week01

CV in self-driving cars, face recognition, GAN

SNN（脉冲神经网络）

CNN

1. Data：Augmentation, Selection Strategy
2. Network: Initialization strategy, learning strategy, optimization strategy, framework-modeules, Layers.
3. Loss: Loss type, loss selection, update strategy, evalution

CV

1. low level: 从图像能直接得到的信息，basic operations, Convolution
2. mid level：链接high level和Low level的桥梁，feature extraction, Feature points, Corner point, SIFT
3. high level：bounding box，从2维怎么构建3维信息，real CV tasks by using CNN, classification

Fantastic Transformation, Style Transfer, Image Generation, 2D To Depth, 3D objects from 2D, Slam

pandapow

Research: AutoML, Acceleration,

Applications: **Image**/Video Classification, Object Detection, Keypoint Detection, **Recognition**, **Segmentation**, Voxcel, Tracking, 2D-3D/3D-2D, Image Captioning, **Image Transfer**, Mixed Inputs, GAN, Slam.

Engineering: Modified Models, **Light Models,** Accelerating Algorithm

Python主要针对有research意愿的人。**C++**主要针对有工程意愿的人。

Caffe、**PyTorch**上手快

**Coding Preparation**: 200/300+题，medium难度40min做对4/5道，Lintcode/Leetcode

LintCode：<https://www.lintcode.com/problem/> 账号：wjq332608421 密码：wjq613613

Tiny Project, Kaggle：Cancer Detection, Cactus Identification, Protein Atlas Image Classification.

Reflection Removal, Super Resolution, Face Frontalization, Add/Remove Sth

Video -> Sound, Video + Sound -> Classification, 2-Stream Videos -> Classification

CVPR, ECCV, ICCV – 偏应用

NIPS – 偏理论

PAMI – 顶刊，较滞后

SIGGRAPH, arXiv, Kaggle

冈萨雷斯-《数字图像处理-第三版》

合恩-《矩阵分析》

Bishop-《Pattern Recognition and Machine Learning》

week02

Convolution:

1-dim convolution: kernel (需180度翻转)，相关性操作correlation (不需180度翻转)：1）kernel很多时候是对称的，翻不翻转都是一样的；2）操作行为类似；3）kernel的值是学习学出来的，对转不转是没有意义的。

2-dim convolution: 补0（padding）数为kernel/2取整的值时，卷积后的数据大小不变。

Image convolution: 图像分不同的channel，每个channel都是一个2维的，每个channel需分别和各自的kernel做卷积。

First-order derivation（一阶导）：相邻像素的差。x为坐标，f(x)是灰度值。相邻 – 人为规定，4邻或者8邻。

Second-order derivation（二阶导）

上式的严格定义如下：

– 拉普拉斯算子

一阶导、二阶导在图像中找信息，能找到角点、边缘。

边缘（Sobel）：二阶导更敏感，二阶导能产生双边缘的效果。-> Sharpen（锐化，Laplacian）：让图像看起来更加清晰（=原图+边缘）。

卷积能快速算一阶导、二阶导，所以用卷积的方法。

钝化：让图像更加模糊。 -> Gaussian Kernel（加权平均，平滑处理）

一阶导：

二阶导：

二阶导拆分成两次一阶导，计算速度更快。

Bilateral Filtering：非边缘部分变平，同时保留边缘。

Feature Points/descriptor：represents pf objects/pixels.

什么算好的特征点？

1）非常具有信息量；

定义一个window：w

指窗口的整体平移量。V的值就是平移后和平移后的灰度值相减。

如果(x,y)对应的是角点，那么新的窗口对于(x,y)就有更大的响应。

泰勒展开：

为高阶无穷小项。

若上式大于某一个阈值thre，则判断其为角点。k的值一般为0.04。为矩阵的特征值。其中：

2）旋转不变性、对光亮不敏感

3）尺度不变性（距离远近看一个物体，物体类别不会变）

SIFT：Scale-Invariant Feature Transform.

1. Generate Scale-Space: DoG（Difference of Gaussian）：用一系列的高斯核对图像进行平滑，模拟出不同尺度下图像的样本，然后将模拟出的高斯图像做差，得到所谓的轮廓线；在不同尺度的空间下找到极值点，找到极值点后再对极值点进行精确的定位，消除所谓的边缘棱角，然后生成descriptor。-> 在尺度层面上找边缘。
2. Scale-Space Extreme Detection：尺度空间内的极值点（该点跟周围的点相比，最大或最小）。20m看到的图中的点，除了与自身所在图相邻的点进行比较外，还要与10m、40m看到的图像对应的点进行比较。
3. Accurate Keypoint Localization



指尺度。

1. Eliminating Edge Responses：海森矩阵。

DoG：too sensitive to edge, unstable.

1. Orientation Assignment：解决旋转不变性。
2. Keypoint Descriptor：对角度、尺度、边缘没有影响，而且非常精准。

FAST（faster version of sift）、SURF、BRIEF、HoG（detection）、OrB（SLAM）

像素值梯度、梯度的幅值

中值滤波（求中值）

RANSAC

homegraphy matrix

Feature Pyramid Networks：尺度上的变化有金字塔的效果。

SIFT的流程：如何做到尺度不变、旋转不变的。

泰勒展开

插值问题

Bag of Words（BoW）

Image Stitching

1. Find feature points in each image
2. Use RANSAC to find keypoint matches
3. Use homography matrix to get transferring info
4. Merge two images.

week03

well-posed（1、A solution exists; 2、Solution is unique；3、Solution’s behavior is continuous） \ ill-posed（如给黑白照片上颜色，解不唯一）

Supervised Learning

1、Who supervises who? A standard supervises the system.

Regression: Object Detection, Keypoint Detection（人脸关键点检测）, Prediction。

Classification: Image/Video Classification, Segmentation（分割，Instance Segmentation, Semantic Segmentation）。

识别 \ 分类 \ 检测（Bounding Box）

Unspuervised Learning（没有标准答案）

Q: How to type a video within 1,000,000,000 videos in a short time?

A: Classification. 1. 标注成本问题。

A: Clustering: 生成树状结构

**Linear Regression**：训练的过程就是寻找参数的过程。

Cost：hypotheses和reality之间的差距，目的就是最小化Cost。

Gradient Desceng Algo

while not converge {

}

过大，容易导致梯度爆炸。

Extension – Multiple Variables：

表示第i个样本。

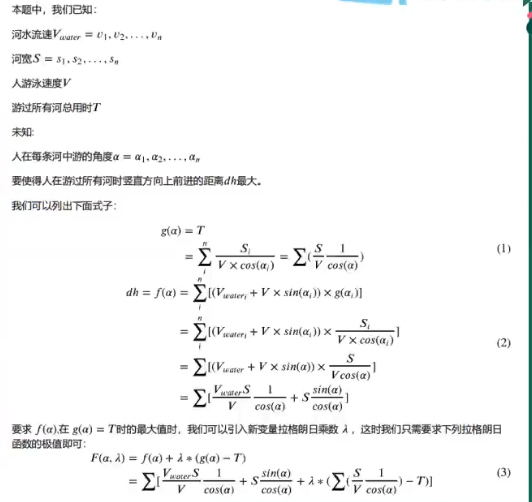
Feature Scaling

**Logistic Regression**: Classification

Cost Function（二分类Cross entropy，对于多分类使用Softmax）:

Sigmoid函数：，其导数为：

Multi-Classes: Softmax



week04

**Neural Network**

Activation Function：非线性作用。

：activation of unit in layer 。

神经元个数怎么确定？经验。

Classification：其输出表示属于哪个类别的概率。

：sample ID；

：# of samples;

：output ID

：# of outputs

若层数较多，通过Gradient Descent求参数变得不现实了。

**Back Propagation**：Chain Rule。（会推导）

Chain Rule存在的问题：

1. 梯度消失/爆炸。

**Regularization（惩罚项）**：防止过拟合，相当于让值变小，让拟合曲线变得更加平滑。

对于线性回归，加入惩罚项后，Gradient Descent项变为：

L2 Regularization：，Ridge Regression

L1 Regularization：，Lasso Regression

L2和L1的异同、特点？

|  |  |
| --- | --- |
| L2 regularization | L1 regularization |
| Computational efficient due to having analytical solutions | Computational inefficient on non-sparse cases |
| Non-sparse outputs | Sparse outputs |
| No feature selection | Built-in feature selection（能使一些参数为0） |

week05

RANSAC算法用于拟合和匹配：<https://www.jianshu.com/p/4b5de31a8894>

fast median filter

图像分割：阈值法（cv2.threshold）、漫水法（cv2.floodFill）、水域法、金字塔分割、均值漂移分割、背景减除。

阈值法：大津阈值（cv2.THRESH\_OTSU）、自适应阈值。

图像滤波：均值滤波、加和滤波、中值滤波、高斯滤波、双边滤波、盒子滤波。

边缘检测：Sobel、Scharr、拉普拉斯、Canny、形态学梯度。

遥感图像：GDAL库。

医学图像：呈层级数据

彩色空间：RGB、CMY、CMYK、HIS。

图像变换：拉伸、缩小、Warp、旋转；一致调整变换或均匀调整变换、图像金字塔、非均匀映射、仿射变换、透视变换；极坐标映射、对数极坐标、任意映射。

图像常见的传统特征描述：LBP、Gabor、HOG、SIFT、Hough变换，颜色直方图、频谱特征提取、圆周频率滤波，矩特征、Harris角点检测

week06：SVM、非监督学习

**SVM（Support Vector Machine）**

，

，，系数不影响平面性质，可将替换为1。截距项对应。

上两式可转换为：

两个平面之间的margin为：

等价于：，且需满足。

拉格朗日乘子法：将约束条件与目标函数融合在一起。

Soft margin：，相当于将平面整体平移了一下。

其中，越大，对误分类的惩罚越大，更可能过拟合。其可看作正则化系数的倒数关系。

Kernel：能产生非线性的效果，类似激活函数。

令，其为高斯核。

越大，kernel更平滑，拟合能力更低，bias会增加，variance会减小。

只要样本越趋近于，值越趋近于1；越远离，值越趋近于0。

linear、polynomial、Gaussian（RBF）、Laplacian（在高斯核基础上去掉2次方）、Sigmoid。

n：# of features；m：# of samples.

Linear Regression

Logistic Regression / Classification

L2 Loss

Sigmoid Function

Cross Etropy / Multi-Label

Neural Network

L1 / L2 Regularization

Overfit / Zigzag

BP / Cost Function / Hypothesis

Gradient Vanishing ./ Explosion

SVM / Kernel / Derivation

Normal Equation

Unsupervised Learning

**K-Means**

收敛：新找德中点与之前德中点很靠近。

需要自己手动实现。

初始化方法：K-Means++[2007, Arthur & Vassilvitskii]

离得远，被选为新中心点的概率大。

K chosen：elbow method / ISODATA

参考sklearn各种k-means方法

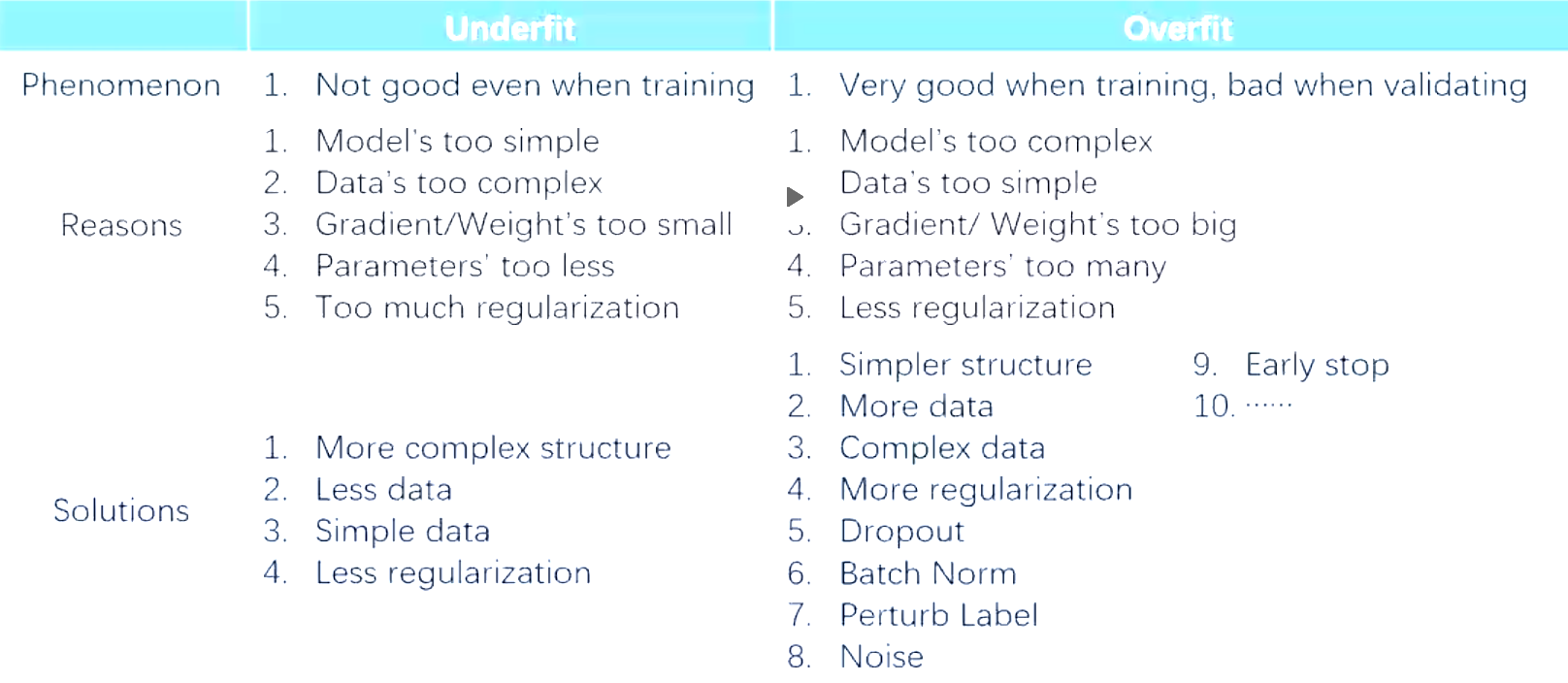
Training / Validation / Test Set

Validation：1）训练时起到测试的作用；2）训练超参数。

交叉验证：将训练集分成9份，拿出一份当作Validation，这样验证9次。以此来决定超参数。将此方法训练得到的模型用于Test数据上。

**Underfit / Overfit**

有时候训练一定轮次之后，training loss先减小后增加，亦出现Overfit。



Perturb Label：按样本0.1比例将实际的标签为1，将其标为0，即进行随机标注。

High Bias: Underfit.

High Variance: Overfit.

正则项的越小，越容易过拟合，产生的Loss越小。

|  |  |
| --- | --- |
| To reduce high variance | + |
|  | -complexity |
|  | +samples |
| To reduce high bias | - |
|  | +complexity |

Decision Tree: ID3 / C4.5 / CART（Gini Index）

week07 Outline & Layers

分类、**检测**、Image Transfer（图像增强）。

CV两个项目选一个，具体项目是？

1. 图像拼接；2）LocNet；3）动物多分类实战；4）3D BBX；5）人脸关键点检测（Open Pose）；6）人脸转正；7）照片反光移除。

convolution + pooling layers + 非线性

Mask R-CNN：检测（画出Bounding Box）、分类、分割（像素级的分类）

Wild Face、3D图像分类，Image Caption（看图说话）

CNN：Data -> Network -> Loss

Data：Augmentation、Selection Strategy

Network：Initialization Strategy、Learning Strategy、Optimization Strategy、Framework->Modules->Layers

Loss：Loss Type、Loss Selection、Update Strategy、Evaluation

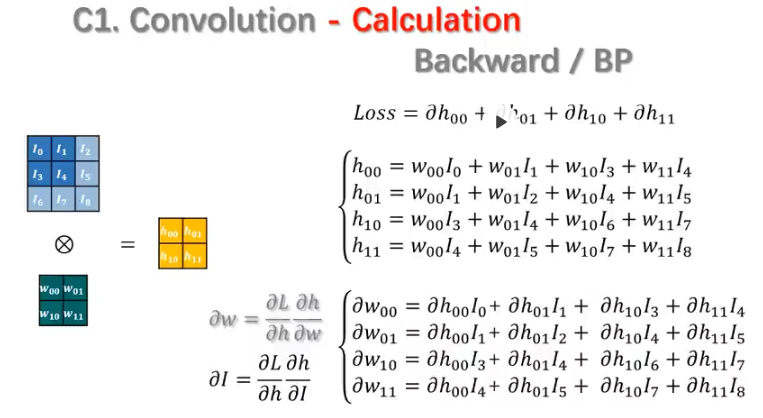
Fundamental Layers:

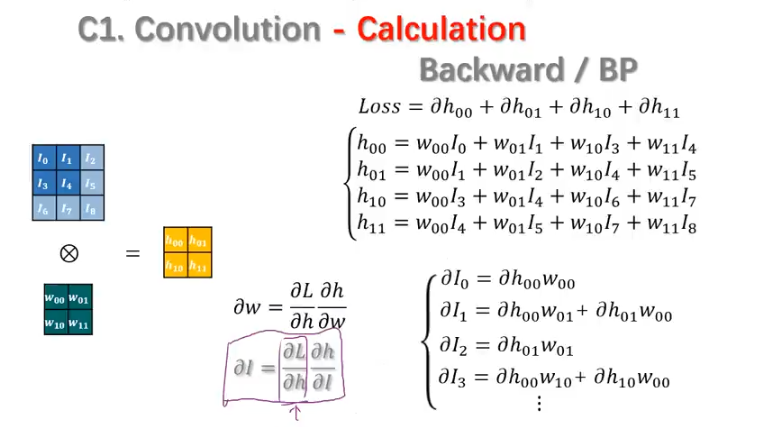
Convolution

：kernel的个数。（Padding,；Dilation – 中间隔了几个眼，默认为1；Stride）

做卷积的时候，实际采用矩阵相乘的形式。。

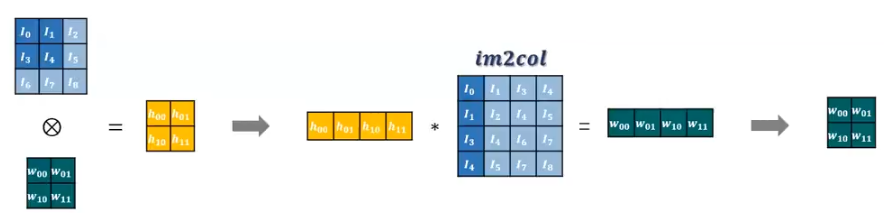
其中，为transposed weight matrix。为1列9行的数据。





Transposed Conv / Deconv / Fractionally Strided Conv：由小图变成大图。（分割）

im2col：



Dilated Conv：Get better view（用于分割）

C3D / Convolution 3D

W\*H\*3\*4：其中4指Temporal info。

kernel为：kw\*kh\*3\*3，第2个3指时间维度上按3帧3帧进行卷积。batch用16帧比较好。

作业：

Write code of 2D image median filtering。注意1）Padding；2）Stride；3）Kernel Size；4）How to get the median number。

Depthwise Convolution（可分离卷积）

Group Convolution

Deformable Convolution（可形变卷积）

ReLU – nonlinearity

1. 非线性是啥？
2. 目的是啥？
3. ReLU是什么？
4. 以及为啥要用ReLU？不会梯度消失，也不会梯度爆炸；easy to implement, calculate。

缺点：负半轴为0，人为消除掉了一半的信息。

1. Which one is better?

ReLU could be 1st choice, PReLU looks better, Never use sigmoid in middle layers.

1. Any other types？ Maxout, SeLU

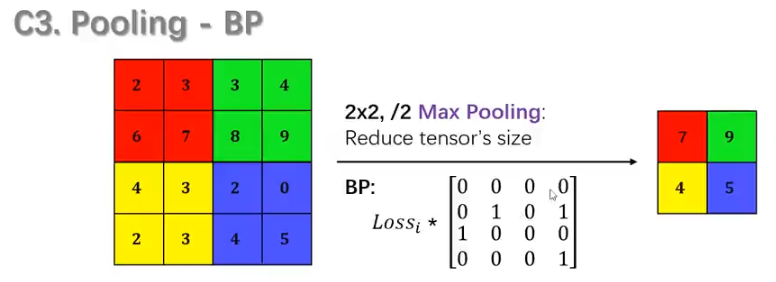
pooling

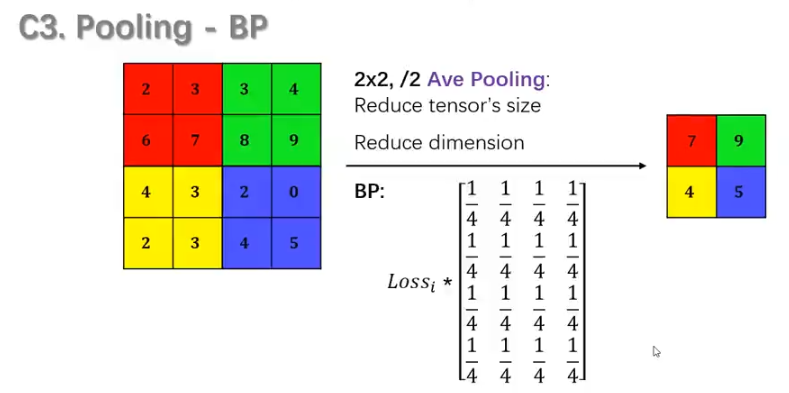
1. What is pooling？
2. 作用是什么？减小tensor的大小。
3. 有哪些池化？

max pooling：获得具有最高响应的像素值。

avg pooling：reduce dimension，get feature vector，used to change tensor’s shape at the end of the network.

1. How to do BP?





Fully connected layers

将previous tensor（Feature map）进行flatten，再进行Neural Network处理。Flatten后的每个点称为response。

Too many parameters, calculations, too mush resources needed, 很容易过拟合。

Functional Layers

1. Dropout Layer

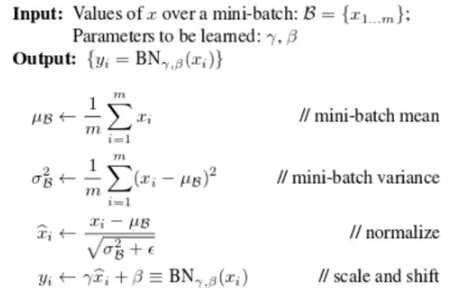
训练时随机地按一定比例p去掉一些连接。测试时，使用比例p乘以各层节点获得。两者的期望一样。但为了在测试的时候降低乘以p的计算量，可在训练的时候，将按比例去掉一些连接后的点除以p，以达到一样的效果。

1. Batch Normalization

网络越深，训练的效果越不好，梯度消失的现象越严重，因此而引入BN。

什么是比较好的可用于训练的数据？层和层之间应独立同分布、no internal covariate shift.

BN即解决数据Internal Covariate Shift的问题及保持层和层之间独立同分布，该问题主要是由于Sigmoid存在梯度饱和的现象使得梯度消失。



好处：

1. 可以用稍微大点的learning rate；
2. 加速训练
3. 不容易过拟合，能减少dropout的使用，也没必要使用data augmentation。

Usually：Conv -> BN -> ReLU -> Pool

Some argus： Conv -> ReLU -> BN -> Pool

1. Group Norm
2. Instance Norm
3. Layer Norm

Q：1）在什么领域内用？2）什么时候去用？3）怎么去用？4）相互间的比较？

week08 Network Structure

trello总体项目说明。

GAN：本身是强理论推导的。

1986 Hinton BP

1989 Yann LeCun把BP算法应用到CNN，1998年提出LeNet，现代CNN的鼻祖。Hinton在此基础上发展出Dropout和ReLU。

2012 AlexNet（Hinton）

backbone架构：

1. Deeper：VGG-16/19，
2. Module：不同的层归结出不同的功能性单元，GoogleNet

功能性的架构

1. Detection
2. Segmentation

History 0-1st generation：LeNet-5，AlexNet，ZFNet

2nd generation：VGGNet，GoogleNet（InceptionNet）

3rd generation：ResNet，DenseNet，CapsNet（Hinton）

Feature Map：经过卷积之后的结果。

LeNet-5、AlexNet、ZFNet：无任何本质区别，只是核更大一些、层更多一些。

Input -> n \* (Conv -> ReLU -> Pool) -> 1|2 \* FC -> Output

VggNet：所有的kernel大小都是3\*3，Why？经过两次3\*3后能看到5\*5的范围，与直接5\*5相比，参数更少。

2\*Conv3 = 3\*3\*C\*2=18C

1\*Conv5 = 5\*5\*C=25C

GoogleNet（Yangqing Jia）

DepthConcat：按深度拼接，28\*28\*50与28\*28\*100拼接成28\*28\*150，拼接时要保证size是一样大的。

1\*1卷积的作用：改变channel数量。

Inception Module：网络可按照一个模块一个模块相互拼接。

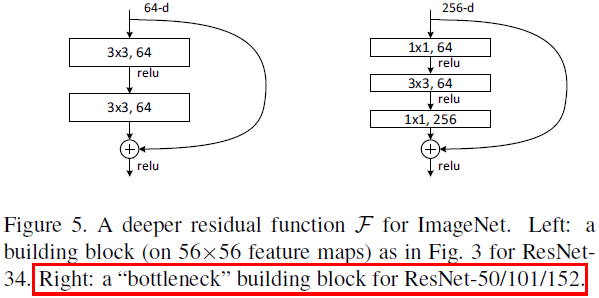
Global Avg Pool：考虑到全部的特征都能其作用，可省去全连接层（容易过拟合、参数量大、计算复杂）。7\*7\*100使用7\*7的Avg Pooling，即得到1\*1\*100的Feature Vector。

Auxiliary Softmax：在不同分辨率的情况下都能够产生结果，将这些结果融合以获得最终的结果。

The deeper the network, the more parameters, the worse results we’ll get. Why? too deep, accuracy saturated, Gradient Vanished.

ResNet（Kaiming He）：Bottleneck

skip connection / shortcut



Elementwise Addition（元素级相加）：shortcut的内容与conv后的内容之间元素级相加。

有了skip connection就不会有梯度消失的现象。

在直线A路，若conv/pool之后的size缩小为原来的一半，shortcut分支如何处理？添加convs使size也缩小一半。

Why 初始阶段使用7\*7kernel？1）在初始阶段希望获得更多的原始信息；2）大的kernel并不意味着很多的参数；3）与VGG相比，达到同样的size时，ResNet的参数明显小于VGG。

50层的ResNet有49层的卷积层，只要1层Avg Pooling层。

Small kernel always better？No

Identity Mapping In Deep Residual Networks：针对ResNet出现的问题进行结构上的改良，使层数能达到1000多层。

ResNeXt

Why ResNet is better？1）解决梯度消失的问题；2）其模型结构可以被看做模型融合。

DenseNet：1）deeper；2）fewer params；3）better results。

缺点：需要巨量的内存。

离最终结果越近的层，越重要。

Light Frameworks：1）much less parameters；2）much faster；3）focus on end devices。

History Version1：SqueezeNet -> MobileNet -> ShuffleNet -> Xception

Version 2：SqueezeNext -> MobileNetV2 -> ShuffleNetV2

Version 3: AutoShuffleNet

优化网络结构：shuffle net

直接减少网络参数：squeeze net

优化conv：mobile net, winograd（算法上的改变）

Delete FC：squeeze net, lightCNN

SqueezeNet：1）晚点儿用pooling；2）less channel number；3）use more 1\*1 kernel。

Structure：Fire Module

MobileNet：Depthwise Convolution（参数计算，其为极端情况下的1\*1Group Conv）

MobileNet-V2：Inverted Residual, Linear Bottlenecks

Inverted Residual：在每层中间增加channel数来增加参数数量，然后在下一层前再缩回去。

Linear BottleNecks：with no ReLU at the end of a bottleneck.

ShuffleNet：Group Conv, Channel Shuffle

ShuffleNet-V2：

1. MAC（memory assces cout）：希望输入和输出的channel数尽量一致。
2. 过多的GConv会增加MAC的花销。
3. 网络碎片化会降低并行性，网络应尽可能使一条，分支越多效果越不好。
4. ElementWise操作的花销是相当大的。

multi-scale(pyramid), hourglass, siamese

FLOPs：浮点运算次数。

输入Tensor：H\*W\*Cin

输出Tensor：H\*W\*Cout

Kernel size：k

# of parameters: k\*k\*Cin\*Cout + Cout

FLOPs: [k\*k\*Cin\*Cout + Cout]\*H\*W（只考虑乘法，若考虑加法，直接乘以2即可）

for FC：

# of parameters: Nin\*Nout + Nout

FLOPs = # of parameters

MobileNet：0.573 GFLOPs

ShuffleNet：0.136 GFLOPs