# CNN

CNN的三个特征

* Some patterns are much smaller than the whole image
* The same patterns appear in different regions.
* Subsampling the pixels will not change the object

自行定义Filter Matrix: Each filter detects a small pattern (3 × 3). 准确估计特征大小很重要？

如何确定Stride及padding大小

卷积层与全连接层

卷积层主要用来提取特征，卷积层后一般链接softmax层，后对特征结果进行flatten，注入到全连接层的神经网络中。

**Each filter is a channel**

The number of the channel is the number of filters

CNN-Summary

* Problems of fully connected neural networks 全连接神经网络

即全连接层，输出与输入单元全交互；随着输入图片的大小极速增长；有一定距离的不同像素相关性很差

* Locally connected neural networks 本地连接神经网络

Sparse connectivity: a hidden unit is only connected to a local patch (weights connected to the patch are called filter or kernel) 稀疏连接特性，每个图像只与一个本地块（filter或者kernel连接）

**感受野** Inspired by biological systems, where a cell is sensitive to a small subregion of the input space, called a **receptive field.** Many cells are tiled to cover the entire visual field.

受生物系统的启发，在该系统中，细胞对输入空间的一小部分区域敏感，称为感受野。许多单元被平铺以覆盖整个视野。即局部特征就能判断整体

The design of such sparse connectivity is based on domain knowledge.

基于域知识来设计稀疏性连接

从空间局部模式，堆叠多层信息到filters上，致使图像变成全局模式。隐藏层结点的层数越高，对应的感受野也越大。

感受野被定义为卷积神经网络特征所能看到输入图像的区域，换句话说特征输出受感受野区域内的像素点的影响。即不同神经网络层受输入图像大小的影响，输入图像的大小定义为感受野。

CNN的参数主要集中在全连接层上

如何判断filter及image 的depth？

Filters always extend the full depth of the input volume

其中，activation map是指convolve (slide) over all spatial locations，其中，大小等于(ImageSize-FilterSize) / Stride + 1

padding = “value”，N = [(W-F+2P)/S]+1，这里表示的是向下取整再加1即求商忽略余数

6 filters of size 5x5x3 其中，6是数量，即想要提取的feature数目

Relu与sigmoid等激活函数的区别

Pooling layer (max pool with 2x2 filters and stride2)

• makes the representations smaller and more manageable

• operates over each activation map independently

一些经典的CNN结构

例如AlexNet, VGG, GoogleNet, ResNet, Network in Network, Wide ResNet, ResNeXT, Stochastic Depth, DenseNet, SENet, FractalNet, SqueezeNet, Inception

Inception网络——GoogLeNet网络的综述

一、Inception网络（google公司）——GoogLeNet网络的综述

获得高质量模型最保险的做法就是增加模型的深度（层数）或者是其宽度（层核或者神经元数），但是这里一般设计思路的情况下会出现如下的缺陷：

1.参数太多，若训练数据集有限，容易过拟合；

2.网络越大计算复杂度越大，难以应用；

3.网络越深，梯度越往后穿越容易消失，难以优化模型。

解决上述两个缺点的根本方法是将全连接甚至一般的卷积都转化为稀疏连接。为了打破网络对称性和提高学习能力，传统的网络都使用了随机稀疏连接。但是，计算机软硬件对非均匀稀疏数据的计算效率很差，所以在AlexNet中又重新启用了全连接层，目的是为了更好地优化并行运算。现在的问题是有没有一种方法，既能保持网络结构的稀疏性，又能利用密集矩阵的高计算性能。

Inception采用不同大小的卷积核，意味着不同大小的感受野，多尺度下的特征融合。

Inception的作用：代替人工确定卷积层中的过滤器类型或者确定是否需要创建卷积层和池化层。

Inception V2结构

作者提出可以用2个连续的3x3卷积层(stride=1)组成的小网络来代替单个的5x5卷积层，这便是Inception V2结构。用两个3x3代替了5x5的大卷积（首次出现在VGGNet），用于降低参数量，降低过拟合，以及提出了BN方法。

BN：有效的正则化方法，对每一个mini-batch数据内部进行标准化处理，使输出规范化到N(0,1)正态分布。BN起到了正则化作用，并减少或者可以取消Dropout，优化网络结构。每个特征都有均值为0，方差为1的分布。

### 类似于BN的之前的正则化方法：whitening，白化

白化（Whitening）是机器学习里面常用的一种规范化数据分布的方法，主要是PCA白化与ZCA白化。白化是对输入数据分布进行变换，进而达到以下两个目的：

使得输入特征分布具有相同的均值与方差。其中PCA白化保证了所有特征分布均值为0，方差为1；而ZCA白化则保证了所有特征分布均值为0，方差相同；去除特征之间的相关性。

BN层的作用及使用方法：

解决的问题：Internal Covariate Shift内部协变量移位——在深层网络训练的过程中，由于网络中参数变化而引起内部结点数据分布发生变化的这一过程被称作Internal Covariate Shift。

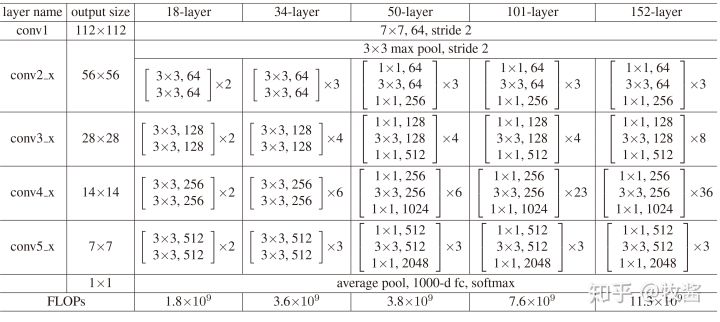
实现了网络中层之间的解耦，允许每一层独立学习，提高整个网络的学习速度。

以及后续更新了Inception V3, Inception V4 and Inception ResNet。

Inception V3中将NxN的卷积核用Nx1即1xN的两个卷积核进行代替，在网络前期的分解效果并不好，适用于中度大小的feature map，适用于mxm大小（m建议在12到20之间）。

ResNet

ResNet 已定义的网络架构



可以将这五种ResNet分为两类：(1) 一种基于BasicBlock，浅层网络ResNet18, 34都由BasicBlock搭成；(2) 另一种基于Bottleneck，深层网络ResNet50, 101, 152乃至更深的网络，都由Bottleneck搭成。

每种ResNet都是4个layer构成，其中block均指四层结构

model = ResNet(Bottleneck, [3, 4, 6, 3], \*\*kwargs)

一般结构layer(Conv layer, Max pooling layer, BN layer and Relu)

**PyTorch ResNet代码中的downsample的作用**

实现残差连接，将前一层的原始数据downsample，搭配后面conv层的数据，实现同维度残差连接

模型压缩

模型压缩方法，可以分为权重优化和模型结构设计。权重优化也可分为剪枝和量化。

剪枝，是将模型中冗余的权重去掉，以达到模型瘦身的目的。比如，Deep compression[1]，通过权值剪枝、权值量化和权值编码，能够将模型的体积减小49倍。

权重量化，则将权值以低码率进行存储，从而减小模型的体积，比如，XNornet[2]模型，对输入的feature maps和权值均进行二进制量化，实现58x的模型压缩和32倍的加速。

如何解决过拟合等验证集结果不够好的情况：

调整Learning Rate或者gradient算法（及optimizer）

或者增加regularization或者dropout层

以及通过data augmentation

调整初始化参数方法也能帮助模型实现加速收敛

## data augmentation

翻转变换 flip、随机修剪（random crop）、色彩抖动（color jittering）、平移变换（shift）、尺度变换（scale）、对比度变换（contrast）、噪声扰动（noise）、旋转变换 / 反射变换 （rotation/reflection）等

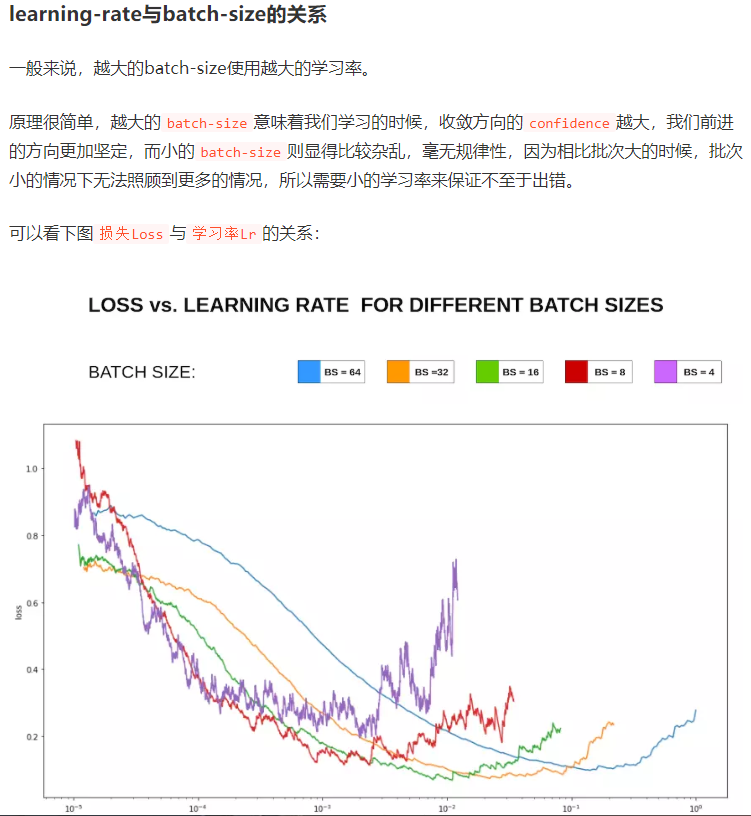
图像翻转，图像切割与图像白化

## 模型改进

1. 权重衰减（weight decay）：对于目标函数加入正则化项，限制权重参数的个数，这是一种防止过拟合的方法，这个方法其实就是机器学习中的 l2 正则化方法，只不过在神经网络中旧瓶装新酒改名为 weight decay [3]。
2. dropout：在每次训练的时候，让某些的特征检测器停过工作，即让神经元以一定的概率不被激活，这样可以防止过拟合，提高泛化能力 [4]。
3. 批正则化（batch normalization）：batch normalization 对神经网络的每一层的输入数据都进行正则化处理，这样有利于让数据的分布更加均匀，不会出现所有数据都会导致神经元的激活，或者所有数据都不会导致神经元的激活，这是一种数据标准化方法，能够提升模型的拟合能力 [5]。
4. LRN：LRN 层模仿生物神经系统的侧抑制机制，对局部神经元的活动创建竞争机制，使得响应比较大的值相对更大，提高模型泛化能力。

## 改变学习率与Batch Size，提升模型性能

随后面epoch的增加，逐渐减小学习率



余弦退火(cosine annealing)和热重启的随机梯度下降

余弦就是类似于余弦函数的曲线，退火就是下降，余弦退火就是学习率类似余弦函数慢慢下降。

热重启就是在学习的过程中，学习率慢慢下降然后突然再回弹(重启)然后继续慢慢下降。

## 差分学习率与迁移学习

迁移学习：例如通过ImageNet的预训练权重来学习两个物体二分类的权重；

卷积层设置的学习率应该更低一些，而全连接层的学习率可以适当提高：这就是差分学习率的意思，在不同的层设置不同的学习率，可以提高神经网络的训练效果，具体的介绍可以查看下方的连接。

## 权重初始化方法

常用权重初始化方法：kaiming\_normal或者xavier\_normal

## 交叉验证

交叉验证是Kaggle比赛中特别推崇的一种技巧，我们经常使用的是5-折(5-fold)交叉验证，将训练集分成5份，随机挑一份做验证集其余为训练集，循环5次，这种比较常见计算量也不是很大。

## 多尺度训练

多尺度训练是一种直接有效的方法，通过输入不同尺度的图像数据集，因为神经网络卷积池化的特殊性，这样可以让神经网络充分地学习不同分辨率下图像的特征，可以提高机器学习的性能。也可以用来处理过拟合效应，在图像数据集不是特别充足的情况下，可以先训练小尺寸图像，然后增大尺寸并再次训练相同模型，这样的思想在Yolo-v2的论文中也提到过

多尺度训练并不是适合所有的深度学习应用，多尺度训练可以算是特殊的数据增强方法，在图像大小这一块做了调整。如果有可能最好利用可视化代码将多尺度后的图像近距离观察一下，看看多尺度会对图像的整体信息有没有影响，如果对图像信息有影响的话，这样直接训练的话会误导算法导致得不到应有的结果。