链滴

CV 岗位精选面试题（11-20）

作者：julyedu
原文链接：https://ld246.com/article/1622111804719
来源网站：链滴

<h2 id="添加微信-julyedufu77-回复--11---领取最新升级版-名企AI面试100题-电子书--">添加微信：julyedufu77，回复“11”，领取最新升级版《名企 AI 面试 100 题》电子书！！</h2>
<h2 id="11-当参数量----样本量时候--神经网络是如何预防过拟合-">11、当参数量 >> 样本时候, 神经网络是如何预防过拟合?</h2>

正则化 2. Early Stopping 3. Dropout 4. 数据增强

过拟合即在训练误差很小,而泛化误差很大，神经网络时避免过拟合的方法：
1、正则化
正则化的思想十分简单明了。由于模型过拟合极有可能是因为我们的模型过于复杂。因此，我们要让我们的模型在训练的时候，在对损失函数进行最小化的同时，也需要让对参数添加限制，这个限也就是正则化惩罚项。
假设我们模型的损失函数为：

加入正则项 L 后，损失函数为：

常用的正则化有两种：L1 正则和 L2 正则
L1 正则表达式：

其中 w 代表模型的参数，k 代表模型参数的个数。
L2 正则表达式：

其中 w 代表模型的参数，k 代表模型参数的个数。
L1 正则与 L2 正则的思想就是不能够一味的去减小损失函数，你还得考虑到模型的复杂性，通过制参数的大小，来限制其产生较为简单的模型，这样就可以降低产生过拟合的风险。
它们的区别在于 L1 更容易得到稀疏解。为什么呢？我们先看看一个直观的例子：

假设我们模型只有 w1,w2 两个参数，上图中左图中黑色的正方形是 L1 正则项的等值线，而彩的圆圈是模型损失的等值线；右图中黑色圆圈是 L2 正则项的等值线，彩色圆圈是同样模型损失的等线。因为我们引入正则项之后，我们要在模型损失和正则化损失之间折中，因此我们去的点是正则项失的等值线和模型损失的等值线相交处。通过上图我们可以观察到，使用 L1 正则项时，两者相交点在坐标轴上，也就是 w1,w2 中常会出现 0；而 L2 正则项与等值线常相交于象限内，也即为 w1,w2 非 0。因此 L1 正则项时更容易得到稀疏解的。
而使用 L1 正则项的另一个好处是：由于 L1 正则项求解参数时更容易得到稀疏解，也就意味着出的参数中含有 0 较多。因此它自动帮你选择了模型所需要的特征。L1 正则化的学习方式是一种嵌式特征学习方式，它选取特征和模型训练时一起进行的。
<h2 id="12-什么是感受野-">12、什么是感受野？</h2>
某一层特征图中的一个 cell,对应到原始输入的响应的大小区域。
什么是感受野
感受野(Receptive Field)，指的是神经网络中神经元“看到的”输入区域，在卷积神经网络中，feature map 上某个元素的计算受输入图像上某个区域的影响，这个区域即该元素的感受野。
卷积神经网络中，越深层的神经元看到的输入区域越大，如下图所示，kernel size 均为 3×3，stride 均为 1，绿色标记的是 Layer2 每个神经元看到的区域，黄色标记的是 Layer3 看到的区域，具体，Layer2 每个神经元可看到 Layer1 上 3×3 大小的区域，Layer3 每个神经元看到 Layer2 上 3×3 大

的区域，该区域可以又看到 Layer1 上 5×5 大小的区域。
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2021/05/150a7bb3bfa84b9e918b2664e98d9a28.jpeg?imageView2/2/interlace/1/format/jpg">
所以，感受野是个相对概念，某层 feature map 上的元素看到前面不同层上的区域范围是不同，通常在不特殊指定的情况下，感受野指的是看到输入图像上的区域。
<h2 id="13-简述你对CBIR-Content-based-Image-Retrieval基于内容的图像检索-的理解">13、述你对 CBIR(Content-based Image Retrieval 基于内容的图像检索)的理解</h2>
通过对比特征点\特征值的相似度,判断两个图片是否相近
基于内容的图像检索
基于内容的图像检索(CBIR, Content Based Image Retrieval)是相对成熟的技术领域，在工业界有广泛的应用场景，如搜索引擎（Google、百度）的以图搜图功能，各电商网站（淘宝、Amazon、bay）的相似商品搜索，社交平台（Pinterest）的相似内容推荐等。
基于内容的图像检索流程
图像内容检索流程与文本检索流程类似，但二者信息表征方法不同。文本通过词频计算 BoW 来征一段文本内容，而图像则使用视觉特征来表示。Google 团队 2003 年[1]提出的视频内容检索方法鉴文本检索流程，使用局部特征构建视觉词袋向量(Bag-of-Visual-Words，BoVW), 也称 BoF(Bag-of Features), 来表示图像。这里的视觉单词是指量化后的视觉特征。Video-Google[1]中检索系统也分构建词库、构建索引和检索三部分。下图是视觉词库构建流程:
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2021/05/b3851a4224254347b6442f8350ec4134.jpeg?imageView2/2/interlace/1/format/jpg">
对图像提取若干个局部特征描述子，如 SIFT，对这些描述子进行量化。量化器通常通过聚类得：对特征描述子集合进行 k-means 聚类，聚类后得到的 k 个质心即为视觉单词。描述子 desc 的量结果 q(desc)为与 desc 最相近的质心的索引。所有质心构成了视觉词表。图像中的特征单词的词频成了该图像的向量描述 BoVW。假设视觉词表中的单词个数为 N，那么 BoVW 向量的长度为 N，向中的元素为对应单词出现在该图像中的频次或者采用采用 td-idf 权重更新向量中每个元素值。
基于得到的视觉词库，计算所有图像(或视频中帧)数据的 BoVW 向量。检索进程启动时，将目标据库中所有图像的 BoVW 向量构建索引。输入一副检索图像，提取该图像的 BoVW 特征，然后与目库向量进行距离比对，查找近邻向量。 最直观的查找方法是蛮力查找即将查询向量 q 与所有的 BoVW 向量进行距离计算。这种穷举方式对大数据集或高维向量的查找效率非常低。为改进这个问题，Video Google[1]提出采用倒排文件 IVF 结构进行索引构建，IVF 索引结构如下图所示。图中 i 表示每个视单词。
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2021/05/27d1c04c865d425bbcb27deb5888f8c1.jpeg?imageView2/2/interlace/1/format/jpg">
由于词向量通常是很稀疏的，我们无需遍历目标库中的所有文件，因而可以通过建立倒排文件，每个单词构建一个列表，列表中是所有包含当前单词的图像 meta 信息。检索时，只需要计算那些与前查询图像包含相同单词的图像的 BoVW 向量间的距离即可，即通过减小搜索范围来降低搜索复杂。
Video-Google 提供了经典的基于内容的图像检索流程，核心技术可以总结为两点：特征提取和邻查找。后续图像检索基于大多基于此思想，针对不同业务场景下的数据特点，对涉及的特征提取和邻查找技术进行优化，最终目标是提取能够高效表征图像的特征向量，进行快速视觉内容查找。
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2021/05/ae03fac552354aeb874da5c9e20b45a1.jpeg?imageView2/2/interlace/1/format/jpg">
更多内容可以查看：赵丽丽：基于内容的图像检索技术：从特征到检索
【TPAMI 重磅综述】 SIFT 与 CNN 的碰撞：万字长文回顾图像检索任务十年探索历程（上）：L LI：【TPAMI 重磅综述】 SIFT 与 CNN 的碰撞：万字长文回顾图像检索任务十年探索历程（上）
论文 SIFT Meets CNN: A Decade Survey of Instance Retrieval：
https://arxiv.org/pdf/1608.01807.pdf
<h2 id="14-什么是计算机视觉单词模型-">14、什么是计算机视觉单词模型？</h2>

参考 1：提取图像特征，然后用 KMeans 聚类得到视觉词汇表，用词汇表描述衣服图像。
参考 2：
引言
最初的 Bag of words，也叫做“词袋”，在信息检索中，Bag of words model 假定对于一个本，忽略其词序和语法，句法，将其仅仅看做是一个词集合，或者说是词的一个组合，文本中每个词出现都是独立的，不依赖于其他词 是否出现，或者说当这篇文章的作者在任意一个位置选择一个词汇不受前面句子的影响而独立选择的。
Bag-of-words 模型是信息检索领域常用的文档表示方法。在信息检索中，BOW 模型假定对于个文档，忽略它的单词顺序和语法、句法等要素，将其仅仅看作是若干个词汇的集合，文档中每个单的出现都是独立的，不依赖于其它单词是否出现。也就是说，文档中任意一个位置出现的任何单词，不受该文档语意影响而独立选择的。例如有如下两个文档：
1：Bob likes to play basketball, Jim likes too.
2：Bob also likes to play football games.
基于这两个文本文档，构造一个词典：
Dictionary = {1:”Bob”, 2. “like”, 3. “to”, 4. “play”, 5. “basketball”, 6. “also”, 7. “football”, 8. “games”, 9. “Jim”, 10. “too”}
这个词典一共包含 10 个不同的单词，利用词典的索引号，上面两个文档每一个都可以用一个 10 维向量表示（用整数数字 0~n（n 为正整数）表示某个单词在文档中出现的次数）：
1：[1, 2, 1, 1, 1, 0, 0, 0, 1, 1]
2：[1, 1, 1, 1 ,0, 1, 1, 1, 0, 0]
向量中每个元素表示词典中相关元素在文档中出现的次数(下文中，将用单词的直方图表示)。不，在构造文档向量的过程中可以看到，我们并没有表达单词在原来句子中出现的次序（这是本 Bag-of words 模型的缺点之一，不过瑕不掩瑜甚至在此处无关紧要）。
一、原理
考虑将 Bag-of-words 模型应用于图像表示。为了表示一幅图像，我们可以将图像看作文档，即干个“视觉词汇”的集合，同样的，视觉词汇相互之间没有顺序。

图 1 将 Bag-of-words 模型应用于图像表示
由于图像中的词汇不像文本文档中的那样是现成的，我们需要首先从图像中提取出相互独立的视词汇，这通常需要经过三个步骤：
（1）特征提取
（2）特征表示
（3）单词本的生成
通过观察会发现，同一类目标的不同实例之间虽然存在差异，但我们仍然可以找到它们之间的一共同的地方，比如说人脸，虽然说不同人的脸差别比较大，但眼睛，嘴，鼻子等一些比较细小的部位却观察不到太大差别，我们可以把这些不同实例之间共同的部位提取出来，作为识别这一类目标的视词汇。
而 SIFT 算法是提取图像中局部不变特征的应用最广泛的算法，因此我们可以用 SIFT 算法从图像提取不变特征点，作为视觉词汇，并构造单词表，用单词表中的单词表示一幅图像。
接下来，我们通过上述图像展示如何通过 Bag-of-words 模型，将图像表示成数值向量。现在有个目标类，分别是人脸、自行车和吉他。
第一步：利用 SIFT 算法，从每类图像中提取视觉词汇，将所有的视觉词汇集合在一起，如下图 2 所示：

第二步：利用 K-Means 算法构造单词表。K-Means 算法是一种基于样本间相似性度量的间接类方法，此算法以 K 为参数，把 N 个对象分为 K 个簇，以使簇内具有较高的相似度，而簇间相似度低。SIFT 提取的视觉词汇向量之间根据距离的远近，可以利用 K-Means 算法将词义相近的词汇合并作为单词表中的基础词汇，假定我们将 K 设为 4，那么单词表的构造过程如下图 3 所示：

第三步：利用单词表的中词汇表示图像。利用 SIFT 算法，可以从每幅图像中提取很多个特征点这些特征点都可以用单词表中的单词近似代替，通过统计单词表中每个单词在图像中出现的次数，可将图像表示成为一个 K=4 维数值向量。
图 3 单词本的生成
上图中，我们从人脸、自行车和吉他三个目标类图像中提取出的不同视觉词汇，而构造的词汇表，会把词义相近的视觉词汇合并为同一类，经过合并，词汇表中只包含了四个视觉单词，分别按索引标记为 1，2，3，4。通过观察可以看到，它们分别属于自行车、人脸、吉他、人脸类。
<h2 id="15-简述什么是Local-Feature-局部特征算子--">15、简述什么是 Local Feature(局部特征子)？</h2>
局部图像算子是图像特征的局部表达，它反映了图像上具有的局部特性，适合于对图像进行匹配检索等应用。
视觉单词 BOVW 也是一种 Local Feature。
1.局部算子分类：
1）基于分布的算子；
2）空间频率技术；
3）微分算子：
2.局部特征建立依赖的空间
1）归一化的 Laplacian 尺度空间
<ol start="2">
Difference of Gaussian

<ol start="3">
局部区域检测算法

1）Harris points 旋转不变量 特征点周围 41×41 像素区域 大小固定
2）Harris-Laplace regions 旋转和尺度不变量 检测角点结构特征
3）Hessian-Laplace regions 旋转和尺度不变量 特征点是由 Hessian 决定的空间极大值和 Laplacian-of-Gaussian.尺度空间极大值，与 DoG 检测近似，但是在尺度空间能获得更高的准确度，并且尺度选择上的准确度也高于 Harris-Laplace 。检测的准确性影响算子的执行力。
4）Harris-Affine regions 仿射不变量 由 Harris-Laplace 检测子检测位置和尺度，附近的仿射基于二次动差矩阵的 affine adaptation 程序决定
5）Hessian-Affine regions 仿射不变量 由 Hessian-Laplace 检测子检测位置和尺度，附近的射由 affine adaptation 程序决定
<ol start="4">
局部区域描述子

1） SIFT 描述子 是一个 3D 梯度位置方向直方图，位置被量化到 4×4 局部栅格，梯度角度分为 8 个方向，算子为 4×4×8＝128 维
2）Gradient location-orientation histogram (GLOH)，GLOH 是 SIFT 描述子的一种延伸，为增强其鲁棒性和独立性。以对数极坐标在半径方向建立三个带（6，11，15）和 8 个角度方向，形成 7 个位置带，中心带在半径方向不分块。梯度方向量化为 16 个带，形成 272 维矢量，利用 PCA 降维/p>
3）Shape context 与 SIFT 描述子相似，但是基于边缘 Shape context 是一个边缘点位置和方的 3D 直方图，以对数极坐标在半径方向建立三个带（6，11，15）和 4 个角度方向，生成 36 维描子
4）Geometric histogram 在一个区域内描述边缘分布直方图
5）PCA-SIFT 描述子 以特征点周围 39×39 像素块形成 3024 维矢量，用 PCA 降维 36 维
6）Spin image 是一个量化像素位置和强度的直方图 ，在 5 个圆环中计算 10 个强度带，生成 50 维算子
7）Steerable filters and differential invariants 使用与高斯卷积后的导数
8）Complex filters
<ol start="9">

Moment invariants
Cross correlation

6.匹配方法：
基于阈值的匹配
基于最近邻匹配：如果 DB 是 DA 的最近邻区域，且之间的距离小于阈值则区域 A 与区域 B 是配的
基于次最近距离与最近距离之比
<ol start="7">
描述子维数影响

低维算子：steerable filters ,complex filters, differential invariants
基于微分的算子，导数的阶数影响着算子的维数，对于 steerable filters 三阶导数和四阶导数都保持算子的独立性，并且导数的阶数对算子匹配的准确度影响显而易见，但是对 complex filters 和 differential invariants 影响较小。并且 steerable filters 计算到四阶导数时效果比 differential invariants 效果好。
高维算子：GLOH，PCA-SIFT，cross correlation 算子 维数过高与过低效果都不理想。对于 GLH 算子，128 维匹配效果高于 40 维和 272 维，对于 PCA-SIFT36 维效果好于 20 维和 100 维，对于ross correlation 则 81 维匹配效果好于 36 维和 400 维。
8.对不同图像变换的适应性
1）仿射变换。 利用 Hessian Affine 和 Harris Affine 检测特征点，然后对不同的局部算子测试效果最好的是 SIFT 算子。并且利用 Hessian Affine 比 Harris Affine 的效果好，因为基于拉普拉斯尺度选择与 Hessian 算子相结合可以获得更准确的结果。
2）尺度变换 大多算子表现良好
3）旋转变
<h2 id="16-KD-Tree相比KNN来进行快速图像特征比对的好处在哪里-">16、KD-Tree 相比 KNN 进行快速图像特征比对的好处在哪里?</h2>
参考解析 1：极大的节约了时间成本．点线距离如果大于最小点，无需回溯上一层，如果小于最点,则再上一层寻找。
参考解析 2：

什么是 KNN

1.1 KNN 的通俗解释
何谓 K 近邻算法，即 K-Nearest Neighbor algorithm，简称 KNN 算法，单从名字来猜想，可简单粗暴的认为是：K 个最近的邻居，当 K=1 时，算法便成了最近邻算法，即寻找最近的那个邻居。/p>
用官方的话来说，所谓 K 近邻算法，即是给定一个训练数据集，对新的输入实例，在训练数据集找到与该实例最邻近的 K 个实例（也就是上面所说的 K 个邻居），这 K 个实例的多数属于某个类，把该输入实例分类到这个类中。

如上图所示，有两类不同的样本数据，分别用蓝色的小正方形和红色的小三角形表示，而图正中的那个绿色的圆所标示的数据则是待分类的数据。也就是说，现在，我们不知道中间那个绿色的数据从属于哪一类（蓝色小正方形 or 红色小三角形），KNN 就是解决这个问题的。
如果 K=3，绿色圆点的最近的 3 个邻居是 2 个红色小三角形和 1 个蓝色小正方形，少数从属于数，基于统计的方法，判定绿色的这个待分类点属于红色的三角形一类。
如果 K=5，绿色圆点的最近的 5 个邻居是 2 个红色三角形和 3 个蓝色的正方形，还是少数从属多数，基于统计的方法，判定绿色的这个待分类点属于蓝色的正方形一类。
于此我们看到，当无法判定当前待分类点是从属于已知分类中的哪一类时，我们可以依据统计学理论看它所处的位置特征，衡量它周围邻居的权重，而把它归为(或分配)到权重更大的那一类。这就是K 近邻算法的核心思想。
我们看到，K 近邻算法的核心在于找到实例点的邻居，这个时候，问题就接踵而至了，如何找到

居，邻居的判定标准是什么，用什么来度量。这一系列问题便是下面要讲的距离度量表示法。
<h2 id="17-简述encode和decode思想">17、简述 encode 和 decode 思想</h2>
解析 1：
Encoder-Decoder(编码-解码)是深度学习中非常常见的一个模型框架，比如无监督算法的 auto-ncoding 就是用编码-解码的结构设计并训练的；比如 image caption 的应用，就是 CNN-RNN 的码-解码框架；再比如神经网络机器翻译 NMT 模型，往往就是 LSTM-LSTM 的编码-解码框架。因此准确的说，Encoder-Decoder 并不是一个具体的模型，而是一类框架。Encoder 和 Decoder 部分以是任意的文字，语音，图像，视频数据，模型可以采用 CNN，RNN，BiRNN、LSTM、GRU 等等所以基于 Encoder-Decoder，我们可以设计出各种各样的应用算法。
Encoder-Decoder 框架有一个最显著的特征就是它是一个 End-to-End 学习的算法；本文将以本-文本的例子作为介绍，这样的模型往往用在机器翻译中，比如将法语翻译成英语。这样的模型也叫做 Sequence to Sequence learning[1]。所谓编码，就是将输入序列转化成一个固定长度的向量解码，就是将之前生成的固定向量再转化成输出序列。
解析 2：
一、Encoder-Decoder（编码-解码）
介绍
Encoder-Decoder 是一个模型构架，是一类算法统称，并不是特指某一个具体的算法，在这个架下可以使用不同的算法来解决不同的任务。首先，编码（encode）由一个编码器将输入序列转化一个固定维度的稠密向量，解码（decode）阶段将这个激活状态生成目标译文。
回顾一下，算法设计的基本思路：将现实问题转化为一类可优化或者可求解的数学问题，利用相的算法来实现这一数学问题的求解，然后再应用到现实问题中，从而解决了现实问题。（比如，我们解决一个词性标注的任务（现实问题），我们转化成一个 BIO 序列标注问题（数学模型），然后设计系列的算法进行求解，如果解决了这个数学模型，从而也就解决了词性标注的任务）。
Encoder：编码器，如下:

Decoder：解码器，如下:

合并起来，如下：

更具体一点的表达如下所示：

说明：

不论输入和输出的长度是什么，中间的“向量 c”长度都是固定的（这是它的缺陷所在）。
根据不同的任务可以选择不同的编码器和解码器（例如，CNN、RNN、LSTM、GRU 等）
Encoder-Decoder 的一个显著特征就是：它是一个 end-to-end 的学习算法。
只要符合这种框架结构的模型都可以统称为 Encoder-Decoder 模型。

在图像中的应用：比如 SegNet 图像分割算法

<h2 id="18-输入图片尺寸不匹配CNN网络input时候的解决方式--三种以上-">18、输入图片尺寸

匹配 CNN 网络 input 时候的解决方式？（三种以上）</h2>
解析：

two-fixed 方法:直接对输入图片 Resize 缩放；
one-fixed 方法: 固定一边，缩放另一条边；
free 方法:去掉 FC 全连接层加入全局池化层，或者使用卷积层替换全连接层；

网络之所以要输入固定大小的图片，主要是因为网络中存在 FC 全连接层，而且全连接层的一个点是参数量大容易导致过拟合，关于这部分解释说明可以参考第 9 题“如果最后一个卷积层和第一个连接层参数量太大怎么办?”
卷积层替换全连接层
在经典分类网络，比如 LeNet、AlexNet 中，在前面的卷积层提取特征之后都串联全连接层来做类。目前很多网络比如 YOLO 系列、SSD 以及 Faster RCNN 的 RPN，MTCNN 中的 PNet 等都使卷积层来代替全连接层，一样可以做到目标分类的效果，而且具有以下优点：

更灵活，不需要限定输入图像的分辨率；
更高效，只需要做一次前向计算；

全连接层和卷积层只要设置好了对应的参数，可以在达到相同输入输出的效果，在这个意义上，数学上可以认为它们是可以相互替换的。
将全连接操作转化成卷积操作，也就是卷积最后一层的 feature map 如果使用卷积操作是将每神经元 Flatten 之后 dense 连接到后面的若干神经元，以 AlexNet 为例，最后一层为 256x7x7，得后面的 4096 个神经元，但是如果使用 7X7 的卷积核对前面的 FeatureMap 进行继续卷积(padding=)，不也可以得到 4096X1X1 的向量吗，如果图片大一些，例如 384x384，那没 AlexNet 最后一层大小就是 256X12X12 经过一个 7x7 的卷积核之后就是 4096x6x6 了，这时候这 6x6=36 个神经元有了位置信息。如下图所示:
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2021/05/6cfaaba121054ca8ae1da8d7cf6eb587.jpeg?imageView2/2/interlace/1/format/jpg">
卷积替代全连接的优点:

对输入分辨率的限制

如果网络后面有全连接层，而全连接层的输入神经元个数就是固定的，那么反推上层卷积层的输是固定的，继续反推可知输入网络的图片的分辨率是固定的。例如，LetNet 由于由全连接层，输入只能是 28 x 28 的。
如果网络中的全连接层都用卷积层替代，网络中只有卷积层，那么网络的输出分辨率是随着输入片的分辨率而来的，输出图中每一个像素点都对应着输入图片的一个区域（可以用 stride,pooling 来算）。
<ol start="2">
计算效率比较

同样以 LeNet 来做例子，如果一个图片是 280 x 280 的分辨率，为了识别图片中所有的数字（了简单，假设每个数字都是在这个大图划分为 10 x 10 的网格中），那么为了识别这 100 个位置数，那么至少需要做 100 次前向；而全卷积网络的特点就在于输入和输出都是二维的图像，并且输入和出具有相对应的空间结构，我们可以将网络的输出看作是一张 heat-map，用热度来代表待检测的原位置出现目标的概率，只做一次前向就可以得到所有位置的分类概率。
FCN(Fully Convolutional Networks for Semantic Segmentation)论文链接：
https://arxiv.org/pdf/1411.4038.pdf
<h2 id="19-FCN与CNN最大的区别-">19、FCN 与 CNN 最大的区别？</h2>
FCN 中用卷积层替换了 CNN 中的全连接层

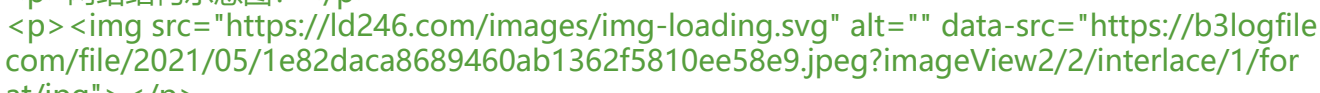
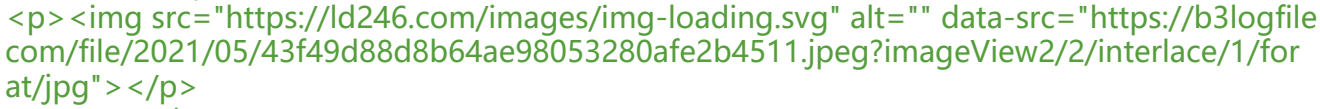
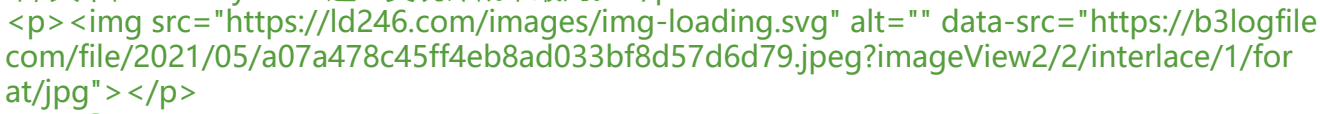
FCN 概述

CNN 做图像分类甚至做目标检测的效果已经被证明并广泛应用，图像语义分割本质上也可以认是稠密的目标识别（需要预测每个像素点的类别）。
传统的基于 CNN 的语义分割方法是：将像素周围一个小区域（如 25*25）作为 CNN 输入，做练和预测。这样做有 3 个问题：

像素区域的大小如何确定；
存储及计算量非常大；
像素区域的大小限制了感受野的大小，从而只能提取一些局部特征；

2.FCN 原理及网络结构
一句话概括原理：FCN 将传统卷积网络后面的全连接层换成了卷积层，这样网络输出不再是类而是 heatmap；同时为了解决因为卷积和池化对图像尺寸的影响，提出使用上采样的方式恢复。核思想：

不含全连接层(fc)的全卷积(fully conv)网络，可适应任意尺寸输入。
增大数据尺寸的反卷积(deconv)层，能够输出精细的结果。
结合不同深度层结果的跳级(skip)结构，同时确保鲁棒性和精确性。

网络结构示意图：

网络结构详图：输入可为任意尺寸图像彩色图像；输出与输入尺寸相同，深度为：20 类目标 + 景=21 类。

3 CNN 与 FCN
CNN
通常 CNN 网络在卷积层之后会接上若干个全连接层, 将卷积层产生的特征图(feature map)映射一个固定长度的特征向量。以 AlexNet 为代表的经典 CNN 结构适合于图像级的分类和回归任务，因它们最后都期望得到整个输入图像的一个数值描述（概率），比如 AlexNet 的 ImageNet 模型输出个 1000 维的向量表示输入图像属于每一类的概率(softmax 归一化)。
比如：下图中的猫, 输入 AlexNet, 得到一个长为 1000 的输出向量, 表示输入图像属于每一类的率, 其中在“tabby cat”这一类统计概率最高。

FCN
FCN 对图像进行像素级的分类，从而解决了语义级别的图像分割（semantic segmentation）题。与经典的 CNN 在卷积层之后使用全连接层得到固定长度的特征向量进行分类（全联接层 + softax 输出）不同，FCN 可以接受任意尺寸的输入图像，采用反卷积层对最后一个卷积层的 feature map 进行上采样, 使它恢复到输入图像相同的尺寸，从而可以对每个像素都产生了一个预测, 同时保留了原输入图像中的空间信息, 然后在上采样的特征图上进行逐像素分类。
最后逐个像素计算 softmax 分类的损失, 相当于每一个像素对应一个训练样本。下图是 Longjon 用于语义分割所采用的全卷积网络(FCN)的结构示意图：

简单的来说，FCN 与 CNN 的区域在把于 CNN 最后的全连接层换成卷积层，输出的是一张已经abel 好的图片：
这些抽象的特征对分类很有帮助，可以很好地判断出一幅图像中包含什么类别的物体，但是因为失了一些物体的细节，不能很好地给出物体的具体轮廓、指出每个像素具体属于哪个物体，因此做到

确的分割就很有难度。
基于 CNN 的分割方法与 FCN 的比较
传统的基于 CNN 的分割方法：为了对一个像素分类，使用该像素周围的一个图像块作为 CNN 输入用于训练和预测。这种方法有几个缺点：

存储开销很大。例如对每个像素使用的图像块的大小为 15x15，然后不断滑动窗口，每次滑动的口给 CNN 进行判别分类，因此则所需的存储空间根据滑动窗口的次数和大小急剧上升。
计算效率低下。相邻的像素块基本上是重复的，针对每个像素块逐个计算卷积，这种计算也有很程度上的重复。
像素块大小的限制了感知区域的大小。通常像素块的大小比整幅图像的大小小很多，只能提取一局部的特征，从而导致分类的性能受到限制。

而全卷积网络(FCN)则是从抽象的特征中恢复出每个像素所属的类别。即从图像级别的分类进一延伸到像素级别的分类。
更多内容参考：语义分割--全卷积网络 FCN 详解 - 我的明天不是梦 - 博客园
<h2 id="20-遇到class-imbalanced-data-数据类目不平衡-问题怎么办-">20、遇到 class-imbalanc d data（数据类目不平衡）问题怎么办？</h2>
解析 1:
设置不同类的权重 weighted loss
batch-wise balanced sampling(循环平衡采样)
解析 2：
当你在对一个类别不均衡的数据集进行分类时得到了 90% 的准确度（Accuracy）。当你进一步析发现，数据集的 90% 的样本是属于同一个类，并且分类器将所有的样本都分类为该类。在这种情下，显然该分类器是无效的。并且这种无效是由于训练集中类别不均衡而导致的。
首先举 2 个关于类别不均衡的例子：

在一个二分类问题中，训练集中 class 1 的样本数比 class 2 的样本数是 60:1。使用逻辑回归进分类，最后结果是其忽略了 class 2，即其将所有的训练样本都分类为 class 1。
在分类任务的数据集中，有三个类别，分别为 A，B，C。在训练集中，A 类的样本占 70%，B 的样本占 25%，C 类的样本占 5%。最后我的分类器对类 A 的样本过拟合了，而对其它两个类别的本欠拟合。

什么是类别不均衡问题：
类别数据不均衡是分类任务中一个典型的存在的问题。简而言之，即数据集中，每个类别下的样数目相差很大。例如，在一个二分类问题中，共有 100 个样本（100 行数据，每一行数据为一个样的表征），其中 80 个样本属于 class 1，其余的 20 个样本属于 class 2，class 1:class2=80:20=4:1 这便属于类别不均衡。当然，类别不均衡问同样会发生在多分类任务中。它们的解决方法是一样的。此，为了便于讨论与理解，我们从二分类任务入手进行讲解。
类别不均衡问题是现实中很常见的问题
大部分分类任务中，各类别下的数据个数基本上不可能完全相等，但是一点点差异是不会产生任影响与问题的。
在现实中有很多类别不均衡问题，它是常见的，并且也是合理的，符合人们期望的。如，在欺诈易识别中，属于欺诈交易的应该是很少部分，即绝大部分交易是正常的，只有极少部分的交易属于欺交易。这就是一个正常的类别不均衡问题。又如，在客户流失的数据集中，绝大部分的客户是会继续受其服务的（非流失对象），只有极少数部分的客户不会再继续享受其服务（流失对象）。一般而已如果类别不平衡比例超过 4:1，那么其分类器会大大地因为数据不平衡性而无法满足分类要求的。因在构建分类模型之前，需要对分类不均衡性问题进行处理。
在前面，我们使用准确度这个指标来评价分类质量，可以看出，在类别不均衡时，准确度这个评指标并不能 work。因为分类器将所有的样本都分类到大类下面时，该指标值仍然会很高。即，该分器偏向了大类这个类别的数据。
<h2 id="添加微信-julyedufu77-回复--11---领取最新升级版-名企AI面试100题-电子书---">添加微：julyedufu77，回复“11”，领取最新升级版《名企 AI 面试 100 题》电子书！！</h2>