**A.1x1卷积的作用是什么？**

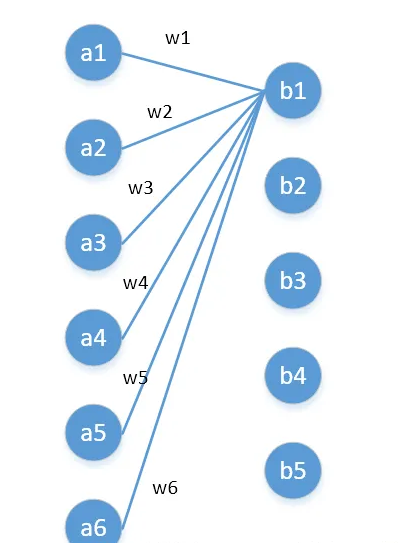
可以当成是全连接网络。全连接和1x1卷积最大的区别在于可以不限制输入的大小，全连接之前需要将tensor拉成一维的，然后做全连接，这个时候参数的数量会受到上下层神经元个数的影响，所以一般会限制输入图像。

左边6个神经元，分别是a1—a6，通过全连接之后变成5个，分别是b1—b5

左边6个神经元相当于输入特征里面的channels：6

右边5个神经元相当于1\*1卷积之后的新的特征channels：5

左边 W\*H\*6 经过 1\*1\*5的卷积核就能实现全连接



卷积层之后会进行激活函数的处理，增加非线性映射的次数

可用于降维，减少计算量，如resnet的bottleneck结构，或者升维，如mobilenetv2

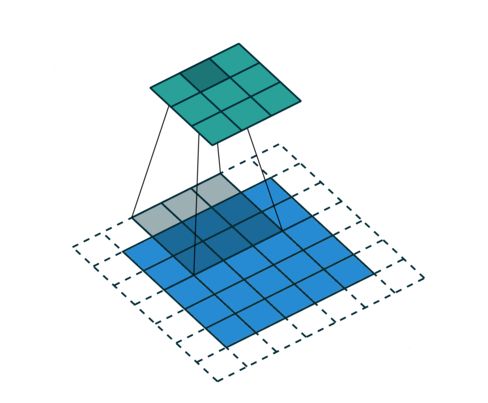
卷积之前会有BN层操作，实现对不同特征进行归一化操作

实现不同channel上的跨通道信息交互和特征融合

**A.常见的卷积方式有哪些，它们的作用是什么?**

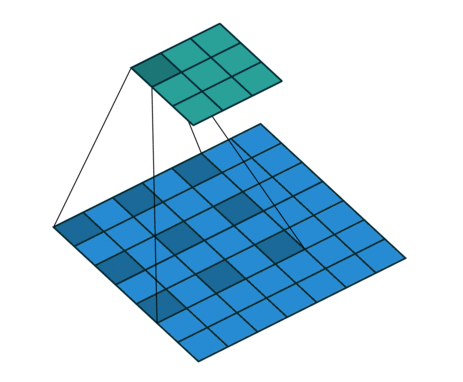
1.普通卷积

卷积在数学上用通俗的话来说就是输入矩阵与卷积核（卷积核也是矩阵）进行对应元素相乘并求和，所以一次卷积的结果的输出是一个数，最后对整个输入输入矩阵进行遍历，最终得到一个结果矩阵，说白了就是一个卷积核在图像上滑动，并求取对应元素相乘求和的过程。



1.扩张卷积

扩张卷积也叫作空洞卷积或者膨胀卷积，是在标准的卷积中加入dilation rate，称为扩张率，指的是卷积核的点的间隔数量，以此来增加模型的感受野。常规的卷积操作dilatation rate为1。扩张的卷积为卷积层引入另一个参数，称为扩张率。扩张率为2的3x3的扩张卷积与5x5的普通卷积具有相同的感受野。



3.转置卷积

转置卷积 转置卷积又叫反卷积、逆卷积。主要是将提取的特征图还原到和原图同样的尺寸大小。

4.深度可分离卷积

深度可分离卷积由Depthwise卷积和Pointwise卷积组成。

深度可分离卷积的过程是:

① 用16个1通道的3×3大小的卷积核分别与输入的16通道的数据做卷积。这里使用了16个1通道的卷积核，输入数据的每个通道用1个3×3的卷积核卷积，得到了16个通道的特征图，我们说该步操作是depthwise（逐层）的。

② 在叠加16个特征图之前，接着用32个16通道的1×1大小的卷积核在这16个特征图进行卷积运算，将16个通道的信息进行融合，用1×1的卷积进行不同通道间的信息融合，我们说该步操作是pointwise（逐像素）的。这样我们可以算出整个过程使用了3×3×16+（1×1×16）×32 =656个参数。

通过上述计算可以看出，使用深度卷积的方式参数656，标准卷积参数为4068，标准卷积比深度卷积多出很多参数。

5.分组卷积

分组卷积最早在AlexNet中出现，由于当时的硬件资源有限，训练AlexNet时卷积操作不能全部放在同一个GPU处理，因此作者把feature maps分给多个GPU分别进行处理，最后把多个GPU的结果进行融合。

从一个具体的例子来看，Group conv本身就极大地减少了参数。比如当输入通道为256，输出通道也为256，kernel size为3×3，不做Group conv参数为256×3×3×256。实施分组卷积时，若group为8，每个group的输入通道和输出通道均为32，参数为8×32×3×3×32，是原来的八分之一。而Group conv最后每一组输出的feature maps应该是以concatenate的方式组合。

AlexNe作者认为group conv的方式能够增加 filter之间的对角相关性，而且能够减少训练参数，不容易过拟合，这类似于正则的效果。

6.可变形卷积

可变形卷积是指卷积核在每一个元素上额外增加了一个参数方向参数，这样卷积核就能在训练过程中扩展到很大的范围。

**A.怎么理解卷积核，常见的卷积核大小有哪些，它的大小如何选取?**

卷积核的作用是提取图像的特征，然而一个卷积核是不够的，因为一个卷积核只能反应图像的某一个特征，所以我们需要多个卷积核，这些不同的卷积核可以提取到图像不同的特征，从而让我们的模型学习图像特征的能力更强。

卷积核可以理解为提取特征的算子。

它的大小选取主要根据你的输入输出的维度进行选取。比较常用的大小是1x1、3x3。

使用比较小的卷积核的时候可能无法表示其特征，如果采用较大的卷积核则会导致复杂度极大的增加。多个小的卷积核叠加使用要远比一个大的卷积核单独使用效果要好的多，这样堆叠可以减少参数量。

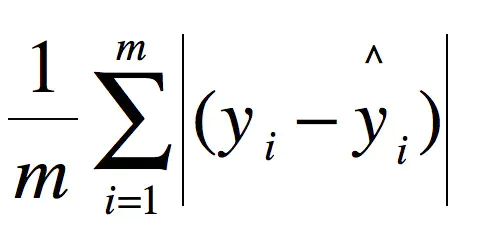
但在此过程中会引入激活函数，所以模型的计算复杂性也会有所改变，具有更多的非线性后，提高了模型的表达能力，模型的拟合效果会更好。

**A.常见的损失函数有哪些，介绍下它们的特点、应用。**

损失函数也叫作目标函数，是我们模型训练时给模型优化的一个指标，一个方向。

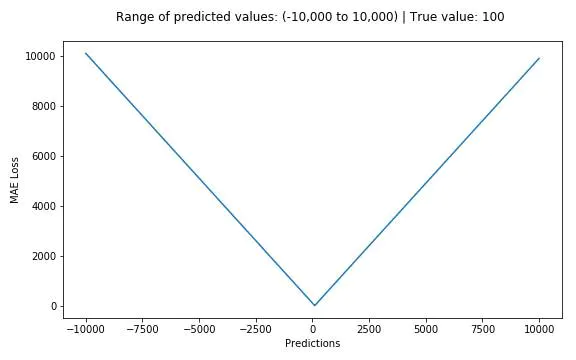
1. MAE（Mean Absolute Error）

MAE也叫作平均绝对值误差，也是用于回归的损失函数。它是通过计算预测值和真实值之间的绝对值之和来当做计算方法。取值范围是0到正无穷。



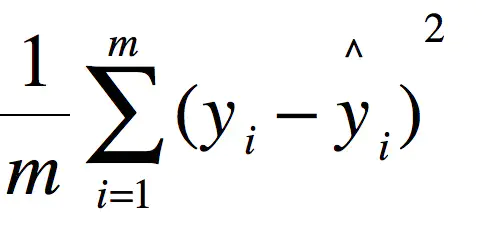
MAE也叫作L1 loss。一般来说，L1正则会制造稀疏的特征，大部分没有用的特征的权重都会被置零。

MAE不会那么容易受异常点的影响，因为损失和误差之间是线性关系，而因为平方放大的关系，下面的MSE则会更加容易受到异常点的影响。但是由于MAE的梯度都保持为1，所以这是很不利于模型训练。



2. MSE（Mean Squared Error）

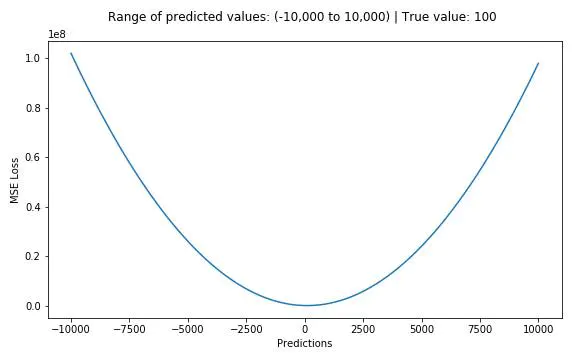
MSE也叫作均方误差，通过求预测值和真实值之间距离的平方和来当做计算方法，大部分回归问题都使用MSE。



取值范围是从0 到正无穷。MSE也是常用的回归损失函数。

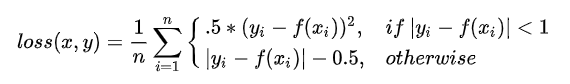
另外，MSE也叫L2 loss。L2正则会不会直接让权重置零，而是平滑地缩小值，使特征权重比较平均。

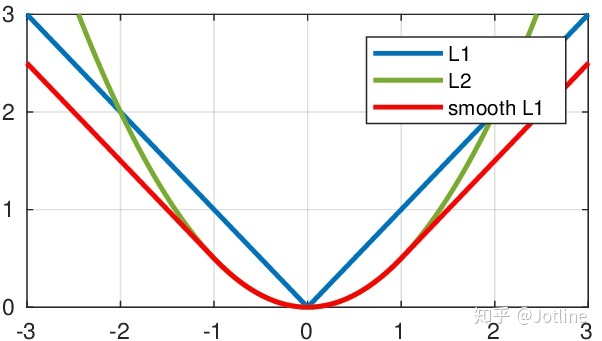
因为MSE每处的梯度都不同，所以一般来说MSE会比MAE更快地收敛。



3. Smooth L1 loss

和名字一样，是平滑的意思。它的函数公式如下：





当预测值和真实值之间差距较小的时候，使用L2 loss；而当差别比较大的时候就用L1 loss的平移函数。因此，可以说Smooth L1 loss是L1 loss和L2 loss的结合体。因此：

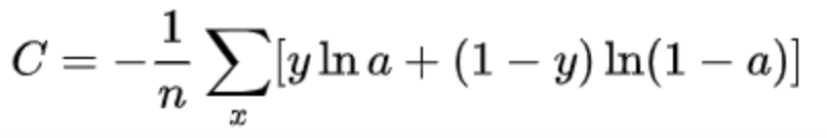
当预测值和真实值过大时，梯度值比较小，较为稳定不容易发生梯度爆炸。

当预测值和真实值过小的时候，梯度不会太大。

相比于L1 loss，可以收敛得更快。

相比于L2 loss，对离群点、异常值不敏感，梯度变化相对更小，训练时不容易跑飞。

4. CE（CrossEntropyLoss）



CrossEntropyLoss是交叉熵损失函数，可以用于二分类也可以用于多分类，经常和softmax一起使用，因为说将两者放在一起算起来会快，数值也更加稳定，一个作为激活函数，一个作为损失函数。softmax把分类输出标准化成概率分布，输出的结果哪个值大，就属于哪个值。cross-entropy（交叉熵）刻画预测分类和真实结果之间的相似度。交叉熵越小，两个概率分布越相似。

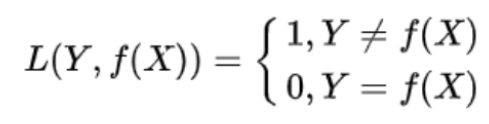
5. BCE Loss和BCEWithLogitsLoss

BCE Loss和BCEWithLogitsLoss可用于单标签的二分类任务或者多标签的二分类任务，对每一个值进行sigmoid运算，输出的值为0-1。当多个类别同时计算时，每个类别之间是相互独立的，所以全部之和不一定为1。

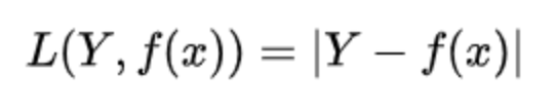
BCEWithLogitsLoss = Sigmoid+BCELoss，当网络最后一层使用nn.Sigmoid时，就用BCELoss，当网络最后一层不使用nn.Sigmoid时，就用BCEWithLogitsLoss。另外，BCEWithLogitsLoss是将Sigmoid()和BCELoss()整合起来，比 纯粹使用BCELoss()+Sigmoid()更数值稳定。

另外，还有其他的损失函数：

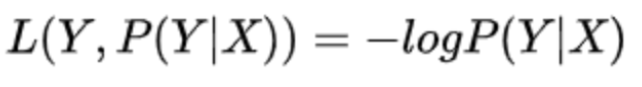
0-1损失函数



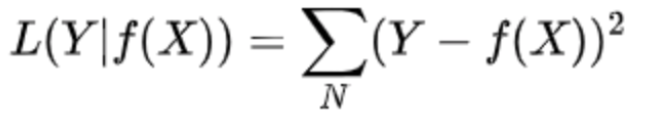
绝对值损失函数



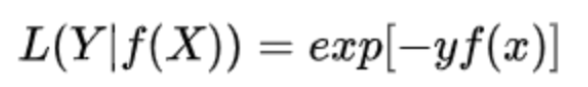
log对数损失函数



平方损失函数



指数损失函数



hinge损失函数

