卷积神经网络原理与应用-物联网导论课程调研报告

朱睿 15352454

1.摘要：

本篇报告详述了卷积神经网络的操作原理，并且介绍分析了几种较前沿的卷积神经网络结构。

2.引言：

从 2012 年夺得 ILSVRC 2012 冠军的 AlexNet[1]，紧追其后的 2014 年网络双雄 VGG[2]，GoogLeNet[3]，2016 年具有突破意义的 DenseNet[4]，ResNet[5]，再到如今百花齐放的 ShuffleNet[6]，MobileNet[7]，ResNeXt[8]，卷积神经网络从多全连接层，大卷积核的“臃肿”结构逐渐变化成少全连接层，小卷积核的“轻巧”结构，加上 DenseNet 和 ResNet 无比巧妙的网络结构设计，重新利用分组卷积，现在的前沿卷积神经网络愈发能够在低参数量，低计算量的条件下完成繁重的图像分类任务。

然而由于神经网络的“黑箱”特性，目前大多数论文都是通过实验得到的优秀结果来反推某个结构为什么能提升性能，对于数学方面的证明相对较少。所以本篇报告着重介绍卷积神经网络的操作原理以及一些新颖的网络结构。

3.卷积神经网络：

3.1 前馈神经网络，随机梯度下降法：

这一部分我在四月份的数值计算课程已经完成了相关调研与实验，详情请参考我提交的另外一份报告《数值方法在 BP 神经网络中的应用》或者参考我的 github：https://github.com/KimSoybean/My-firsht-BP-network

3.2 卷积操作：

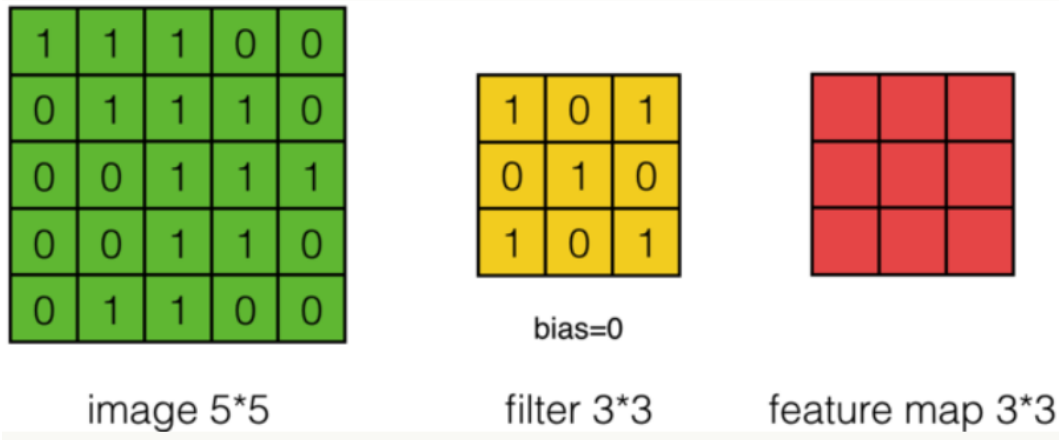
卷积神经网络设计的初衷就是应用在图像特征提取上，因此我直接使用图像与卷积核的例子来说明：

卷积核：一个卷积核就是一个三个维度的滤波器，通道数×高度×宽度，高度和宽度可以自己定义，但目前一般是使用 3×3（**提取特征**）或者 1×1（**压缩通道数或扩张通道数**），通道数和输入的图片通道数相同。回想信号系统课的卷积操作，就是数轴上滑动线性乘积求和，卷积核同理，在三维的图片上，只在图片的宽度方向滑动，遇到边缘则进入下一行继续滑动，同时在通道维度线性乘积求和。最后的输出就是特征映射。一般的图片是 3×高度×宽度，3 个通道指的是 Red，

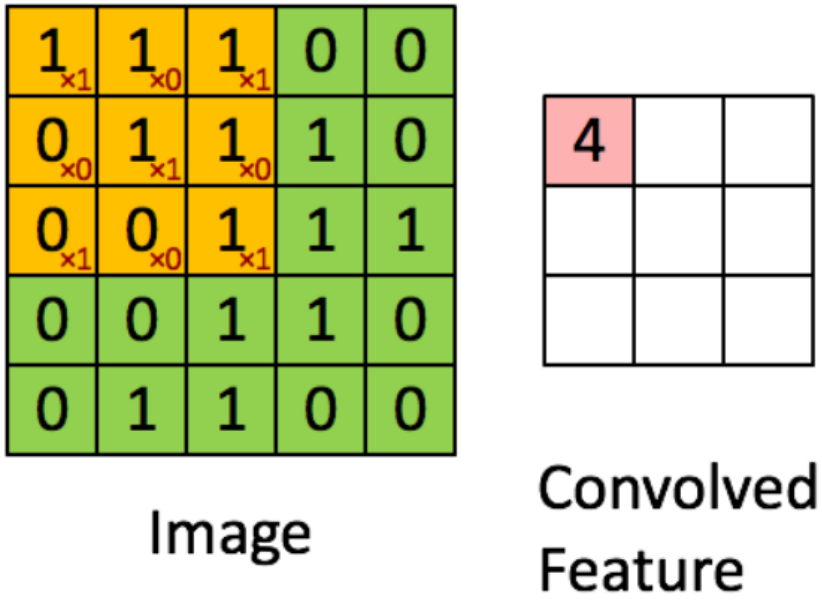
Green，Blue 三个通道，特征映射也可以暂时理解成一种图片，只不过是多通道×高度×宽度。各种深度学习的框架都有步长和边缘的参数设置，步长就是卷积核每次滑动的长度，这个步长决定了输出的特征映射的宽度和高度和每一层的感受野，边缘这个参数是为了补充图像和卷积核尺寸步长不匹配而用的。

多说无益，用图片来解释很直观。

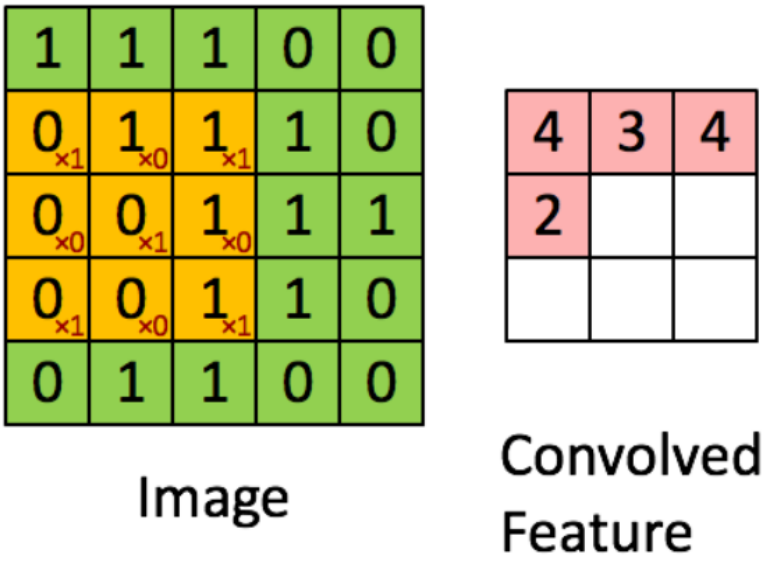
下面的图片来自于知乎（自己的图不入眼且不直观，就借用较好的例子来阐述）：



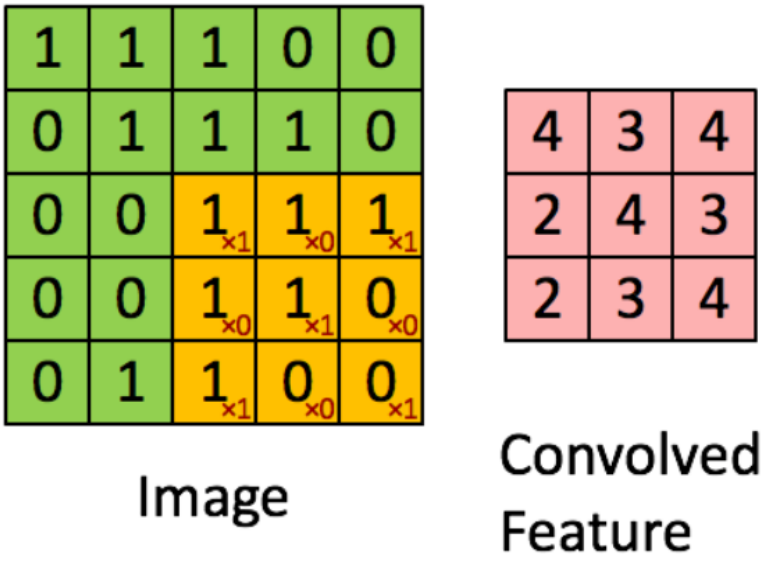
原图片　　卷积核　　即将要输出的特征映射



第一次卷积



第四次卷积(已经换行)



第九次卷积（卷积结束）

如果是刚入门的话，动手算一次会有很大的收获。可以观察到，在每次卷积的时候卷积核的参数并没有发生变化，这就是卷积神经网络的一个重要特点，**参数共享**。

3.3 池化操作：

池化，也叫下采样，其实它的滑动方式和卷积核相同，在此不赘述。池化层只消耗计算量，并不会增加参数。池化的操作可以这样想像：定义一个运算方式高度×宽度，池化会利用这个运算方式在特征映射的每个通道进行滑动采样。常用的池化有最大值池化，就是在特征映射的每个通道的高度×宽度对应的区域里面选择一个最大的值采样保留，注意是每个通道。平均池化就是取均值采样的计算。因此池化并不增加参数量。有兴趣可以查一下反卷积的操作，也就是上采样。下采样能将特征映射的高度和宽度变小，上采样相反。

4.经典网络结构介绍：

4.1 AlexNet，VGG，GoogLeNet：

AlexNet：

AlexNet 使用尺寸较大的卷积核，较多的全连接层，参数量和计算量都非常大，但是作为卷积神经网络的前辈，并且突破了沉寂已久的图像分类领域瓶颈，很棒。其使用的分组卷积，在 2016 年被发现很有价值，在减少计算量和参数量方面非常有效

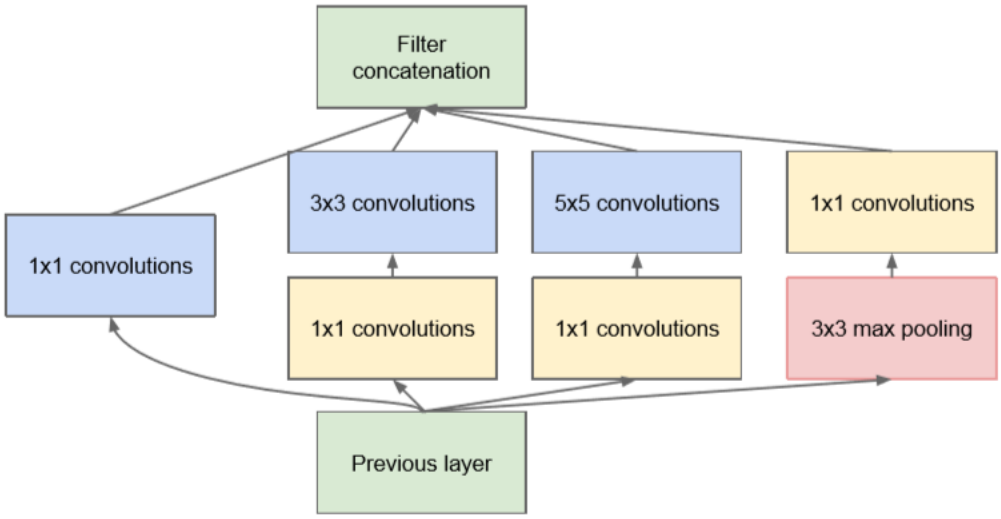
VGG：

VGG 网络不使用大尺寸卷积核，通体网络采用 3×3,1×1 的卷积核，网络层数不多，但是非常“臃肿”，每一层的通道数很多，也有较多的全连接层。参数量运算量都比较大。我个人认为 VGG 网络有一种稳定性，就是因为较大的通道数，使得模型较为稳定，有几种前沿的目标检测框架 FasterRCNN[9]，SSD[10]就是用 VGG 为主干网络的。

GoogLeNet：

谷歌的网络有 4 个版本，这里指的是 Version1。谷歌的论文很好，每次都能够给出许多实验结论，调参经验。Version1 大量使用了 1×1 结构，作为**压缩通道数或者增大通道数**的手段（因为 1×1 卷积不增大感受野），来降低参数量和计算量，虽然这种结构并不是谷歌第一次提出的，详情可以参考 Network in Network[11]这篇论文。Version1 还有一个亮点，它的每一个模块将具有多个不同感受野的特征映射输出到下一个卷积层，实现了某种意义上的特征重用。谷歌对

这种结构很执着，一直到 2017 年还在坚持这样的结构。然而我认为，的确有特征的重用，但是本质上还是增大了每一层输入的通道数，增加了更完备的统计信息，还不如像 VGG 干脆利落，直接增大通道数。GoogLeNet 比 VGG 的参数少了许多，是因为引入了 1×1 卷积的压缩方式并且减少了全连接层。



GoogLeNet Version1 的一个模块

4.2 Batch Normalization：

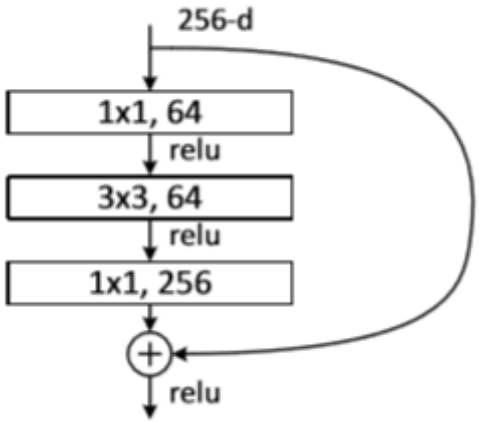
Batch Normalization[12]（后面简称为 BN）是 GoogLeNet Version2 的论文提出的一种正则化手段，经过后面大量的网络，实验证明了 BN 的正确性和适用性，2016 年之后的新网络基本都会在每一个卷积层之后带上 BN 层，使用批处理的统计特性将上一层输出的特征映射的分布拉回方差为 1，均值为 0 的正态分布上，这样能够完全利用 sigmoid 激活函数的非线性部分，使得反向传播的时候梯度值适中而不是趋于 0（层数加深会出现梯度弥散现象，收敛变慢），然而如果每层的分布都是（0,1）的正态分布的话，那么整个网络和一个线性函数相差无几，所以作者设置了要学习的参数，即在强行拉回（0,1）正态分布之后加入微小偏移，再一次重新分布，只不过改动较小。BN 层在一定程度上解决了梯度弥散问题（即梯度消失，随着层数增加梯度会逐渐减小，趋于 0），我在实验里面发现，BN 层不仅能加速收敛，还有提升模型准确率的作用

4.3 Residual Network

ResNet 是第一个上 100 层的网络。为什么其他网络没有成功堆到 100 层呢？

因为梯度弥散与过拟合。什么叫做过拟合？举个易懂的例子，一个人在某一科目的两万道例题(例题是给答案的)得到了很好的成绩，但是在考同一科目的两百道题(没有答案)的时候成绩有些差，即没有学习到举一反三的程度。两万题就是训练集，两百题就是测试集，过拟合就是因为模型参数量太大，导致模型过度依赖训练集。而卷积神经网络在没有减参数的技巧下直接增加许多层的话，参数会增长很快，容易过拟合。而梯度弥散则会造成反向传播的梯度趋向于 0，使得收敛变慢，网络无法收敛也就谈不上成功建模。我没有细致研究过梯度弥散，有兴趣可以研究这篇论文：https://arxiv.org/abs/1702.08591。

ResNet 很好的解决了这两个问题，一方面使用了 GoogLeNet V1 的 1×1 卷积核大量减少参数（1×1 操作不仅减少参数，还增加了层数），另一方面使用了 GoogLeNet V2 的 BN 层，减缓梯度弥散。当然，最重要的是引入了残差结构，使得收敛加快，加速收敛在本篇论文中描述的话需要很大篇幅，可以参考原作者的另一篇论文[13]：https://arxiv.org/abs/1603.05027

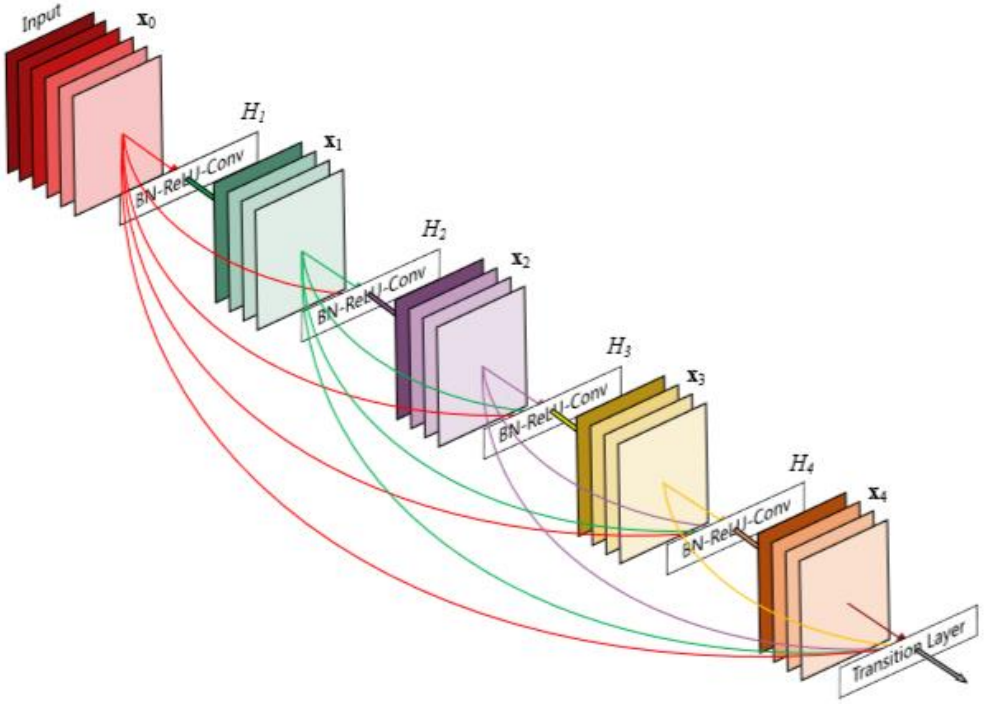


残差结构

4.4 DenseNet:

DenseNet 很巧妙地减少了参数和计算量，我们用一种思考的方式来想，如何不增大卷积核的输出通道个数，而增大输入下一层的特征映射的通道数呢。DenseNet 给我们的答案就是，使用前面层输出的特征映射不就可以了。这样不仅可以有效增强每一层输入的统计特性，还能够重利用前面层输出的特征（**存疑**）

DenseNet 的作者就是利用这样的技巧，设置一个模块，每个模块内的每一个 3×3（提取特征的卷积层）输出的特征映射大小相同，每一层输出的特征映射都会在通道维度上与模块内的后面每一层进行拼接，从而有效增大了通道数量，增强统计特性。当然，DenseNet 也使用了 BN 层加速收敛，1×1 的卷积核来压缩参数。



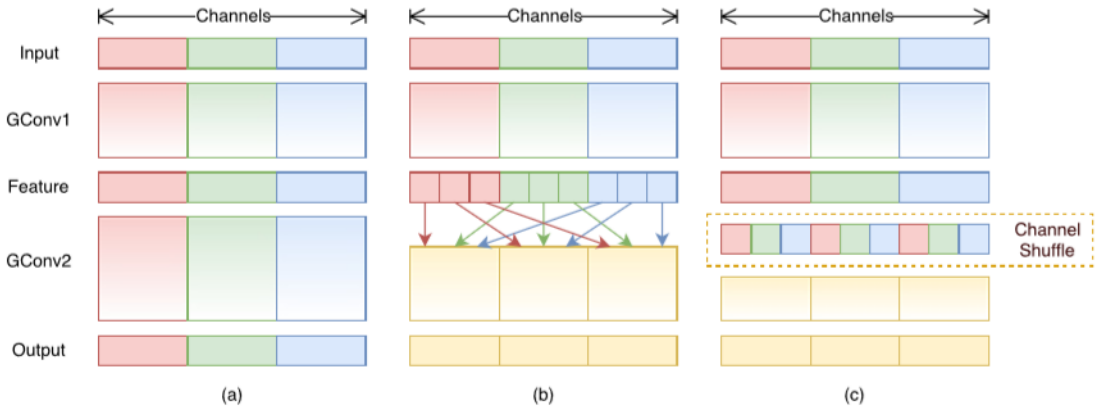
DenseNet 的一个模块

DenseNet 虽然有效的减少了参数量和计算量，但是训练模型时需要的显存是很可怕的。DenseNet 原文说 DenseNet 实现了特征的重用，我对此表示存有疑问（自己的实验也有证明）。我的观点和 Dual Path Network[14]论文的观点一样，**Resdial 结构实现了特征重用，Dense 结构探索产生了新特征**。Dual Path Network 就不在此详述了，文章将 ResNeXt(使用分组卷积减少参数)与 DenseNet 结合，得到了性能很好的网络。文章的亮点在于使用了 RNN 的方式得出了一个结论：Resdial 结构实现了特征重用，Dense 结构探索产生了新特征。我自己也推导过这个问题，很同意这个观点。

4.5 Shuffle Net

分组卷积是一个很有趣的结构，就是卷积核按照组数一分为多个，分别进行卷积，要注意的是**分组的卷积核参数不共享**，是不同的参数。这一点很重要，我在我自己的实验中尝试过，收敛非常慢。最后产生的特征映射会在通道维度拼接起来，在这一拆一拼之间，参数减少了，计算量减少了。实际上最近使用分组卷积的网络都有一个共性，就是将分组卷积省出来的参数，计算量，都均分给不同层数（直接增加通道数），还是先压缩后膨胀的老方法，只不过压缩的方式换成了分组卷积。

shuffle 这个单词是什么意思呢？洗牌，重排列。因为分组卷积之后再分组卷积会造成通道被分割，信息不能很好地流通，所以作者想出一个办法进行重排列。从而更好地使用了统计信息。



Shuffle Net 的重排列方式

5.结语：本次论文写到这里已经够了，算是我对自己两个月科研经历的一个小总结，还有许多有趣巧妙的结构还没有介绍，比如 DW 卷积，膨胀卷积，这些结构都是作者花心思，做实验想出来的。我个人认为深度学习不是调参，后传要学好线性代数，前传要学好微积分，损失目标函数要学好凸优化，调参经验要认真做实验用好画图工具，增加新操作方式需要熟练 C++。我不认为深度学习是一个黑箱，媒体整天自嗨 AI 是为了赚流量，只有学者和工程师在努力着，学者还要承受那一份经济压力。

6. 引用文章：

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *neural information processing systems*

[2] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *international conference on learning representations*,.

[3] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *computer vision and pattern recognition*,, 1-9

[4] Huang, G., Liu, Z., Weinberger, K. Q., & Der Maaten, L. V. (2016). Densely Connected Convolutional Networks. *arXiv: Computer Vision and Pattern Recognition*,

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *computer vision and pattern recognition*,, 770-778.

[6] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. arXiv preprint arXiv:1707.01083, 2017. 1, 2, 3, 5, 7

[7] A.G.Howard,M.Zhu,B.Chen,D.Kalenichenko,W.Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 1, 2, 7

[8] Xie, S., Girshick, R. B., Dollar, P., Tu, Z., & He, K. (2016). Aggregated Residual Transformations for Deep Neural Networks. *computer vision and pattern recognition*,, 1492-1500.

[9] Ren, S., He, K., Girshick, R. B., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149.

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *european conference on computer vision*,, 21-37.

[11] Lin, M., Chen, Q., & Yan, S. (2014). Network In Network. *international conference on learning representations*,.

[12] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *international conference on machine learning*,, 448-456.

[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. *european conference on computer vision*,, 630-645.

[14] Dual Path Networks
Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, Jiashi Feng
https://arxiv.org/abs/1707.01629