

1 基本概念

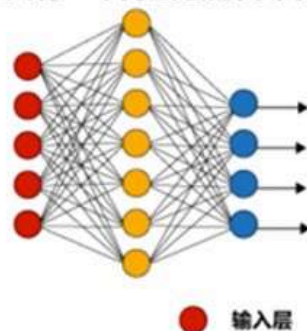
- 深度学习 (Deep Learning, DL) : 通过组合低层特征形成更加抽象的高层特征或类别, 从而从大量的输入数据中学习有效特征表示, 并把这些特征用于分类、回归和信息检索的一种技术。
- 人工神经网络: 是一种模仿动物神经网络行为特征, 进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度, 通过调整内部大量节点之间相互连接的关系, 从而达到处理信息的目的。
- 深度神经网络: 包含多个隐含层的神经网络。
- 模型: 可以理解成算法和数据的合集。
- 模型训练: 利用模型的算法, 使用深度神经网络进行权值的训练, 最终得出一个最优解。
- 模型预测: 使用训练完成的模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

1. 基本概念

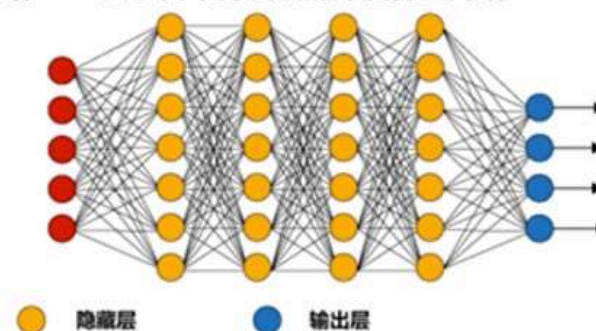
- 人工神经网络：是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。
- 深度神经网络：包含多个隐含层的神经网络。

神经网络和深度学习

只有一个隐藏层的简单神经网络

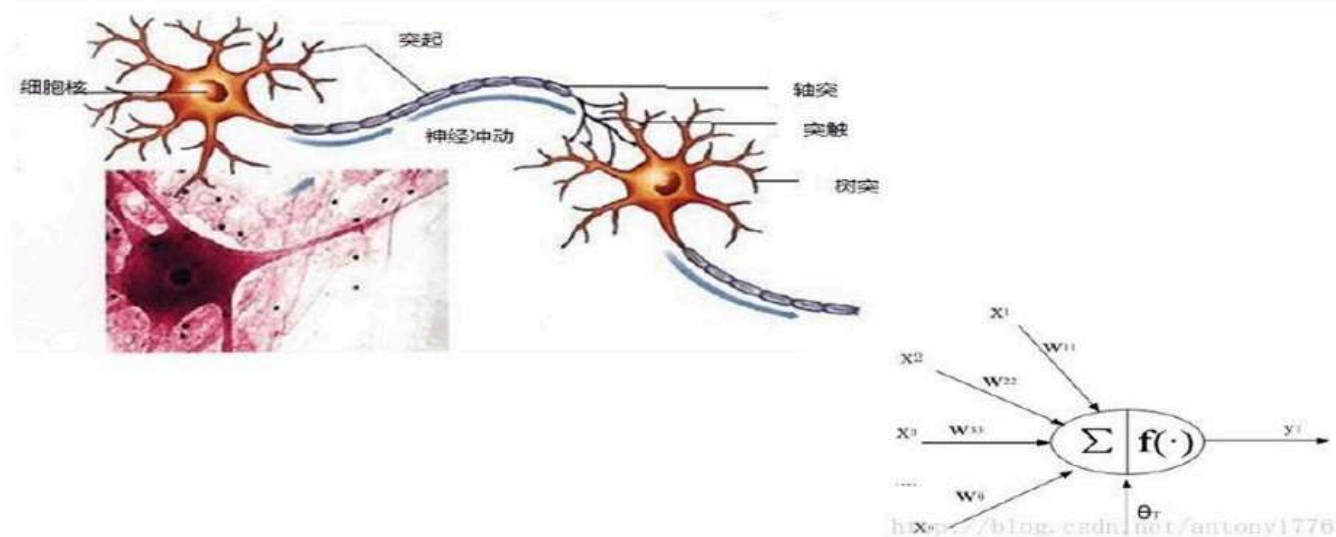


大于两个隐藏层的深度神经网络



3.1 人工神经网络 (ANN)

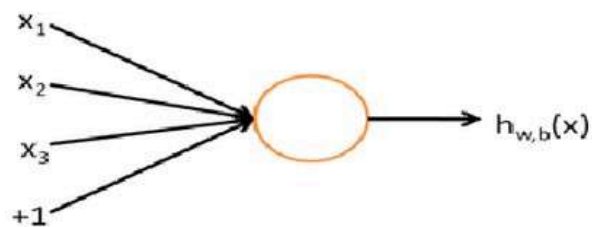
- 人工神经网络 (Artificial Neural Networks) 是一种模仿生物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点（神经元）之间相互连接的权重，从而达到处理信息的目的。



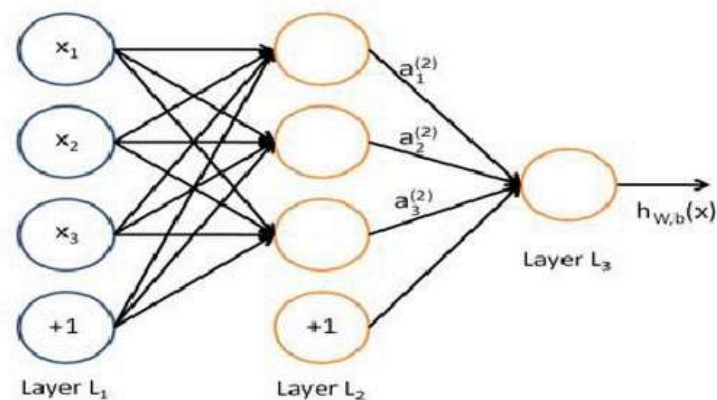
14

3.1 人工神经网络 (ANN)

- 神经网络



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



$$a_1^{(2)} = f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)})$$

$$a_2^{(2)} = f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)})$$

$$a_3^{(2)} = f(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)})$$

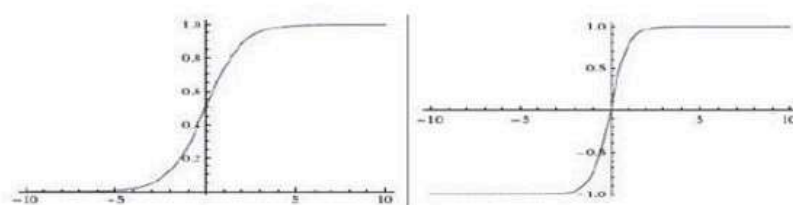
$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)} + b_1^{(2)})$$

3.1 人工神经网络 (ANN)

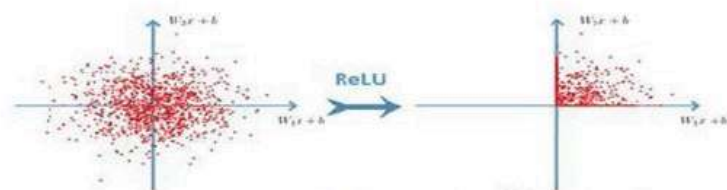
- 人工神经网络的重要概念:

- 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:

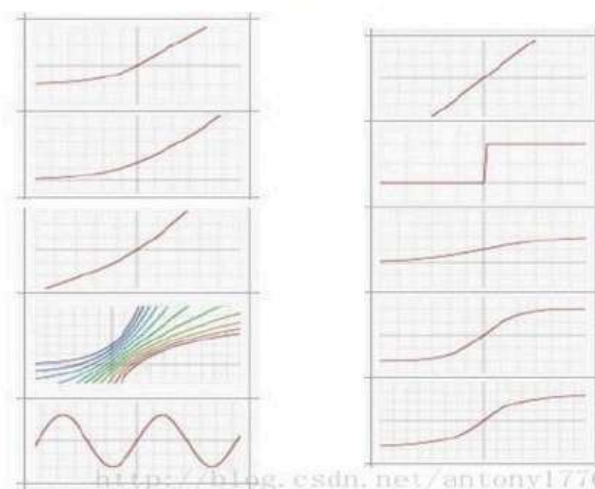


Sigmoid



Relu

Others

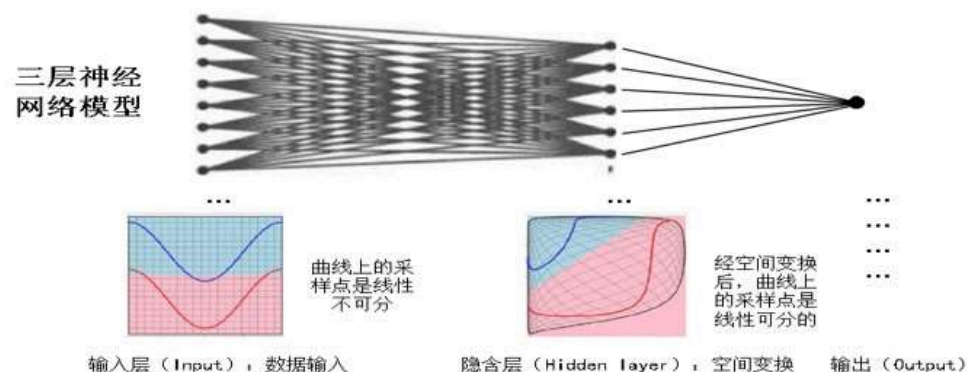


3.1 人工神经网络 (ANN)

- 人工神经网络的重要概念:

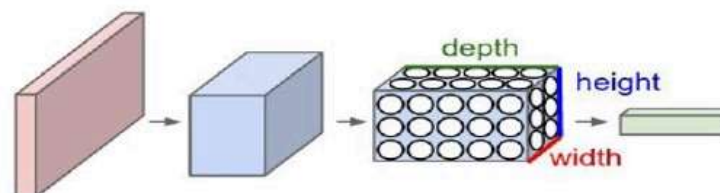
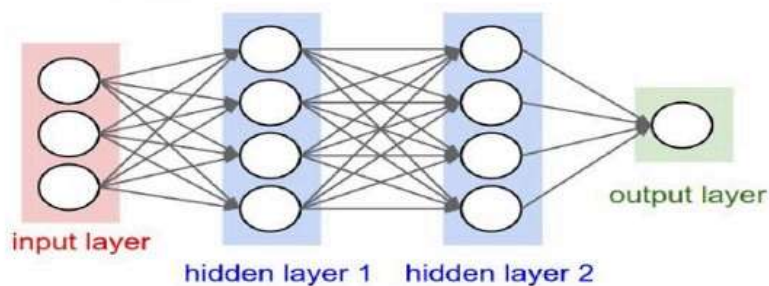
激励函数很重要，无论是对建立神经网络的模型，还是理解神经网络。首先要了解，它有以下几个影响：

- 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！



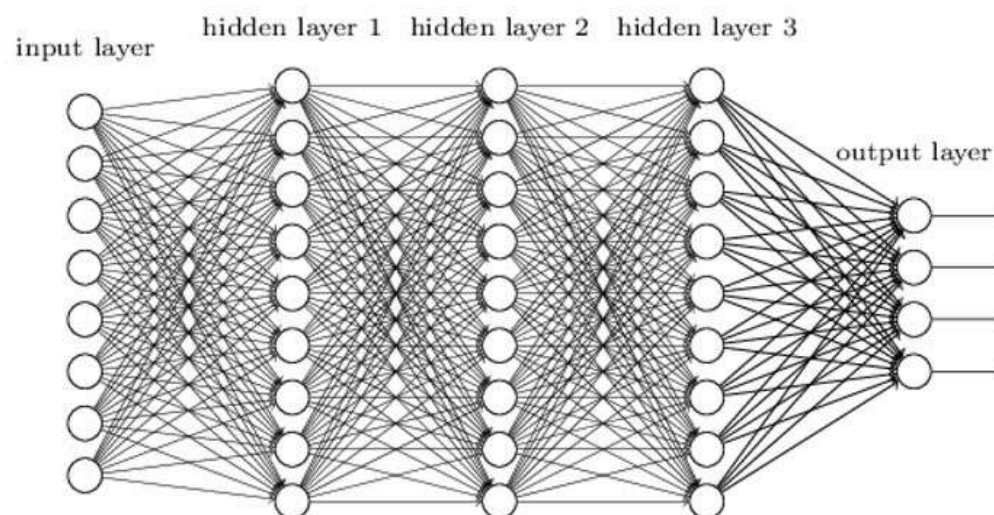
3.1 卷积神经网络（ CNN ）

- 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

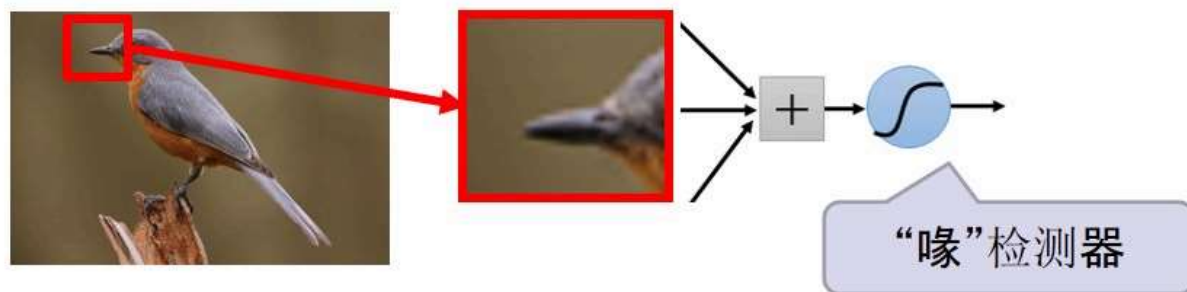
- 我们知道学习规模小更好。
- 从这个完全连接的模型中，我们真的需要所有的连接吗？
- 可以共享其中一些吗？



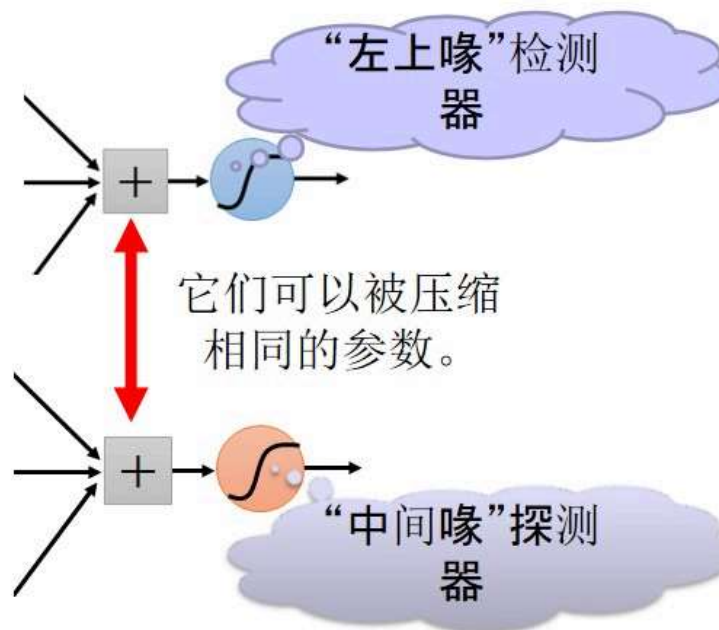
考虑学习图像：

- 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

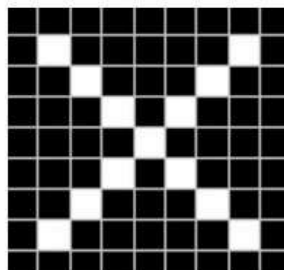


相同的模式出现在不同的位置：它们可以被压缩！
训练很多这样的“小型”探测器又如何呢？每个探测器必须“四处移动”。

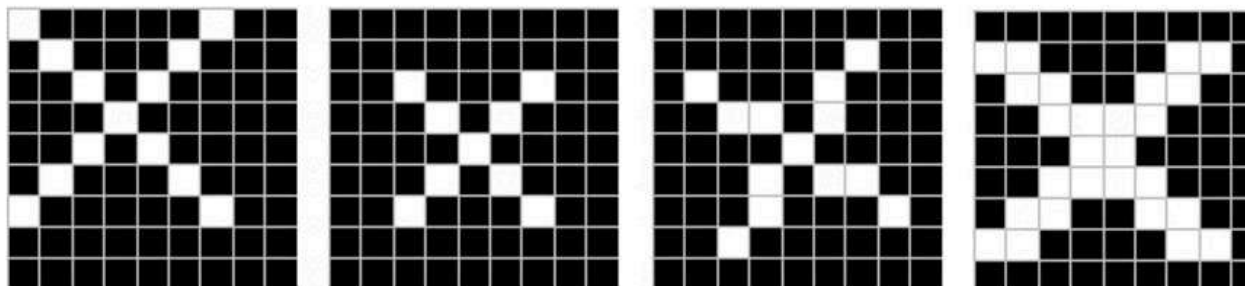




比如，我现在要训练一个最简单的**CNN**，用来识别一张图片里的字母是**X**还是**O**。

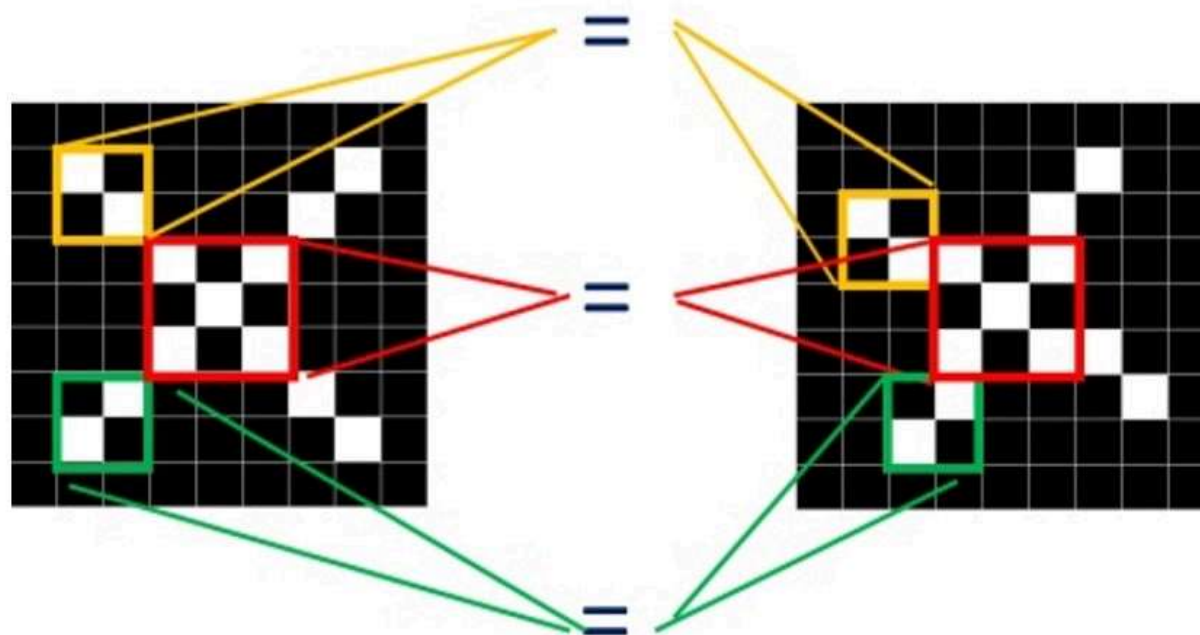


以下四个都是**X**，但它们和之前那张**X**明显不一样。（“欠拟合”）





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。

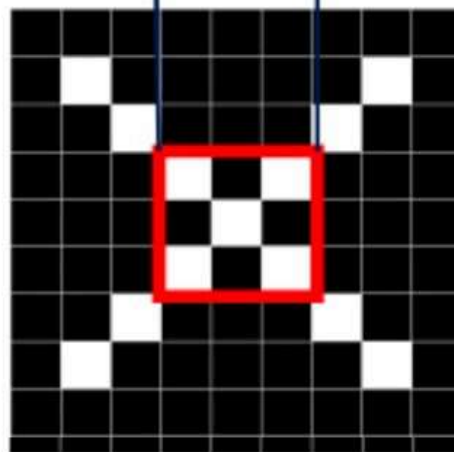


从标准的X图中我们提取出三个特征（feature）

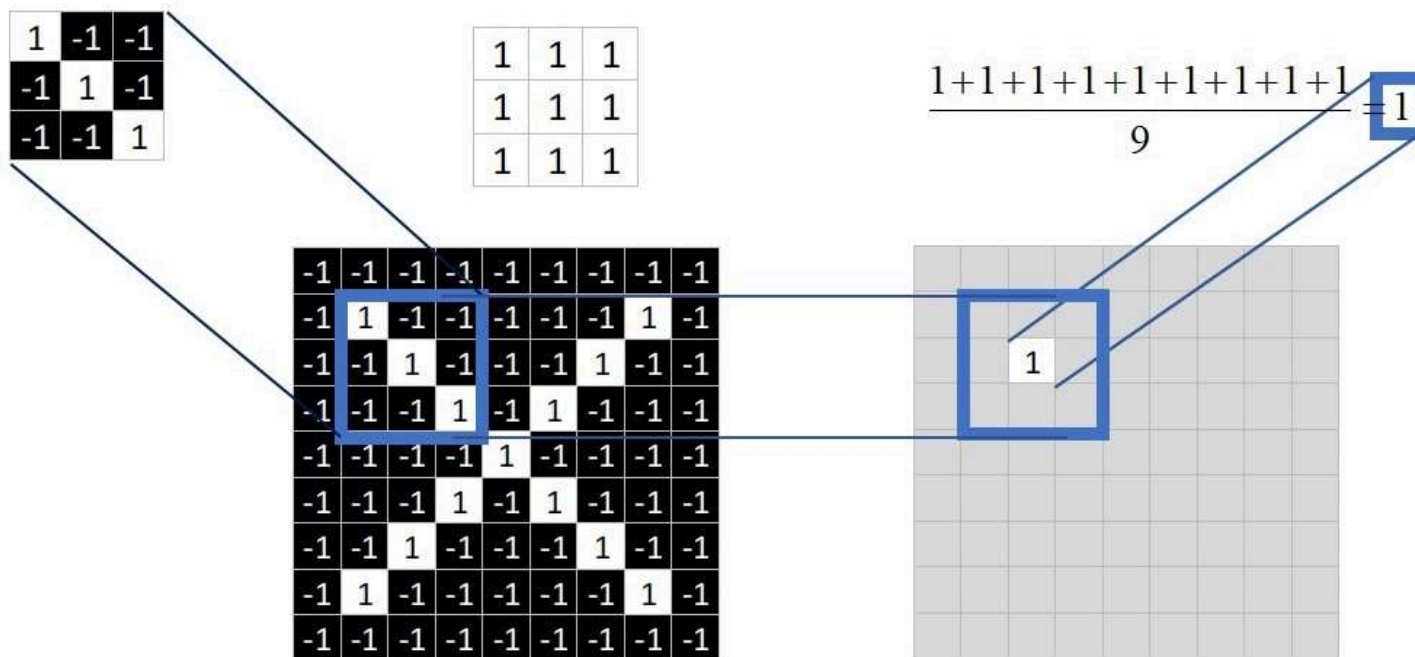
1	-1	-1
-1	1	-1
-1	-1	1

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

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

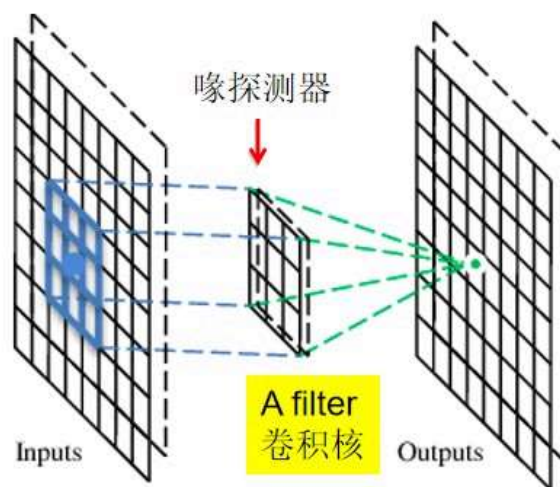


feature在CNN中也被成为卷积核（filter）



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

这些是要学习的网络参数。

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 黑白图片

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
小图案 (3 x 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

6 x 6 图片

点积

3

-1

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

卷积核 1

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

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 图片

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

卷积核 1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

卷积

步长=1

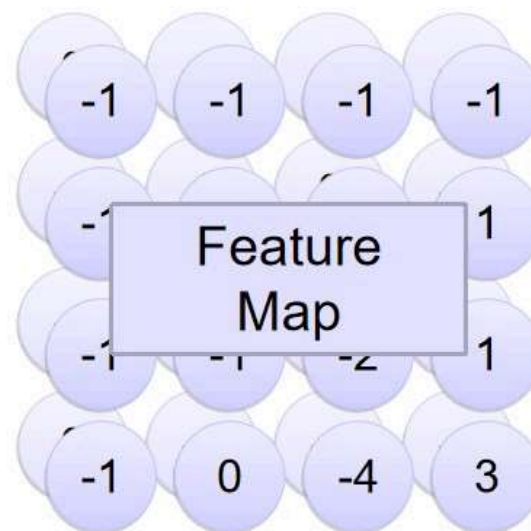
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 图片

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

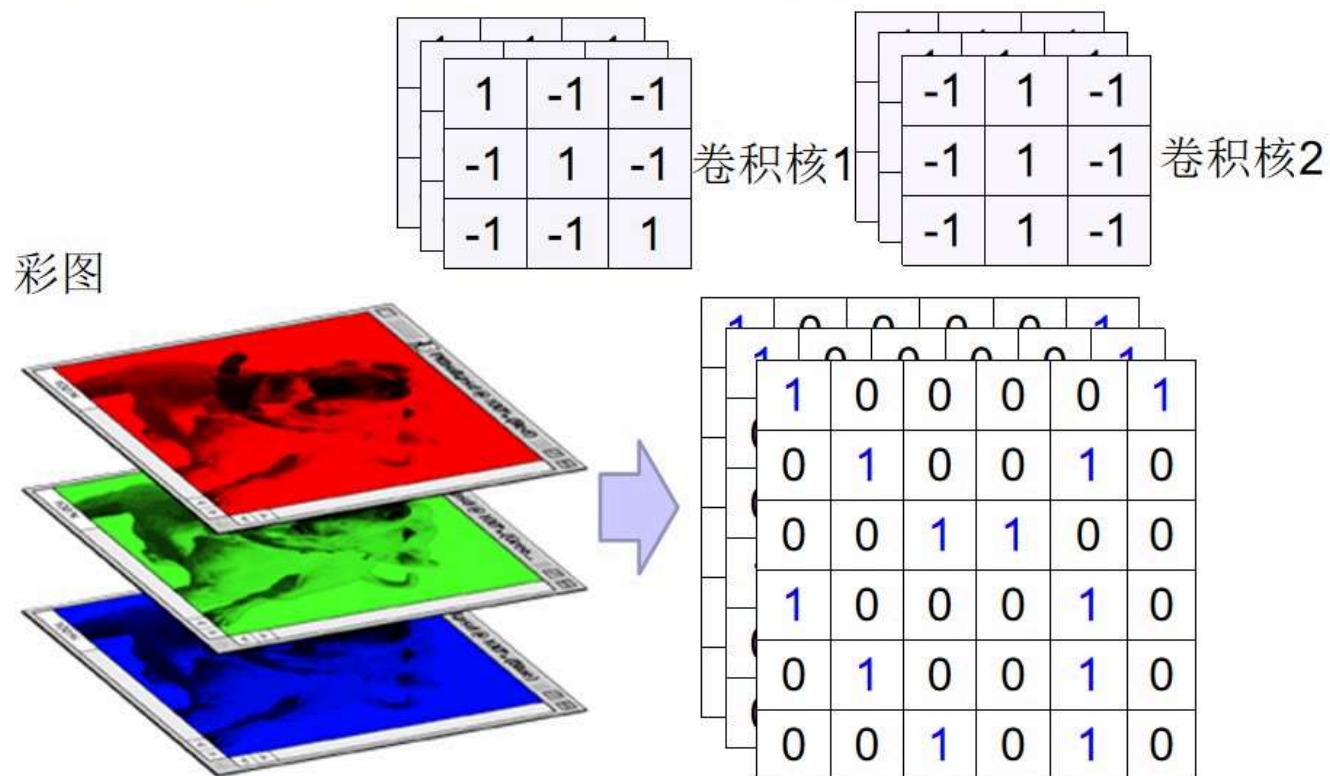
卷积核 2

对每个卷积核重复此操作

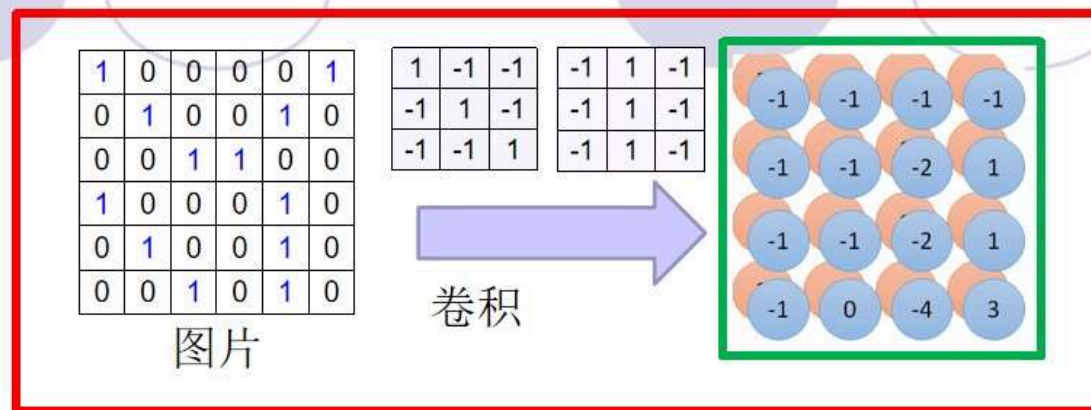


Two 4 x 4 images
Forming 2 x 4 x 4 matrix

彩色图像: RGB 3通道

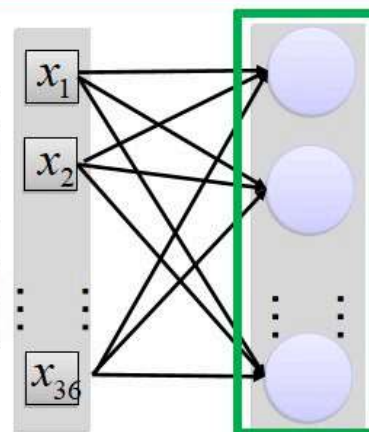


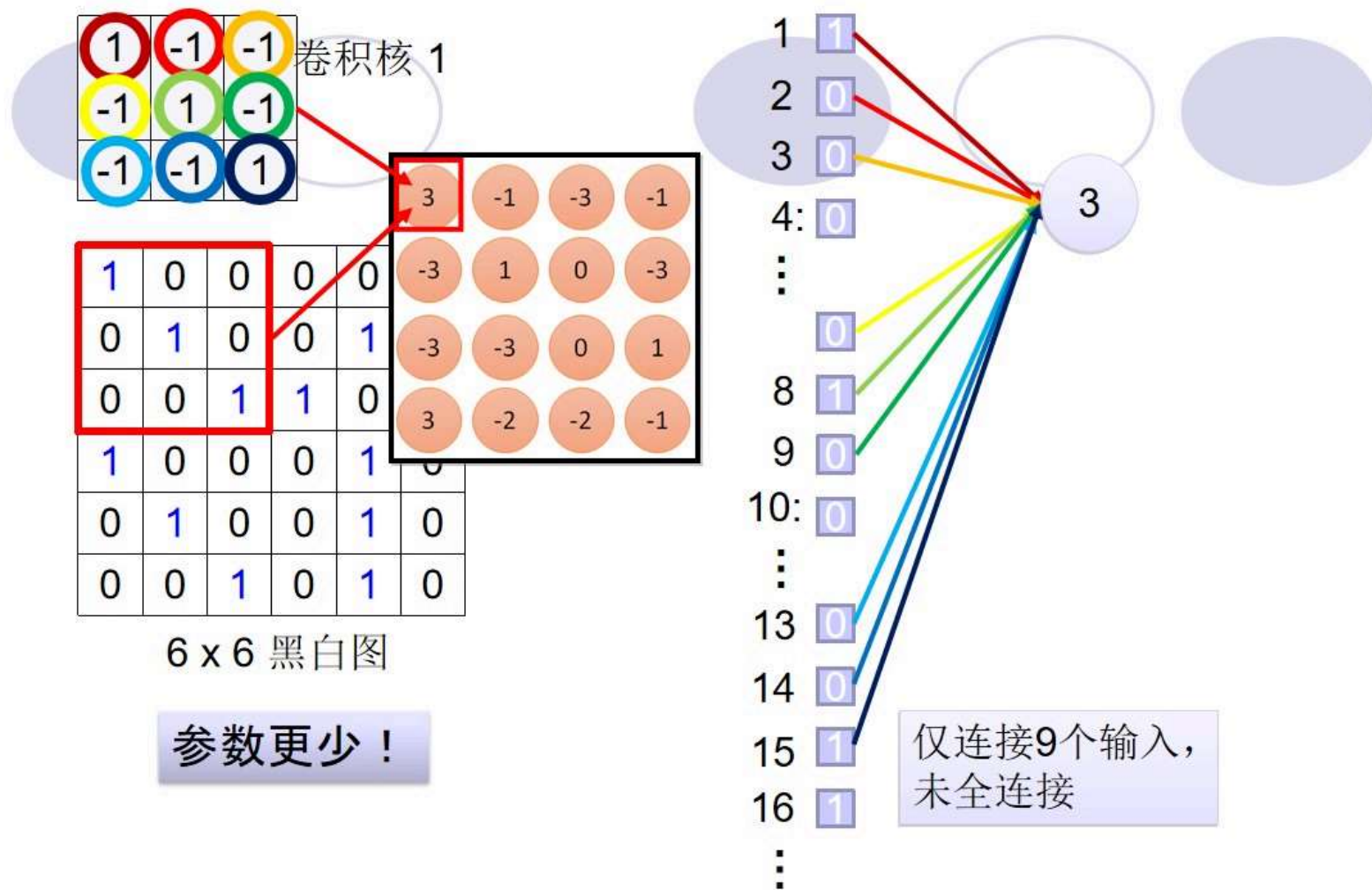
卷积 v.s. 全连接

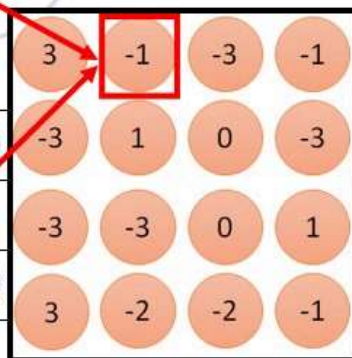
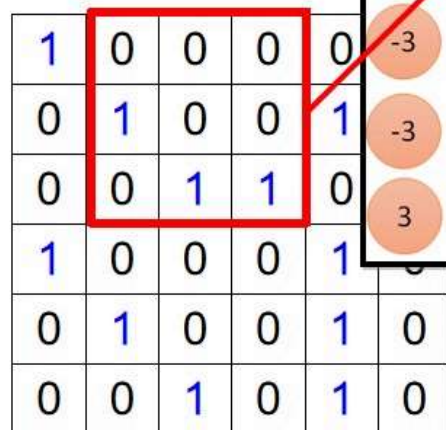


全连接

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

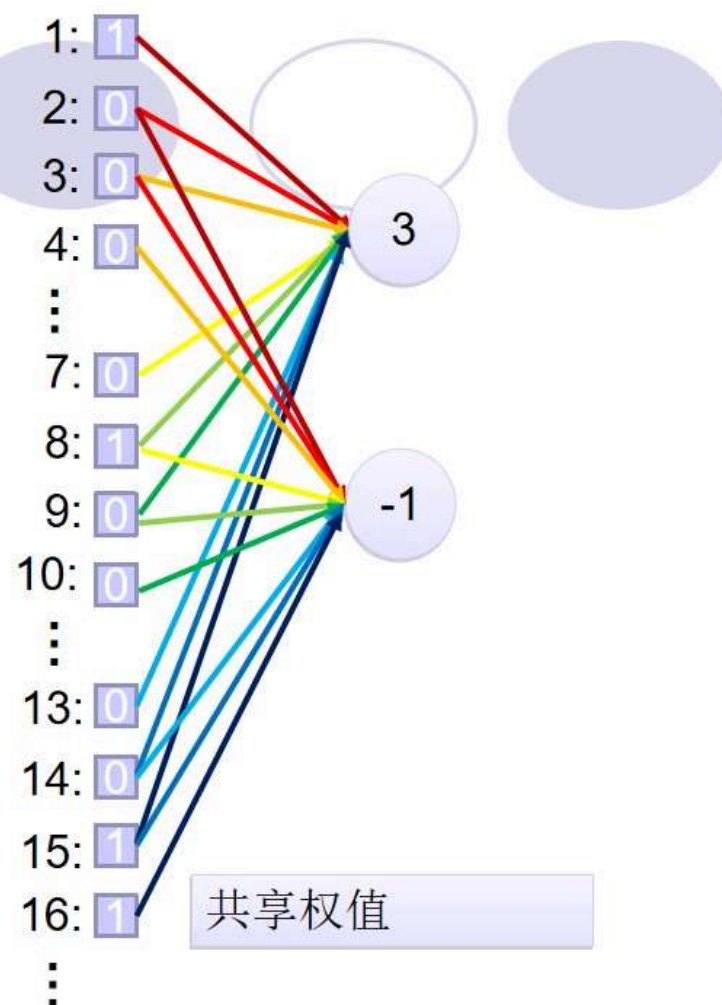






参数更少

参数进一步减少！



单选题 1分

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。

☐ A 错

☒ B 对

单选题 1分

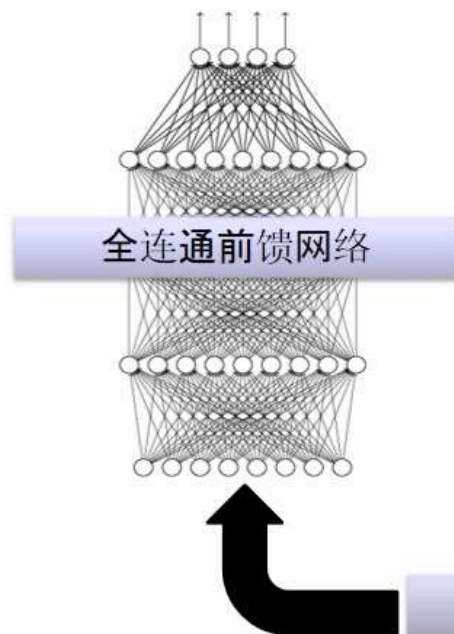
卷积是一种有效提取图片特征的方法。一般用一个正方形卷积核，遍历图片上的每一个像素点。图片与卷积核重合区域内相对应的每一个像素值乘卷积核内相对应点的权重，然后求和，再加上偏置后，最后得到输出图片中的一个像素值。

A 错

B 对

完整 CNN

cat dog



卷积

池化

卷积

池化

可重复多次

扁平化

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

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

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。

30

单选题 1分

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

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

更小的图

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 图片

卷积

池化

-1

1

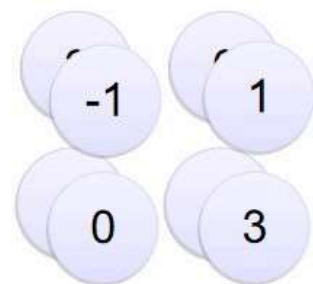
0

3

2 x 2 图片

每个卷积核
是一个通路

完整 CNN



新图

更小的图

通道数是卷积核数



Convolution

Max Pooling

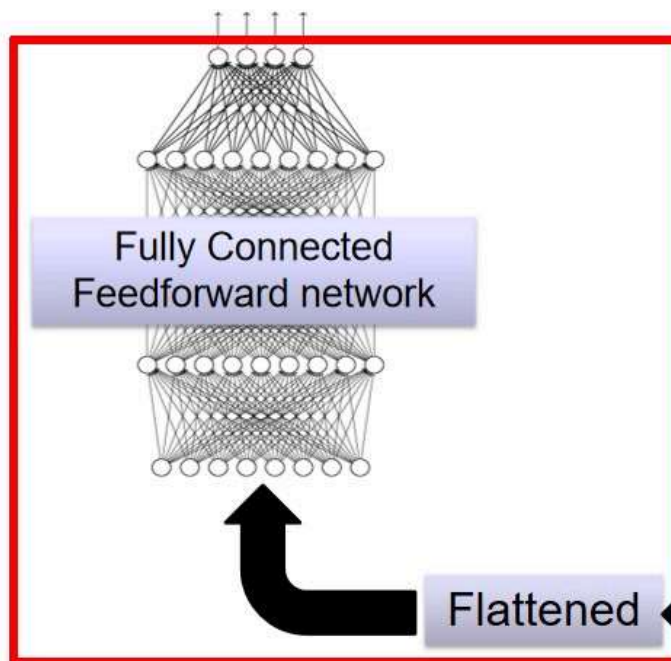
Convolution

Max Pooling

可重复

The whole CNN

cat dog



Convolution

Max Pooling

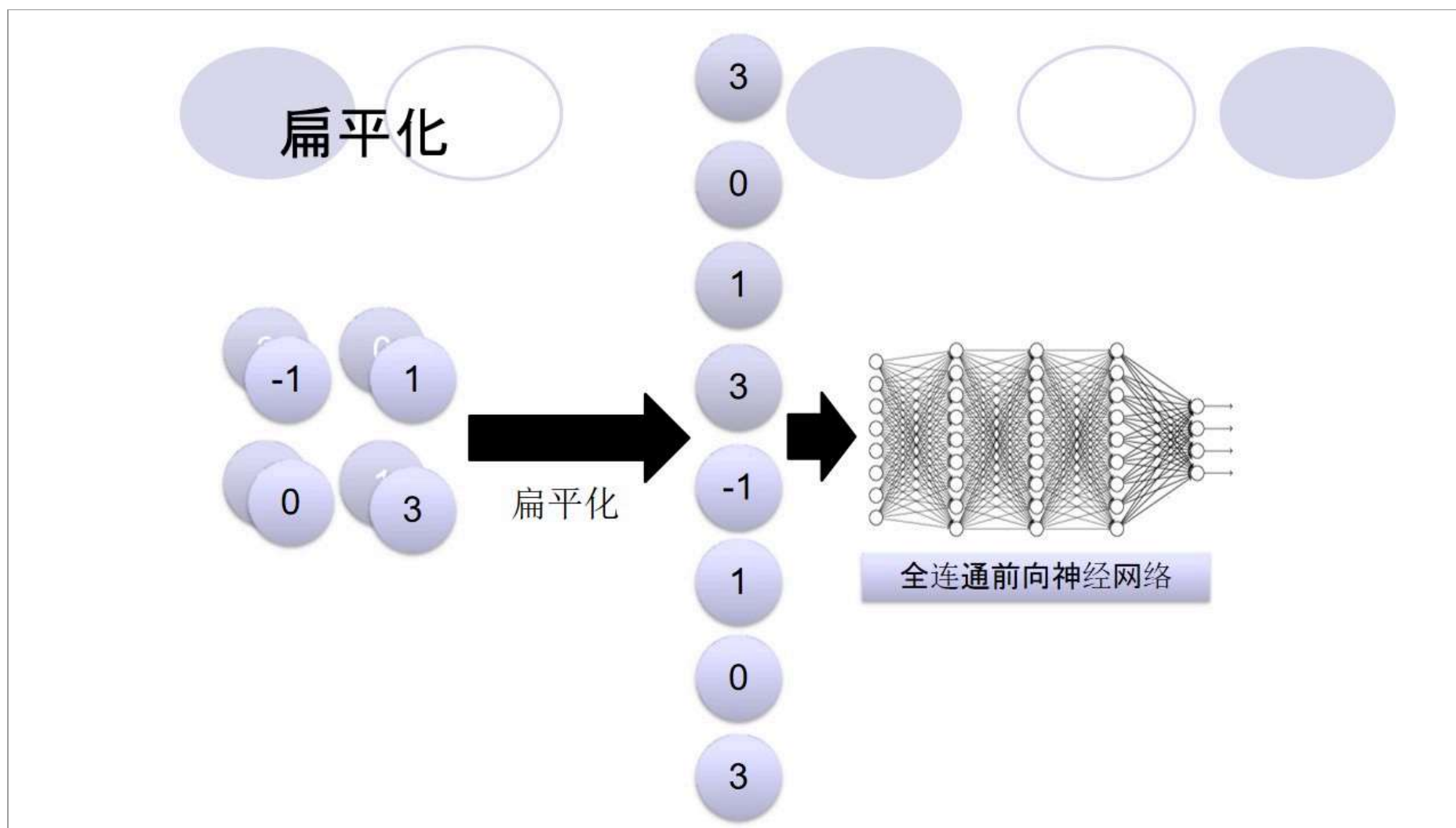
A new image

Convolution

Max Pooling

A new image

Flattened





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

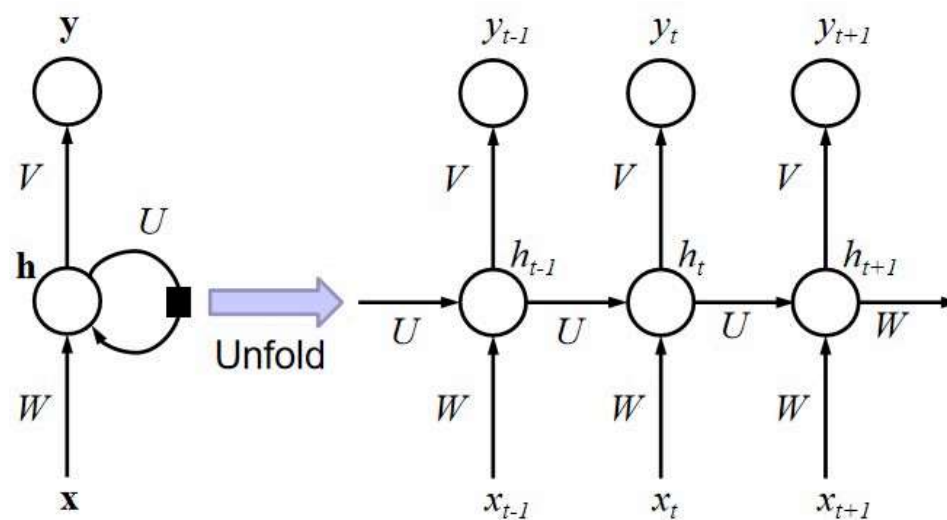
卷积神经网络视频

- 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×tamp=1634889046&unique_k=Yxkf1s 33:09
- https://www.bilibili.com/video/BV1qb411P7JD?spm_id_from=333.999.0.0 15:24
- https://www.bilibili.com/video/BV1R5411w715?from=search&seid=14451113221349708057&spm_id_from=333.337.0.0 8:32

三. 循环神经网络

2. 循环神经网络的模型结构

循环神经网络（Recurrent Neural Network, RNN）是一种对序列数据建模的神经网络，即一个序列当前的输出与前面的输出也有关。其吸收了HMM模型的有限序列关联的思想。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不在无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。RNN模型的连接如图所示。



RNN模型结构图

三. 循环神经网络

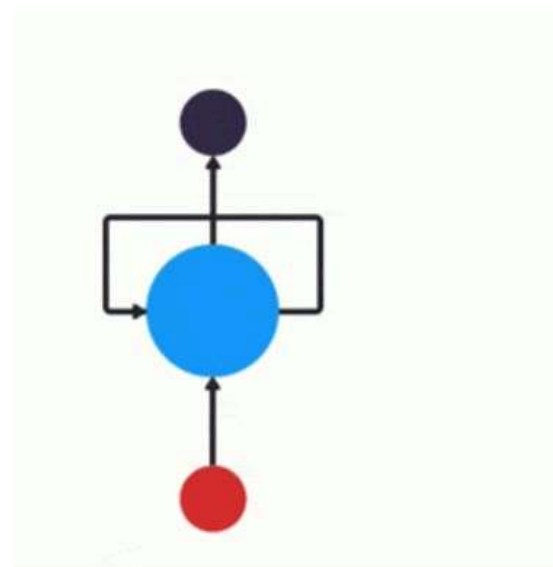
与传统神经网络的区别：

传统神经网络(包括CNN)，输入和输出都是互相独立的。如图像上的猫和狗是分隔开的，但有些任务，后续的输出和之前的内容是相关的。

RNN引入“记忆”的概念，也就是输出需要依赖之前的输入序列，并把关键输入记住。循环2字来源于其每个元素都执行相同的任务。它并非刚性地记忆所有固定长度的序列，而是通过隐藏状态来存储之前时间步的信息。

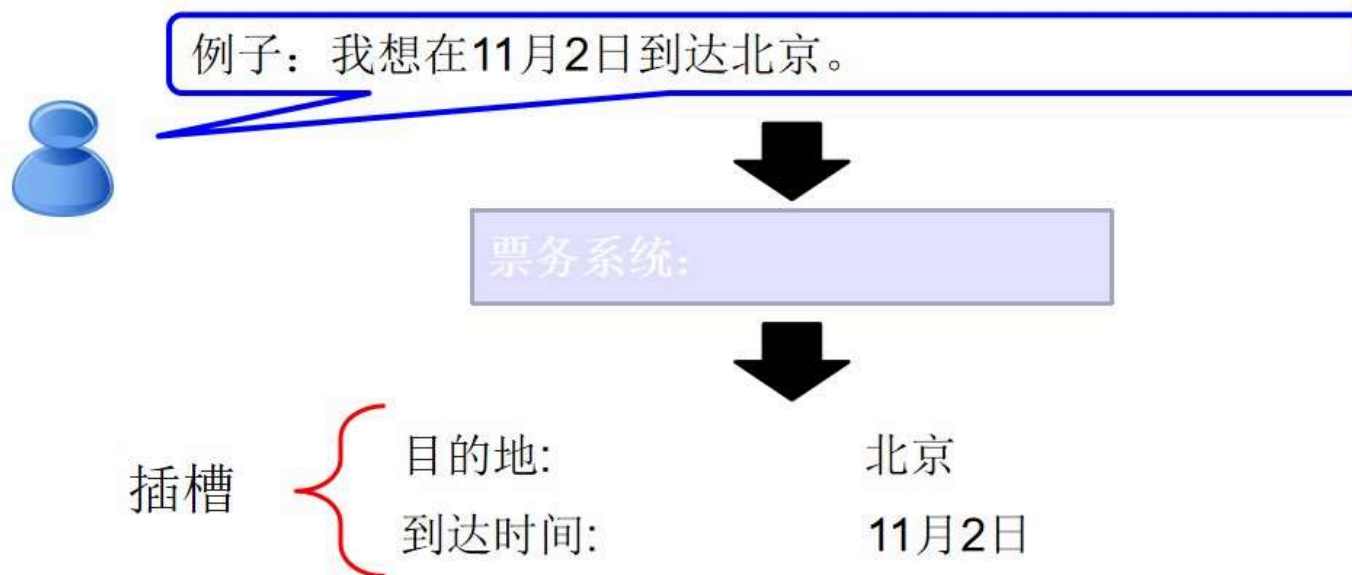
RNN跟传统神经网络最大的区别在于每次都会将前一次的输出结果，带到下一次的隐藏层中，一起训练。如右图所示：

2. 循环神经网络的模型结构



循环神经网络(RNN)的应用示例

- 插槽填充Slot Filling

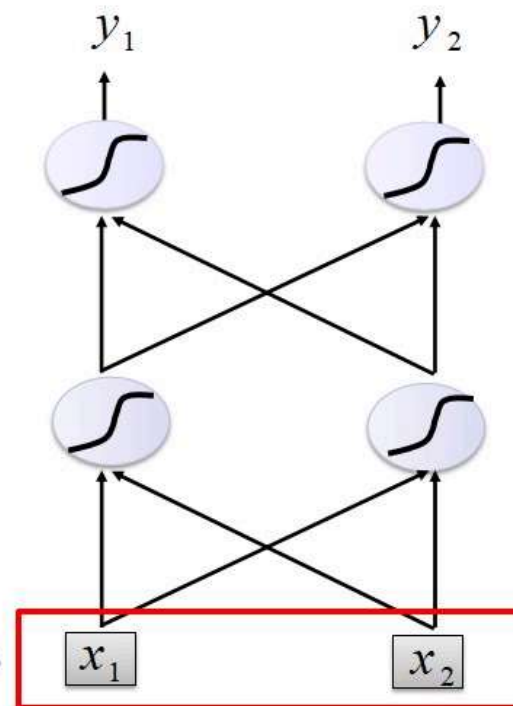


应用示例

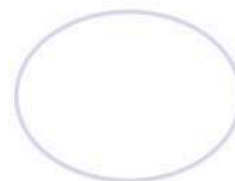
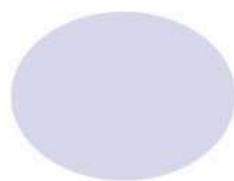
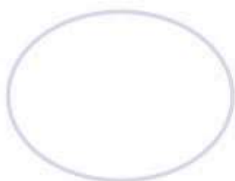
插槽填充的解决方法？

输入：一个词
(每个词都表示为一个向量)

北京



1-of-N编码



如何将每个单词表示为一个向量?

1-of-N编码

例子: 词汇表 = {苹果, 包, 猫, 狗, 大象}

该向量是词典大小。

每个维度对应于词汇表中的一个词

目标词维度为1, 其它为0

苹果 = [1 0 0 0 0]

包 = [0 1 0 0 0]

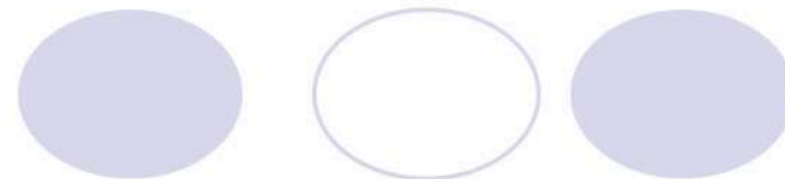
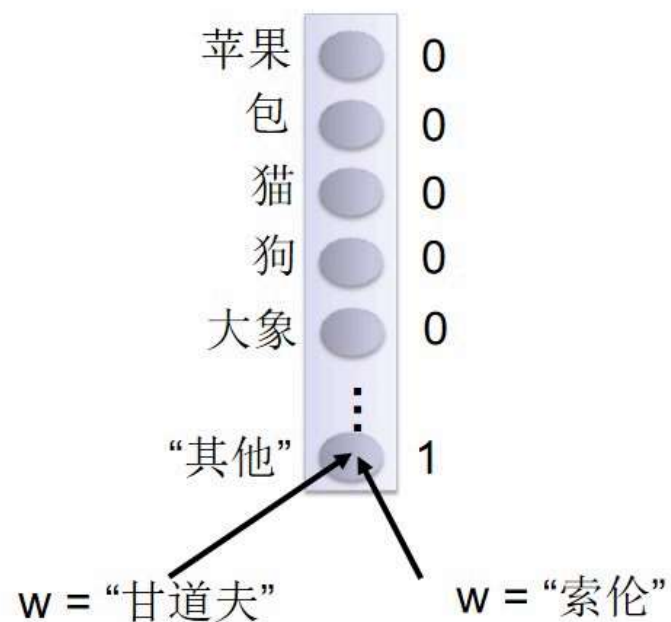
猫 = [0 0 1 0 0]

狗 = [0 0 0 1 0]

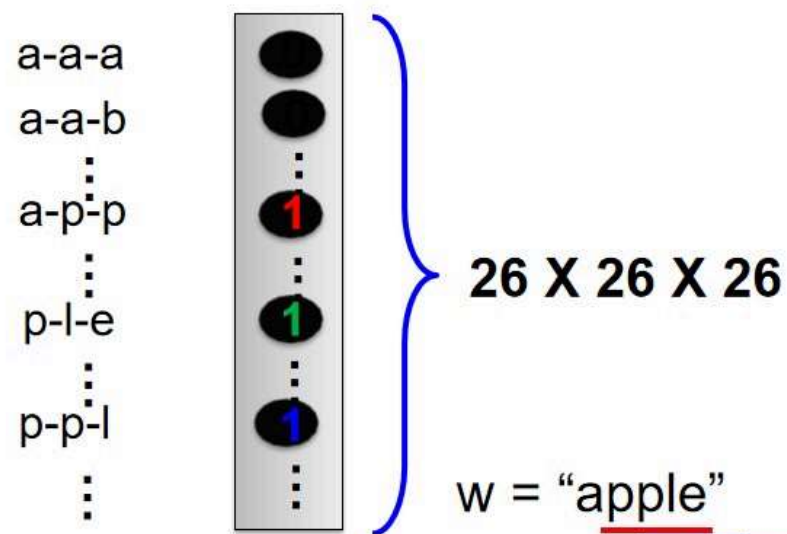
大象 = [0 0 0 0 1]

不在 1-of-N 词汇表的词

“其他”维度



Word hashing 词哈希



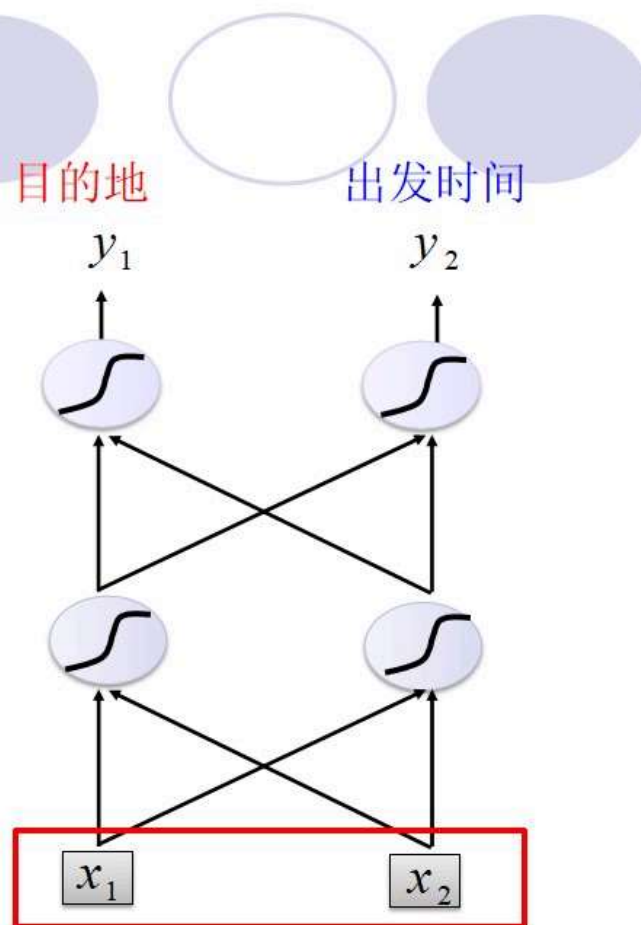
应用示例

通过前馈网络解决插槽填充？

输入：一个词
(每个词都表示为一个向量)

输出：
输入词属于某插槽的概率分布

北京



应用示例

arrive **Beijing** on November 2nd

other **dest** other time time

问题?

leave **Beijing** on November 2nd

place of departure

神经网络需要记忆!

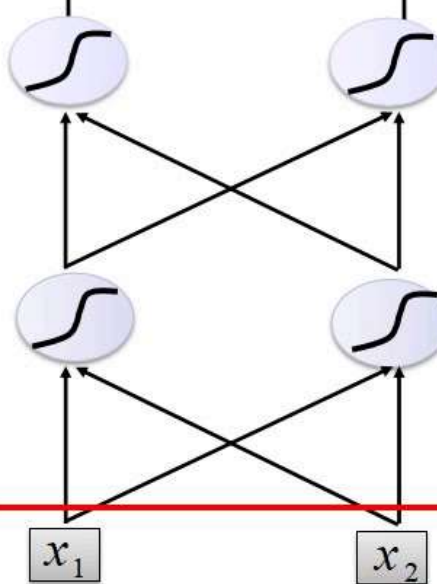
Taipei

目的地

出发时间

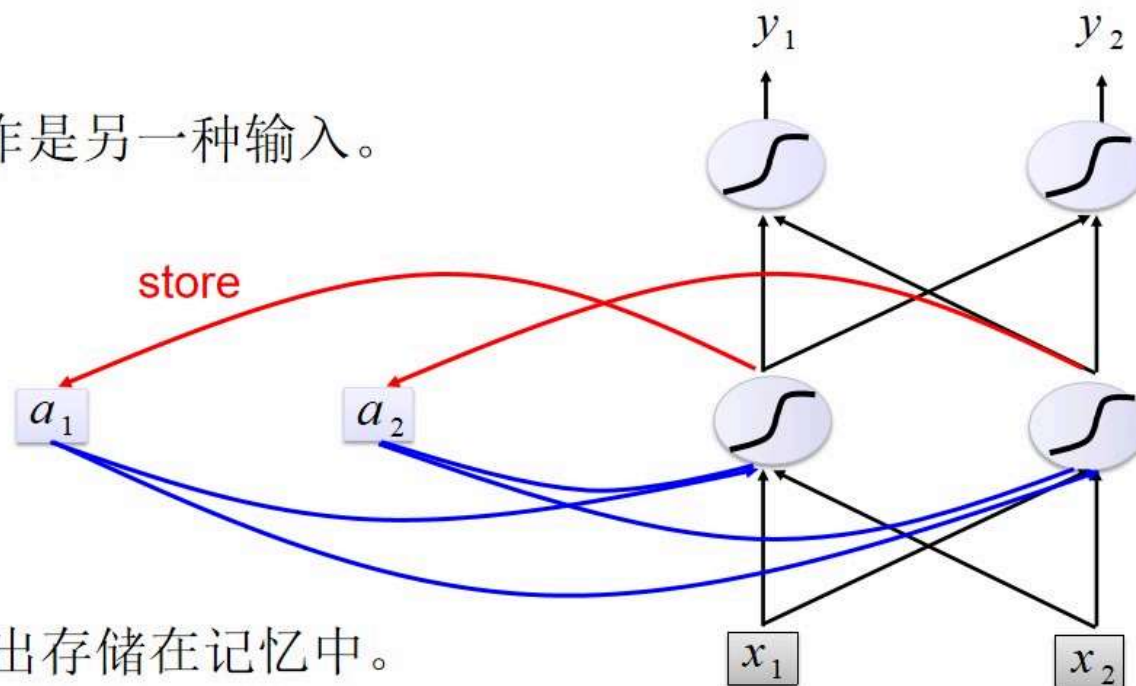
y_1

y_2



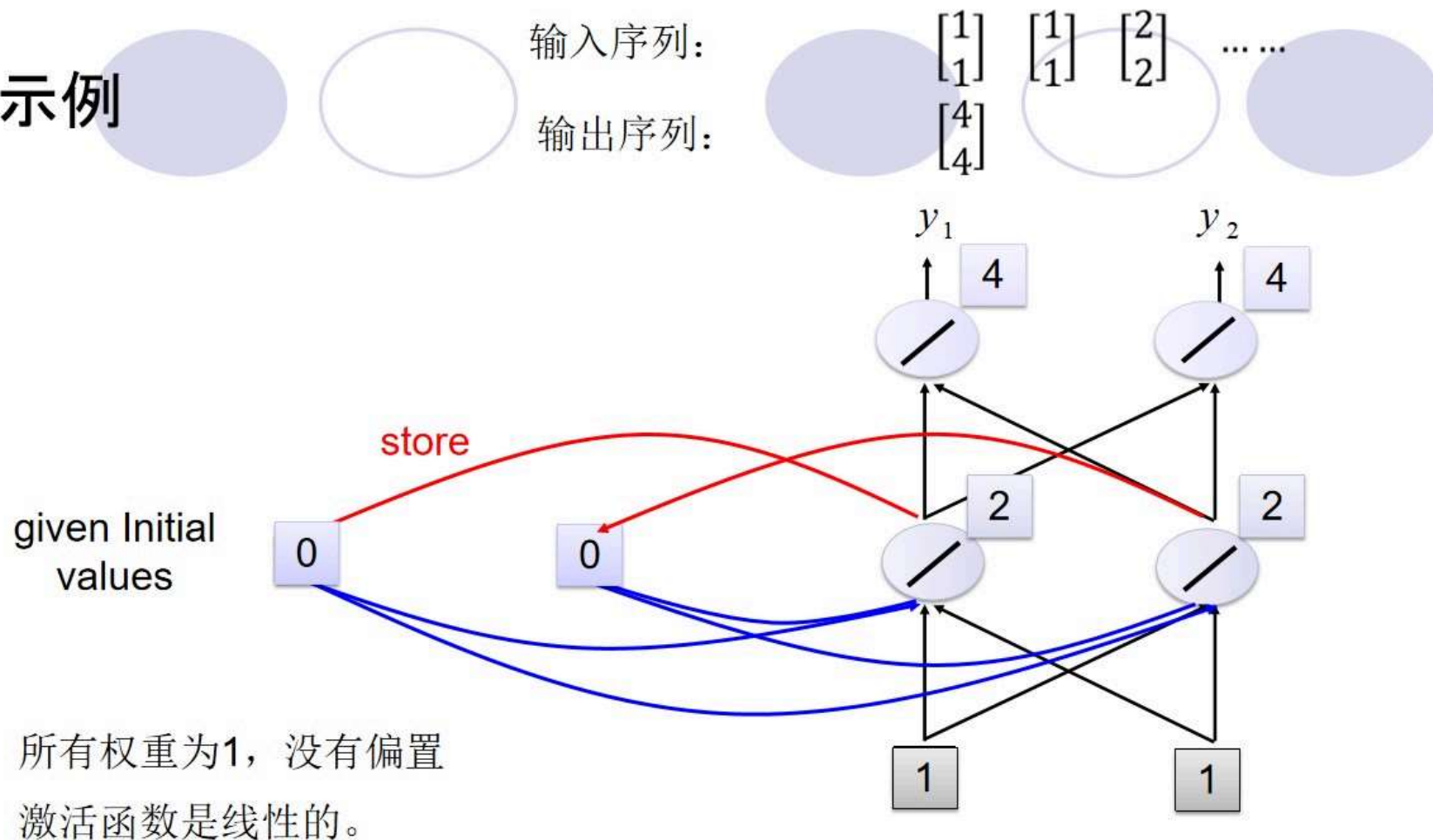
循环神经网络(RNN)

记忆可以看作是另一种输入。



隐藏层的输出存储在记忆中。

示例



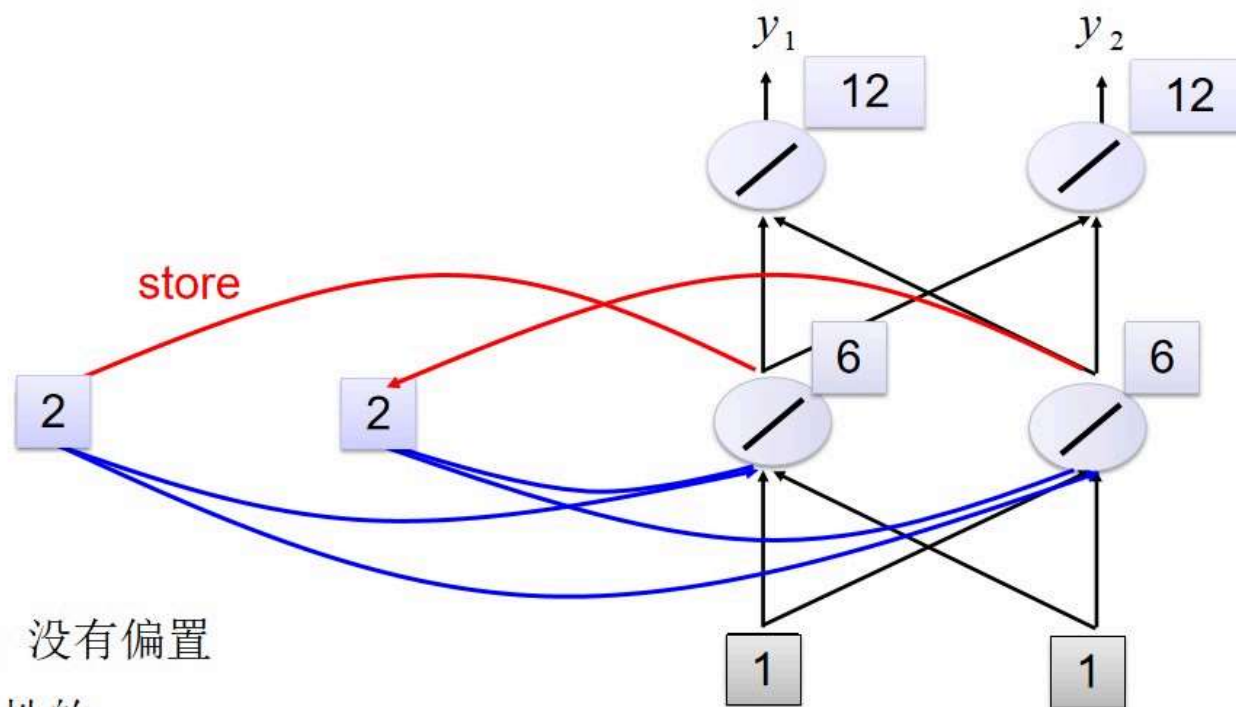
示例

输入序列:

输出序列

$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$...

$\begin{bmatrix} 4 \\ 4 \end{bmatrix}$ $\begin{bmatrix} 12 \\ 12 \end{bmatrix}$



所有权重为1，没有偏置
激活函数是线性的。

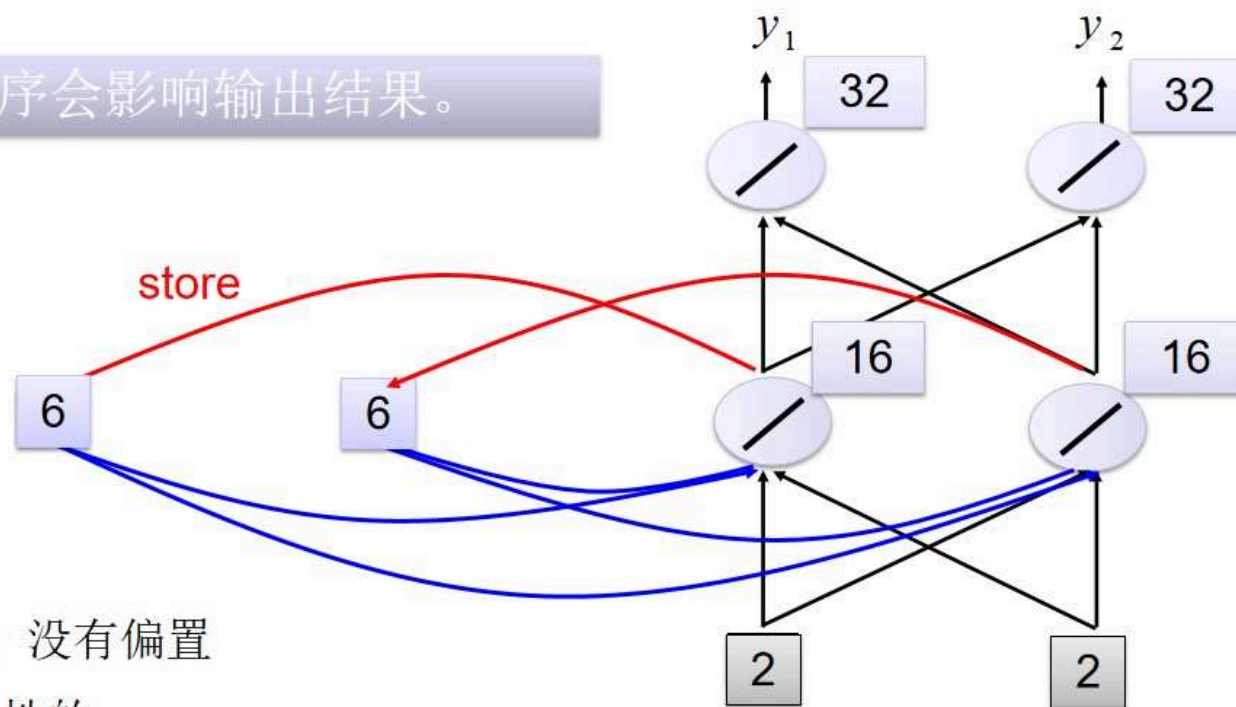
示例

输入序列:

输出序列:

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \dots$$
$$\begin{bmatrix} 4 \\ 4 \end{bmatrix} \begin{bmatrix} 12 \\ 12 \end{bmatrix} \begin{bmatrix} 32 \\ 32 \end{bmatrix}$$

更改序列顺序会影响输出结果。



所有权重为1，没有偏置
激活函数是线性的。

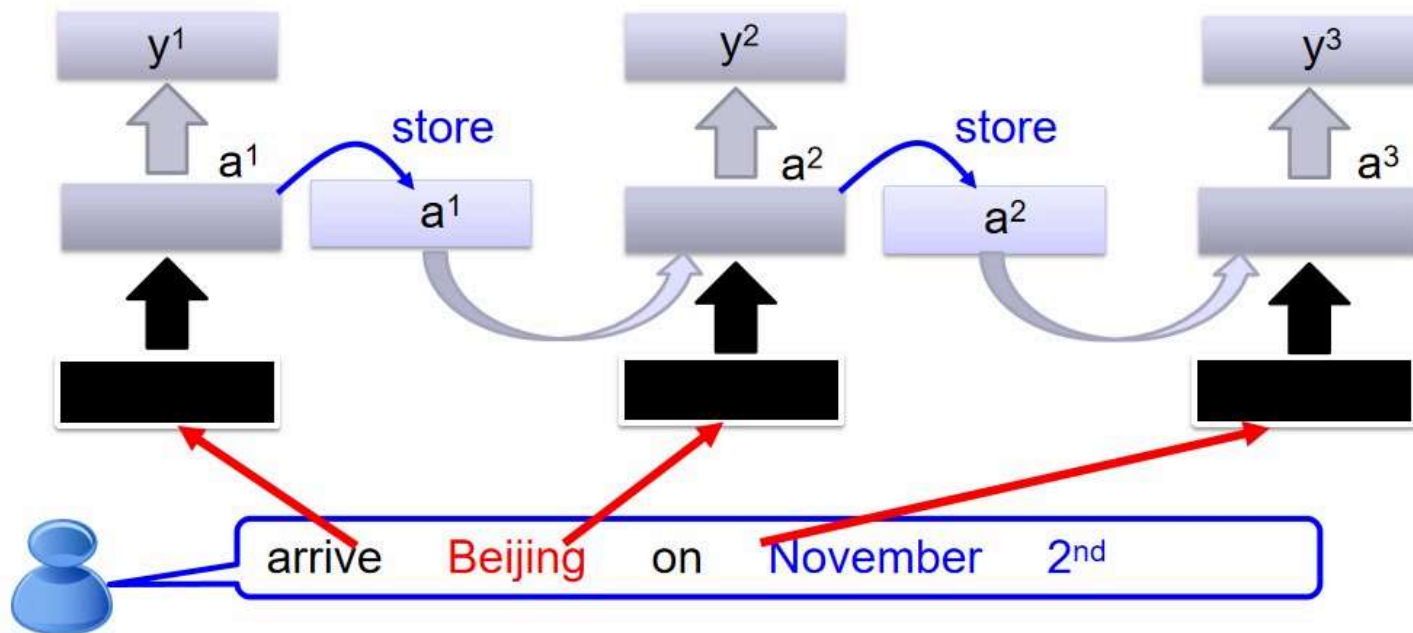
RNN

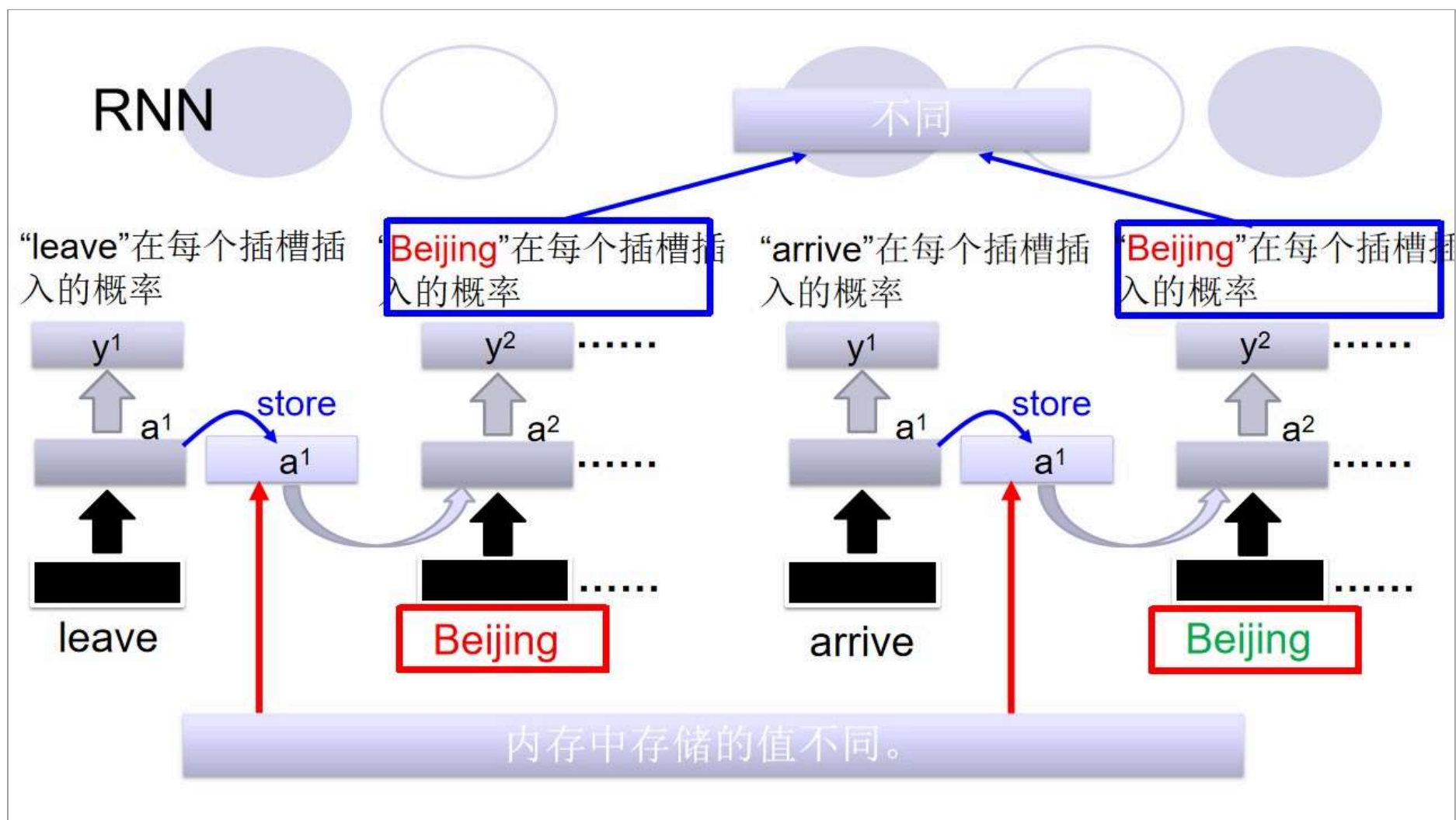
同一个网络不断重复使用。

“arrive” 在每个插槽插入的概率

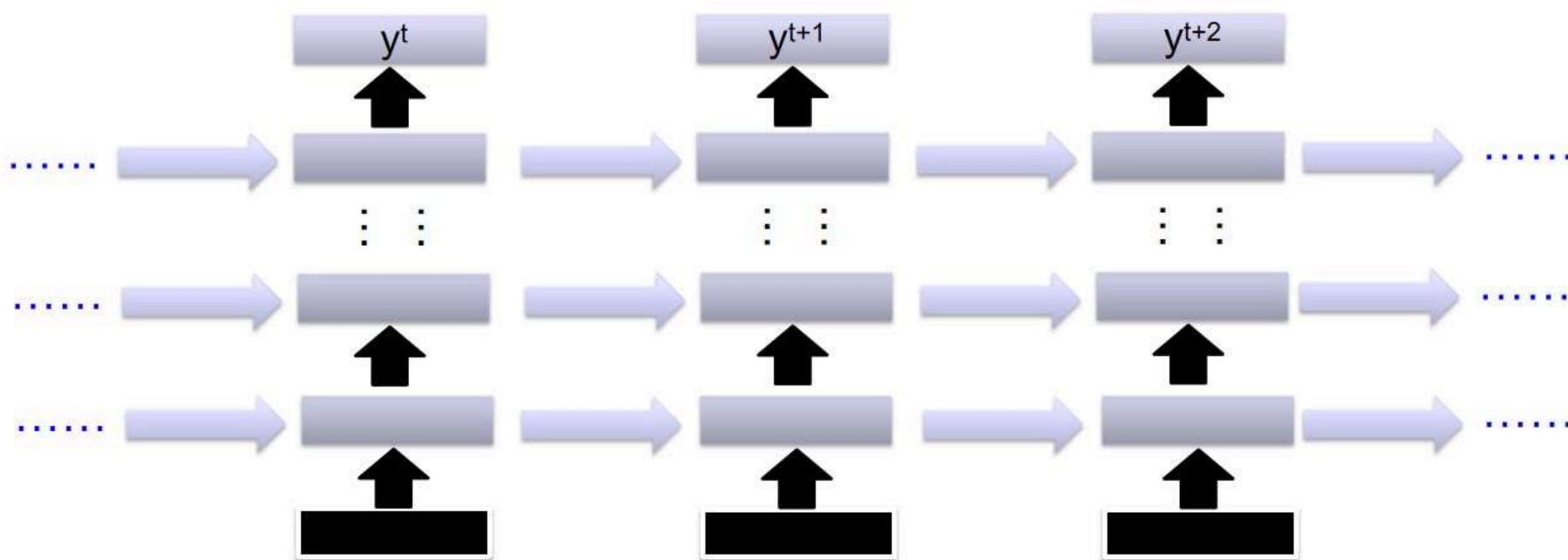
“Beijing” 在每个插槽插入的概率

“on” 在每个插槽插入的概率

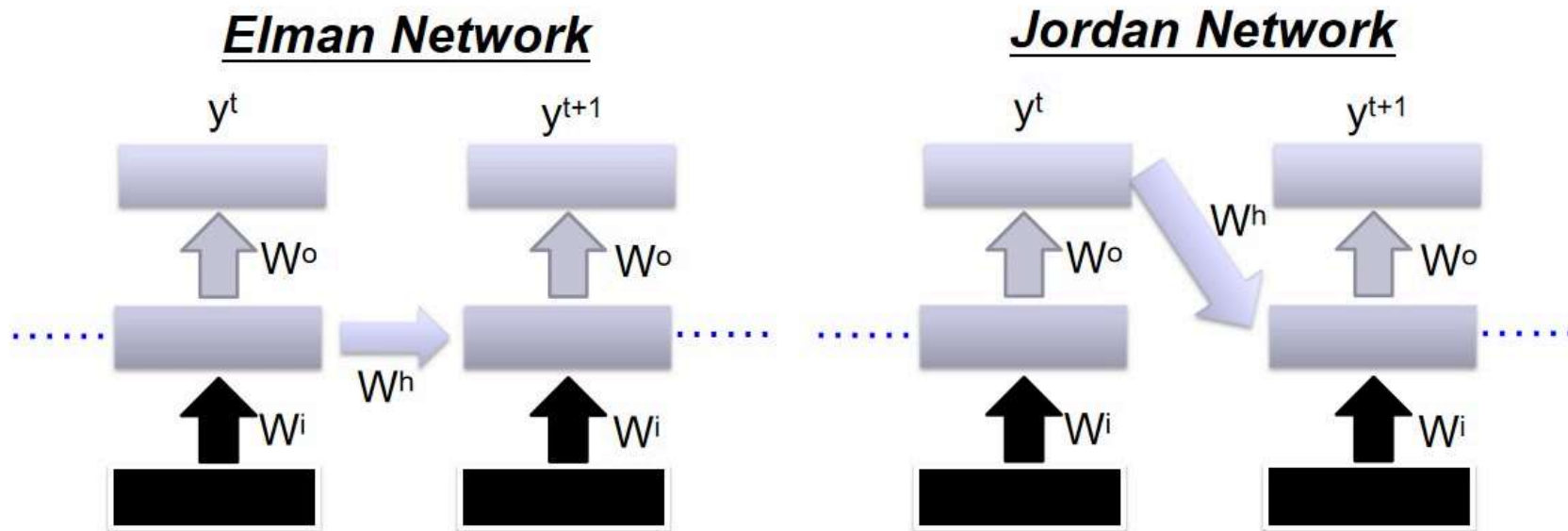




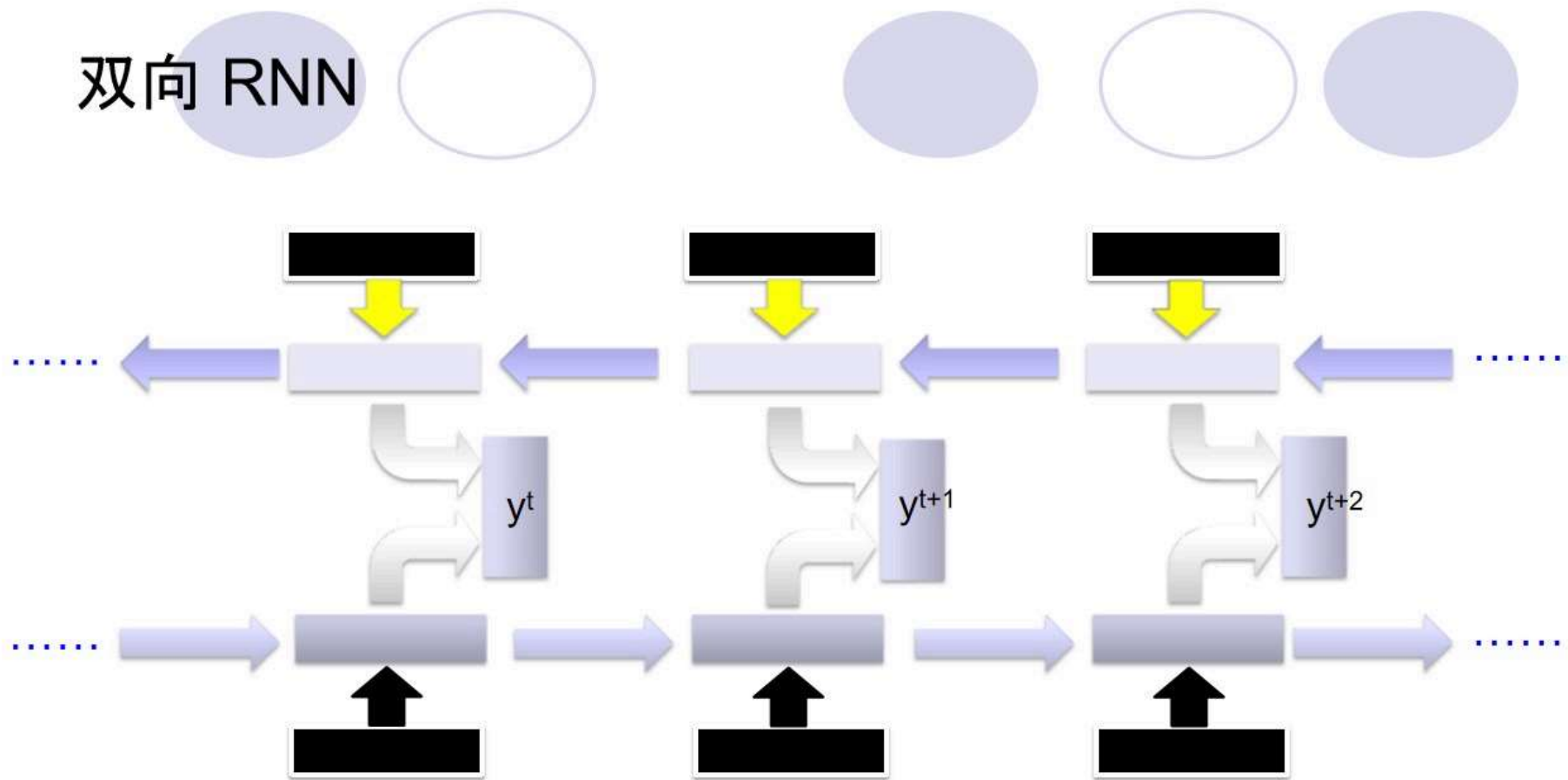
当然可以很深.....



Elman 网络和 Jordan 网络



双向 RNN





- https://www.bilibili.com/video/BV1z5411f7Bm/?spm_id_from=333.337.search-card.all.click&vd_source=e81dcbaa3b668b4eedfa281d2c2b69c3