Learning by doing

王超 2016.7

参加此次天津大学机器与数据挖掘中心举办的暑期研讨班收获颇多。而且这次机会也来之不易，从拿到报名资格到顺利参会，过程曲曲折折，但是只要你想去，是什么都挡不住的。

7.16 刘昕专题

上午 Deep Learning基础

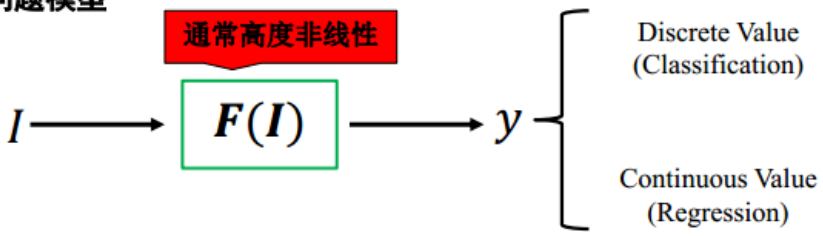
从以下四个方面展开：深度学习导引；深度学习的历史发展；深度学习基础；推荐阅读。分享非常精彩、细致、接地气。

讲者的精彩视频和PPT后面会统一分享给大家。本报告就几点自己遇到的问题和印象深刻的东西做一些总结。

1. 计算机视觉的方法模型

虽然关注CV算有半年时间了，但是对于计算机视觉的问题模型没有坐下来考虑过，之前更多的是关注具体的问题：如分类，识别，检测等任务；Feifei Li的说法：让计算机看懂世界。本次刘昕分享了一个最基本的问题模型：

- **问题模型**



2. 计算机视觉的研究方法

两段式研究方法：首先要做图像的特征表示，提取图像局部特征，纹如理、边缘等，然后解决视觉问题，如分类，识别等。

Prof. Songchun Zhu的一幅形象的漫画: 柜台上是： We treat all illness（我们可以解决计算机视觉的所有任务）；背后是一堆中药药材（各种图像提特征的方法）；开药方的老中医就根据不同的illness（问题）选择不同的特征（药材）；最后放到药壶里用Boosting熬制（利用不同特征训练自己的分类器，组合多个基分类器提高准确率）

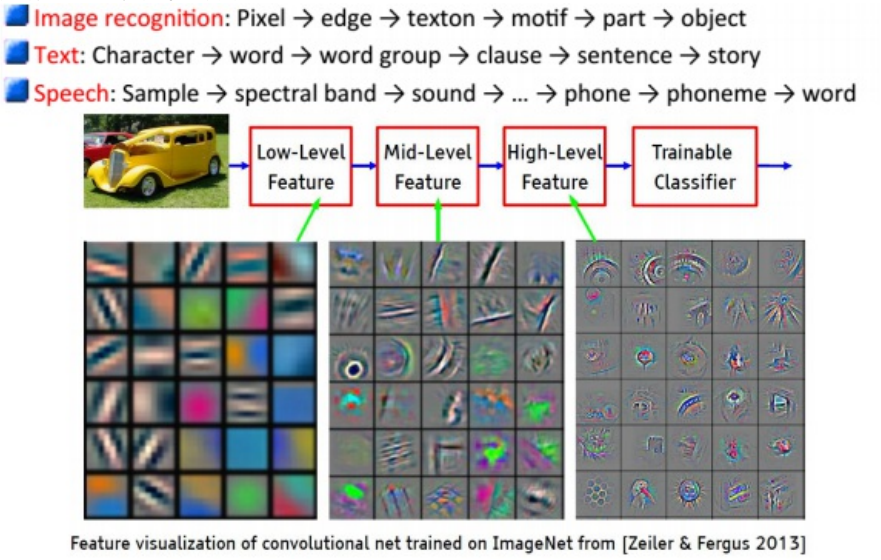
DL的崛起给CV界带来了巨大冲击；从百家争鸣到deep learning一统江山。DI带来了这些变革：

显示的学习非线性映射F(I)；

分层非线性->逐层的语义抽象（CNN类比大脑视觉神经系统的从低层到高层的不同分区：人类视觉系统的研究对于DL的启发很关键，交叉->更好地创新）；

End to End学习，与传统两段式最大的不同，不需要中间特征提取的过程，直接输入图像得到结果；
协同增效的优势： 我的理解是所有的阶段使用同样的数据，这样共同利用一个资源可以使产生一个整体的效应。

3. 深度学习概念深入
该部分讲了感受野和层级特征的概念：

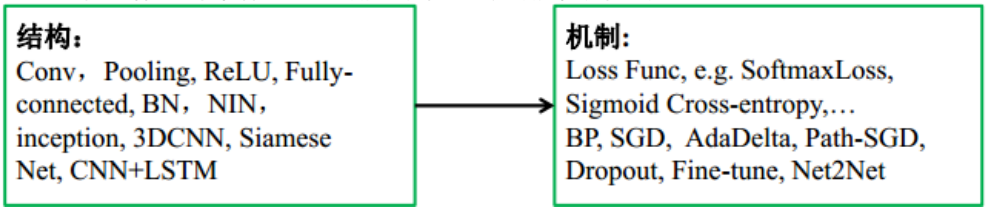


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

图像、文本、语音的不同层级、层层进行抽象，从low level的图像特征到high level的语义特征。此处有一个问题，CNN不同卷积层提取的特征结果和用传统方法提取的特征结果是否相似。

感受野：一个感觉神经元的感受野是指这个位置里适当的刺激能够引起该神经元反应的区域。可以这样理解：一个image patch对该神经元有最大的activation，那么这个patch就是这个神经元的感受野。

4. 看待DL的统一视角
刘昕博士给了一个不错的观点：结构+机制。DL是结构和机制的统一。

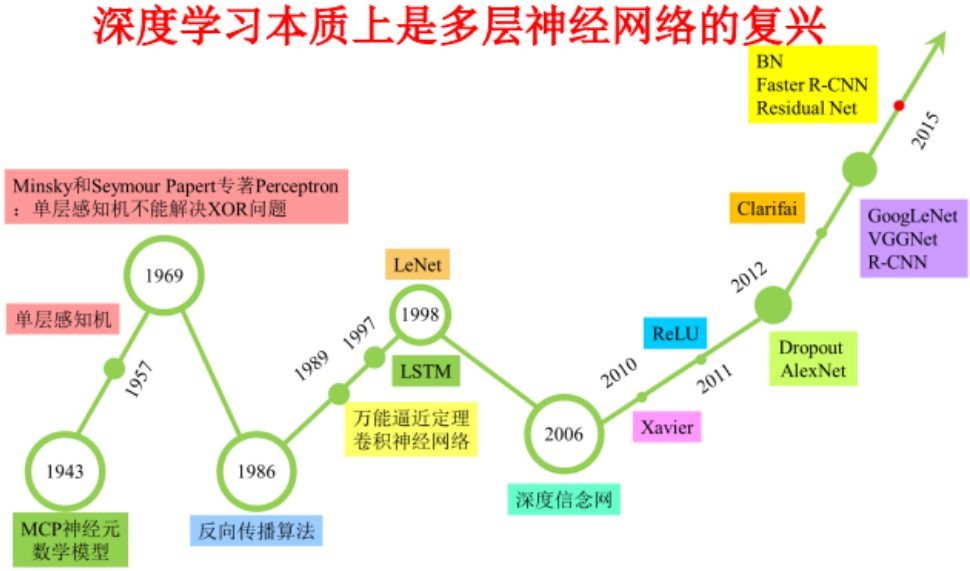


网络的基本材料在上个世纪已经基本建立了，现在更多的研究是在优化问题上，如SGD、momentum、Dropout等。

$$y_{\text{train}} = \begin{cases} \frac{x}{1-p} & \text{if } u > p \\ 0 & \text{otherwise} \end{cases} \qquad \text{Where, } u \sim U(0,1)$$

即使采用了Dropout，总得期望是不变的E（期望）= E（x）

5. 深度学习的历史发展

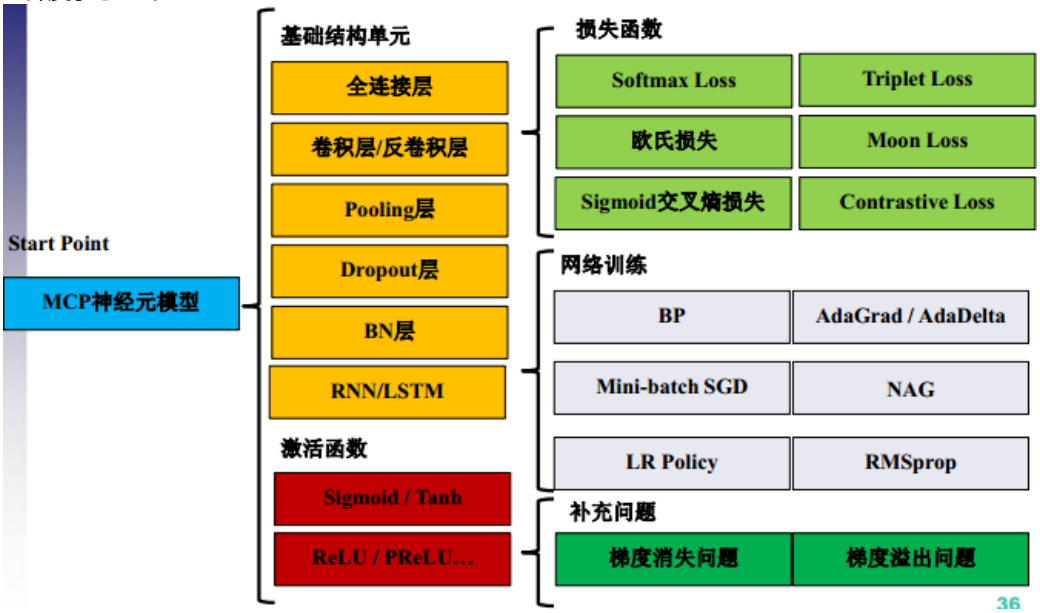


深度学习在前辈的坚持与努力下，经历了岁月的峥嵘到今天蓬勃发展，新事物的诞生总是曲折的螺旋上升的。有这样几点：20年磨一剑，1986年RBM提出，2002年Hinton提出CD算法，再到DBN提出DL概念。在这个发展历程中Hinton自己也反思了这20年间的无作为：

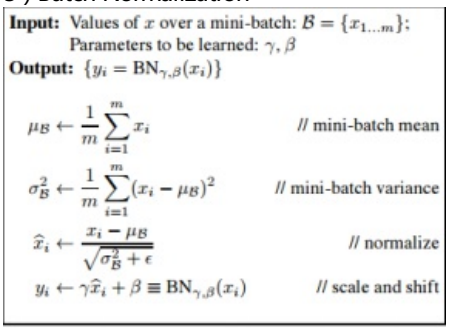
Our labeled datasets were thousands of times too small；
Our computers were millions of times too slow；

We initializaed the weights in a stupid way；
We used the wrong type of non-linearty。
无论岁月峥嵘，能够坚持到最后的人是幸运的。
2011年，在Hinton的介绍下微软研究者首先用DNN在语音识别领域取得了突破。2012年，AlexNet（5个卷积层，3个全连接层，ReLU，Dropout，LRN，GPU），ImageNet。

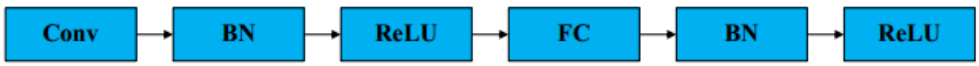
6. 深度学习基础



1）多通道卷积的快速实现，Im2Col（caffe）将卷积操作转换为矩阵运算，提升了运算速度。
2）反卷积操作，实现信号的复原；FCN利用反卷积操作实现上采样得到分割结果。
3）Batch Normalization



利用逐层尺度的归一化，避免梯度的消失和溢出；加速网络的收敛速度5到20倍，正则化技术提高泛化能力。如何使用：



4）LSTM: 引入Gate结构：Forget Gate, Input Gate, Output Gate；打破了RNN的局限：无法建立长时间的依赖。更好的理解LSTM，具体参考该博客(干货): http://colah.github.io/posts/2015-08-Understanding-LSTMs/
5）Loss Function：介绍了多种LF针对不同任务做优化。

- **Cross Entropy Loss分析**

对于离散的概率分布p和q, 交叉熵计算如下：

$$H(p,q) = -\sum_x p(x)\log q(x)$$

其中p(x)是one-hot encoding, q(x)是Softmax层输出

此处有一个数值稳定性问题的讨论：为什么softmax和Cross Entropy loss放在一起，因为在计算梯度时，log（x）的导数是1/x, 当x很小时，容易浮点溢出。所以数学上对的，数值上不一定OK。这也是自己在写layer的时候需要注意的问题---数值稳定性。
softmax loss 和 cross loss的区别：softmax输出维之间是相关的（是这一类那就不是其他类）；cross多标签分类（维之间是独立的：既可以是，这个也可以是那个）
关于怎样读源码：结合问题研究caffe源码，这样既高效（能解决问题）又有成就感。
6) BP, GD, mini-batch SGD with momentum, NAG.
7) 关于鞍点的讨论：最新的一个研究（search）：陷入局部最优的概率P很小很小，所以很多情况下找不到最优解是因为这个高度非凸的深度网络存在鞍点，实际上：高度非凸网络优化的难点不是陷入局部最优点，而是无处不在的鞍点。

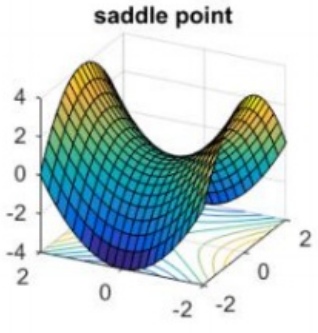
- 鞍点不等价于局部最值点

二元函数$f(x, y)$鞍点的数学条件(一阶导数为0，Hessian矩阵indefinite

$$\frac{\partial f}{\partial x}=0,\quad \frac{\partial f}{\partial y}=0,\quad \text{and}\quad \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2}-\left[\frac{\partial^2 f}{\partial x\,\partial y}\right]^2<0$$

以 $y=x_1^2-x_2^2$ 为例，点（0,0）是鞍点，

一阶导数是零向量，但并不是极小值点。



8）Learning Rate 策略

- Fixed
  - Learning Rate固定不变

```
base_lr: 0.01
lr_policy: “fixed”
```

- Step
  - Learning Rate在每隔stepsize轮迭代后减少gamma倍

```
base_lr: 0.01
lr_policy: “step”
gamma: 0.1
stepsize: 100000
```

- Polynomial
  - Learning Rate依多项式曲线下降

$$LR(t)=\text{base\_lr}\times(1-\frac{t}{T})^{power}$$

```
base_lr: 0.01
lr_policy: “poly”
power: 0.5
```

- Inv
  - Learning Rate随迭代次数增加而下降

$$LR(t)=\text{base\_lr}\times(1+\text{gamma}*\text{iter})^{-power}$$

```
base_lr: 0.01
lr_policy: “inv”
gamma: 0.0001
power: 0.75
```

7.一些推荐阅读内容：参见PPT。

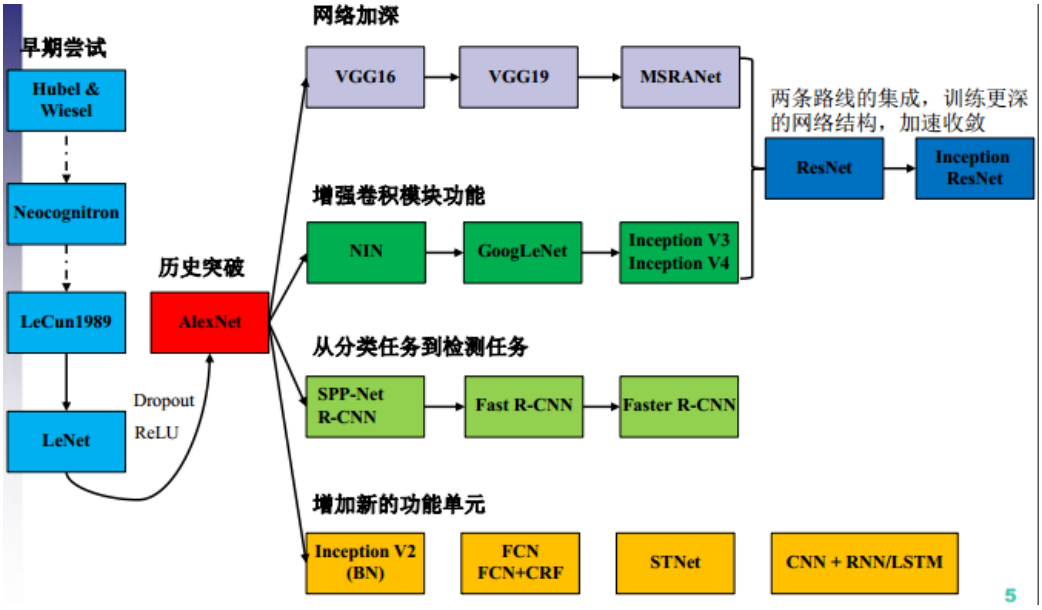
下午

CNN基础与Caffe实践

下午的主题主要从以下几个方面展开：CNN结构演化；Caffe源码与用法浅析；CNN实战技巧：Caffe；开放话题。

1. CNN演化脉络

一幅图概括了CNN的风雨历程和重要突破。



1）LeCun论文的彩蛋

All simulations were performed using the backpropagation simulator SN (Bottou and LeCun 1988) running on a SUN-4/260.

The nonlinear function used at each node was a scaled hyperbolic tangent. Symmetric functions of that kind are believed to yield faster convergence, although the learning can be extremely slow if some weights are too small (LeCun 1987). The target values for the output units were chosen within the quasilinear range of the sigmoid. This prevents the weights from growing indefinitely and prevents the output units from operating in the flat spot of the sigmoid. The output cost function was the mean squared error.

Before training, the weights were initialized with random values using a uniform distribution between $-2.4/F_i$ and $2.4/F_i$ where F_i is the number of inputs (fan-in) of the unit to which the connection belongs. This technique tends to keep the total inputs within the operating range of the sigmoid.

...rom empirical study (supported by theoretical arguments), the stochastic gradient was found to converge much faster than the true gradient,

Highlights:

1) 正切激活函数收敛更快！

2) Sigmoid归一＋欧氏损失！

3) 网络参数初始化！神秘的2.4是怎么回事？

4) SGD比GD收敛快！

a 为什么Tanh收敛快

$$\tanh'(x) = 1 - \tanh(x)^2 \in (0,1)$$

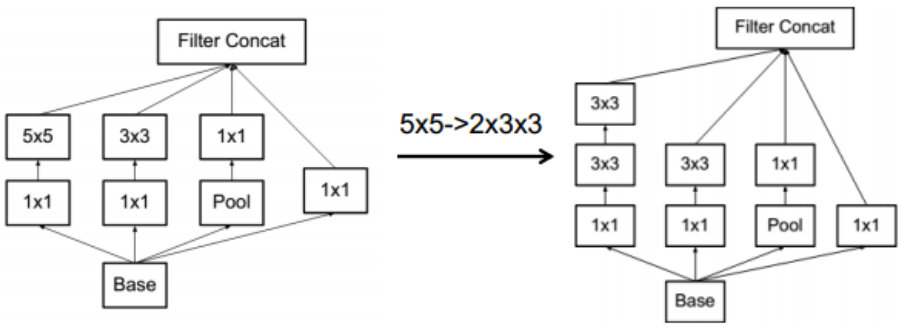
tanh(x)的梯度消失问题比sigmoid要轻，$s'(x) = s(x) \times (1 - s(x)) \in (0, \frac{1}{4})$

b 利用sigmoid函数归一化限制欧式损失的梯度范围，防止梯度的溢出

2) 网络

a GoogleNet的Inception结构利用两层3*3的卷积核代替5*5的卷积核，网络更深，参数更少；加辅助Loss减少梯度消失（BN可以搞定）Inception结构的理论可以认为是一个网络设计的trick：为什么这个结构就work，很多是通过实验试出来的。

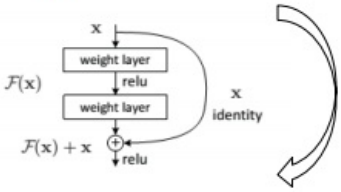
- 更深的Inception结构
- 5x5卷积核拆成两层3x3卷积核



b ResNet

shortcut; Identity

■ 2015-Deep Residual Net



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

c FCN 加 CRF：分割

DeepLab Liang-Chieh Chen's ICLR15

- **结构设计（FCN + Fully-Connected CRF）：**

1）双线性插值上采样粗的Score Map

2）基于全连接CRF优化（Refine）分割结果

2. Caffe

1）Caffe源码文件夹结构：

cmake：Caffe的编译配置文件；

data：ILSVRC等数据源（包含download脚本）；

examples：Caffe的例程；

include/caffe: 源码头文件；

matlab：Matlab接口；

models：参考模型Alexnet、GoogLeNet等：资源Model zoo；

python：Python接口；

src：实现文件（.cpp/.cu）；

tools: 数据转换，均值计算等工具。

2）Caffe核心概念：

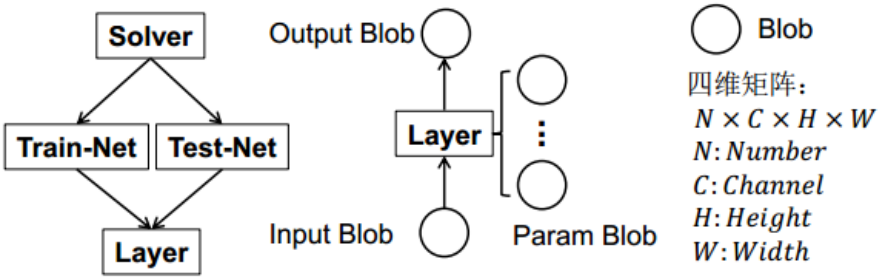
src/caffe/proto/caffe.proto公共数据结构定义

Solver: 求解器：进行网络参数的更新

Net：网络结构，运算中间结果

Layer：网络基本结构：Data、Conv、ReLU、Pooling，Loss

Blob：物理数据结构，本质是对SyncedMemory的封装



3）Caffe各个层的特点功能分析：data层；卷积层；激活函数层；全连接层；BN层等；特殊功能层：Split层、Slice层（Blob按维度切片）；concat层（多个Blob在指定维度链接：Inception结构中不同size卷积核的串联）；Eltwise层（逐元素计算；ResNet中实现short-cut连接的相加操作）

CNN实践技巧

1）Fine-tune： Fine-tune is Everywhere in CNN

2）模型集成：

ImageNet测试阶段的融合：五种Crop方式，镜像翻转

不同尺度的融合

不同Crop-size的融合
不同网络结构融合：VGG+GoogLeNet
不同损失函数的融合

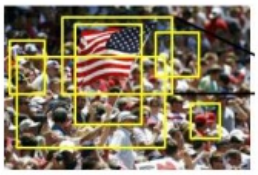
3）与经典方法结合
如：ICCV15 Marr Prize：Deep Neural Decision Forests：将全连接层实现为Decision Tree
用WPCA对CNN输出的特征进行降维

综合应用：

图像预处理

- 全局策略
  - 缩放图像以保持短边长为256像素
  - 按上中下或左中右采样三张256x256图像
- 局部策略
  - 基于Selective search每张图像采样若干局部区域
  - 滤除(1)长或宽小于原图20%(2)长宽比>2的区域
  - 缩放所有局部区域至256x256

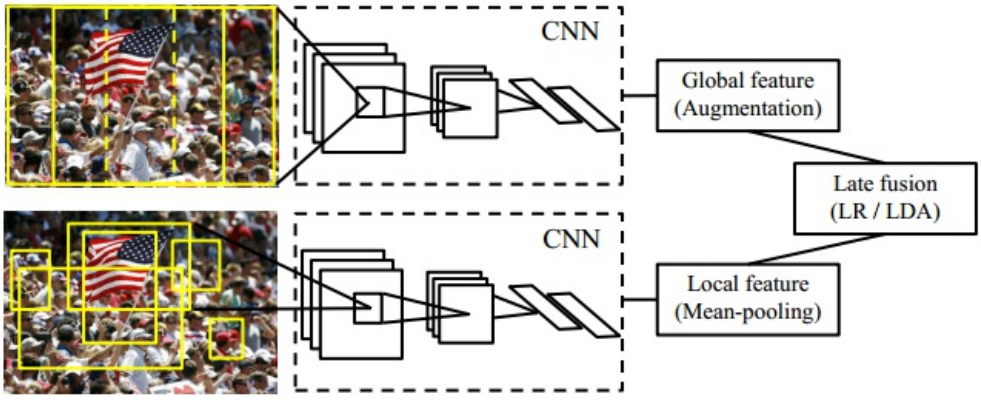
全局策略



局部策略

方法框架

- 模型融合：局部与整体，不同网络结构，不同后端方法
- Fine-tune
- 与经典算法的结合



4）Multi-task Learning：包含多个损失函数；多个数据源，使用连个DataLayer，设置对应batch_size，然后做concat，然后传递到下一层。

5）网络结构调参
如何得到比AlexNet更简单但性能更好的网络:

- Conv1层卷积核从11x11缩小到9x9，更小的卷积核可以提取更精细的特征
- 删除所有LRN层：LRN层的参数设置需要经过交叉验证，同时实践也表明，经过恰当的参数初始化并使用类似于Batch Normalization的归一化操作之后，LRN层对提升网络的泛化能力并无增益
- 删除Group操作。AlexNet采用Group操作是为了利用两块显卡进行并行训练，而目前已经有更好的方法,e.g.，Parameter Sever
- 5x5的卷积核拆分为两层3x3的卷积核，网络深度增加的同时没有增加计算量
- 第二个全连接层节点数从4096调整为2048
- 总的来说，把网络变深变瘦

开放话题：

深度学习打破了问题和方法的边界

- 物体识别与人脸识别边界的模糊化
  - 人脸识别和图像检索底层技术大一统
- 物体检测与物体识别边界的模糊化
  - Faster R-CNN多任务网络同时做物体检测与识别
- 端到端思维吃掉了越来越多传统方法的中间步骤
  - 砸掉马车夫饭碗的不是其他马车夫，而是蒸汽机（LBP和SIFT在人脸和图像领域各领风骚多年，被DL一齐打败）
- 只会Matlab不灵了?
  - 过去读CV的PHD/Master，Matlab几乎包治百病，现在得会：Linux, Shell/Vim, C++, Python, Matlab, CUDA…

刘昕的工作对自己的启示：

关于Caffe代码的阅读: 了解Caffe结构的基础上，先阅读核心部分的代码：

src/caffe/proto/caffe.proto公共数据结构定义

Solver: 求解器：进行网络参数的更新

Net：网络结构，运算中间结果

Layer：网络基本结构：Data、Conv、ReLU、Pooling，Loss

Blob：物理数据结构，本质是对SyncedMemory的封装

尤其对所有的Layer层进行深入研究

然后针对自己遇到的问题阅读相关的代码，在源码中寻找答案，这样解决问题更高效，也更有趣。

Caffe框架的修改：可以参考其他版本框架的实现：

Caffe on Spark（Yahoo开源，多机多卡）

https://github.com/yahoo/CaffeOnSpark

LRCN(LSTM for Video Representation and Image Caption)

https://github.com/jeffdonahue/caffe/tree/recurrent

Faster R-CNN (Object Detection)

https://github.com/rbgirshick/py-faster-rcnn

MXNet：集成了多种编程黑科技，如高度省显存版本，值得阅读和学习。

画网络结构的工具：http：//ethereon.github.io/netscope/#/editor

可视化分析工具：Jupyter

PPT里讲到的所有工作，刘昕博士都是亲自做过用过，认真研究过的，所以他能够这么清楚流畅的讲出来；所以最重要的是要动手实践；CV领域理论研究与代码实现能力同样重要Knowing what to solve & how to solve and solve it in a smart way。

感谢别人做了这样的工作，让自己少走了好多弯路，这就像开源了知识，这样的share越多，整个领域的进步才会更快。

接下来要做的几件事：

1. get到了一些有用的东西，投入到ILSVRC2016中，在这个过程中学习更好地使用Caffe，设计网络，掌握一些tricks的使用。

具体：从caffe核心开始；然后研究各种layer，尤其是特殊功能的layer：split、slice、concat、Eltwise的使用；注意实现过程的数值稳定性问题。

2. 根据两个PPt的思路与 相关资料，实践加整理（尤其该领域代表paper的整理和一些网络的实现改进，使用好pycaffe），用这个暑

期的时间去接近他讲述PPT的状态。

3. 对自己的要求：理论和工程实践要同时work（因为这句话真的不是笑话：Theory is when you know everything but nothing works. Practice is when everything works but no one knows why. and此处省略15词）

4. 更好的利用网络资源搜索引擎和一些好的网站，博客，公开课、开源项目等；很多之前的问题都能通过互联网search到。