

Deep Learning Technology and Application

Ge Li

Peking University

卷积神经网络的发展

- ① 卷积核的设计
- ② 卷积层的设计
- ③ Pooling 层的设计

带有卷积权值的网络

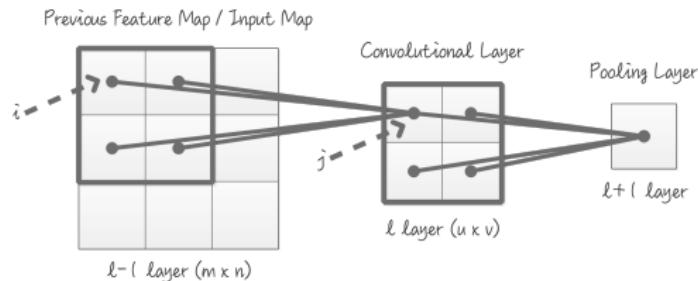
设 α_{ij} 为第 i 个输入特征图与第 j 个卷积核之间的连接强度；

$$a_j^l = f \left(\sum_{i=1}^{N_{in}} \alpha_{ij} (x_i^{l-1} * k_i^l) + b_j^l \right)$$

其中：

- N_{in} 为输入特征图的个数；
- $\sum_i \alpha_{ij} = 1$ 且 $0 \leq \alpha_{ij} \leq 1$ ；
- 可以设 $\alpha_{ij} = \frac{\exp(c_{ij})}{\sum_k \exp(c_{kj})}$ ；
- 接下来的运算，将针对一个确定的输出特征图 j ，因此上述公式可以简化为： $\alpha_i = \frac{\exp(c_i)}{\sum_k \exp(c_k)}$ ；

带有卷积权值的网络



接下来看一下，推导 α_{ij} ，即 α_i 的求解方法：

$$\frac{\partial J}{\partial \alpha_i} = \frac{\partial J}{\partial z^l} \frac{\partial z^l}{\partial \alpha_i} = \sum_{u,v} (\delta^l \circ (a_i^{l-1} * k_i^l))_{uv}$$

继续求解关于 c_i 的导数：

$$\frac{\partial J}{\partial c_i} = \sum_k \frac{\partial J}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial c_i} = \alpha_i \left(\frac{\partial J}{\partial \alpha_i} - \sum_k \frac{\partial J}{\partial \alpha_k} \alpha_k \right)$$

带有卷积权值的网络

为了能够增强 α_i 的效果，我们可以想办法让它变得更稀疏：

$$\hat{J}(W, b) = J(W, b) + \Omega(\alpha) = J(W, b) + \lambda \sum_{i,j} |(\alpha)_{ij}|$$

在这个条件下， α 如何更新呢？

$$\frac{\partial \Omega}{\partial \alpha_i} = \lambda \operatorname{sign}(\alpha_i)$$

将 α 的求解方式代入：

$$\frac{\partial \Omega}{\partial c_i} = \sum_k \frac{\partial \Omega}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial c_i} = \lambda \left(|\alpha_i| - \alpha_i \sum_k |\alpha_k| \right)$$

因此：

$$\frac{\partial \hat{J}}{\partial c_i} = \frac{\partial J}{\partial c_i} + \frac{\partial \Omega}{\partial c_i}$$

Tiled Convolution

Tiled Convolution :

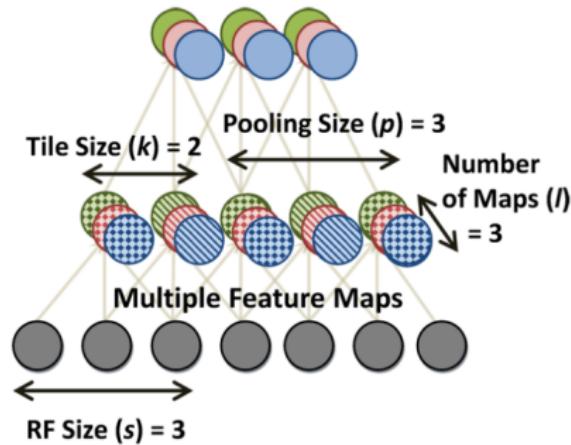
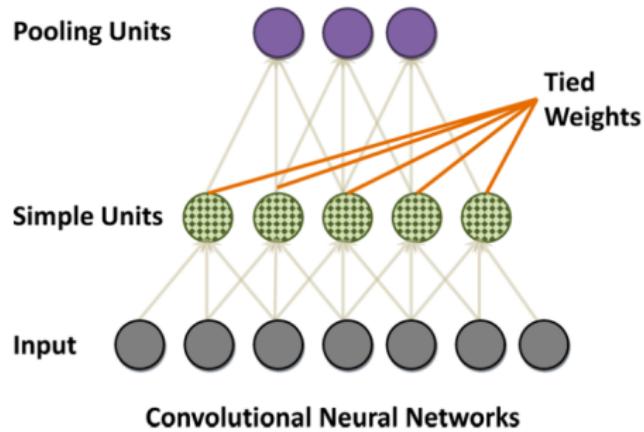
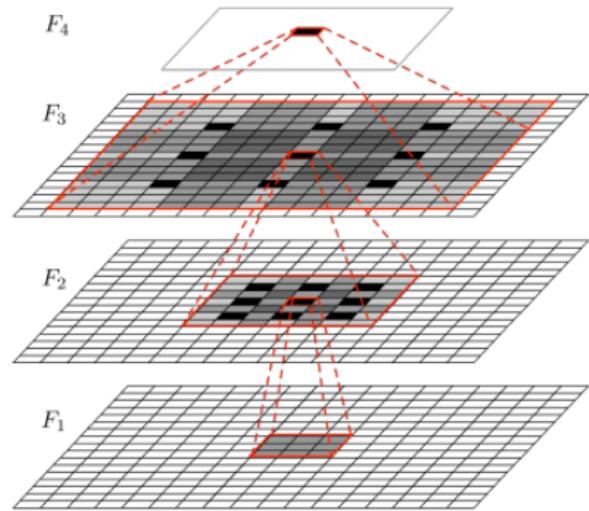


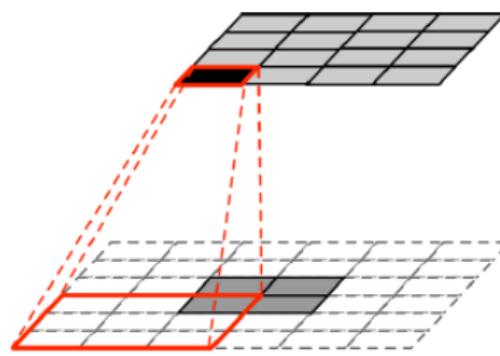
Figure 1: Left: Convolutional Neural Networks with local receptive fields and tied weights. Right: Partially untied local receptive field networks – Tiled CNNs. Units with the same color belong to the same map; within each map, units with the same fill texture have tied weights. (Network diagrams in the paper are shown in 1D for clarity.)

[1] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, A. Y. Ng, Tiled convolutional neural networks, in: NIPS, 2010.

Transposed Convolution, Dilated Convolution



(c) Dilated Convolution



(d) Deconvolution

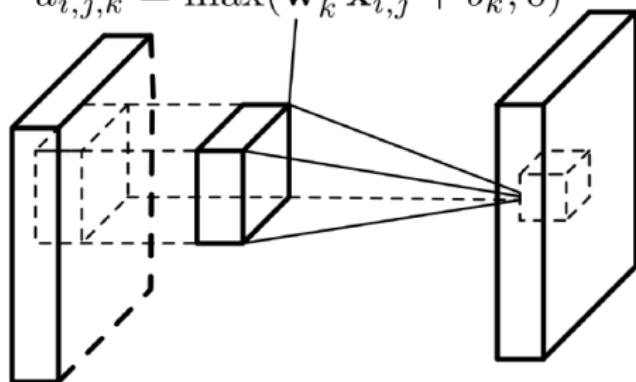
[d] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, 2014.

[c] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: ICLR, 2016.

Network in Network

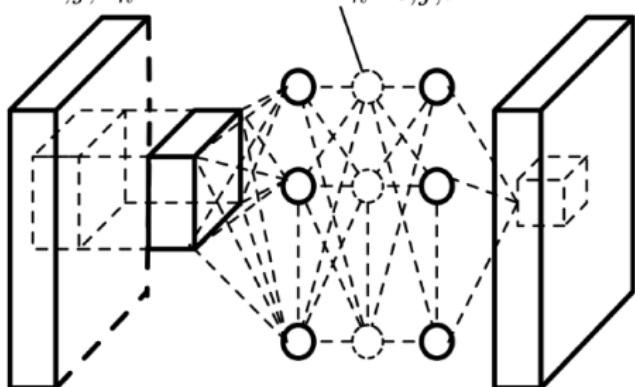
Network in Network:

$$a_{i,j,k} = \max(\mathbf{w}_k^T \mathbf{x}_{i,j} + b_k, 0)$$



(a) Linear convolution layer

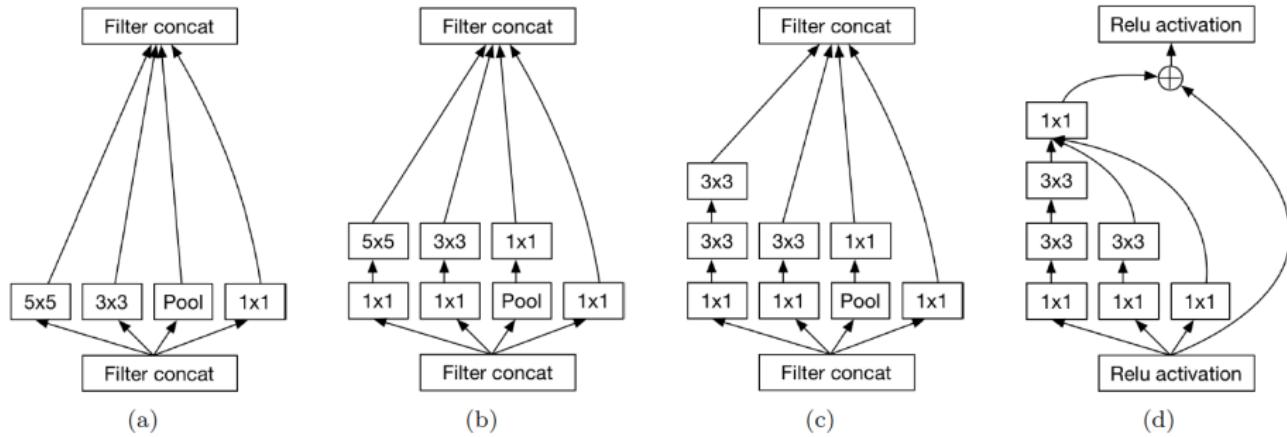
$$a_{i,j,k_n}^n = \max(\mathbf{w}_{k_n}^T \mathbf{a}_{i,j,:}^{n-1} + b_{k_n}, 0)$$



(b) Mlpconv layer

- [1] M. Lin, Q. Chen, S. Yan, Network in network, in: ICLR, 2014.

Inception Module



- [b] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, 2015.
- [c] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: arXiv preprint arXiv:1512.00567, 2015.
- [d] C. Szegedy, S. Ioffe, V. Vanhoucke, Inception-v4, inception-resnet and the impact of residual connections on learning, in: arXiv preprint arXiv:1602.07261, 2016.

L_p Pooling

L_p Pooling[1] 是一种泛化能力较强的 Pooling 方法；

$$y_{i,j,k} = \left[\sum_{(m,n) \in R_{ij}} (a_{m,n,k})^p \right]^{\frac{1}{p}}$$

- 当 $p = 1$ 时, L_p Pooling 相当于“Average Pooling”;
- 当 $p = \infty$ 时, L_p Pooling 相当于“Max Pooling”;

[1]A. Hyvarinen, U. Koster, Complex cell pooling and the statistics of natural images, in: NCNS, 2007.

Mixed Pooling

Mix Pooling[1] 是一种 Max Pooling 与 Mean Pooling 混合的方式：

$$y_{i,j,k} = \lambda \max_{(m,n) \in R_{ij}} a_{m,n,k} + (1 - \lambda) \frac{1}{|R_{ij}|} \sum_{(m,n) \in R_{ij}} a_{m,n,k}$$

- Inspired by random Dropout and DropConnect.
- Experiments in show that mixed pooling can better address the overfitting problems and it performs better than max pooling and average pooling.

[1]D. Yu, H. Wang, P. Chen, Z. Wei, Mixed pooling for convolutional neural networks, in: Rough Sets and Knowledge Technology, 2014.

Stochastic Pooling

- Stochastic pooling[1] randomly picks the activations according to a multinomial distribution.
- Stochastic pooling first computes the probabilities p for each region R_j by normalizing the activations within the region:

$$p_i = \frac{a_i}{\sum_{k \in R_j} (a_k)}$$

- Compared with max pooling, stochastic pooling can avoid overfitting due to the stochastic component.

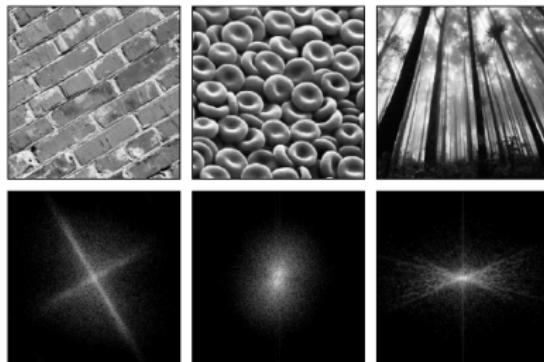
[1]M. D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, in: ICLR, 2013.

Spectral Pooling

通过离散傅立叶变换，将特征图像转换到频域再进行 Pooling[1]：
对于输入 $x \in C^{M \times N}$ ，其对应的傅立叶转换为：

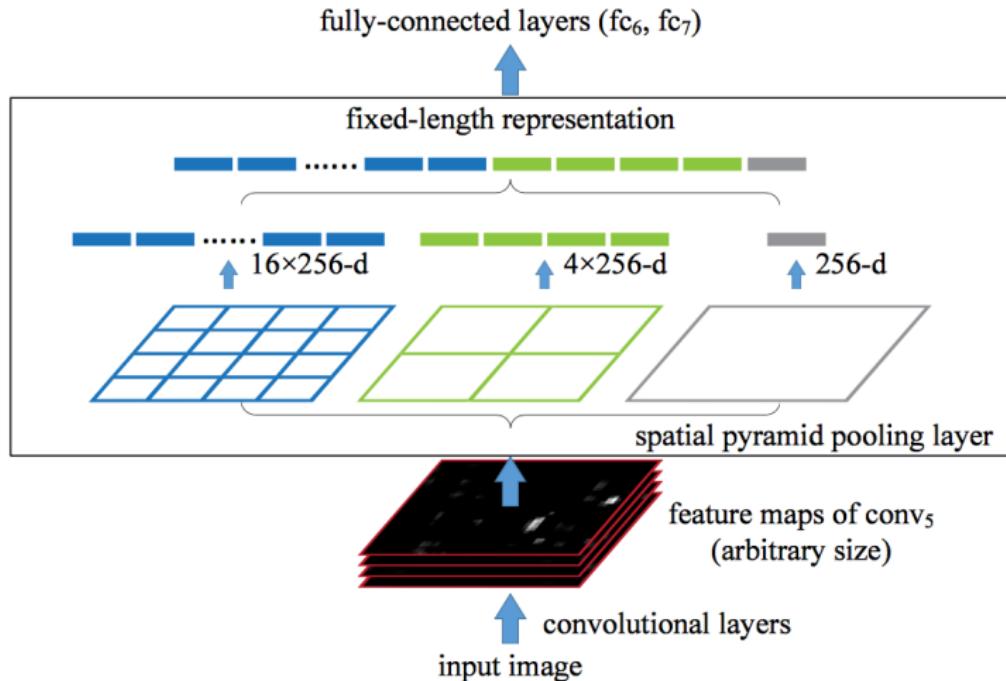
$$F(x)_{hw} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} e^{-1\pi i(\frac{mh}{M}) + \frac{nw}{N}}$$

$$\forall h \in 0, \dots, M-1, \forall w \in 0, \dots, N-1.$$



[1] O. Rippel, J. Snoek, R. P. Adams, Spectral representations for convolutional neural networks, in: NIPS, 2015.

Spatial Pyramid Pooling

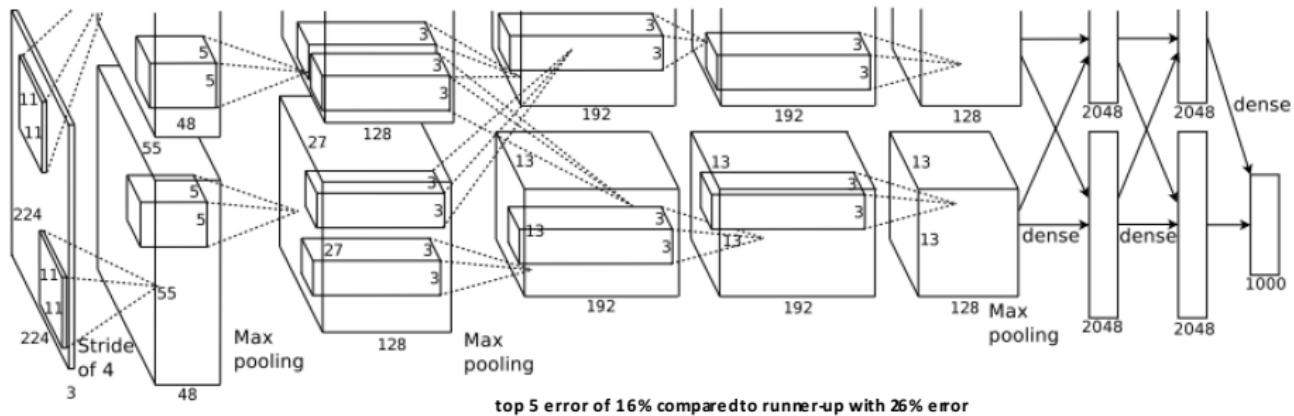


[1] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: ECCV, 2014.

2012 , Alex Net



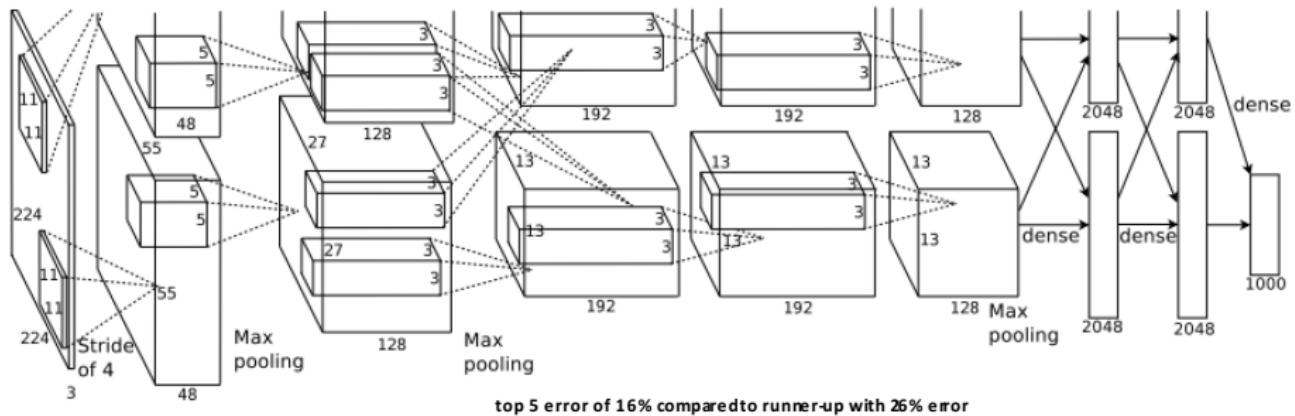
- ◆ 输入层： 224×224 大小的图片，3通道 (RGB)
- ◆ 第一层卷积： 5×5 大小，跨步4，卷积核96个，每个GPU上48个。
- ◆ 第一层Maxpooling：窗口大小 2×2



2012 , Alex Net



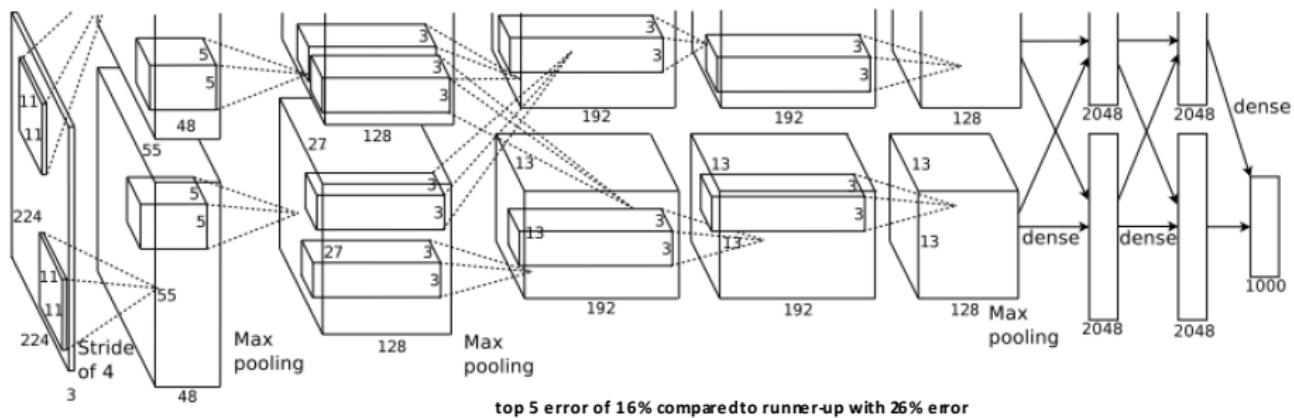
- ◆ 第二层卷积： 3×3 卷积核256个，每个GPU上128个。
- ◆ 第二层Maxpooling：窗口大小 2×2
- ◆ 第三层卷积：与上层全连接， 3×3 的卷积核 384个，每个GPU 192个。



2012 , Alex Net



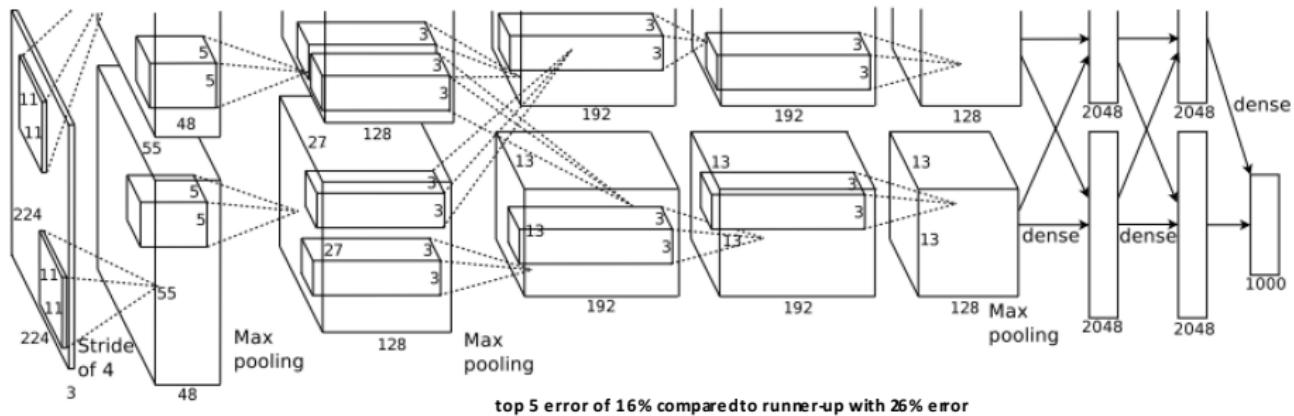
- ◆ 第四层卷积： 3×3 的卷积核384个，两个GPU各192个；
- ◆ 第五层卷积： 3×3 的卷积核256个，两个GPU上个128个；
- ◆ 第五层Maxpooling：窗口大小 2×2



2012 , Alex Net



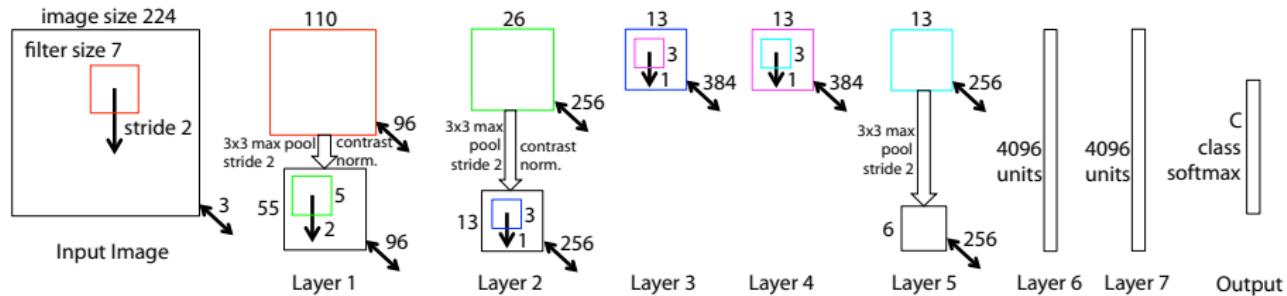
- ◆ 第一全连接：将第五层输出连接为一个一维向量作为输入，共4096维；
- ◆ 第二全连接：连接上层输入4096维；
- ◆ Softmax输出层 · 1000单元 每一维为图片属于该类别的概率



2013 ZF Net



■ Visualizing and Understanding Convolutional Networks



2014/12/9 Network In Network



Table 4: Test set error rates for MNIST of various methods.

Method	Test Error
2-Layer CNN + 2-Layer NN [11]	0.53%
Stochastic Pooling [11]	0.47%
NIN + Dropout	0.47%
Conv. maxout + Dropout [8]	0.45%

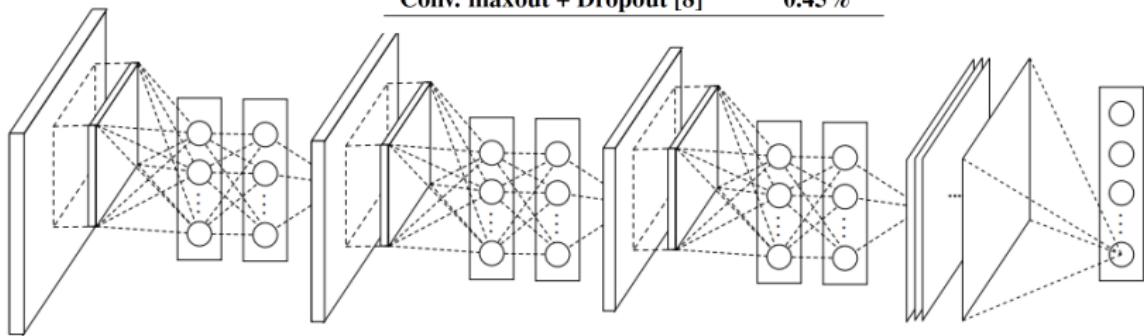
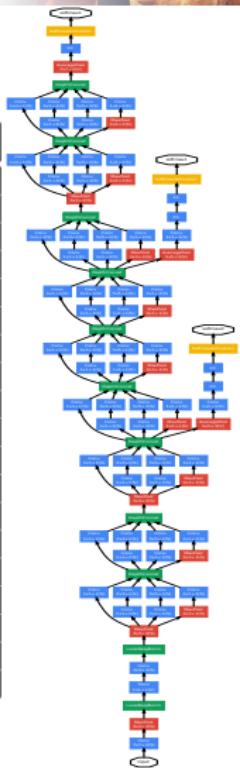


Figure 2: The overall structure of Network In Network. In this paper the NINs include the stacking of three mlpconv layers and one global average pooling layer.

2014 , GoogLeNet



2014 , VGGNet.



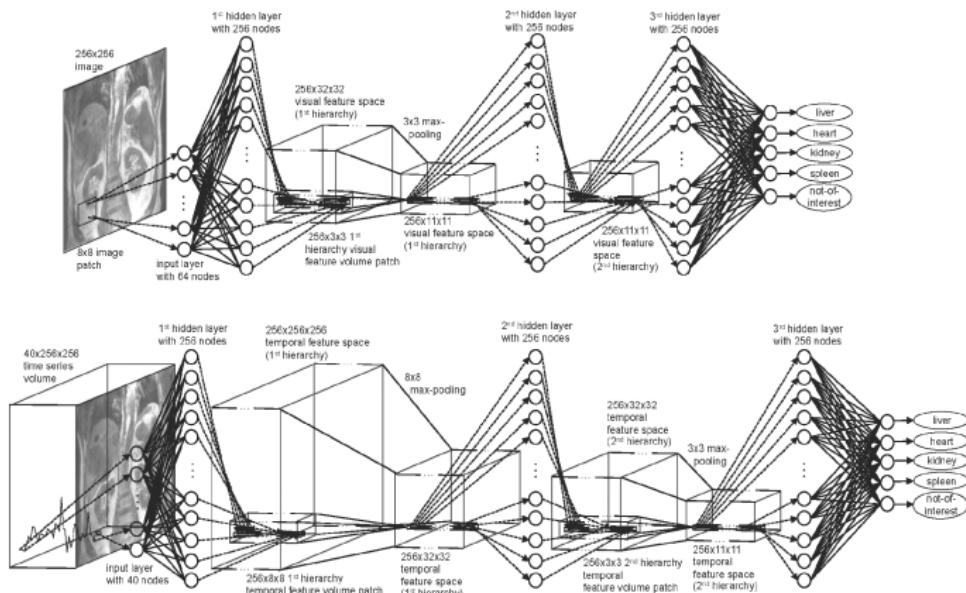
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

2015 , ResNet



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56			3×3 max pool, stride 2		
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	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

Convolution & AutoEncoder



Thanks.