4.1 树、森林、梯度树

- 决策树
- 最美的形式
- **高度的灵活性与表示力**
- **容易过拟合、容易过敏**

- 随机森林
- 鲁棒性很强的算法
- **良好的能力、难过拟合**
- **能力有时不够尤其回归**

- **简单融合+强个体能力**

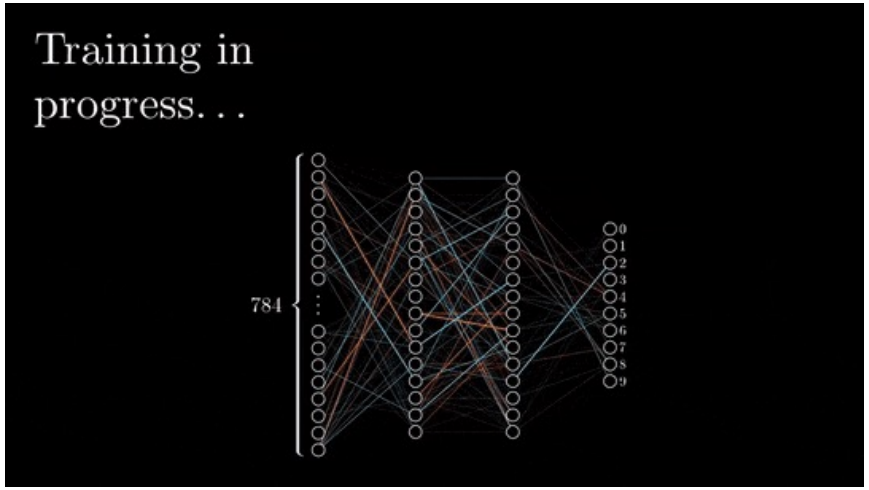
- Bagging 算法
- 将一个个小的**强**算法
- 通过**简单方式**进行融合
- 当发现一个灵活算法容易过拟合时
- **降低Variance**

- 梯度下降树
- 极其敏锐的算法
- **担当底牌的能力**
- **过拟合、难训练**

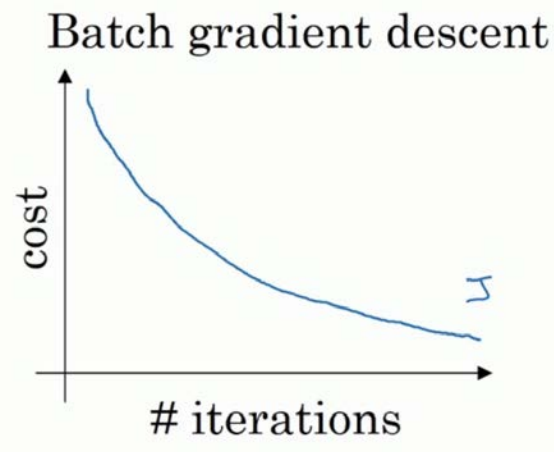
- **复杂融合+弱个体能力**

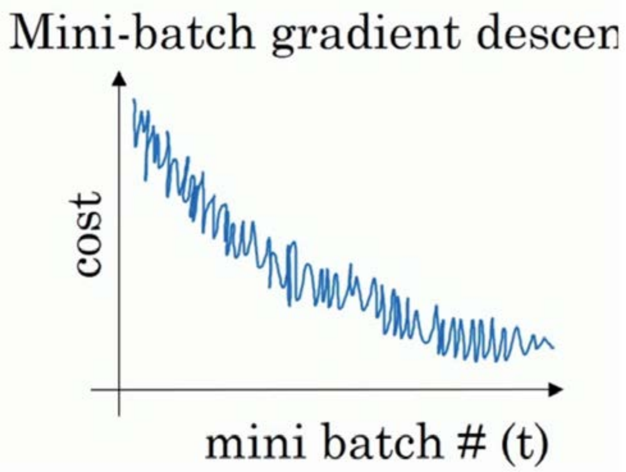
- Boosting 算法
- 将一个个小的**弱**算法
- 通过**复杂方式**进行融合
- 当发现一个问题难求解时
- **降低Bias**

1.2 反向传播（Back Propagation）算法



- Rumelhart, Hinton & Williams (1986)
- 算法流程
- 初始化权重$\boldsymbol{w}$（整张网络）
- 训练过程分为 t = 0, 1, 2, ⋯ T 期
  - 1.随机挑选：随机挑选一组数据$\boldsymbol{x}_{(\boldsymbol{n})}, y_{(\boldsymbol{n})}$
  - 2.前向传播：挑选数据$\boldsymbol{x}_{(\boldsymbol{n})}$ 作为输入，并向前传播直至算出网络总输出
  - 3.反向传播：将输出与真实值$y_{(\boldsymbol{n})}$ 进行比较，并根据链式法则将残差对某一个$\boldsymbol{w}_{ij}^{l}$求导
  - 4.梯度下降：按照减少残差的方向（残差求导的负方向）更新$\boldsymbol{w}_{ij}^{l}$
- 迭代多次后，将最终的$\boldsymbol{w}_{ij}^{l}$作为权重进行构建网络
- 多数情况下，1-3步会（并行）一起做多次**mini-batch**
- 优化$\boldsymbol{w}$的过程道阻且长，充满不确定性
- work but hard，做了许多许多年的机器学习"守门员"





2
0
2
3

经典神经网络

目　录

CONTENT

01

卷积
神经网络

02

循环
神经网络

03

生成模型

04

强化学习

Part 01

卷积神经网络

- 武无第二
- 如何识图?
- 卷积实现
- 网络迭代

ImageNet & Large Scale Visual Recognition Challenge

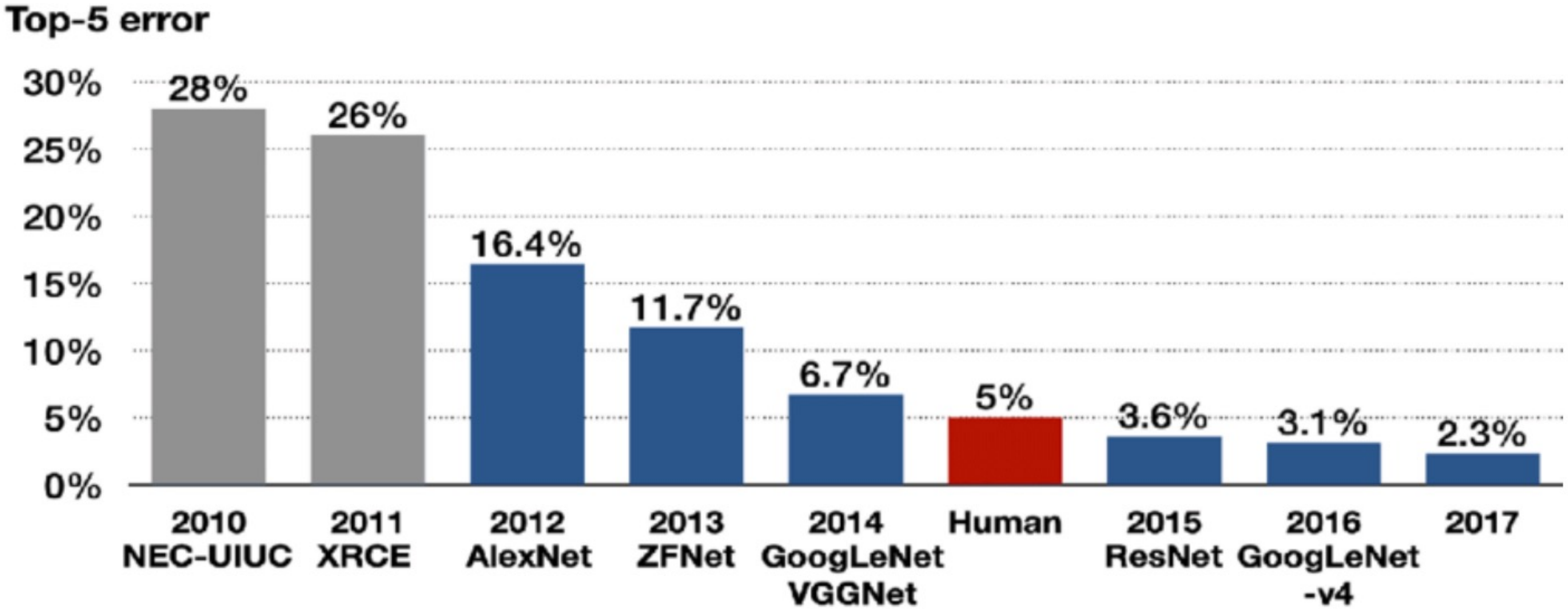
ImageNet 1500万张 2.2万类

ILSVRC

128万训练集，5万验证

10万测试集。1000类

五类图像相关任务



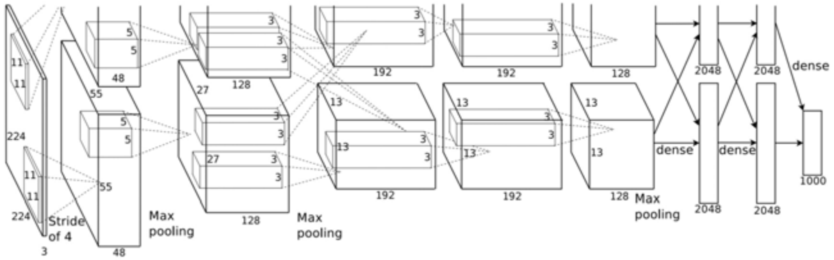


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

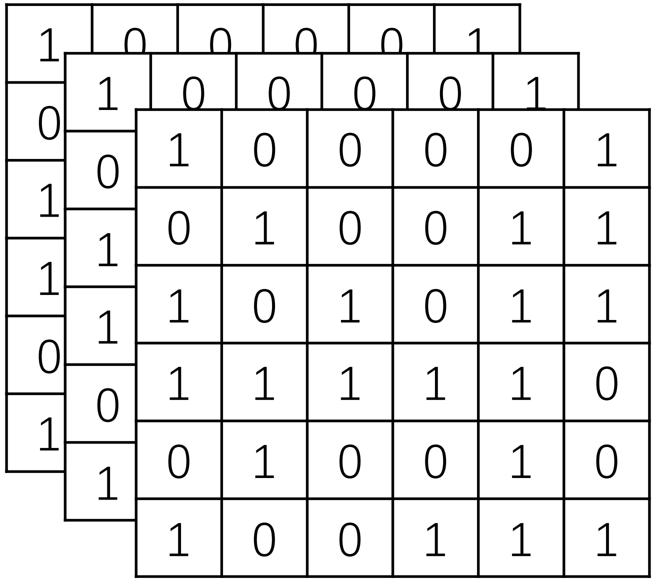
1.2 计算机如何理解一张图像

```
In [6]: lena.shape
Out[6]: (512, 512, 3)

In [14]: lena
Out[14]: array([[[125, 137, 226],
                 [125, 137, 226],
                 [133, 137, 223],
                 ...,
                 [122, 148, 230],
                 [110, 130, 221],
                 [ 90,  99, 200]],
```

512*512 像素

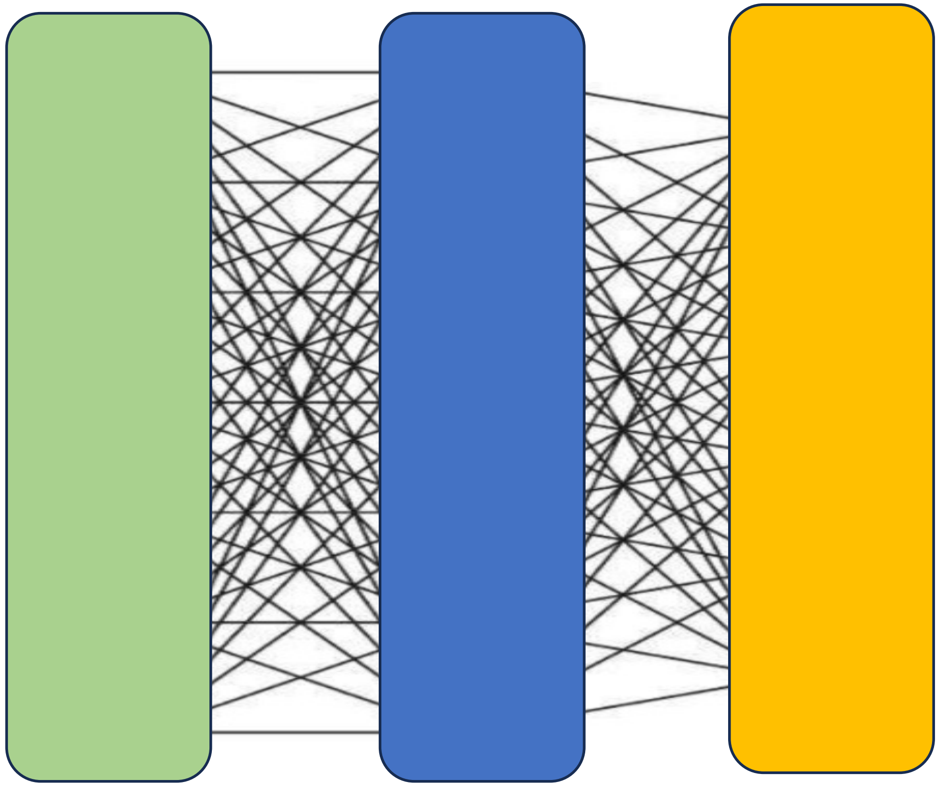
RGB三个通道



512*512*3

张量Tensor

1.2 我们来个Img2Vec



输入层　　隐藏层　　输出层

224*224*3
=150528　　1000　　1000

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

边数（要训练的权重）是多少?

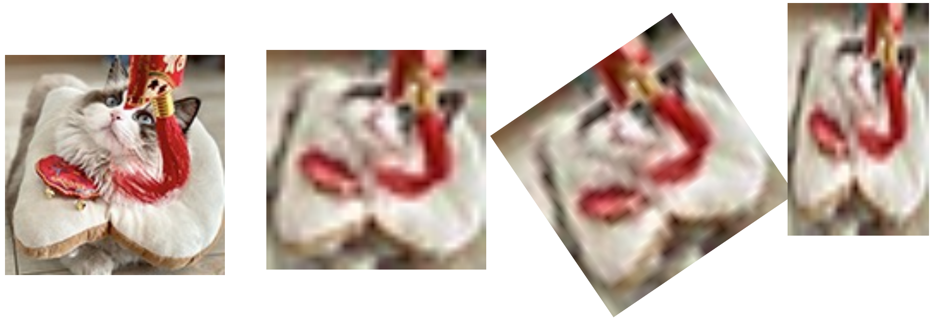
151528000

151M参数

1.2 动机与直觉

- 动机：神经网络或许太过强大（通用）
  - 每一步的全连接或许是过剩的
  - 断开一些连接性能不会受损甚至更好
- 直觉：人如何从一张图里抽取内容
- 我们关心的特征，往往只是图像的一小部分
  - 大量的冗余信息

- 存在的形式可能存在平移、放缩、旋转
- 图片的放缩（合理范围内）不会影响我们的判断



1.2 实现：一点点看图片

64 * 64 image

1.3

黄老爷的望远镜数学化!

滤镜尺寸Filter size = 3； 步长stride = 1

1	**0**	**0**
0	1	0
0	0	1

b = -2.5

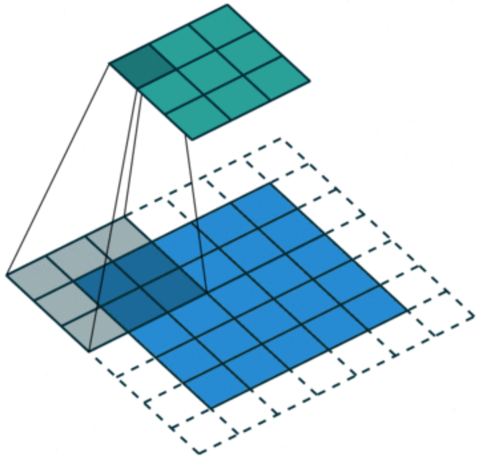
1	0	0	0
0	1	0	0

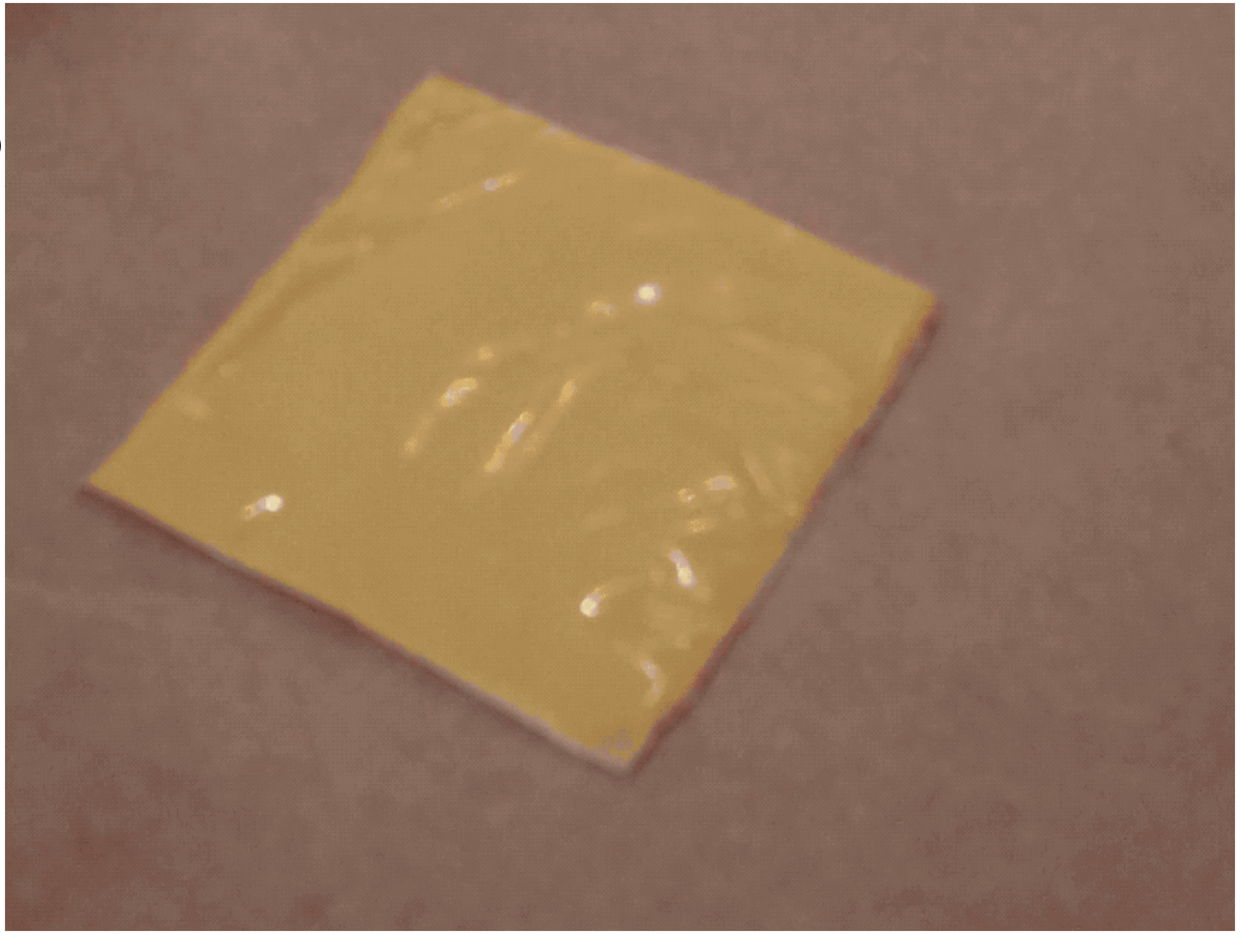
1	**1**	**1**
0	0	0
0	0	0

b

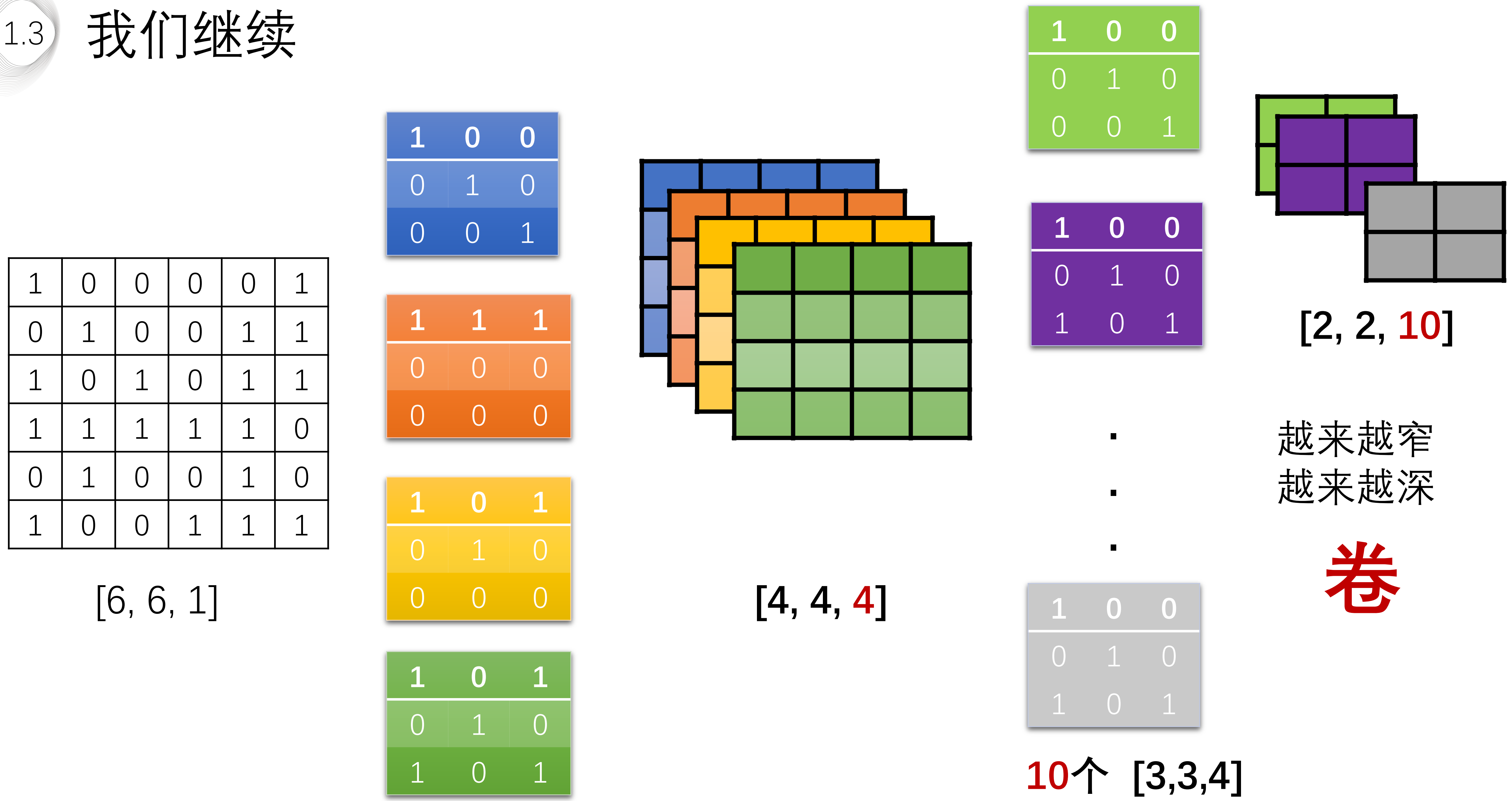
1	0	0	0	0	1
0	1	0	0	1	1
1	0	1	0	1	1
1	1	1	1	1	0
0	1	0	0	1	0
1	0	0	1	1	1

6 * 6 image



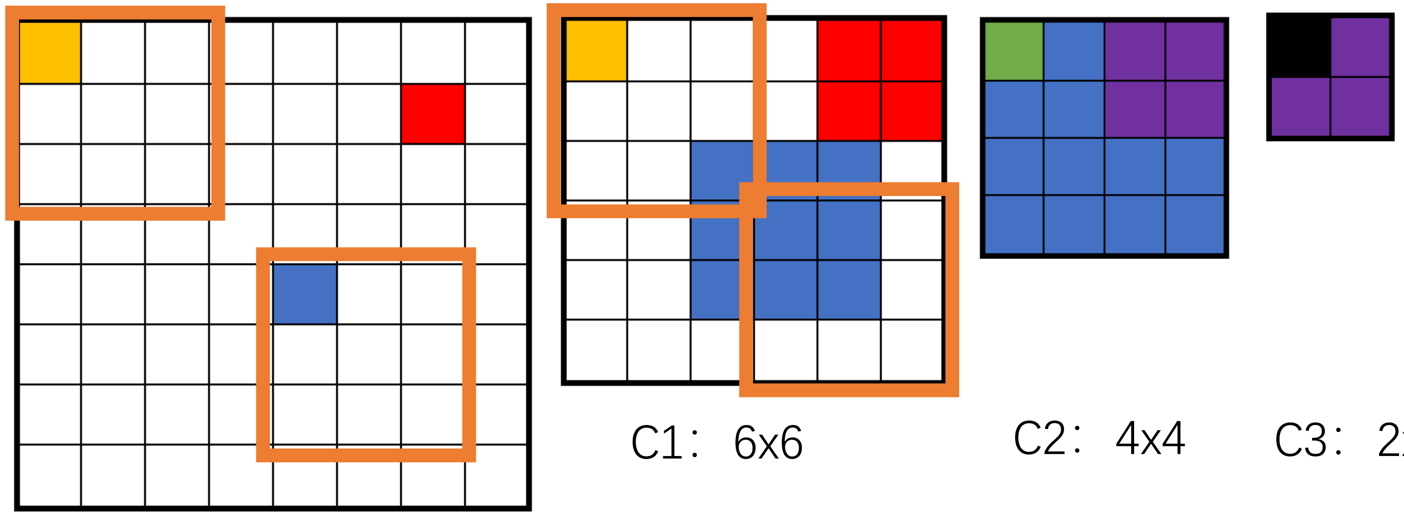


1.3 我们继续

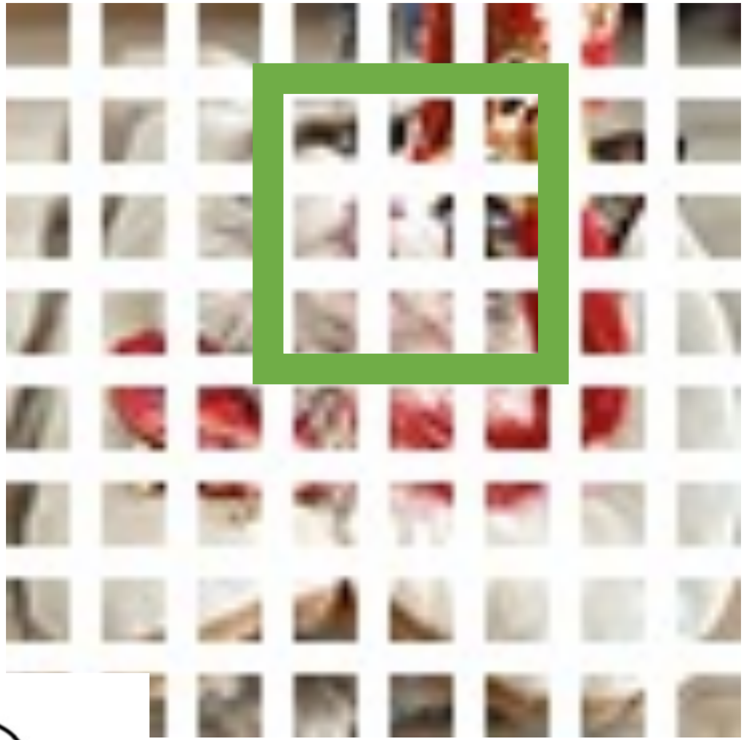


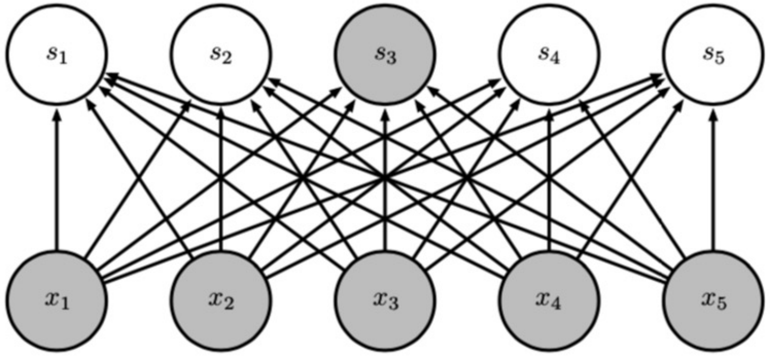
1 0 0 0 0 1
0 1 0 0 1 1
1 0 1 0 1 1
1 1 1 1 1 0
0 1 0 0 1 0
1 0 0 1 1 1
[6, 6, 1]
1 0 0
0 1 0
0 0 1
1 1 1
0 0 0
0 0 0
1 0 1
0 1 0
0 0 0
1 0 1
0 1 0
1 0 1
[4, 4, 4]
1 0 0
0 1 0
0 0 1
1 0 0
0 1 0
1 0 1
1 0 0
0 1 0
1 0 1
10个 [3,3,4]
[2, 2, 10]
越来越窄
越来越深
卷

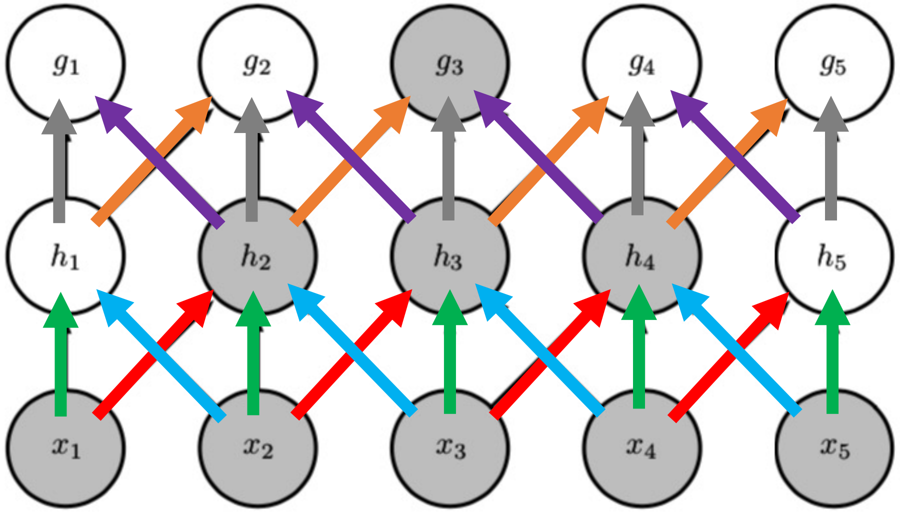
1.3 感受野（receptive field）&参数共享



原图：8x8　3x3的卷积核



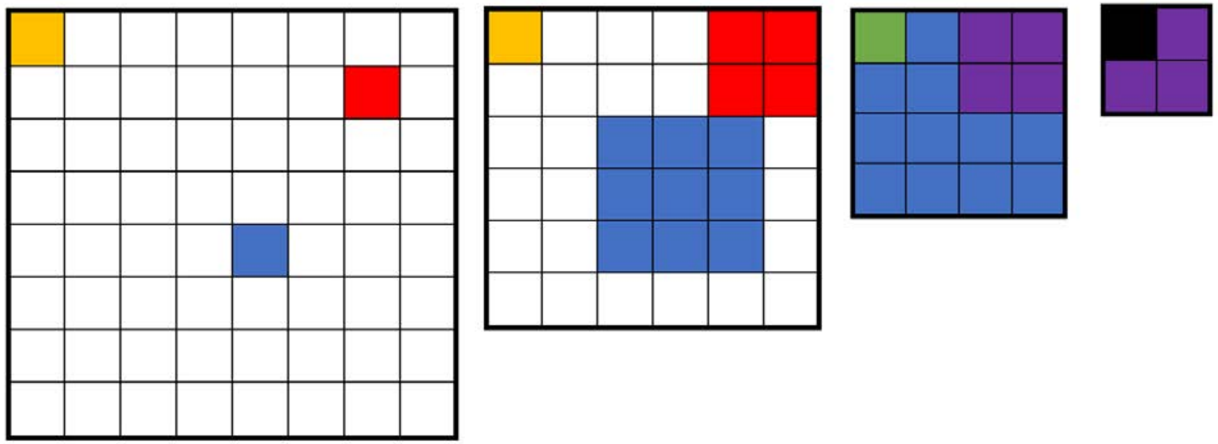




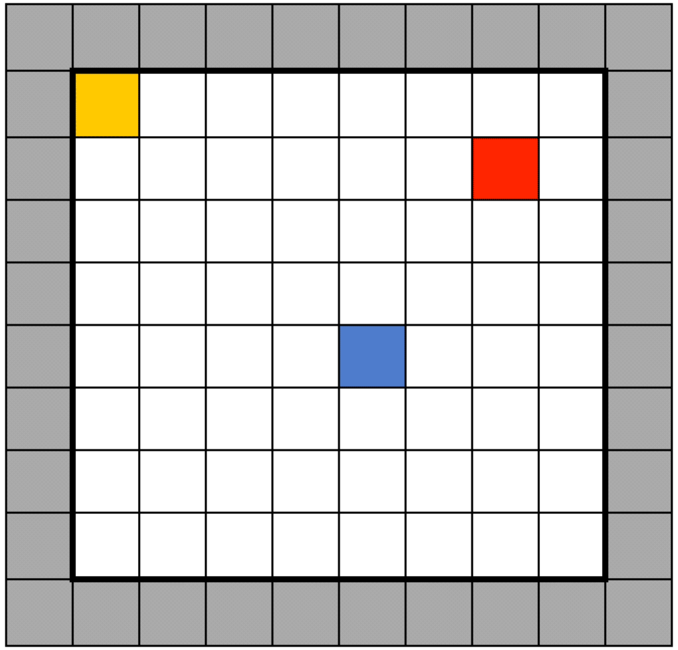
1.3 Padding 填充

1	2	3	3	3	3	2	1
2	4	6	6	6	6	4	2
3	6	9	9	9	9	6	3
3	6	9	9	9	9	6	3
3	6	9	9	9	9	6	3
3	6	9	9	9	9	6	3
2	4	6	6	6	6	4	2
1	2	3	3	3	3	2	1

1	2	3	3	3	3	2	1
2	4	6	6	6	6	4	2
3	6	9	9	9	9	6	3
3	6	9	9	9	9	6	3
3	6	9	9	9	9	6	3
3	6	9	9	9	9	6	3
2	4	6	6	6	6	4	2
1	2	3	3	3	3	2	1

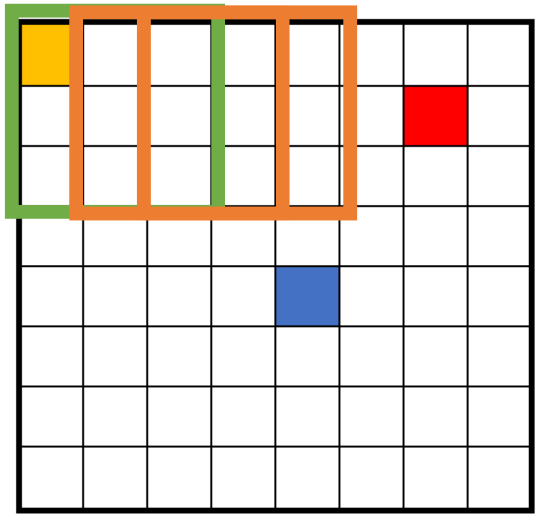


- 卷积使得图越卷越小，很难做很深的网络
- 边缘的权重很小——很难、很慢、很轻地识别边界
- Padding 填充技术：
  - 在周围一圈填补元素（一般为0）
  - 使得图的尺寸下降变慢、原有边界更多更快被检测到
- 假设卷积前边长为$\boldsymbol{m}$,卷积核边长为$\boldsymbol{k}$,padding为单边$\boldsymbol{p}$
  - 卷积后边长为$m+2p-(k-1)$
  - m,k,p可以纵横不一样，即m_1,m_2,k_1,k_2,p_1,p_2



原图

1.3 简化：步长stride 与 池化 pooling



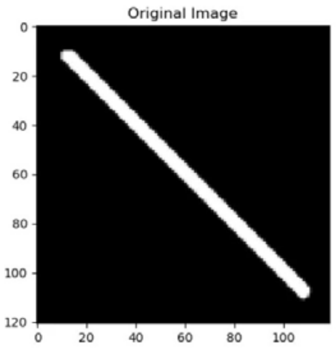
- 有的时候我们担心尺寸下降太慢
  - 真实世界图片尺寸更大
  - 过深的网络带来很多问题：参数量、训练
- Stride 步长：卷积核的移动步长$\boldsymbol{s}$
  - 卷积前边长为$\boldsymbol{m}$, 卷积核边长为$\boldsymbol{k}$, padding为单边$\boldsymbol{p}$
  - 则下一层边长为$\left\lfloor\frac{m+2p-k}{s}\right\rfloor+1$
  - 同样，步长两个方向不一样s_1, s_2
- Pooling 池化：把若干个元素**简单**计算为一个
  - Max Pooling
  - Average Pooling

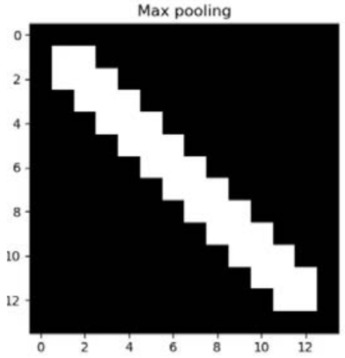
1	2	5	7
1	3	3	5
3	2	4	2
8	7	6	5

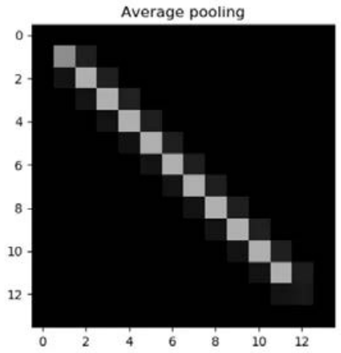
2	2	5	7
1	3	3	5
3	2	4	5
8	7	6	5

3	7
8	6

2	5
5	5

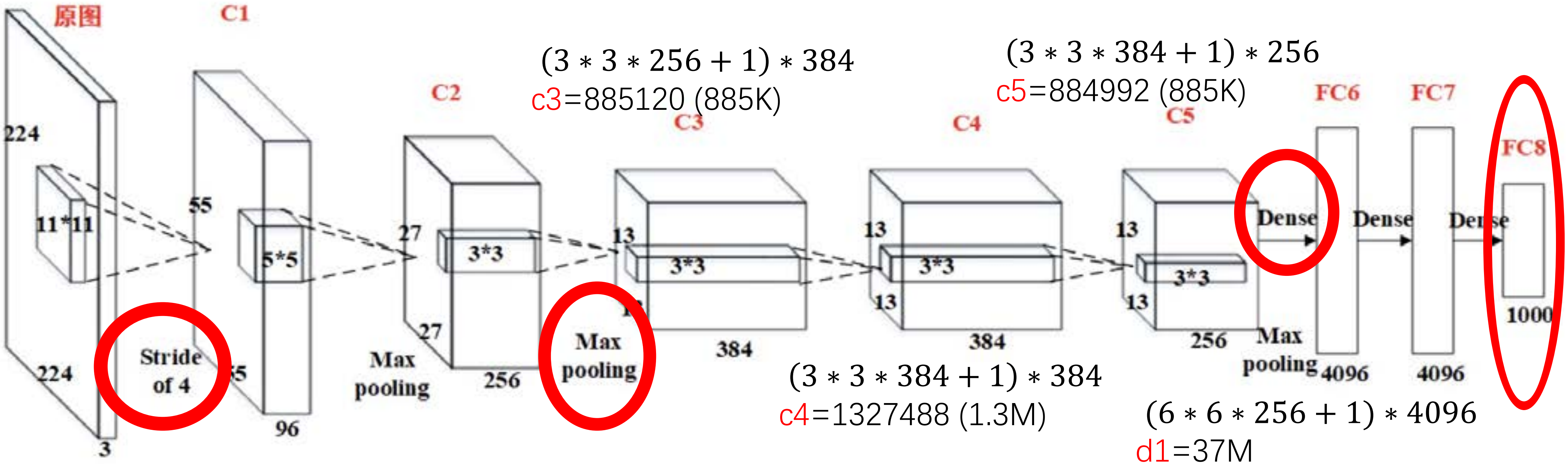






1.4 卷积神经网络

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).



$(11*11*3+1)*96$
c1=34848 (34K)

$(5*5*96+1)*256$
c2=614656 (614K)

$(3*3*256+1)*384$
c3=885120 (885K)

$(3*3*384+1)*384$
c4=1327488 (1.3M)

$(3*3*384+1)*256$
c5=884992 (885K)

$(6*6*256+1)*4096$
d1=37M

$(4096+1)*4096$
d2=16.7M

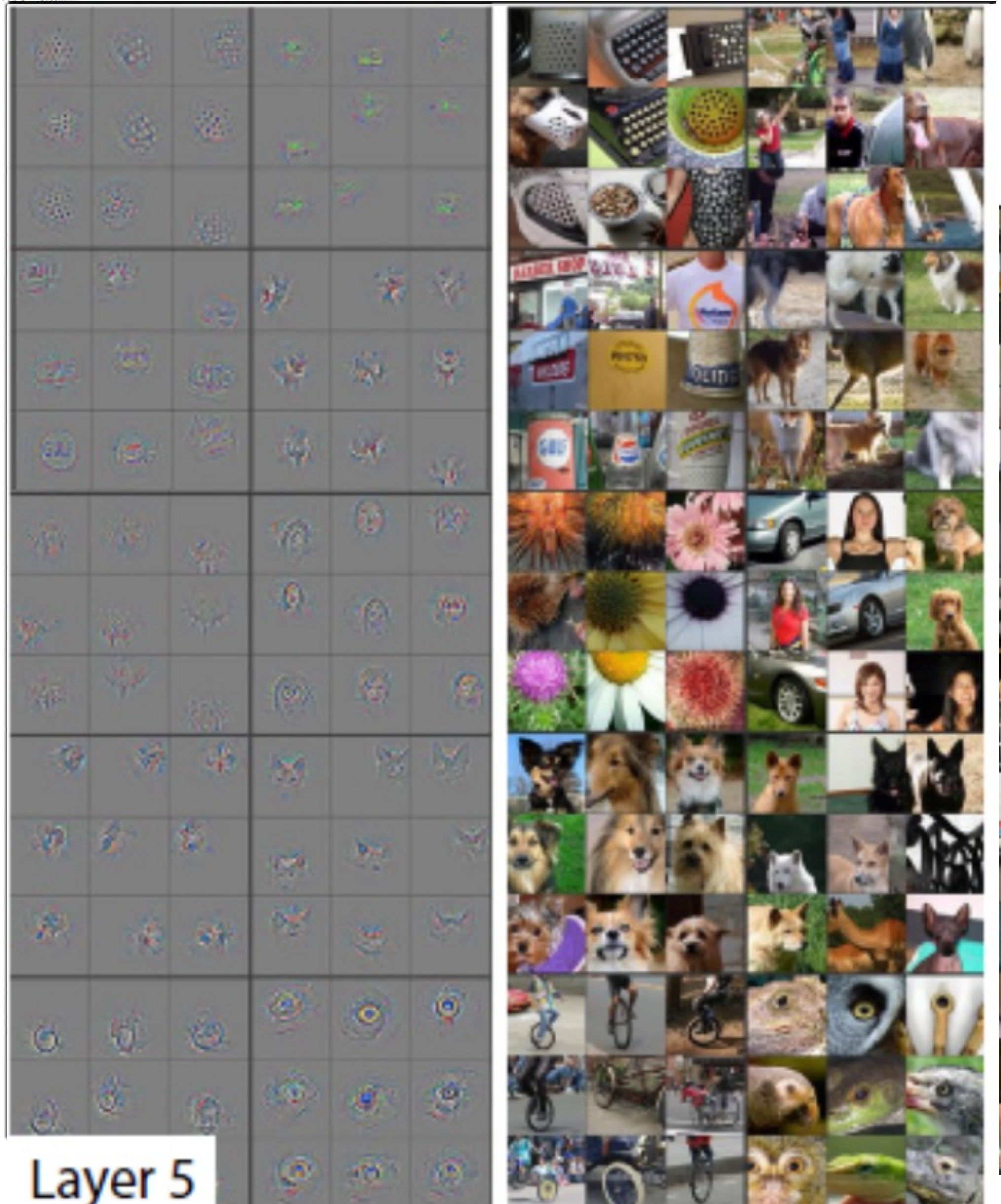
$(4096+1)*1000$
d3=4.1M

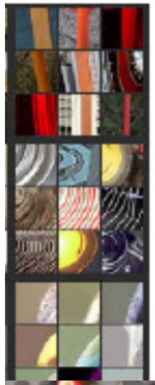
60M参数量

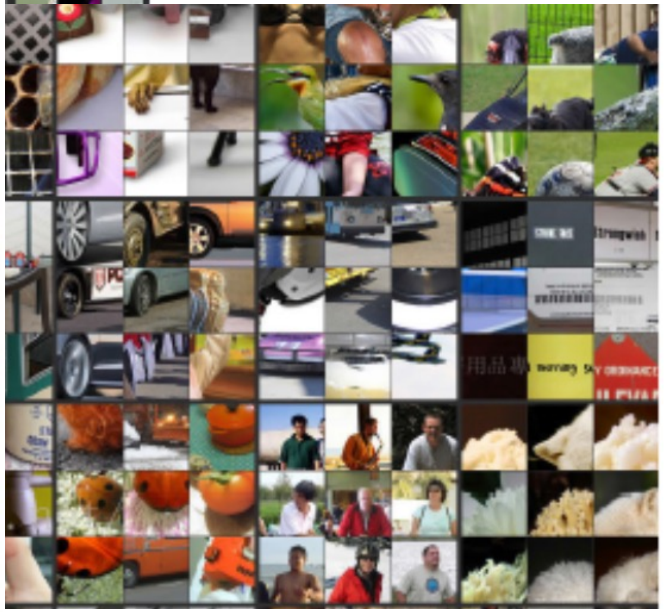
94%是最后几个全连接层

1.4

卷积神经网络：why work?





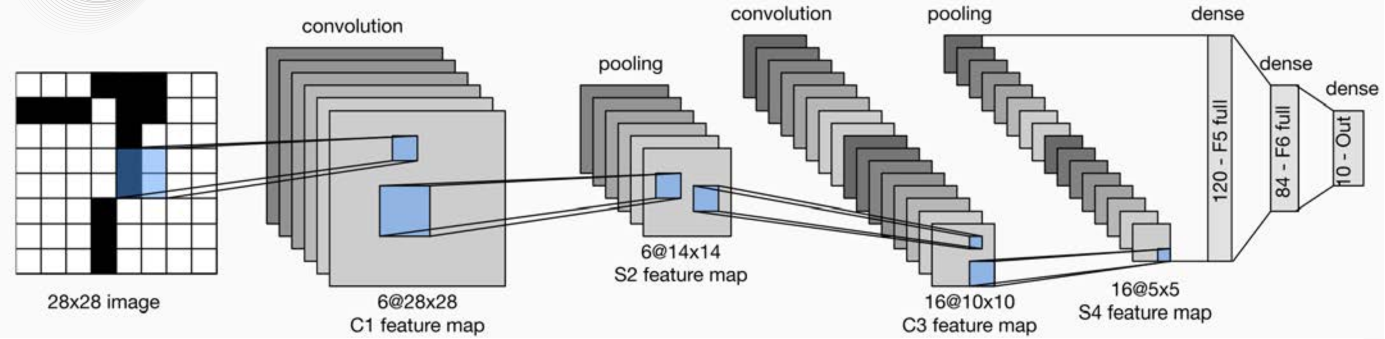


Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer International Publishing, 2014.

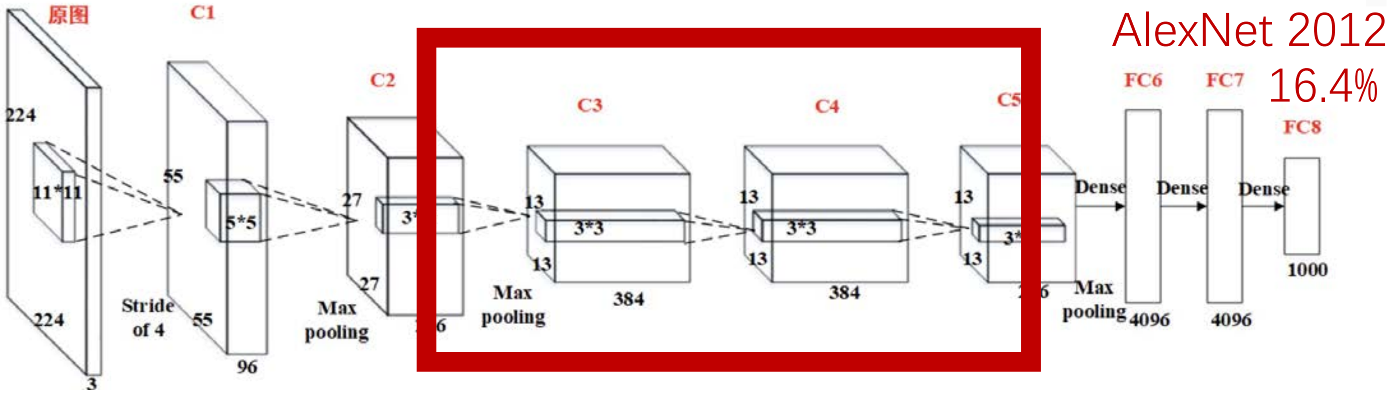
- Layer 1、2 颜色、边缘、无意义
- Layer3 开始学习纹理
- Layer4 标志性特征
- Layer5 完整的、全面的

1.4 从层到块

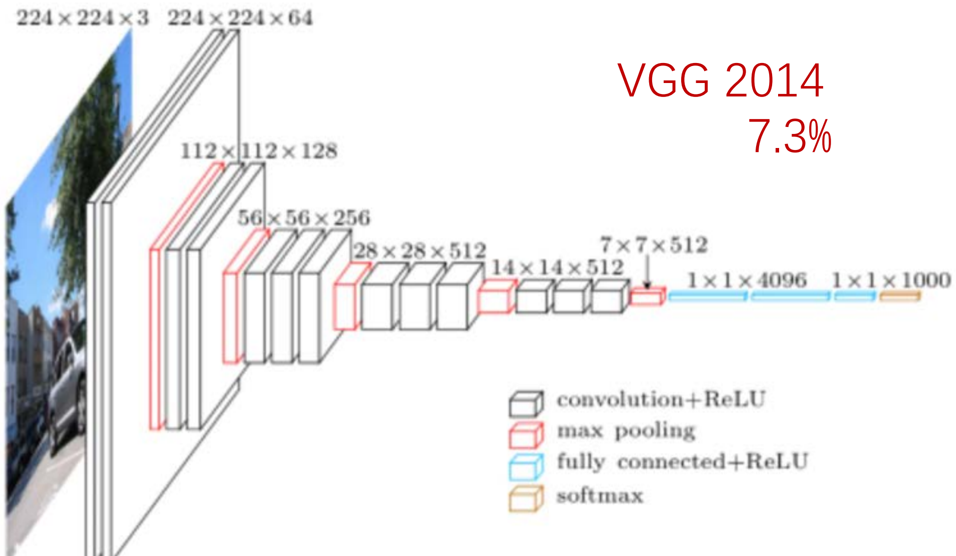
LeNet 1998



AlexNet 2012
16.4%

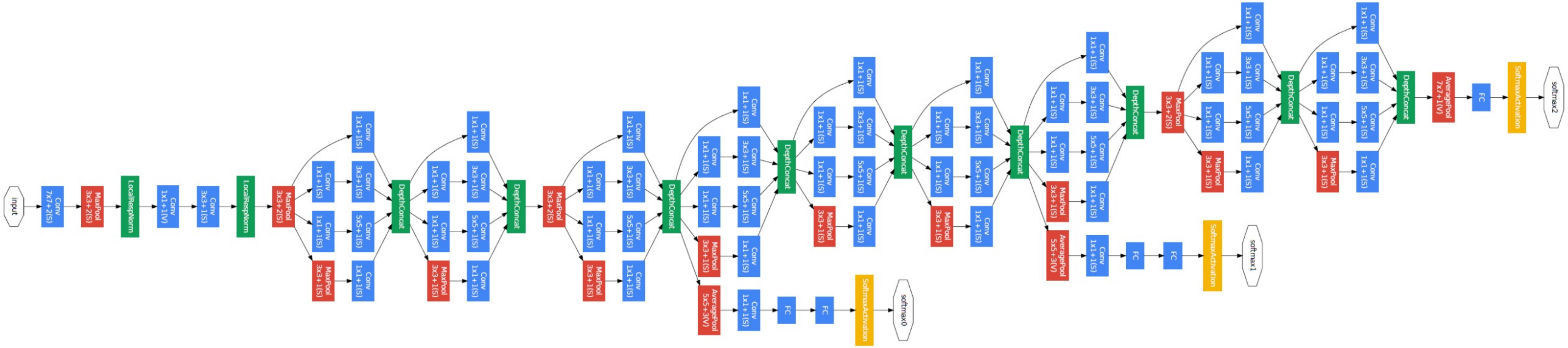


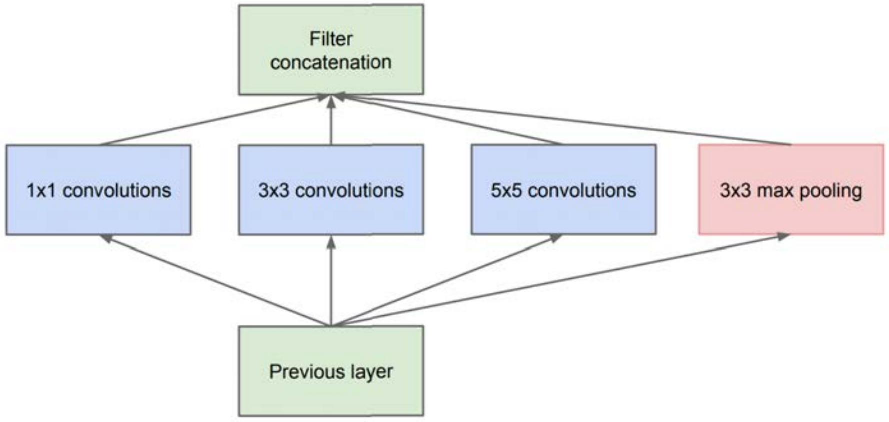
VGG 2014
7.3%



- 越深越好，小而深优于大而浅
- K=3，p=1，大小不变
- （只）需要改变输入输出维度
- 堆许多个k=3,p=1?
- 形成一系列的“**块**”

1.4 GoogLeNet（2014, 6.7%） 22层、100+网络



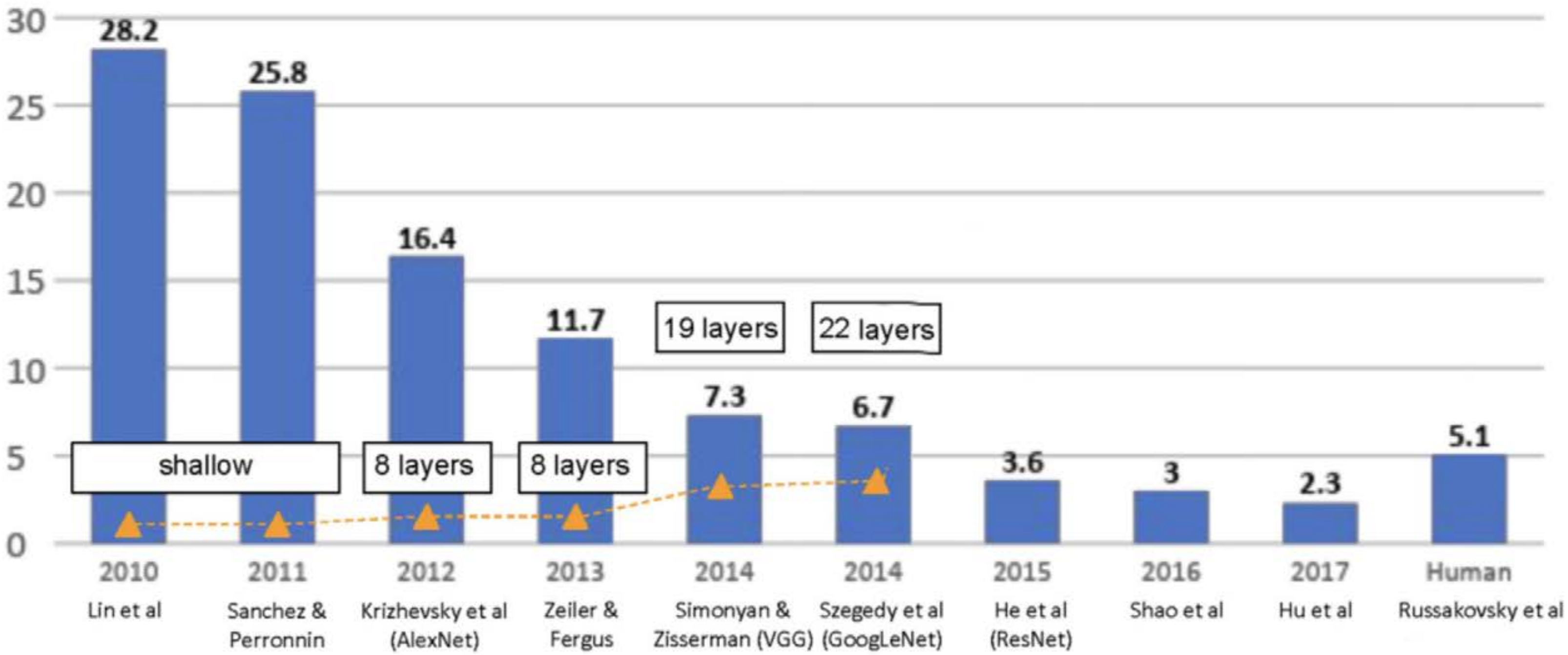


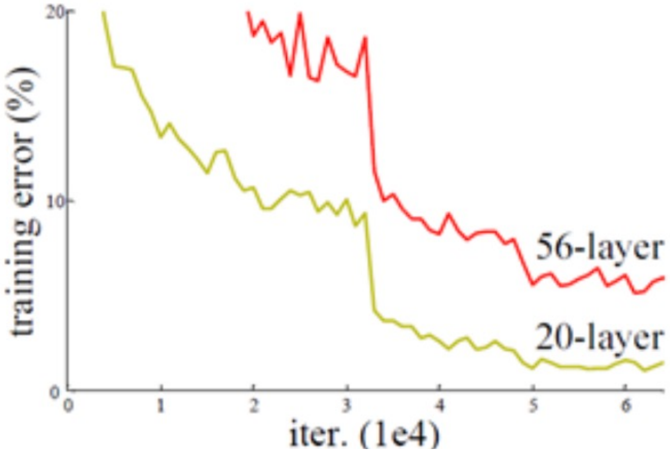
(a) Inception module, naïve version

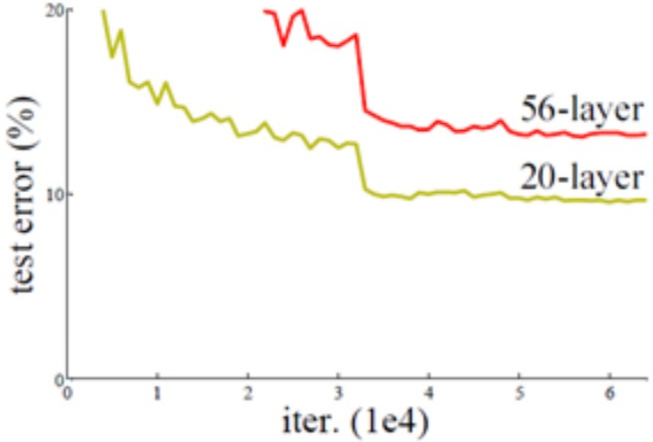
1.4 卷的尽头是什么？

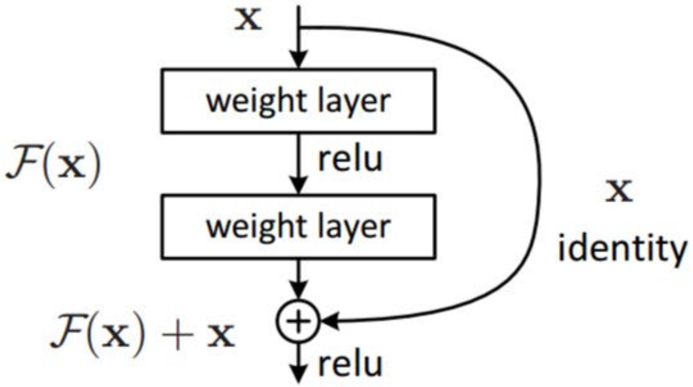
何恺明

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners







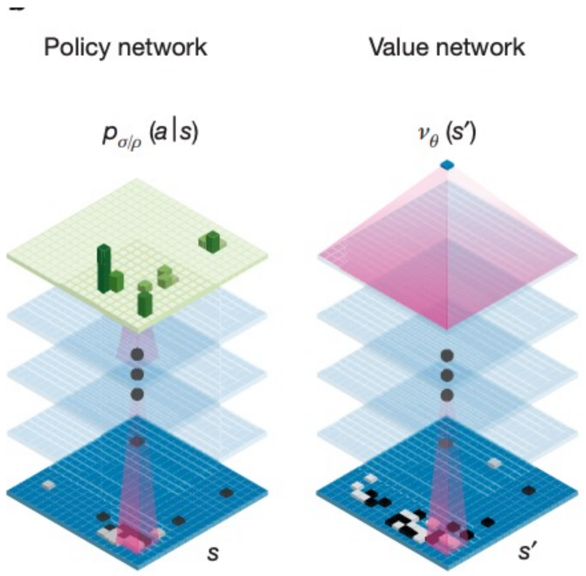


资本主义可以搞股市，社会主义也可以搞嘛……要坚决地试，搞不好可以关掉嘛！

1.5

卷积神经网络：扩展&更改

- 小特征决定
- 放缩旋转存在
- 放大缩小不改变性质
- AlphaGo：没有Pooling
- TextCNN：横条卷积核



Silver et al. 2016

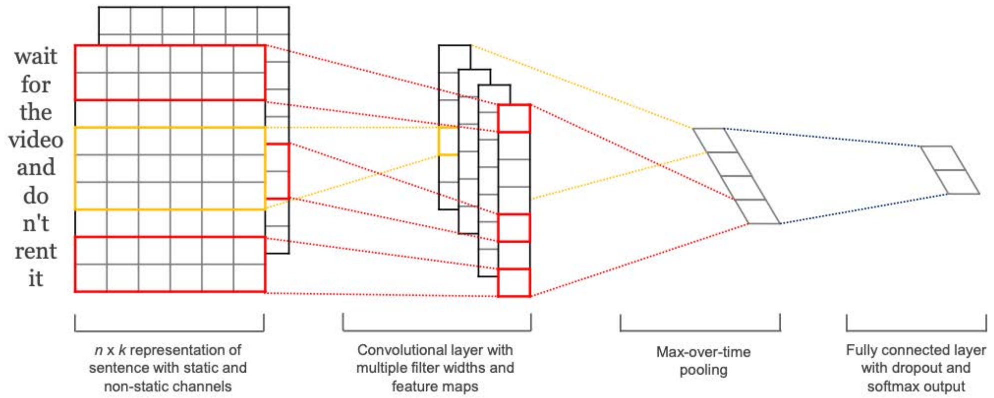


Figure 1: Model architecture with two channels for an example sentence.

Kim 2014

1.5 CNN 应用：基于文本构建情绪指数

Li, J., Chen, Y., Shen, Y., Wang, J., & Huang, Z. (2019). Measuring China's stock market sentiment. *Available at SSRN 3377684*.

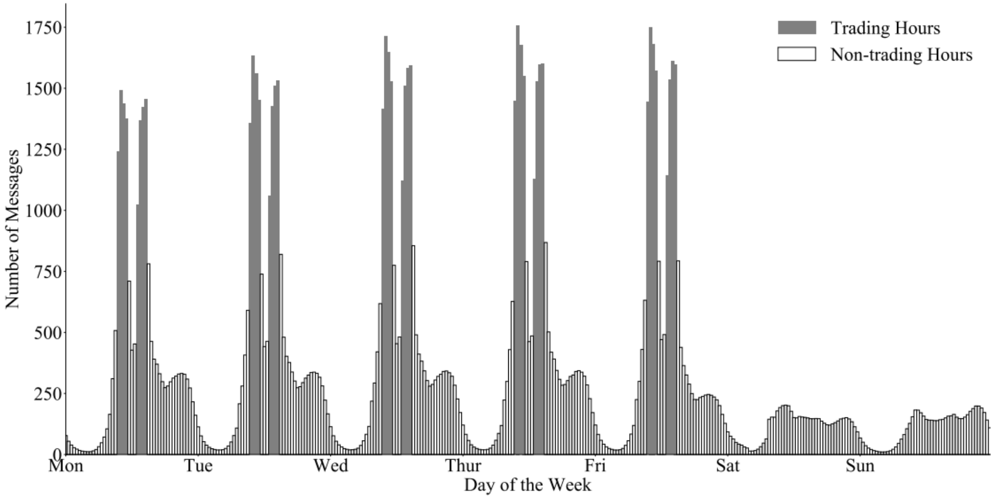


FIGURE I
Message Posting Activity within a Week

Notes: This figure plots the average number of messages of each day of the week at half-hour frequency. The sample period is from July 1, 2008 to February 14, 2018.

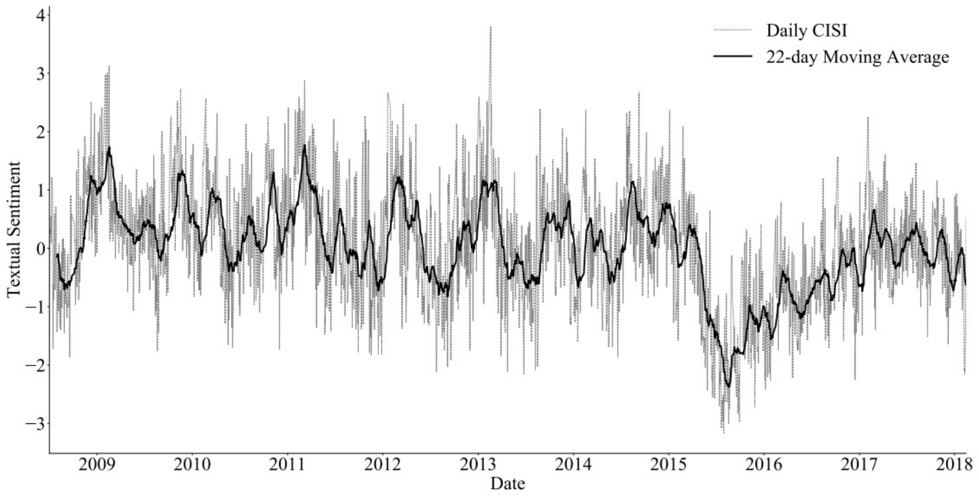


FIGURE III
Time Series of the Daily CISI

Notes: This figure plots the time series of the daily value-weighted CISI textual sentiment index. The daily sentiment index is formed using messages posted between 3:00 pm on trading day $t-1$ and 3:00 pm on trading day t. The index is normalized to have zero mean and unit standard deviation. The solid line is the 22-day moving average of the daily series.