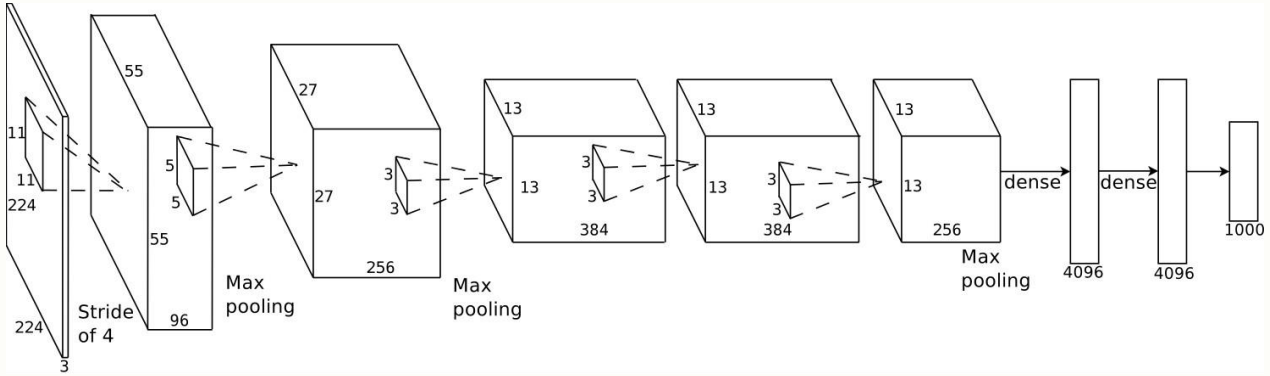
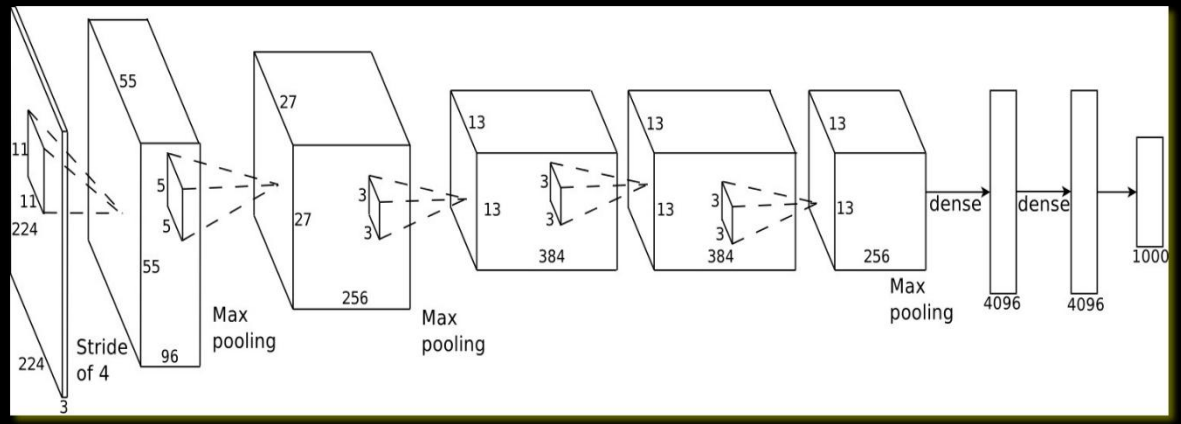


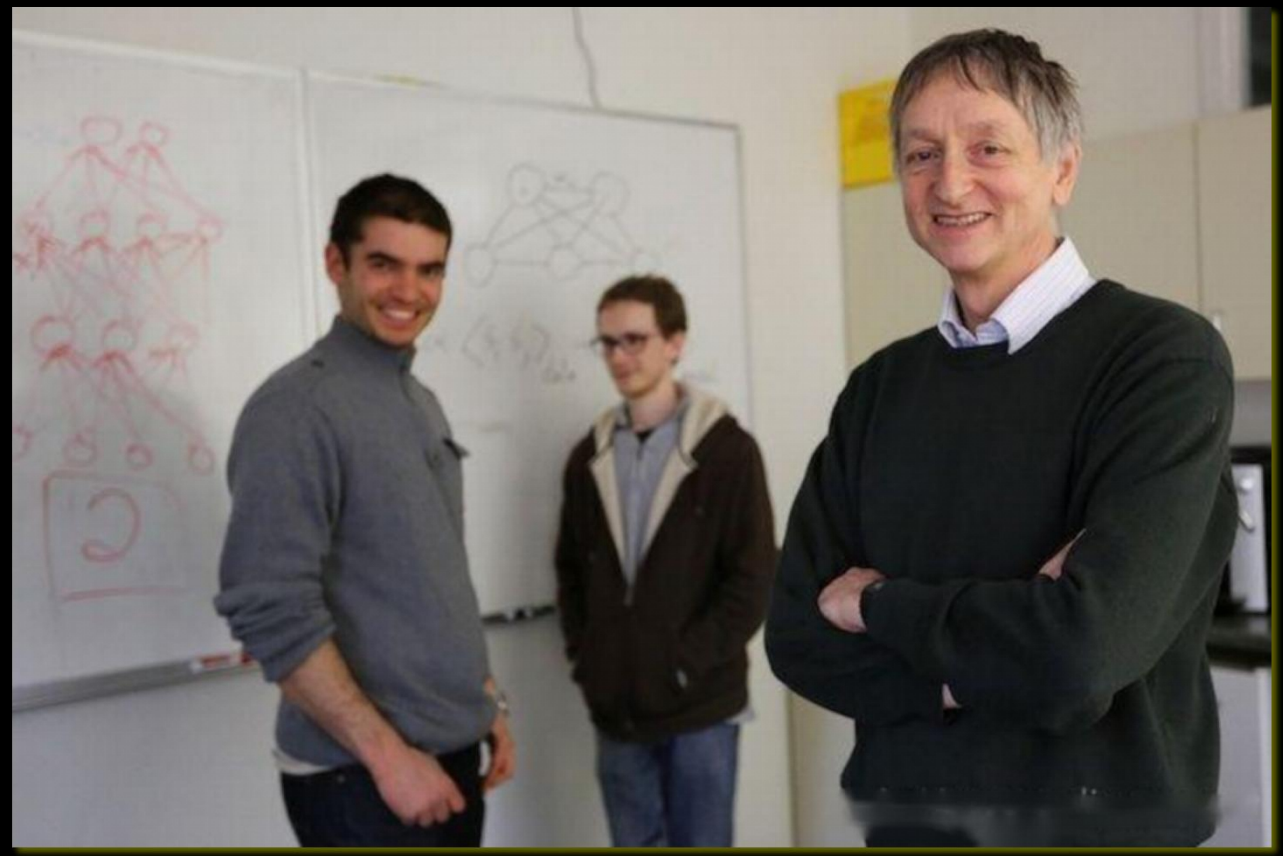
第05讲 深度神经网络

欧 新 宇

Thanks to LiMu-CS329 and LiFeifei-CS231n

划时代的深度神经网络AlexNet



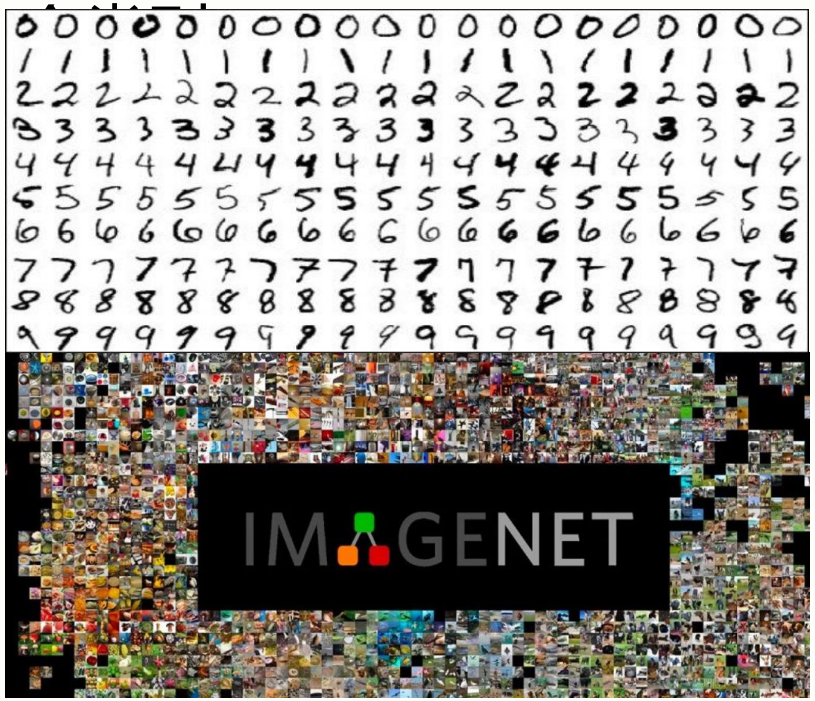


AlexNet和ImageNet简介

AlexNet和ImageNet简介

Imagenet 数据集

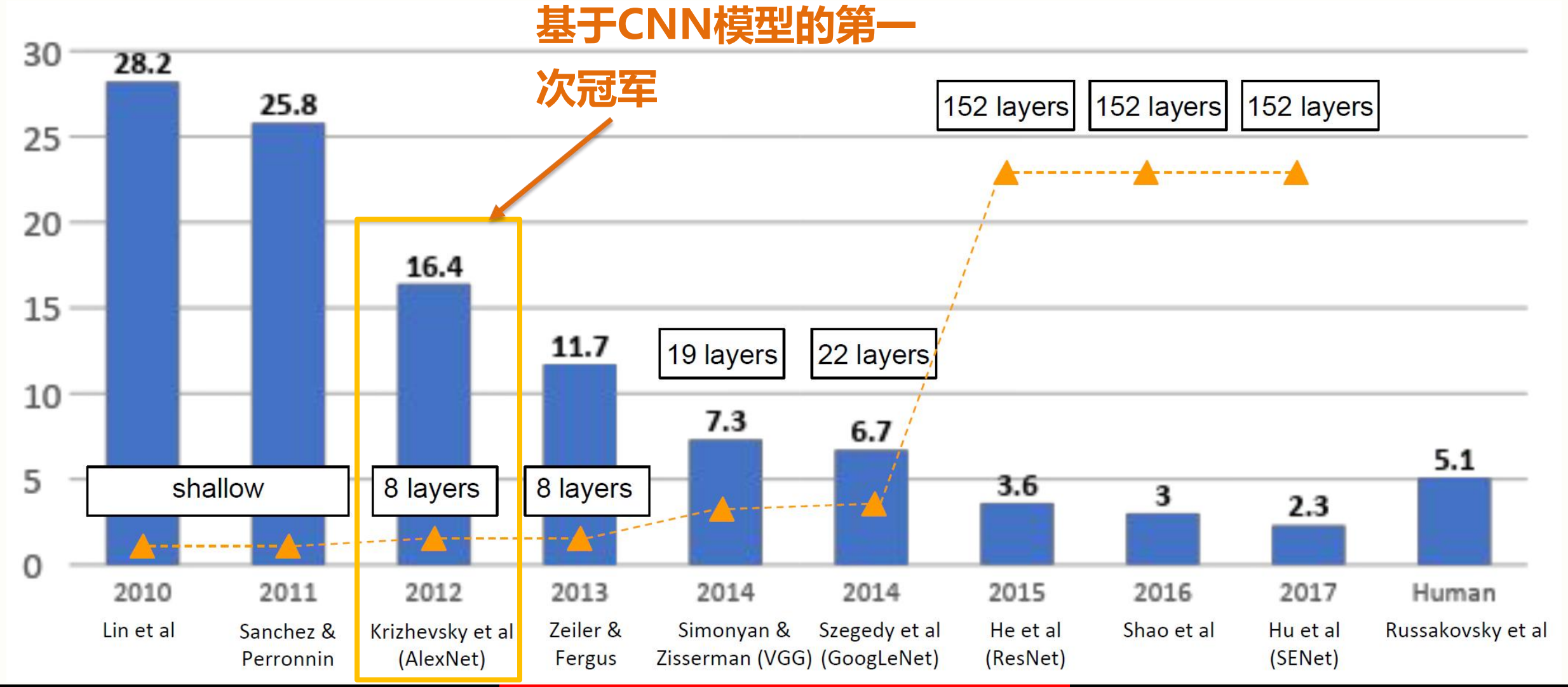
ImageNet图像数据集始于2009年，当时李飞飞教授等在CVPR2009上发表了一篇名为《ImageNet: A Large-Scale Hierarchical Image Database》的论文。完整版的ImageNet由是50,000+众包收集的数据集，该数据集包含1400万图像，分为22000



数据集	MNIST	ImageNet (ILSVRC)
分辨率	28×28	75×56 ~ 4288×2848 (469×387)
train	60,000	1,281,167 (2012年)
val	-	50,000
test	10,000	150,000
total	70,000	1,481,167
类别数	MNIST	1,000

AlexNet和ImageNet简介

ImageNet 大规模视觉识别挑战赛(ILSVRC) 冠军



AlexNet和ImageNet简介

改变历史的AlexNet

2012年，Alex Krizhevsky、Ilya Sutskever在多伦多大学Geoff Hinton的实验室设计出了一个深层的卷积神经网络AlexNet，夺得了2012年ImageNet LSVRC的冠军，且准确率远超第二名（top5错误率为15.3%，第二名为26.2%），引起了举世瞩目的轰动。

AlexNet是具有历史意义的一个网络结构，在此之前，神经网络已经沉寂了很长时间，自2012年AlexNet诞生之后，ImageNet的冠军都被**卷积神经网络（CNN）**包办了，并且层次越来越深，使得CNN逐渐成为图像识别分类的核心算法模型，也带来了深度学习的大爆发。

AlexNet的特点主要包括：

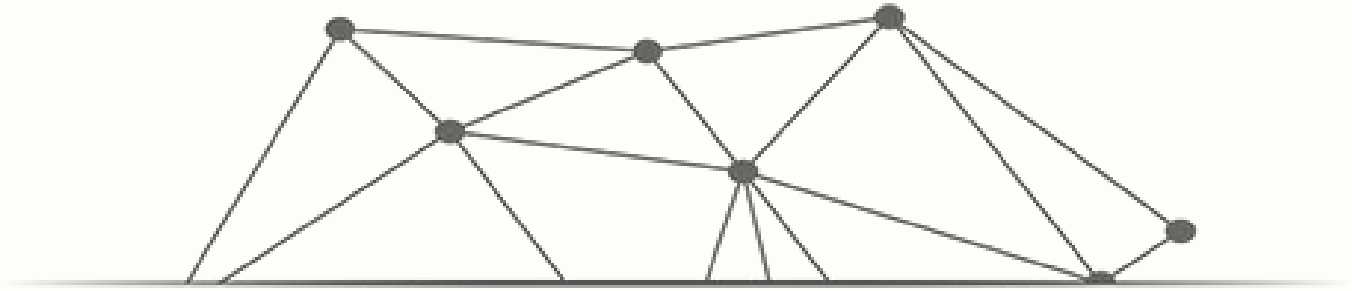
- ✓ **使用了非线性激活函数**：ReLUs
- ✓ **防止过拟合的方法**：Dropout、数据增广（Data augmentation）
- ✓ **其他**：有重叠池化、多GPU实现、LRN归一化层的使用

AlexNet和ImageNet简介

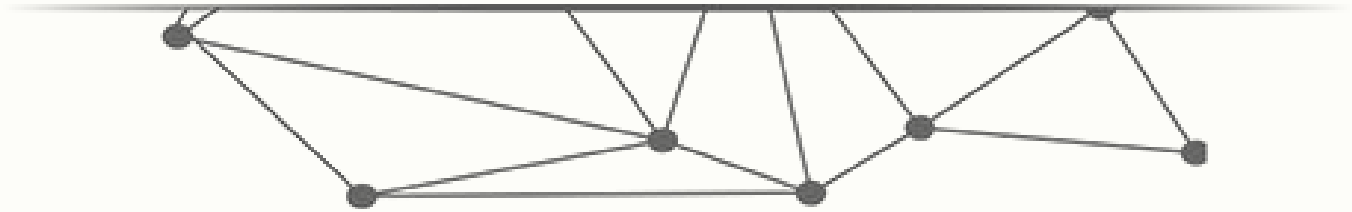
ILSVRC - Imagenet 大规模视觉识别挑战赛

ILSVRC挑战赛(2010-2017)总共举办了8届，2017年后，ImageNet由Kaggle继续维护。图像分类（Classification）和目标检测（Detection）是该竞赛的主要任务，在后期增加了使用VID数据集的视频目标检测, 使用MIT Places2数据集的场景分类（Scene Classification）和使用MIT ADE20K数据集的场景分割（Scene Parsing）。

年	模型/队伍名 (more)	Test top-5 (%)	备注
2012	AlexNet	15.32	7CNNs
2013	ZFNet, Overfeat, Clarifai	13.51, 14.18, 11.20	Overfeat (7 fast models)
2014	VGG, GoogLeNet	7.32, 6.67	7nets (dense eval, 144crop)
2015	**ResNet**	3.57	6 models，首次超越人的识别能力
2016	**Trimps-Soushen, ResNeXt**	2.99, 3.03	公安三所,加州大学圣地亚哥分校
2017	**SENet**	2.25	Momenta 与牛津大学



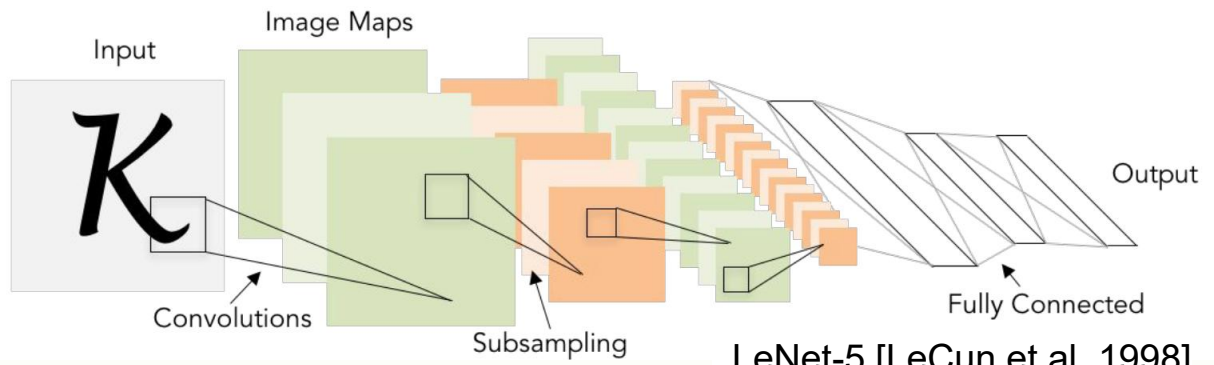
课堂互动 7.1.1



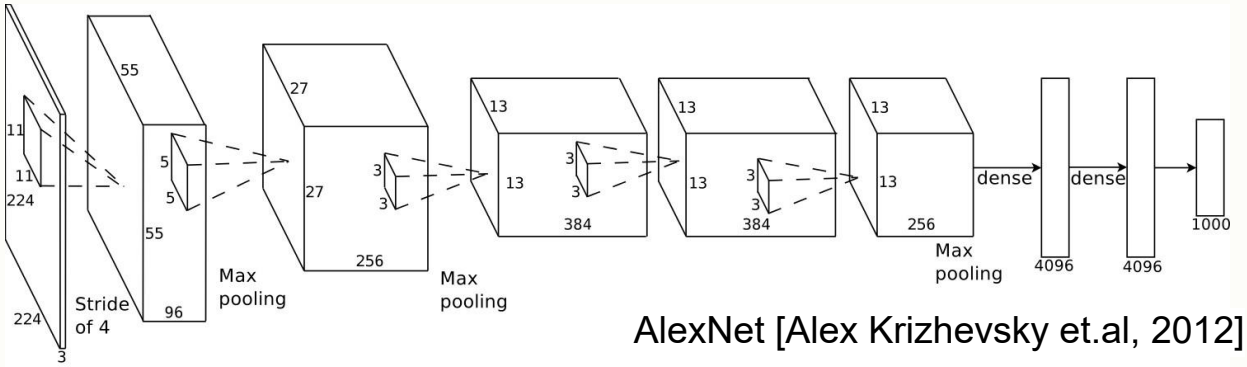
AlexNet的网络体系结构

AlexNet的网络体系结构

从LeNet-5到AlexNet



LeNet-5 [LeCun et.al, 1998]

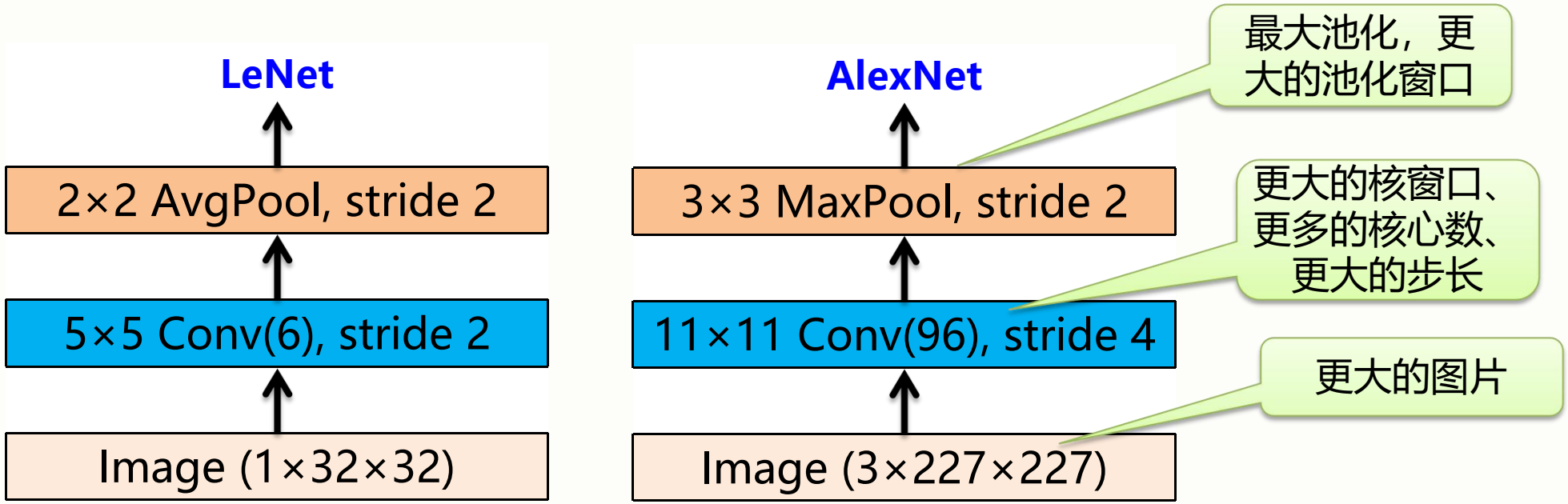


AlexNet [Alex Krizhevsky et.al, 2012]

- 获得了2012ImageNet竞赛冠军
- AlexNet是更深、更大的LeNet
- 主要改进：
  - 丢弃法Dropout
  - ReLU
  - MaxPooling
  - 数据增广
- 计算机视觉的方法论进化
  - 样本 -> 手工特征 -> SVM
  - 样本 -> CNN特征 -> Softmax回归

AlexNet的网络体系结构

从LeNet-5到AlexNet



Architecture:

FC8
FC7
FC6
Max POOL3
CONV5
CONV4
CONV3
NORM2
MAX POOL2
CONV2
NORM1
MAX POOL1
CONV1
Input

Q1: 第一个卷积层的输出尺寸是多少?

A1: (227-11)/4+1=55 => 输出尺寸:

[96×55×55]

Q2: 第一个卷积层的参数总共有多少?

Q3: 第一个池化层的输出尺寸是多少?

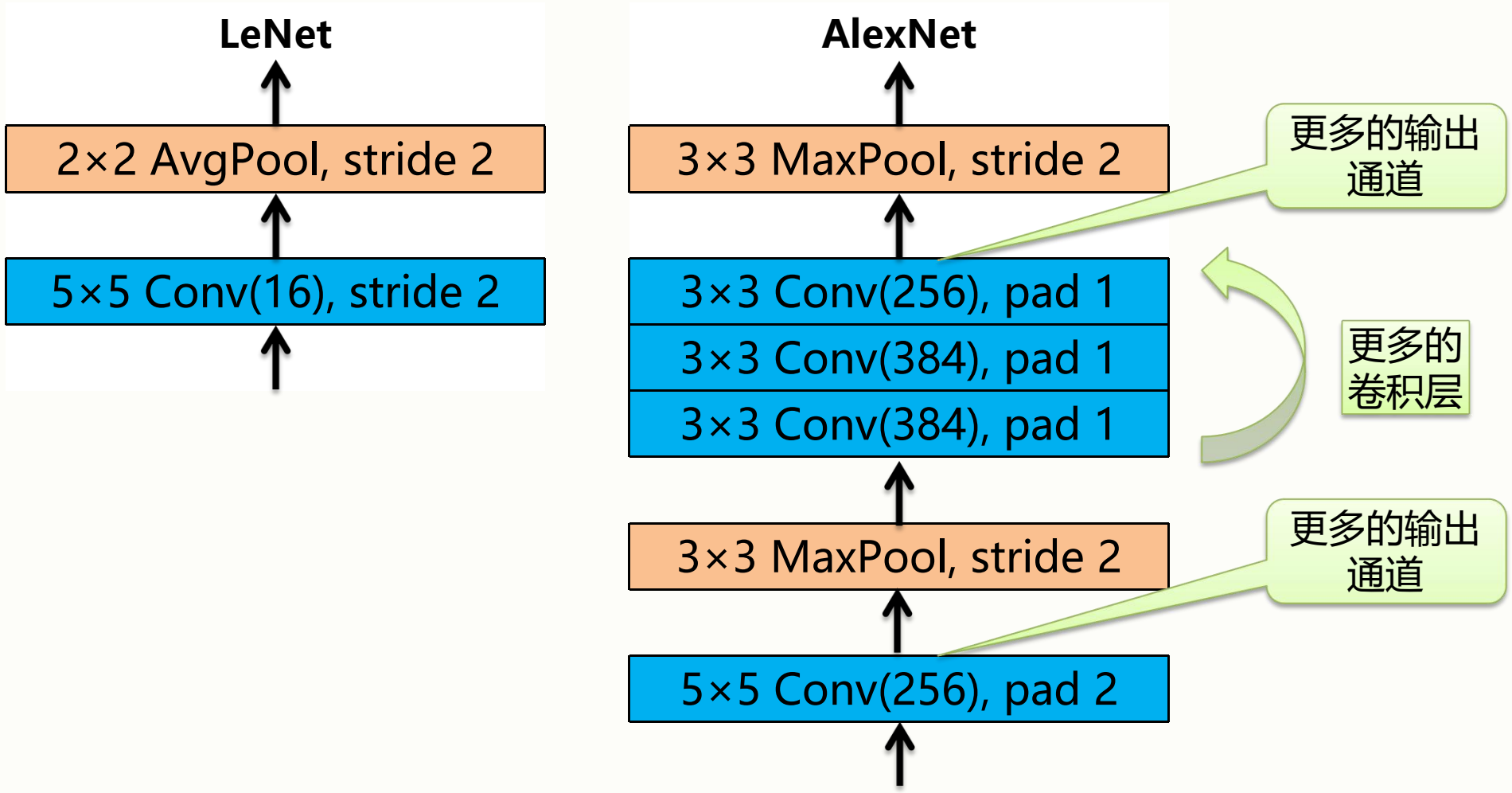
A3: (55-3)/2+1=27 => 输出尺寸:

[96×27×27]

Q4: 第一个池化层的参数总共有多少?

AlexNet的网络体系结构

从LeNet-5到AlexNet

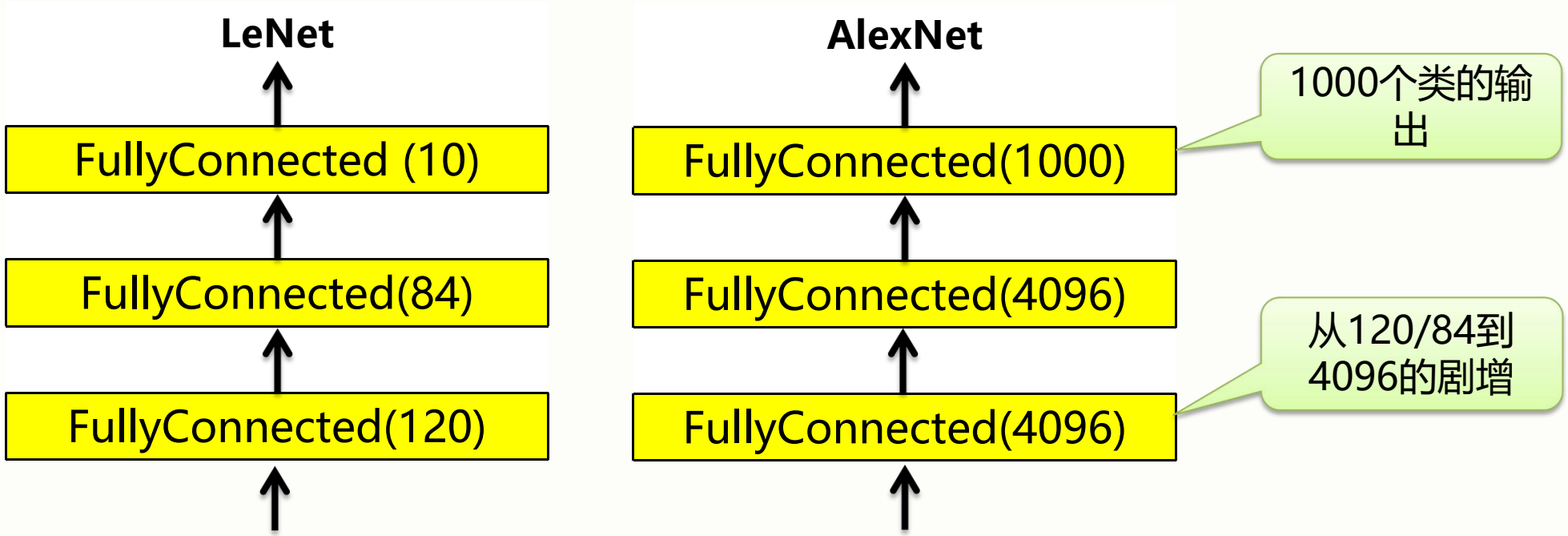


Architecture:

FC8
FC7
FC6
Max POOL3
CONV5
CONV4
CONV3
NORM2
MAX POOL2
CONV2
NORM1
MAX POOL1
CONV1
Input

AlexNet的网络体系结构

从LeNet-5到AlexNet



Architecture:

- FC8
- FC7
- FC6
- Max POOL3
- CONV5
- CONV4
- CONV3
- NORM2
- MAX POOL2
- CONV2
- NORM1
- MAX POOL1
- CONV1
- Input

AlexNet的网络体系结构

计算复杂度

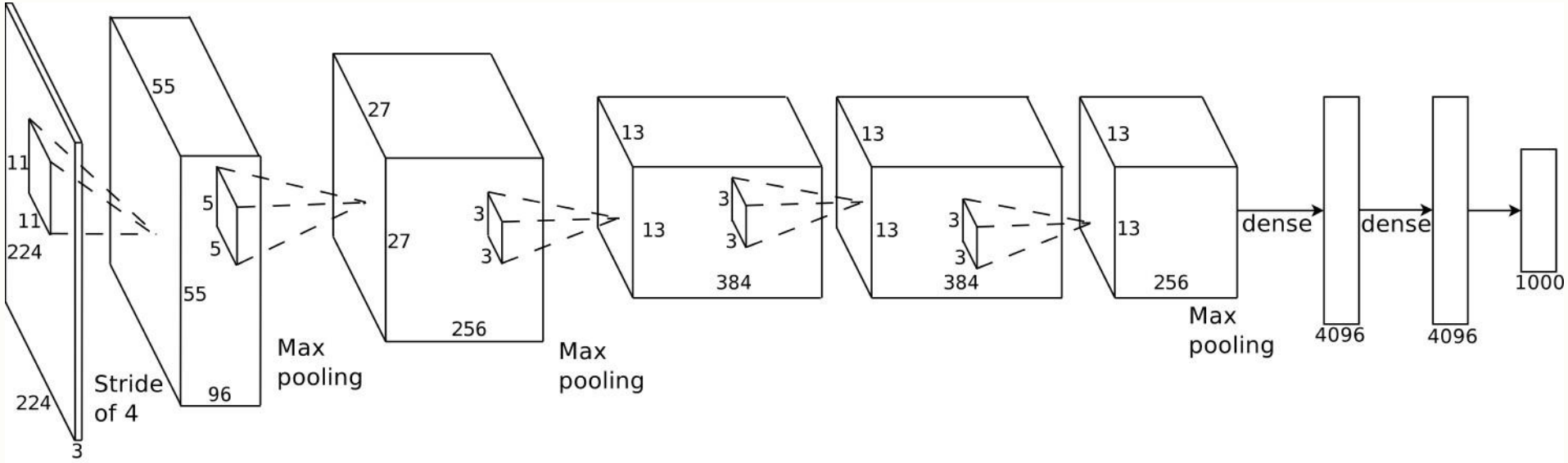
	参数数量		浮点计算次数FLOP	
	LeNet	AlexNet	LeNet	AlexNet
Conv1	156	34944	1.2M	101M
Conv2	2416	614656	2.4M	415M
Conv3-5		3,097,600		445M
FC1	48120	37,752,832	0.48M	26M
FC2	10164	16,781,312	0.1M	16M
总数	61,706	61,100,840	4M	1000M
倍数	1x	990x	1x	250x

Architecture:

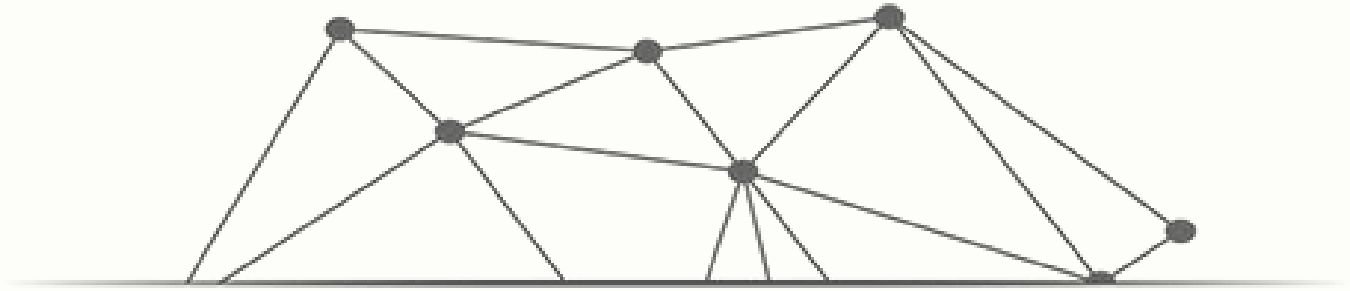
FC8
FC7
FC6
Max POOL3
CONV5
CONV4
CONV3
NORM2
MAX POOL2
CONV2
NORM1
MAX POOL1
CONV1
Input

AlexNet的网络体系结构

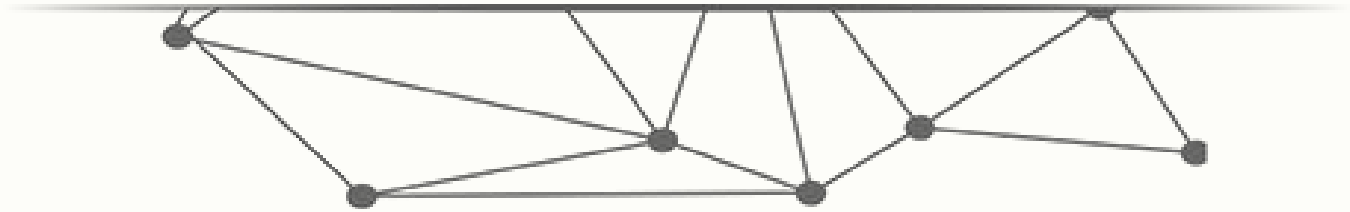
网络体系结构总览



- **模型深度**：7个包含权重的隐层（5个卷积层，2个全连接层）
- **特征学习**：所有特征以高斯白噪声为初始值，然后从数据中学习获得
- **学习类型**：完全的监督学习
- **神经元和参数**：~65万个，~60M个

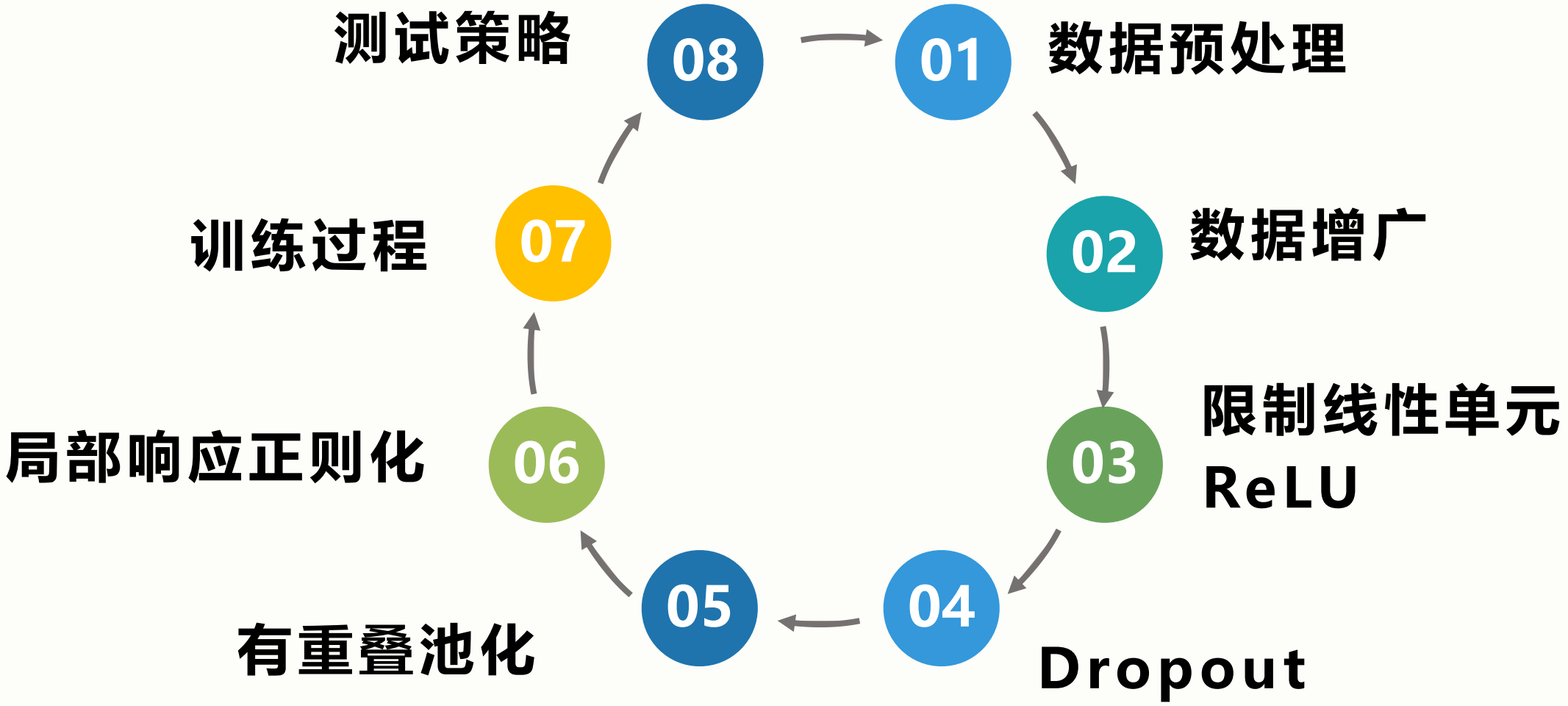


课堂互动 7.1.2



AlexNet的技术细节一

AlexNet的技术细节



数据预处理

数据预处理的基本步骤

- 数据清洗（Data Cleaning）
- 数据标注（Data Annnotations）
- 数据集成（Data Intergration）
- 数据列表生成（List Generation）
- 数据归约（Data Reduction）
- 数据增广（Data Augumentation）

像素归一化

将像素值从[255,0]归一化到[0,1]

均值消除

均值减除和方差归一化

维度规约

量纲统一

数量规约

消除代表性不强的特征

通道转换

HWC转换为CHW

数据类型变换

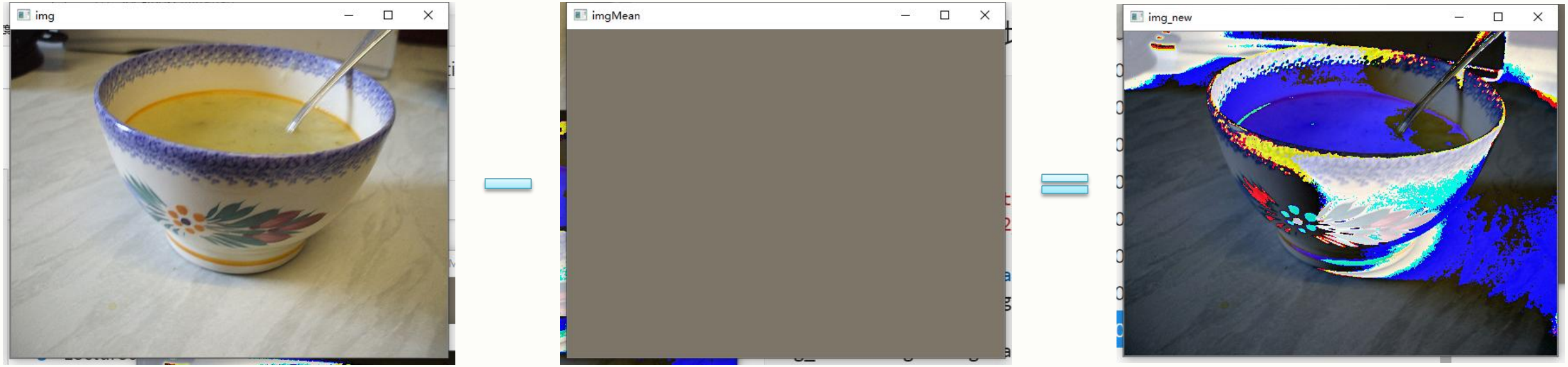
32位浮点型和8位整型等

数据预处理

针对RGB通道分别进行减均值操作

We did not pre-process the images in any other way, except for subtracting the mean activity over the training set from each pixel. So we trained our network on the (centered) raw RGB values of the pixels.

RGB值中心化：对所有图像，都按照色彩通道进行均值减除。均值从Imagenet数据集的训练集（120万图像）求取。该均值为[104,117,123]，适用于绝大多数基于自然场景的样本。



数据预处理

样本预处理之归一化

- **归一化（Normalization）：**

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}}$$

x是某个特征的原始值，x^*是经过归一化后的特征值，x_{max}为该特征在所有样本中的最大值，x_{min}为该特征在所有样本中的最小值。经过归一化后，所有的特征都会被映射到[0, 1]之间。归一化可以在进行多特征计算时，消除量纲对结果的影响，使不同特征具有可比性。

例如，给定一组特征序列

$x = [0.1,\ 2.3,\ -4.5,\ 4.2,\ -2.7,\ 6]$,

则$x^* = ?$

```
import numpy as np
np.set_printoptions(precision=3, suppress=True)
x = np.array([0.1, 2.3, -4.5, 4.2, -2.7, 6])
x_norm = (x-np.min(x))/(np.max(x)-np.min(x))
print(x_norm)
```

```
Last executed at 2020-10-26 22:12:57 in 13ms
```

```
[0.438 0.648 0.    0.829 0.171 1.   ]
```

数据预处理

样本预处理之标准化

- **标准化（Standardization）：**

$$x^* = \frac{x - \mu}{\sigma}$$

x是某个特征的原始值，x^*是经过标准化后的特征值，μ为该特征在所有样本中的平均值，x_{min}为该特征在所有样本中的标准差。经过标准后，所有的特征都会满足均值为0，标准差为1的正态分布。

例如，给定同一组特征序列

$x = [0.1,\ 2.3,\ -4.5,\ 4.2,\ -2.7,\ 6]$,

则$x^* = ?$

```
import numpy as np
np.set_printoptions(precision=3, suppress=True)
x = np.array([0.1, 2.3, -4.5, 4.2, -2.7, 6])
x_std = (x-np.mean(x))/np.std(x, ddof=0)
print(x_norm)
```

```
Last executed at 2020-10-26 22:12:50 in 16ms
```

```
[0.    0.373 0.746 0.695 0.441 1.   ]
```

数据预处理

均值的两种计算方法

标准化和归一化都具有消除量纲的功能，使得原本分布相差较大的特征调整为对模型具有相同权重的影响，在传统机器学习中使用非常普遍。如果期望在最后的计算中，让所有特征都具有相同作用，则使用**标准化**；若希望保留原始数据中标准差所反映的潜在权重关系，则使用**归一化**。此外，标准化更加适合具有较多噪声的大数据场景。

在基于大数据的深度学习中，我们一般有两种方法求数据的均值，一种是按像素进行零均值、另一种是按通道进行零均值。

- ✓ **按像素进行零均值：**针对每个像素的所有通道进行零均值，若样本分辨率为200×200，则最终会有40000个均值。该方法计算相对复杂，而且由于不同样本中的对象是随机分布的，按通道求取均值会导致数据发生一定的偏差。
- ✓ **按通道进行零均值：**按照每个通道进行零均值，若样本包含3个颜色通道，则最终会有3个均值。自然界的样本通常都满足正态分布，只要样本足够多，按通道进行零均值通常都能够获得较好的效果。

数据预处理

零均值的作用

简而言之，零均值有助于避免“Z型更新”的情况。它的好处是：

- 提高神经网络收敛速度，
- 避免像素间因为均值的相互影响导致特征显著性的下降。

值得注意的是，灰度图一般不需要执行均值消除。

除此之外，建议所有来源于自然图像样本都执行数据标准，并按照通道进行均值[104, 117, 123]减除。

数据增广（Data Augmentation）

为什么要使用数据增广

数据增广（Data Augmentation）

How to ...

一种观点认为：神经网络是数据喂出来的，如果能增加训练数据（甚至是提供海量数据），则能够有效避免过拟合并由提高算法准确率；同时，可以进一步增大，加深神经网络，从而进一步提高神经网络模型的拟合能力。

数据增广（DataAugmentation）的提出，被认为是当训练数据有限时，最有效的方法之一。

数据增广（Data Augmentation）

图像在互联网上传播时可能发生的变化

平移

色偏

遮挡

光照

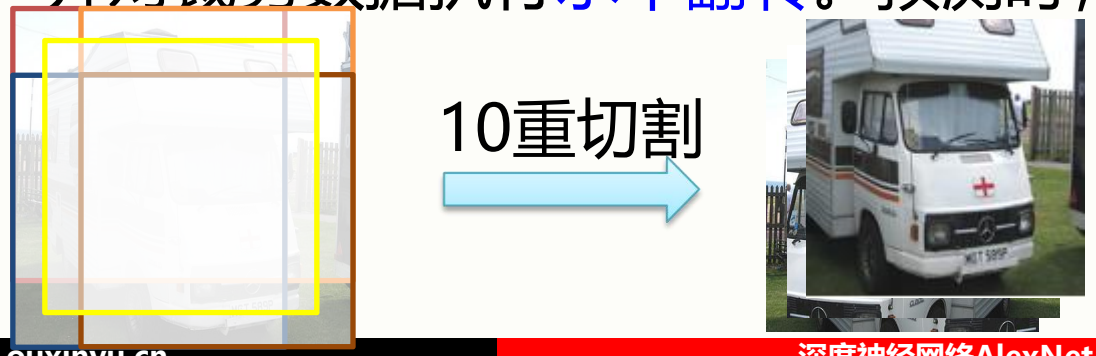
3D
旋转

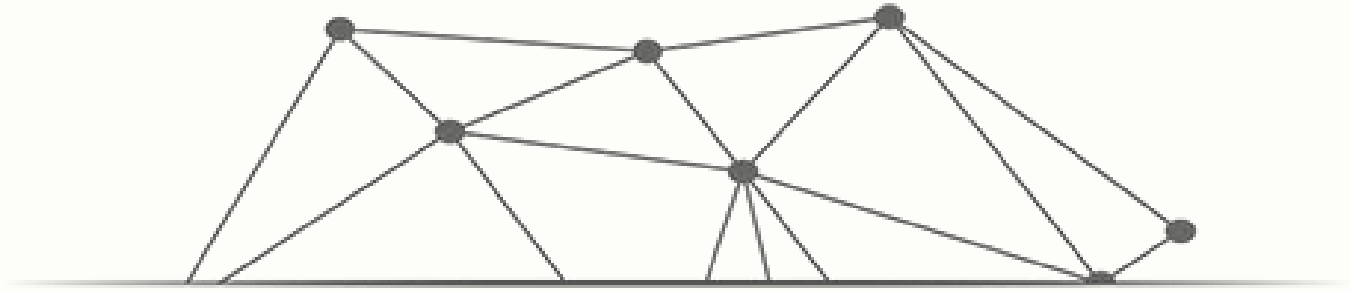
数据增广（Data Augmentation）

Alexnet做了什么?

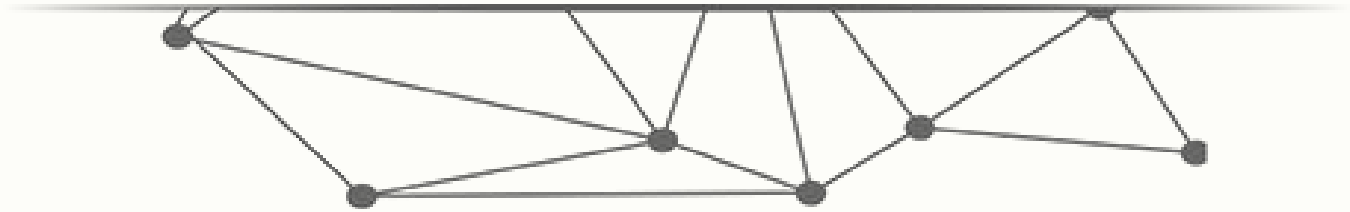
- **训练阶段**，执行顺序裁剪+水平翻转：对256×256的图片进行顺序裁剪至224×224，并对每个切片都执行水平翻转。数据增加量=$(256-224)^2 \times 2 = 2048$倍

- **测试阶段**，执行10重切割在256×256的样本中执行左上，左下，右上，右下，中间5次裁剪，并对裁剪数据执行水平翻转。预测时，对10个样本的预测概率求平均值。





课堂互动 7.1.3

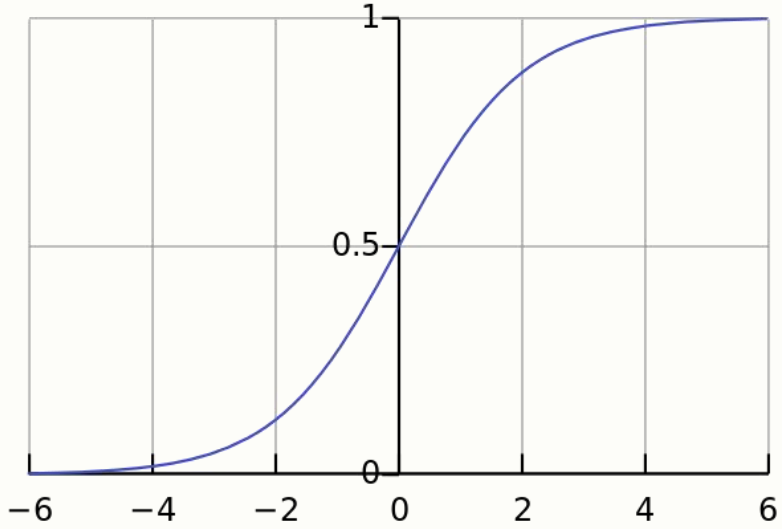


AlexNet的技术细节二

限制线性单元ReLU

Sigmoid的局限性

在早期的神经网络中，使用最频繁的激活函数是Sigmoid函数，因为其很好地解释了神经元受到刺激后是否被激活和向后传递的场景（0：没有被激活；1：完全被激活）。但是，在近几年的深度学习的应用中很少见到他的身影。主要原因是：容易产生梯度饱和。双曲正切函数tanh和Sigmoid函数具有相似的特性。



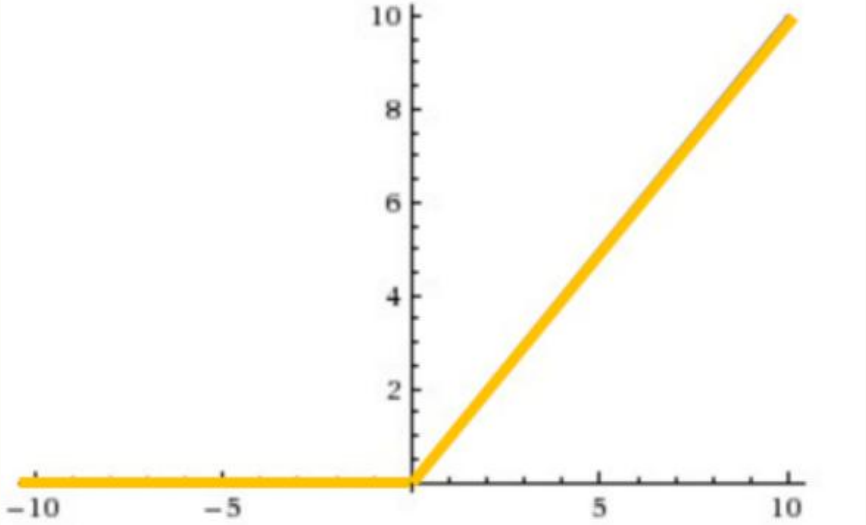
当输入值非常大或非常小的时，神经元的梯度将接近于0，从而产生梯度饱和。

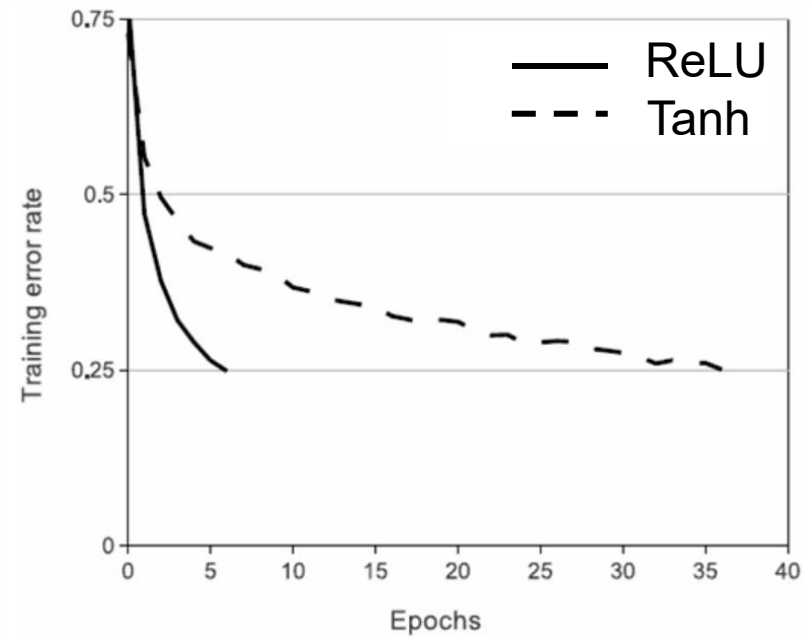
因此，Sigmoid和Tanh又被称为饱和非线性单元。

限制线性单元ReLU

ReLU的优势

ReLU（Rectified Linear Units）限制线性单元，其公式为F(x)=max(0,x)。 当输入信号<0时，输出都是0；当输出信号>0时，输出等于输入。





由于ReLU是线性的，因此其导数始终为1，在反向传播时其计算量大大减少，收敛速度比Sigmoid和Tanh快得多。上图为Imagenet原文中CIFAR数据集上的实验过程，ReLU相比Tanh快6倍。

Dropout

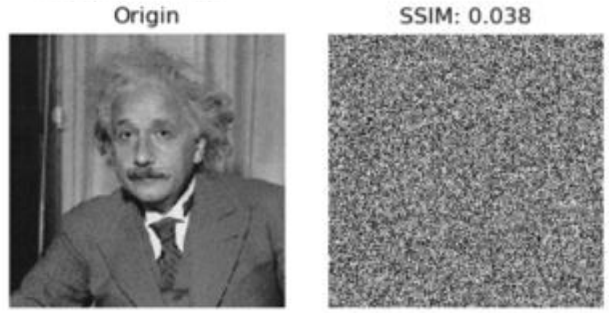
Dropout的引入

动机：好的模型需要对输入数据的扰动足够鲁棒

功能：防止过拟合

原理：

1. 对某全连接层，Dropout通过修改神经网络的结构实现按一定概率将神经元置为0，该神经元不参与前向和反向传播，等同于被删除，此时网络结构不变。
2. 在下一次迭代中，新的一组神经元被重新随机删除（置为0），直至**训练结束**。

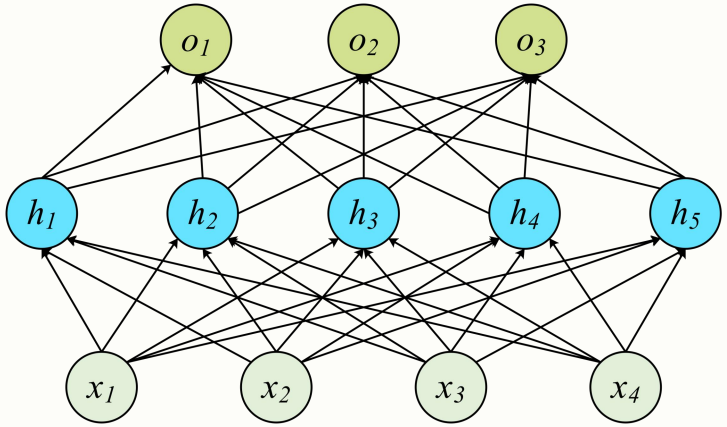


研表究明，汉字的序顺并不一定会影阅响读。比如当你完看这句话后，才发这现里的字全都乱是的。

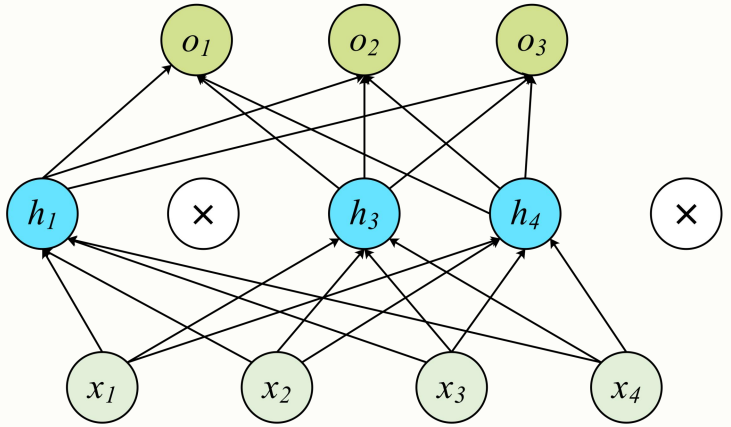
Dropout

Dropout的具体实现

普通的多层感知机



使用Dropout后的多层感知机



$$\boldsymbol{h} = \sigma(\boldsymbol{W}_1 x + \boldsymbol{b}_1)$$

$$\boldsymbol{h}' = \text{dropout}(\boldsymbol{h})$$

$$\boldsymbol{o} = \boldsymbol{W}_2 \boldsymbol{h}' + \boldsymbol{b}_2$$

$$\boldsymbol{y} = \text{softmax}(\boldsymbol{o})$$

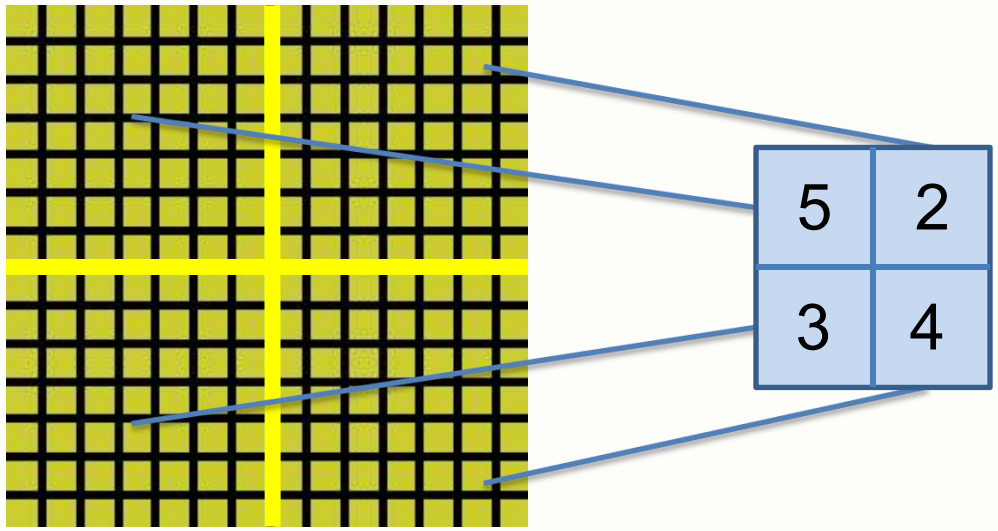
- Dropout是作用在全连接层上的正则项，通过随机丢弃来减少模型对训练样本的依赖，从而降低过拟合。
- Dropout可以被看成是一种生成不同结构模型的方法，通过迭代实现多模型组合。
- Dropout在推理过程中不生效，目的是保证输出的稳定性。
- 在AlexNet中，Dropout被应用在最后2个4096维的全连接层。

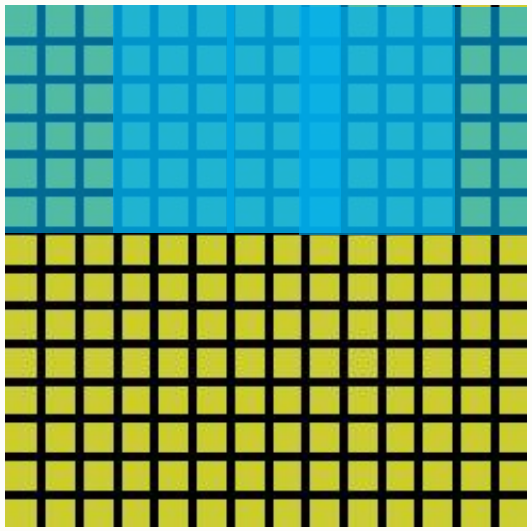
有重叠池化

从无重叠池化到有重叠池化

Pooling层通过汇总近邻组的输出实现了特征图的抽象。

- 在一般的池化中，pool_size = stride，这种池化称为无重叠池化。 例如8×8的特征图，经过步长为2，池化核为2×2的池化后，将变为4×4。
- 当stride<pool_size，则称为有重叠池化。这种方法让Alexnet在Imagenet任务中将错误率降低了0.4%（top-1）和0.3%（top-5）。主要原因是有重叠池化更加不容易过拟合(原文）。





AlexNet

✓ Stride= 2

✓ pool_size=3

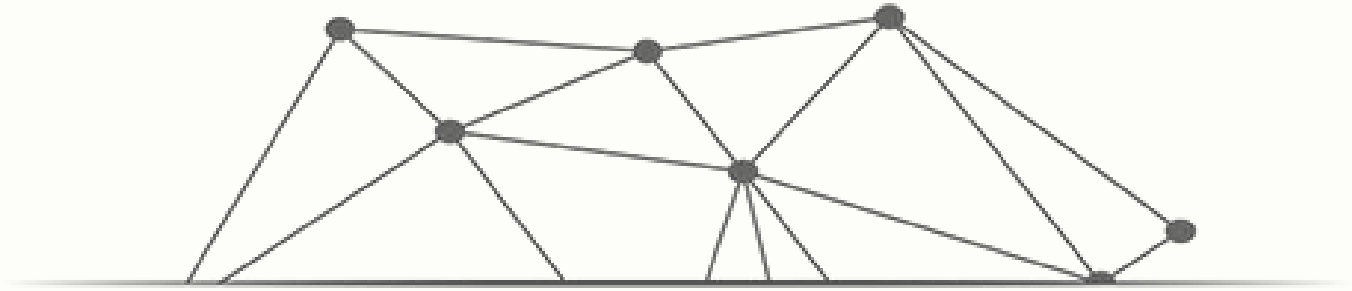
局部正则响应

Local Response Normalization(局部响应正则化)

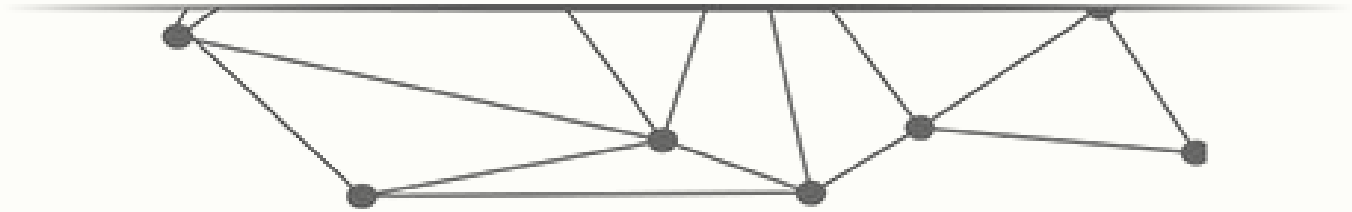
【理论上】Sigmod和Tanh有一定的正则化作用，它们将输出归一化到[0,1]/[-1,1]之间。而ReLU的值域并没有限制，因此需要对其进行正则化。Alexnet提出了一种正则化方法，称为**局部响应正则化 (Local Response Normalization，LRN)**。基本想法是通过LRN层，对局部神经元的活动创建竞争机制，使得其中响应比较大的值变得相对更大，并抑制其他反馈较小的神经元，增强模型的泛化能力。

$$b_{x,y}^{i} = a_{x,y}^{i} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^{j})^2 \right)^{\beta}$$

该方法实现了1.4%，1.2%的Top1和Top5提升。但后续的很多研究提出LRN效果并不是很明显。



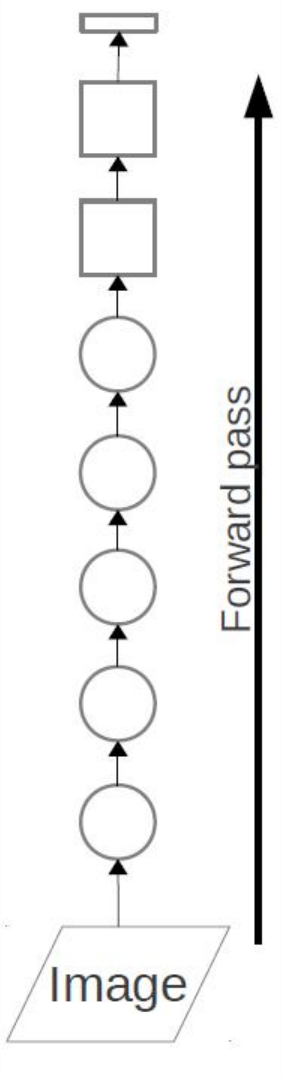
课堂互动 7.1.4



AlexNet的技术细节三

训练细节

训练及输出

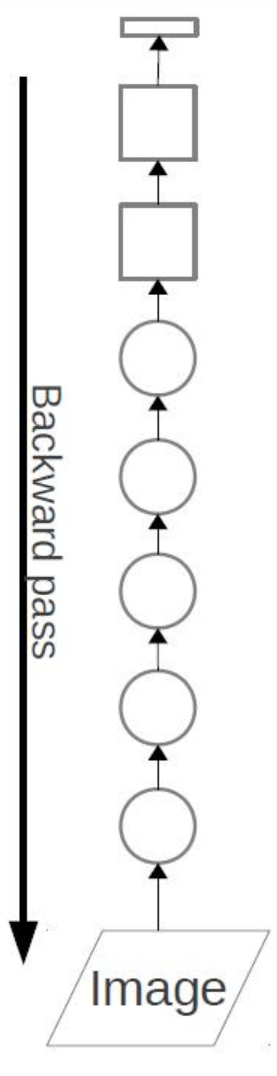


使用随机梯度下降算法和反向传播算法（链式法则）实现梯度更新。

每个类的输出 x_i = 总输入到输出单元 i 的输出 $f(x_i)$。

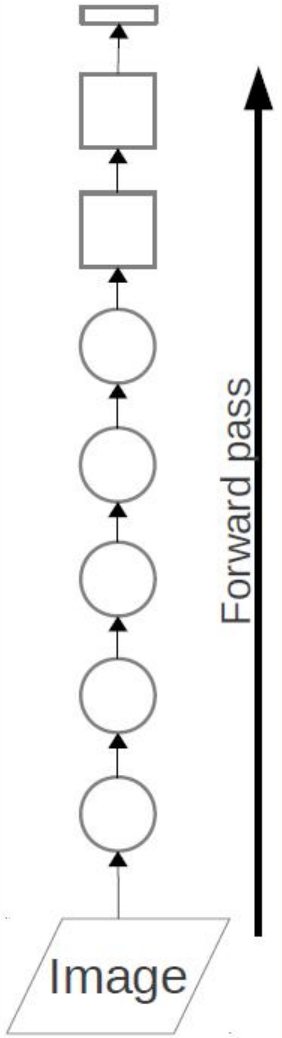
$$f(x_i) = \frac{\exp(x_i)}{\sum_{i=1}^{1000} \exp(x_i)}$$

预测标签为所有输出概率的对数 $\log(x_t)$ 中最大的值的索引。



训练细节

基于动量的随机梯度下降



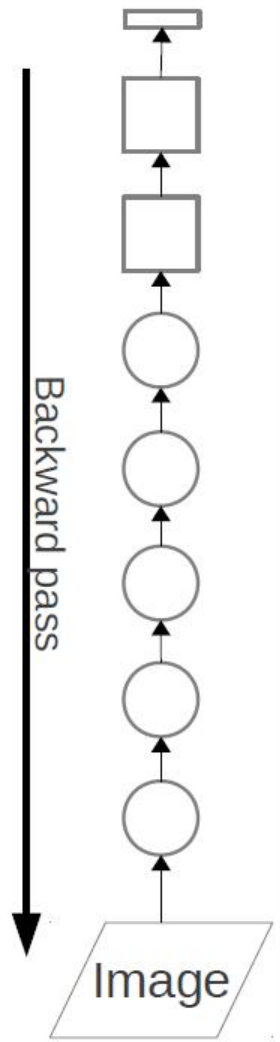
权重（梯度）更新规则：

动量项　　权重衰减项　　梯度项

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

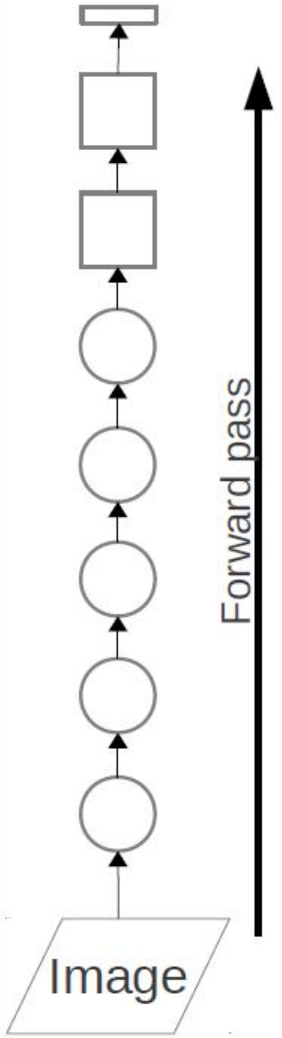
$$w_{i+1} := w_i + v_{i+1}$$

- ✓ Batchsize=128， 其中i为迭代索引
- ✓ 动量Momentum=0.9，权重衰减系数0.0005
- ✓ v为动量变量，ϵ为学习率
- ✓ $\left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$ 为第i个批次 D_i中针对权重w_i的梯度

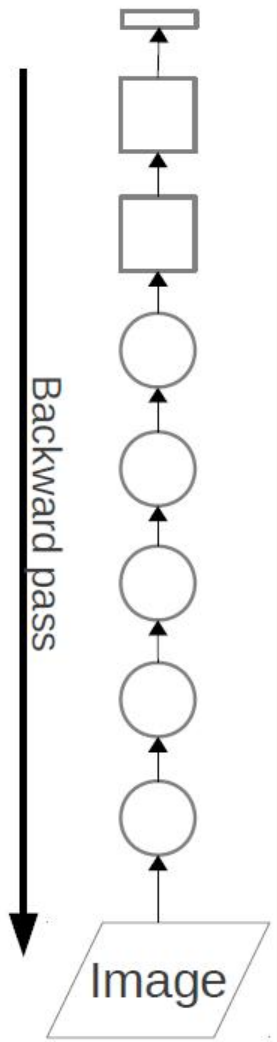


训练细节

其他细节设置

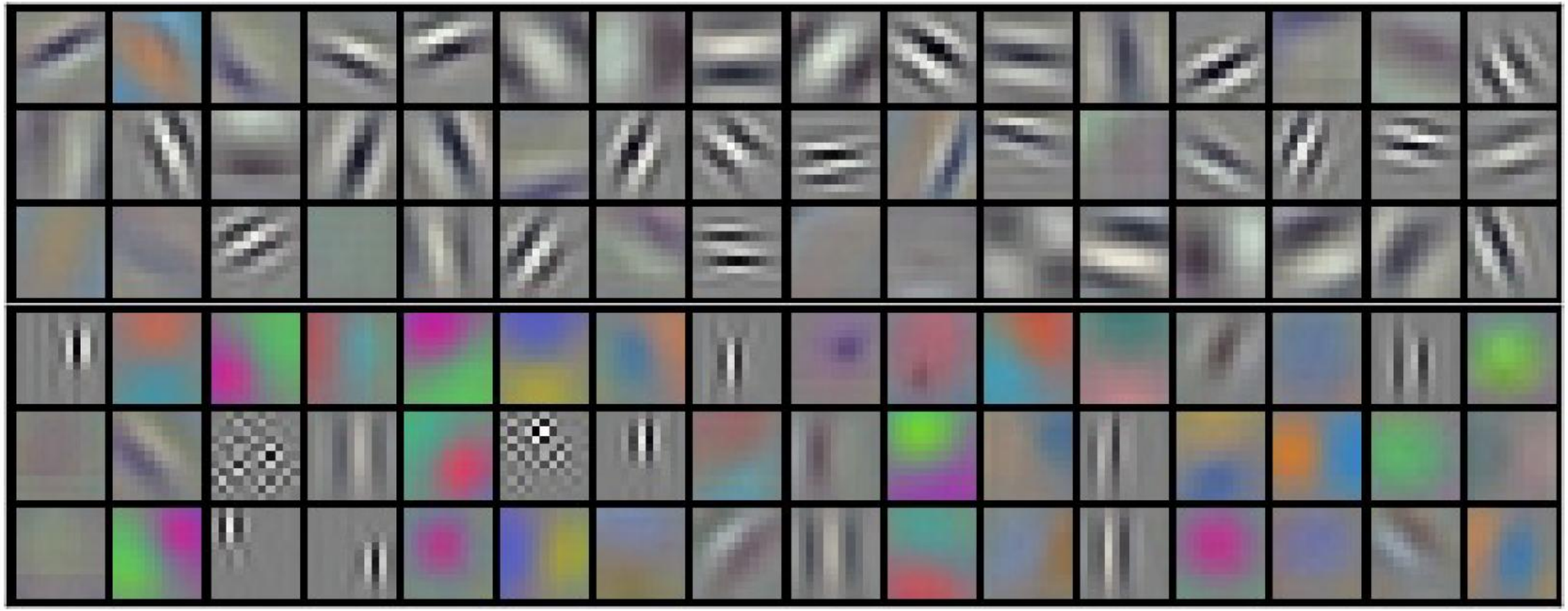


- **权重初始化**：所有权重都使用零均值高斯分布（标准差为0.01）进行初始化。
- **偏置bias**：
  - 第2，4，5个卷积层及全连接层7，8使用常数1进行初始化；
  - 第1，3个卷积层使用常数0进行初始化
- **分段学习率**：所有层使用相同的学习率，并使用启发式学习率策略（验证误差不再下降时，进行一次学习率调节）。
  - 起始学习率learning_rate=0.01，总共训练90个周期
  - 每次验证误差停止下降时，学习率降低10倍，总共降低三次，即：[0.01, 0.001, 0.0001, 0.00001]



训练细节

96个低层（第一层）卷积核学习到的特征

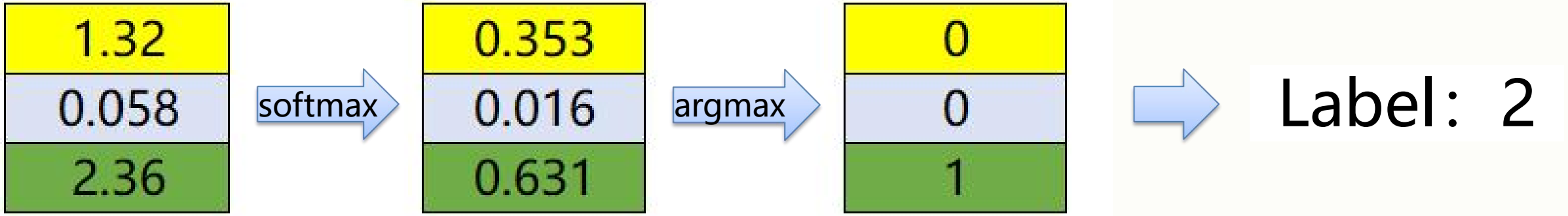


卷积核：11×11×3，从224×224×3的输入样本中学习获得。

测试策略

基于平均概率的标签预测

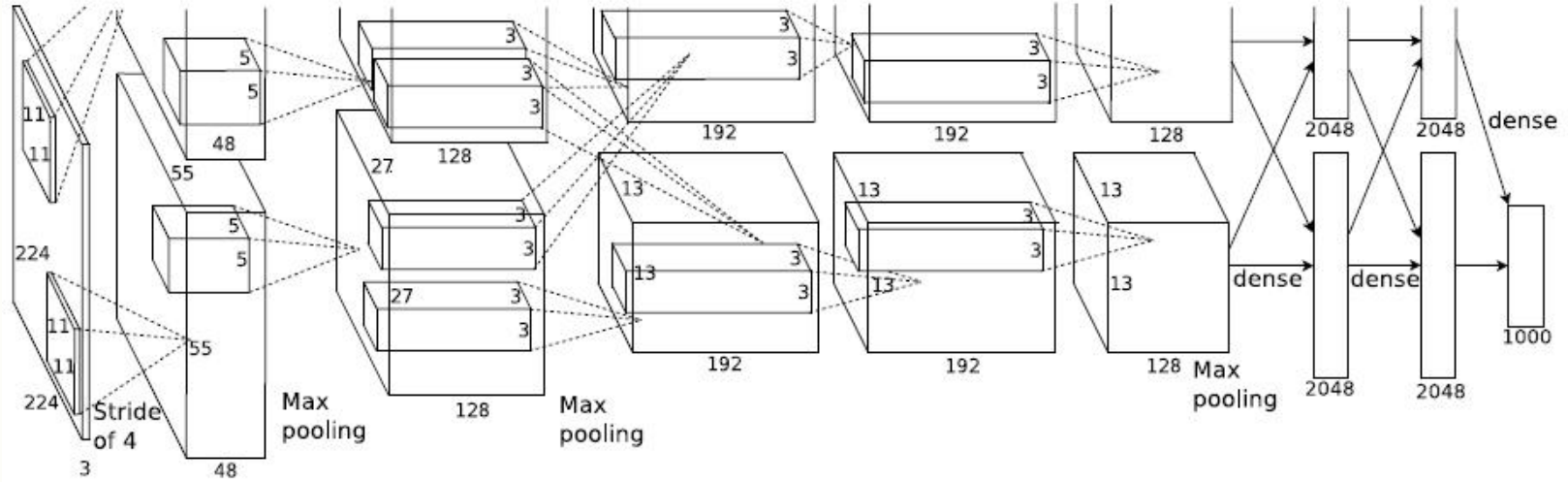
- 模型的预测由10个样本（patch）的平均值获得，即5个224×224的切割patch和它们的水平翻转。
- 基于逻辑回归的Softmax，能够较好的输出基于类别标签的概率分布。因此，输出分数不需要再进行额外的归一化（与SVM不同）。
- 预测结果直接输出分类概率最大的标签索引。



多GPU训练

多GPU训练 - 模型并行

AlexNet当时使用了GTX580的GPU进行训练，由于单个GTX 580 GPU只有3GB内存，这限制了在其上训练的网络的最大规模，因此他们在每个GPU中放置一半核（或神经元），将网络分布在两个GPU上进行并行计算，大大加快了AlexNet的训练速度。这种并行方法称为模型并行。



但由于其复杂性，该方法之后不再被使用，更通用是：**基于Batch的多GPU并行。**

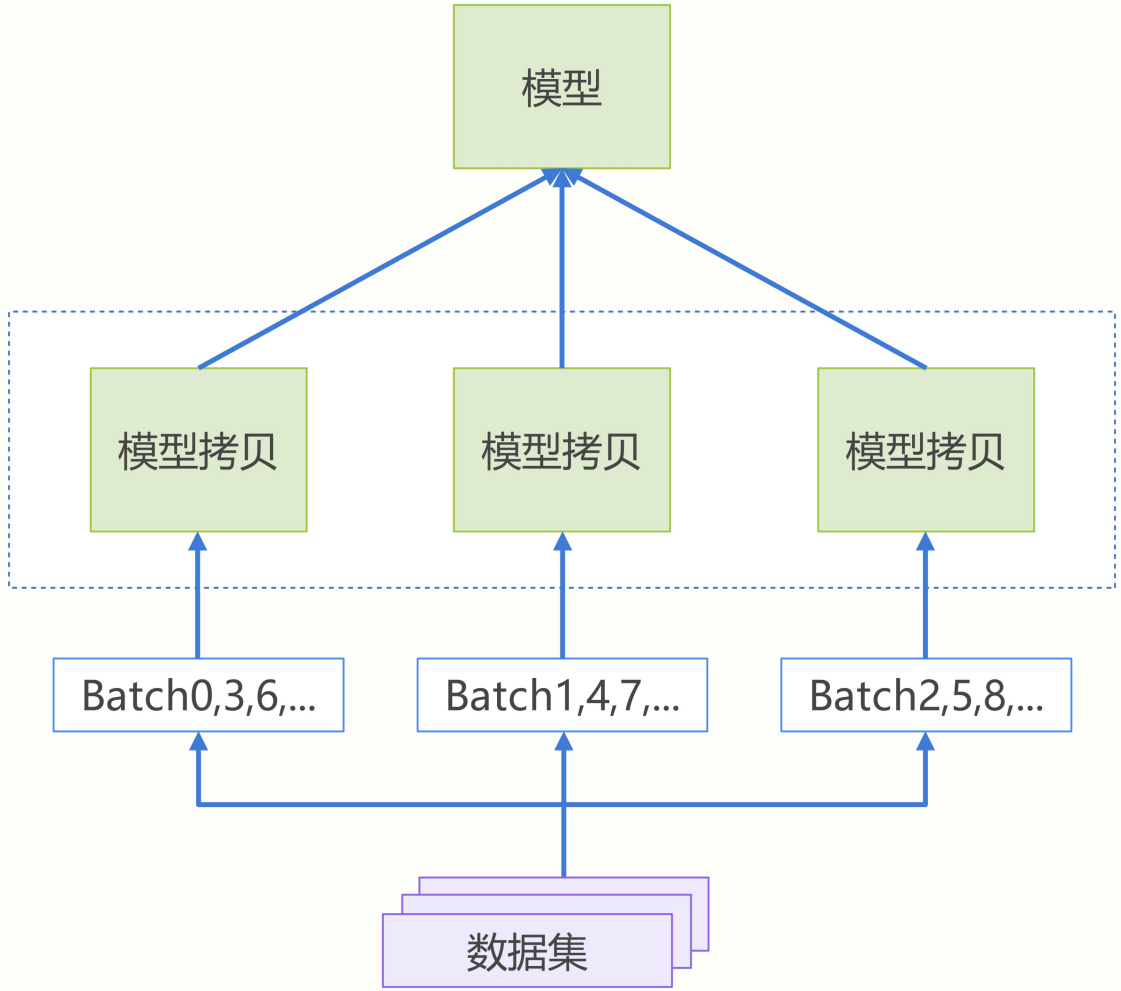
多GPU训练

多GPU训练 - 数据并行

数据并行：

指将建立好的神经网络复制到每台机器上，每台机器上有相同的模型，但是每台机器使用不同的数据进行模型的训练。

不同的数据batch会被分到不同的GPU上进行训练，并在最后进行梯度合并，然后再划分到不同的GPU上执行反向传播更新梯度。

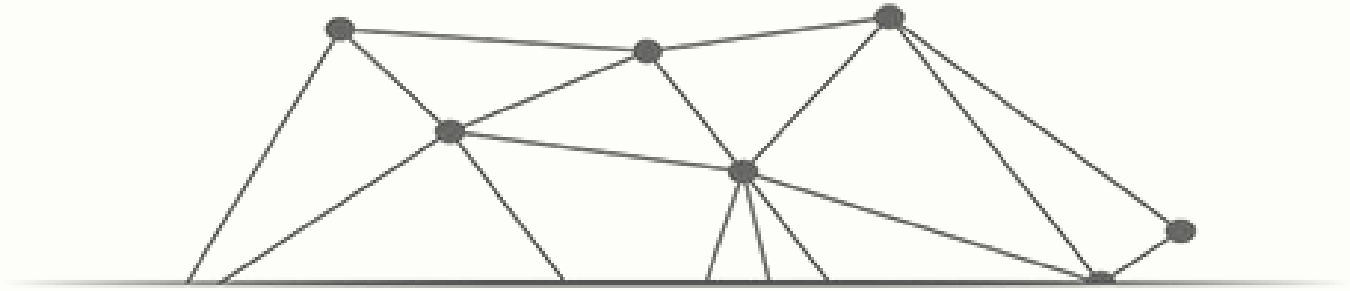


深度神经网络（AlexNet）

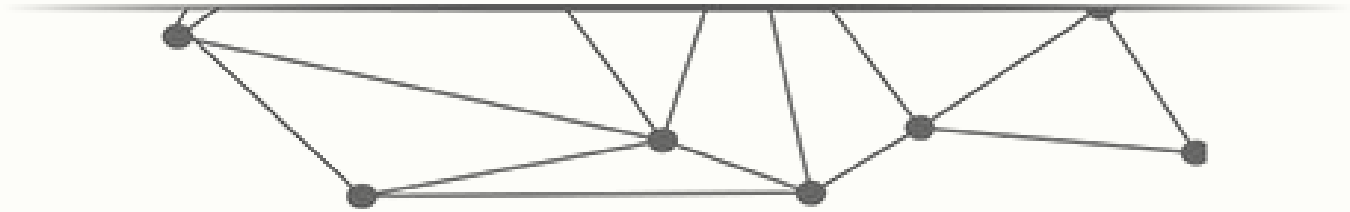
小　结

1. 卷积神经网络的使用；
2. AlexNet是一个更大、更深的LeNet，990倍参数、250倍计算复杂度；
3. 多种优化策略的整合，包括：数据增广、数据预处理（减均值）、Dropout、有重叠池化、非线性激活ReLU、局部正则响应、多GPU训练、多模型融合等；
4. AlexNet赢得了2012年ImageNet竞赛，标志着新一轮神经网络（深度学习）热潮的开始。

虽然以上很多方法都非原创，但**有机的整合**让它们发挥了巨大的作用，这也说明了基础技术与基础技术整合的**重要性**。



课堂互动 7.1.5



欧老师的联系方式

读万卷书 行万里路 只为最好的修炼

QQ：14777591 (宇宙骑士)

Email：ouxinyu@alumni.hust.edu.cn

Website: http://ouxinyu.cn

Tel：18687840023

地址：安宁校区 诚远楼201

南院 智能应用研究院A306-2