

# 深度学习介绍

2023 年

# 目录

- 深度学习概述
- 深度学习常见场景
- 深度学习常用算法介绍
- 深度学习常用框架介绍
- 深度学习未来和展望

## 1.1 深度学习概述

- 深度学习(Deep Learning, DL)由Hinton等人于2006年提出，是机器学习(MachineLearning, ML)的一个新领域。
- 深度学习起源于人工神经网络，它的定义：通过组合低层特征形成更加抽象的高层特征或类别，从而从大量的输入数据中学习有效特征表示，并把这些特征用于分类、回归和信息检索的一种技术。
- 深度学习被引入机器学习使其更接近于最初的目标——人工智能(AI, Artificial Intelligence)。深度学习是学习样本数据的内在规律和表示层次，这些学习过程中获得的信息对诸如文字、图像和声音等数据的解释有很大的帮助。它的最终目标是让机器能够像人一样具有分析学习能力，能够识别文字、图像和声音等数据。

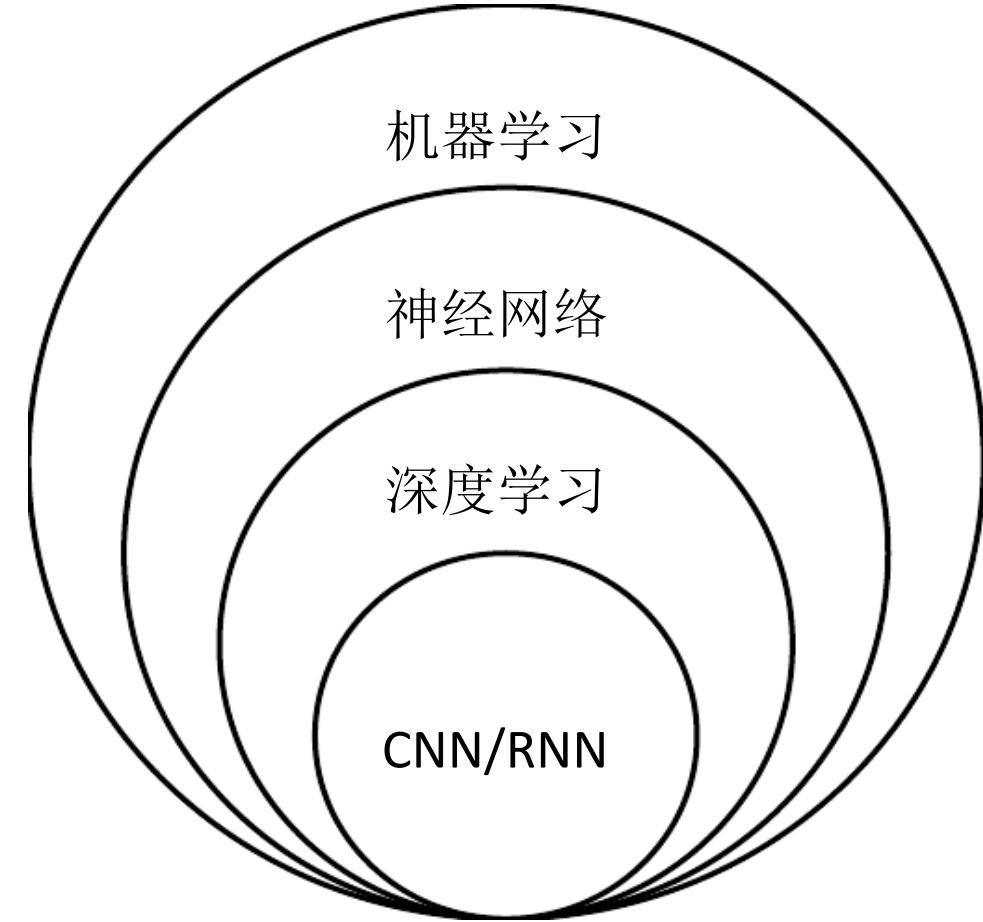
## 1.2 与机器学习关系



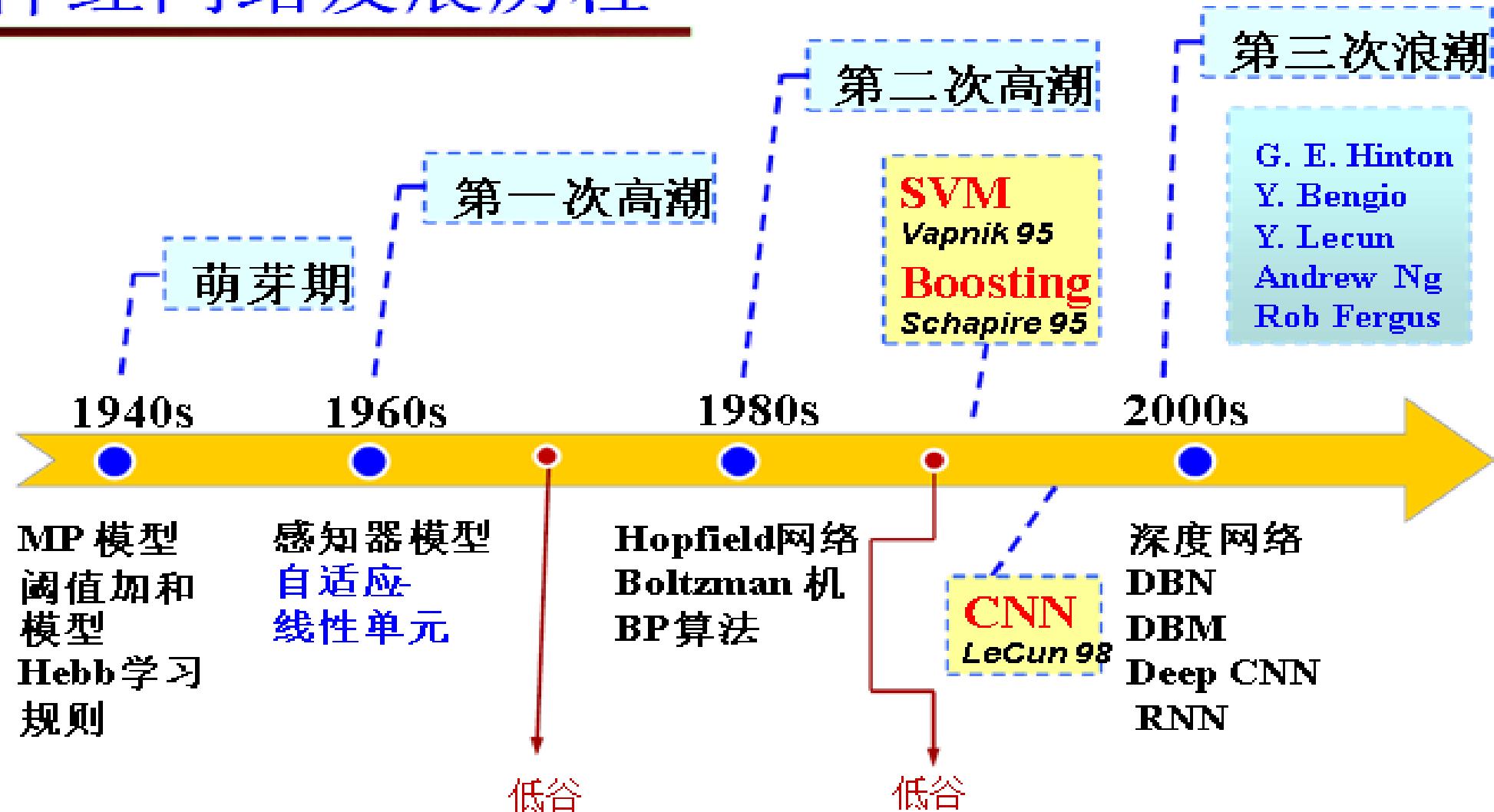
如上图，人工智能是最早出现的，也是最大、最外侧的同心圆；其次是机器学习，稍晚一点；最内侧，是深度学习，当今人工智能大爆炸的核心驱动。

## 1.2 与CNN关系

机器学习，神经网络，  
深度学习之间的关系



# 人工神经网络发展历程



# 人工神经网络发展历程

**Deep Learning, Science 2006 (vol. 313, pp. 504-507)**

## **Reducing the Dimensionality of Data with Neural Networks**

G. E. Hinton\* and R. R. Salakhutdinov

### **两个重要的信息：**

1. 具有多个隐层的人工神经网络具有**优异的特征学习能力**，学习得到的特征对数据有更本质的刻画，从而有利于可视化或分类；
2. 深度神经网络在训练上的难度，可以通过“逐层初始化”  
**(Layer-wise Pre-training)** 来有效克服。

**Neural networks are coming back!**

# 人工神经网络发展历程

- 在语音识别取得重大突破

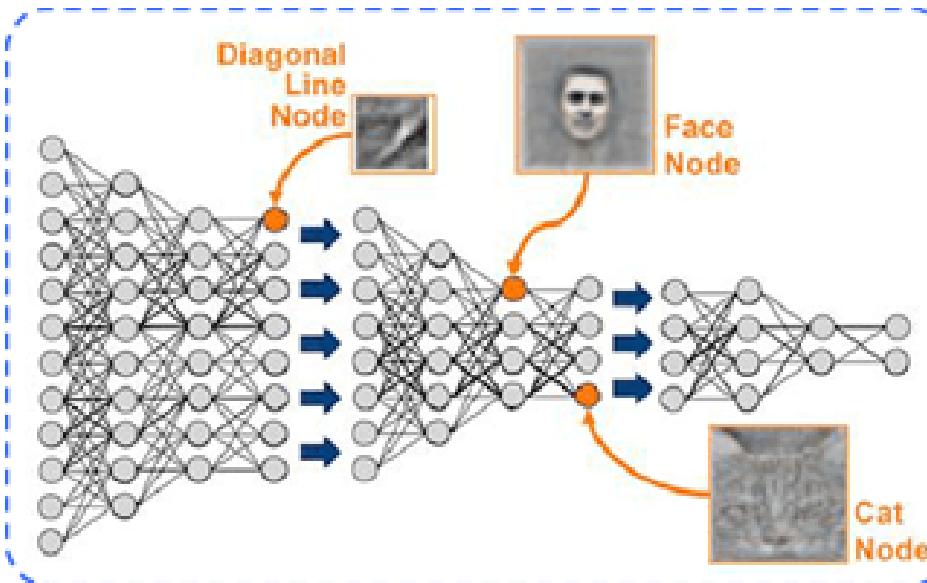


2011年以来，错误率降低20–30%!

2012年11月，微软在天津的一次活动上公开演示了一个全自动的**同声传译系统**，讲演者用英文演讲，后台的计算机一气呵成自动完成语音识别、英中机器翻译，以及中文语音合成，效果非常流畅。

# 人工神经网络发展历程

- Google Brain项目（纽约时报2012年6月报道）



吴恩达

2011年开始，Google Brain 项目采用16000个CPU Core 的并行计算平台训练“深层神经网络”，在图像识别等领域获取巨大成功！

# 人工神经网络发展历程



# 1.3 基本概念

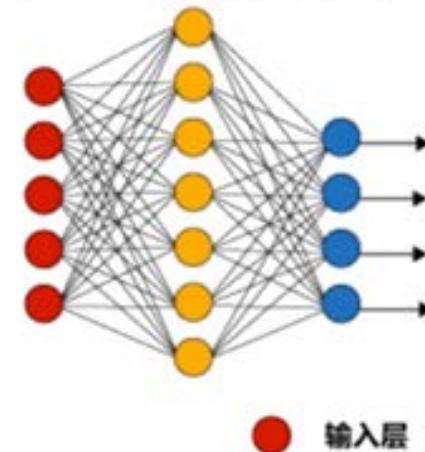
- 深度学习 (**Deep Learning, DL**)：通过组合低层特征形成更加抽象的高层特征或类别，从而从大量的输入数据中学习有效特征表示，并把这些特征用于分类、回归和信息检索的一种技术。
- 人工神经网络：是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。
- 深度神经网络：包含多个隐含层的神经网络。
- 模型：可以理解成算法和数据的合集。
- 模型训练：利用模型的算法，使用深度神经网络进行权值的训练，最终得出一个最优解。
- 模型预测：使用训练完成的模型进行预测，得出分类识别结果。

### 1.3 基本概念

- 人工神经网络：是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。
- 深度神经网络：包含多个隐含层的神经网络。

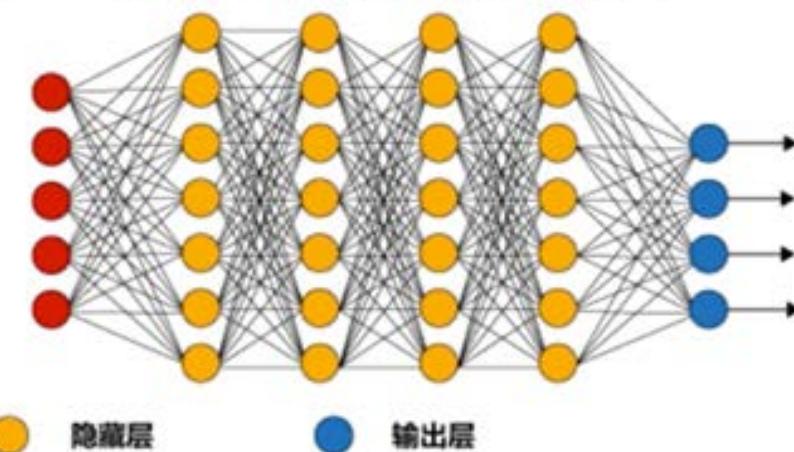
## 神经网络和深度学习

只有一个隐藏层的简单神经网络



● 输入层

大于两个隐藏层的深度神经网络



● 隐藏层

● 输出层

# 目录

- 深度学习概述
- 深度学习常见场景
- 深度学习常用算法介绍
- 深度学习常用框架介绍
- 深度学习未来和展望

## 2 深度学习场景

当前深度学习使用的场景主要在无人驾驶，人脸识别，拍照购，智能客服，文字识别，语音识别，目标检测，图片分类等方面。

人脸识别

文字识别

语音识别

无人驾驶

深度学习

拍照购

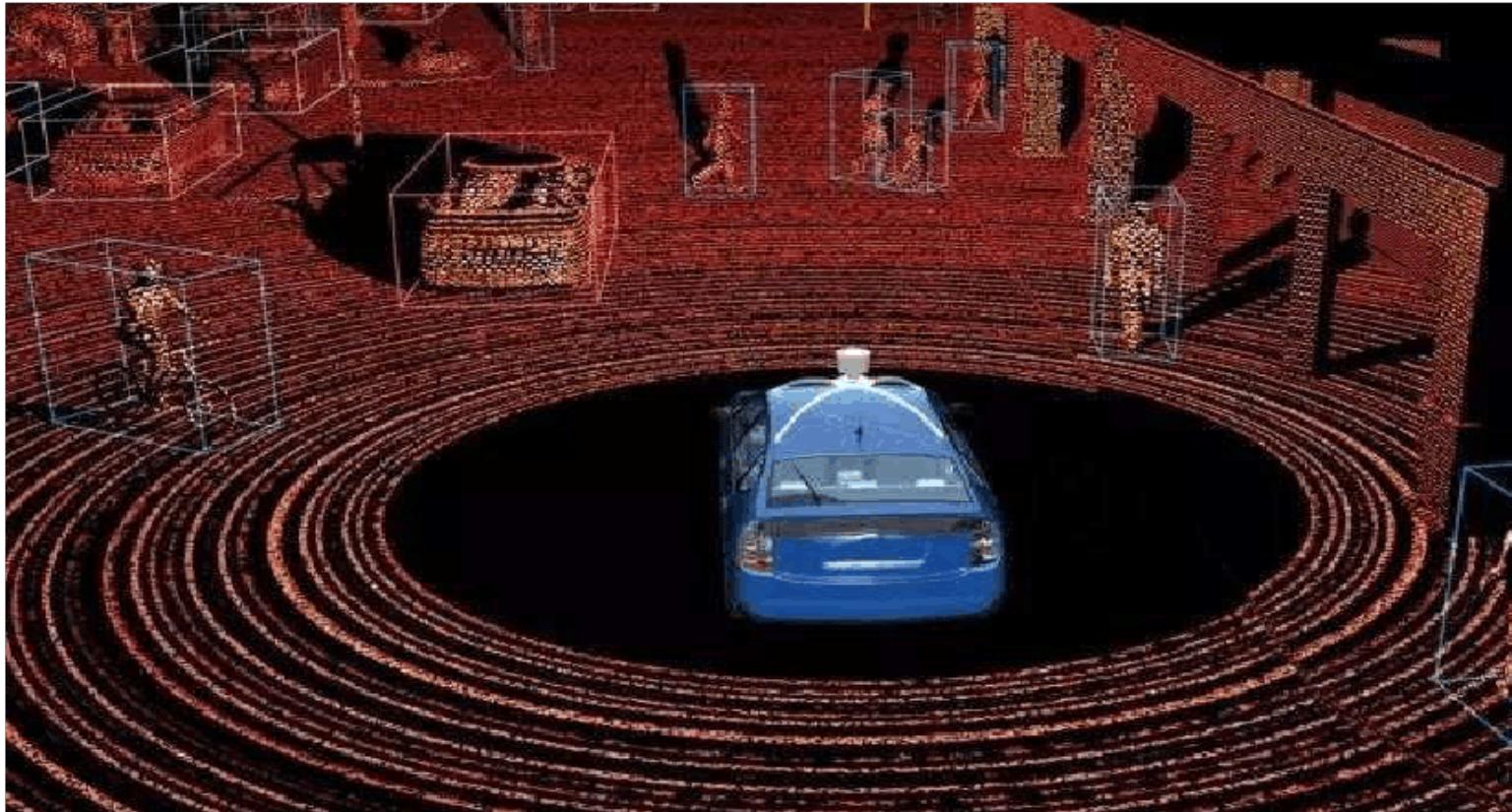
图像分类

智能客服

目标检测

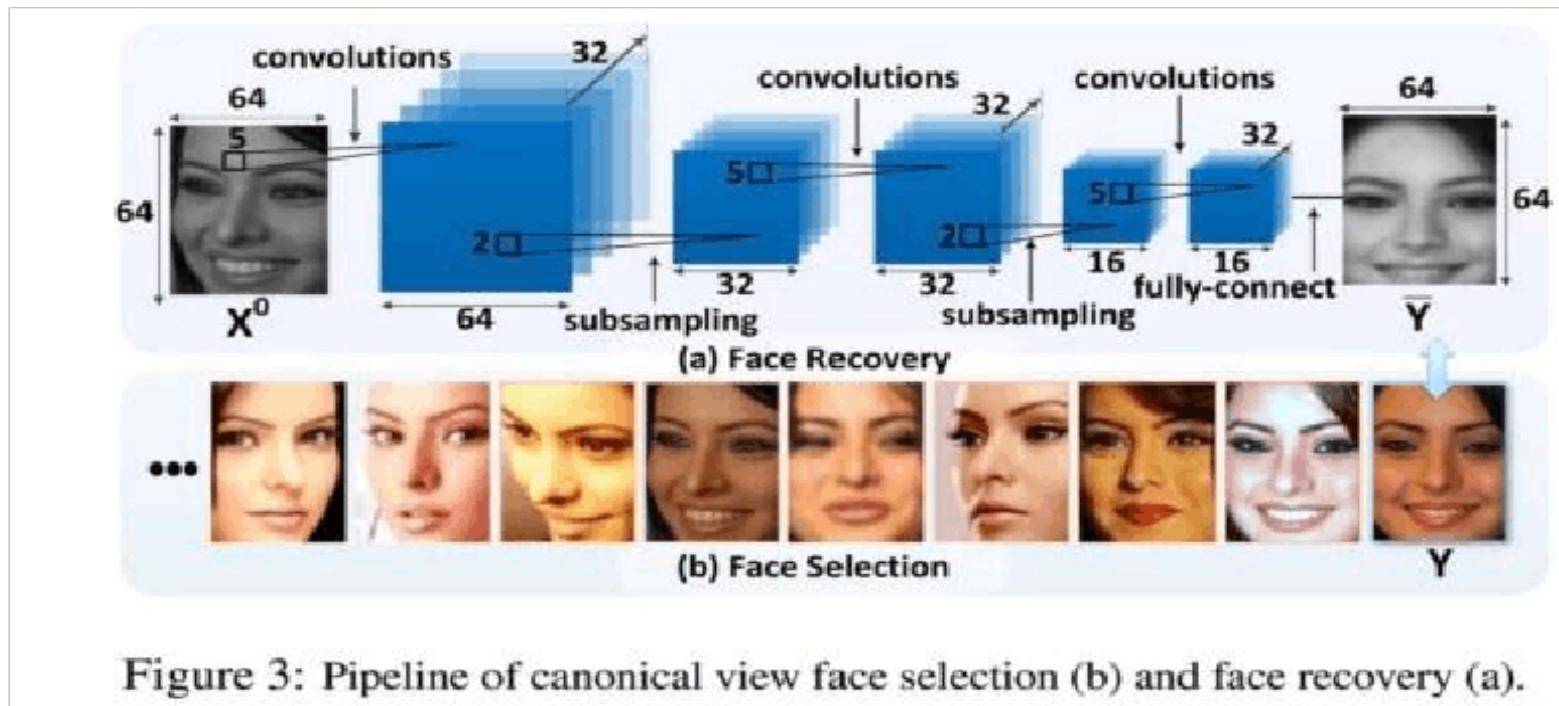
## 2.1 深度学习场景 - 无人驾驶

**无人驾驶**：深度学习利用其深层的神经网络，通过一定的算法能训练出一个识别率非常高的分类器，从而使环境感知部分高精度的完成，为驾驶决策模块提供正确的环境信息，保证无人驾驶正常的完成。



## 2.2 深度学习场景 - 人脸识别

人脸识别：人脸信息的识别，是对人脸的信息加以提取然后进行识别的办法，一个最重要的目标就是分辨不同人的信息，辨别身份。面部识别的主要方式有：几何结构、子空间局部特征以及深度学习。



安防领域

开户审核

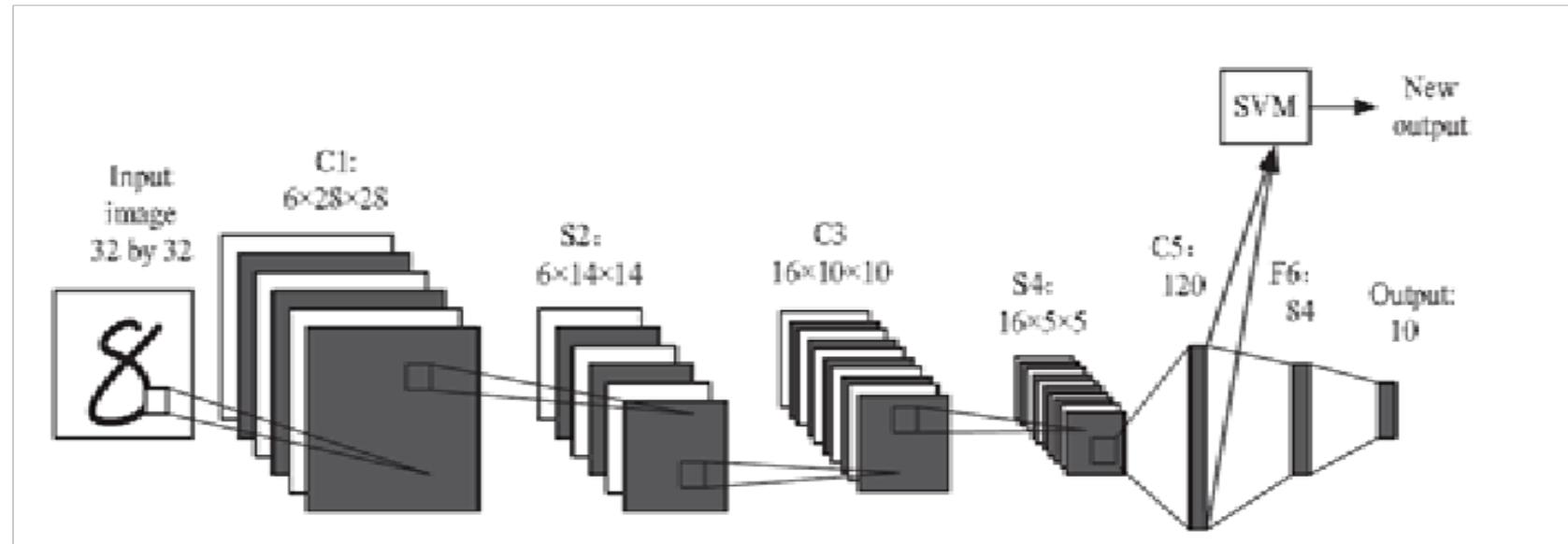
考勤门禁

.....

Figure 3: Pipeline of canonical view face selection (b) and face recovery (a).

## 2.3 深度学习场景 - 文字识别

**文字识别**：基于深度学习的文字识别系统的实现方法，属于图像处理技术领域，将包含文字的图片进行预处理、切分、识别，重组成一段文本，从而实现图片到文本的转换。涉及到图像预处理、图片切分、图片识别和文字重组。



审核

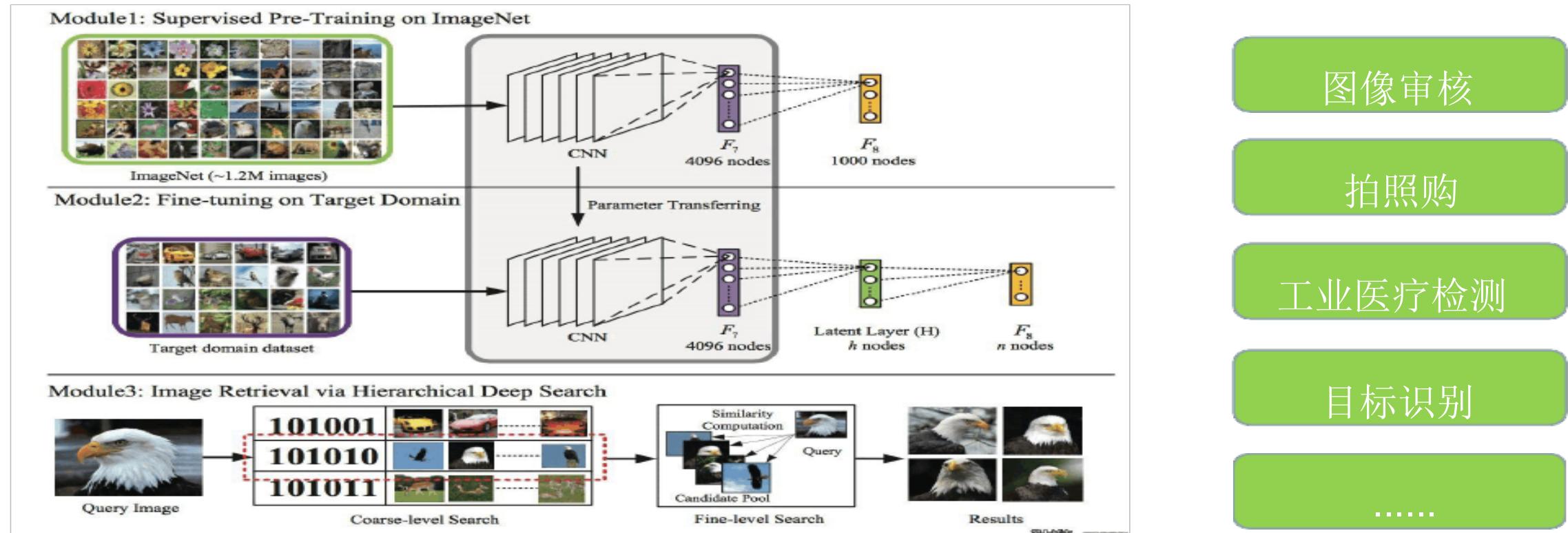
车牌识别

文档识别

.....

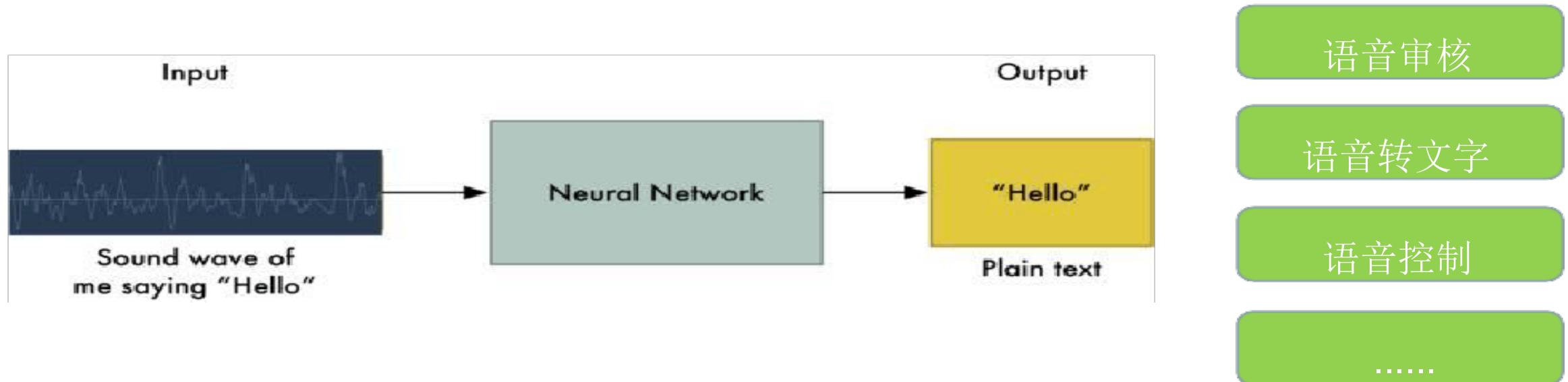
## 2.4 深度学习场景 - 图像识别

图片识别：图像识别与人脸及文字识别类似，主要流程包括图像预处理，图像分割，图像特征提取和图像分类。



## 2.5 深度学习场景 - 语音识别

**语音识别**：一个完整的语音识别系统可大致分为 3部分：语音特征提取、声学模型与模式匹配和语言模型与语言处理。其中声学模型是识别系统的底层模型，并且是语音识别系统中最关键的一部分。

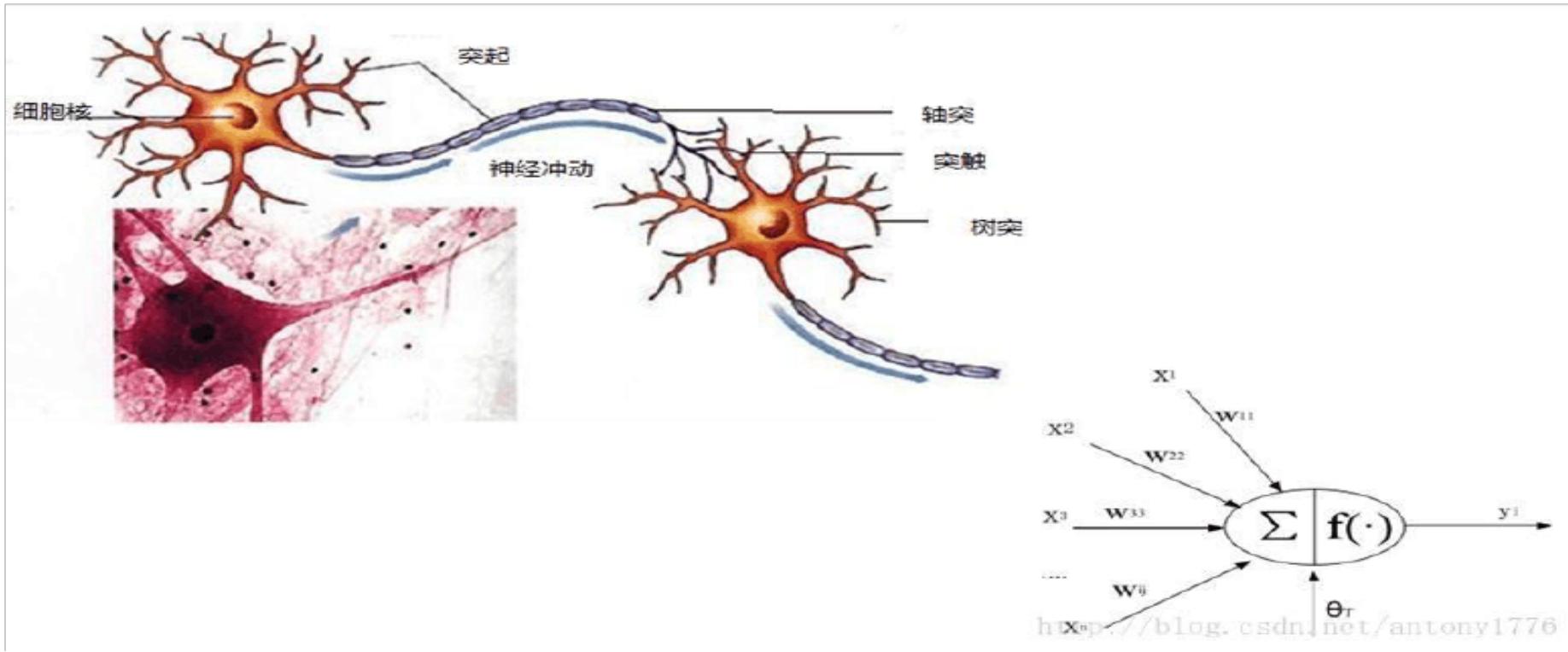


# 目录

- 深度学习概述
- 深度学习常见场景
- 深度学习常用算法介绍
- 深度学习常用框架介绍
- 深度学习未来和展望

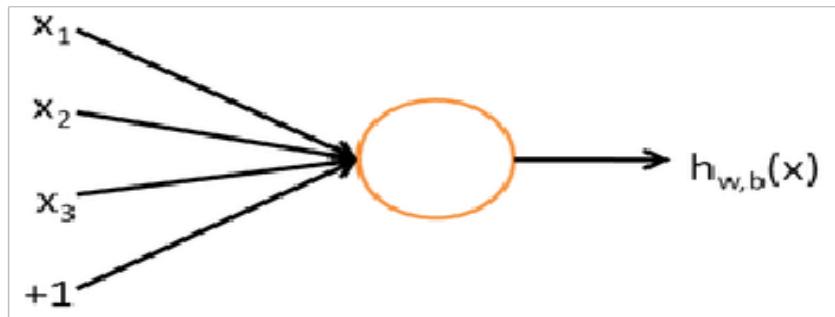
# 3.1 人工神经网络（ ANN ）

- 人工神经网络（ Artificial Neural Networks ）是一种模仿生物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点（神经元）之间相互连接的权重，从而达到处理信息的目的。

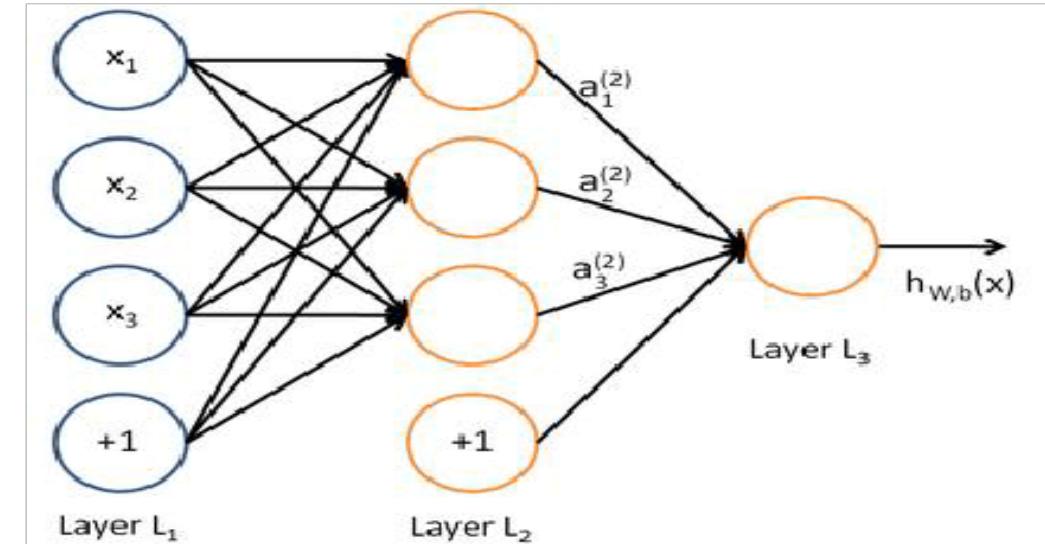


# 3.1 人工神经网络 ( ANN )

- 神经网络



$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^3 W_i x_i + b)$$



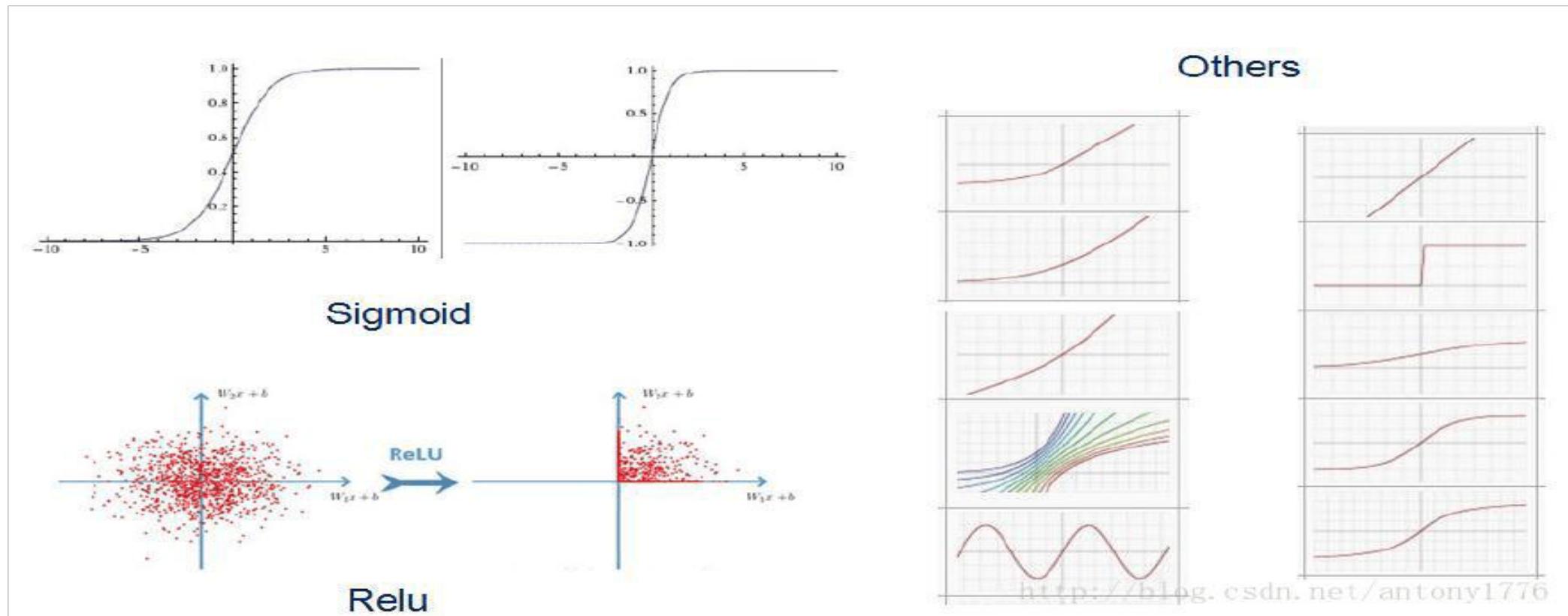
$$\begin{aligned}a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\h_{W,b}(x) &= a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})\end{aligned}$$

# 3.1 人工神经网络（ ANN ）

- 人工神经网络的重要概念：

- 1 权值矩阵：相当于神经网络的记忆！在训练的过程中，动态调整和适应。

- 2 激励函数：

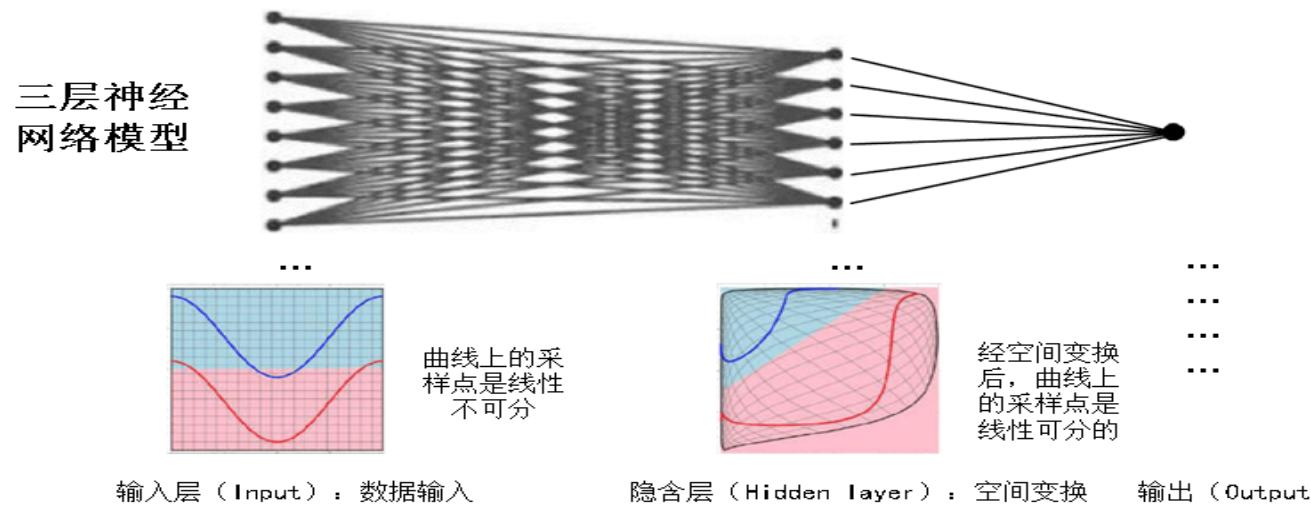


# 3.1 人工神经网络（ ANN ）

- 人工神经网络的重要概念：

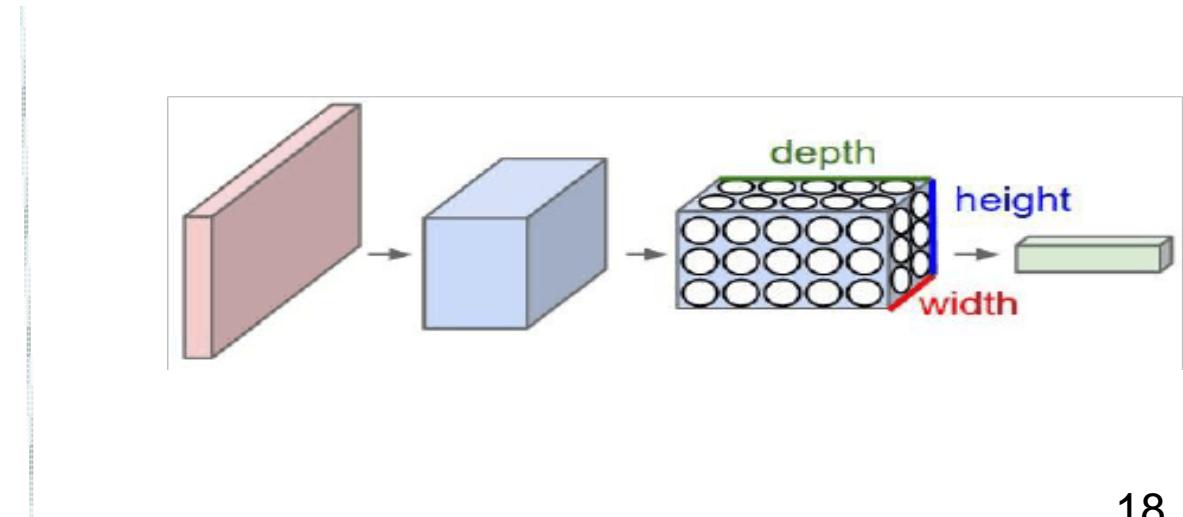
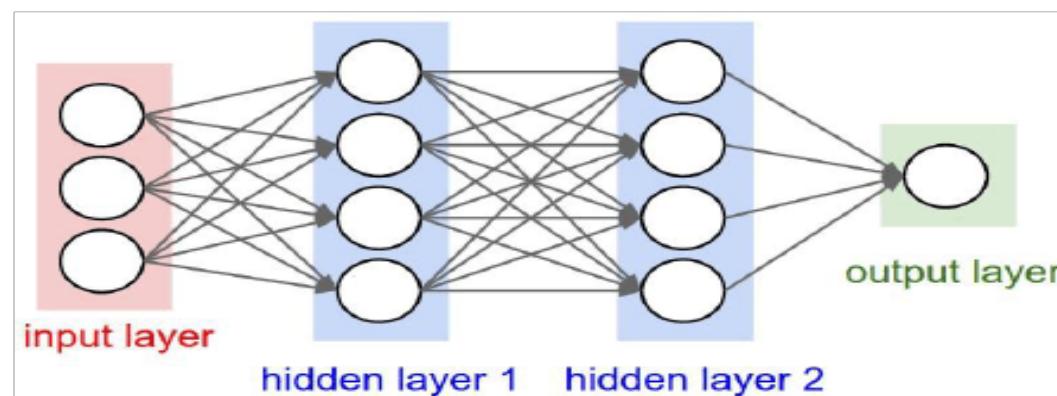
激励函数很重要，无论是对建立神经网络的模型，还是理解神经网络。首先要了解，它有以下几个影响：

- 1 如何能更好的求解目标函数的极值！——高等数学中求解函数极值的知识！可微，单调！
- 2 如何提升训练效率，让梯度的优化方法更稳定；
- 3 权值的初始值，不影响训练结果！



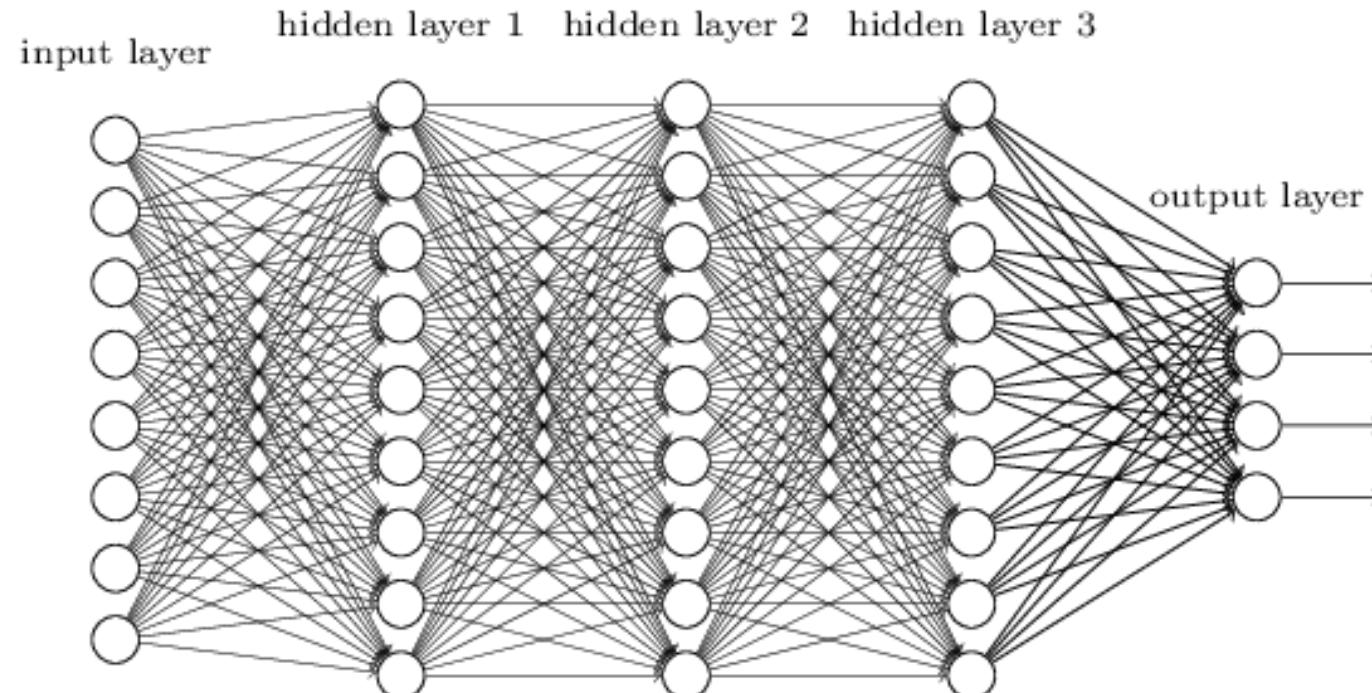
# 3.1 卷积神经网络（ CNN ）

- 卷积神经网络（ Convolutional Neural Networks / CNNs / ConvNets ）与普通神经网络非常相似，它们都由具有可学习的权重和偏置常量（biases）的神经元组成。每个神经元都接收一些输入，并做一些点积计算，输出是每个分类的分数，普通神经网络里的一些计算技巧到这里依旧适用。
- 与普通神经网络不同之处：卷积神经网络默认输入是图像，可以让我们把特定的性质编码入网络结构，使我们的前馈函数更加有效率，并减少了大量参数。



# 卷积神经网络

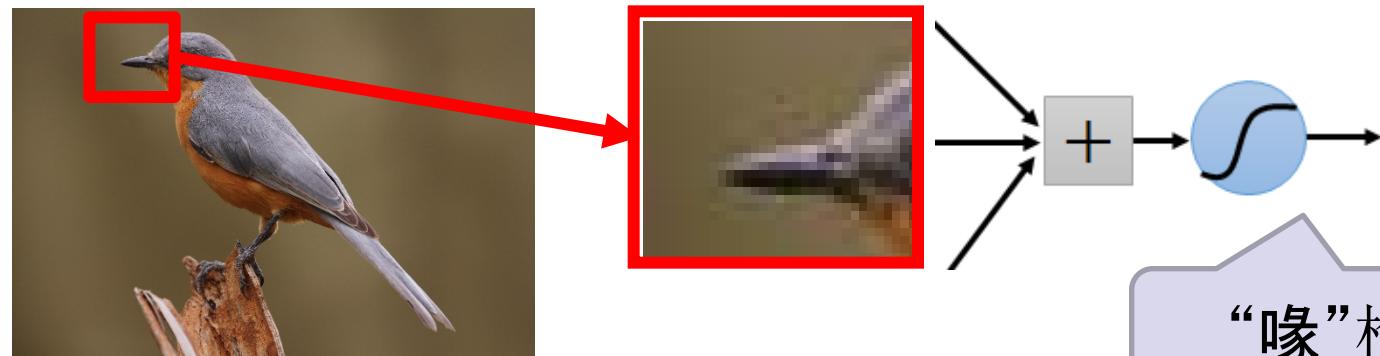
- 我们知道学习规模小更好。
- 从这个完全连接的模型中， 我们真的需要所有的连接吗？
- 可以共享其中一些吗？



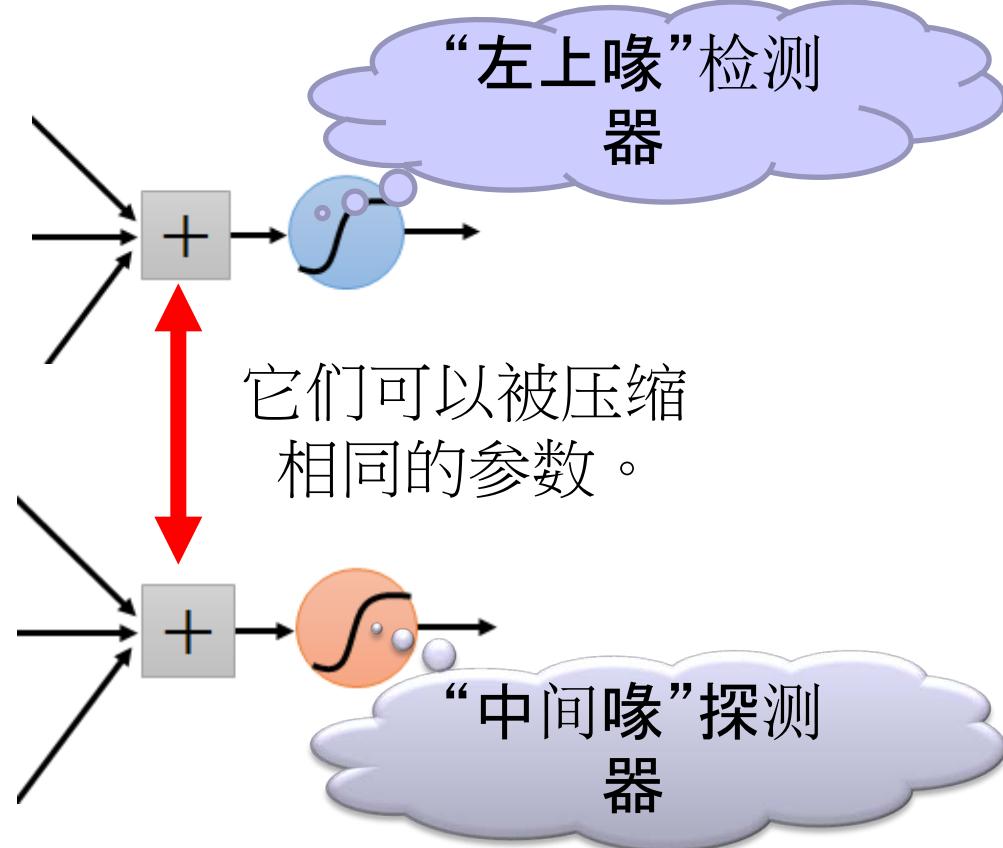
## 考虑学习图像：

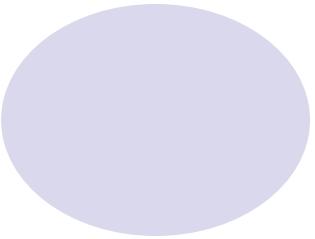
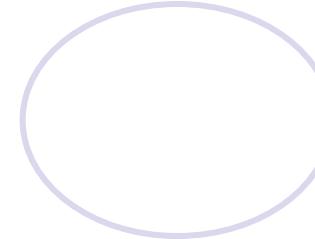
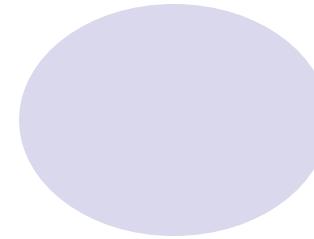
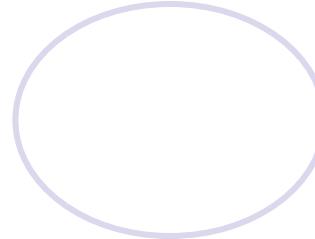
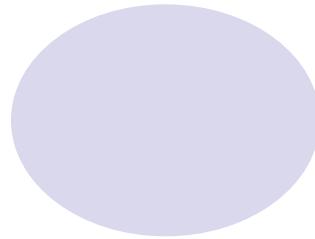
- 有些特征图案比整个图像小很多

可以用更少的参数表示一个小的区域

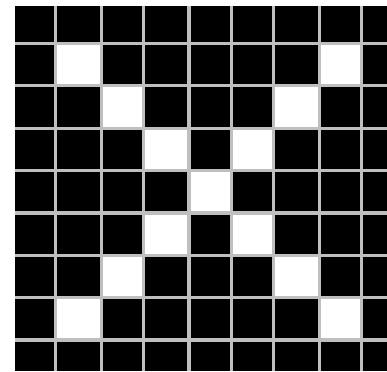


相同的模式出现在不同的位置：它们可以被压缩！  
训练很多这样的“小型”探测器又如何呢？每个探测器必须“四处移动”。

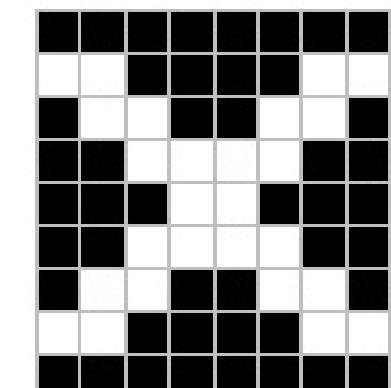
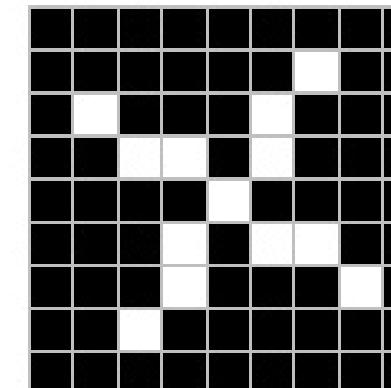
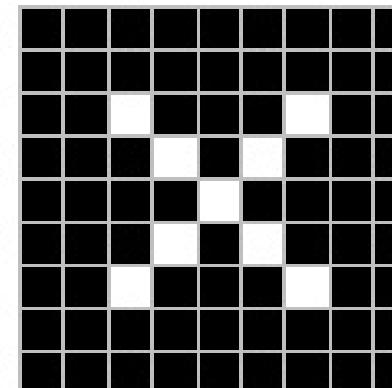
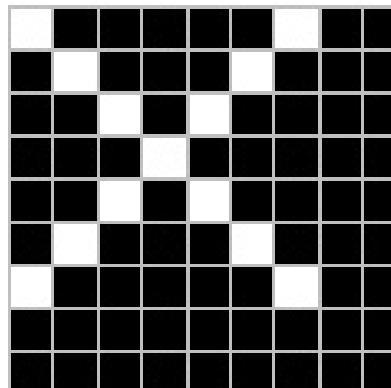


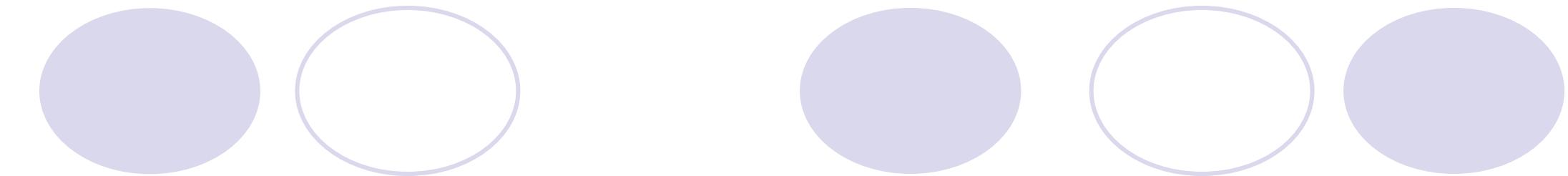


比如，我现在要训练一个最简单的**CNN**，用来识别一张图片里的字母是**X**还是**O**。

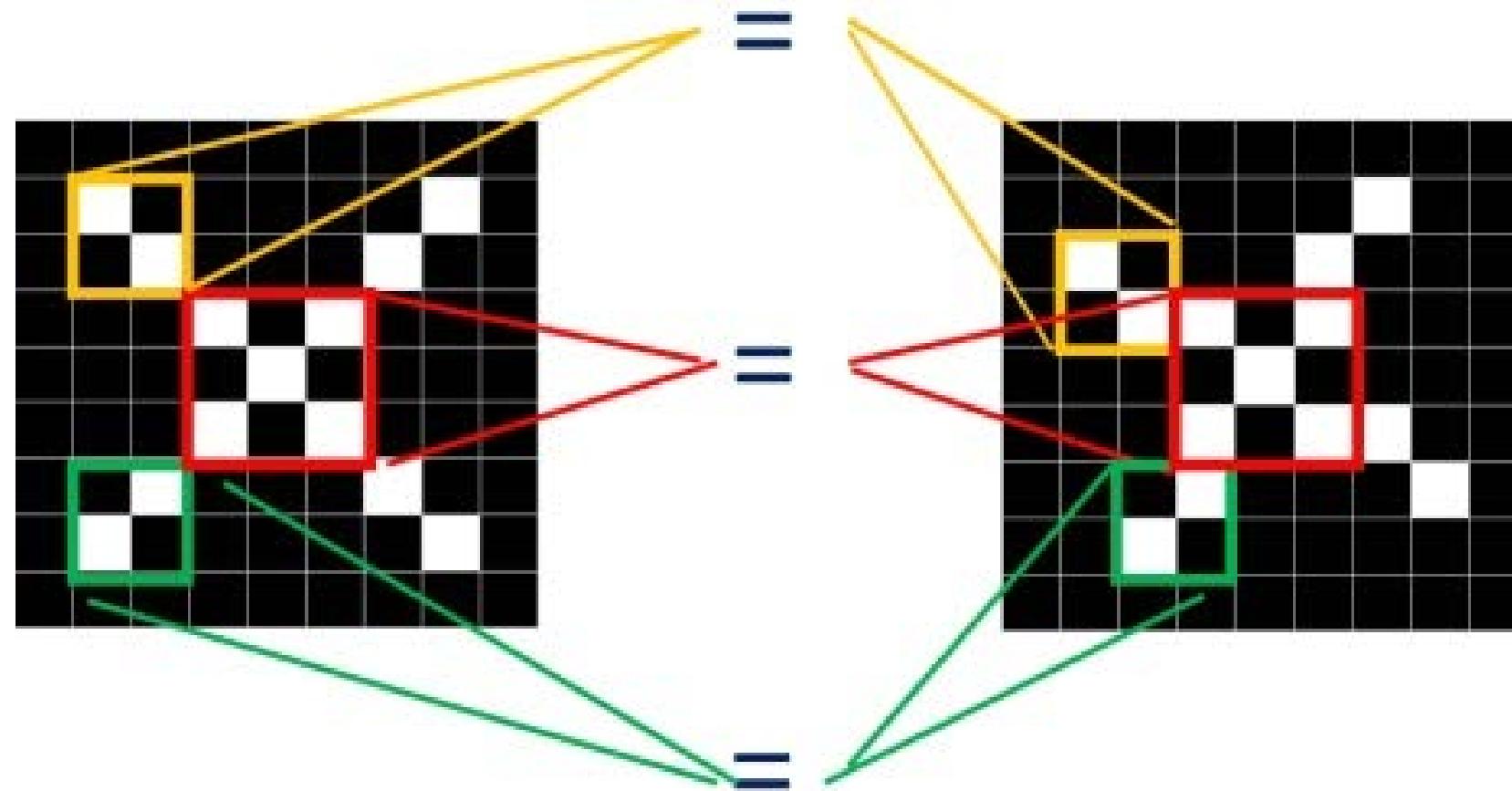


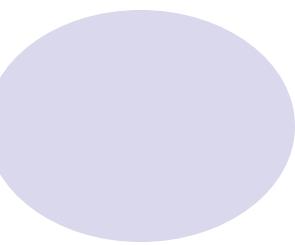
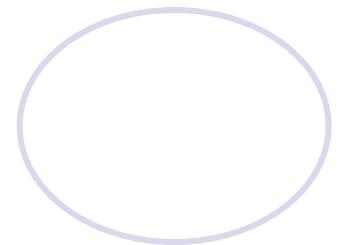
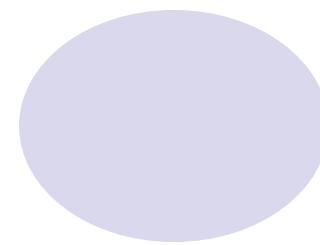
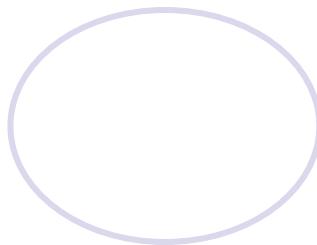
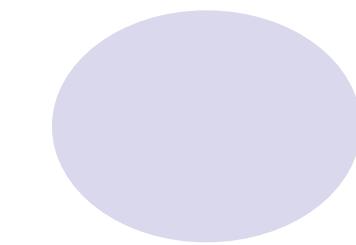
以下四个都是**X**，但它们和之前那张**X**明显不一样。（“欠拟合”）





观察这两张X图，可以发现尽管像素值无法一一对应，但也存在着某些共同点。



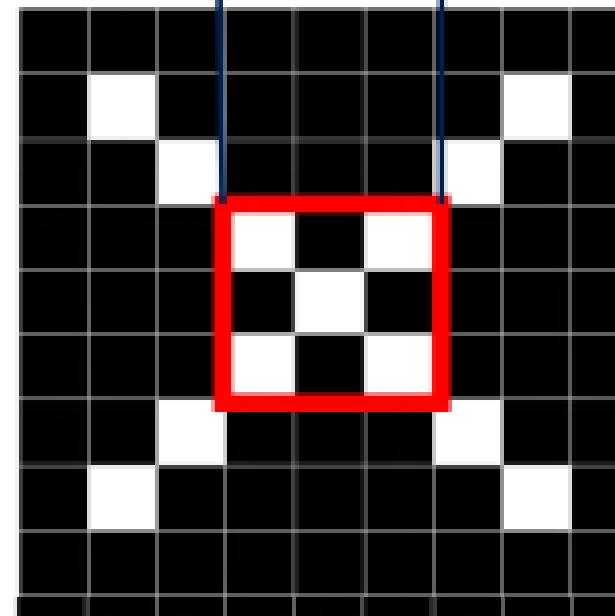


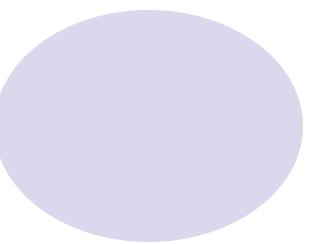
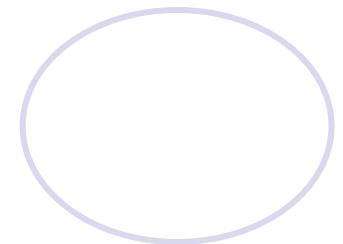
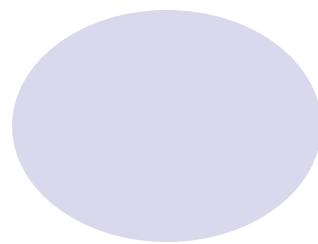
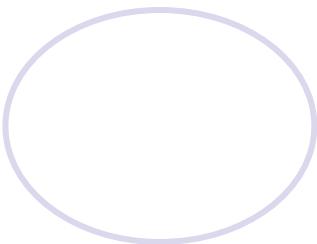
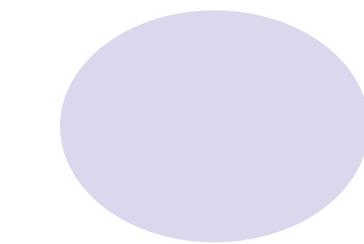
从标准的X图中我们提取出三个特征 (feature)

1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1





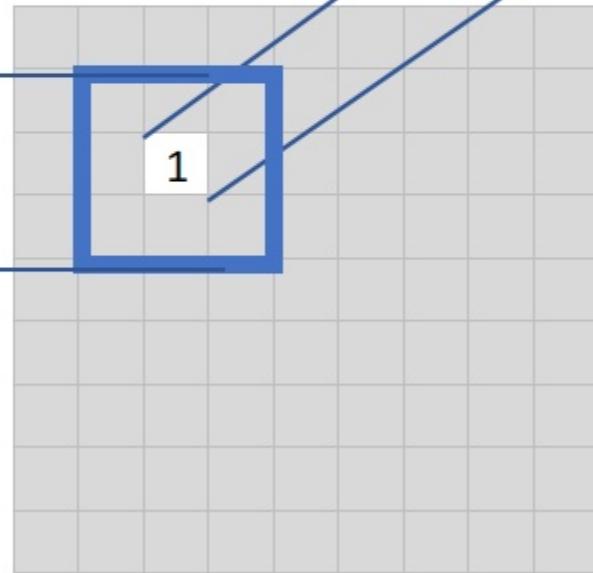
feature在CNN中也被成为卷积核 (filter)

1	-1	-1
-1	1	-1
-1	-1	1

1	1	1
1	1	1
1	1	1

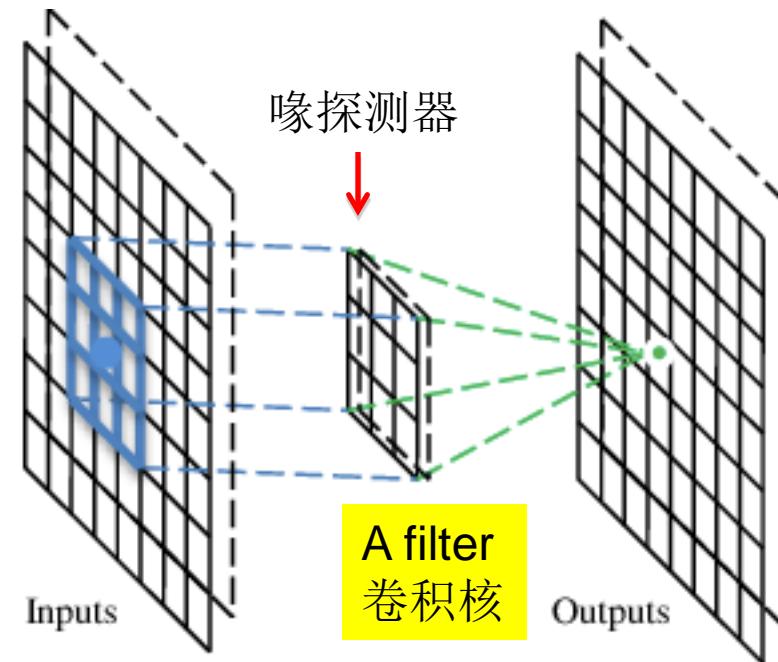
$$\frac{1+1+1+1+1+1+1+1}{9} = 1$$

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1



# 卷积层

CNN是具有一些卷积层的神经网络（以及其他一些层）。卷积层有一些卷积运算的卷积核的集合。



# 卷积

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

6 × 6 黑白图片

这些是要学习的网络参数。

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

-1	1	-1
-1	1	-1
-1	1	-1

卷积核 2

⋮ ⋮

每个卷积核都会检测到一个  
个小图案 (3 × 3)。

卷积

步长=1

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

点积



1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

3

-1

6 × 6 图片

# 卷积

如果步长=2

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

6 x 6 点积

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

3

-3

# 卷积

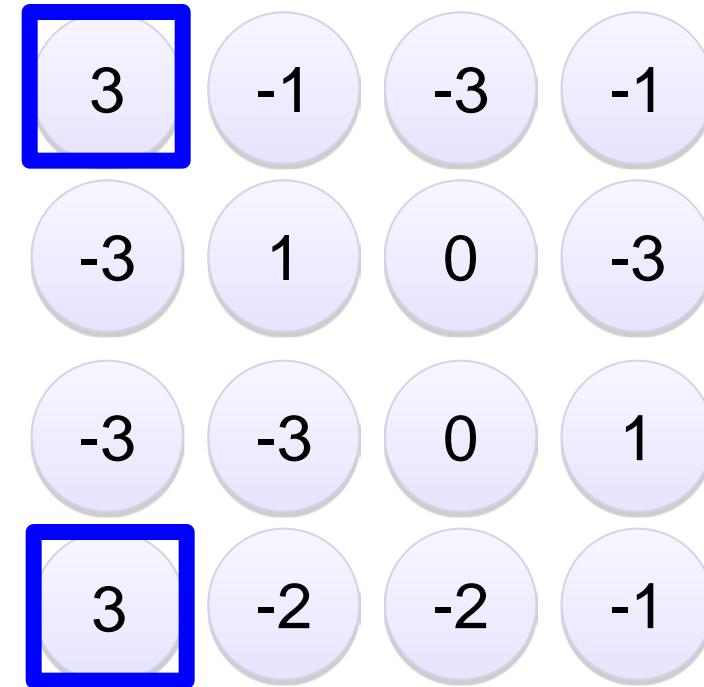
步长=1

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

6 × 6 图片

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1



卷积

步长=1

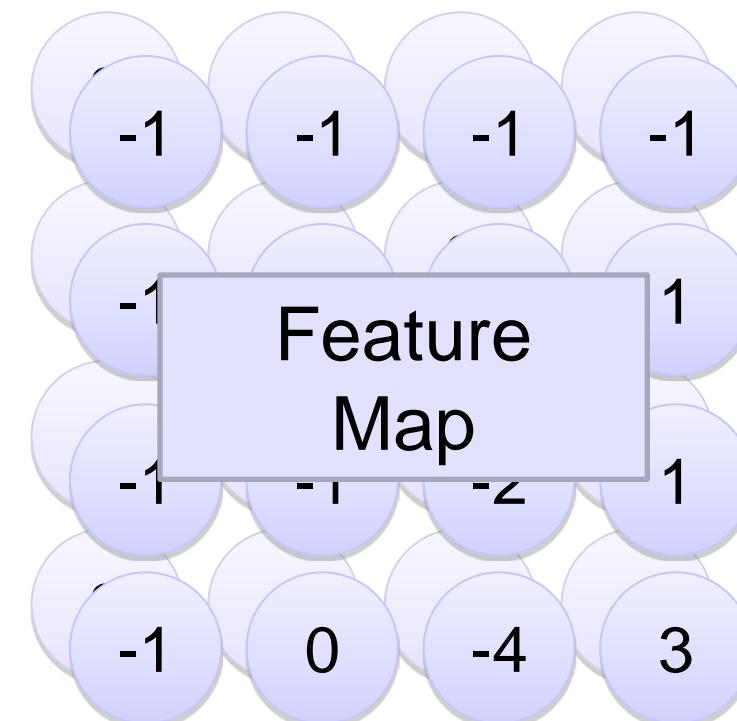
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 × 6 图片

-1	1	-1
-1	1	-1
-1	1	-1

卷积核 2

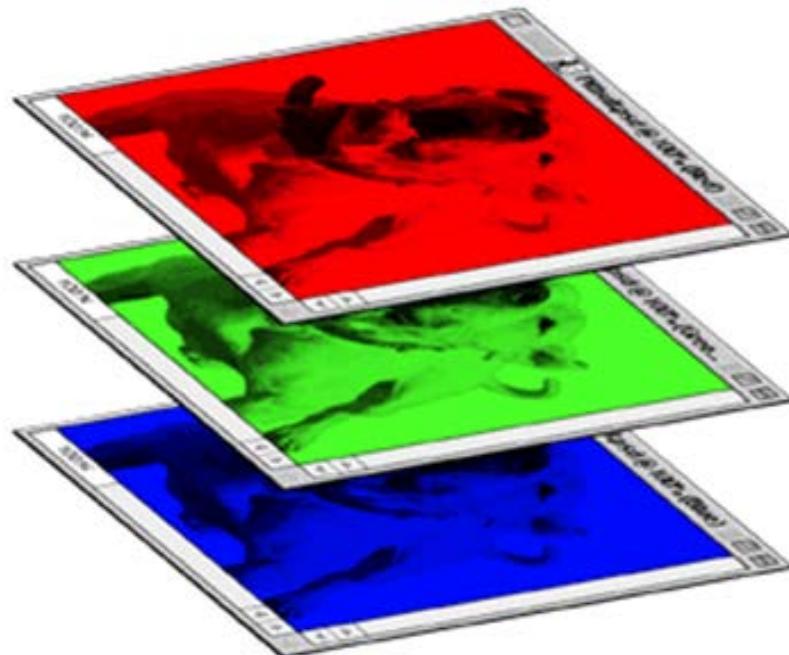
对每个卷积核重复此操作



Two 4 × 4 images  
Forming 2 × 4 × 4 matrix

# 彩色图像: RGB 3通道

彩图



1	-1	-1
-1	1	-1
-1	-1	1

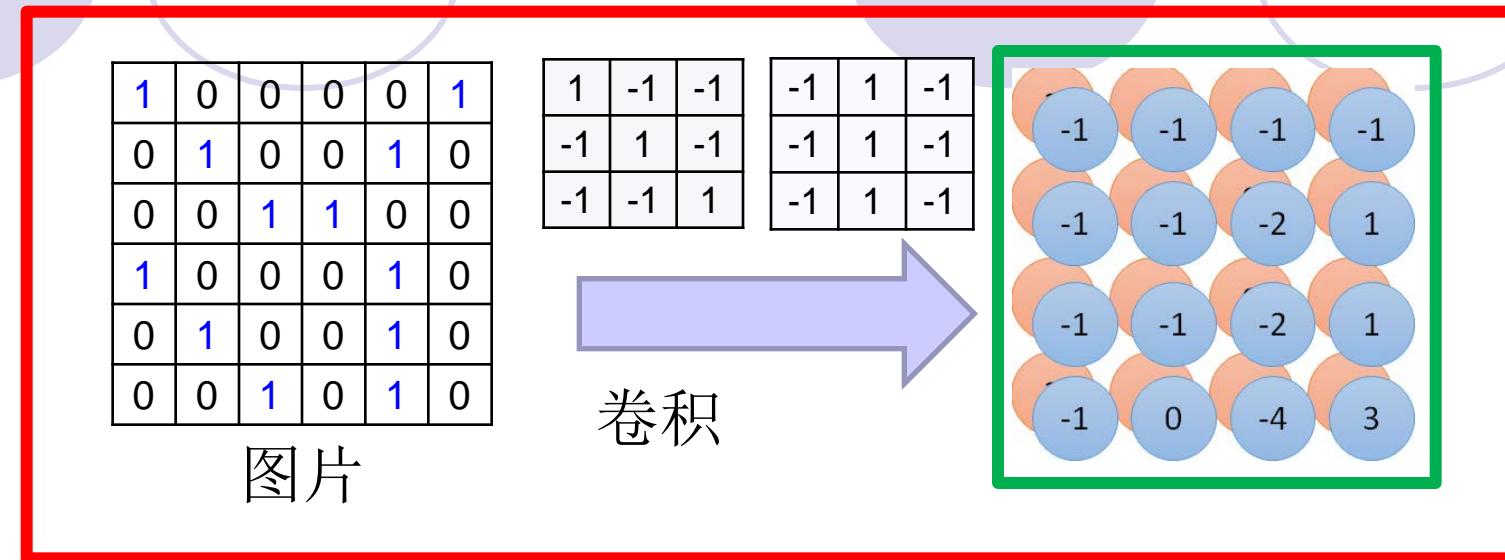
卷积核1

-1	1	-1
-1	1	-1
-1	1	-1

卷积核2

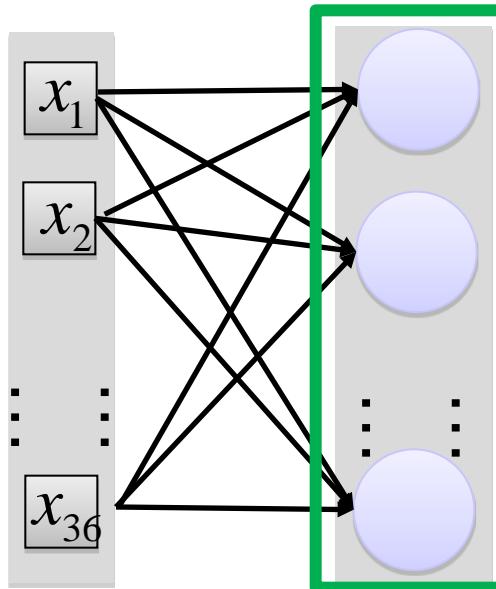
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

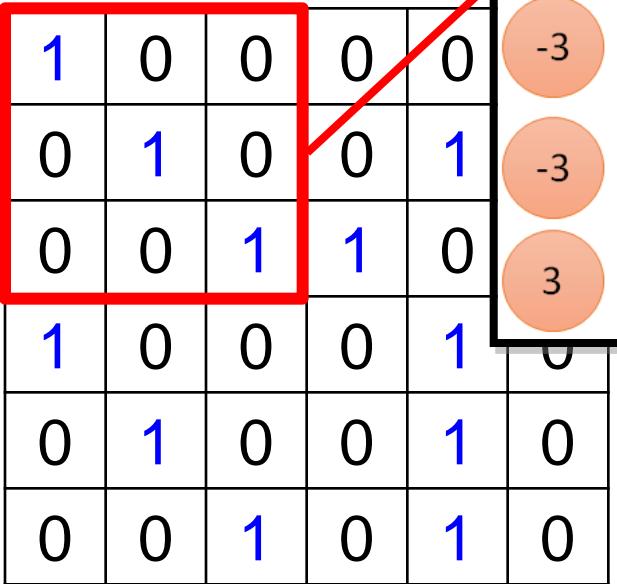
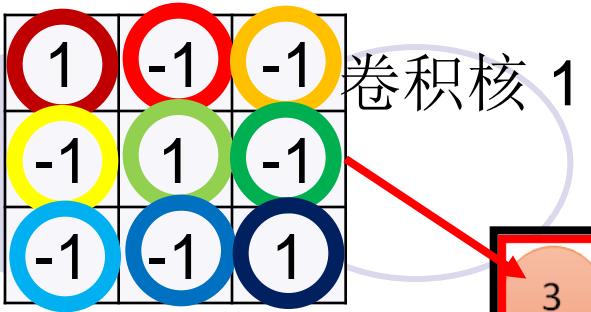
# 卷积 v.s. 全连接



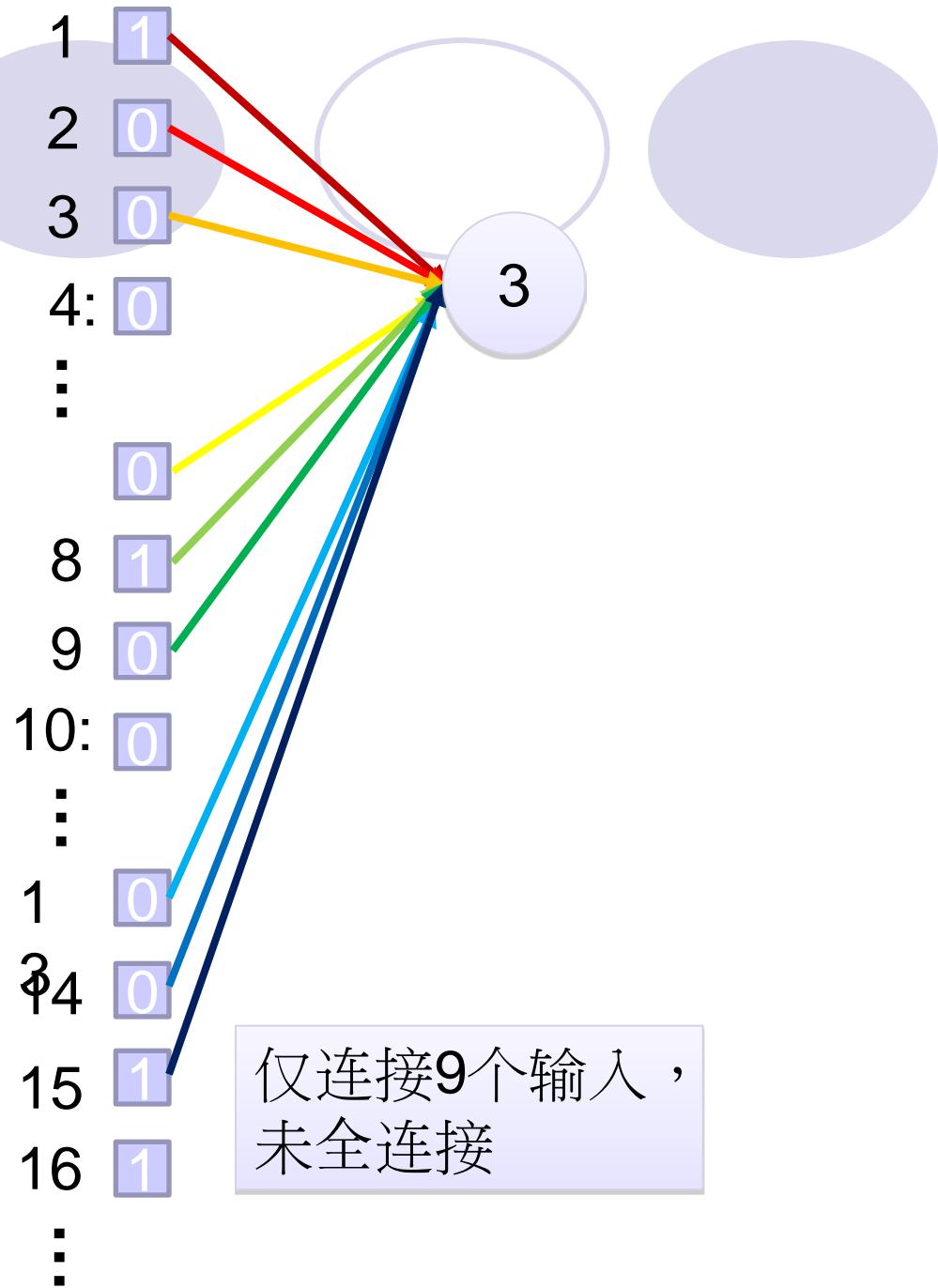
全连接

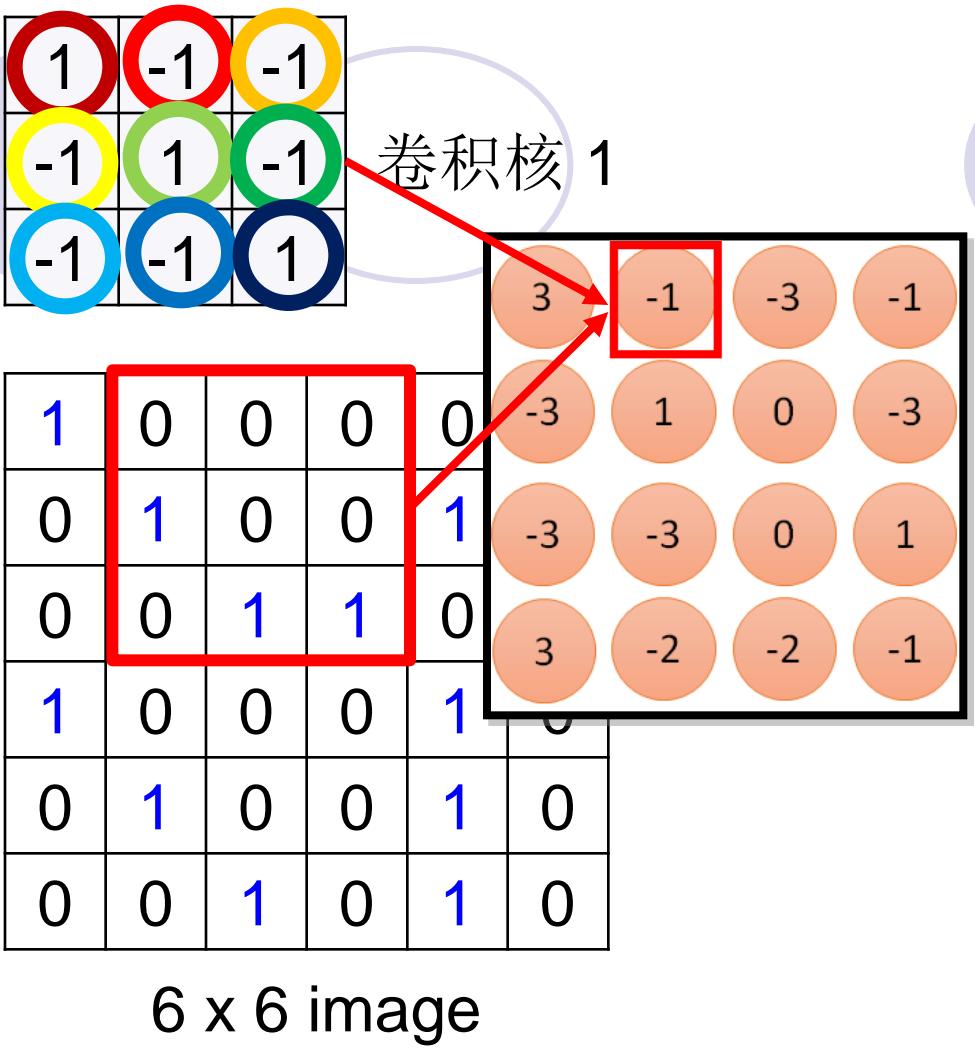
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0





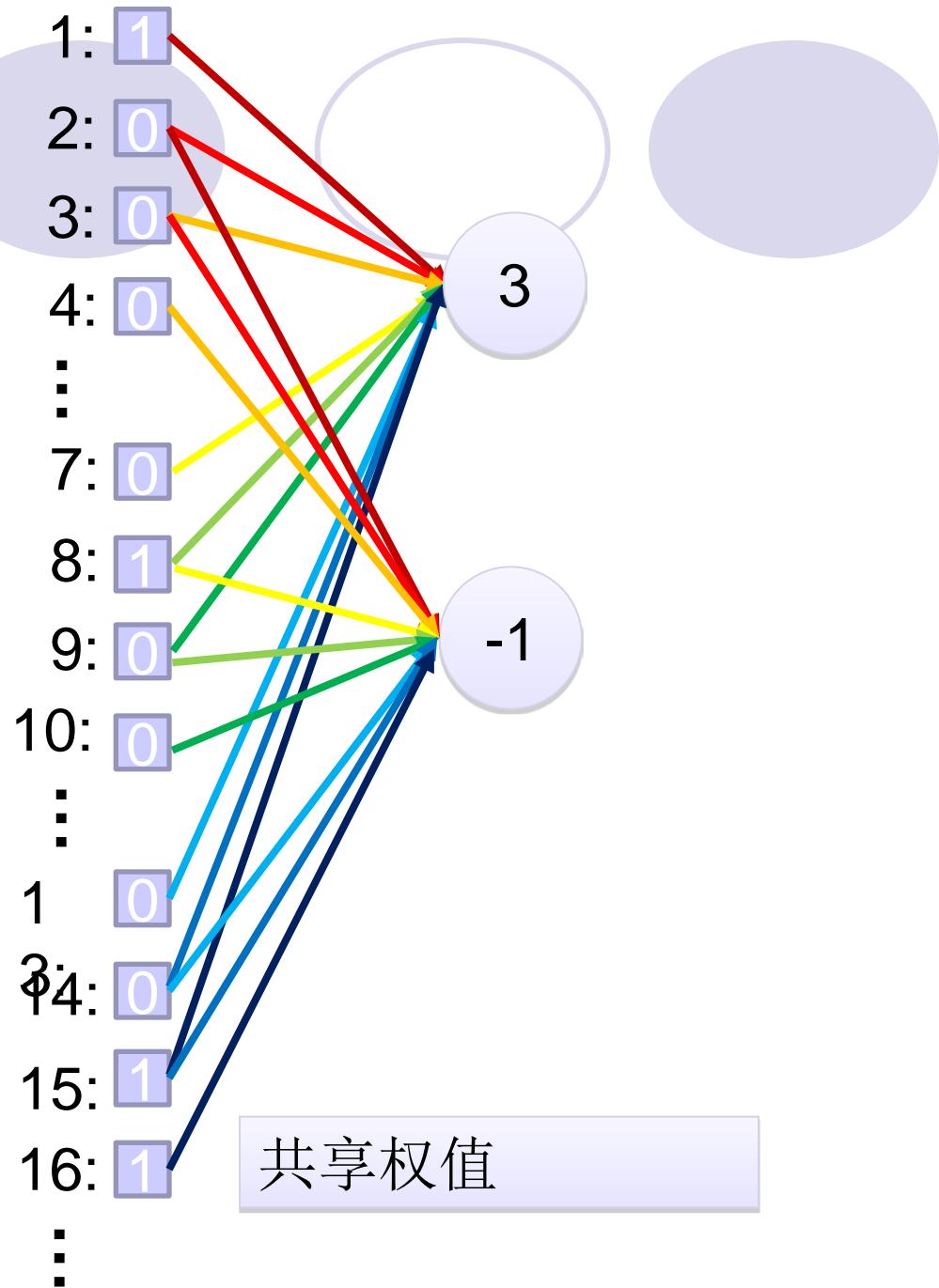
参数更少！





参数更少

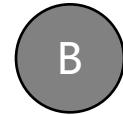
参数进一步减少！



卷积神经网络是一个全连接的神经网络，中间隐层通常包含多个卷积层。



错



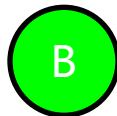
对

提交

卷积是一种有效提取图片特征的方法。一般用一个正方形卷积核，遍历图片上的每一个像素点。图片与卷积核重合区域内相对应的每一个像素值乘卷积核内相对应点的权重，然后求和，再加上偏置后，最后得到输出图片中的一个像素值。



A 错

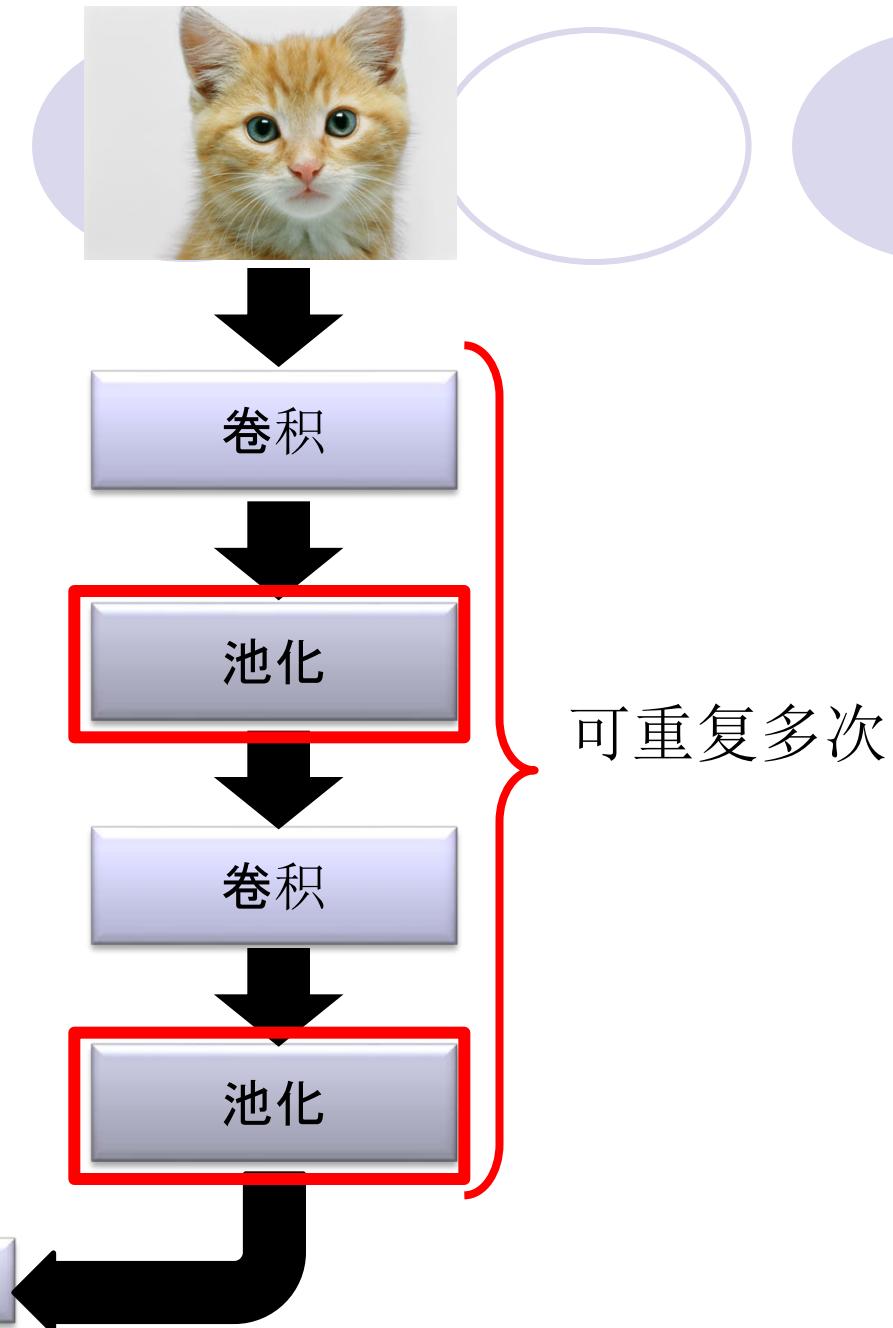
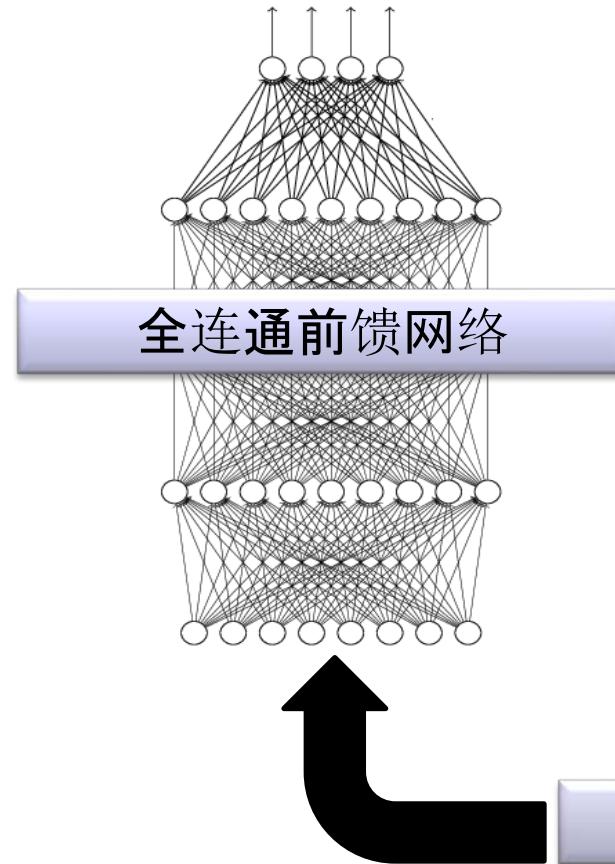


B 对

提交

# 完整 CNN

cat dog .....



可重复多次

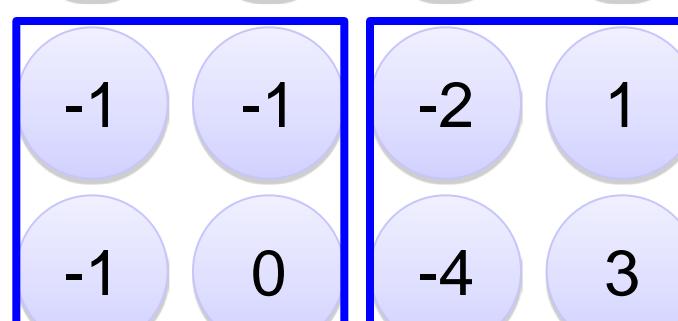
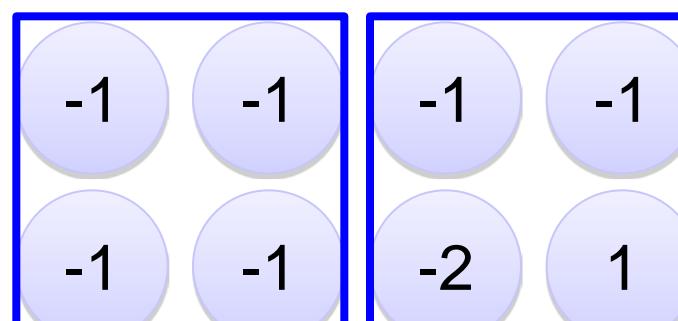
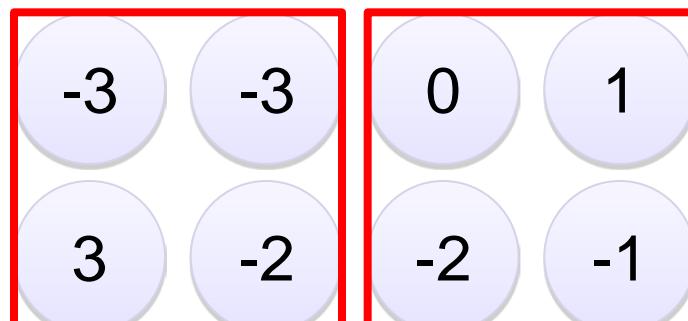
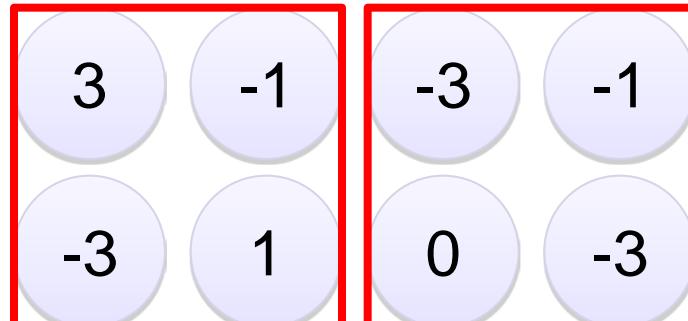
# 池化

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

-1	1	-1
-1	1	-1
-1	1	-1

卷积核 2



# 为什么池化

- 采样像素不会改变对象

鸟



采样

鸟

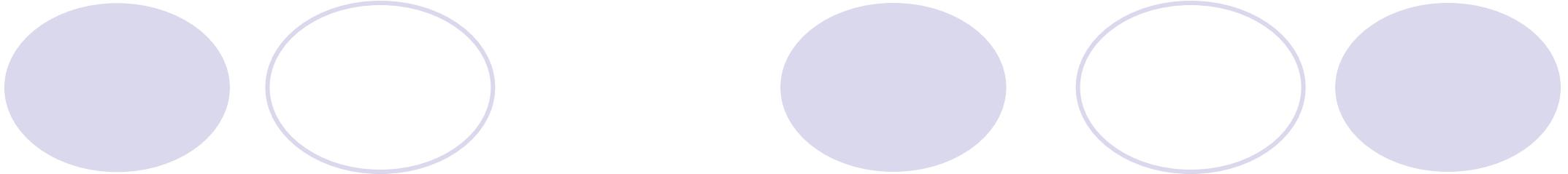


我们可以采样使图像更小

→ 更少的参数来表征图片特征



## 池化

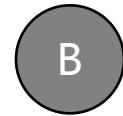


- 通过卷积获得了特征之后，如果直接利用这些特征训练分类器，计算量是非常大的。
- 对不同位置的特征进行聚合统计,称为池化 (**pooling**)。
- 池化常用方法： 平均池化、最大池化。
- 卷积神经网络在池化层丢失大量的信息，从而降低了空间分辨率，导致了对于输入微小的变化，其输出几乎是不变的。

卷积神经网络中，对不同位置的特征进行聚合统计，称为池化（pooling）。池化不会丢失图像的信息，也不会降低其空间分辨率。



A 错



B 对

提交

# CNN以两种方式压缩一个完全连接的网络

- 减少连接数
- 共享连接权值
- 池化进一步降低了复杂性

池化

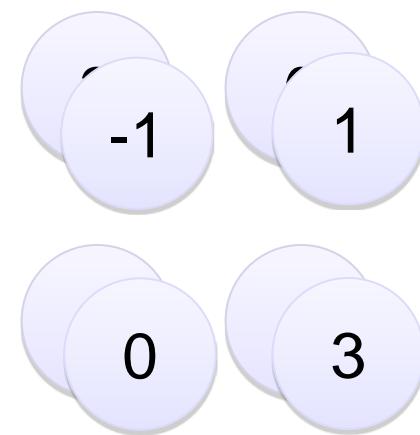
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 × 6 图片

卷积

池化

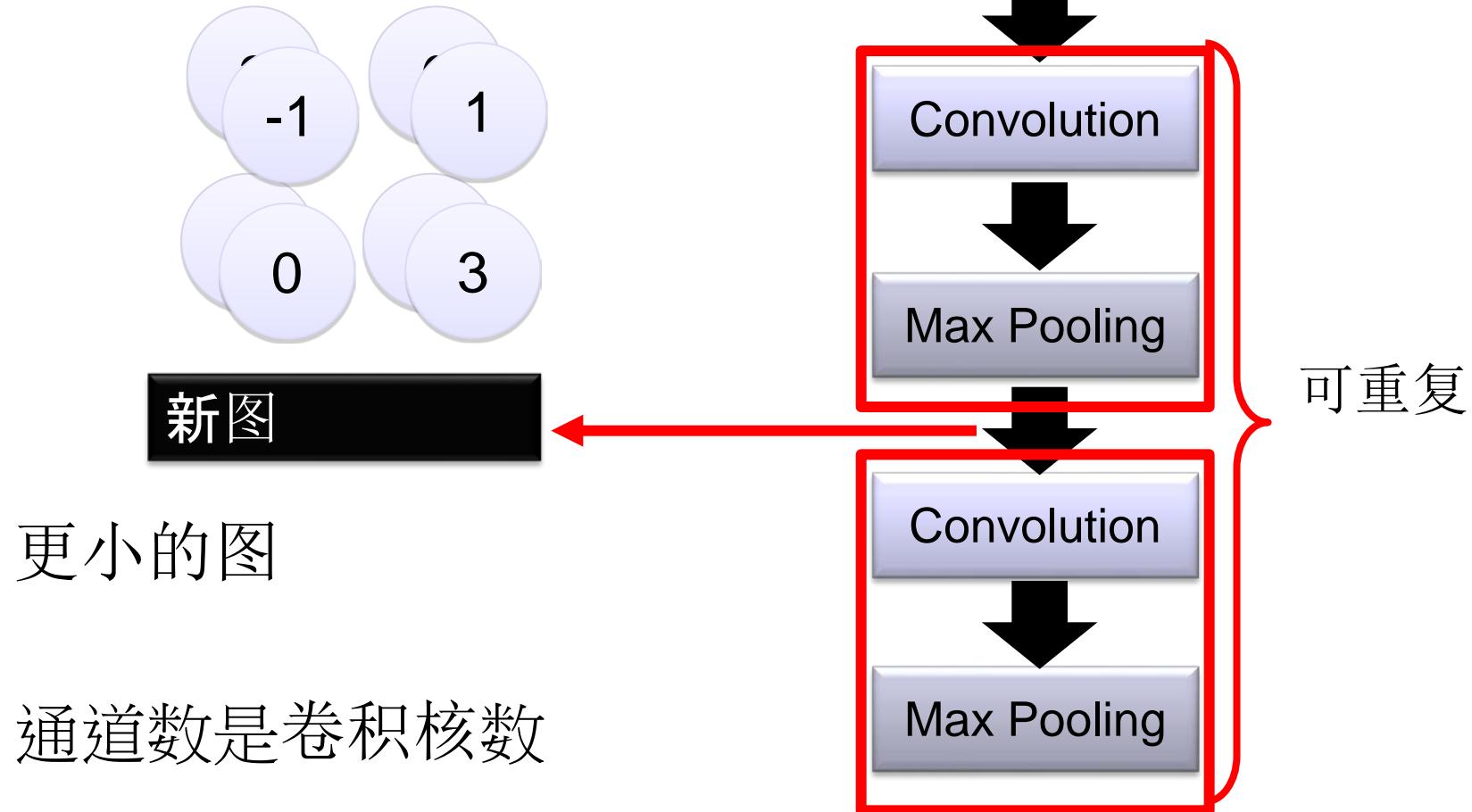
更小的图



2 × 2 图片

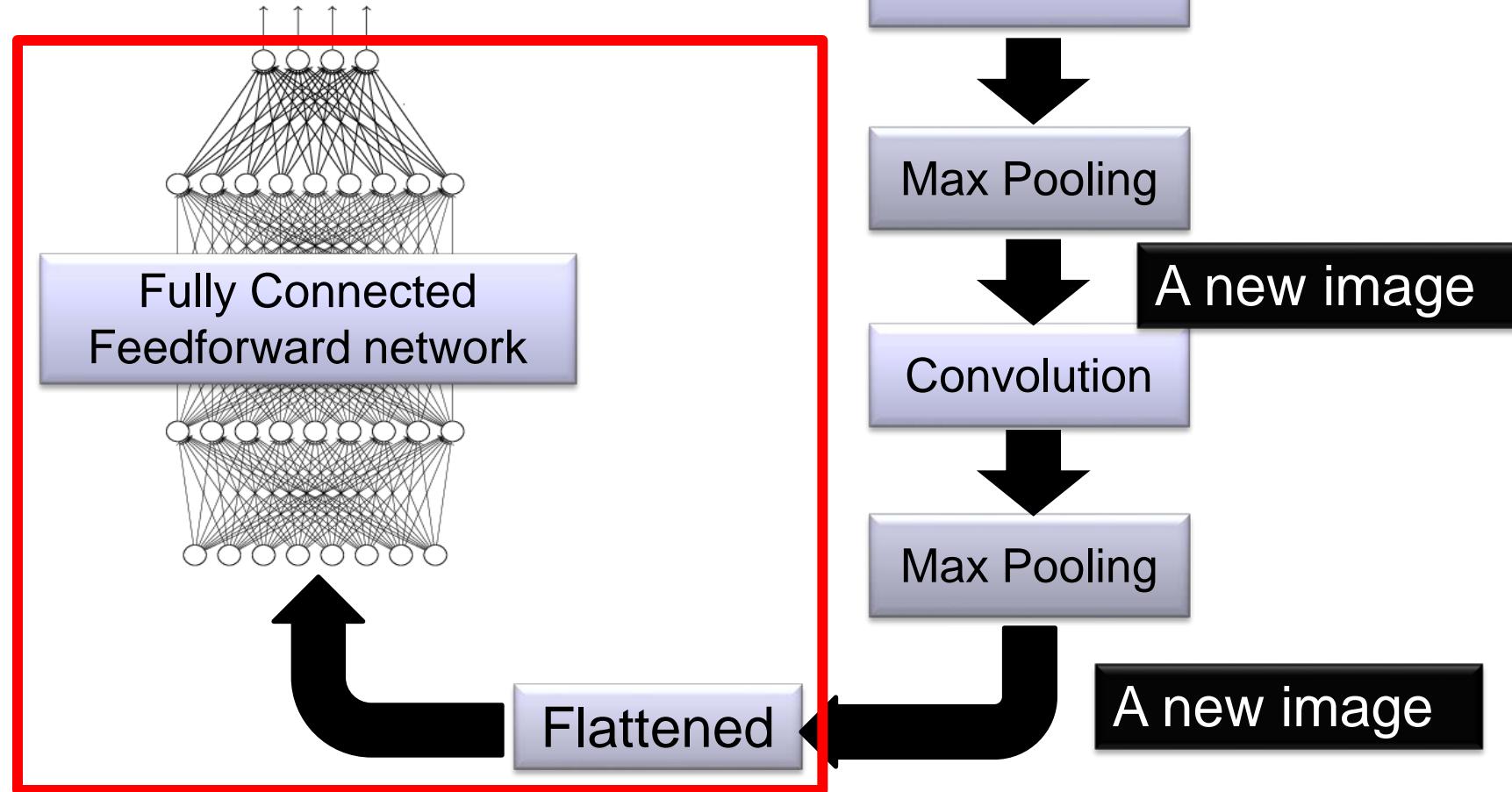
每个卷积核  
是一个通路

# 完整 CNN

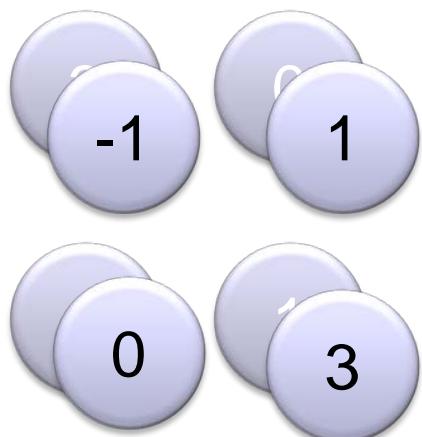


# The whole CNN

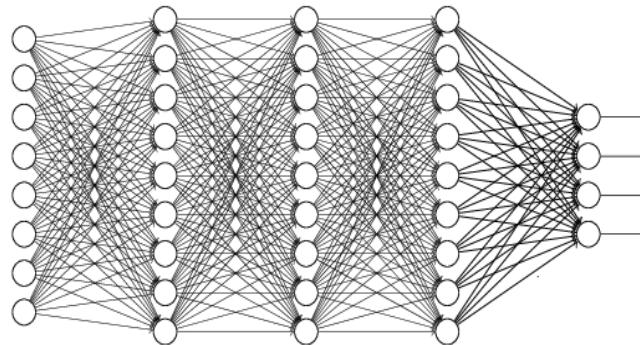
cat dog .....



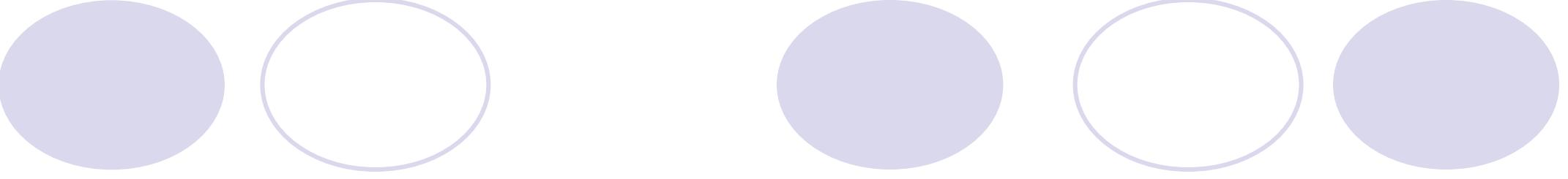
扁平化



扁平化



全连通前向神经网络



## 卷积神经网络（CNN）：卷积+池化+全连接

### 卷积：

- ✓ 局部特征提取
- ✓ 训练中进行参数学习
- ✓ 每个卷积核提取特定模式的特征

### 池化（下采样）：

- ✓ 降低数据维度，避免过拟合
- ✓ 增强局部感受野
- ✓ 提高平移不变性

### 全连接：

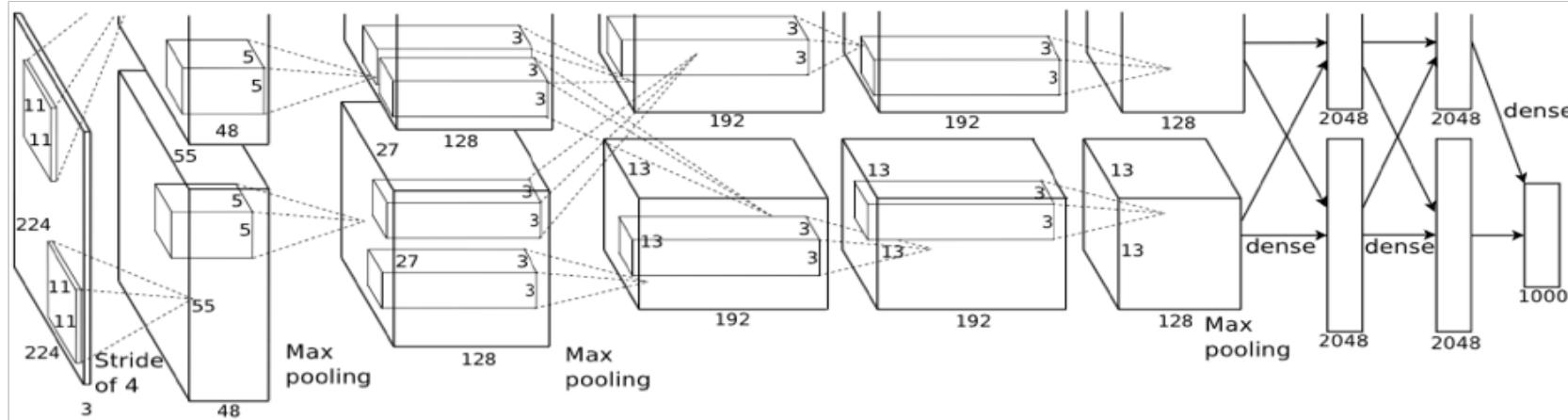
- ✓ 特征提取到分类的桥梁

# 卷积神经网络视频

- [https://www.bilibili.com/video/BV1sb411P7pQ?p=1&share\\_medium=iphone&share\\_plat=ios&share\\_session\\_id=6F58779B-D700-44A1-965F-B250466E9838&share\\_source=WEIXIN&share\\_tag=s\\_i&timestramp=1634889046&unique\\_k=Yxkf1s](https://www.bilibili.com/video/BV1sb411P7pQ?p=1&share_medium=iphone&share_plat=ios&share_session_id=6F58779B-D700-44A1-965F-B250466E9838&share_source=WEIXIN&share_tag=s_i&timestramp=1634889046&unique_k=Yxkf1s) 33:09
- [https://www.bilibili.com/video/BV1qb411P7JD?spm\\_id\\_fro m=333.999.0.0](https://www.bilibili.com/video/BV1qb411P7JD?spm_id_fro m=333.999.0.0) 15:24
- [https://www.bilibili.com/video/BV1R5411w715?from=search&seid=14451113221349708057&spm\\_id\\_from=333.337.0.0](https://www.bilibili.com/video/BV1R5411w715?from=search&seid=14451113221349708057&spm_id_from=333.337.0.0) 8:32

## 3.2 常见网络模型

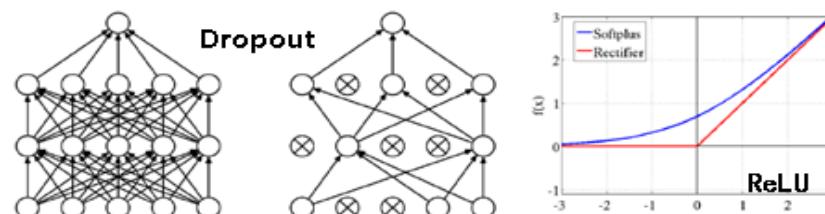
### ● AlexNet



AlexNet在LeNet基础上进行了更宽更深的网络设计，首次在CNN中引入了ReLU、Dropout和Local Response Norm (LRN)等技巧。网络的技术特点如下：

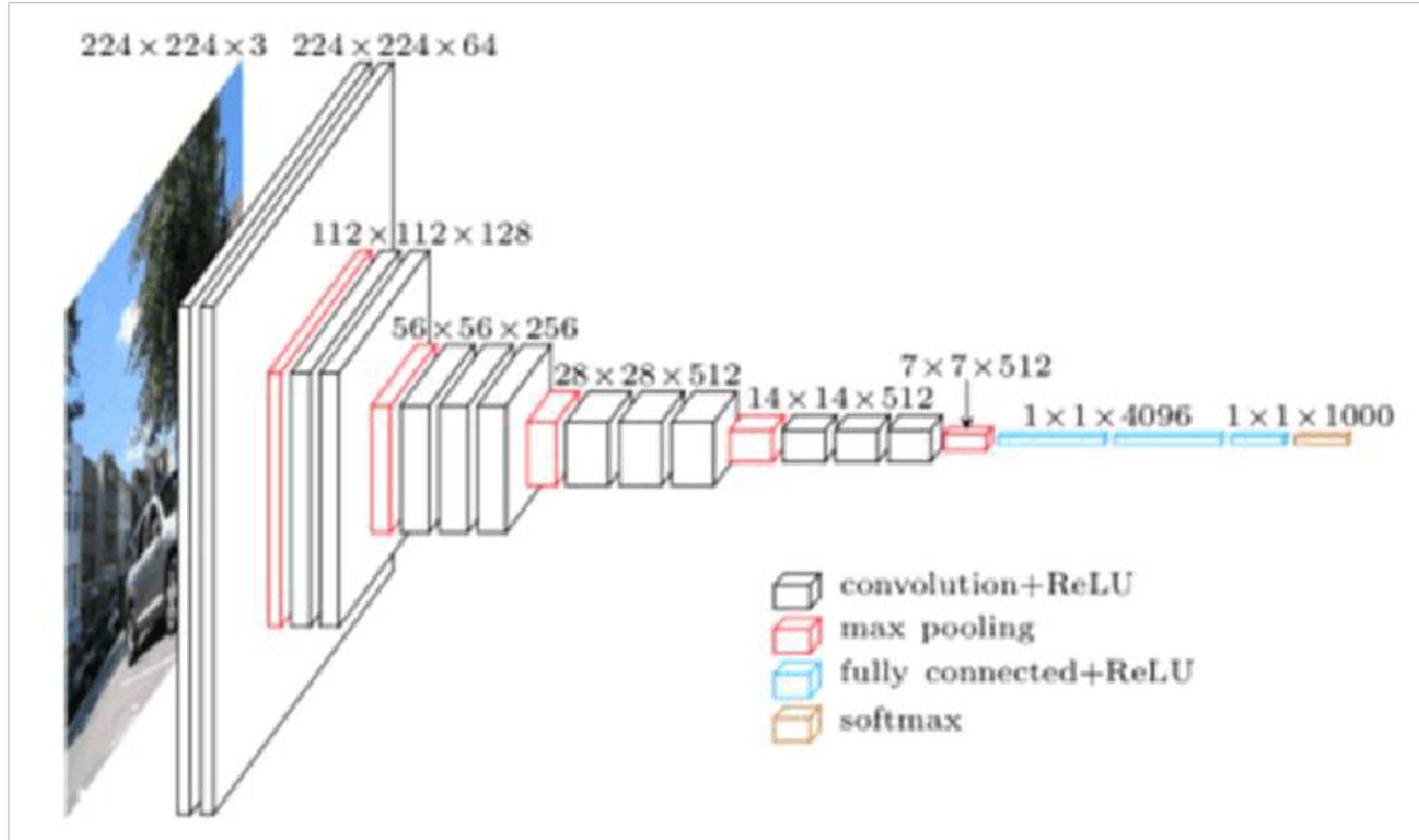
- 使用ReLU (Rectified Linear Units) 作为CNN的激活函数，并验证其效果在较深的网络超过了Sigmoid，成功解决了Sigmoid在网络较深时的梯度弥散问题，提高了网络的训练速率。
- 为避免过拟合，训练时使用Dropout随机忽略一部分神经元。
- 使用重叠的最大池化(max pooling)。最大池化可以避免平均池化的模糊化效果，而采用重叠技巧可以提升特征的丰富性。
- 提出了LRN层 (ReLU后进行归一化处理)，对局部神经元的活动创建竞争机制，使得其中响应比较大的值变得相对更大，并抑制其他反馈较小的神经元，增强了模型的泛化能力。
- 利用GPU强大的并行计算能力加速网络训练过程，并采用GPU分块训练的方式解决显存对网络规模的限制。
- 数据增强。利用随机裁剪和翻转镜像操作增加训练数据量，降低过拟合。

max pooling: 池化时取最大值



## 3.2 常见网络模型

### ● VGG16



## 3.2 常见网络模型

- GoogleNet (InceptionV4)

googlenet的主要思想就是围绕这两个思路去做的：

- 1.深度，层数更深，文章采用了22层，**googlenet**巧妙的在不同深度处增加了两个**loss**来避免上述提到的梯度消失问题，。
- 2.宽度，增加了多种核1x1，3x3，5x5，在3x3前，5x5前，**max pooling**后分别加上了1x1的卷积核起到了降低**feature map**厚度的作用。以下是**googlenet**用的**inception**可以称之为**inception v1**，如下图所示：

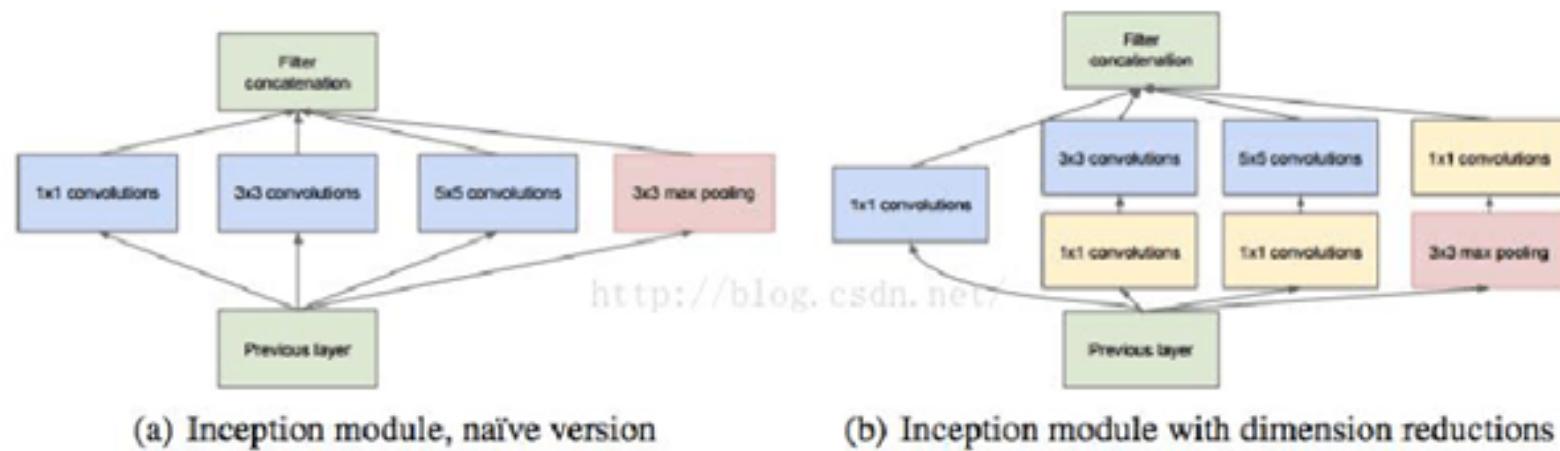


Figure 2: Inception module

## 3.2 常见网络模型



CNN模型比较.doc

### ● 比较

模型名	AlexNet	VGG	GoogLeNet	ResNet
初入江湖	2012	2014	2014	2015
层数	8	19	22	152
Top-5错误	16.4%	7.3%	6.7%	3.57%
Data Augmentation	+	+	+	+
Inception(NIN)	-	-	+	-
卷积层数	5	16	21	151
卷积核大小	11,5,3	3	7,1,3,5	7,1,3,5
全连接层数	3	3	1	1
全连接层大小	4096,4096,1000	4096,4096,1000	1000	1000
Dropout	+	+	+	+
Local Response Normalization	+	-	+	-
Batch Normalization	-	-	-	+

# 目录

- 深度学习概述
- 深度学习常见场景
- 深度学习常用算法介绍
- 深度学习常用框架介绍
- 深度学习未来和展望

## 4.1 开源框架概述

- 深度学习研究的热潮持续高涨，各种开源深度学习框架也层出不穷，其中包括TensorFlow、Caffe、Keras、CNTK、Torch7、MXNet、Leaf、Theano、DeepLearning4、Lasagne、Neon等等。下图是各个开源框架在GitHub上的数据统计（2017年初）。

框 架	机 构	支持语言	Stars	Forks	Contributors
TensorFlow	Google	Python/C++/Go/...	41628	19339	568
Caffe	BVLC	C++/Python	14956	9282	221
Keras	fchollet	Python	10727	3575	322
CNTK	Microsoft	C++	9063	2144	100
MXNet	DMLC	Python/C++/R/...	7393	2745	241
Torch7	Facebook	Lua	6111	1784	113
Theano	U. Montreal	Python	5352	1868	271
Deeplearning4J	DeepLearning4J	Java/Scala	5053	1927	101
Leaf	AutumnAI	Rust	4562	216	14
Lasagne	Lasagne	Python	2749	761	55
Neon	NervanaSystems	Python	2633	573	52

## 4.1 开源框架概述

- Google、Microsoft、Facebook等巨头都参与了这场深度学习框架大战，此外，还有毕业于伯克利大学的贾扬清主导开发的Caffe，蒙特利尔大学Lisa Lab团队开发的Theano，以及其他个人或商业组织贡献的框架。下表是主流深度学习框架在各个维度的评分。

	模型设计	接 口	部 署	性 能	架构设计	总体评分
TensorFlow	80	80	90	90	100	88
Caffe	60	60	90	80	70	72
续表						
	模型设计	接 口	部 署	性 能	架构设计	总体评分
CNTK	50	50	70	100	60	66
Theano	80	70	40	50	50	58
Torch	90	70	60	70	90	76
MXNet	70	100	80	80	90	84
DeepLearning4J	60	70	80	80	70	72

## 4.2 TensorFlow

- TensorFlow最初是由研究人员和Google Brain团队针对机器学习和深度神经网络进行研究所开发的，目前开源之后可以在几乎各种领域适用。
- TensorFlow灵活的架构可以部署在一个或多个CPU、GPU的台式以及服务器中，或者使用单一的API应用在移动设备中。



## 4.3 Caffe

- Caffe由加州大学伯克利的PHD贾扬清开发，全称Convolutional Architecture for Fast Feature Embedding，是一个清晰而高效的开源深度学习框架，目前由伯克利视觉学中心（Berkeley Vision and Learning Center, BVLC）进行维护。（贾扬清曾就职于MSRA、NEC、Google Brain，他也是TensorFlow的作者之一，目前任职于Facebook FAIR实验室。）
- Caffe2脸书 (Facebook) 出品，为生产环境设计，提供在各种平台（包括移动设备）的运行。



## 4.4 Torch

- Torch是一个有大量机器学习算法支持的科学计算框架，其诞生已经有十年之久，但是真正起势得益于Facebook开源了大量Torch的深度学习模块和扩展。Torch另外一个特殊之处是采用了编程语言Lua(该语言曾被用来开发视频游戏)。
- PyTorch是基于Torch的衍生，支持Python语言，实现了机器学习框架 Torch 在 Python 语言环境的执行。



## 4.5 Theano

- 2008年诞生于蒙特利尔理工学院,Theano派生出了大量深度学习Python软件包，最著名的包括Blocks和Keras。Theano的核心是一个数学表达式的编译器，它知道如何获取你的结构。并使之成为一个使用numpy、高效本地库的高效代码，如BLAS和本地代码（C++）在CPU或GPU上尽可能快地运行。它是为深度学习中处理大型神经网络算法所需的计算而专门设计的，是这类库的首创之一（发展始于2007年），被认为是深度学习研究和开发的行业标准。

Theano  
Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. It can generate efficient CPU or GPU code for these expressions.

## 4.6 Deeplearning4j

- Deeplearning4j是“for Java”的深度学习框架，也是首个商用级别的深度学习开源库。Deeplearning4j由创业公司Skymind于2014年6月发布，使用Deeplearning4j的不乏埃森哲、雪弗兰、博斯咨询和IBM等明星企业。DeepLearning4j是一个面向生产环境和商业应用的高成熟度深度学习开源库，可与Hadoop和Spark集成，即插即用，方便开发者在APP中快速集成深度学习功能。

## 4.7 MXNet

- 出自CXXNet、Minerva、Purine 等项目的开发者之手，主要用C++ 编写。  
MXNet 强调提高内存使用的效率，甚至能在智能手机上运行诸如图像识别等任务。



## 4.8 CNTK

- CNTK (Computational Network Toolkit) 是微软研究院 (MSR) 开源的深度学习框架。它最早由 start the deep learning craze 的演讲人创建，目前已经发展成一个通用的、跨平台的深度学习系统，在语音识别领域的使用尤其广泛。



# 目录

- 深度学习概述
- 深度学习常见场景
- 深度学习常用算法介绍
- 深度学习常用框架介绍
- 深度学习未来和展望

# 5 未来与展望

- 无监督半监督学习所占比重会越来越大。
- 用于学习的硬件设备会越来越强大，效率越来越高。
- 有向移动端转移的趋势。
- 可视化开发，开发框架会越来越普遍，入门更加简单。
- 小数据样本的训练所占比重会增大。