深度学习总结

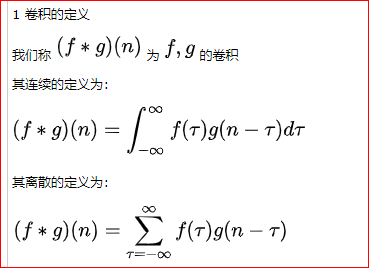
# 说明

本文档主要总结深度学习中相关的概念

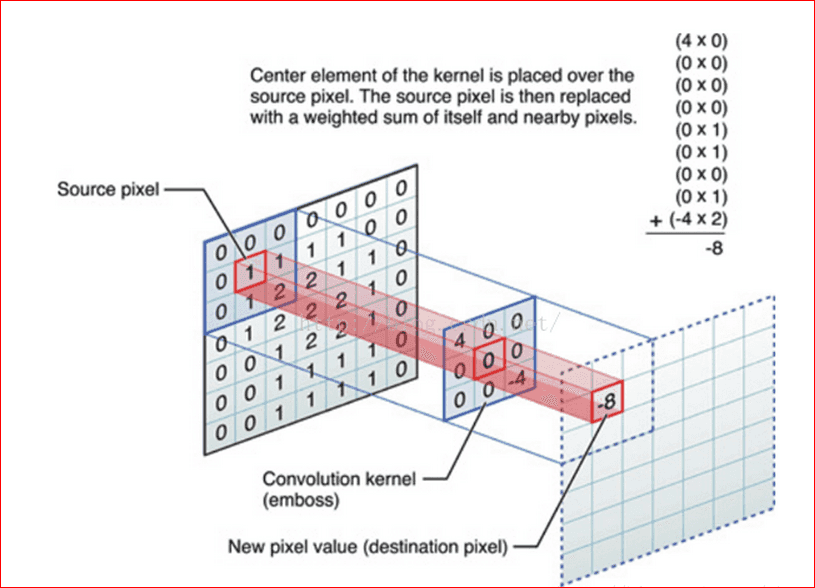
# 相关概念

## 卷积

从数学上讲，卷积是一种运算，其定义如下：



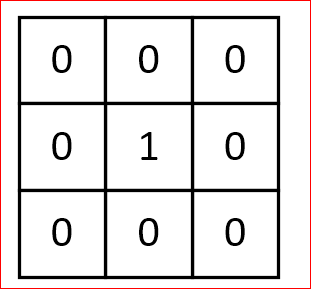
数字图像是一个二维的离散信号，对数字图像做卷积操作其实就是利用卷积核（卷积模板）在图像上滑动，将图像点上的像素灰度值与对应的卷积核上的数值相乘，然后将所有相乘后的值相加作为卷积核中间像素对应的图像上像素的灰度值，并最终滑动完所有图像的过程。



卷积核选择上一般是奇数，例如3\*3，5\*5，7\*7等，当处理边缘像素时，经常通过补零或其他值的方式进行操作；此外卷积核所有元素之和一般等一1，这时为了原始图像的亮度守恒；当然也可以不等于1，当大于1时，图像就会变亮，小于1时，图像变暗。

不同的卷积核在图像处理上有不同的意义：

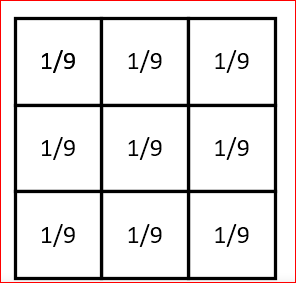
### 如下是一个没有任何作用的卷积核：



该卷积核处理后，图像的像素值不会发生任何变化。

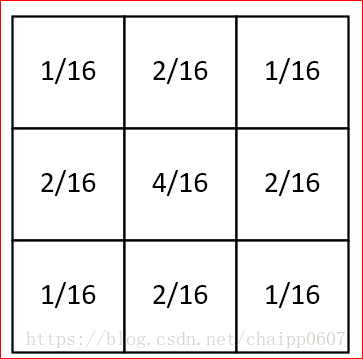
### 平滑均值滤波

当把卷积核选择为如下样子时



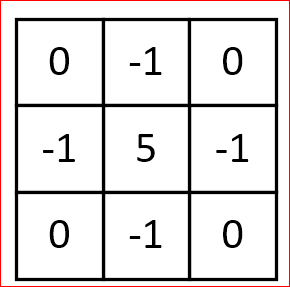
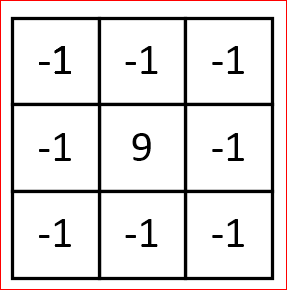
会将待处理像素周围9个像素的平均值代替中间像素，所以会起到图像平滑的效果。

### 卷积核选择为如下取值，则得到高斯平滑



高斯平滑突出了中心点在目标图像中的权重，有更好的平滑效果。

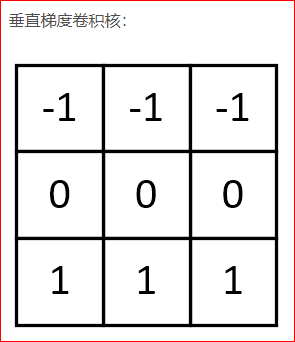
### 选择如下卷积核，则得到图像锐化效果：



该卷积核利用了图像中的边缘信息比周围像素有更高的对比度，通过卷积更强化了这种对比，从而使图像显的菱角分明，画面清晰。

### 梯度Prewitt

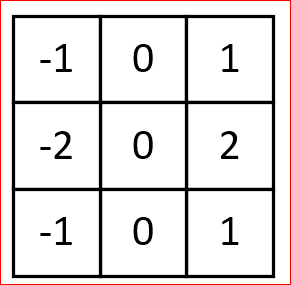
通过选择如下两个卷积核，可以在水平和垂直方向上得到较好的边缘检测效果



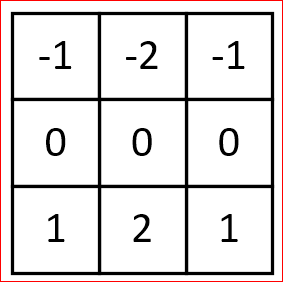
### Soble边缘检测

Soble边缘检测更强调和边缘相邻的像素点对边缘的影响，卷积核可以选择成下面这个样子

水平梯度：

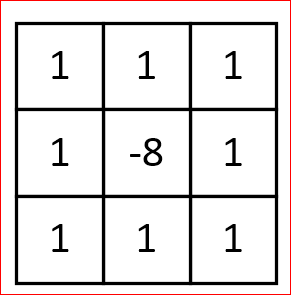


垂直梯度：



### 梯度laplacian

Laplacian是一种图像锐化方法，同时也可以做边缘检测，而且边缘检测的应用并不局限于水平或垂直方向，卷积核可以选择如下样子：



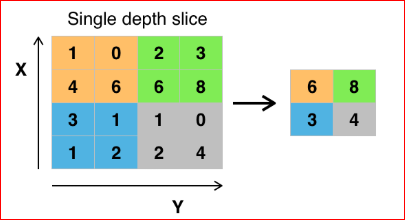
## 池化

池化是卷积神经网络中的一个重要概念，实际上是一种形式的降采样，有多种不同形式的非线性池化函数，最常见的是“最大池化”，其运作过程是，将图像划分为若干个区域，在每个区域上输出最大值。

CNN的卷积层之间会周期性的插入池化层。

当前最常用的池化层是每间隔2个元素从图像划分出2\*2的区块，然后对每个区块中的4个数取最大值，从而减少计算量。

下图是一个“最大池化”的样例：



经过池化以后，相比输入，输出结果会变小，池化层的引入是仿照人的视觉系统，对输入对象进行降维和抽象。

池化层的功效：

1：特征不变型：池化操作关注是否存在某些特征而不是特征的位置。

2：特征降维：池化相当于在空间范围内做了维度约减，从而使的模型可以抽取更广范围的特征，同时减少下一层的输入大小。

3：在一定程度上防止过拟合。

## 激活函数

在神经网络中，如果不使用激活函数，则每一层的输出都是上一层输入的线性映射，无论神经网络有多少层，输出都是输入的线性组合。

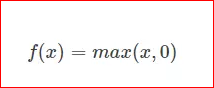
为了应对这种情况，引入了非线性函数做为激活函数，这样输入到最终输出之间就不是简单的线性映射，从而可以逼近任意函数。

激活函数的作用是增加神经网络的非线性，以便神经网络可以接近任意函数。

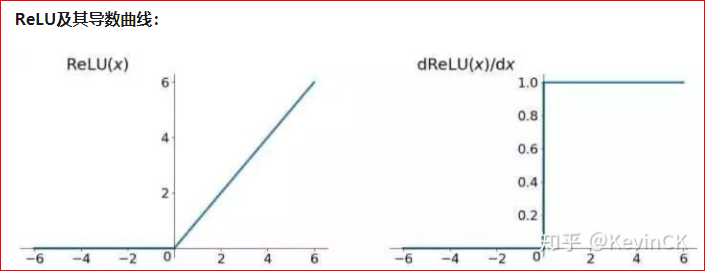
常见的激活函数有sigmod, relu, tanh，LReLU, PReLU, RReLU, ELU, softplus, softsign等。

### 激活函数relu

Relu的函数定义如下



即将输入中的非最大值设置为0，其函数图型与导数图型如下：



从图上可以看出，relu的有效导数是1，解决了神经网络中梯度消失的问题，从而使的网络可训练；同时relu是非线性函数（一阶导数不是常数）

Relu在x>0，导数取值为1，可保证在链式反应中梯度不会消失。

Relu在x<0，导数取值为0有以下特点：

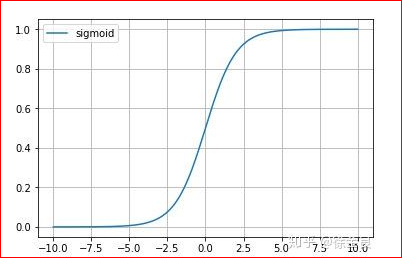
深度学习的目的是根据大批量数据样本，从错综复杂的数据关系中找到关键特征，从数学的角度也就是将密集矩阵转换为稀疏矩阵，保留其中的关键信息，relu，x<0输出为0，其实就是一个去除噪音，稀疏矩阵的过程。当然，relu x<0输出为0，可能会导致训练模型无法学习到有效特征（尤其是学习率过大的时，会导致神经元处于dead状态），因此当使用relu的模型中，学习率不能设置太大。

### 激活函数sigmod

Sigmod激活函数定义如下：



其函数图像形式如下：



Sigmod函数的值域是（0，1），其导数形式如下：

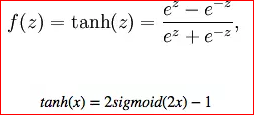


通过sigmod函数可以将任何值转换到（0，1）之间，可以用于二分类（结果大于0.5可以认为是1，结果小于0.5可以认为是0），在神经网络中，如果使用sigmod函数作为激活函数，具有如下的一些特征：

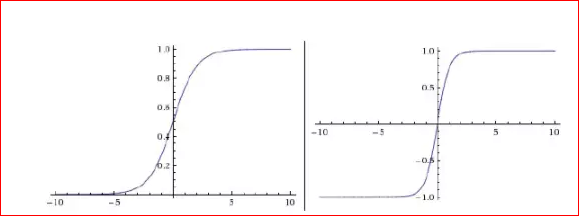
1. sigmod函数的输出范围在0~1之间，均值为0.5，需要做数据偏移，不放变下一层学习。
2. 当输入x很小或很大时，存在导数很小的情况，随着神经网络层数的增加，梯度后向传播时，基本无法引起参数的扰动，即传递loss信息到前面的网络很困难，从而导致出现梯度消失现象。

### 激活函数tanh

Tanh函数也被称为双正切函数，值域范围在[-1, 1]之间。函数定义如下



该函数曲线形式如下图：

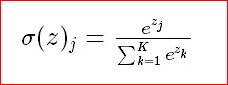


Tanh函数也存在梯度消失的问题。

Tanh函数是完全可微分的。

### Sofmax函数

Softmax函数常用于多分类神经网络输出，函数定义如下：



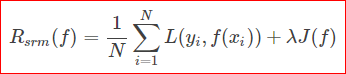
在应用上，softmax函数将一个k维的向量（a1,a2,a3,a4…）映射到（b1,b2,b3,b4..）其中bi时一个0到1的常数，且其和时1.0，然后可以根据bi的取值大小进行分类。

Softmax建模使用的分布是多项式分布。

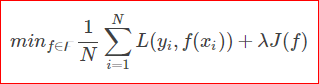
## 损失函数

损失函数是用于评估模型预测值与真实值之间的不一致程度，一般是一个非负值实值函数，损失函数越小，模型的鲁棒性越好。

损失函数是经验风险函数的核心部分，也是结构风险函数的重要组成部分，模型的结构风险函数包括了经验风险和正则项，可以表示成如下形式：



其中前一部分描述经验风险项，后一部分为正则化，正则化用于控制模型的复杂度，所以可以通过最小化结构风险的方式找最优模型，即求解如下最优化问题：

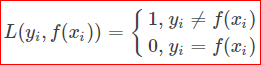


当样本数量足够时，经验风险最小化可以达到很好的学习效果，但在样本容量有限时，模型容易产生过拟合现象。

常见的损失函数有如下这些。

### 0-1损失函数

0-1损失函数定义如下，经常被用于分类任务中。



如果预测值与数据标记值相同，则为0，否则为1

### 平方损失函数

平方损失函数的定义如下：



平方损失函数经常用于回归问题中，权重可以直接初始化，然后通过梯度下降不断更新，从而达到最小。

### 绝对损失函数

绝对损失函数定义如下：



绝对损失函数也经常用于回归问题中。

### 对数损失函数（交叉熵损失函数）

对数损失函数的定义如下：



对数损失函数适用于逻辑回归问题中。

### 铰链损失函数

函数定义如下：



该损失函数在SVM中使用较多，如果模型分类正确，则损失为0，否则损失为1-mi(w)。在SVM模型中，该损失函数的详细定义如下：



### 指数损失函数

指数损失函数的定义如下:



该损失函数在Boosting算法中使用较多。

## 训练过程中的Batch\_size

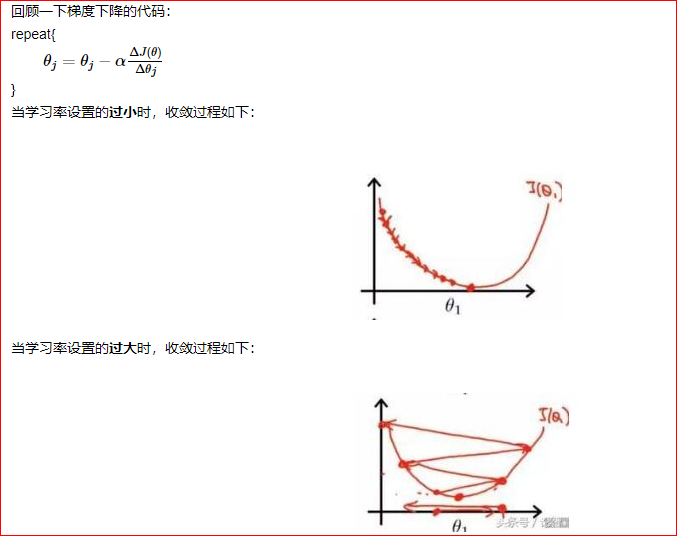
Batchsize：是指一次迭代训练中的样本个数，也是一次参数更新的样本数。

在数据集样本确定的情况下，设置不同的batchsize取值具有不同的效果。

当batchsize取值较大时，总体训练次数减少，可以在更短的时间内完成训练，但较大的batchsize会导致模型的泛化能力下降（在一定范围内，增加batchsize有助于训练收敛稳定，但随着batchsize的增加，模型性能会出现下降）；同时大的batchsize需要更大的内存或显存支持。

## 学习率learning\_rate

学习率learning\_rate控制模型训练的进度，当学习率过小时，训练过程不容易收敛，学习率过大时，可能导致梯度在最小值左右震荡，无法收敛到最小值。



# 计算机图像处理问题

计算机视觉中，关于图像主要有四类任务

1：分类，即给定一张图像或视频，判断里面包含什么类别的目标。

2：定位，定位出目标在图像中的位置。

3：检测，即定位出目标的位置，同时给出目标是什么。

4：分割，即解决像素属于哪个目标