**一、图像处理基础理论**

1. **图像降噪的常用方法**

降噪：y = x + e，y是你观察到的带噪音的图像，e是噪音，x是干净无噪音的图像。只已知y，外加e的概率分布，降噪问题需要你去寻找最接近真实值的x。

高斯滤波、中值滤波

1. **图像增强的常用方法**

为了改善图像的视觉效果，提高图像的清晰度，突出某些感兴趣的特征，抑制不感兴趣的特征，空间域增强、频域增强

1. **举例几种边缘检测算子**

一阶的有Roberts Cross算子，Prewitt算子，Sobel算子，Canny算子，Krisch算子，罗盘算子，二阶有拉普拉斯算子，Sobel算子是典型的基于一阶导数的边缘检测算子

1. **常见的插值算法**

最邻近插值算法也叫做零阶插值算法，主要原理是让输出像素的像素值等于邻域内离它距离最近的像素值，计算简单但易出现像素不联系，出现锯齿状；双线性二次插值；双线性三次插值

1. **饱和度、对比度、清晰度等概念说明**

饱和度指色彩的纯洁性，也叫饱和度或彩度，一种颜色的饱和度越高，它就越鲜艳；反之，一种颜色的饱和度越低，它就越接近于灰色

对比度指的是一幅图像中明暗区域**最亮的白和最暗的黑**之间不同亮度层级的测量，差异范围越大代表对比越大，差异范围越小代表对比越小

清晰度指影像上各细部影纹及其边界的清晰程度

1. **Hog特征是什么**

方向梯度直方图（Histogram of Oriented Gradient, HOG）特征是一种在计算机视觉和图像处理中用来进行物体检测的特征描述子。它通过计算和统计图像局部区域的梯度方向直方图来构成特征

1. **腐蚀、膨胀、锐化等概念**

膨胀就是求局部最大值的操作，腐蚀就是求局部最小值的操作

图像锐化*(image sharpening)*是补偿图像的轮廓，增强图像的边缘及灰度跳变的部分，使图像变得清晰

**二、深度学习图像视觉**

**1.1x1卷积的作用**

卷积核越大，它生成的 featuremap 上单个节点的感受野就越大，随着网络深度的增加，越靠后的 featuremap 上的节点感受野也越大。因此特征也越来越抽象，不增加感受野的情况下，让网络加深，为的就是引入更多的非线性，升降维度，全连接层等效作用

**2.depthwise卷积的概念**

不同于常规卷积操作，Depthwise Convolution的一个卷积核负责一个通道，一个通道只被一个卷积核卷积，无法扩展特征图维度，因此需要与Pointwise Convolution配合使用，可起到参数缩减的作用

**3.反卷积的概念及实现原理**

反卷积：反卷积也被称为转置卷积，反卷积操作并不能还原出卷积之前的图片，只能还原出卷积之前图片的尺寸，在GAN中使用较多

**4.3D卷积和2D卷积的区别**

2D ConvNets在每次卷积运算之后就会丢失输入信号的时间信息，3D卷积才能保留输入信号的时间信息，从而产生输出卷

**5.卷积核大小选取的原则**

选择卷积核大小的原则是：小而深。可达到大的卷积核的效果但参数会减少，会有更好的泛化能力

当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，我们应该选择多个相对小的卷积核来进行叠加卷积

**6.卷积层减少参数的方法**

使用池化操作；使用堆叠的小卷积核代替大卷积核；使用深度可分离卷积，将原始的K × K × C的卷积核分成K × K × 1和1 × 1 × C两部分操作；应用1 × 1卷积

**7.pool层如何传递导数**

因为Pooling操作使得feature map的尺寸变化，这使得梯度无法对位的进行传播下去，但只**需要保证传递的loss（或者梯度）总和不变，因此**mean pooling和max pooling的反向传播也是不同的

mean pooling的前向传播就是把一个patch中的值求取平均来做pooling，那么反向传播的过程也就是把某个元素的梯度等分为n份分配给前一层，这样就保证池化前后的梯度（残差）之和保持不变

max pooling的前向传播是把patch中最大的值传递给后一层，而其他像素的值直接被舍弃掉。那么反向传播也就是 把梯度直接传给前一层某一个像素，而其他像素不接受梯度，也就是为0，max pooling操作和mean pooling操作不同点在于需要记录下池化操作时到底哪个像素的值是最大，也就是max id ，这个变量就是记录最大值所在位置的，因为在反向传播中要用到

无论max pooling还是mean pooling，都没有需要学习的参数。因此，在卷积神经网络的训练中，Pooling层需要做的仅仅是将误差项传递到上一层，而没有梯度的计算

**8.CNN里池化的作用**

将特征图下采样，对感受域内的特征进行筛选，提取区域内最具代表性的特征，能够有效地降低输出特征尺度，进而减少模型所需要的参数量，更关注是否存在某些特征而不是特征具体的位置，防止过拟合或有可能会带来欠拟合

**9.CNN反向传播的过程细节**

**10.梯度消失，梯度爆炸的原因，以及如何解决**

如果网络使用sigmoid激活函数，误差在向前传递的时候，经过sigmod单元，需要乘sigmod的梯度，而sigmoid的梯度最大是0.25，因此越向前传递，误差就越小了，这就是梯度消散

误差在经过全连接或者卷积层时，也要乘以权重w，如果w都比较大，大过sigmod造成的减小，这样越往前误差就越来越大，梯度爆炸了

**11.FPN结构原理**

将深层特征图进行上采样与浅层特征图相融合，其背后的思路是为了获得一个强语义信息，这样可以提高检测性能，将处理过的低层特征和处理过的高层特征进行累加，这样做的目的是因为低层特征可以提供更加准确的位置信息，而多次的降采样和上采样操作使得深层网络的定位信息存在误差，因此我们将其结合其起来使用，这样我们就构建了一个更深的特征金字塔，融合了多层特征信息，并在不同的特征进行输出

**低层的特征语义信息比较少，但是目标位置准确；高层的特征语义信息比较丰富，但是目标位置比较粗略**

**12.怎么增加感受野，怎么计算感受野**

感受野是卷积神经网络(CNN)每一层输出的特征图(feature map)上的像素点在原始输入图像上映射的区域大小，感受野的计算从最深层开始向浅层计算，逐步计算到第一层

第一层卷积层的输出特征图像素的感受野大小等于滤波器大小，深层卷积层的感受野大小和它之前所有层的滤波器大小和步长有关系，计算感受野大小时，忽略了图像边缘的影响，即不考虑padding的大小，RF = ((RF-1)\*stride) + f\_size

**13.CNN权值共享的意思是什么**

其实权值共享这个词说全了就是整张图片在使用同一个卷积核内的参数，比如一个3\*3\*1的卷积核，这个卷积核内9个的参数被整张图共享，而不会因为图像内位置的不同而改变卷积核内的权系数, 当然CNN中每一层不会只有一个卷积核, 权值共享后意味着每一个卷积核只能提取到一种特征（即每一个卷积核只能提取到到一个特征

**14.BN层的作用，以及在训练时和测试时如何使用，有何不同**

现存问题：各层参数需要不断适应新的输入数据分布，降低学习速度。（学习率不能大，训练速度缓慢），深层输入的变化可能趋向于变大或者变小，导致浅层落入饱和区，使得学习过早停止。（容易梯度饱和、消失。）每层的更新都会影响到其它层，因此每层的参数更新策略需要尽可能谨慎地选择。（调参困难）

**BN就是通过一定的规范化手段，把每层神经网络任意神经元这个激活输入值的分布强行拉回到均值为0方差为1的标准正态分布**

BN层位于激活函数之前，它先对数据进行标准化，在进行缩放和平移, 标准化是防止数据达到饱和（即激活值分布在大部分都接近0或1），这样会使数据对激活函数不敏感, BN层就是为了让让每一层的值在一个有效范围内传递下去

**15.模型压缩方法、模型量化、模型蒸馏等概念**

**16.如何计算卷积层的复杂度和参数量**

**17.求解网络的FLOPs**

**18.k、p、s、input size 之间的关系**

**19.BP神经网络反向传播推导**

**20.前向传播、反向传播代码演示**

**21.GN、BN、LN的区别**

独立同分布的数据可以简化常规机器学习模型的训练、提升机器学习模型的预测能力

Batch Norm就是在深度神经网络训练过程中使得每一层神经网络的输入保持相同分布

白化包含两个目的：去除特征之间的相关性 —> 独立；使得所有特征具有相同的均值和方差 —> **同分布**

BN特别依赖**batch Size，**与**BN仅针对单个神经元不同，LN综合考虑整个一层的信息**

Batch Norm：batch方向做归一化，算NHW的均值；

Layer Norm：channel方向做归一化，算CHW的均值；

Instance Norm：一个channel内做归一化，算H\*W的均值；

Group Norm：将channel方向分group，然后每个group内做归一化，算(C//G)HW的均值；

**22.softmax里面的数值上下溢出问题**

**23.二元交叉熵和多元交叉熵损失函数原理**

线性回归解决的是回归问题，预测值是实数范围，逻辑回归则相反，解决的是分类问题，预测值是[0,1]范围。所以逻辑回归名为回归，实为分类

分类标签为0、1，预测结果在[0,1]之间，即样本为 0 或 1 的概率g(x)=1 if g(x)>=0.5 else g(x)=0

描述两种分布之间的差异程度，就需要引入交叉熵函数

同时考虑两种概率分布的熵叫做：交叉熵（cross-entropy)

用KL（相对熵）散度来衡量这两个分布的差异，KL散度=交叉熵-真实分布的信息熵，而这个真实分布的信息熵在数据集确定之后就确定下来的了，这一项就可以当成一个常数项，所以我们如果想让KL散度越小，只需要让交叉熵越小越好了，因此就可以直接将逻辑回归的损失函数直接定义为交叉熵。

交叉熵函数的曲面是非常陡峭的，在模型效果差的时候学习速度比较快，是非常有利于梯度下降的迭代的，所以逻辑回归里面使用交叉熵作为损失函数而不是使用均方误差作为损失函数

**24.dropout的原理及使用方法**

**25.神经网络的权重初始化方法**

**26.BN层为什么能够加速模型收敛**

神经网络学习过程本质就是为了学习数据分布，一旦训练数据与测试数据的分布不同，那么网络的泛化能力也大大降低；另外一方面，一旦每批训练数据的分布各不相同(batch 梯度下降)，那么网络就要在每次迭代都去学习适应不同的分布，这样将会大大降低网络的训练速度

**27.正则化和标准化的概念及实现方法**

**28.注意力机制的理解Attention**

**29.模型评价指标有哪些**

**30.AP、mAP 的计算过程**

**31.为什么分类问题用交叉熵损失，如何得来**

交叉熵是同时考虑两种概率分布的熵，相对熵才是描述两种概率分布的差异，但是因为相对熵=交叉熵-真实信息熵，而真实信息熵为常数，所以要想相对熵小，就要求交叉熵小，因此才称交叉熵是'两种概率分布的差异’，但实际上它不是，只不过有等效作用而已

**32.逻辑回归中损失函数的实际意义**

**33.sigmoid的优缺点**

**34.为什么relu用的比较多**

**35.SGD每一步在干什么，及原理**

**36.adam的优缺点及原理**

**37.正则化的本质，以及为什么要使用正则化**

范数是一个向量空间或矩阵上所有向量的长度和大小的求和

**38.L1和L2正则化的区别**

L1正则化是指权值向量W中各个元素的绝对值之和，通常表示为||w||1->稀疏矩阵->特征选择

L1正则化可以产生稀疏权值矩阵，即产生一个稀疏模型，可以用于特征选择

稀疏矩阵指的是很多元素为0，只有少数元素是非零值的矩阵，即线性回归模型的大部分**系数**都是0

L1正则化就是在loss function后边所加正则项为L1范数，加上L1范数容易得到稀疏解（0比较多）, **L1 正则化的引入就是为了完成特征自动选择的光荣使命，它会学习地去掉这些无用的特征，也就是把这些特征对应的权重置为 0。**

L2正则化是指权值向量W中各个元素的**平方和然后再求平方根**

L2正则化可以防止模型过拟合（overfitting）, L2正则存在的意义就是加速权重衰减的速度，也就是加大了对权重的惩罚

**39.过拟合、欠拟合的原因及解决方法**

**40.数据样本不平衡的处理方法**

**41.yolo中的正负样本及需求比例**

**42.anchor free检测算法**

**43.ROI pooling和Allign的细节**

**44.Focal loss的概念**

**45.NMS算法过程**

**46.RPN算法过程**

**47.使用3x3卷积核最大的优势是什么**

**48.卷积的局限是什么**

**49.工程上如何对卷积操作进行优化**

**50.auc是什么**

**51.列举常用经典网络及其特征**