第二章 视觉相关算法理论分析

2.1 神经网络

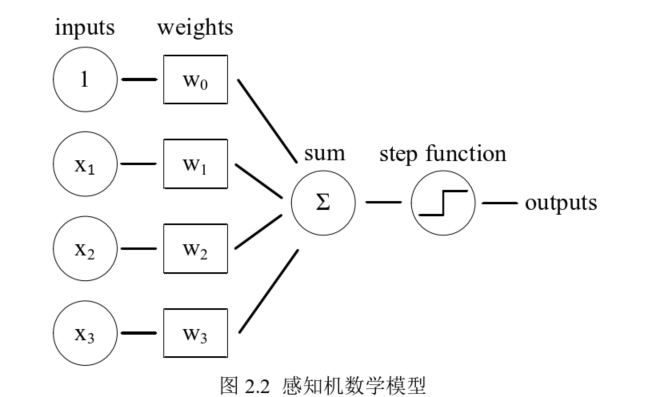
2.1.1神经元

人工神经网络最初起源于1943年，沃伦麦克洛克和沃尔特皮茨发表的神经活动中内在思想的逻辑演算，普遍认为MP模型是人工神经网络的起点，1957年罗森布莱特提出感知器的概念，人们认为只要将足够多的神经元组合在一起就可以解决任何问题正是这一概念的产生彻底带起了人工神经网络的第一次研究热潮，由于当时计算机性能的局限与对神经网络错误的认知，认为其只能解决线性问题，还有过拟合的无法解决，一直到2006年，随着计算机算力的提升与存储能力的增强，BP算法创始人辛顿再次提出深度网络和深度学习的概念，再次让神经网络提上热潮。

这些概念的追述其实来自生物智能的神经元，神经元主要包括树突、轴突、突触等概念，如图所示。



神经元通过树突与其他神经元树突的连接，多个神经元与多层次的连接最终形成大规模的神经网络，每个神经元都对最终结果产生一定量的微量影响与传递，正是这一特性，在实际应用中使用感知器作为神经元的代替模型，模仿神经元的传递信息过程。神经元的数学模型感知机如下图2.2所示，从图中可以得知出感知机拥有多个输入与输出的节点，对应了神经元的多个树突与突触，在视觉算法的数学模型中，这些输入对应着每个图像的所有像素值，权重是通过随机初始化，紧接着后续训练产生确定的，对应大脑中从对新事物第一次见面一无所知，到多次见面有了先验经验认知。经过输入与权重相乘之后求和，代表了当前神经元的对输入图像的所有认知，之后经过一个激励函数，激励函数可以等同于对神经元的一个刺激，激励函数也可以叫点火函数，这使与人脑的运行规则联系在一起，当神经元的输入足够大时才会触发点火的激励函数进行输出，最后输出到下一个神经元上。



感知机是一个简单模仿神经元运作的数学模型，他们相同点都是有多个输入和一个输出，但实际还是有着巨大的差距，生物神经元有着更加复杂的电化学反应，感知机只是简化动态模拟这一过程，下面以一个4个分别输入4个神经元为例，数学公式如下，





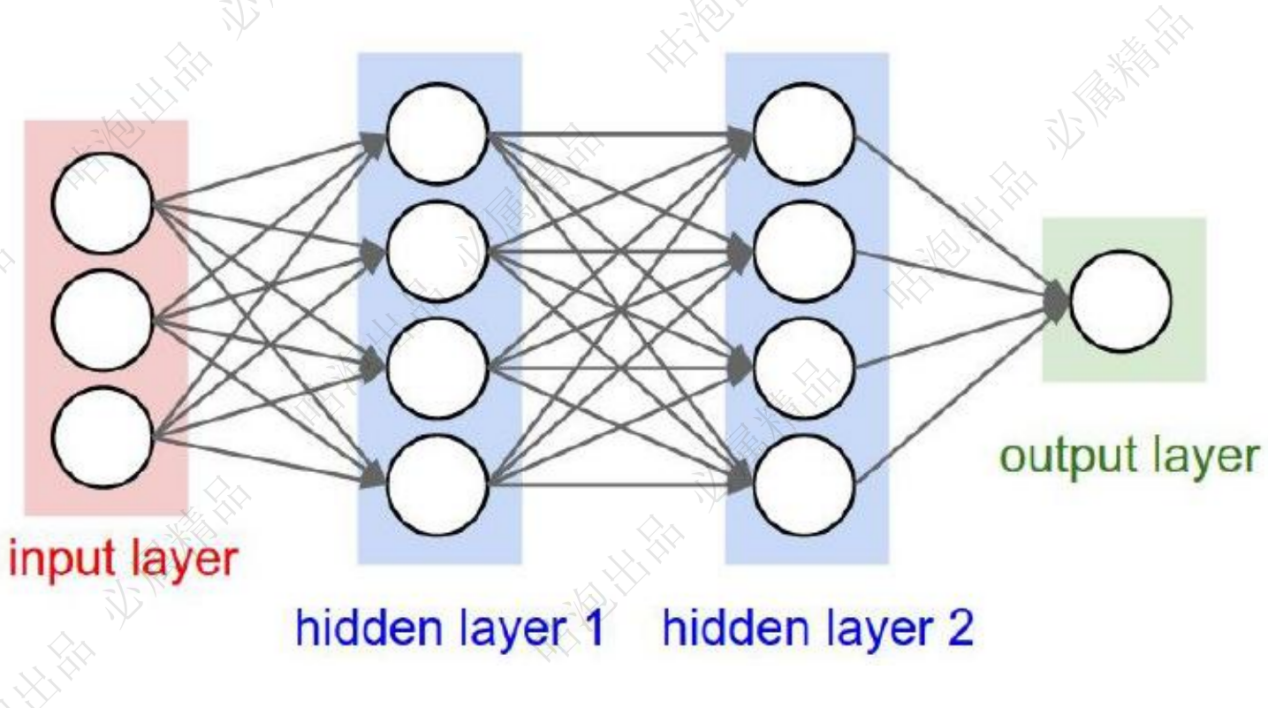




公式中𝑎x为感知机输出数据，Wxy为权重，𝑥x为输入数据，bx为线性偏置，经过输入与权重的累积加上偏置，再通过激活函数进行输出。

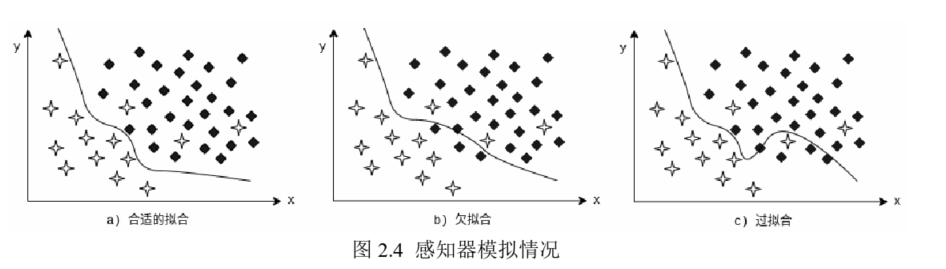
2.1.2多层感知机与神经网络

因为感知机是线性分类模型，无法拟合复杂函数，因此引入多层感知机，也称为人工神经网络，感知机之间相互叠加形成不同的层，从而完成各种逻辑功能，如图所示为2个隐藏层，由于输入层不参与计算故这是层数为3的多层感知机。加入隐藏层后理论上可以拟合任何数学函数变化，通过增加隐藏层，增加网络深度，可以使网络更好的拟合复杂函数从而完成复杂回归或分类任务。



拟合能力是评估神经网络性能的重要评价标准之一，下图为各种拟合状态，过拟合和欠拟合都是不好的表现，过拟合表示泛化能力弱，加入正则化是防止过拟合的有效手段，

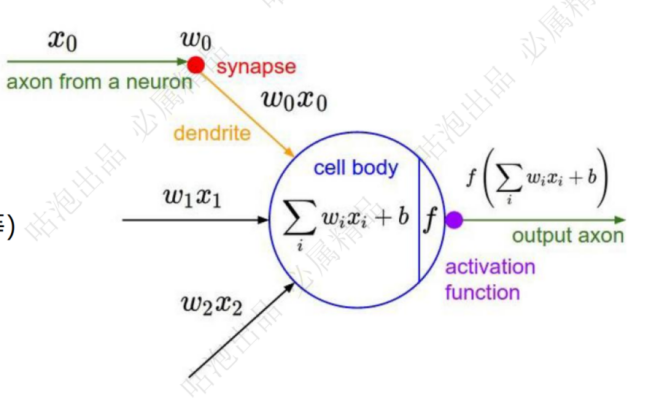
|  |  |  |
| --- | --- | --- |
|  |  |  |
| 合适拟合 | 欠拟合 | 过拟合 |



神经网络的训练就是通过不停通过反向传播优化隐层和输出层参数，让神经网络计算输出结果尽量与真实结果吻合。神经元数量影响了优化拟合度性能，而正则化惩罚λ则体现了对过拟合的纠正，效果如下图所示。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 4个神经元 | 8个神经元 | 20个神经元 |
|  |  |  |
| λ=0.001 | λ=0.01 | λ=0.1 |

激活函数是将非线性加入到神经网络的重要手段，其流程如下图，如果没有激活函数，那么无论多少层输入输出都是线性组合，都是原始的感知机，然而并不能满足现实生活中的非线性映射，故需要加上激活函数增加非线性，其要求必须满足非线性、可微性、单调性。常用激活函数有Sigmoid、Relu、Tanh。



正向传播（前向传播）是基于已有权重的网络模型，输入图像目标通过隐层的权重累积加偏置、激活完成一级一级迭代，将原始输入像素数据逐步转换为抽象特征，最终通过全连接输出分类结果。

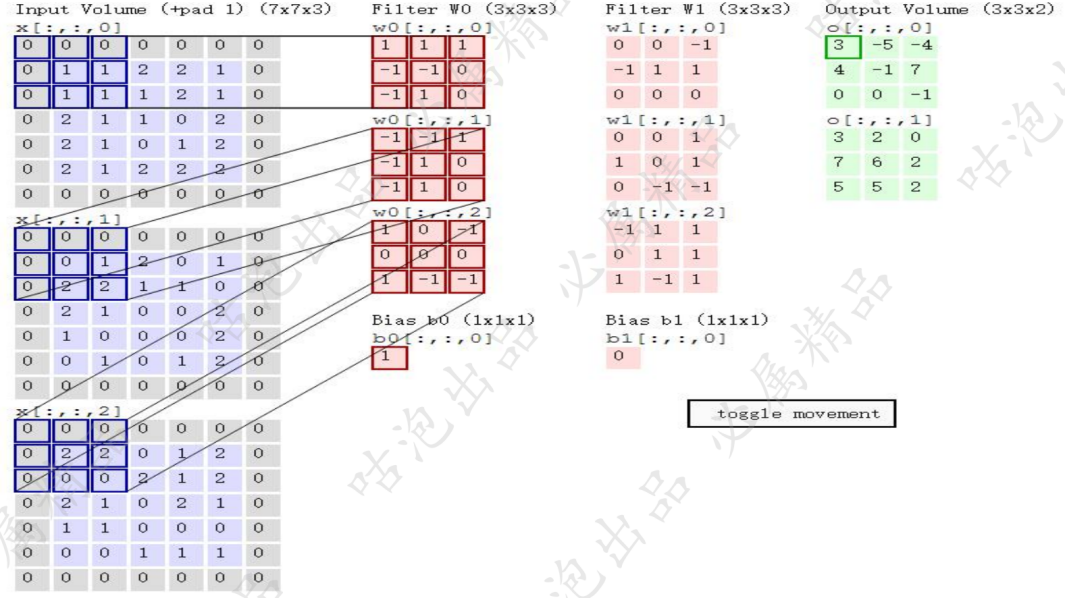
反向传播是根据正向传播结果与真实值计算损失函数，再用梯度下降方式通过链式法则计算出权重与偏置对损失的影响，最后更新权重与偏置参数。在训练网络过程中，正向传播与反向传播相互依赖，二者合一才是一个完整的训练迭代循环。

2.2 传统卷积神经网络

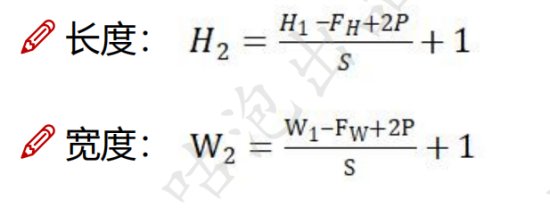
深度学习大致有两个主流方向，一是视觉CV领域，另一个是自然语言NLP领域。在视觉领域目前还是离不开CNN卷积神经网络。主要分为卷积层、池化层、全连接层、激活函数。常用卷积层有Conv2D、Conv3D、DeConve，池化层有Max Pooling、General Pooling、Average Pooling等，常用激活函数有Sigmoid、Relu、Mish。

2.2.1卷积层

卷积层是卷积神经网络的核心，也占据了大部分运算，其作用是从输入数据中提取特征，特点是计算速度快，省参数，计算机的优化相对于卷积运算更加友好。深度学习发展中选择放弃全连接的原因是，在处理很高维度的图像输入数据时，全连接运算量太大，而恰巧人脑在进行图像处理时也采用了像卷积一样的局部原理，一次只去图像一部分进行处理分析，卷积则模拟了这个过程，卷积层是先进行卷积核的滑动数乘累加再加上偏置组成，如下图所示。



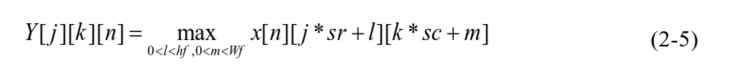
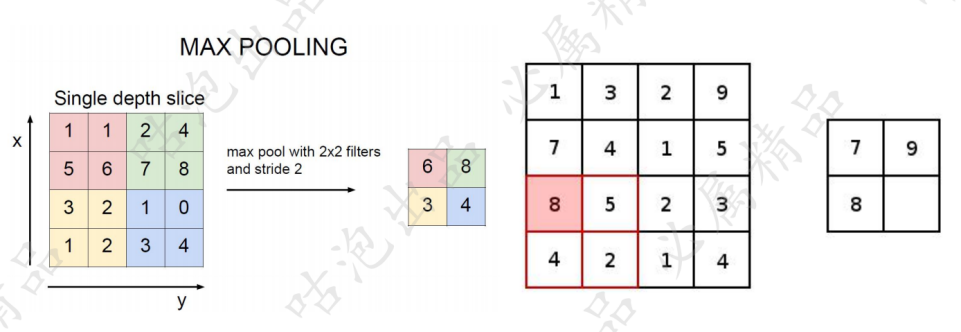
如图对于一个5×5×3的输入数据，先进行padding为1的边缘填充，增加了边缘数据的利用率，卷积核为两个3×3×3矩阵，经过卷积核滑动窗口遍历输入数据做矩阵乘法再加上偏置得到输出，输出维度为3×3×2。卷积层一般作为下采样，是将图像数据进行特征提取，往往会使数据维度的w和h变小，chanel变大。卷积层的参数包括滑动窗口步长S、卷积核尺寸F、边缘填充P、卷积核个数H。卷积计算公式如下。

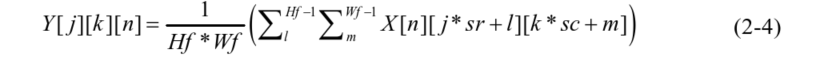
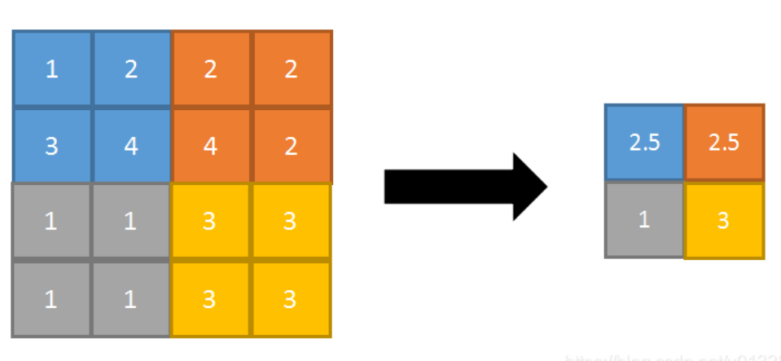


滑动窗口步长S、卷积核尺寸F、边缘填充P决定了输出特征图的长度宽度，而卷积核个数决定了特征图个数。

2.2.2池化层

上述的卷积层是用于对输入图像数据的特征提取，在特征提取的过程中，会出现局部过拟合现象，比如输入一张人像数据，经过卷积层后虽然图像被一定的压缩，但仍有大部分区域是有着重复的特征，池化层就是针对这一问题提出的，对特征图池化的过程中，减少了过拟合，更重要的是降低了特征他的维度，极大简化网络复杂度，减少计算量，也可以增加非线性，提高模型泛化能力。池化一般会使用最大池化Max Pooling和平均池化Average Pooling两种，其公式如下。



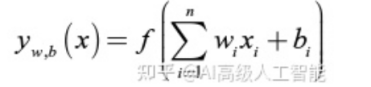


公式中的代表特征图的数据，代表池化后输出数据，公式和公式几代表了局部最大值处理和局部平均化处理。以下图为例，4×4的特征层被池化层2×2矩阵最大池化后输出了2×2的矩阵特征图，其中每个元素都是对应2×2区域的最大代表值。同样的以下图为例，是阵特征图4×4被2×2平均池化后池化输出结果为2×2矩阵，每个元素都是对应区域的所求平均值。

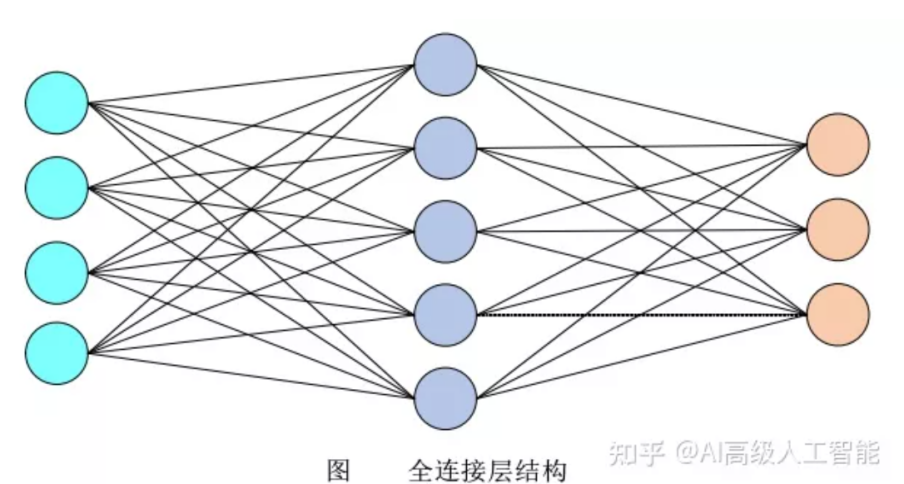
由以上内容可见，池化过程较为简单，只是想当于对特征数据的一次降维处理，只有简单的最大值比较或求平均值运算，最大池化优点是能很好的保留纹理特征，平均池化的优点是很好的保留背景。

2.2.3全连接

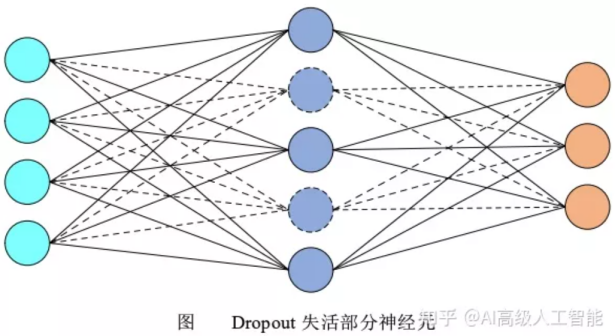
经过了卷积层的特征提取和池化层的降维处理，提取出了适量大小并有代表性的新特征图。而全连接把得到的所有特征数据矩阵进行重组，获得完整信息，与卷积池化的提取压缩信息不同，用到了得到的所有信息，全连接一般位于卷积神经网络的末尾，将重组后的信息首位相连，整合所有特征并映射到样本标记空间。全连接对前层输出特征进行加权求和，再讲结果送至激活函数，最后完成目标分类，其加权计算公式如下：



其中wi是全连接的权重系数，xi是上层第i个神经元的输出值，bi为当前偏置量，全连接结构图如下

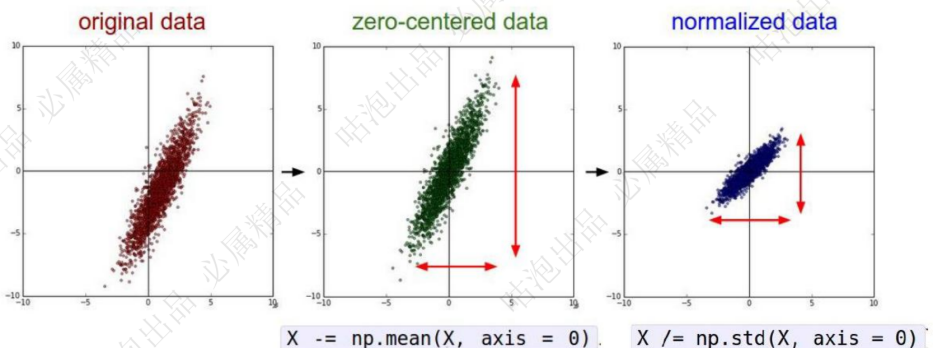


由于全连接层神经元全部互相连接，其参数量很大，几乎占整个网络模型的 80%以上，导致全连接层中的各个神经元输出的特征信息基本都是重复的，这使得网络训练中更易出现过拟合，所以使用在全连接层后添加 Dropout 的方法来避免。Dropout 层在网络训练时会使一部分神经元失效，这样就可以降低一些神经元间的关联，减少网络过拟合，提高泛化能力。Dropout 原理如下图所示：



2.2.3批归一化处理和激活函数

批归一化处理（Batch Normalization），简称BN，由于现在神经网络训练深度越来越大，过深会导致“梯度爆炸”和“梯度消失”的问题，这意味着当神经网络超过一定深度后效果反而会更差，无法继续增加深度通过大算力增强拟合性，而BN算法正是针对这一现象，于2015年被Google提出，解决了一定的网络深度问题。如下图所示针对，BN主要分为三步，首先对同一训练批次中数据求均值与方差；然后使用均值与方差对这一批次数据做归一化处理，通过对均值做减法将数据中心拉向原点，再对方差做除法将数据在各维度保持近似防止相差过大，完成归一化处理；最后对数据进行拉伸操作与偏移操作，也就是乘法与加法，最后操作是为了防止归一化操作失效，则选择不使用批量归一化操作。在进行批量归一化操作过程中将批量大小设置大一些可以得到更有效的均值与方差，从而使效果更好。具体表达式如公式 至 所示。



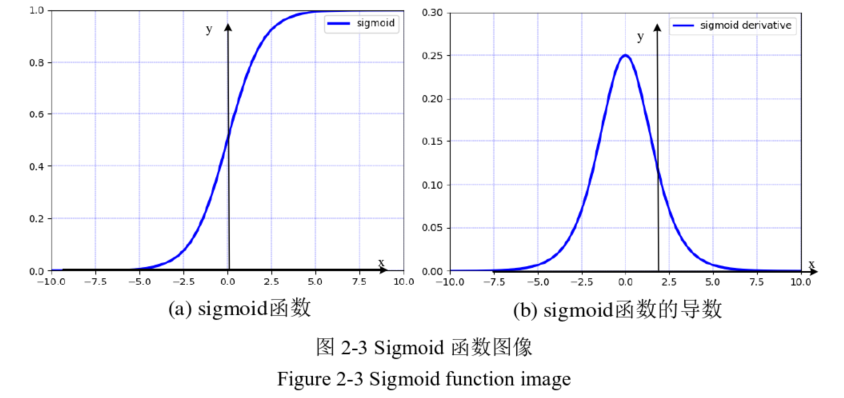




其中m表示批量大小，也就是一批次的样本个数，x表示输入样本数据，μ和σ分别表示均值与标准差，λ和β分别表拉伸参数和偏移参数，ε是为了防止分母为零而加入的小数值。

BN操作可以实现每层输入的数据进行归一化，从而使得每层网络的数据分布维度不会相差过大，统一后会有助于深度学习网络的快速收敛，目标函数相对更圆滑，改善了反向传播的梯度流，加快模型训练速度。

深层卷积神经网络大部分是由卷积层、池化层、BN层按一定序列排列的组合，逐步将图像数据提取为集合度越来越高的特征图，但是上述三种网络层只能拟合线性问题，无法处理生活种更为常见的非线性问题，因此引入了激活函数，激活函数最主要特点为神经网络提供非线性特征，使网络可拟合更加复杂的非线性问题，卷积神经网络中常用激活函数有Relu、Sigmoid、Mish函数。



Sigmoid激活函数：它能将任意连续输入的值都转换为0-1之间的输出值，对应关系为非线性，其数学表达式如式 所示：

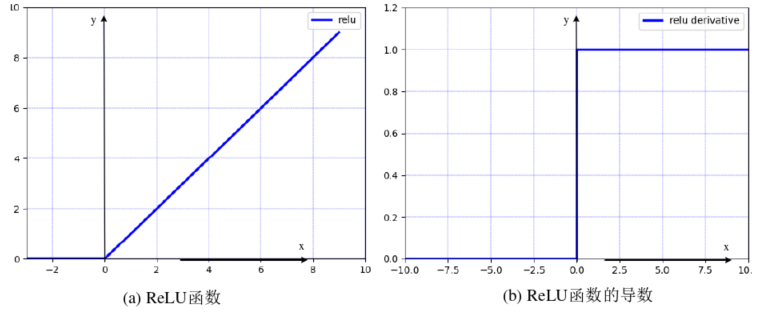


该激活函数的几何函数图像如图几所示，当输入值为无穷小或无穷大时，输出值则会变成无限趋近于零，这种情况导致卷积网络在训练中出现梯度爆炸和梯度消失的现象，从而导致模型无法继续收敛。由图也能看出其图像中心不在原点，因为这样导致后层神经元输入非0均值的信号，故会使权重更新效率降低，同事因为指数运算的参加，也会导致计算量增加造成计算成本高。

ReLU激活函数：ReLU函数是当下卷积神经网络常用的激活函数，如图所示，该函数表现为当输入为负值时一律将其输出变为0；当输入为非负值时，输出保持不变为原值，该函数数学表达式如式所示：

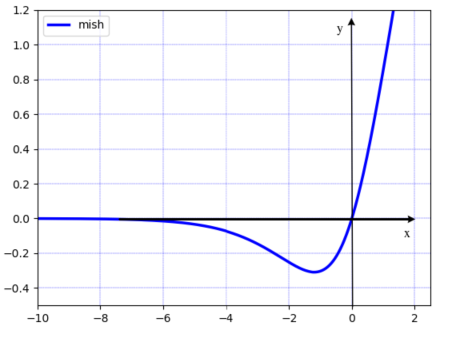


该激活函数可以有效改善Sigmoid函数的梯度消失、梯度爆炸的现象，在输入大于0时不会存在饱和问题，梯度不会衰减从而解决了梯度消失问题，让我们可以直接以有监督方式训练，而无需依赖无监督的逐层预训练，同时该函数数学表达式简单，在一定程度降低了计算压力，然而，该激活函数也存在一定不足，当输入为负值时输出恒为0，导致部分神经元无法激活，也称为“神经元死亡”现象。



Mish激活函数：该激活函数与ReLU最主要区别在于当输入为负值时，输出不在为恒等于0的数值，有效避免了神经元死亡现象。该函数数学表达式如式所示：

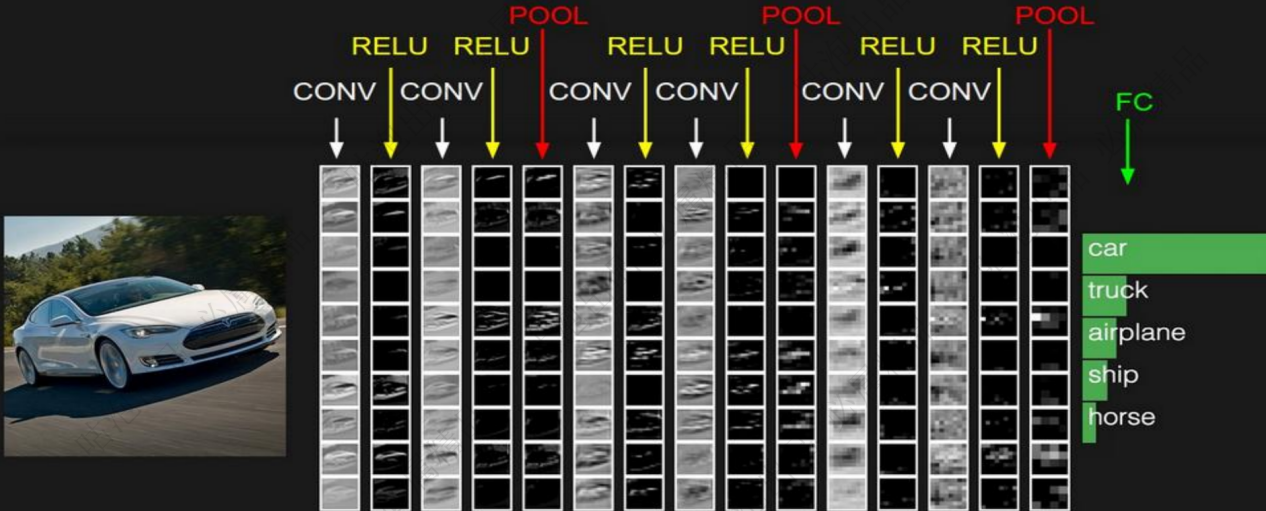




如图所示，该激活函数主要特点有在负值时不是直接截断，而是允许较小负梯度的流动，保证了一定的数据流通，并且无边界的特点避免了饱和问题，比如Sigmoid激活函数在极限情况下梯度趋向1，Mish则巧妙避开了这点，同时因为Mish图像每个位置都是平滑的，所以是的梯度下降效果比Relu更加平缓有效。

###正向传播

卷积神经网络是由卷积层、池化层、激活函数、全连接经过一定的排列，将一个图像进行逐步的特征提取，从只能人类看懂的数据逐步转换为计算机能够识别的特征图数据，最后根据不同的特征图的数据，进行判断分类得出结果，结果并不是一个确定的分类，而是一系列分类结果的权重系数，具体流程如下图所示。



该示意图将一个汽车图片输入卷积网络中，从图中不难看出，连续两次卷积和激活再加上一次池化是一次完整的下采样流程，会将图片矩阵数据进行降维处理，但会得到更多张相同大小的特征图，也就是说降低了数据的h和w维度，增加了通道维度，但总的仍然是减少了数据量，故称为下采样。在3次小循环后最后一层加入FC全连接，将所有特征图展开为一维数据，在做权重相乘与偏置，最终用Softmax激活函数输出分类结果。

###反向传播

反向传播算法（Back Propagation，BP）是目前常用的训练神经网络的参数更新算法，其主要思想为损失函数求误差，首先将图像数据输入到神经网络中计算，将得到的推理值与真实值进行损失函数计算求得损失，再利用损失值依次从输出层像中间隐藏层、输入层逐层进行反向传播，在传播过程会更新每一层的参数。其中随机梯度下降（Stochastic Gradient Descent，SGD）算法是一种常用的梯度更新算法，每次训练模型会从数据集中选择一个样本进行训练，并完成参数更新。其数学表达式如式所示



其中(x,y)表示输入卷积神经网络的数据，ω表示神经网络中需要更新的参数，η代表学习率大小。因为SGD算法每次只会选择数据集中一个样本进行训练与参数更新，因此算法的更新速度较快，但由于样本选择是有随机性，训练的结果易陷入局部最优解，这也是SGD算法的问题有着不小的优化空间。

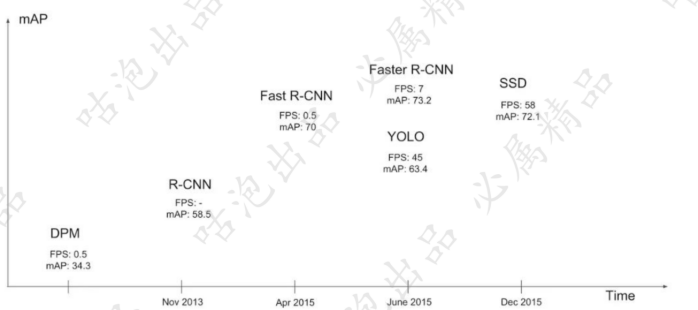
对应着解决问题就发明了小批量梯度下降算法（Mini-batch Gradient Descent）可以恰好弥补SGD算法的不足，MGBD算法的特点是每次不止选择一个样本而是选择一小批样本进行训练，这样的好处就是既不会有很复杂的计算量，训练速度也不会太慢，减少了模型总迭代次数。

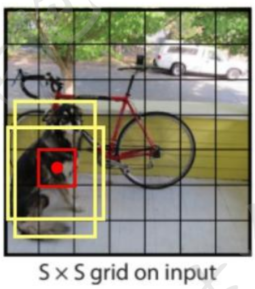
YOLO目标检测算法

2.3.1YOLOv1

深度学习算法大致分为两类，一类是两阶段式，代表为Faster-rcnn、Mask-Rcnn系列，他的特点是速度较慢，大约5FPS的速度，但整体的识别效果是不错的，相比较单阶段多了一个预选框的判断循环；而另一类是单阶段式，也称为one-stage，代表为YOLO系列，最核心的优势为处理速度非常快适合做实时检测任务和剪枝后移植嵌入式做线下实时设备，缺点也是效果比双阶段差一点。其代表作就是YOLO系列算法，YOLO算法以其快速性与准确性著称，版本也一直在更新迭代中，现有的版本号有YOLOv1至YOLOv7版本，但无论到哪一版本其核心都是以YOLOv1为主导进行研究的，故本小节开始为YOLOv1算法原理进行阐述。

YOLOv1的输入参数为固定的448×448×3的RGB三通道的图像。在实际训练过程中，将整个图像划分为7×7的网格，如下图所示。其特点是将检测问题转化为了回归问题，单纯CNN就解决了大部分问题，在刚提出的时代引起很大波动，代表了单阶段检测的强势崛起。下图是之前算法框架的map值，其中YOLO的平均精准度虽然不是最高，但他的速度已经可以独占鳌头。

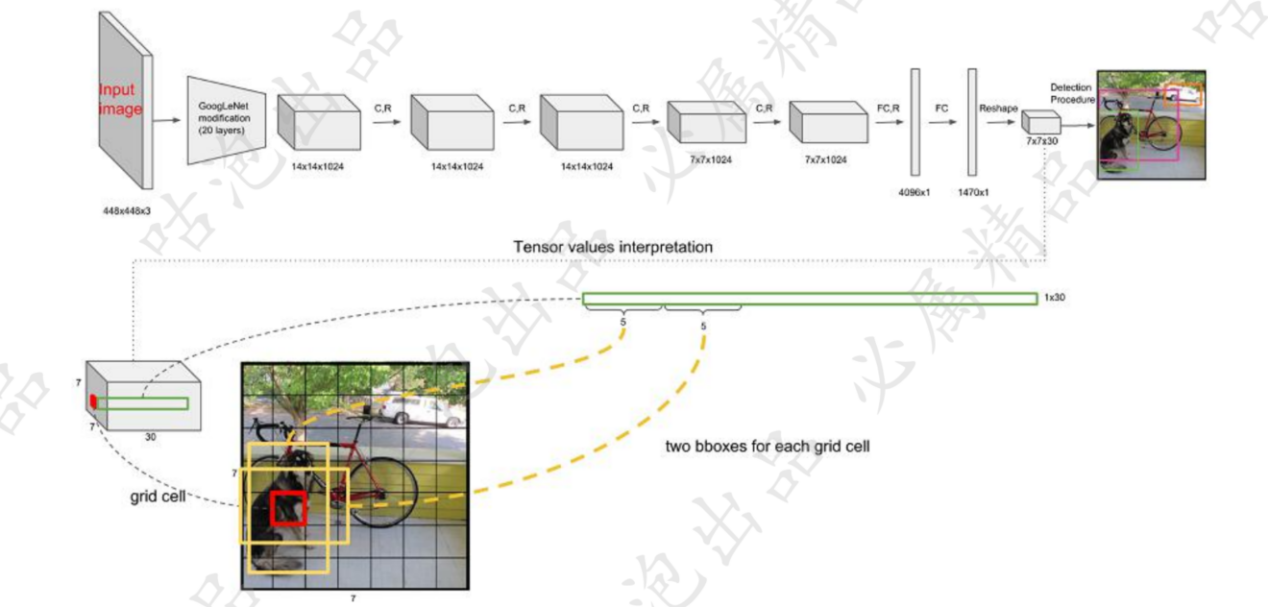




在图中每个格子都会预测两个预测框（bounding box）作为预测首选项，当一个真实待测目标恰巧也在该网格中时，该网格会执行预测工作，每个预测框的位置由四个数值表示，其中有中心点坐标两个以及框的长度与宽度，并且各个预测框有个重要概念表示预测的准确性，即iou的值，其值来自于预测框与真实框的交集与并集的比值，其代表了预测框的准确度，而每个预测框会利用置信度来表示是否存在目标与iou的乘积，其表达公式如式所示。



公式中，confidence代表置信度，IOU代表交并比。YOLOv1的输出结果为7×7×30，其中的7×7代表划分成的每个网格，而30代表特征向量。前面两种预测框的坐标和置信度占据了10个空间，剩下的20个则是判定为某个类别的概率，其中的20则代表该算法最多可以预测20种类别。该网络一共有24层卷积层作为特征提取层，如图2.11所示，各层之间使用最大池化层用作数据降维，各个层均使用Leak Relu为激活函数。最后使用2个全连接层。



YOLO算法成功之处就在于将目标检测任务转变为了回归问题，不同于双阶段任务，YOLO直接通过一次回归即可得到目标的坐标，同时得到了每个目标的概率值。YOLO最独具特点的就是简单粗暴的将整个图像划分为了7×7的网格。每个网格需要检测2个预选框，总共也只需要98个判断对象，随之带来的改变就是速度的极大提高，同时准确率并没有降低很多，满足大部分工作要求，这就是YOLO能风靡一时的重要因素。

2.3.2YOLOv2

YOLOv2使用的特征提取Backbone为Darknet-19，是基于Inception的定制网络，其中包括了卷积层、最大池化层。BN（batch normalization）层，通过3×3的卷积核进行特征提取与1×1卷积核进行特征压缩，还有池化层的下采样，进行5次池化后特征图变为原来1/32，将原来输入的416×416图像5次降采样为13×13，特征图通道也增加至1024个，取消了FC全连接，使用1×1的卷积核省了很多参数。YOLOv2是YOLOv1基础上改进的，具体有如下改进。

1. 批归一化处理（BN）

为了解决反向传播过程中梯度消失和梯度爆炸，提出了BN批归一化处理，同时降低一些由于数据维度不匹配的敏感性，训练中能起到正则化效果，模型收敛变得更加容易，其次舍弃了Dropout的随机神经元抑制，虽然可以避免过拟合，但也会带来一些特征损失，故将它舍弃。经过Batch Normalization处理后的网络会提升2%的mAP，现在Batch Normalization处理已经成为了各大知名模型的必备处理方式。

1. 使用分辨率更高的图像

YOLOv1使用的训练策略是用224×224分辨率图像进行训练，来训练backbone的参数，但在进行测试阶段却使用了448×448的较高分辨率样本进行测试，其中分辨率有个突变过程会导致模型的不适应性，对收敛会产生一定影响。而YOLOv2的策略是在进行224×224的图像训练完成后，会再加入10次448×448的较高分辨率样本的训练进行参数的微调，让模型有个向高分辨率的适应过程，最后再进行448×448的样本测试，这一缓冲操作将模型的mAP提高了大约 4％。

1. 使用先验框

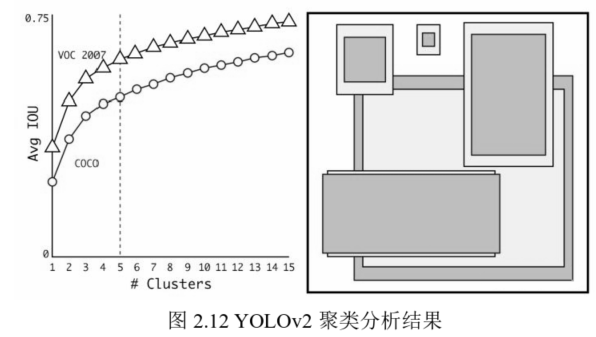
借鉴Fasterr RCNN使用先验框，在每个网格中都会预先设定一组大小尺寸不同的先验框，用来检验出各个位置的正样本，提高一定的召回率。这些先验框不是一直不变的，会在训练中选出每个正样本对应的不同先验框的概率值，最后通过非极大值抑制（NMS）来最终确定候选框，最后进行训练中的微调使之与真实目标越来越靠近。相比较上一版本预测框总数从v1的98个上升到了845个预测框，召回率从81％上升为88％，mAP虽然下降了0.2%，但整体来说提升效果更明显。

1. 聚类提取先验框大小

相比于YOLOv1的先验框均为人为设定大小与长宽比，而YOLOv2会根据数据集做一个统计，来设计出更符合真实数据情况的先验框，这样也降低了后面反向传播过程自主调节先验框的难易程度。首先具体做法是对数据集中样本的标注边框进行统计和k-means聚类提取，从而选出一些与真实标注框尺寸最为接近的的先验框。其中k-means聚类过程中最关键的是求真实框与预测框的距离，不同于传统欧式距离法，而引入了IOU概念的特殊距离算法，公式如下所示：



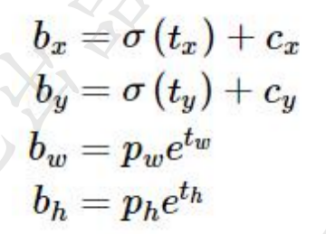
式中d表示相差距离，box与centroid表示被测边框，IOU是两个框所围面积的交并比，即重叠区域比上并集的所有区域而得到的值，IOU的值取值范围在0-1，当完全重合时IOU为1，故可以用1减去IOU来表示两框相距的距离。最终经过k-means聚类了coco数据集与voc2007等大型公开数据集的结果如图所示。



左侧图表示在聚类不同数量k的先验框下，平均IOU的曲线关系，明显可得k越大越满足条件，可为了综合考虑性价比最终选择曲线拐点k为5时效率最高，故本模型选择了先验框数量为5。效果与人为设定相比，可跟比自己高出接近一倍的人为设定框效果相比，进步明显，右图表示聚类出的5种先验框的尺寸，形状多为长方形边框。

1. 约束预测边框位置

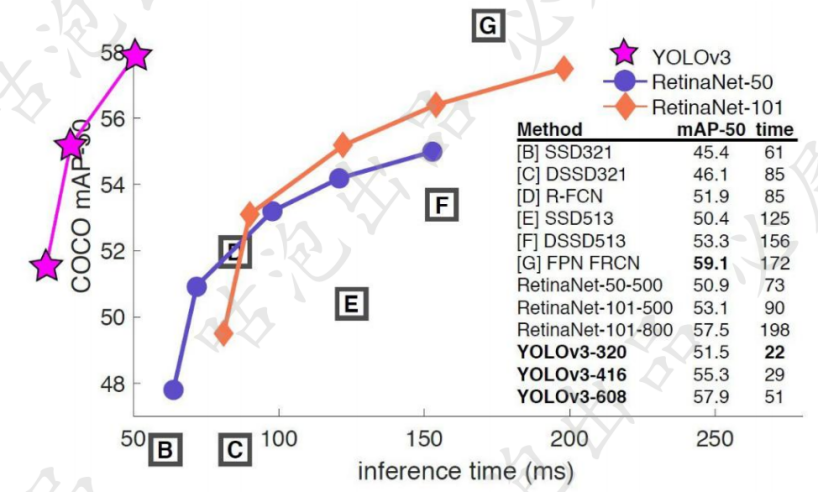
与FasterRCNN算法相比，在初期训练中预测框位置震荡严重，导致收敛速度极慢，YOLOv2对预测框公式进行了调整，将预测框的偏移量限制在网格单元内，而不是之前的全局偏移量。公式如下所示。



其中bx与by表示预测框中心坐标，bw和bh分别表示预测框的宽与高，pw和ph是先验框的宽和高，σ是sigmoid函数，t是要学习的参数，cx和cy是网格左上角到图像左上角的距离，网格的大小要做归一化，每个网格宽高均为1×1，总图像大小变为13×13。其中偏置相是由t经过sigmoid函数做处理，输出控制在1以内，这样就保证了训练过程中预测框位置波动不会太大，控制在一个网格中，使得模型学习效率提高，结果更易收敛。

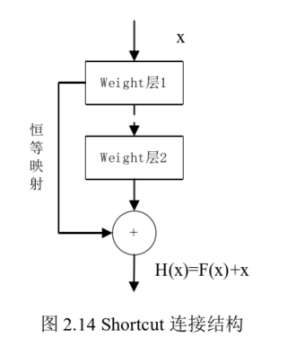
2.3.3YOLOv3

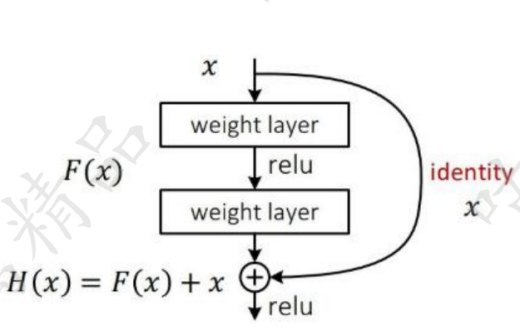
YOLOv3在原来基础上最大的改进是优化了网络结构，更加适合做小物体检测，特征提取做的更加细致，来满足不同尺度的物体检测，先验框分类更加丰富，加入softmax可以处理多标签的预测。是当时的集大成者，无论从速度还是准确来讲都远高于当时其他模型，对比如图所示。



1. Darknet-53网络结构

Darknet53框架加入了Shortcut连接模块，该模块主要借鉴了Resnet的网络架构，借鉴了残差学习单元结构，如图所示，在保证对特征表达不失真的前提下又避免了网络深度过大带来的梯度问题。





残差块（shortcut connections）分为两部分，一部分是映射部分（x），另一部分是残差部分（F(x)），可以用公式表示。当没有shortcut connection（图右侧从 x到⨁的箭头）时，残差块就是一个普通的2层网络。残差块中的网络可以是全连接层，也可以是卷积层。设第二层网络在激活函数之前的输出为H(x)。如果在该2层网络中，最优的输出就是输入x，那么对于没有shortcut connection的网络，就需要将其优化成H(x)=x；对于有shortcut connection的网络，即残差块，如果最优输出是x，则只需要将 F(x)=H(x)−x 优化为0即可。后者的优化会比前者简单。



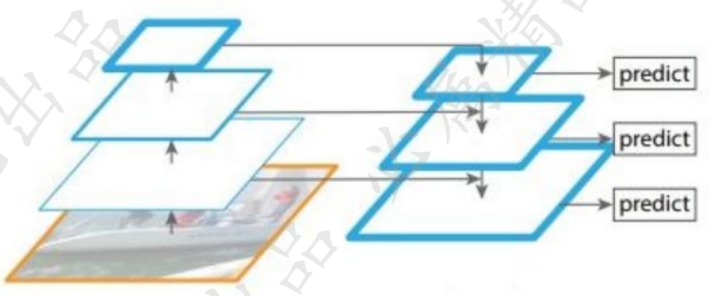
H(x)是求和前的网络映射，H(x)是从输入到求和后的网络映射，残差的思想是去掉相同的主题部分，更加强调微小变化。

通过实验表明，ResNet模型的准确行与模型深度成正相关，对前人的网络模型有较好兼容性，它最大的特点就是帮助打破了深度学习无法继续挖掘深度的瓶颈，可以更好的堆叠层数和挖掘深层特征信息。因此被应用到YOLO算法中。

YOLOv3的核心架构中取消了全连接和池化层，全部使用卷积层代替，好处是节约了参数，减少了计算量，同时没有使用池化的特征损失，下采样的过程由卷积层的移动步长设为2来实现。

1. 多尺度检测

多尺度检测的意思是在不同大小的尺寸情况下对结果进行预测，该网络构建了图像金字塔如图所示，先经过下采样降低特征图维度，随后在后面的一些网络层进行一次上采样，最后将两种采样过程中相同维度的特征进行拼接操作，不同深度与尺度的特征图都会结果产生影响，这样来实现多尺度检测的目标，故YOLOv3的小目标识别得到了优化。其中不同分辨率的特征图对结果的影响是不同的，分辨率越高意味着信息量越多，对细节的描述也越多，提供了更多的语义信息，可以区分背景与正样本，同时更适合做小目标检测。而特征图分辨率越低的情况下表示其感受野较大，从语义信息来讲其包含的更宏观的信息，对整体的把握更全面，故其适合做大目标识别。可无论是大尺度还是小尺度分辨率都需要结合在一起，共同决定最后的输出结果。其中YOLOv3为了检测不同尺度的物体，设计了3个scale的先验框，每个scale中包括3种先验框，一共为9种不同尺寸先验框，一定程度上提高了召回率。

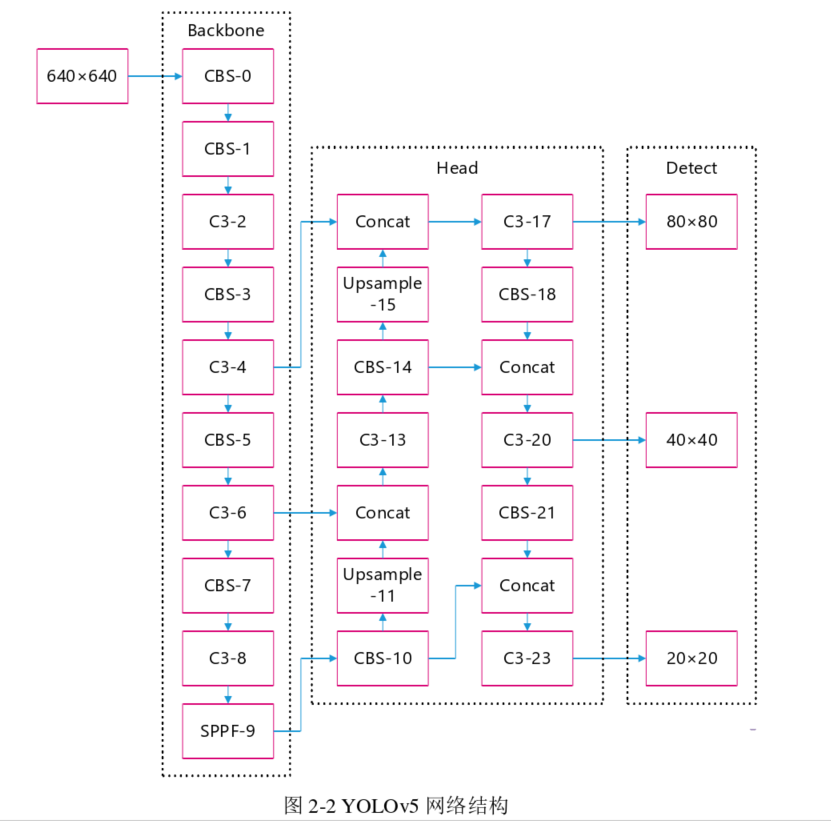


1. 目标类别预测

因为现实生活中对目标的预测中，有时同一个目标会有多个身份，需要多标签的支持，所以使用了多个Logistic分类器代替之前广泛使用的softmax，因为Softmax分类器只能通过对比Score的大小来确定最可能的类别，认定一个目标只能为一个类别，而Logistic可以使用多个独立的分类器解决多标签问题，且不影响最终准确率。

2.3.4YOLOv5

YOLOv5结构分为4大块，分别是输入端、骨干网络（Backbone）、颈部层（Neck）和输出端。输入端是一些数据的预处理操作，其中包括Mosaic、mixup和copy-paste数据增强、自适应锚框计算还有图片的自适应缩放；骨干网络包括了Focus模块、CBS模块和SPP模块；颈部包括了FPN+PAN的特征融合结构。YOLOv5基本模块包括由Conv、Bn批标准化、Leaky-relu激活函数组成的CBL模块、Resunit残差网络模块、由卷积层和多个Resunint模块拼接而成的BottleneckCSP模块和采用最大池化方式进行多尺度融合的SPP模块，以上模块间通过维度扩充的张量拼接操作和维度不变的张量相加操作进行融合。如图为YOLOv5的网络结构，其中分为各种尺寸深度的模型，除了深度以外其他基本一致，故本节以大小适当的YOLOv5s版本为主介绍其网络架构。



1. 输入端

Mosaic数据增强是为了弥补小目标样本不足而导致的平均精度低，此操作会将随机选择4张图片数据进行随机裁剪、翻转、旋转、缩放等方式拼接，来增加小目标样本的数量，提高mAP。

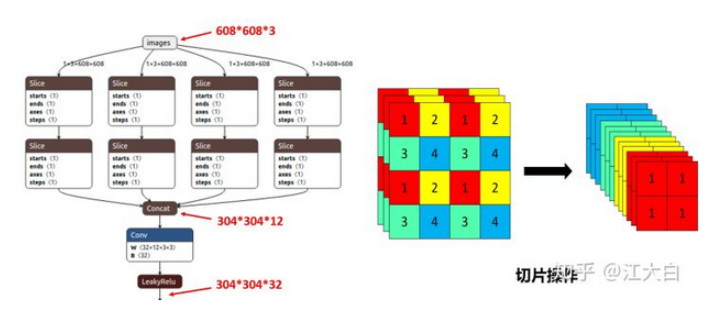
自适应锚框设计是YOLOv5起初在Coco数据集中聚类出9种不同大小的初始锚框，其尺寸如下图所示，因为Coco是一个庞大的数据集，故其聚类出的先验框尺寸是具有普遍性的，能符合非常大部分的样本数据条件。在网络训练中会在预定义框的基础上输出bounding box和真实框groundtruth进行对比，计算损失再反向更新，会自适应计算出不同训练集的最佳锚框尺寸。



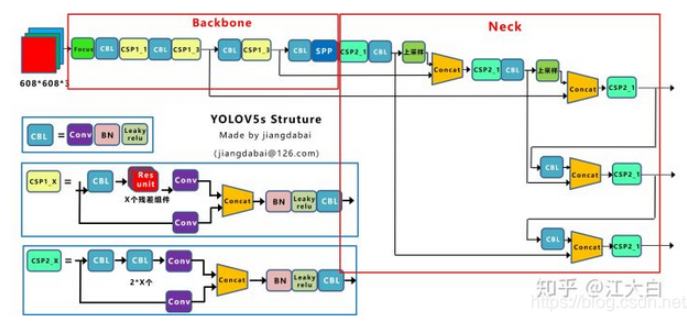
自适应图片缩放是为了统一输入网络中的图片尺寸，虽然之前的版本也有此功能，但YOLOv5对此做了改进，不会盲目的填充黑边来实现图片比例的统一，而是在letterbox函数中通过计算得到最佳的缩放比例，使自适应填充黑边最少，以此来加快了模型推理速度。

1. 主干网络

Focus结构是YOLO系列中YOLOv5最先使用的一种结构，其中最关键的步骤是对图像的切片操作，从输入图像开始每隔2×2的区间取一个像素点，将每次取出同一位置的信息拼接在一起形成4份比原来更小的图像，w与h尺寸变为原来的一半。具体以YOLOv5s结构为例，如左图所示原始输入图像维度为608×608×3，经过Focus操作后变为了304×304×12的特征图，通道数变为4倍，w与h降为原来一半，随后在进行一次32个卷积核的卷积层，最终特征图尺寸变为了304×304×32，可以实现模型训练加速。

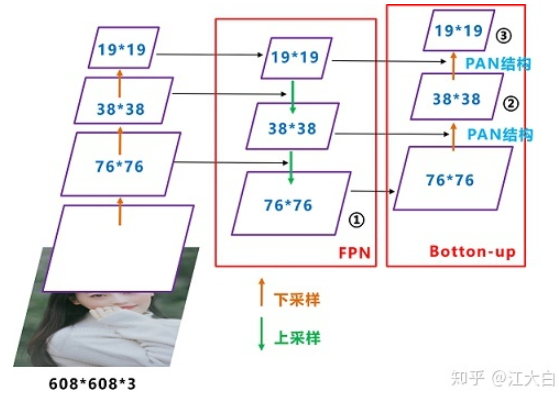


BottleneckCSP结构借鉴了CSPNet设计思路，但不同于YOLOv4中只在骨干网络中使用，而是分成CSP-1与CSP-2两种，分别应用于骨干网络和Neck层。其网络架构如图所示，其中CBL包含了卷积层、批归一化、leaky-relu激活函数，CBL与卷积层、批归一化、激活函数经过残差网络一起组成CSP模块，最后经过以上结构的排列组合形成了YOLOv5的骨干网络。其作用就是一层一层的下采样提取特征图，供后续模块使用特征图进行目标识别。



1. 颈部网络

YOLOv5的Neck层与上个版本类似，如图所示均采用了FPN与PAN的组合。其中FPN负责上采样，在特征金字塔中自下而上的将底层的语义特征信息进行传递，同时与初次下采样时相同深度的同维度数据进行融合，表达出强语义信息特征；其次PAN负责下采样，在特征金字塔中自上而下的将形状、位置等微小细节信息与底层特征进行融合，更好的表达出强定位信息特征。经过两种特征金字塔的上采样与下采样，将不同大小感受野的特征进行融合，对不同检测层的参数聚类操作，从而得到最终待预测特征图。

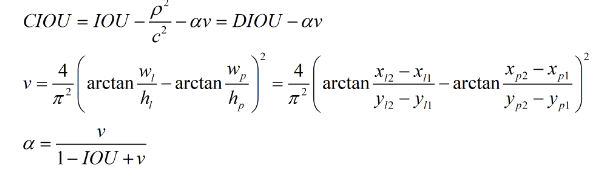


1. 头部检测层

Head层与之前版本类似，均使用三种不同尺寸特征图进行预测，使用了CIOU-Loss损失函数及非极大值抑制方式进行最后预测。损失函数的值代表了模型预测结果与真实值的差异，损失越小说明模型拟合的越好，通过训练模型来减小损失函数的值，使模型变得更收敛。

损失函数是判断模型效果好坏的重要方式。

YOLO系列的损失函数，分为分类损失（cls-loss）、定位损失（box-loss）、置信度损失（obj-loss）。分类损失是计算矩形框与标定类别是否一致的误差，定位损失是计算预测框与标定框之间的位置距离误差，置信度损失是计算网络的置信度误差。几种主流的损失函数中，IOU-Loss只计算了预测框与真实框的重叠面积，没有考虑其他尺度的影响；DIOU-Loss加入了真实边框与预测框的中心距离；GIOU-Loss考虑了真实矩形框与预测框的重叠面积、中心距离，但由于形状的不同也会影响结果，长宽比的不同也需要考虑，故为了进一步提高训练的稳定性与收敛速度，YOLOv5提出了CIOU-Loss，将重叠面积、中心点距离、宽高比都加入了计算。其计算公式如下。

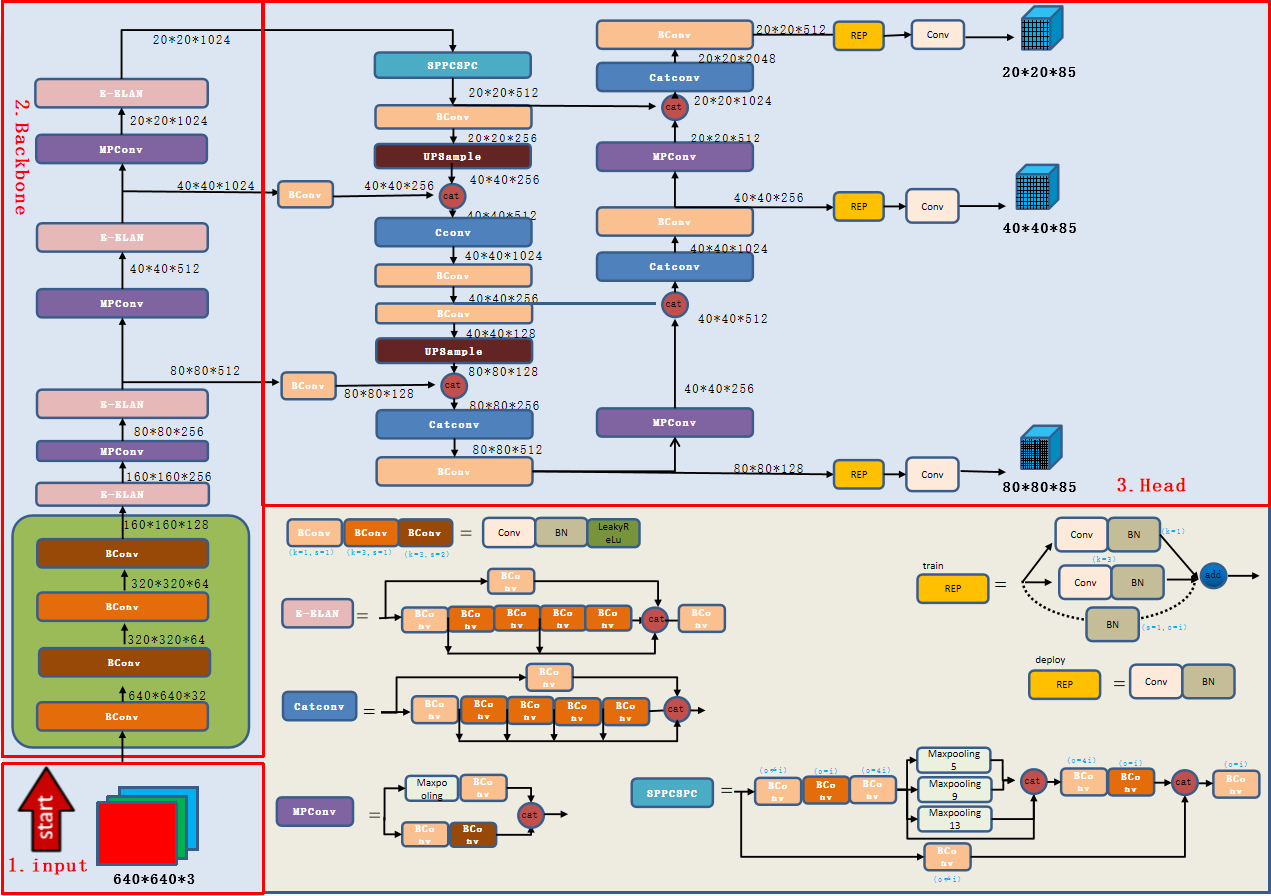




其中ρ为框A和框B的中心点距离，c为框A和框B的最小包围矩形的对角线长度，v为框A、框B的宽高比相似度，α为v的影响因子。反正切arctan函数的取值范围是0~Π/2，那么v的取值范围为0~1，当框A、框B的的宽高比相等时v取0，当框A、框B的的宽高比相差无限大时v取1。当框A、框B的距离无限远，且宽高比差别无限大时DIOU取-1，v取1，alpha取0.5，此时CIOU取-1-0.5=-1.5；当框A、框B完全重叠时CIOU取为1。CIOU取值范围是-1.5~1。在最后预测阶段，每个预测框都会输出一个预测的结果，选择最优预测框使用NMS极大值抑制去掉多余的预测框，当多个框重合度很高时，说明是同一个目标，取置信度最高的作为最终预测框。

2.3.5YOLOv7

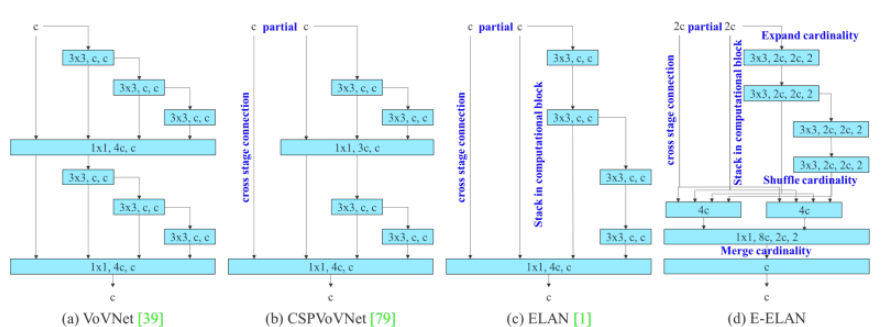
YOLOv7最为当下最先进的YOLO系列，其特点为集合了当下不同种类框架的最优秀的环节，集众家之所长打造出最强网络框架，其在5至160FPS范围内无论是速度还是精度均超过了目前已知最为流行的目标检测器。其整体框架如图所示，分为input输入层、backbone骨架、head头部层，因为其基本框架是基于YOLOv5基础上做的，所以框架整体变化不大，主要体现在一些细节的打造和研究，使得YOLOv7的效果更快更好。下面是一些与之前版本优化改进的部分。



###扩展更有效的层聚合网络

为了设计更高效的结构，一般会考虑减少参数量、计算量或者计算复杂程度来考虑，另一种是考虑输入到输出中间的通道数量、点乘规模、模型尺寸大小等方面来提升。如图所示为层聚合网络的更新迭代过程，CSPVoVNet是VoVNet的优化版，加入分析了梯度路径来使不同层的权重参数能学习得到更大差异性与代表性的特征。ELAN为高效远程注意力网络，利用shift卷积提取图像信息，加上多尺度注意力模块共同搭建而成，相比提高了计算速度，为了设计更加高效的网路，提出了E-ELAN为拓展高效远程注意力网络，使用了expand、shuffle、merge cardinality等模块对ELAN进行增强，首先使用group卷积来增加通道数和 计算块，

随后将计算块的特征图洗牌到多个组，然后连接合并，这样每个组中的特征图通道数和初始结构的通道数是相同的，最后将所有组的特征都相加。所提出的扩展E-ELAN完全不改变原架构的梯度传输路径，而是利用群卷积来增加新增特征的基数，并以shuffle和merge的基数方式将不同组的特征进行组合。这种操作方式可以增强不同特征映射学到的特征，提高参数和计算的使用。



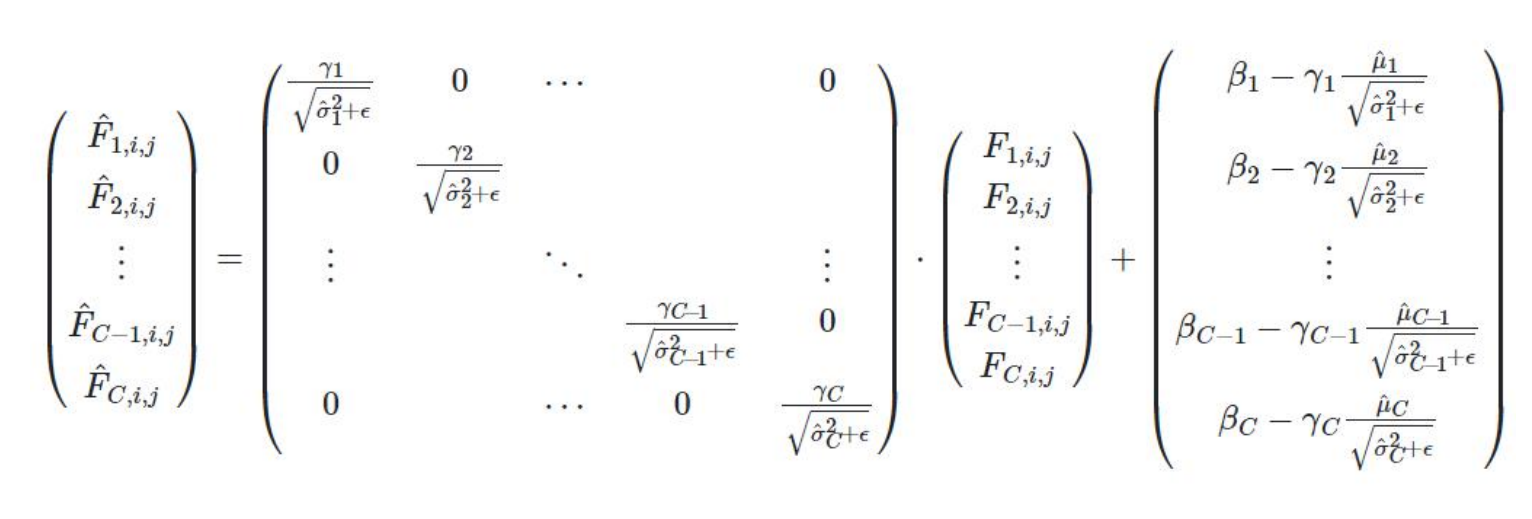
正样本分配策略

在原YOLO系列中，网格分的越细则正样本分配越细，召回率提高的同时也会增加计算量，而YOLOv7的正样本分配策略是在网格相同数量前提下，增加了正样本的召回率，其核心思想是真实框的中心落在那个网格，那么对应的锚框即为正样本，预测过程中选择单一的网格作为锚框选择的话，可能会导致复杂背景下仍有正样本被负样本所遮挡影响，故在其真实框中心所在位置，将此点在网格的上下左右各移动0.5个网格单位长度，此范围内涉及的网格均作为正样本所在锚框位置，如图所示中心点靠近当前所在网格右下方，那么下方网格和右方网格也算作被正样本选中的网格，正样本的考虑范围变为原来3倍，故提高了召回率。除此真实框与预测锚框的长宽比例的差异也被限定在了4倍以内，防止出现过于不靠谱的预测框来减少计算量。除此还增加了AUX辅助头输出，在原有的预测输出基础上又增加了一层辅助输出，原来的3层输出变为了6层输出，辅助头同样参与模型预测过程，但在结果预测时仅为主预测头提供参考，主要决定依旧取决于主预测头。



模型结构重参数化

其主要目的是为了优化速度，主要因为英伟达GPU和CUP对3×3卷积有着深度的推理加速优化，因此与其他种类卷积核相比3×3卷积的计算密度更高、速度更快。所以优化方向为尽可能只使用3×3卷积作为特征提取的工具，YOLOv7中完成了卷积与BN批归一化处理的合并，二者功能集成在一个3×3的卷积中。其旨在训练阶段资源较为充足的情况下，既不降低预测的准确行又可以加快预测速度，所以训练过程可以更加复杂一些去迭代更完美的权重参数。卷积与BN融合的第一步过程如图所示。



将卷积核拆为对角矩阵，并将BN的参数与之数乘得到图中矩阵，

WBN∈RCXC和bBN∈RC是BN的参数，Wconv∈RC和bconv∈RC是卷积层在BN之前的参数因此得到的公式为式，故权重参数为式，偏置为式。





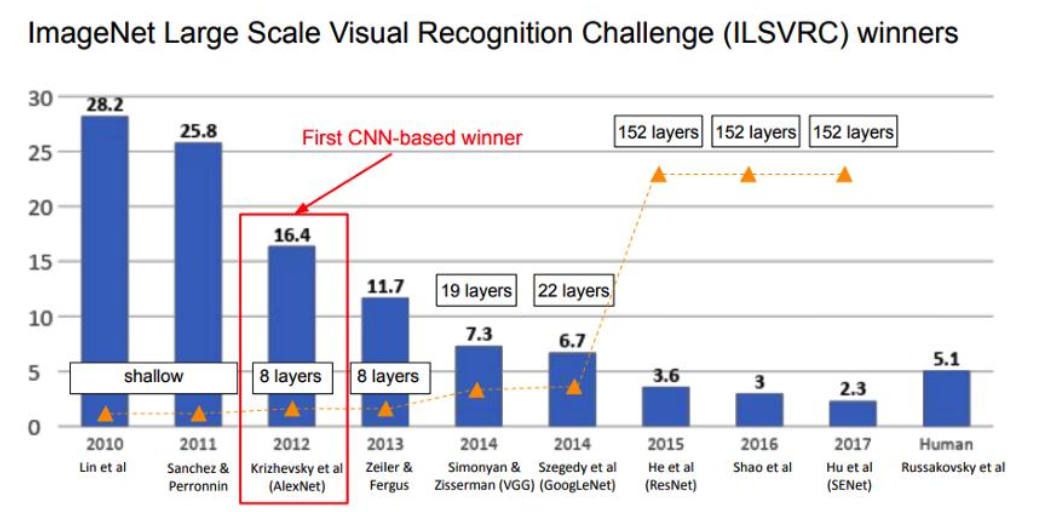
则结果相当于一个1×1的卷积核，再将它拓展为3×3卷积核，转化流程如图所示。



其计算方式没有改变，相当于在卷积核做了pad为1的拓展，同时要在原始输入中也做pad为1的拓展加零处理，最终实现卷积与批处理融合代替为了一个3×3的卷积，但需注意的是，此过程只能在预测阶段使用，因为只有预测阶段参数才是已知能够当做常量来处理，训练阶段无法使用，需要逐步训练更新参数。实现了预测阶段的无损压缩，提高了预测阶段的实时性。

AleNet

CNN已经成为图像识别领域最流行的深度学习网络结构。经典网络包括AlexNet、VGGNet。AlexNet的问世也开启了深度学习研究的大门，是后续许多深度学习模型的基础。其首先提出了卷积神经网络（CNN）这一概念，这是计算机视觉史的重要时刻巨大的提升了机器视觉准确率，其历史发展如图所示，AlexNet的创新之处在于使用了非线性非饱和函数ReLU代替了原有的Softmax函数。局部响应归一化( LRN )也被用来提高准确率和泛化能力。此外，由于池化层的输出相互重叠和覆盖，使用了比轮询尺寸更小的步长，提高了特征丰富度并最小化了信息损失，AlexNet的第一层使用较大的卷积核来更好地提取位置等全局信息。前层足够大，可以获得更大的感受野，为后面的层提供更多的信息。由于信息量大，特征映射和逐像素分类器可以紧密相连，从而增强处理不同变换的能力。虽然AlexNet具有较大的卷积核，加深了网络的深度，但却增加了梯度消失的风险，且准确率较低。



VGG

所示的模型是视觉几何组在2014年ILSVRC竞赛中开发的图形处理单元。（GPU） 支持的是一个简单的卷积神经网络模型。它是一种深度学习算法，在ILSVRC竞赛中的成功率为89%。属于 20，000 多个类别的 1400 多万张图像的数据库提取从训练集（ImageNet 训练数据库）计算的平均图像值。在将VGGNet结构添加到ESA模型之前，检测精度为98.5%，但当添加VGGNet时，该精度值达到98.75%。

