模式识别与深度学习实验6报告

|  |  |  |  |
| --- | --- | --- | --- |
| 实验结果的正确性(60%) |  | 表达能力(10%) |  |
| 实验过程的规范性(10%) |  | 实验报告(20%) |  |
| 加分(5%) |  | 总成绩(100%) |  |

实验报告

一、实验目的（介绍实验目的）

|  |
| --- |
| 本实验使用完成对象识别与定位任务。即输入一张图片，输出对应图片，其中标注所包含的对象以及每个对象的位置。可以将本任务拆分成两个任务：对象定位和对象识别。  对象识别问题可以通过网络达到很好的效果，该任务的难点在于找到图片中存在对象的区域（对象定位）。针对该问题，最简单的方法即遍历图片中所有可能的位置，地毯式地搜索不同大小、不同位置、不同长宽比的区域，进而从其中找到最优可能存在对象的区域，但是该方法查找效率太低，不具有实用性。  本次实验基于系列论文，实现了网络（网络参数加载现有参数，因需要在大量数据集上进行比较长时间的训练才能达到好的效果），巧妙解决了对象定位问题。 |

二、实验环境（介绍实验使用的硬件设备、软件系统、开发工具等）

|  |
| --- |
| 操作系统：家庭中文版；  版本：；  安装与版本管理：使用维护多虚拟环境；  深度学习加速： + ；  安装：配置第三方源； |

三、实验过程（介绍实验过程、设计方案、实现方法、实验结果等）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 相关工作 之前的物体检测系统使用分类器来完成物体检测任务。为了检测一个对象，这些检测系统要在一张图片的不同位置、不同尺寸、不同长宽比的边界框中使用分类器来判断该区域是否存在某一对象。例如，系统要使用一个滑动窗口在图片上滑动，进而利用分类器判断滑动窗口汇总是否存在物体。  提出了候选区的方法解决对象定位问题。即先从图片中搜索出一些可能存在对象的**候选区**，大概个左右，然后使用分类器对每个候选区的对象进行识别，之后再对边界框的位置进行微调，相比于大幅度提升了对象识别和定位的效率。但是执行起来还是比较慢的，因为这些环节分开训练，检测性能难以进行优化。 论文阅读与学习 系列有很多论文，是从逐步改进得到的，因此该节分别对，，进行解读。 **YOLO1** 是系列的第一个版本。即，只需处理一次图像就可以完成对象识别与定位两个任务。该论文将检测问题转换成回归问题，输入的图像经过一个神经网络可以获得一些特征，进而利用这些特征得到对象边框和该边框中的对象属于特定类别的概率。相比于其他的神经网络，突出的优点就是快（本次实验没有考虑检测的时间问题），此外，因为相比于其他检测系统能学到更加抽象的物体的特征，使得其可以从真实图像领域迁移到其他领域；缺点是在物体定位时更容易出错，但是因为查看图片会捕获图片的全部信息，因此基本不会在背景上检测出不存在的物体（即将背景看作待检测对象）。  系统大致分为三个步骤，如图一所示：   1. 将输入图像调整为； 2. 在图像上运行单个卷积神经网络； 3. 根据模型的置信度对得到的检测进行阈值化；   的想法很简单，如图二所示：将图像输入一个单独的网络，预测出对象的边界框，以及这个边界框中的对象属于某个类别的概率。使用一整幅图像进行训练，同时可以直接优化检测性能。    图一、系统总体流程  中使用统一检测，即使用一个神经网络就可以检测出所有对象。该网络使用图像中的全部特征来预测每一个边界框。  的设计使得可以进行端到端的训练、可以保证实时处理速度，同时可以保证较高的准确率。  检测系统首先将输入图像划分为个栅格，图二中将输入图像划分为的栅格。如果**一个对象的中心落到某个栅格内，则该栅格（对应的特征映射）负责检测该对象**。每个栅格预测个边界框，同时预测这些边界框的得分。得分反映的是模型对该边界框中存在对象的可能性大小的预测。论文中将该得分定义为：  如果该边界框中不存在对象（没有人工标记的对象落在一个栅格中），则为，则得分也是；如果该边界框中存在对象（有人工标记的对象落在一个栅格中），则得分是真实边界框和预测边界框之间的。  针对边框的预测值由个预测值组成，分别是。其中，表示边界框的中心相当于当前栅格左上角位置的偏移量（偏移量一定位于，具体解释见部分）。是边界框的宽度和高度（实际训练中，宽度和高度被归一化到，具体解释见部分），是上文提到的数值大小。每个栅格还要预测个类条件概率概率：。即在该栅格中包含一个对象的前提下，该对象是某个类别的概率。（针对每个栅格而言）综上所述，预测结果编码成的张量。  在测试阶段，需要将每个栅格的类条件概率和每个边界框的相乘，即计算。该乘积结果既包含了边界框中预测的类别的概率信息（因为参与计算），也反映了边界框中是否含有对象以及边界框的准确度（因为参与计算）。  方法的缺陷：尽管每个栅格预测个边界框，但是只能选择值最大的边界框作为对象检测的输出。因此每个栅格最多输出一个对象，该方法能够检测对象的最大数量受限于划分的栅格的数量。此外，当对象占画面比例比较小的时候，比如图片中有鸟群，每个栅格只能检测出其中的一个，将导致很多较小的对象不能被检测出来。    图二、系统流程  由卷积神经网络实现。网络的初始卷积层从图像中提取特征，全连接层预测输出维张量，每个的维度对应原图个栅格中的一个，中含有类别预测和边界框坐标预测。总的来说就是让栅格负责类别信息，让边界框负责坐标信息（部分负责类别信息，因也算类别信息）。网络架构如图三所示。    图三、网络架构  网络训练：  首先需要在具有个类别的数据集上对该网络的卷积层进行预训练。在预训练阶段，使用图三所示的网络中前个卷积层，之后连接上一个平均池化层和一个全连接层（此时网络的输出是）。论文中提到，上述网络训练**一周**之后，达到了和模型相当的效果。  将卷积层训练好之后，就可以将该网络转换成执行检测任务。（论文中提到说在预训练网络中增加卷积层和全连接层可以改善性能。所以在预训练网络的基础上增加了个卷积层和个全连接层，随机初始化权重；此外，因为检测经常是需要细粒度的视觉信息的，因此将网络输入从转换成）  最后一层既预测类概率也预测边界框的坐标。使用图片的宽度和高度对边界框的**宽度**和**高度**进行正则化，因此最终预测的数值是的；此外，边界框的坐标设置成相对栅格左上角位置的偏移量，因此也是间的。  关于激活函数：最后 一层使用线性激活函数，其他层都使用激活函数，其中。  关于损失函数：该模型的输出中使用平方和误差函数。因为该函数容易进行优化，但是不完全符合最大化平均精度的目标。对象分类误差和对象定位误差赋予相同的权重，效果不理想。而且，很多栅格中不包含任何对象，这就把这些栅格的置信度分数推到，往往超过了包含对象的细胞的梯度。这可能导致模型不稳定，进而导致训练早期发散。为了解决该问题，对损失函数中的不同部分施加不同的权重：增加边界框预测误差的权重（用表示），减少不包含对象的边界框的预测置信度的损失（用表示），论文中，。平方差损失函数同等对待大边界框和小边界框，但是同样大小的误差，对于大边界框的重要程度要小于相对于小边界框的。为了解决这个问题，预测边界框宽度和高度的平方根，而不是直接预测宽度和高度。  总体的损失函数如下：    当待检测对象处于多个栅格中时，可能在多个栅格中都被定位出来。可以用非极大值抑制进行去除重复检测的对象，可以令最终的提高。  的不足之处：对相互靠得很近的物体（爱在一起并且中点都落在同一个格子上的情况）和很小的群体检测效果不好，因为一个网格中只能预测两个框，并且只属于一类；测试图像中，当同一类物体出现不常见的长宽比的时候泛化能力比较弱；对于损失函数的问题，定位误差是影响检测效果的主要原因，尤其是大小物体的处理上，还有待加强。 YOLO2 该版本的针对原来的版本进行了很多改进，获得了性能提升。主要改进如下：  1、批正则化：在所有卷积层上添加批正则化，批正则化能使收敛性得到显著改善，同时消除了对其他形式正则化的需求。  2、高分辨率分类器：所有最先进的检测方法都使用在上预训练的分类器。训练分类器时，输入图像的分辨率是，但是在检测的时候图像分辨率是，即该网络在将任务切换到学习目标检测的同时将输入调整到新的分辨率，这样的切换可能导致网络无法快速适应，从而导致效果较差。首先在数据集上以的分辨率对网络进行个迭代周期的微调，使网络进行调整来适应更高分辨率的输入。该方法使网络的准确率得到了一定的提升。  3、带有先验框的卷积：从中移除了全连接层，并基于先验框来预测边界框。首先，消除了一个池化层，使网络卷积层的输出具有更高的分辨率；此外，还缩小的网络，输入图像大小是而不是，这样做的原因是希望特征映射中有奇数个位置，这样特征映射图中就有一个中心（论文中提到，特别大的待检测对象往往占据图像的中心，所以在正中心的位置预测该对象能取得比预测稍微靠边位置更好的效果。中在卷积层对图像下采样倍，因此输入的图像，可以得到的特征映射）。  4、维度聚类：在中，虽然网络可以在最后对先验框进行微调获得更好的边界框，但是好的先验框可以令网络更容易学习。则现在训练集边界框上运行均值聚类，获得更好的先验框。论文作者进行实验，获得较好的的。  5、直接位置预测：针对每个边界框，该网络都会预测个值，分别是，，，，（这四个值也就是模型的预测输出）如果目标栅格距离图像左上角的距离是，并且该栅格对应的边界框的宽度和高度是和，坐标的损失采用平方误差损失，则网络的预测值会是下式所示。即预测边界框坐标相对栅格坐标的偏移，从而将和限制在之间。这样，可以保证网络训练时更加稳定。      图四  6、细粒度功能：即为了提升检测比较小的对象时的效果，不仅仅从的特征图中提取特征，也从比较大的特征图中提取特征（类似）。  7、多尺度训练：原始使用的输入，前文提到为了使用先验框将输入大小更改为。但是模型中没有全连接层，因此可以输入任意大小。为了使可以健壮地处理不同大小的图像，在训练时每几次迭代之后就改变输入图像大小（图像进行倍的下采样，因此输入图像大小必须是的倍数，训练时采用） YOLO3 在上做了进一步的改进。  1、类别预测方面：将原本的单标签改进为多标签分类，因此网络结构上就将原来用于单标签多分类的层换成用于多标签分类的逻辑回归层（使用就是假设一个图像只属于一个类别，但是在一些复杂场景下，一个对象可能属于多个类，比如说类别中有和两个类，这就是多标签分类，需要使用逻辑回归层对每个类别做二分类）。  2、多尺度预测：预测多种不同尺度的边界框。在进行基本特征提取时，用网络最后一个卷积层预测一个三维张量编码；同时，也要从前面的图层中得到特征图。将多个特征图连接到一起，就可以找到早期特征映射中的上采样特征和细粒度特征，并获得更有意义的语义信息。（基本类似中的方法）  3、使用新网络进行特征提取，该网络融合，和，该网络由连续的卷积和卷积组合而成，称为（因为使用了类似残差网络降低了训练难度，因此极大地加深了网络深度）。网络结构如下所示：    图五、 总结 综上，的基本思想是首先利用特征提取网络提取输入图像中的特征，得到一定大小的特征映射，比如大小的特征映射；然后将输入图像划分为个栅格（栅格大小和特征映射大小对应，对应位置的特征映射表示对应栅格的特征）；接着如果某个对象的中心坐标落在某个栅格中，则由该栅格来预测这个对象。虽然每个栅格都会预测个边界框（中，中，中），但是只有和真实边界框的最大的边界框才是真正用于预测对象的。 代码实现 1、搭建网络  此处使用官方提供的配置文件来搭建图五所示的。为了根据配置文件搭建网络，首先需要明白其含义。中使用了中类型的层，分别介绍其参数意思如下：  卷积层参数如下所示：表示是否进行批正则化，表示卷积核个数，表示卷积核大小，表示卷积核移动步长，表示边界填充大小，表示激活函数。   |  | | --- | | [convolutional] batch\_normalize=1 filters=32 size=1 stride=1 pad=1 activation=leaky |   层类似残差网络中的跳过连接层。参数是，则表示层的输出是该层前面层输出的特征映射和当前层开始，从后向前数三层输出的特征映射叠加得到。   |  | | --- | | [shortcut] from=-3 activation=linear | |  | |  |   层，即上采样层。使用双线性上采样方法对前一层的特征图通过因子的步幅进行上采样。   |  | | --- | | [upsample] stride=2 | |  | |  |   层的属性可以有一个参数或者两个参数。当只有一个参数时（例如该值是），该层将从层向后输出第四层的特征映射；当有两个参数时，该层输出两个层特征映射连接的结果。   |  | | --- | | [route] layers = -1, 61 | |  | |  |   层即探测对象的层。属性中有个先验框，但是只使用参数对应的先验框，针对下面的例子就是使用第个先验框。探测层的每个栅格只使用个不同的先验框，而探测层总共有个不同的尺度，因此一共有个先验框。   |  | | --- | | [yolo] mask = 0,1,2 anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326 classes=80 num=9 jitter=.3 ignore\_thresh = .5 truth\_thresh = 1 random=1 | |  | |  |   不是网络中的模块，而是描述了关于网络输入和训练参数的一些信息。在前向传递的过程中没有用到。   |  | | --- | | [net] batch=1 subdivisions=1 width= 416 height = 416 channels=3 momentum=0.9 decay=0.0005 angle=0 saturation = 1.5 exposure = 1.5 hue=.1 learning\_rate=0.001 burn\_in=1000 max\_batches = 500200 policy=steps steps=400000,450000 scales=.1,.1 | |  | |  |   搭建网络的大部分工作在中完成。该文件主要完成读取配置文件，将参数信息转换成一个网络。该网络的前向传播过程比较麻烦，因为有层和层，因此需要将每层输出的特征映射做一个缓存，在遇到这些层时，编写正确的下标将对应的特征映射取出。具体代码见代码文件和响应注释。  本次实验主要学习了完成对象检测的原理，搭建了响应的网络，实现了前向传播过程，以及针对测试图像得到正确的输出。无法自行训练网络，因中的卷积层训练代价太大了。因此，加载了训练好的权重文件作为网络权重。  网络权重文件是一个二进制文件，其中包含了以序列方式进行存储的网络权重。在整个网络中，只有批正则化层和卷积层有权重；此外，权重文件的权重时按照网络结构按照层进行存储的。因此，要按照网络结构逐层读取响应数量的权重。否则，可能导致权重加载错误。函数实现加载网络权重功能。 实验结果及分析   图六、输入图像，输出图像对比  某个测试结果如图六所示。从图中可以看到有些对象被正确识别，有些对象没有被识别。这是因为只能检测在训练数据集中出现的对象类别。训练时使用的是数据集，因此只能检测种对象。如果图像出的对象都是训练数据集中出现的，才可能检测出来，此时效果如图七所示。    图七、输入图像，输出图像对比 |