# 深度学习理论与实践练习题006

友情提示：

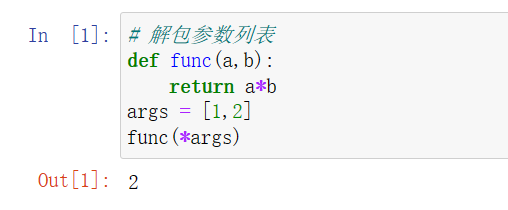
1. 编程题可用pycharm或vscode编辑、调试，也可用其他方式，形式不限；
2. 编程题附上代码及结果截图；
3. 深度学习理论部分题目，可手写拍照；

## Python

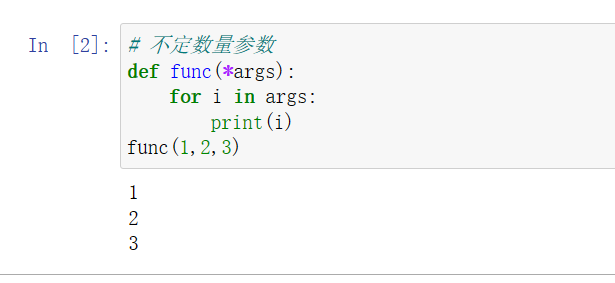
1. 单星（\*）和双星（\*\*）运算符的作用，并举例说明?

单行（\*）作用：

1. 用在函数调用或列表/元组前面，解包可迭代对象，将其中的元素作为独立的参数传递给函数或用于构建列表/元组

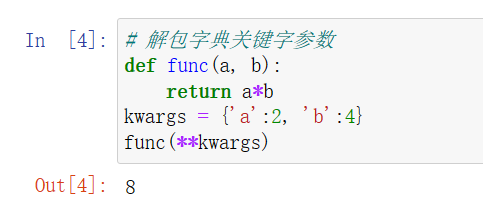


1. 用在函数定义上，，用于接受不定数量的参数，这些参数被收集为一个元组

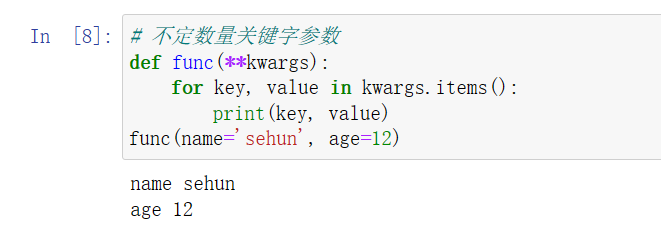


双星（\*\*）作用：

1. 用在函数调用或字典前面，解包字典，其键值对作为关键字参数传递给函数或用于构建新的字典



1. 用在函数定义时，用于接受不定数量的关键字参数

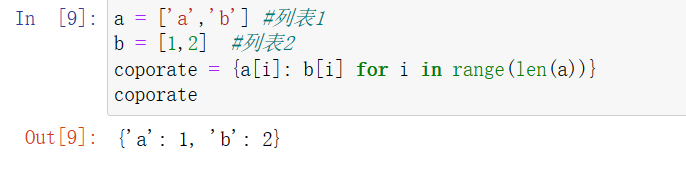


1. 如何将两个列表合并成一个字典，形式如下：

a = ['a','b'] #列表1

b = [1,2] #列表2

# 合并后：{'a':1,'b':2}；



## 深度学习框架

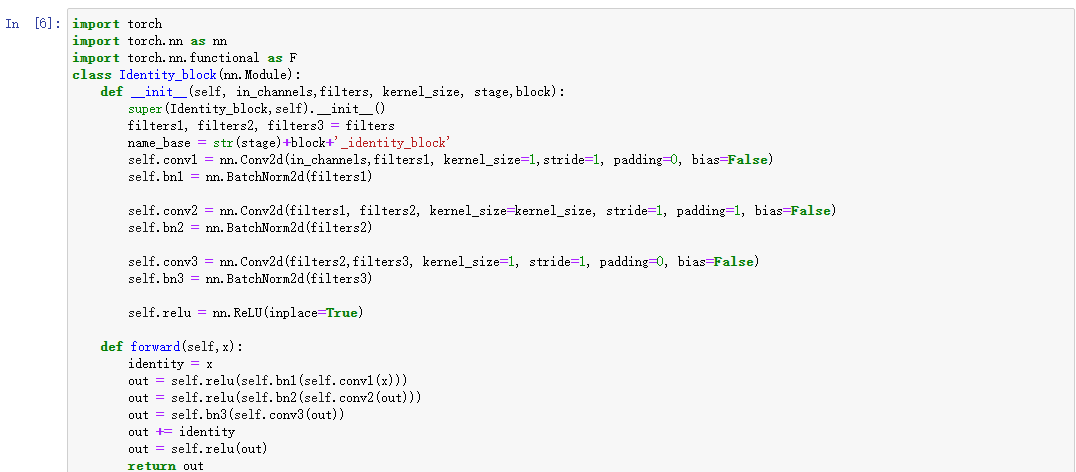
1. Pytorch中保存训练过程日志；

Pytorch中可以使用标准模块logging来记录训练过程中的信息：损失值、准确率等，如代码所示：



1. 基于flower数据集，使用pytorch实现resnet网络；

Resnet50框架







## 深度学习理论知识点

1. 分类损失函数有哪些及其原理；
2. 均方差损失Mean Squared Error Loss

原理：预测数据和原始数据对应点误差的平方和的均值。公式如下：

 ，其中N为样本个数

1. 平均绝对误差损失 Mean Absolute Error Loss

原理：衡量的是预测值与真实值之间距离的平方误差幅度。公式如下：



优点：收敛速度快，能够对梯度给予合适的惩罚权重，使梯度更新的方向可以更加精确。

缺点：对异常值十份敏感，梯度更新的方向很容易受到离群点所主导，不具备鲁棒性。

1. Huber Loss（能增强MSE函数对离群点的鲁棒性）

Huber Loss 定义如下



δ 是 HuberLoss 的参数，y是真实值， 是模型的预测值

当预测偏差小于 δ 时，它采用平方误差,

当预测偏差大于 δ 时，采用的线性误差。

相比于最小二乘的线性回归，HuberLoss降低了对离群点的惩罚程度，所以 HuberLoss 是一种常用的鲁棒的回归损失函数。

1. 分位数损失Quantile Loss

原理：对不同分位数的预测误差赋予不用的权重。公式如下：



1. 交叉熵损失Cross Entropy Loss（经常用于分类问题，需要计算各类别的概率，所以交叉熵损失函数都是与sigmoid函数或者softmax函数成对出现）

二分类的交叉熵损失函数：



其中 表示类别为1， 表示预测类别为1的概率。

多分类的交叉熵损失函数：



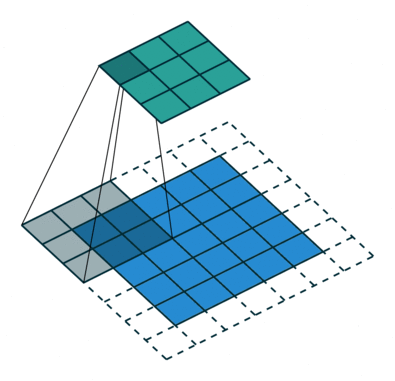
表示类别有n个。

详情可见网址：https://mp.weixin.qq.com/s/\_mrnK\_KDYJrgmoTc6di0NA

1. 卷积有哪些类型并逐一说明；

常规卷积、空洞卷积、多尺度卷积、分组卷积、可分离卷积、形变卷积、转置卷积

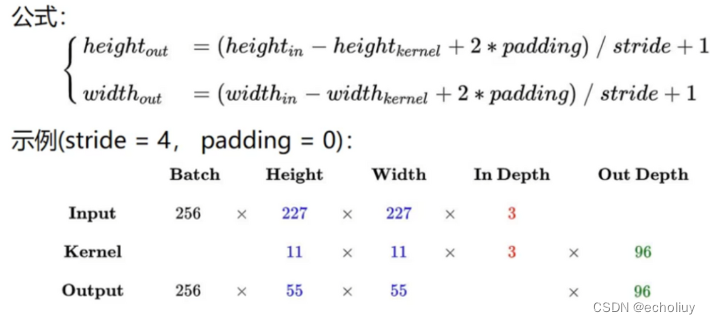
1. 常规卷积



计算方式：\*\*卷积核在输入特征图上滑动并计算内积

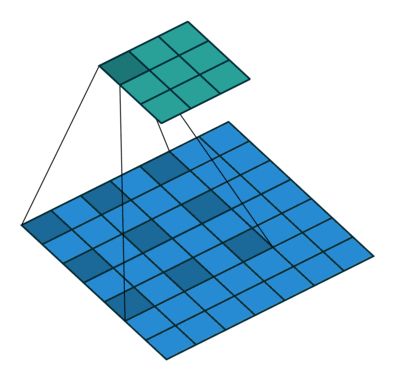
\*\*超参数：\*\*kernel size：卷积核尺寸大小；padding：输入填充的圈数；stride：卷积核滑动步长。

输出尺寸计算公式：



1. 空洞卷积

空洞卷积或者膨胀卷积，是在标准的卷积核中注入空洞，以此来增加模型的感受野。如下图，使用3内核进行2D卷积，扩展率为2且无填充



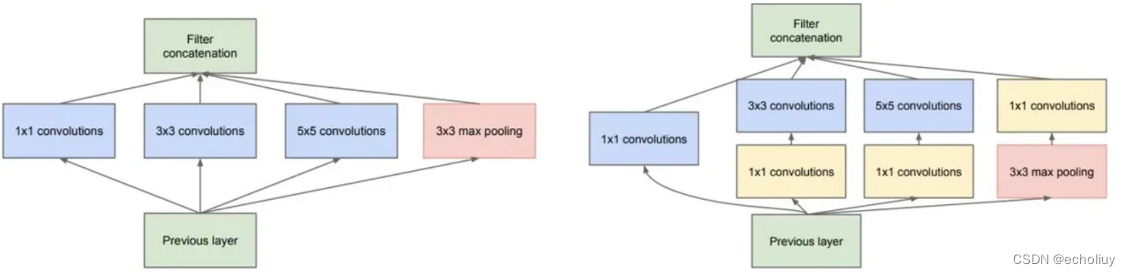
目的：常规卷积感受野太小。感受野是指特征图上某个点代表输入图像尺寸的大小，网络的浅层是小感受野，捕获小尺寸目标，网络深层是大感受野，捕获大尺寸目标。

计算方式：在卷积核元素之间注入空洞，然后进行常规卷积运算。

超参数：空洞率r(dilation rate)，即卷积核元素之间插入的r-1个空洞

与常规卷积的关系：K = k + (k-1)(r-1)，k为常规卷积核的大小，K为空洞卷积核的大小，r为空洞率。

1. 多尺度卷积



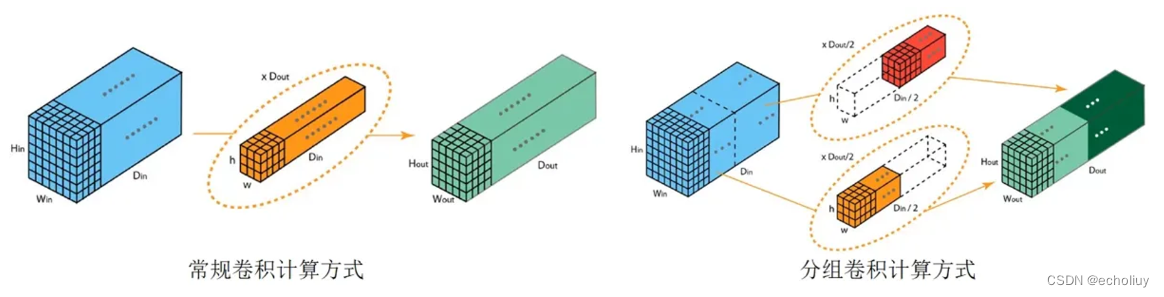
目的：输入图像中目标的尺寸是多变的。

计算方式：用多分支卷积来提取不同感受野的特征

缺点：上图左图计算量太大，改进后的右图插入1×1的卷积降通道，减少了计算量。但手工痕迹太重，SKNet可以自适应的决定哪个分支的信息更重要。

用途：Inception系列

1. 分组卷积



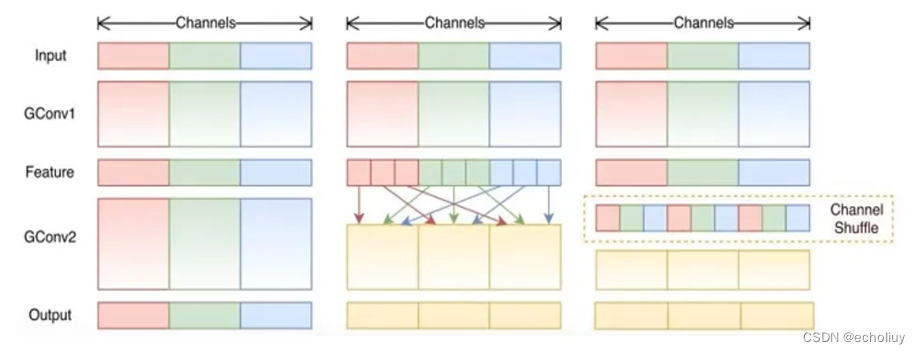
目的：降低参数量和计算量

计算方式：将输入特征图的channel进行分组，然后每个组内部进行卷积，最终将结果concate拼接。

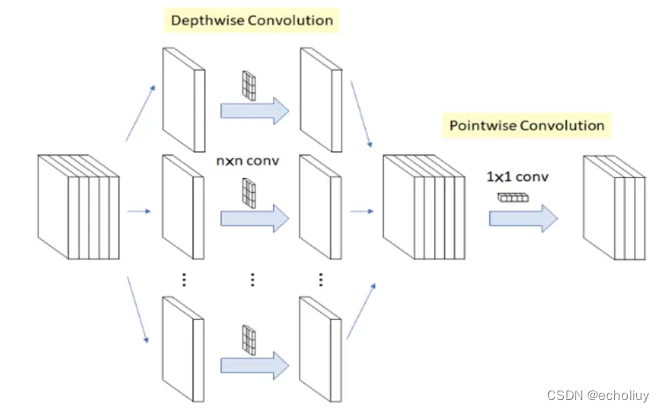
存在问题：信息流通不通畅，如下图的ShuffleNet网络结构：

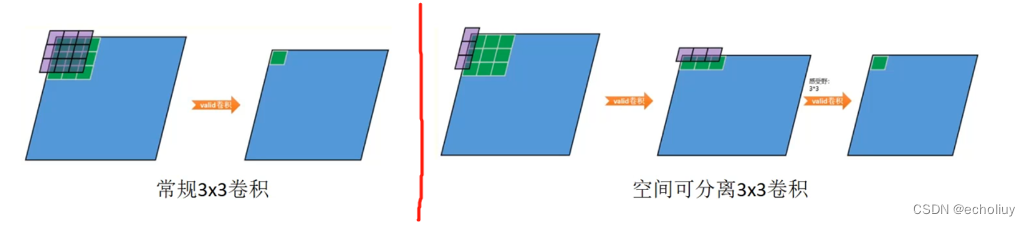
用途：ResNeXt，shuffleNet

解决办法：通道混洗(切分后互相糅杂)



1. 可分离卷积





解决问题：神经网络参数多，计算量大.

计算方式：将原n×n的卷积，分开计算变为1×n和n×1两步。即逐层卷积(Depthwise Convolution) + 逐点卷积(Pointwise Convolution)

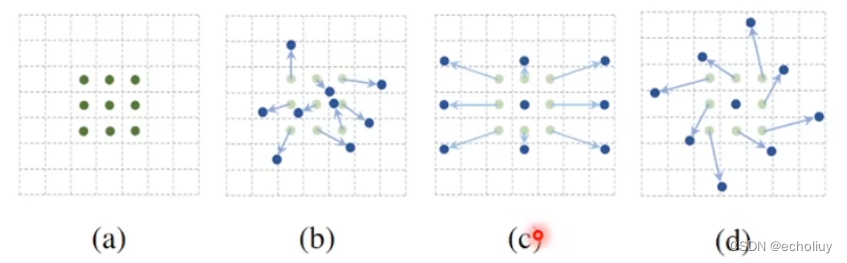
Depthwise Convolution：对输入特征图的每一个通道只是用一个卷积核。

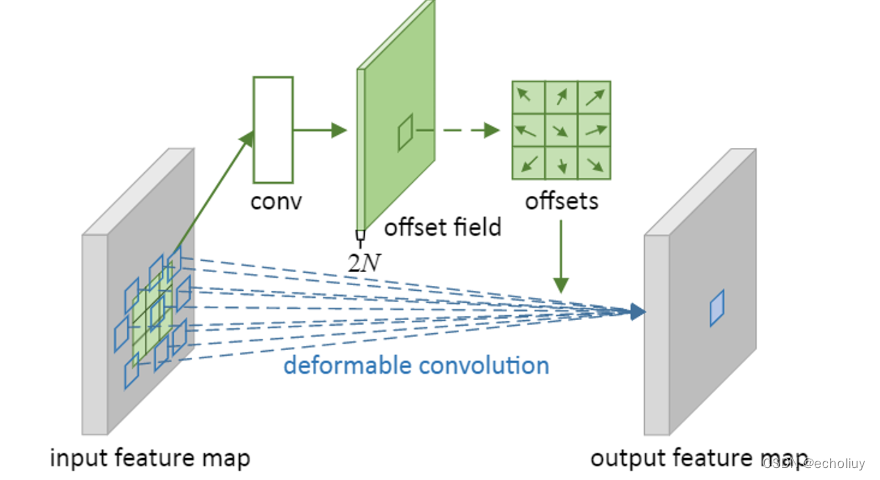
Pointwise Convolution：常规的1×1卷积，目的是跨通道融合。

用途：MobileNet

缺点：用在网络深层是还好，浅层时对精度损失过大。

1. 形变卷积（DCN）->扩大感受野





解决问题：应用于目标检测场景，一个待检测的目标可能有各种形状，用常规的卷积是很难把这些形状信息捕获到的。

计算方式：

学习采样点的偏移，根据偏移量计算，得到新的特征图。

在正常计算卷积过程中，增加一个分支，用来去学习偏移量。首先经过一个卷积层，得到一个偏移特征图，shape和原始特征图是一样的，但是通道是2N，2代表原始特征图上的每个点是有2个维度可以移动的，N代表卷积核的尺寸。相当于对每一个卷积核的像素点学习对应输入特征图的偏移量。接着拿到当前卷积核对应的偏移量，映射到原图上进行像素偏移，进行正常卷积得到结果即可。

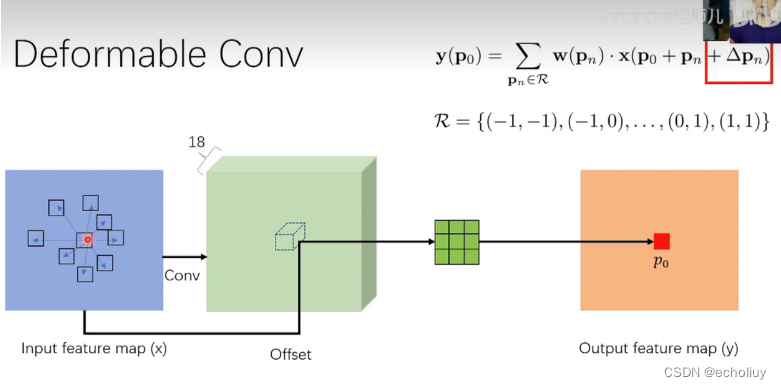
缺点：计算量大，且无法tensorRT加速。

注意：形变卷积不是对卷积核的形变，而是对卷积核对应的采样点的偏移。

用途：目标检测应用其模块如：DCN v1，DCN v2。

具体细节：

offset=18表示九个偏移的x,y值，共18个数



1. 转置卷积

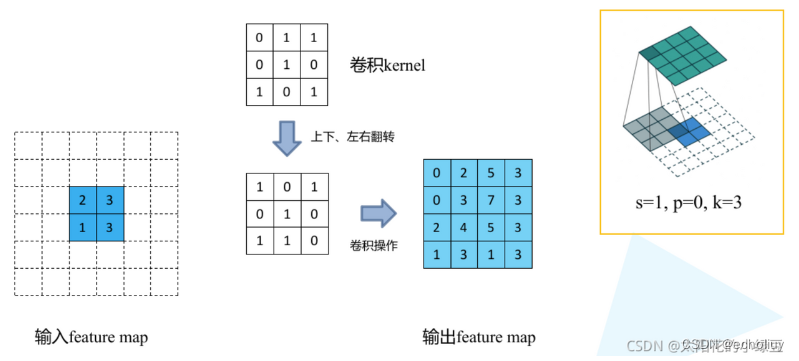
转置卷积刚刚说了，主要作用就是起到上采样的作用。但转置卷积不是卷积的逆运算（一般卷积操作是不可逆的），它只能恢复到原来的大小（shape）数值与原来不同。

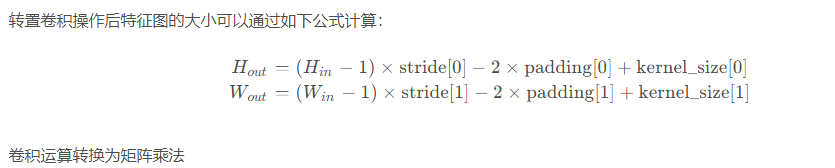
在输入特征图元素间填充s-1行、列0（其中s表示转置卷积的步距）

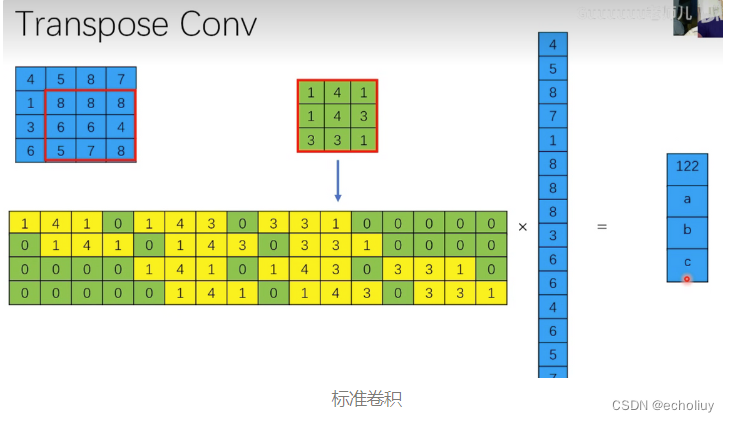
在输入特征图四周填充k-p-1行、列0（其中k表示转置卷积的kernel\_size大小，p为转置卷积的padding，注意这里的padding和卷积操作中有些不同）

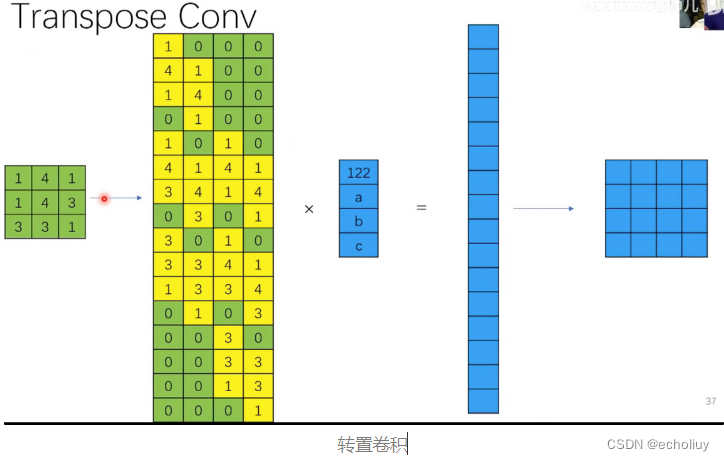
将卷积核参数上下、左右翻转

做正常卷积运算（填充0，步距1）









链接：

<https://blog.csdn.net/echoliuy/article/details/130707080>

https://blog.csdn.net/yohnyang/article/details/129237840