



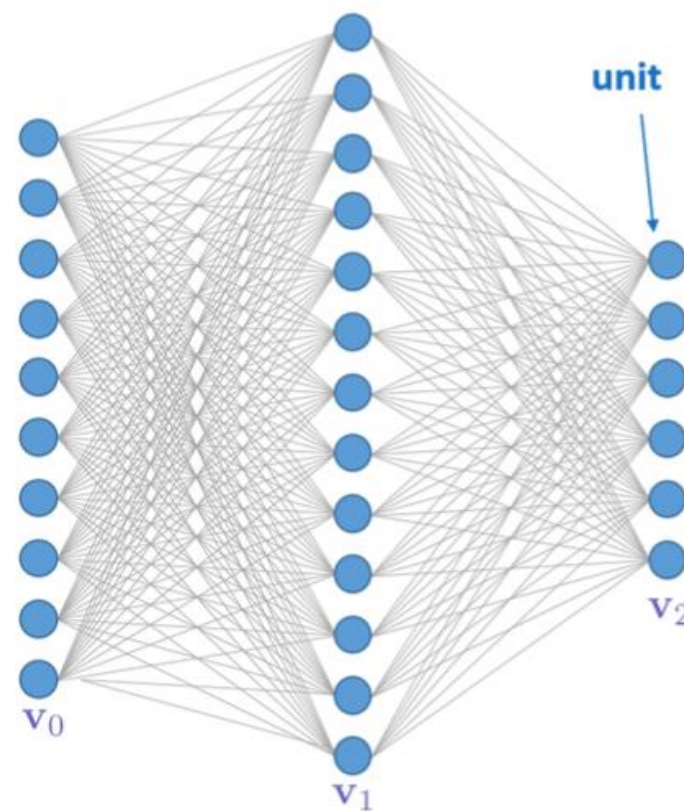
卷积神经网络CNN

CNN

卷积神经网络是近年发展起来，并广泛应用于图像处理，NLP等领域的一种多层神经网络。

传统BP处理图像时的问题：

1. 权值太多，计算量太大
2. 权值太多，需要大量样本进行训练。

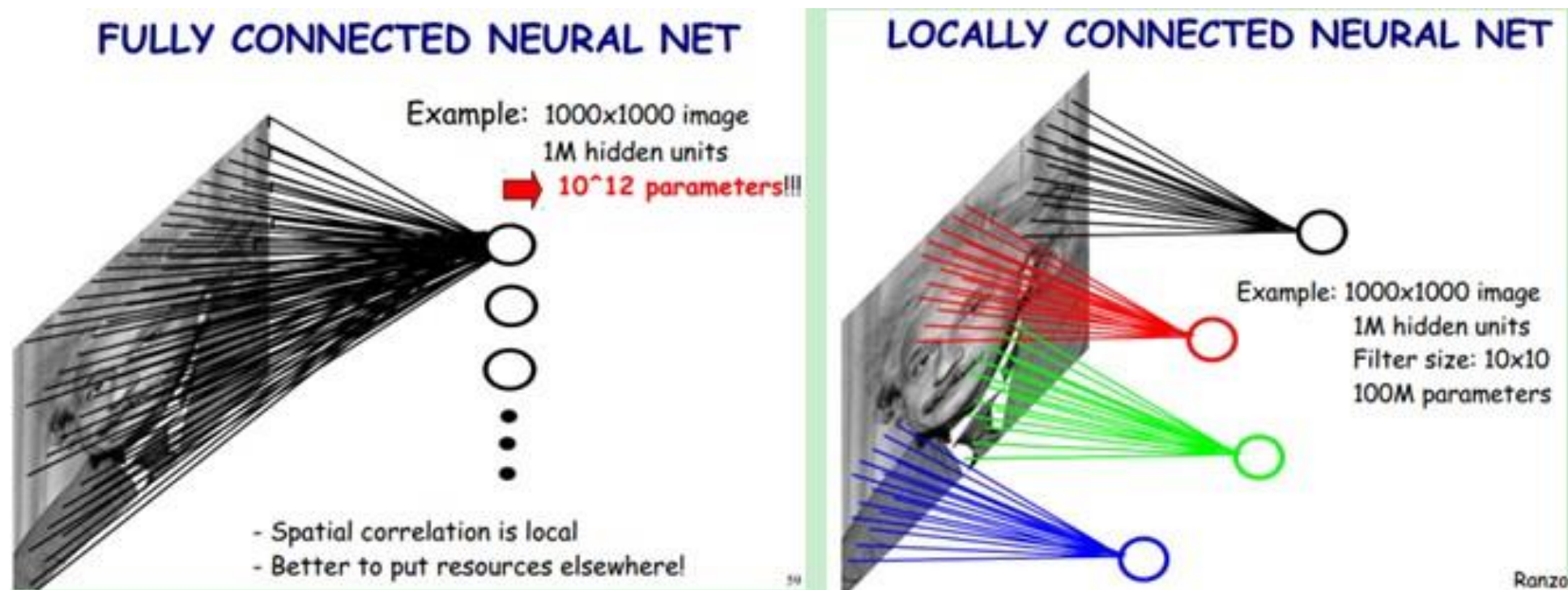


局部感受野

1962年哈佛医学院神经生理学家Hubel和Wiesel通过对猫视觉皮层细胞的研究，提出了感受野(receptive field)的概念，1984年日本学者Fukushima基于感受野概念提出的神经认知机(neocognitron)可以看作是卷积神经网络的第一个实现网络，也是感受野概念在人工神经网络领域的首次应用。

局部感受野和权值共享

CNN通过局部感受野和权值共享减少了神经网络需要训练的参数个数



卷积计算

卷积核/滤波器

1	0	1
0	1	0
1	0	1

特征图:feature map

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

Image

4		

Convolved Feature

$$1*1+1*0+1*1+0*0+1*1+1*0+0*1+0*0+1*1 = 4$$

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

Image

4	3	4
2	4	

Convolved Feature

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

Image

4	3	4
2	4	3
2	3	4

Convolved Feature

不同步长的卷积

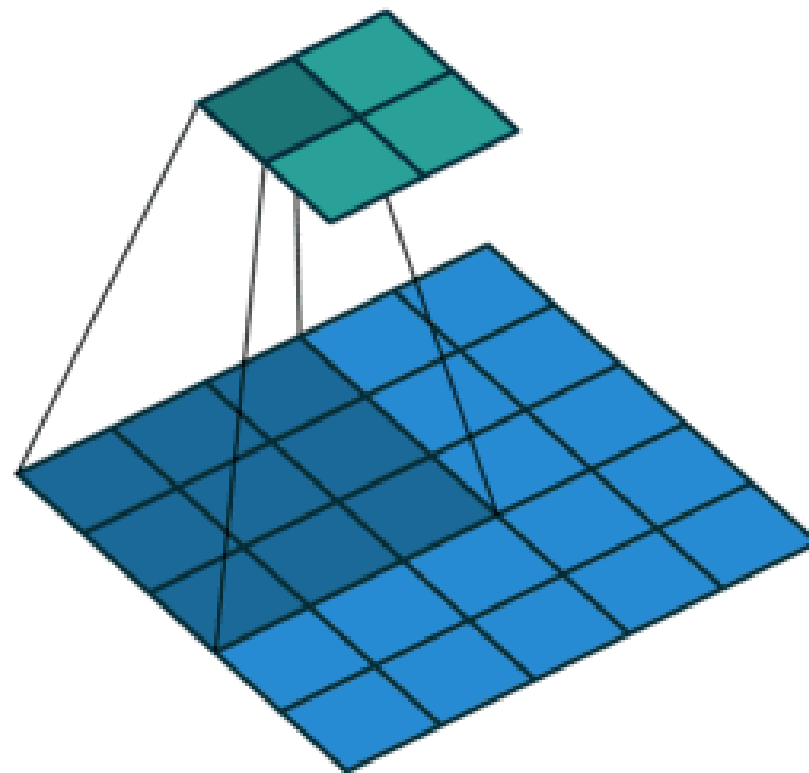
1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

步长为1

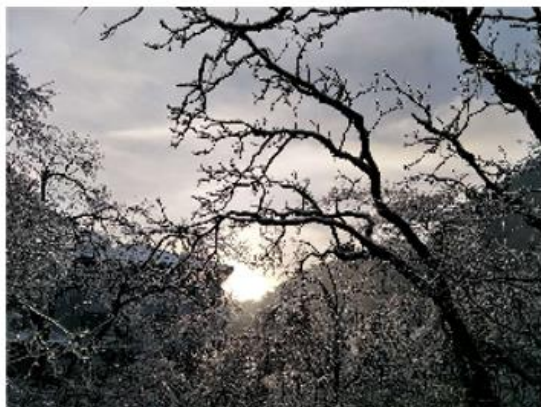
4		

Convolved
Feature



步长为2

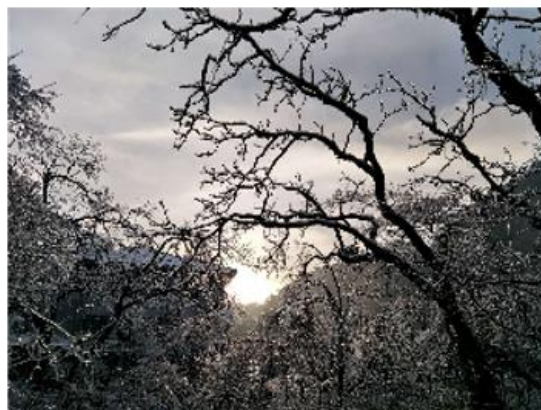
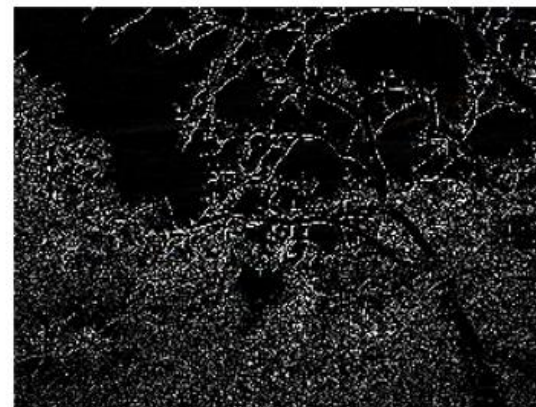
濾波器



*

-1	-1	0
-1	0	1
0	1	1

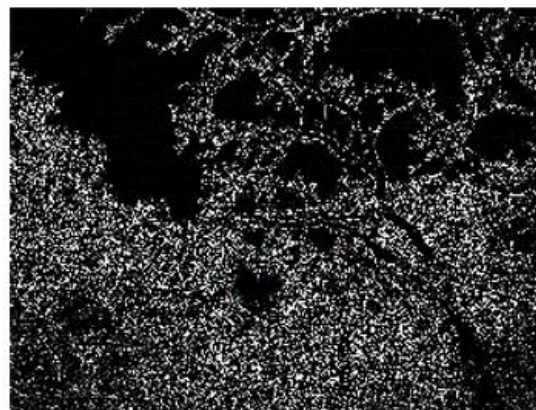
=



*

-1	-1	-1
-1	8	-1
-1	-1	-1

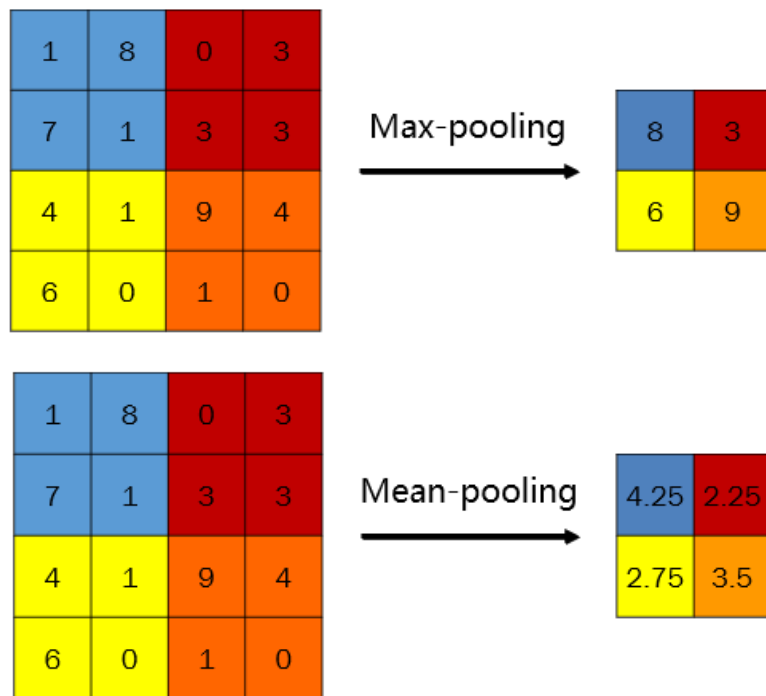
=



池化Pooling

Pooling常用的三种方式：

- 1.max-pooling
- 2.mean-pooling
- 3.stochastic pooling



Padding

SAME PADDING:

给平面外部补0

卷积窗口采样后得到一个跟原来大小相同的平面

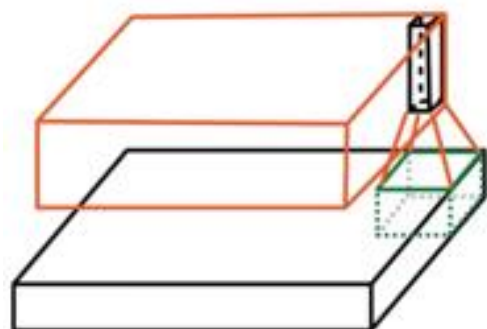
VALID PADDING:

不会超出平面外部

卷积窗口采样后得到一个比原来平面小的平面

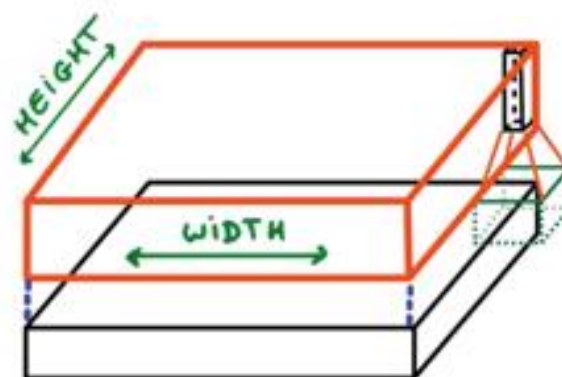
VALID PADDING

CONVOLUTIONAL
LINGO



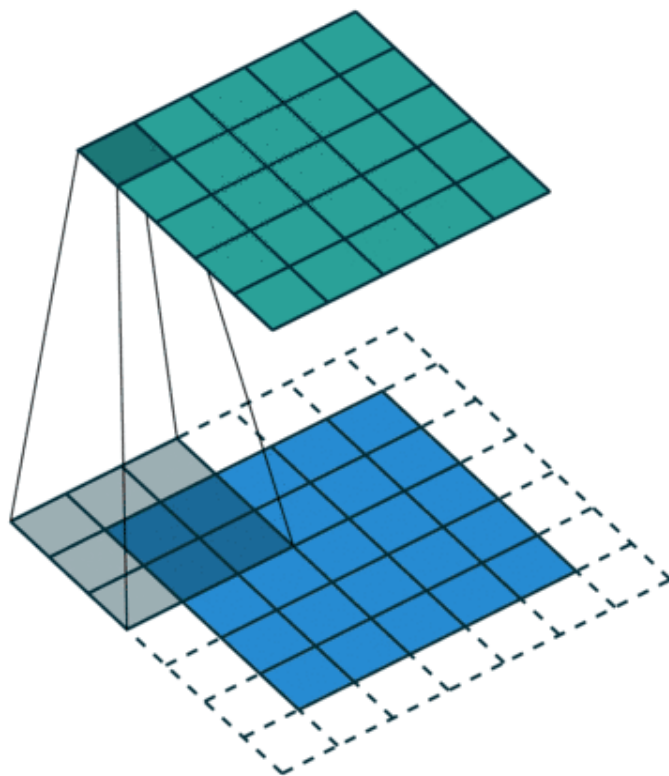
SAME PADDING

'VALID' PADDING
'SAME' PADDING

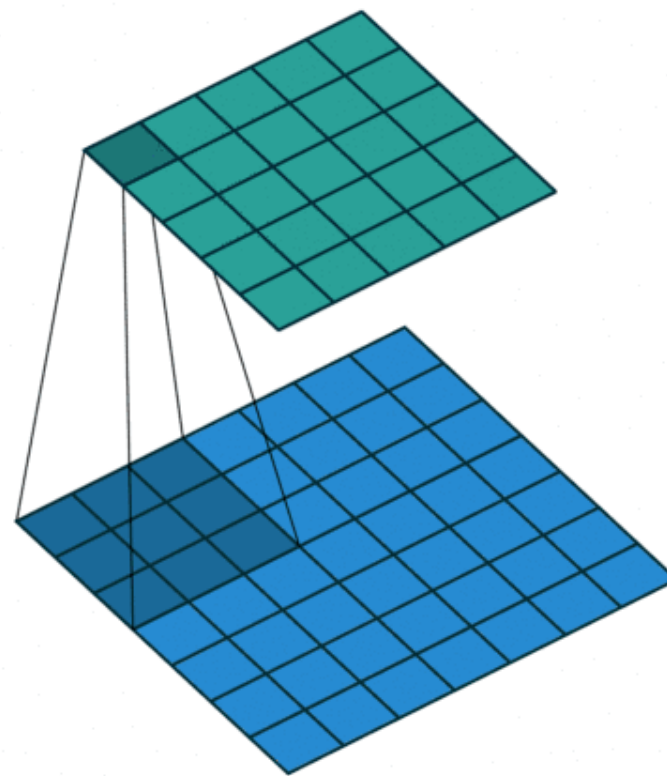


Padding

SAME PADDING



VALID PADDING



Padding

SAME PADDING:可能会给平面外部补0

VALID PADDING:不会超出平面外部

假如有一个 28×28 的平面，用 2×2 步长为2的窗口对其进行卷积/池化操作

使用SAME PADDING的方式，得到 14×14 的平面

使用VALID PADDING的方式，得到 14×14 的平面

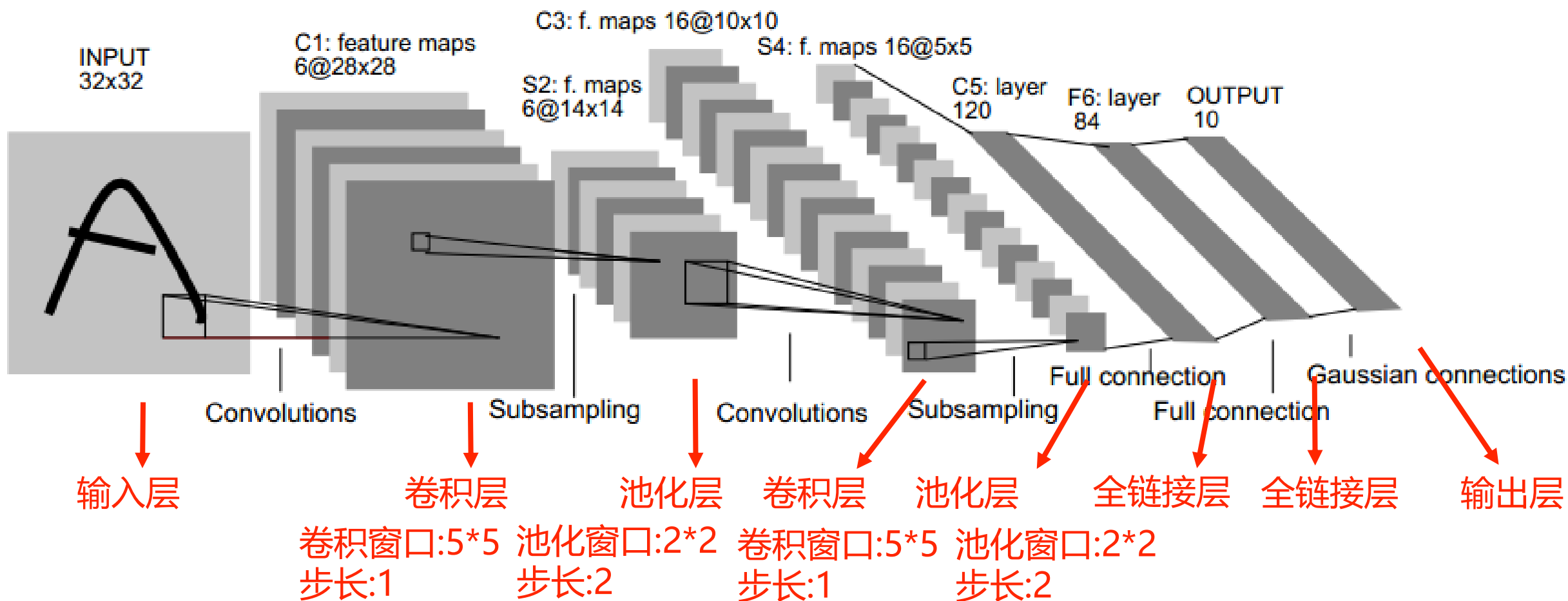
假如有一个 2×3 的平面，用 2×2 步长为2的窗口对其进行卷积/池化操作

使用SAME PADDING的方式，得到 1×2 的平面

使用VALID PADDING的方式，得到 1×1 的平面

LeNET-5

LeNET-5是最早的卷积神经网络之一，曾广泛用于美国银行。手写数字识别正确率在99%以上。

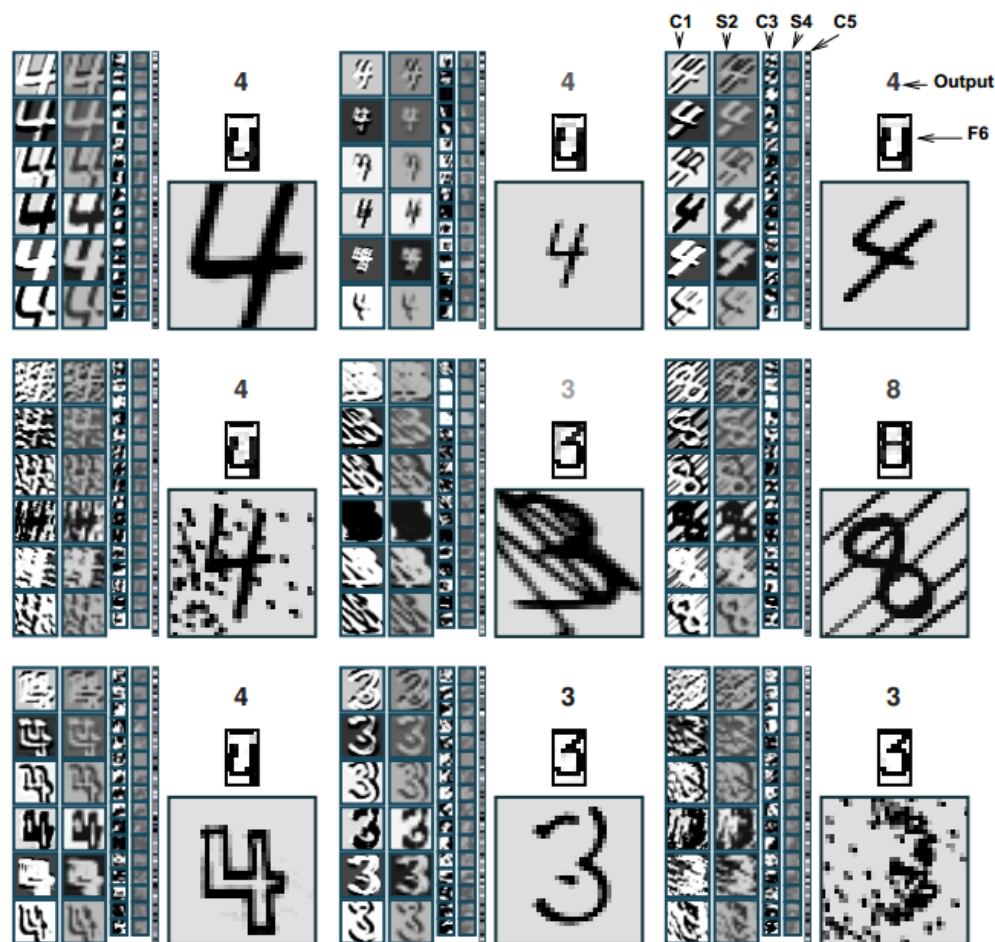


LeNET-5

可视化：

<http://scs.ryerson.ca/~aharley/vis/conv/>

<http://scs.ryerson.ca/~aharley/vis/conv/flat.html>





ImageNet介绍

ImageNet

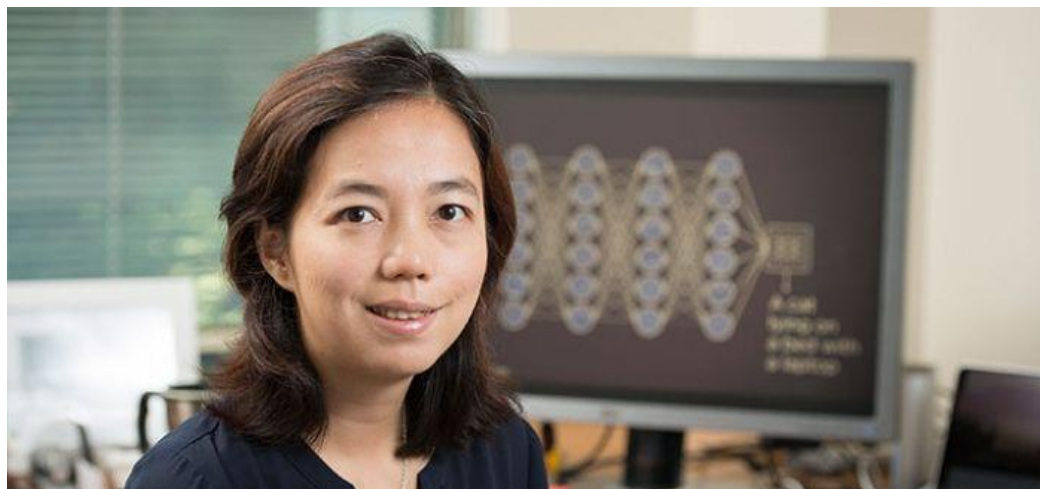
ImageNet是一个计算机视觉系统识别项目，是目前世界上图像识别最大的数据库。一共有1500万张左右的图片，被分为22000个左右的类。是由斯坦福教授李飞飞领导建立的。

[TED演讲：我们怎么教计算机理解图片？](#)



李飞飞

1976年出生于北京，长在四川，16岁随父母移居美国新泽西州。
 1999年毕业于普林斯顿大学，2005年获得加州理工学院电子工程博士。
 2009年加入斯坦福大学担任助理教授，并于2012年担任副教授（终生教授），和斯坦福人工智能实验室与视觉实验室主任。
 2017年1月入职Google，担任谷歌云首席科学家。
 2018年9月卸任谷歌云首席科学家，回归斯坦福大学当教授。



ILSVRC

ILSVRC:ImageNet Large Scale Visual Recognition Challenge

ImageNet Challenge

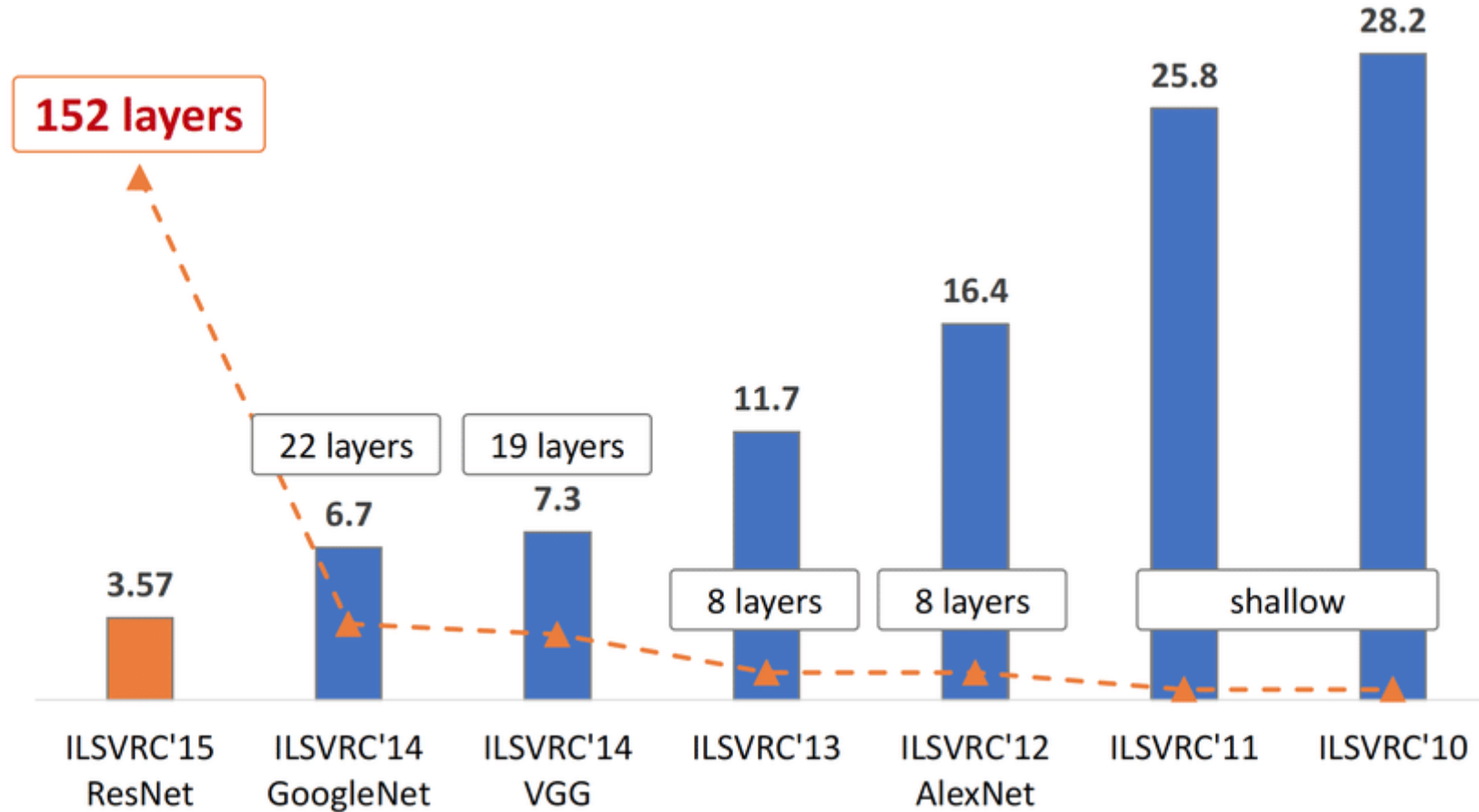
IMAGENET

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



ILSVRC

ILSVRC: ImageNet Large Scale Visual Recognition Challenge

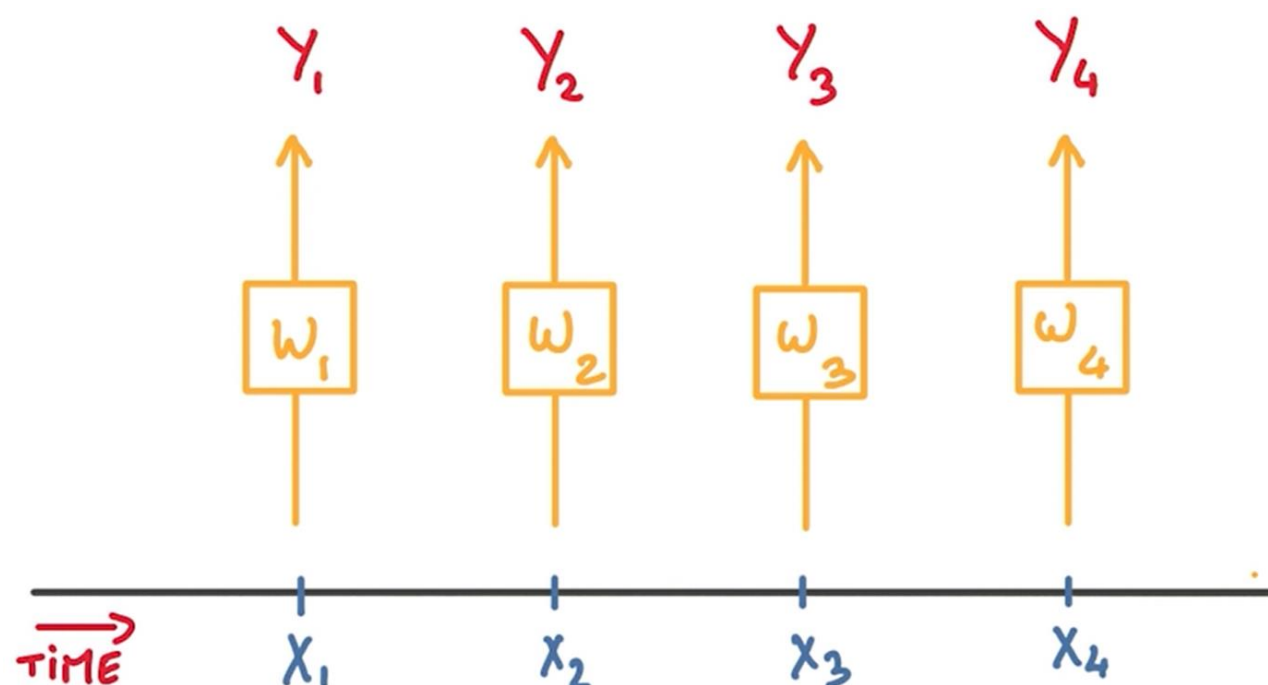


3

序列模型

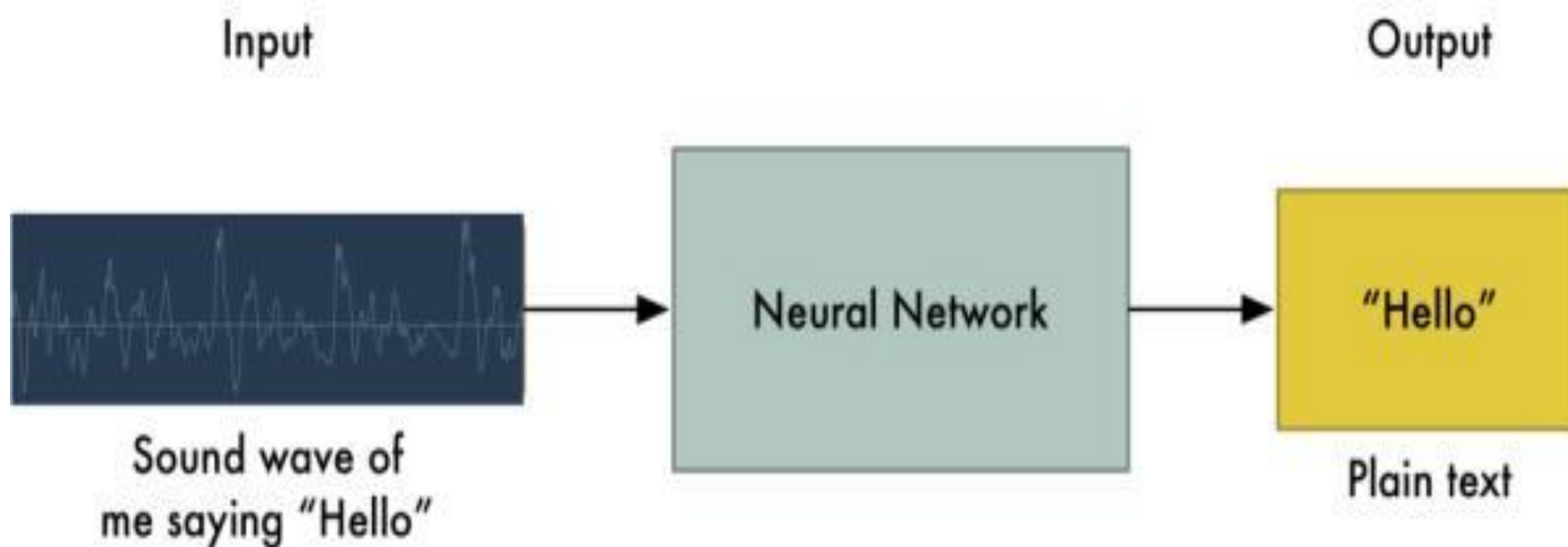
RNN (Recurrent Neural Network)

1986年 Rumelhart 等人提出循环神经网络(recurrent neural network), 简称RNN。RNN跟我们之前学习过的神经网络都不太一样, 它是一种序列模型。比如卷积网络是专门用来处理网格化数据(例如图像数据)的神经网络, RNN是专门用来处理序列数据的神经网络。所谓的序列数据指的是跟序列相关的数据, 比如一段语音, 一首歌曲, 一段文字, 一段录像等。



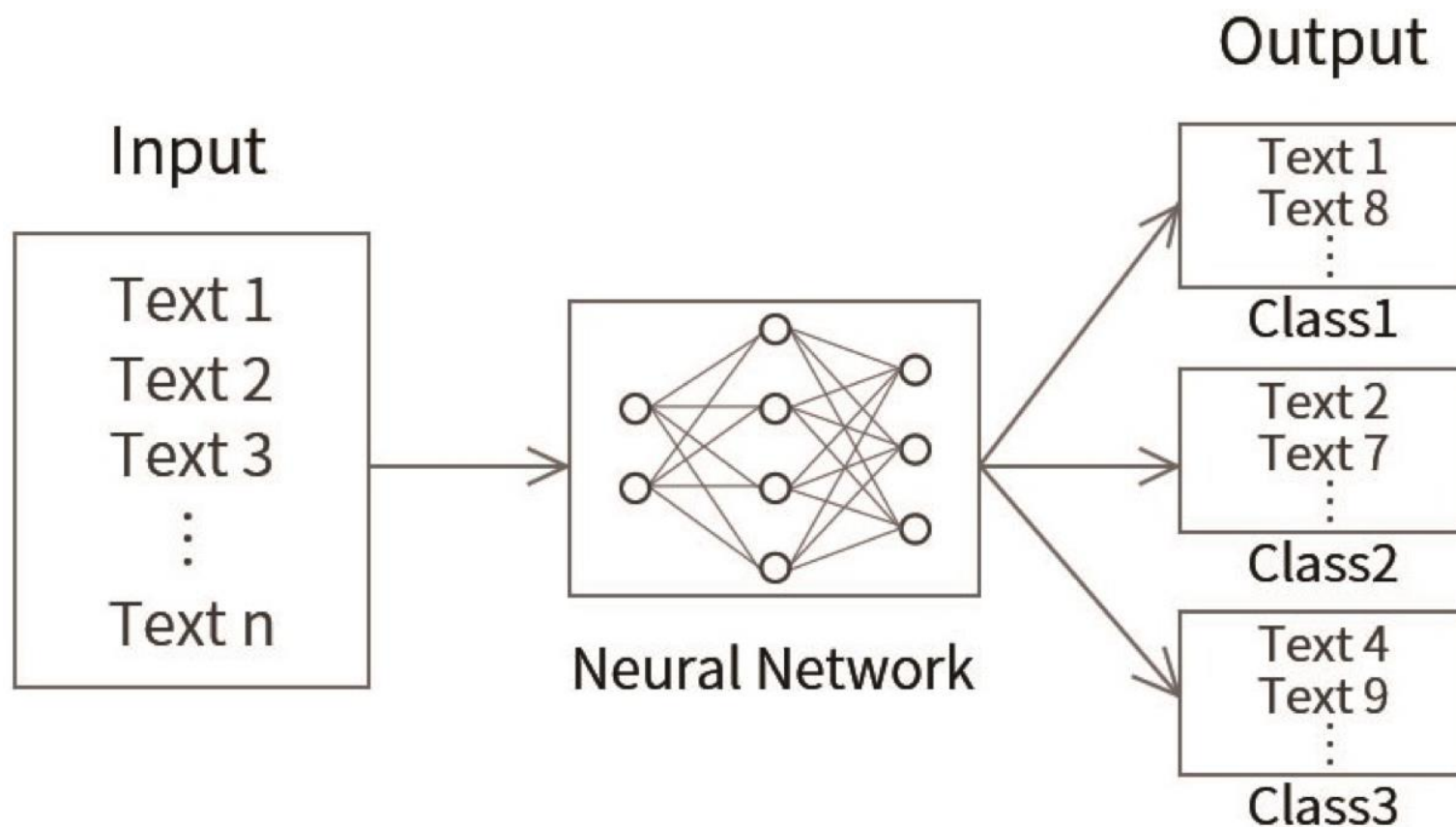
序列模型应用：语音识别

把语音转换成为文字



序列模型应用：文本分类

把文章，邮件或用户评论等文本数据做分类



序列模型应用：机器翻译

例如把中文翻译成英文



The screenshot displays the Google Translate web interface. At the top, the Google logo is on the left, and a link to '关闭即时翻译' (Close real-time translation) is on the right. Below the logo, the word '翻译' (Translate) is shown in red. The interface features a language selection bar with buttons for '英语' (English), '中文' (Chinese), '德语' (German), and a '检测语言' (Detect language) dropdown. A swap button is located between the language bars. The source language is set to '中文(简体)' (Chinese Simplified), and the target language is '英语' (English). A blue '翻译' (Translate) button is on the right. The input text area contains the Chinese sentence: '谷歌公司发布了新一代深度学习计算框架Tensorflow，我们可以使用Tensorflow进行机器翻译。' (Google has released a new generation of deep learning computing framework Tensorflow, we can use Tensorflow for machine translation). Below the input, there are speaker icons for audio playback and a character count '52/5000'. The output text area shows the English translation: 'Google has released a new generation of deep learning computing framework Tensorflow, we can use Tensorflow for machine translation.' Below the output, there is a '提出修改建议' (Suggest an edit) link and a '点击图标下载 App' (Click icon to download app) link. At the bottom right, there are buttons for 'Android' and 'iOS' app downloads.

Google

翻译

关闭即时翻译

英语 中文 德语 检测语言

中文(简体) 英语 日语

翻译

谷歌公司发布了新一代深度学习计算框架Tensorflow，我们可以使用Tensorflow进行机器翻译。

Google has released a new generation of deep learning computing framework Tensorflow, we can use Tensorflow for machine translation.

提出修改建议

点击图标下载 App

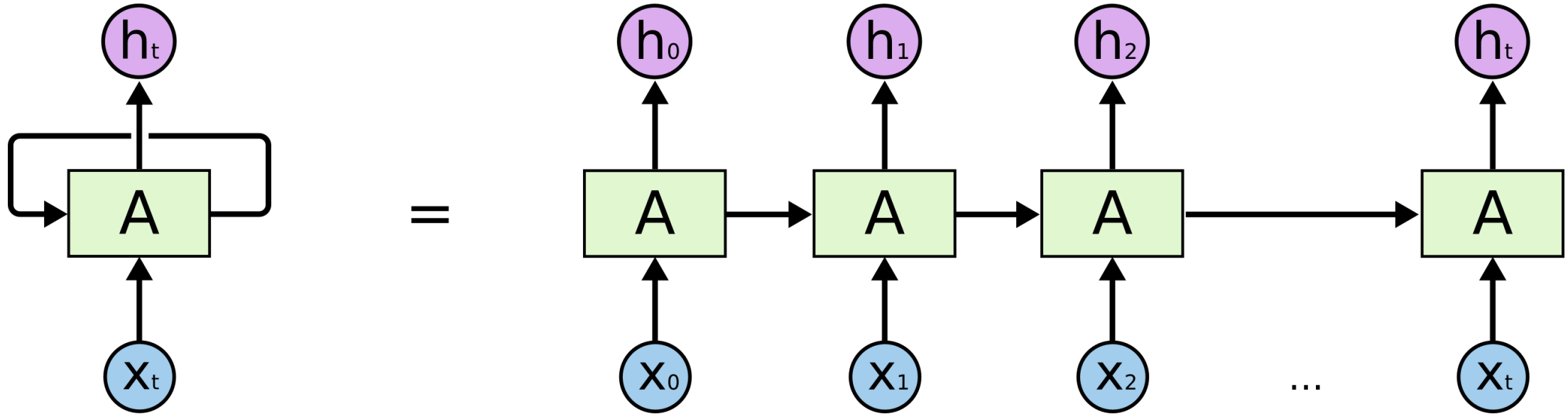
Android iOS

序列模型应用：分词标注

给一段文字做分词标注，标注每个字对应的标号。假如使用4-tag(BMES)标注标签，B表示词的起始位置，M表示词的中间位置，E表示词的结束位置，S表示单字词。可以得到类似如下结果：

“人/B 们/E 常/S 说/S 生/B 活/E 是/S 一/S 部/S 教/B 科/M 书/E ”

RNN (Recurrent Neural Network)





RNN (Recurrent Neural Network)

Elman network^[10]

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$

$$y_t = \sigma_y(W_y h_t + b_y)$$

Jordan network^[11]

$$h_t = \sigma_h(W_h x_t + U_h y_{t-1} + b_h)$$

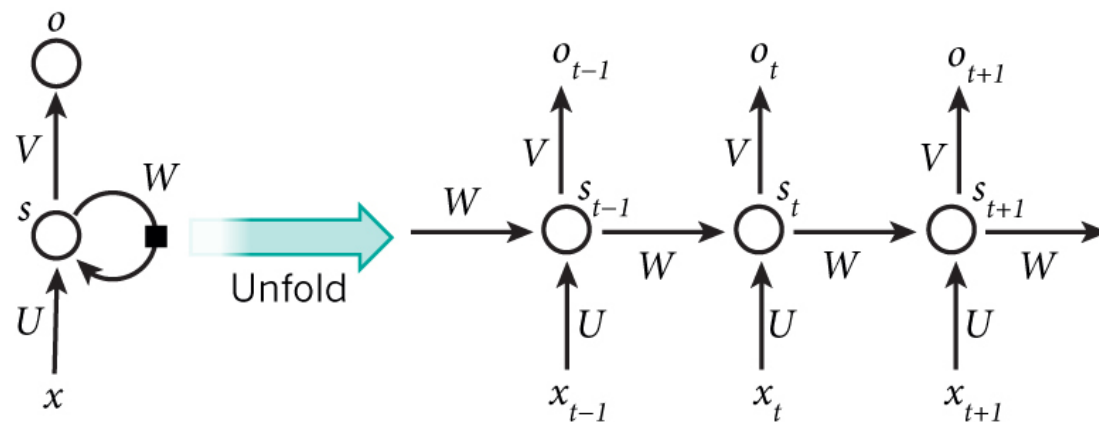
$$y_t = \sigma_y(W_y h_t + b_y)$$

Variables and functions

- x_t : input vector
- h_t : hidden layer vector
- y_t : output vector
- W , U and b : parameter matrices and vector
- σ_h and σ_y : Activation functions

隐层 h_t 接收的是上时刻的隐层 (hidden layer) h_{t-1}
还是上时刻的输出 (output layer) y_{t-1} ,
可以分成了两种 RNN :

Elman network 接收上时刻的隐层 h_{t-1}
Jordan network 接收上时刻的输出 y_{t-1}



RNN (Recurrent Neural Network)

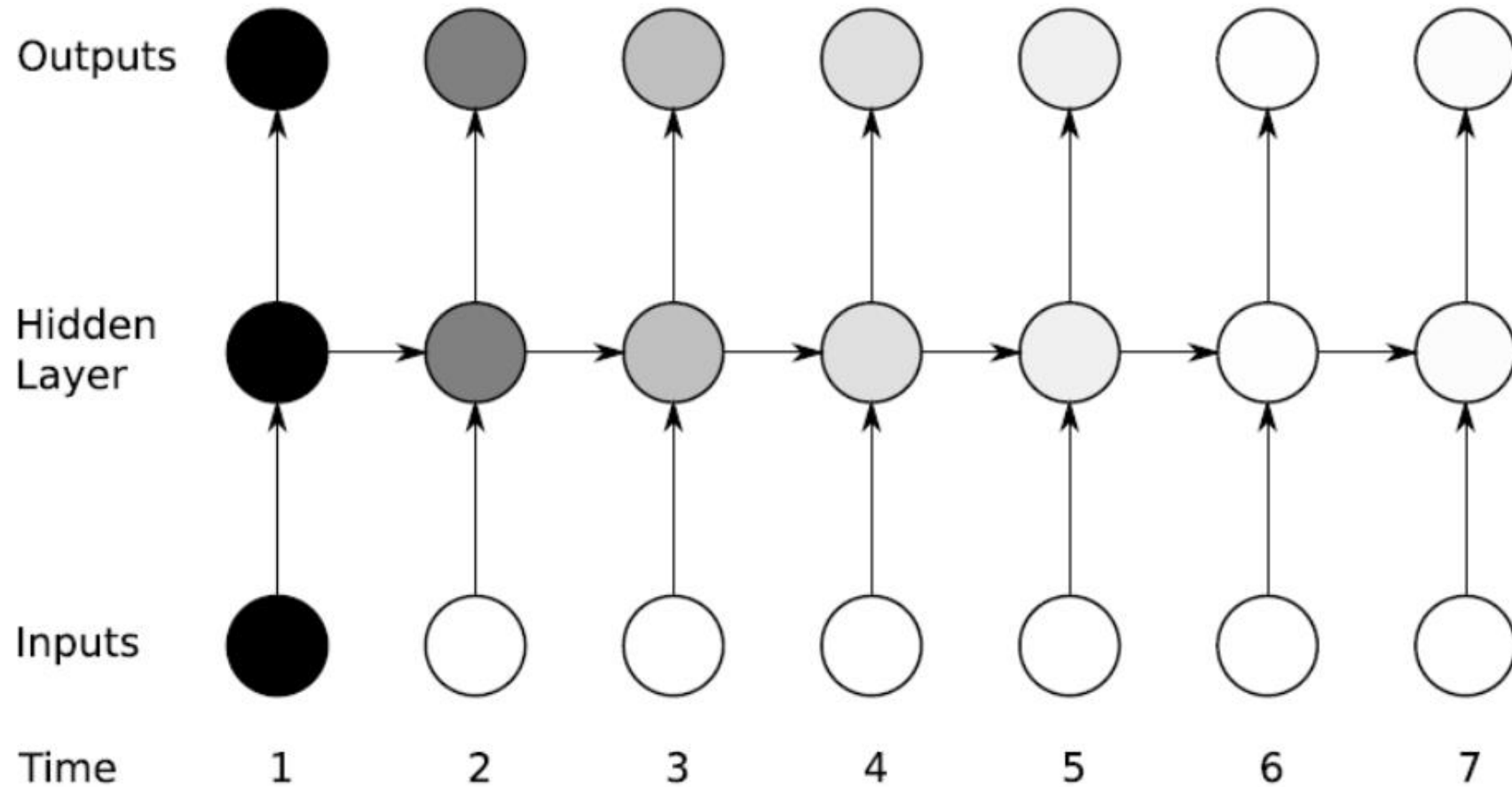
RNN一个重要的用法就是通过之前的信息来决策当前的问题。

比如就像我们看电影，我们要根据电影之前的情节，才能理解现在的情节。

例子1：有一朵云飘在（ ）

例子2：我从小生长在美国。。。我可以说一口流利的（ ）

RNN (Recurrent Neural Network)



4

长短期记忆网络LSTM

LSTM(Long Short Term Memory)

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

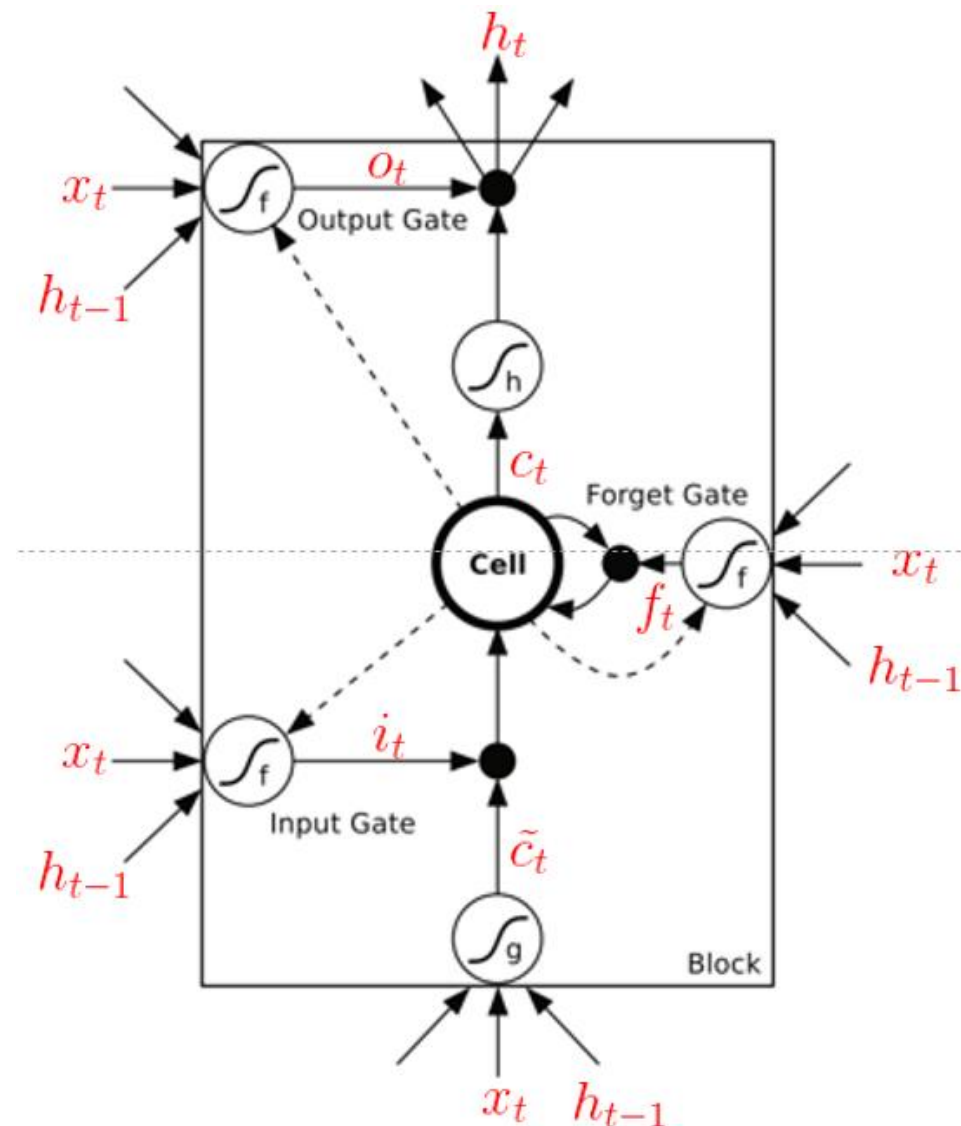
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = o_t \circ \sigma_h(c_t)$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

i_t 输入门信号
 f_t 忘记门信号
 \tilde{c}_t Cell输入信号
 c_t Cell输出信号
 o_t 输出门信号
 h_t block输出信号

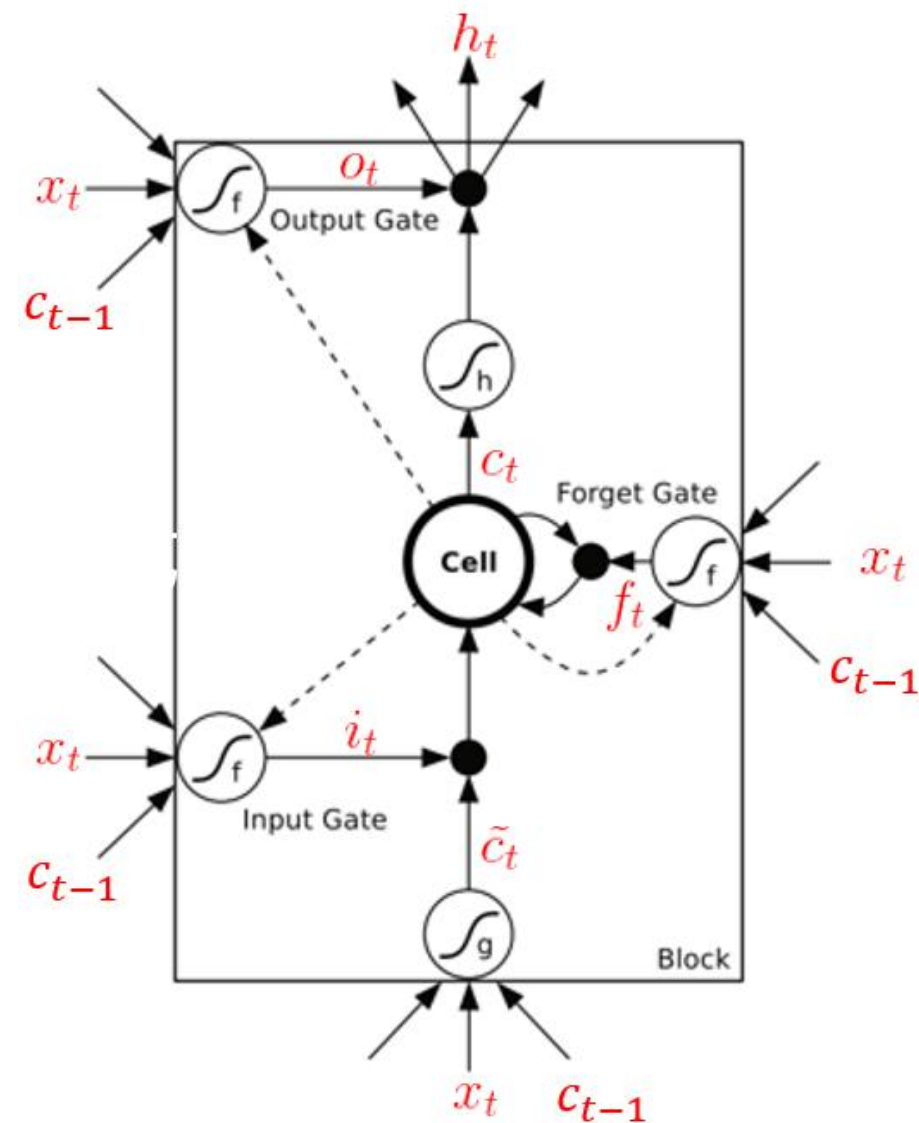
x_t 第t个序列输入
 h_{t-1} 第t-1个序列输出
 σ_g sigmoid函数
 σ_c tanh函数
 σ_h tanh函数或线性函数



Peephole LSTM

$$\begin{aligned}
 f_t &= \sigma_g(W_f x_t + U_f c_{t-1} + b_f) \\
 i_t &= \sigma_g(W_i x_t + U_i c_{t-1} + b_i) \\
 o_t &= \sigma_g(W_o x_t + U_o c_{t-1} + b_o) \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c c_{t-1} + b_c) \\
 h_t &= o_t \circ \sigma_h(c_t)
 \end{aligned}$$

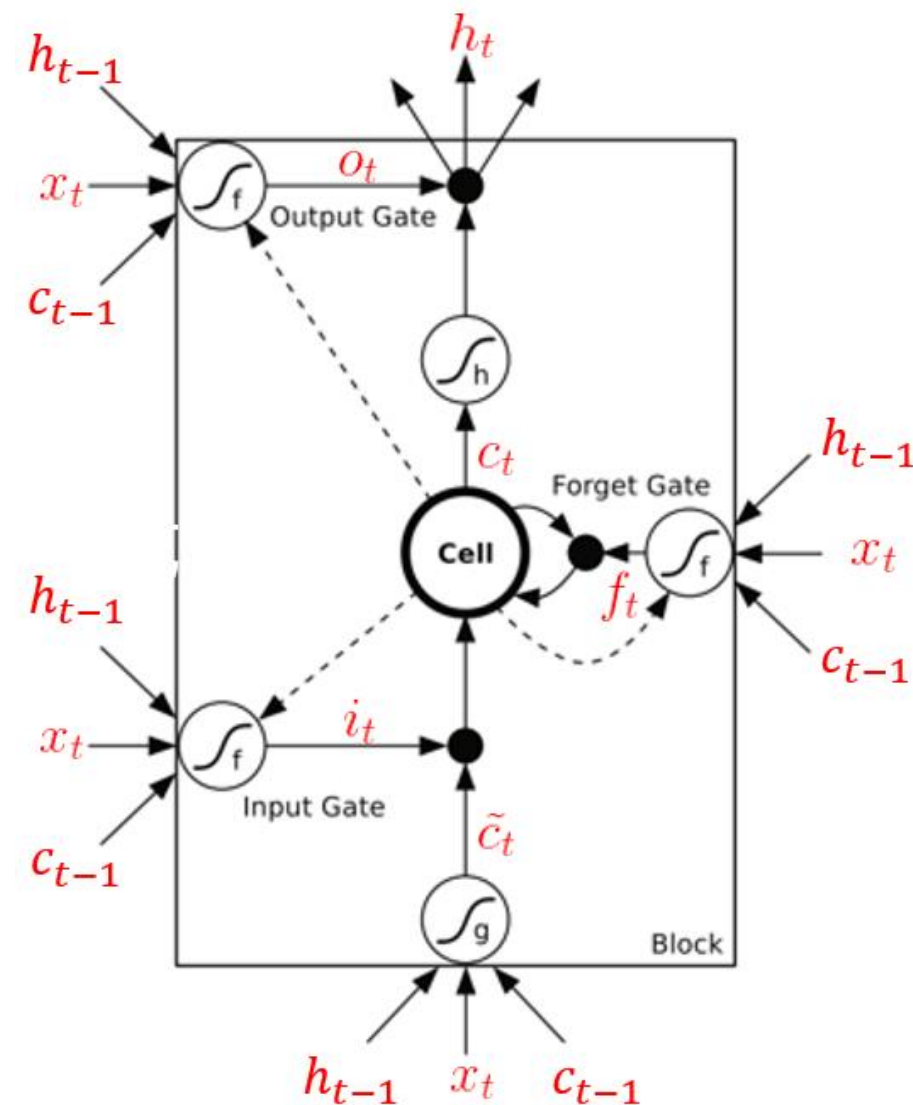
和之前的公式做比较，发现只是把 h_{t-1} 都换成了 c_{t-1} ，即三个门的输入都改成了 $[x_t, c_{t-1}]$ 。因为是从 cell state 里取得信息，所以叫窥视孔 (peephole)。



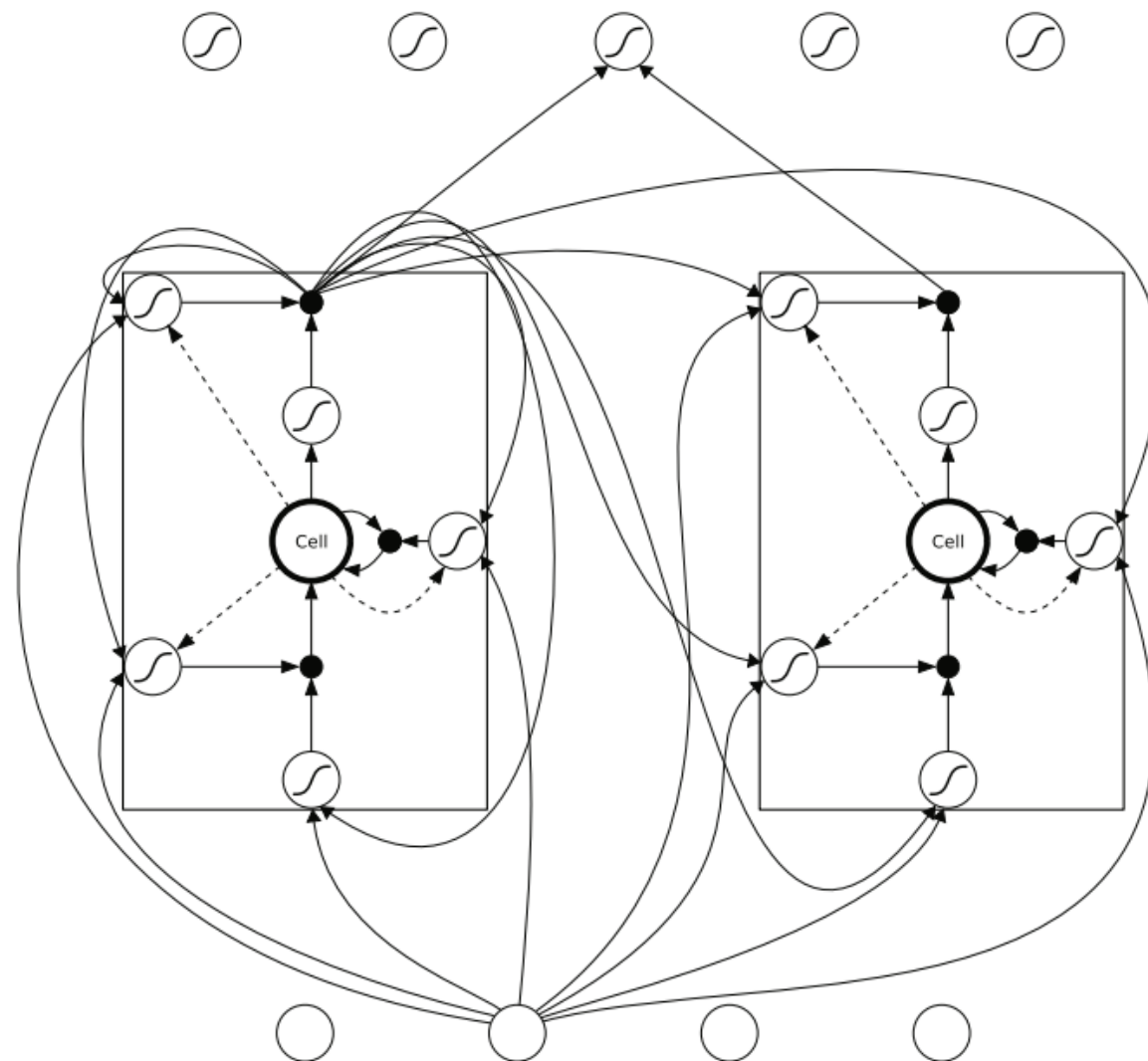
FC-LSTM(Fully-connected LSTM)

$$\begin{aligned}
 f_t &= \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \circ c_{t-1} + b_f) \\
 i_t &= \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \circ c_{t-1} + b_i) \\
 o_t &= \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \circ c_{t-1} + b_o) \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \\
 h_t &= o_t \circ \sigma_h(c_t)
 \end{aligned}$$

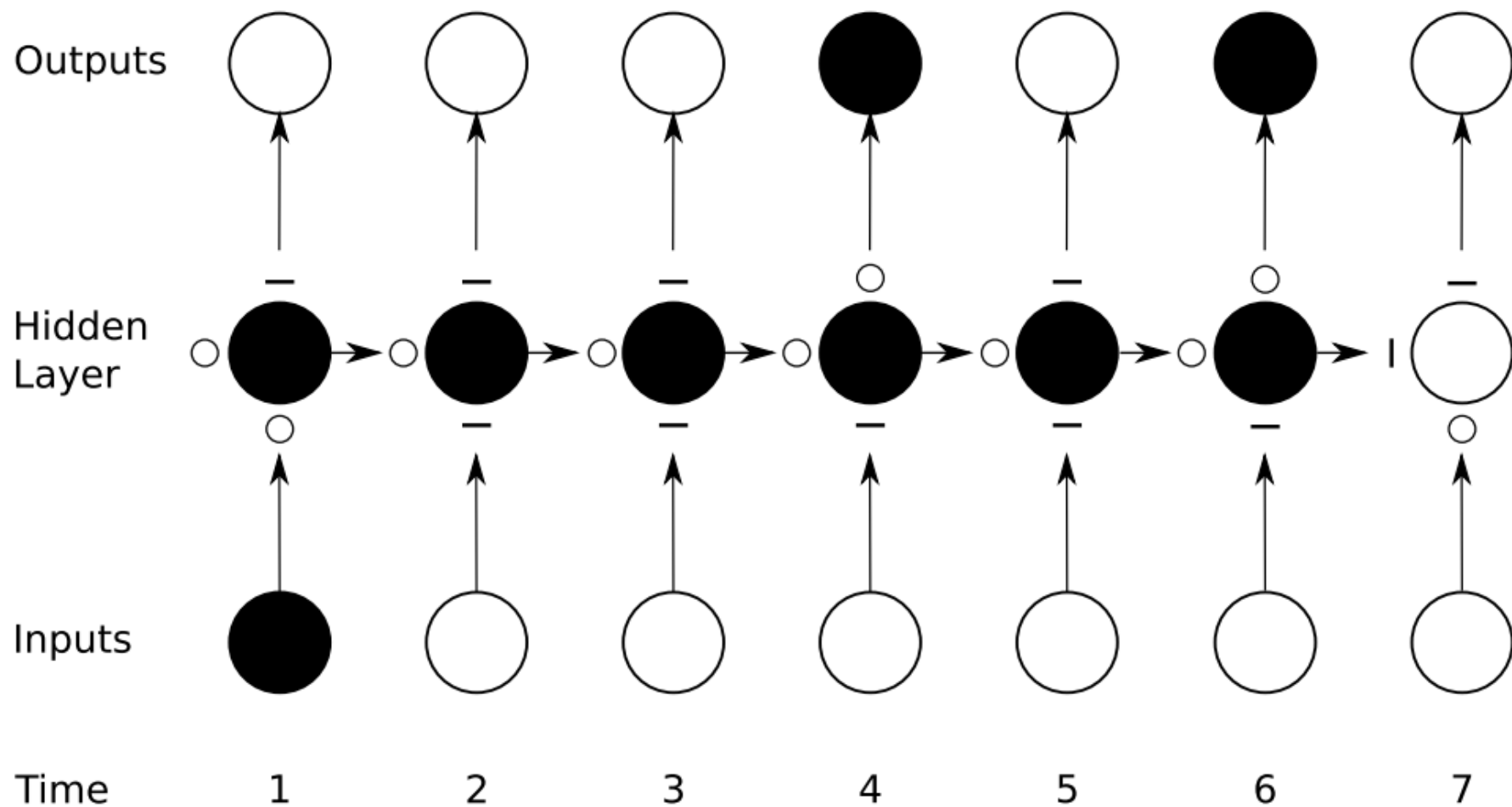
三个输入分别是 $[x_t, h_{t-1}, c_{t-1}]$



LSTM(Long Short Term Memory)



LSTM(Long Short Term Memory)



5

门控循环单元GRU

GRU(Gated Recurrent Unit)

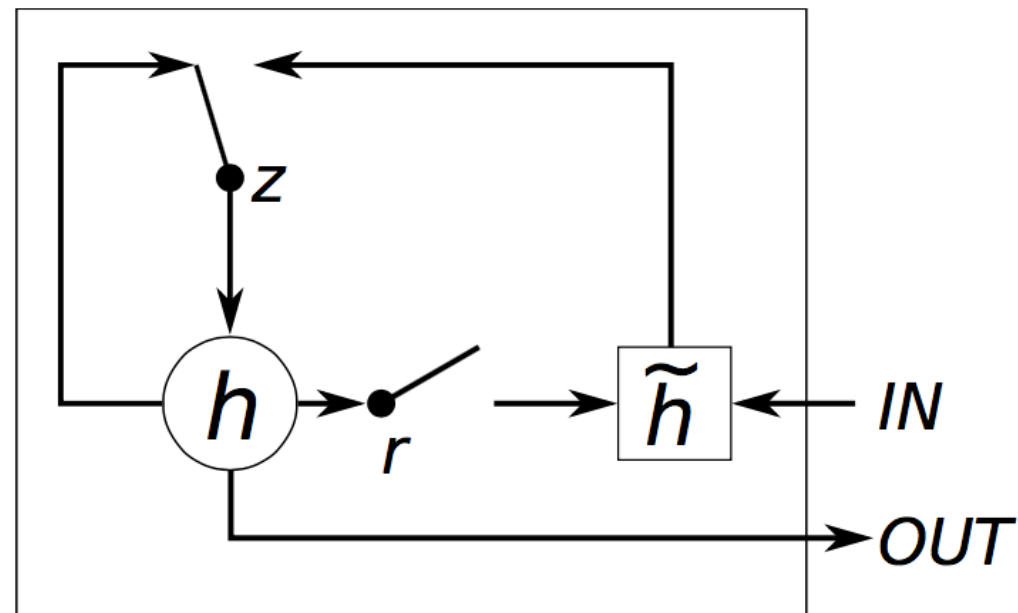
GRU 这个结构是 2014 年才出现的，效果跟 LSTM 差不多，但是用到的参数更少。将忘记门和输入门合成了一个单一的更新门。

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

$$\tilde{h}_t = \tanh(W \tilde{x}_t + U(r_t \circ h_{t-1}))$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t$$



z_t 是更新门(update gate)，决定 h_t 的更新情况

r_t 是重置门(reset gate)，决定是否要放弃 h_{t-1}

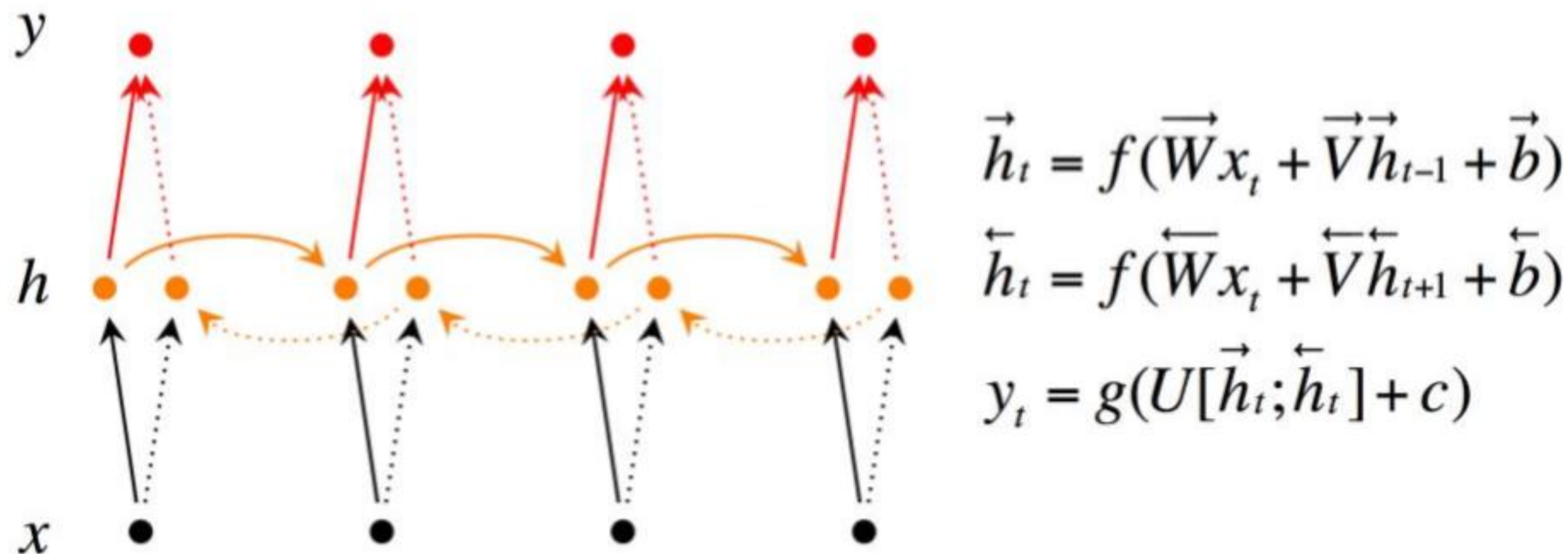
\tilde{h}_t 是候选输出，接收 $[x_t, h_{t-1}]$

h_t 是当前输出，接收 $[h_{t-1}, \tilde{h}_t]$



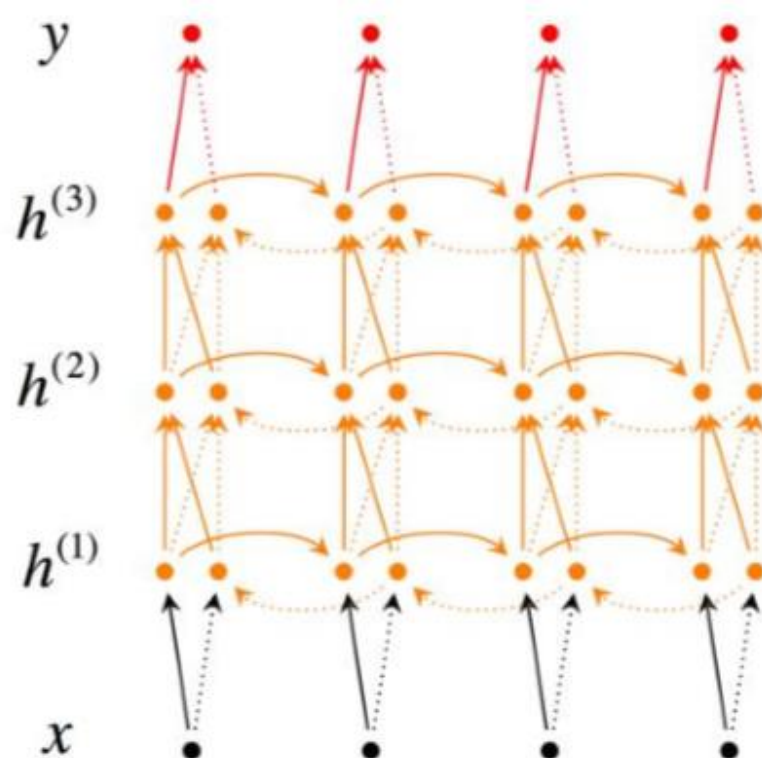
其他RNN模型

双向RNN(Bidirectional RNNs)



双向的 RNN 是同时考虑“过去”和“未来”的信息，输入（黑色点）沿着黑色的实线箭头传输到隐层（黄色点），再沿着红色实线传到输出（红色点）。黑色实线做完前向传播后，在 Bidirectional RNNs 却先不急着重后向传播，而是从末尾的时刻沿着虚线的方向再回传回来。

Stacked Bidirectional RNNs



$$\vec{h}_t^{(i)} = f(\vec{W}^{(i)} h_t^{(i-1)} + \vec{V}^{(i)} \vec{h}_{t-1}^{(i)} + \vec{b}^{(i)})$$

$$\overleftarrow{h}_t^{(i)} = f(\overleftarrow{W}^{(i)} h_t^{(i-1)} + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)})$$

$$y_t = g(U[\vec{h}_t^{(L)}; \overleftarrow{h}_t^{(L)}] + c)$$

图中是双向的三层 RNNs，堆叠多层的RNN网络，可以增加模型的参数，提高模型的拟合能力。每层的 hidden state 不仅要输给下一时刻，还是当做是此时刻下一层的输入。



THANKS