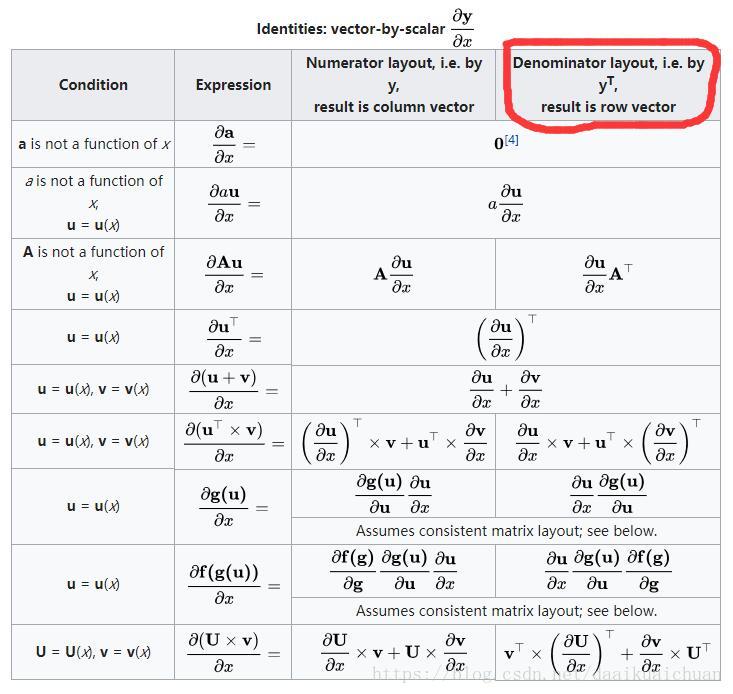




## 11.2 标量对向量求导



## **11.3 向量对标量求导**



# 15 拉格朗日乘子法

<https://blog.csdn.net/THmen/article/details/87366904>

拉格朗日乘子法的本质就是通过引入拉格朗日乘子（λ），将条件约束问题转化为无约束问题。

问题描述：求函数在约束下的极值。

1 引入拉格朗日算子λ，将问题转化为求的极值

2 将F分别对x，y，λ求偏导，并令各偏导表达式为0.

3 对偏导方程进行整合，进而直接求出x，y或通过λ求出x，y。

# 16 动态规划问题

<https://www.zhihu.com/question/39948290>

<https://www.bilibili.com/video/BV18x411V7fm?from=search&seid=11935627856360280076>

动态规划（DP，Dynamic programming）解决的是重复计算的问题。方法是先将分问题的计算结果储存下来，在后面直接调用结果，而不用再计算一遍。

# ----------------------II 机器/深度学习部分------------------------------

# 1 特征缩放（feature scaling）

## 1.1 为什么要进行特征缩放（特征缩放的好处）？

（1）原始数据各项特征的量纲不同，进行模型训练时，越大的量纲对模型的影响越大，进而会导致得到不好的结果。特征缩放可以使数据变为纯量（标量），消除量纲影响，。

（2）特征缩放可以使特征分布接近类圆形，使梯度下降的步子更平稳，捡块收敛速度。

## 1.2 特征缩放常用方法

特征缩放的常用方法分两种，一种是归一化，指将特征值缩放至指定的区间中，如[0,1], [0,255]。二为标准化，将原分布转化为均值为0，方差为1的分布（不一定是标准正态分布）。

**归一化**

（1）min-max normalization



最后将特征的数值范围缩放至[0,1]中。

（2）mean normalization



最后将特征的数值范围缩放至[-1,1]中。

**标准化**



# 2 熵、交叉熵

<https://blog.csdn.net/tsyccnh/article/details/79163834>

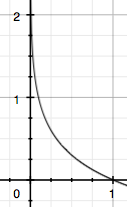
## 2.1 信息量

事件的信息量的大小与事件发生的概率有关，发生概率小的事件发生了，我们获得的信息熵大；发生概率大的事件发生了，我们获得的信息量小。

设X是一个随机变量，事件{x1,x2,x3...xn}∈X，那么对于事件xi，我们获得的信息量为：



因为xi表示概率，故H(x)的图像如下：



## 2.2 熵

信息量衡量的对象是事件X的某一种情况，熵就是用来衡量该事件的所有情况的信息量的大小，也就是X的所有情况的信息量的和。

设事件X有n种情况，{x1,x2,x3...xn}∈X，则有：



对于只有两种可能性的问题，如抛硬币等二分类问题，事件的熵可以表示为：



## 2.3 相对熵（KL散度）

相对熵又称KL散度,如果我们对于同一个随机变量 x 有两个单独的概率分布 P(x) 和 Q(x)，我们可以使用 KL 散度（Kullback-Leibler (KL) divergence）来衡量这两个分布的差异

维基百科对相对熵的定义：

In the context of machine learning, DKL(P‖Q) is often called the information gain achieved if P is used instead of Q.

即如果用P来描述目标问题，而不是用Q来描述目标问题，得到的信息增量。

离散型随机变量X的KL散度的计算公式：



连续型随机变量X的KL散度计算公式：



上式中，p代表事件的真实分布，q代表模型估计的分布，n代表事件可能发生的情况。

容易看出，KL越小，模型预测的分布与真实分布越接近。

## 2.4 交叉熵(cross entropy)

对KL计算公式进行变形：



前一部分就是事件的熵，后一部分就是交叉熵：



原本用交叉熵来表示模型的精度就够了，但是由于事件的熵是不变的(事件的真实分布只有一个)，故舍去前一部分，直接用交叉熵作为模型精度的评判标准。

在机器学习中，p(xi)代表事件的真实分布，也即是y’，q(xi)代表预测的分布，也就是y，所以交叉熵在机器学历里的表达式为：



# 3 线性回归的基本前提

线性回归需要满足四个前提假设

## 3.1 Linearity 线性

应变量和每个自变量都是线性关系。

## 3.2 Indpendence 独立性

对于所有的自变量，它们各自的误差项相互之间是独立的。

自变量间相互独立与各自变量的误差间相互独立是等价的。

## 3.3 Normality 正态性

不同特征的误差项服从正态分布N（0，σ2）。

不同的特征的观测值y都服从正太分布。

## 3.4 Equal-variance 等方差

所有的误差项具有同样方差。

不同的特征都服从相同方差的正太分布，但他们的均值不同。

这四个假设的首字母，合起来就是**LINE**，这样很好记。

**如何理解误差服从均值为0的正太分布N(0,σ2)？**

即误差在0的两边近似对称，那么观测值y在均值两边的数量相近，且与均值的差距之和也近似相等。

# 4 颜色直方图

颜色直方图对图片进行描述地一项指标，他表示每种颜色在图片中的占比，而不关注每种颜色的位置。

# 5 top1 error & top5 error

## top1 error：

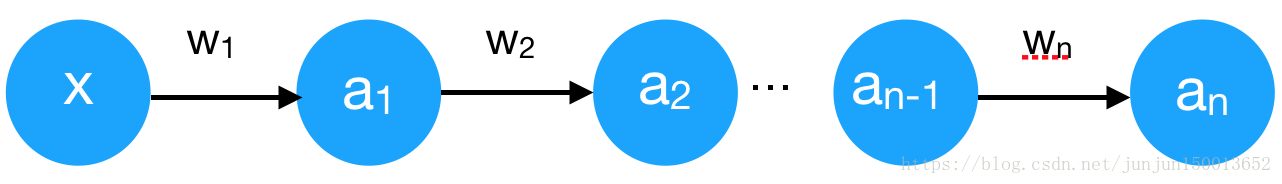


## top5 error：

ImageNet中图片可能由1000多个类别，某张图片可能可以分为好几个类别。输入一张图片，输出的是其属于这1000个类别的概率，取前5个类别，如果正确的标签在这前5个类别中，则看作分类正确，否则分类错误。



# 6 梯度消失与梯度爆炸



考虑一个最简单的神经网络，每一个神经层只有一个神经元，没有偏置参数。输入为x，第i个神经元的输入为zi，输出为ai，激活函数为sigmoid，先进性前向传播：





…



设C为代价函数，求解C对的梯度：



也即：



## 梯度消失（梯度弥散）

sigmoid函数的导数在越大或越小的时候，梯度都越趋近于0；在0处梯度取最大值，也只有0.25。

输入的数据一般都会进行标准化（均值为0，方差为1），所以权重一般|wi|<1（可能跟均值，方差有关）,所以对于每层神经层，都有。随着网络的加深，呈指数级缩小，越来越趋近于0。结合梯度下降的公式，这就会导致越靠近输入层的权值越难被更新，以至于难以得到好的训练结果。

## 梯度爆炸

梯度爆炸是由于初始权值设置过大导致的，也即当时，随着网络的加深，呈指数级增长，最后会导致参数更新的很大。

# 7 Batch Normalization

## Internal Covariate Shift

神经网络在训练过程中，随着底层神经元参数的更新，在相同输入的情况下，底层神经元的输出也会变化，而且这很有可能产生离群点，使得数据的分布偏离类圆形，加大训练难度，耗费训练时间。（数据分布越偏离类圆形，你越不敢使用大些的学习效率，因为说不定步子就迈大了，导致偏离了正确方向，进而加大了训练难度）。而且网络越深，这种偏离可能越严重，后果越恶劣。

Google将这种由于底层参数的更新而导致高层参数学习更新困难的现象定义为ICS。

## Batch Normalization

BN通过对每层神经元的输入进行标准化，可以使每层神经元的输入属于相同分布（均值为0，方差为1），进而解决ICS问题。

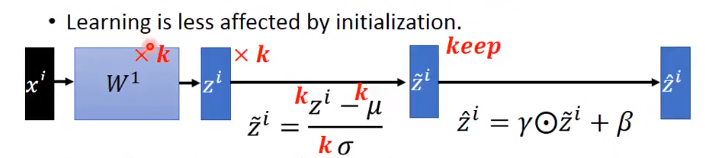
**BN的好处在于：**

（1）减少模型训练时间

①可以使用更大的学习效率。因为数据分布更趋近于圆形，说以哪怕步子迈得大一点，也不会偏离方向。

②对于sigmoid，tanh等激活函数，由于BN将岩本分布拉回到0附近，所以会缓解梯度消失/爆炸问题，能使网络训练得更快，减少模型训练时间。

（2）减少初始值对训练得影响



（2）BN有一定的正则化效果

在Batch Normalization中，由于我们使用mini-batch的均值与方差作为对整体训练样本均值与方差的估计，尽管每一个batch中的数据都是从总体样本中抽样得到，但不同mini-batch的均值与方差会有所不同，这就为网络的学习过程中增加了随机噪音，与Dropout通过关闭神经元给网络训练带来噪音类似，在一定程度上对模型起到了正则化的效果（当然没有正则化，Dropout那么明显）。

**BN的缺点：**不适用于batch太小的情况，因为难以体现总体的均值与方差。

**引入BN的缩放参数γ与位移参数β**，是因为在执行BN时，可能改变数据的原始分布，这两个参数是希望能还原样本的分布。这是两个可以学习的参数

# 8 白化（Whitting）

白化是指数据的预处理，也即对数据进行归一化，加速模型训练。白化分两种，一种是PCA白化，另一种是ZAC白化。

PCA白化是指先对数据进行PCA处理，然后再进行归一化。

ZCA白化是指再PCA白化的基础上，再将数据还原会原始样本空间。

由于白化计算成本过高，现在差不多被摈弃了。

# 9 Softmax

softmax没有分类作用，它只是将前一个layer的分类结果表示为相对概率的结果，以便于观测和比较。softmax函数的定义如下：



其中i为类别索引，n为类别总数，Si表示某个元素的指数与所有元素的指数和之比，也即某个元素所代表的相对概率大小。

<https://blog.csdn.net/qq_32642107/article/details/97270994>

为什么前一层神经元的输出能表示分类结果呢？

假设最后一层（输出层）为softmax层，倒数第二层为全连接层，可以将导数第二层转化为全卷积层，所有神经元各自代表的卷积核分别对倒数第三层输出的feature map进行卷积，会得到一个输出向量，那么这个向量代表的就是全卷积层对前一层的feature map的整合结果，也即得分，那么这也是全部连接层的输出，代表每种整合方式的得分。此时，分类已经结束了，但是输出向量或许不是那么直观，于是通过softmax转化为相对概率，代表的是每个分类结果为正确的概率。

# 17 CNN

CNN的经典结构是：卷积/池化层进行特征提取，全连接层进行特征整合，将数据从特征空间映射到标签空间，进行分类。

## 17.1 CNN的三个特点

（1）局部感受野（局部连接）

人类观察图片的方式：先观察小块局部图像，再将局部图像连接，慢慢拼接，一步一步得到了对图片的整体认识。

体现在神经网络上就是，每个神经元不与整个图像的每个像素连接，而是与部分区域连接，提取局部特征，再在高层进行整合。

（2）参数（权值）共享

根据图片的“静态性“，相同成分的局部区域在图片上的不同位置，仍然可以用同一套参数去检测。隐藏层不同的神经元连接图片的不同的局部区域，因为检测的特征不因为位置的改变而受影响，所以可以使用相同的参数，这就大大减少了参数量。此时参数个数与隐藏层神经元个数无关，仅与卷积核大小有关。

由全连接改为局部连接不会减少参数量，但是加上权值共享，就能大大地减少参数量。

（3）池化

虽然通过局部连接&权值共享降低了参数量，但参数量还是较多。池化通过最大池化，均值池化，提取出局部区域中具有代表性地特征，可以进一步减少参数量。

从感受野的角度看，池化将特征空间进行压缩，能进一步增大模型的感受野，并保留空间信息，但是会丢失一部分数据。

池化还有一定的平移，旋转不变性。

## CNN中的相关计算

卷积后feature map大小计算：



要保持feature map大小不变，padding大小的计算：



池化后feature map的大小计算与卷积的计算方式相同，只需要注意：

卷积向下取整，池化向上取整。

感受野大小计算：

在卷积神经网络中，感受野的定义是 卷积神经网络每一层输出的特征图（feature map）上的像素点在**原始图像**上映射的区域大小。感受野边长计算公式为：



其中计算公式如下



## 卷积核大小的选择

<https://www.jianshu.com/p/06002775e0d4>

<https://www.zhihu.com/question/38098038/answer/271448608>

喜欢小而深，拒绝大而短

通过小的卷积核的堆叠，可以替代大的卷积核，感受野不变。用小而深替代大而短的好处在于：

中间经过更多非线性函数，使模型的表达能力增强，更具判别力。

参数更少，训练加快。

## 全卷积网络

<https://www.cnblogs.com/gujianhan/p/6030639.html>

CNN的经典结构是：卷积/池化层进行特征提取，全连接层进行特征整合，将数据从特征空间映射到标签空间，进行分类。

### 使用全连接层的问题

但是对于全连接层而言，框架一定下来，全连接层的参数矩阵的形状也就定了，那么也就意为着全连接层的输入维度是确定的，进而整个模型的输入也是要确定的。 这也就意味着不符合模型默认大小的输入需要经过变换（裁剪/上采样），这对模型产生了一定的限制。

### 用卷积层替换全连接层的可行性

卷积层其实就是全连接层的特殊情形，它是在全连接层的基础上应用了局部连接和权值共享的假设。如果全连接层的前一层是卷积层，则将全连接层的权值矩阵resize成前一个卷积层输出的feature map的形状，要转换的全连接有多少个神经元，那么就resize成多少个multi-channel 的卷积核；如果前一层是全连接层，那么前一层的输出multi-channel的，高和宽都是1的feature map，那么这第二个卷积层的卷积核应该是：通道数等于前一层神经元个数，高和宽都为1。其他的转换相同。

### 将普通CNN转化为FCN的好处

解除了全连接层对于输入图片大小的限制，提高模型的泛化性。

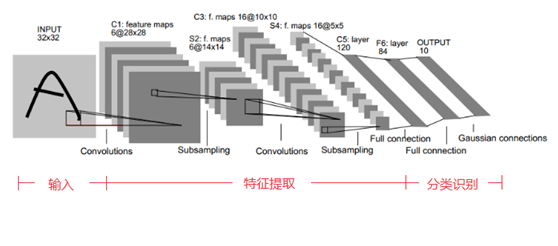
CNN在训练时往往还是使用全连接层进行训练，只是在测试时转换为FCN。如果输入的图片大于默认的图片大小，那么模型最后输出的可能是一个multi-channel的feature map，对于这个每一个channel的特征图，我们给予另一个名字：heat map。

那么如何从heat map得到最后的分类结果呢？

如果最后的heat map是1000\*6\*6的，也就是说有1000个维度为6\*6的heat map，将这1000个heat map上采样到原图大小。对于原图中的每个像素，逐个像素地求其在这1000个上采样得到的图片中（概率）最大值描述，作为该像素地分类结果。最后就产生原图逐像素的分类结果。

## 17.2 LeNet

<https://my.oschina.net/u/876354/blog/1632862>



lenet的目的是进行手写体识别。lenet的输入是32\*32的手写体图片，特征提取部分由3个卷积层和2个池化层组成，分类部分是一个全连接层，最后再加上一个输出层。

（1）卷积层1：6@28×28

卷积核：5×5，步长：1

（2）池化层2：6@14×14

池化单元：2×2，步长：2

（3）卷积层3：16@10×10

卷积核：5×5，步长：1

需要注意的是卷积层3的卷积核并不是对池化层2的feature map的所有channel相连，而是连接了3层，4层或是6层，这样做有利于增加特征数量。

（4）池化层4：16@5×5

池化单元：2×2，步长：2

（5）卷积层5：120@1×1

卷积核：5×5，步长：1

因为卷积核的大小与feature map大小相同，所以可以认为是全连接。当然这只是一个巧合。

（6）全连接层6：84

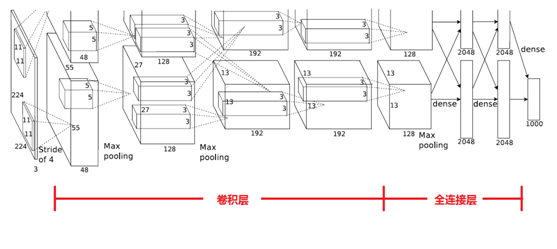
84个神经元的设计来源于输出层。

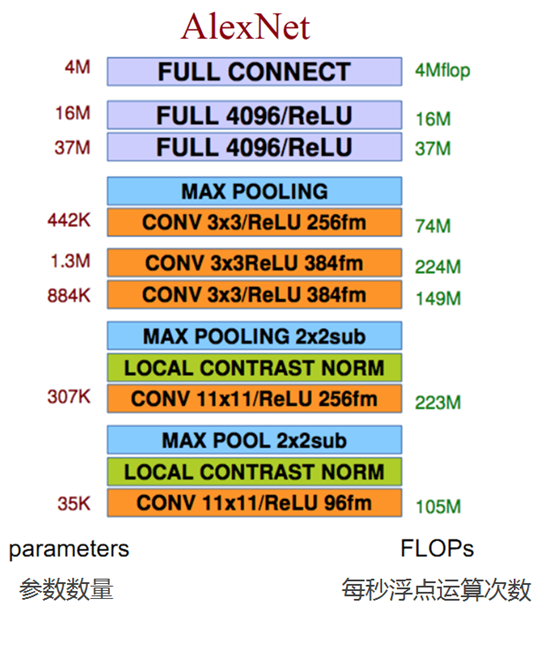
（7）输出层7：10

10个神经元是因为数字范围从0~9.

## 17.3 AlexNet

<https://my.oschina.net/u/876354/blog/1633143>



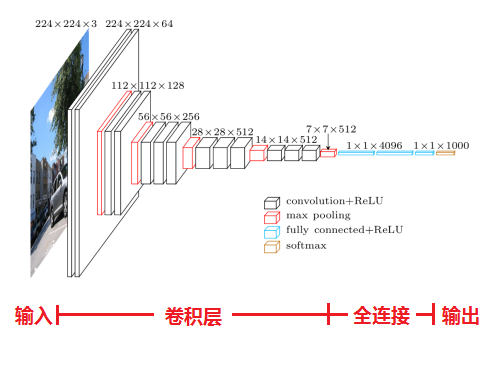


AlexNet之所以能够成功，跟这个模型设计的特点有关，主要有：

* 使用了非线性激活函数：ReLU
* 防止过拟合的方法：Dropout，数据扩充（Data augmentation）
* 其他：多GPU实现，LRN归一化层的使用

## VGG

<https://my.oschina.net/u/876354/blog/1634322>



从总体上来看，VGG是在模仿AlexNet，但是具体来讲，VGG又有很多的改进。VGG的特点：

**①结构简洁**

VGG由5层卷积层、3层全连接层、softmax输出层构成，层与层之间使用max-pooling（最大化池）分开，所有隐层的激活单元都采用ReLU函数。

**②用小卷积核和多卷积子层替代大卷积层**

好处有：减少参数数量、使模型非线性表达能力更强

**③小池化核**

相对于AlexNet的3×3的池化核，VGG采用的都是2×2的池化核。

**④ 通道数更多**

VGG网络第一层的通道数为64，后面每层都进行了翻倍，最多到512个通道，通道数的增加，使得更多的信息可以被提取出来。

**⑤层数更深，感受野更大**

由于卷积核专注于扩大通道数、池化专注于缩小宽和高，使得模型架构上更深更宽的同时，控制了计算量的增加规模（大都在100到150万之间）。

**⑥全连接转卷积（FCN）**

使模型输入图片的大小不受限制。