

VGG模型

深度与层数

从a到e有不同的层数，深度越大参数量越大

卷积层

1. 卷积核为 3×3 ，步长为1，有效的捕捉前后左右中的信息，两层3卷积核==5感受野，三层==7感受野，而且因为经过了三次激活函数，比直接用7卷积核的非线性表达能力好
2. 1卷积核的加入：用于增加非线性特征，改变通道数（特征维度），不变化图像大小

训练细节

1. 从多尺度的图像中裁剪不同的crop，使用优化器，正则化技巧：权重衰减（求导），dropout层，学习率分三次下降，先大步后小步
2. 网络初始化：进行预处理，使数据不位于激活函数的饱和区，在饱和区时梯度非常小，得不到信息
3. 预训练：在小模型a中预训练，利用a得到的参数设置初始参数
4. 调整图像大小

测试细节

1. 全连接层改为卷积层，任意尺寸都可以了



4. 图像处理，从转，调整纵横比等
5. multi crop方法，single crop即裁剪小图

4. 部署在多个gpu上，数据并行，把所有数据分到不同gpu中

1. init函数

1. 特征提取器：由多个卷积层和池化层构成
2. 平均池化层：采用平均池化的技术对图像降低像素
3. 分类器：自己定义的，类似于多个全连接层，由sequential函数串联起来的
 1. 一个全连接层，将大小为 $512 \times 7 \times 7$ 的特征映射到大小为4096的向量
 2. 一个ReLU激活函数
 3. 一个dropout层，用于防止过拟合。
 4. 一个全连接层，将4096维的向量映射到4096维的向量
 5. 一个ReLU激活函数
 6. 一个dropout层
 7. 最后一个全连接层，将4096维的向量映射到输出类别的数量

dropout层

Dropout是一种常用的正则化技术，用于减少神经网络模型的过拟合，各种防止过拟合的方法都可以叫正则化，Dropout层在训练期间以一定的概率随机丢弃网络中的一些神经元，使得网络在每次训练迭代中都在不同的子网络上训练，从而减少了神经元之间的依赖关系，提高了网络的泛化能力

具体来说，对于每个训练样本，在前向传播过程中，Dropout会以一定的概率丢弃每个神经元，并将其输出设为0。而在反向传播时，只有那些没有被丢弃的神经元才会更新其权重。这样，Dropout使得网络不能过度依赖于任何一个神经元，从而减少了过拟合的可能性。

Dropout 为什么能减少过拟合？

- A. 模型集成 $p=0.5$ 意味着 2^n 个共享权重的潜在
- B. 记忆随机抹去 不再死记硬背
- C. Dropout: 减少神经元之间的联合依赖性
每个神经元都被逼着独当一面
- D. 有性繁殖 每个基因片段都要与来自另一个随机个体的基因片段协同工作
- E. 数据增强 总可以找到一个图片 使神经网络中间层结果与Dropout后相同 相当于增加了这张图片的数据
- F. 稀疏性

2. make_layer函数

1. 配置参数 cfg: 字典内容中的数字代表每个卷积层内的卷积通道数，M表示最大池化层，不同的cfg参数决定不同的vgg版本，我写的这个为vgg11
2. 列表+=的方法与append大同小异，+=每一次生成的列表会被视为一个新的列表而不是原列表上加内容
3. 代码逻辑：如果遇到M则生成最大池化层，如果不是则生成对应卷积层，并通过手动修改是否使用bn层来判断是否加入bn层，并在每次生成后更新in_channels，