

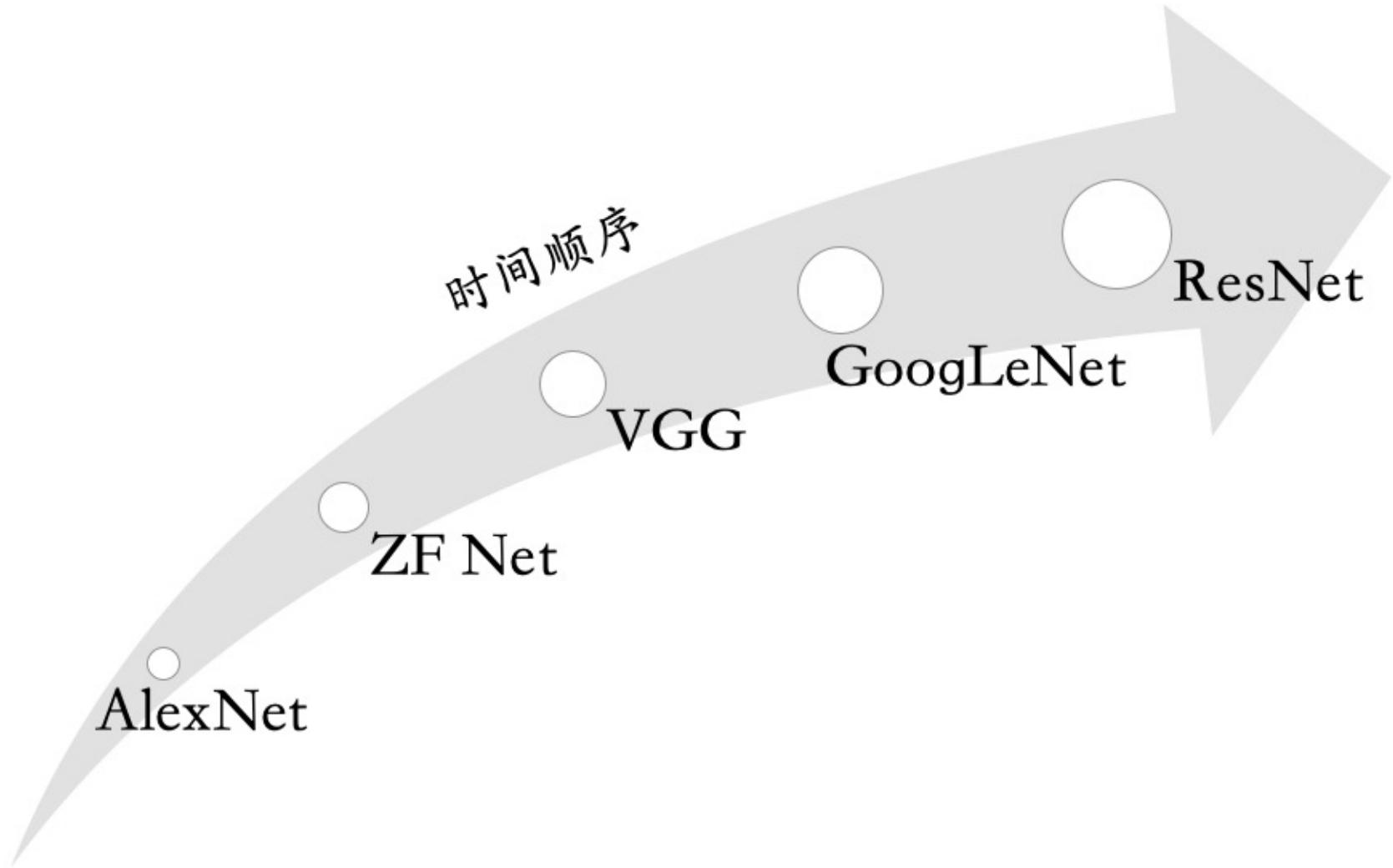


经典CNN概览 (1)

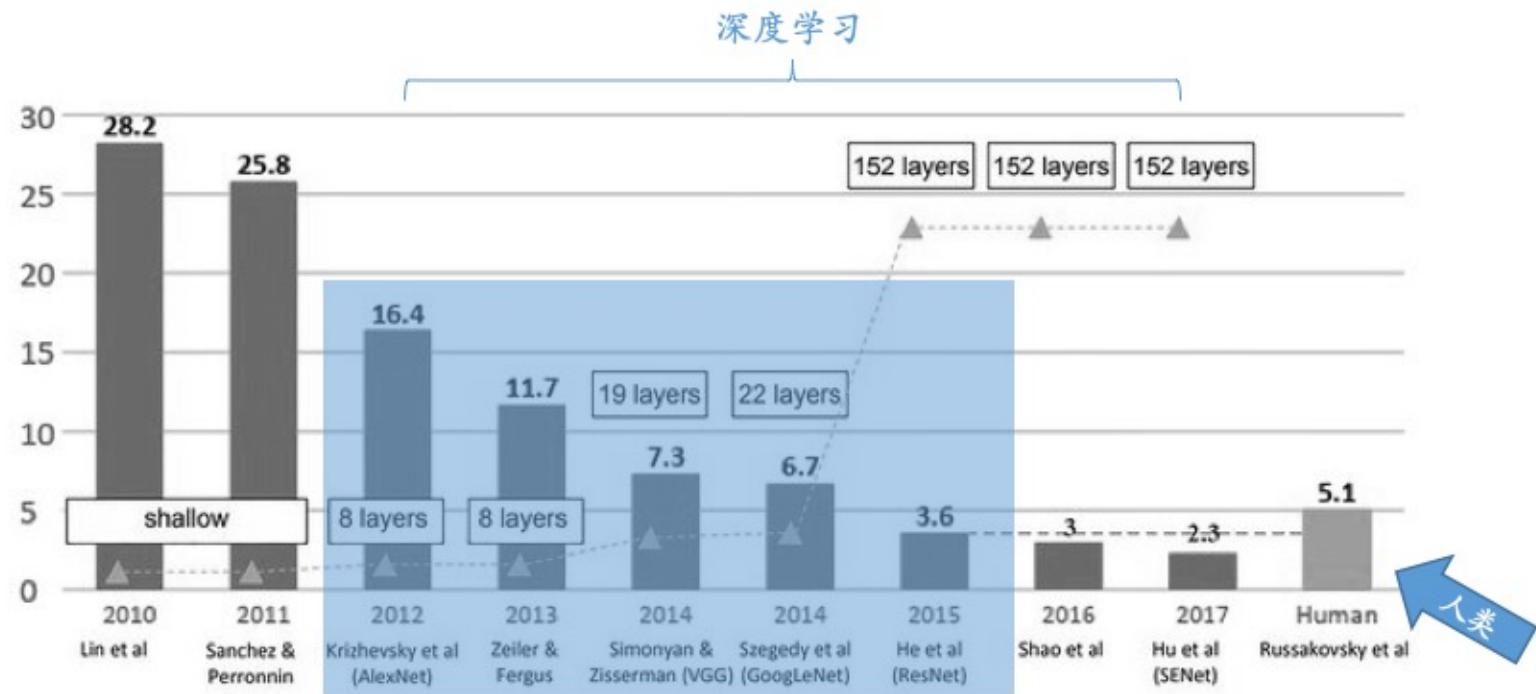
叶山 中国地质大学（北京）

yes@cugb.edu.cn

经典CNN结构



经典CNN结构



ILSVRC 历年冠军的Top 5分类结果错误率

AlexNet (8层)

AlexNet的影响力

- Charles Darwin (1859) “On the origin of species” (达尔文《物种起源》) 引用次数: 6万+
- Claude Shannon (1948) “A mathematical theory of communication” (香农《通信的数学理论》) 引用次数: 14万+
- James Watson and Francis Crick (1953) “Molecular structure of nucleic acids” (沃森和克里克《核酸的分子结构》) 引用次数: 约1.9万
- Oliver Lowry et al. (1951) “Protein measurement with the Folin phenol reagent” (洛瑞等《用Folin酚试剂测量蛋白质》) 引用次数: 30万+ (目前全球引用次数最多的论文)
- Alex Krizhevsky et al. (2012) “ImageNet classification with deep convolutional neural networks” 引用次数: 15万+

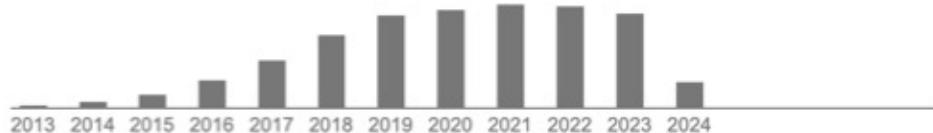
ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Cited by 154119



AlexNet的提出人



Alex Krizhevsky

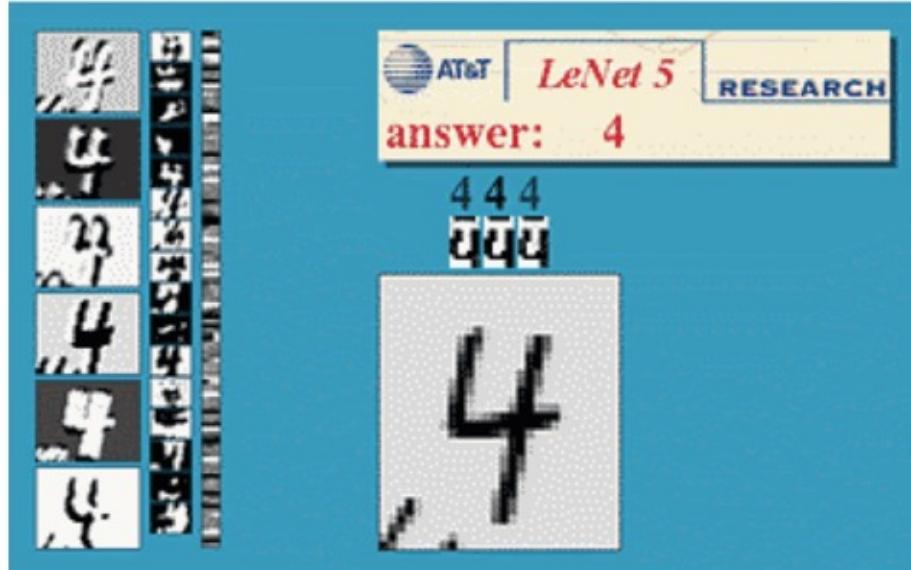


Ilya Sutskever



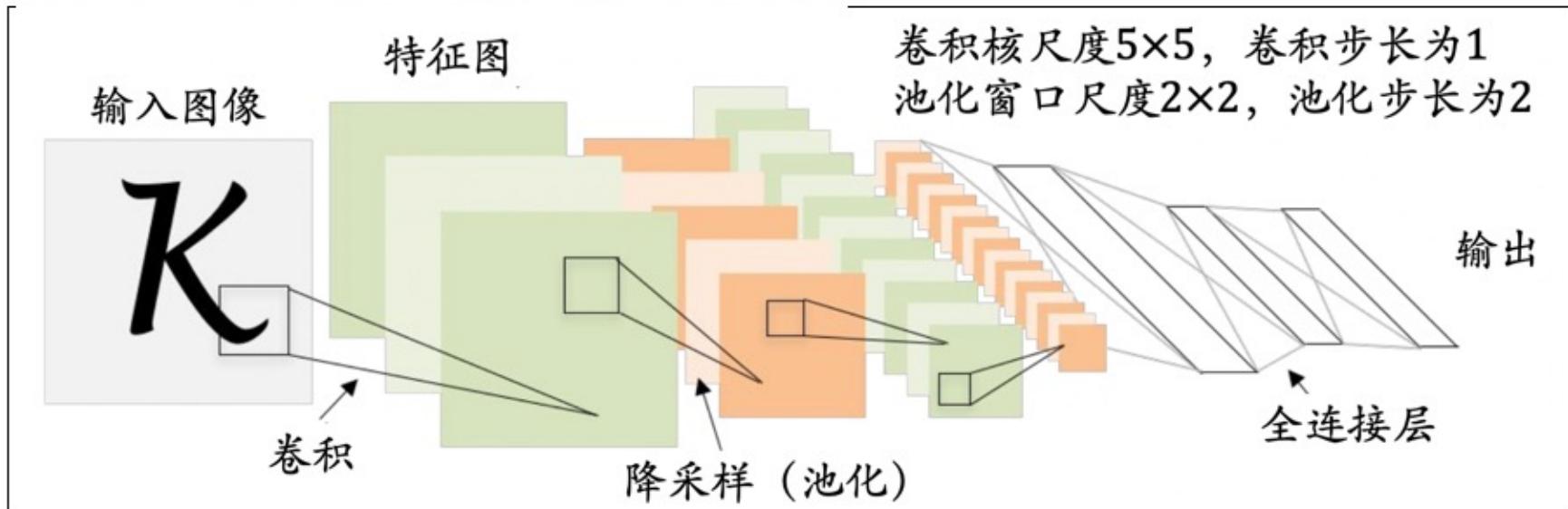
Geoffrey Everest Hinton

AlexNet的源头： LeNet



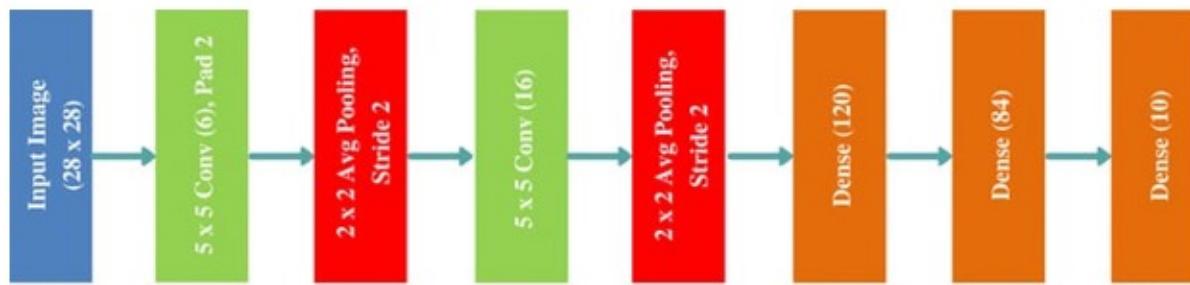
3	6	8	1	7	9	6	6	9	1
6	7	5	7	8	6	3	4	8	5
2	1	7	9	7	1	2	8	4	6
4	8	1	9	0	1	8	8	9	4
7	6	1	8	6	4	1	5	6	0
7	5	9	2	6	5	8	1	9	7
2	2	2	2	3	4	4	8	0	
0	2	3	8	0	7	3	8	5	7
0	1	4	6	4	6	0	2	4	3
7	1	2	8	1	6	9	8	6	

LeNet：由Yann LeCun（杨立坤）发明的手写数字识别网络，最初的设计动机为识别邮编。

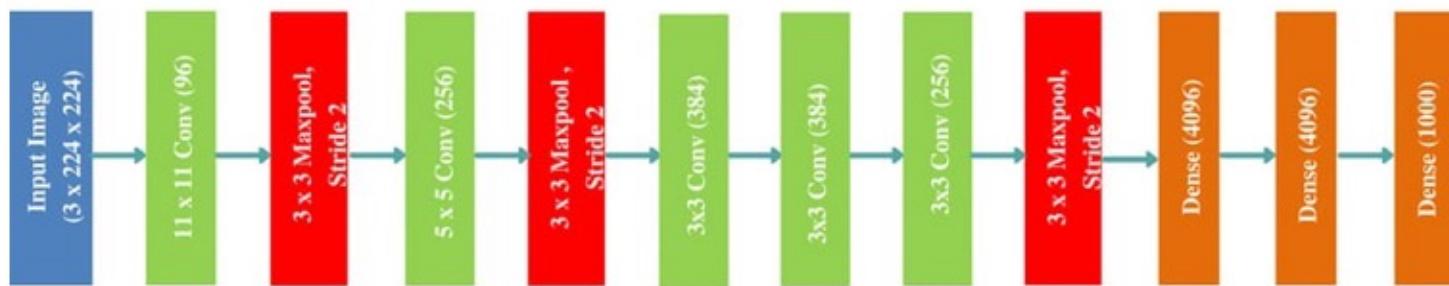


AlexNet比LeNet的进步

增加了层数，使用ImageNet数据集训练，极大地扩展了应用范围（不再局限于手写数字）。



LeNet Architecture



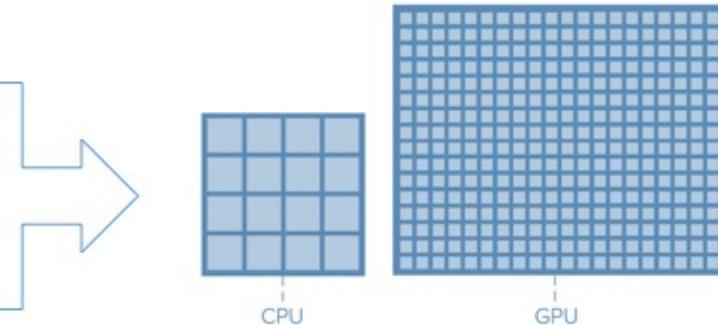
AlexNet Architecture

AlexNet的主要贡献

- 卷积层+全连接层的构架（CNN的构架）
- 首次使用ReLU作为激活函数
- 首次使用Dropout来做抑制过拟合
- 在优化过程中正式采用了动量法
- 第一次使用了数据增强的策略
- 把GPU的并行计算能力引入到深度学习领域

AlexNet+GPU

GPU的每一个核心较弱，但足以应付深度学习每一步的简单运算。通过让更多的核心实现并行计算，提高学习效率。



对比方面	CPU	GPU
全称	中央处理器	图形处理器
组成占比	执行单元占25% 控制单元占25% 缓存单元占50%	执行单元占90% 控制单元占5% 缓存单元占5%
核心数量	通常有4-8个核心	数百甚至上千个核心
核心性能及特点	单个核心更强大，内存也更多，能在不同的指令集之间快速切换（低延迟）	单个核心较弱，内存较少，但能获取大量相同的指令，并高速推进任务（高吞吐量）
执行特点	快速处理步骤和逻辑复杂的任务	将任务分解为简单且独立的部分，以便同时处理
适用领域	更适用于串行计算，尤其是快速执行需要复杂逻辑来控制的任务	专为并行计算而构建：有很多并行的执行单元，而且图形处理是最典型的并行计算案例

AlexNet的结构

如何描述CNN的层数？

- 卷积层+全连接层
- 池化层、激活层、归一化层不会单独出现，可视为卷积层的配套后续操作。

AlexNet有多少层？

- 5个卷积层+3个全连接层，因此AlexNet层数为8。

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出

AlexNet的结构

输入图像的要求：

- 尺度为 $227 \times 227 \times 3$
- 输入前要做去均值处理：
我们比的是相对数值，而非绝对数值



AlexNet的结构

第一个卷积层：

- 卷积核个数： 96
- 卷积核平面尺度： 11×11
- 卷积步长： 4
- Padding： 2

输入图像尺度为 $227 \times 227 \times 3$ ，该层输出的特征图尺度是多少？

$56 \times 56 \times 96$

卷积层后面跟随非线性操作 (ReLU)

输入

卷积层+ReLU，边长11，步长4，数量96



局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出

AlexNet的结构

第一个卷积层：

- 卷积核个数： 96
- 卷积核平面尺度： 11×11
- 卷积步长： 4
- Padding： 2

PyTorch版为
64个卷积核

Keras版
这一层没有
做边缘填充

输入图像尺度为 $227 \times 227 \times 3$ ，该层输出的特征图尺度是多少？

$56 \times 56 \times 96$

大约3.5万个模型参数

输入

输出

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

AlexNet的结构

局部响应归一化层：

- Local Response Normalization Layer (LR Norm)，不改变维度
- 在小范围内增强最大值、抑制较小值，旨在突出特点、增强模型的泛化性能。
- 后来的研究发现，在更深层的网络中，它的作用不明显，却在浪费算力，现在已经被淘汰了。

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层



最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

输出

全连接层+Softmax，神经元个数1000

AlexNet的结构

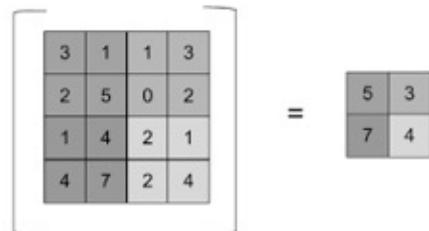
Max Pooling 层：

- 目的：对抗轻微的目标偏移给分类任务带来的影响。
- 池化窗口大小： 3×3
- 池化步长：2
- 步长大于窗口边长，说明相邻两次池化操作有重复的覆盖范围（重叠池化）→旨在抹平空间上的剧烈变化（过于细节的特征），对抗过拟合。现已不常用。

输出维度： $27 \times 27 \times 96$

该层没有模型参数

Max



输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出



AlexNet的结构

第二个卷积层：

- 卷积核个数： 256
- 卷积核平面尺度： 5×5
- 卷积步长： 1
- Padding： 2层0填充

输入维度： $27 \times 27 \times 96$

输出维度： $27 \times 27 \times 256$

输入

输出

卷积层+ReLU， 边长11， 步长4， 数量96

局部响应归一化层

最大池化层， 边长3， 步长2

卷积层+ReLU， 边长5， 步长1， 数量256

局部响应归一化层

最大池化层， 边长3， 步长2

卷积层+ReLU， 边长3， 步长1， 数量384

卷积层+ReLU， 边长3， 步长1， 数量384

卷积层+ReLU， 边长3， 步长1， 数量256

最大池化层， 边长3， 步长2

全连接层+ReLU， 神经元个数4096

全连接层+ReLU， 神经元个数4096

全连接层+Softmax， 神经元个数1000



AlexNet的结构

第二个Max Pooling层：

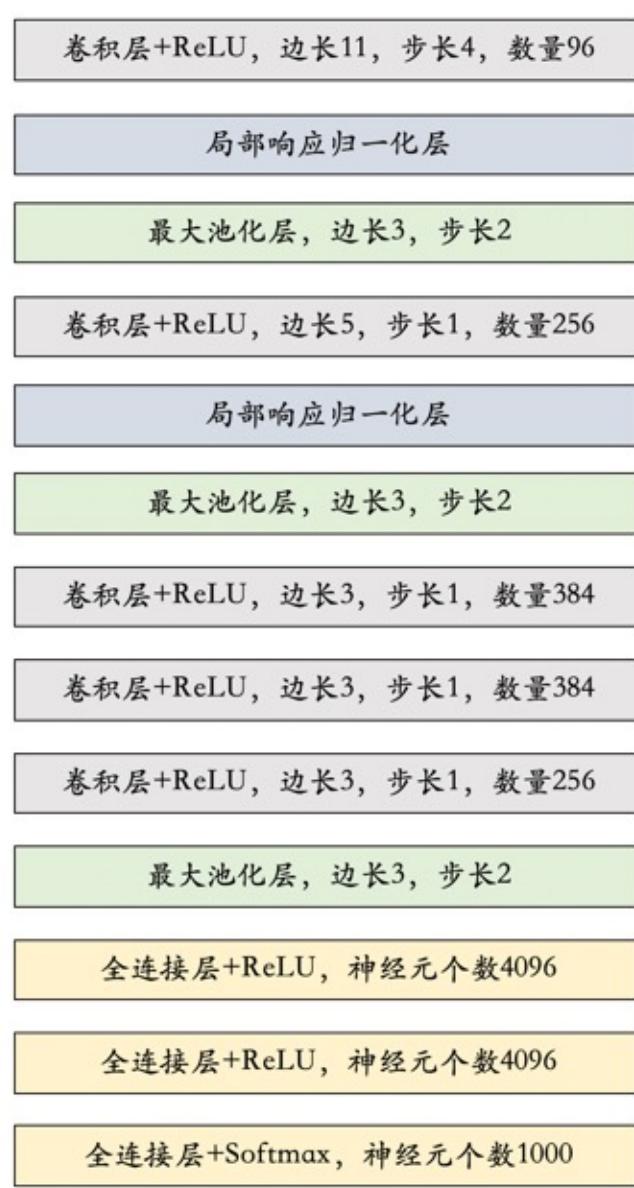
- 池化窗口大小： 3×3
- 池化步长： 2

输入维度： $27 \times 27 \times 256$

输出维度： $13 \times 13 \times 256$

输入

输出

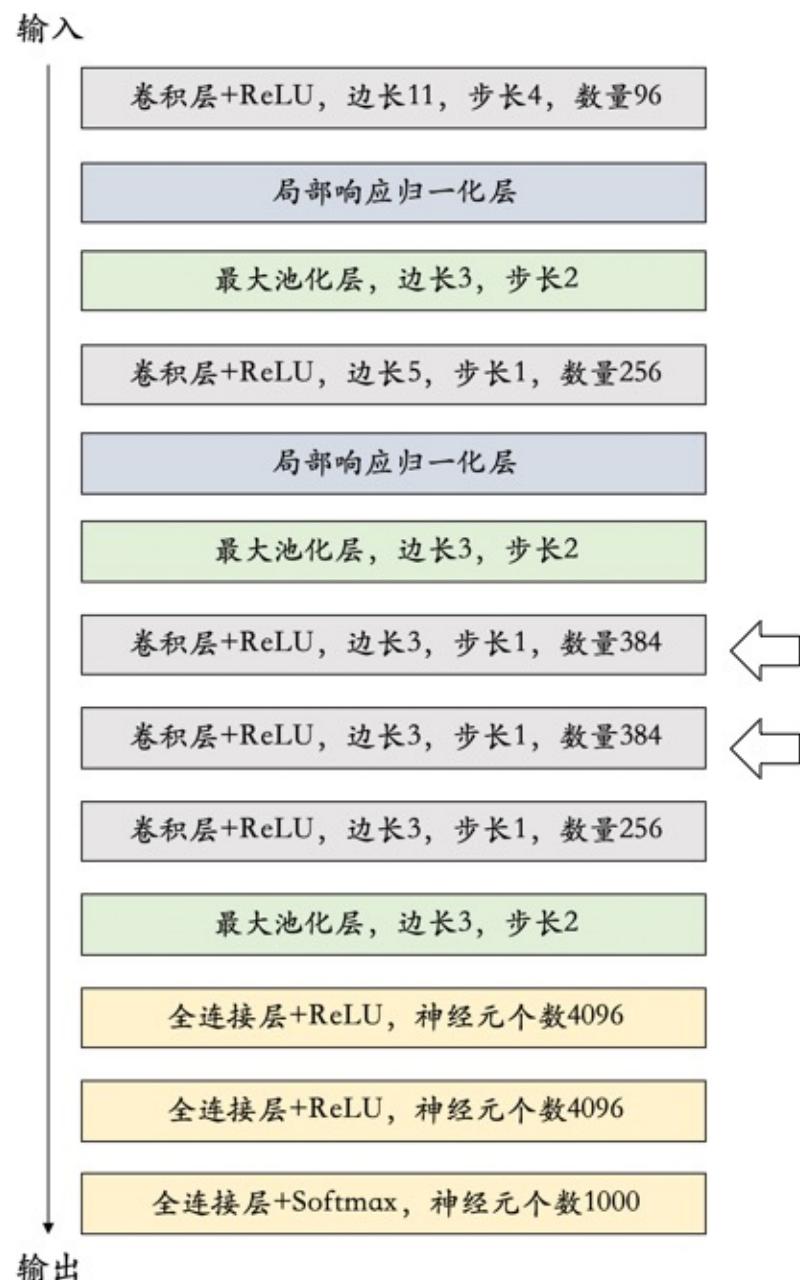


AlexNet的结构

第三、第四个卷积层：

- 卷积核个数： 384
- 卷积核平面尺度： 3×3
- 卷积步长： 1
- Padding： 1层0填充

输出维度： $13 \times 13 \times 384$



AlexNet的结构

第五个卷积层：

- 卷积核个数： 256
- 卷积核平面尺度： 3×3
- 卷积步长： 1
- Padding： 1层0填充

输出维度： $13 \times 13 \times 256$

输入

输出

卷积层+ReLU， 边长11， 步长4， 数量96

局部响应归一化层

最大池化层， 边长3， 步长2

卷积层+ReLU， 边长5， 步长1， 数量256

局部响应归一化层

最大池化层， 边长3， 步长2

卷积层+ReLU， 边长3， 步长1， 数量384

卷积层+ReLU， 边长3， 步长1， 数量384

卷积层+ReLU， 边长3， 步长1， 数量256

最大池化层， 边长3， 步长2

全连接层+ReLU， 神经元个数4096

全连接层+ReLU， 神经元个数4096

全连接层+Softmax， 神经元个数1000



AlexNet的结构

第三个Max Pooling层：

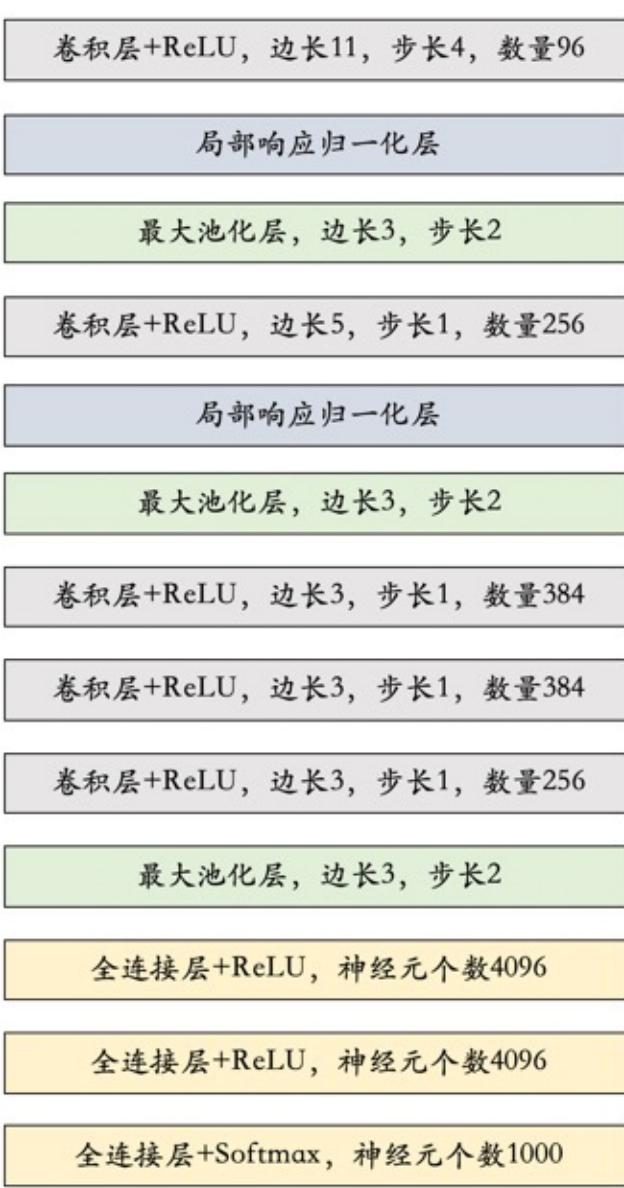
- 池化窗口大小： 3×3
- 池化步长： 2

输入维度： $13 \times 13 \times 256$

输出维度： $6 \times 6 \times 256$

输入

输出



AlexNet的结构

全连接层：

- 输入：把卷积层获得的 $6 \times 6 \times 256$ 的特征矩阵展开（flatten）成向量
- 神经元个数：4096（隐藏层）、1000（输出层）

输入向量长度：9216

输出得分向量长度：1000（分类任务的类别）

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

全连接层+ReLU，神经元个数4096

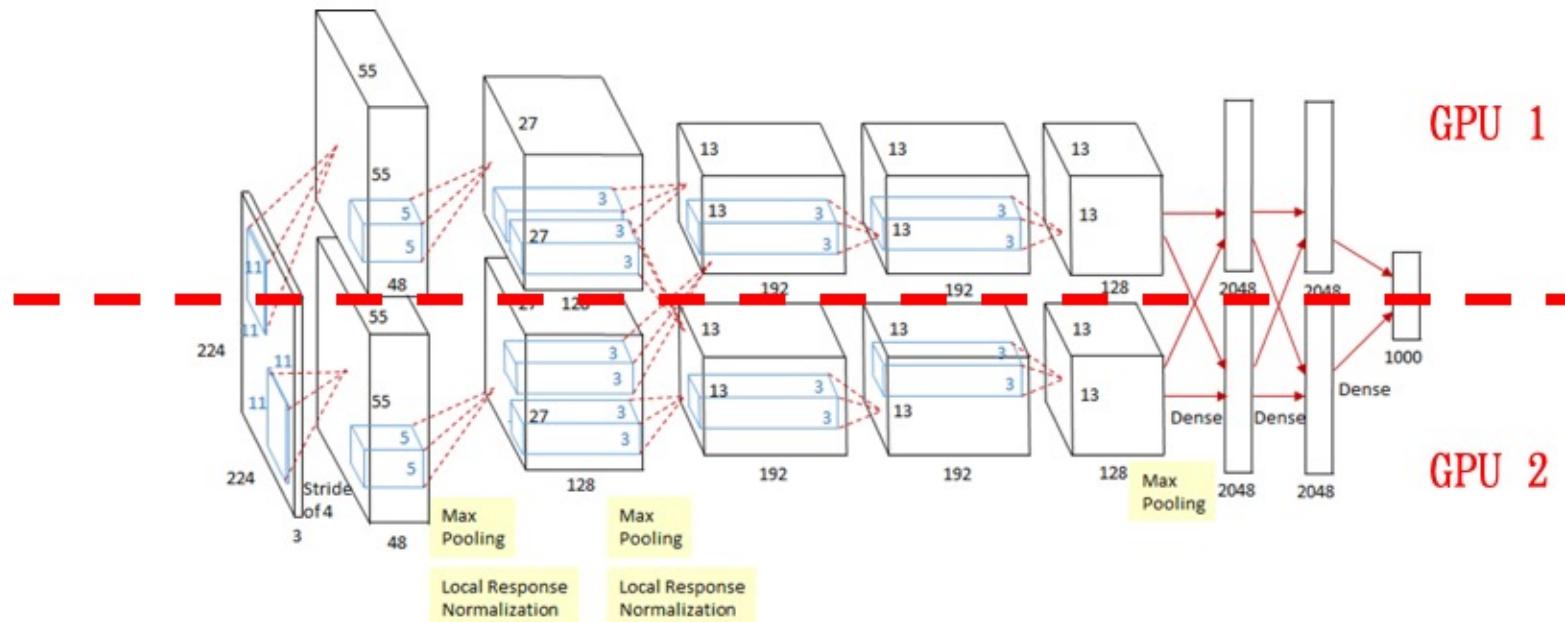
全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出

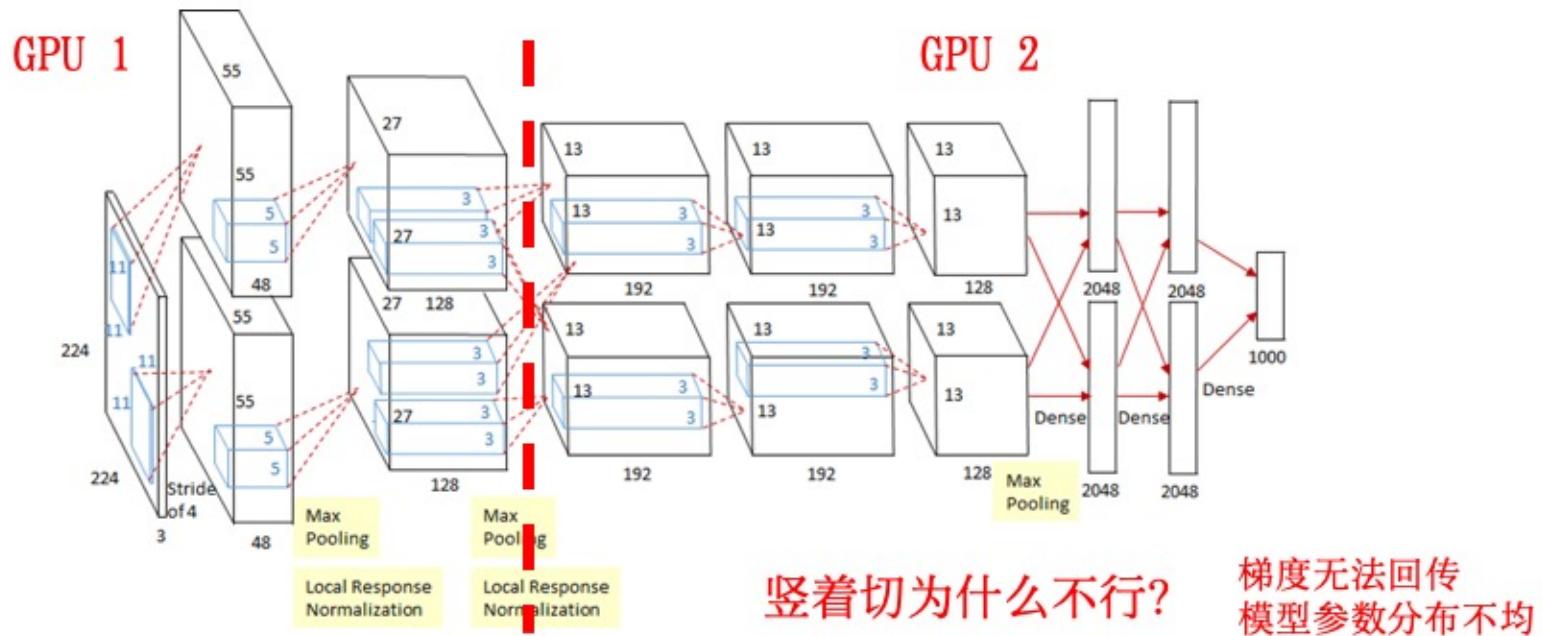
AlexNet的训练

- 卷积层和全连接层是一起训练的（“端对端”学习）。梯度可以从全连接层的输出层，回传到卷积层的图像输入。
- 分割为两部分，在两个显卡中同时训练。

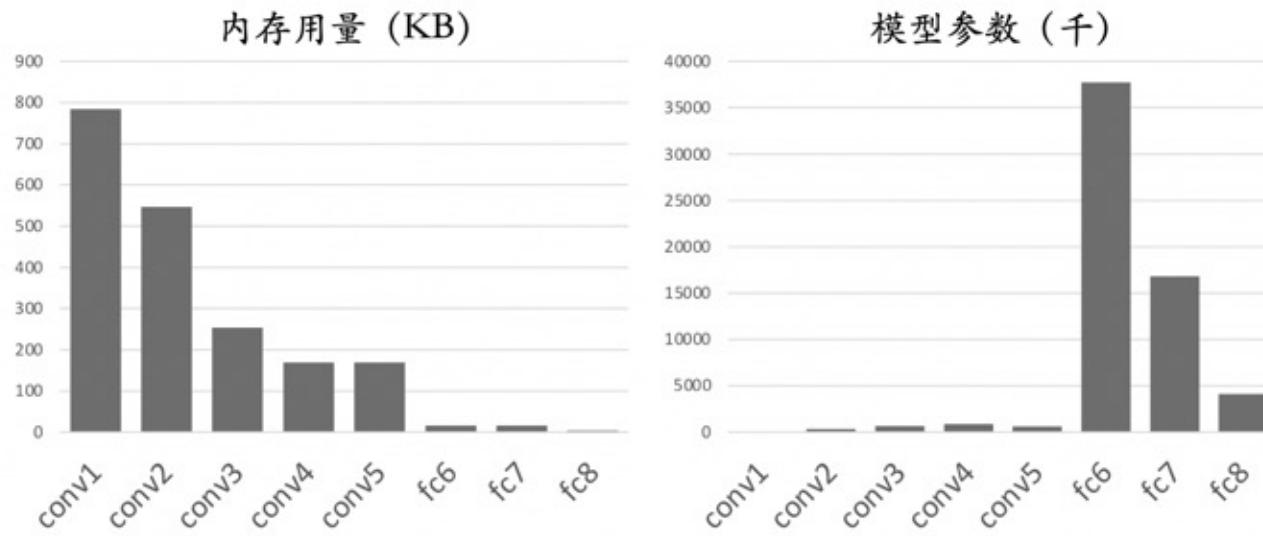


AlexNet的训练

- 卷积层和全连接层是一起训练的（“端对端”学习）。梯度可以从全连接层的输出层，回传到卷积层的图像输入。
- 分割为两部分，在两个显卡中同时训练。



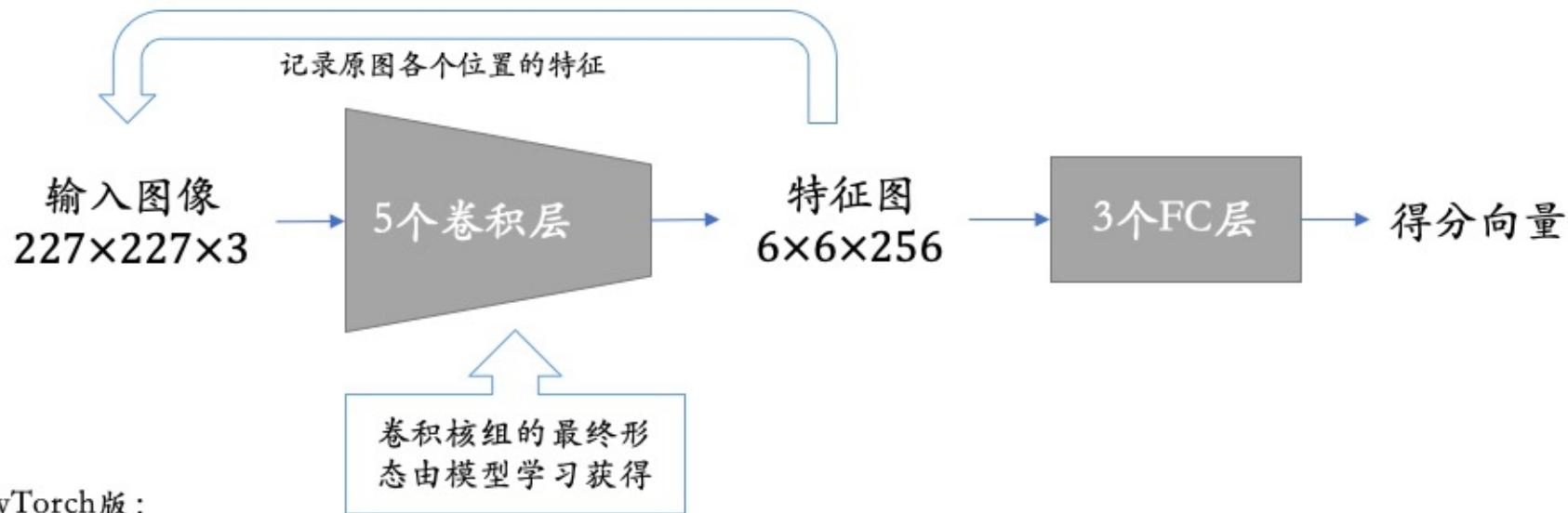
AlexNet的训练



PyTorch版：

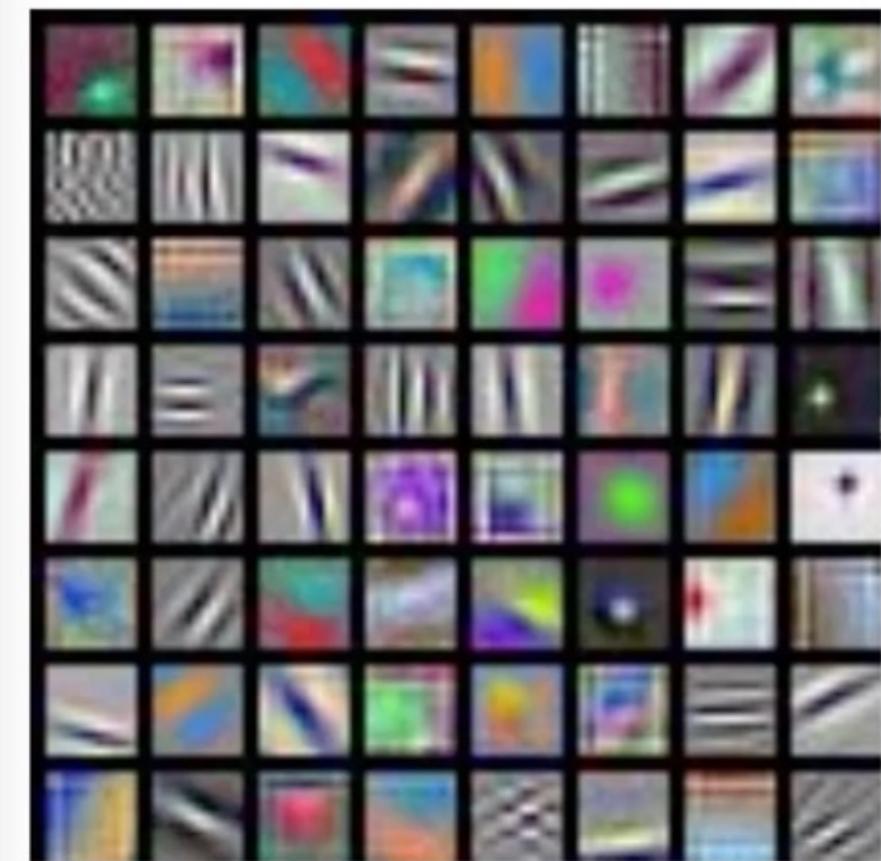
层次	输入维度	卷积核数	神经元数	滤波器边长	步长	Padding	输出维度	内存 (KB)	参数 (千)
卷积层1	227×227×3	64		11	4	2	56×56×64	784	23
池化层	56×56×64			3	2	0	27×27×64	182	0
卷积层2	27×27×64	256		5	1	2	27×27×256	547	307
池化层	27×27×256			3	2	0	27×27×256	127	0
卷积层3	27×27×256	384		3	1	1	13×13×384	254	664
卷积层4	13×13×384	384		3	1	1	13×13×384	169	885
卷积层5	13×13×384	256		3	1	1	13×13×256	169	590
池化层	13×13×256			3	2	0	6×6×256	36	0
Flatten	6×6×256							9216	0
全连接层6	9216		4096				4096	16	37749
全连接层7	4096		4096				4096	16	16777
全连接层8	4096		1000				1000	4	4096

AlexNet的训练

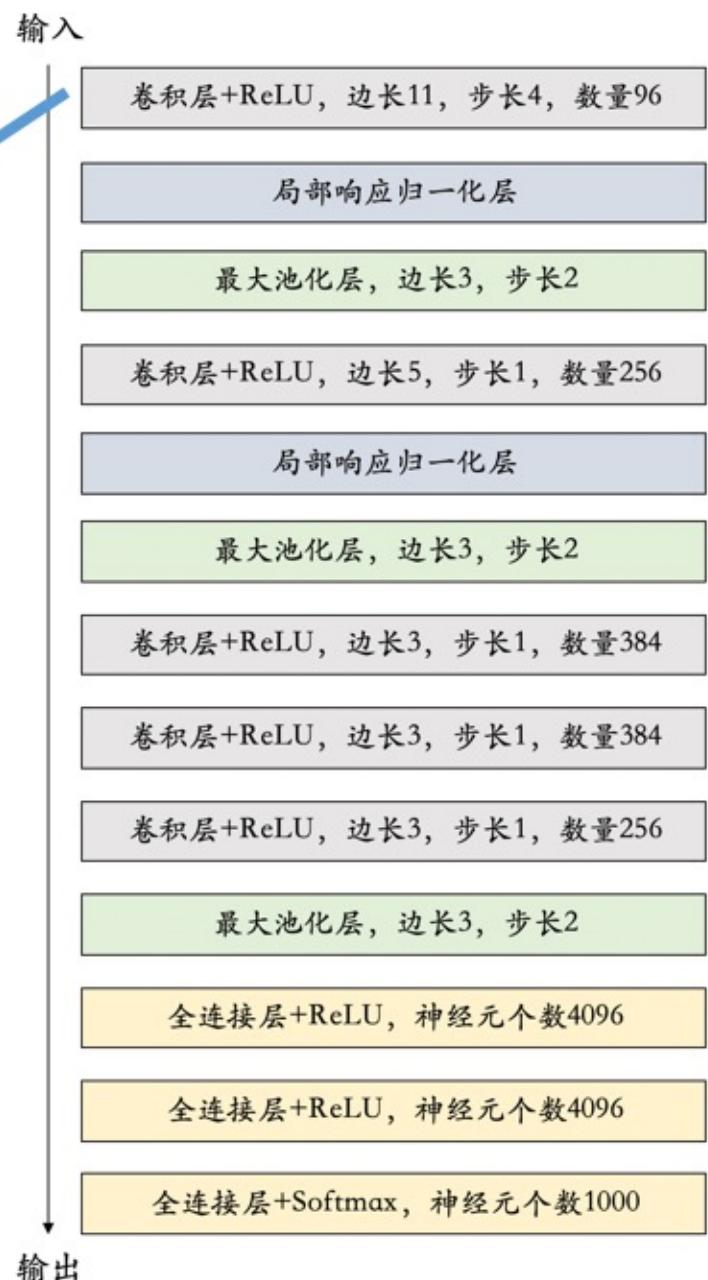


层次	输入维度	卷积核数	神经元数	滤波器边长	步长	Padding	输出维度	内存 (KB)	参数 (千)
卷积层1	$227 \times 227 \times 3$	64		11	4	2	$56 \times 56 \times 64$	784	23
池化层	$56 \times 56 \times 64$			3	2	0	$27 \times 27 \times 64$	182	0
卷积层2	$27 \times 27 \times 64$	256		5	1	2	$27 \times 27 \times 256$	547	307
池化层	$27 \times 27 \times 256$			3	2	0	$27 \times 27 \times 256$	127	0
卷积层3	$27 \times 27 \times 256$	384		3	1	1	$13 \times 13 \times 384$	254	664
卷积层4	$13 \times 13 \times 384$	384		3	1	1	$13 \times 13 \times 384$	169	885
卷积层5	$13 \times 13 \times 384$	256		3	1	1	$13 \times 13 \times 256$	169	590
池化层	$13 \times 13 \times 256$			3	2	0	$6 \times 6 \times 256$	36	0
Flatten	$6 \times 6 \times 256$							9216	36
全连接层6	9216		4096				4096	16	37749
全连接层7	4096		4096				4096	16	16777
全连接层8	4096		1000				1000	4	4096

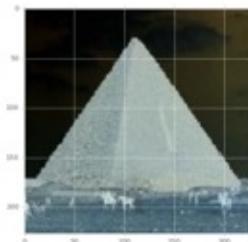
中间过程的可视化



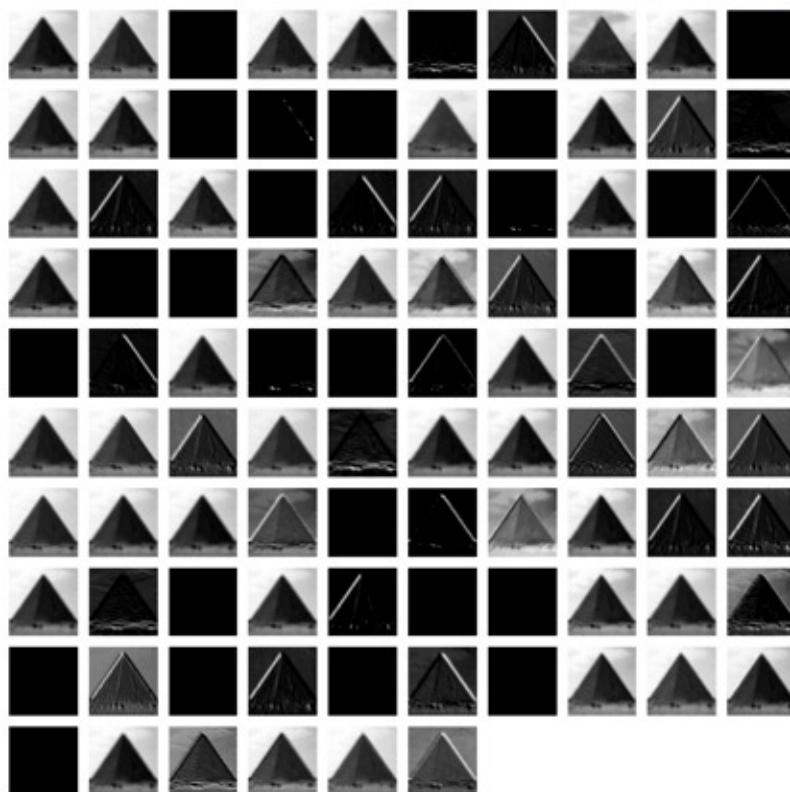
第一个卷积层的卷积核组



中间过程的可视化



输入图像



第一个卷积层的响应图

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

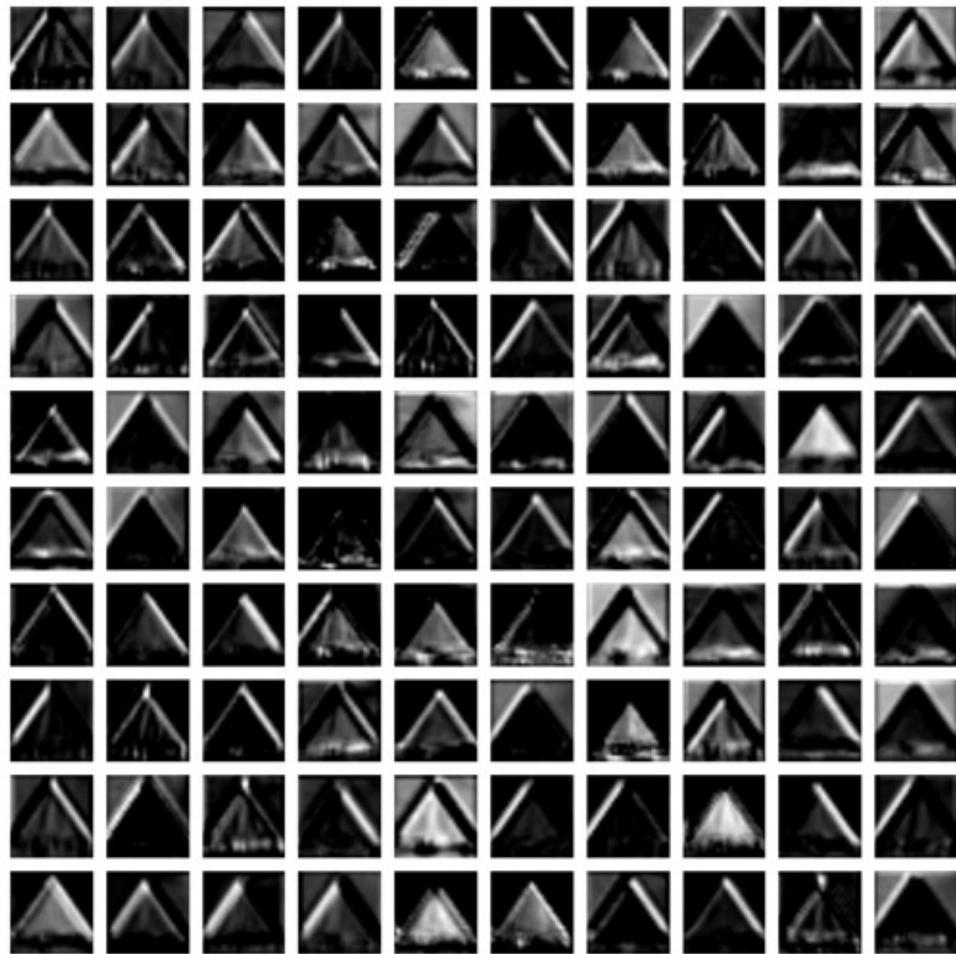
全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出

中间过程的可视化



第二个卷积层的响应图组
(前100个)

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

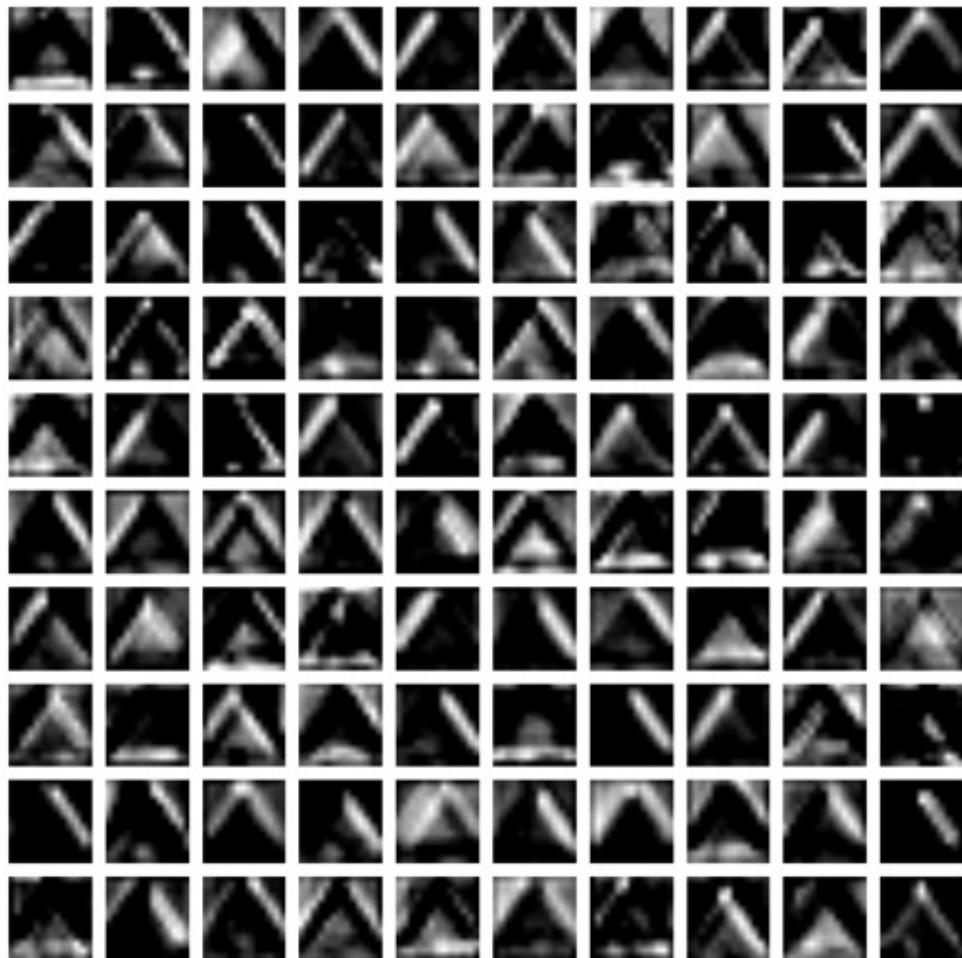
全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出

中间过程的可视化



第三个卷积层的响应图组
(前100个)

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

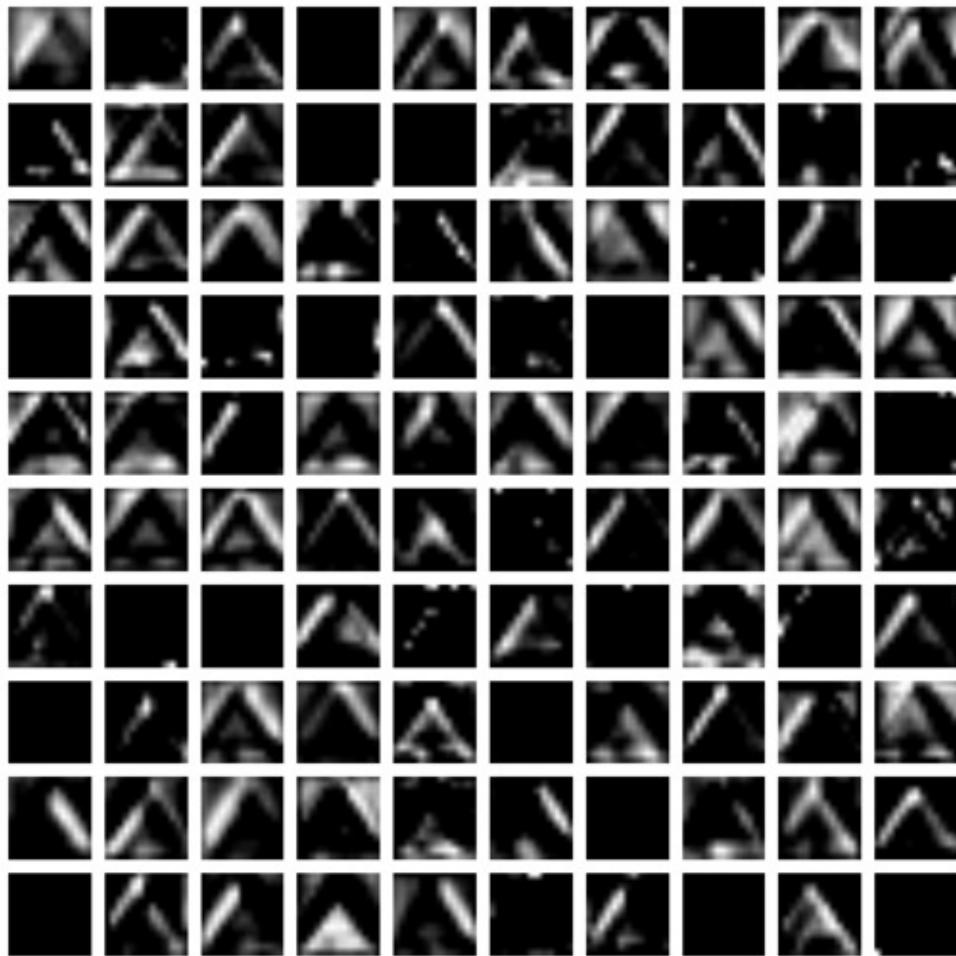
全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出

中间过程的可视化



第四个卷积层的响应图组
(前100个)

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

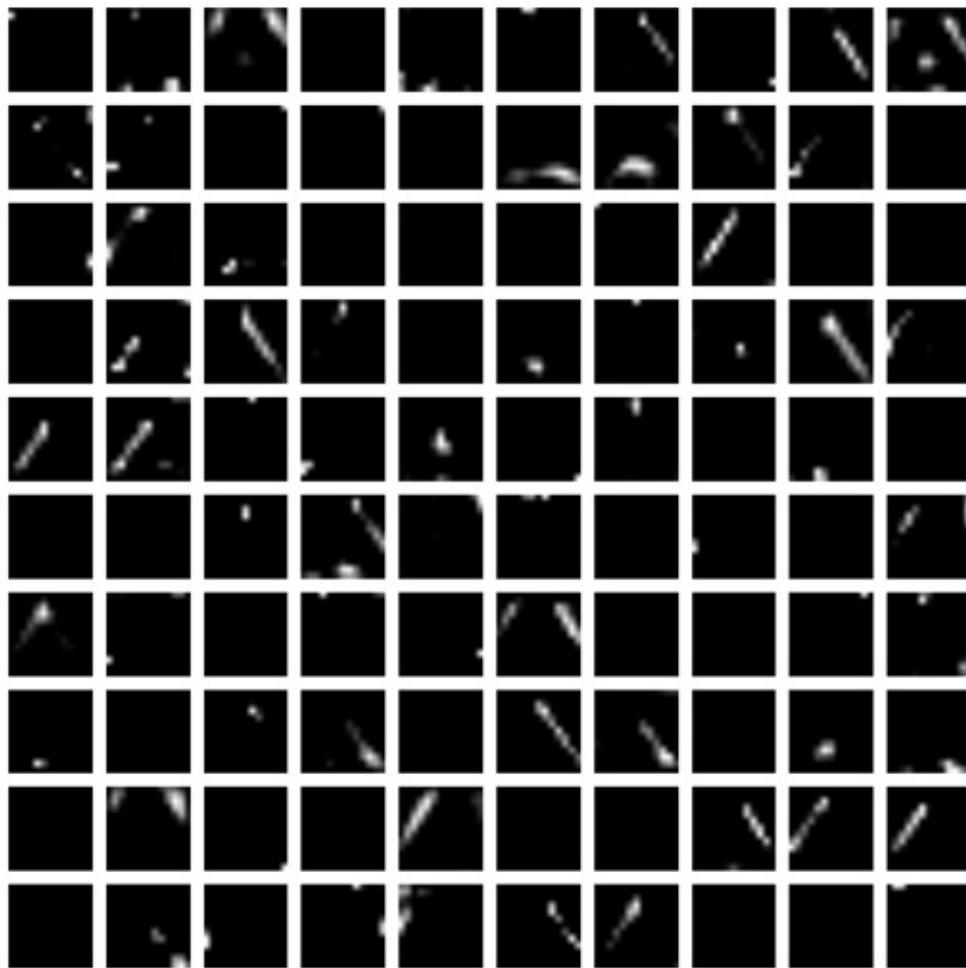
全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出

中间过程的可视化



第五个卷积层的响应图组
(前100个)

输入

卷积层+ReLU，边长11，步长4，数量96

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长5，步长1，数量256

局部响应归一化层

最大池化层，边长3，步长2

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量384

卷积层+ReLU，边长3，步长1，数量256

最大池化层，边长3，步长2

全连接层+ReLU，神经元个数4096

全连接层+ReLU，神经元个数4096

全连接层+Softmax，神经元个数1000

输出