NEURAL NETWORKS AND DEEP LEARNING

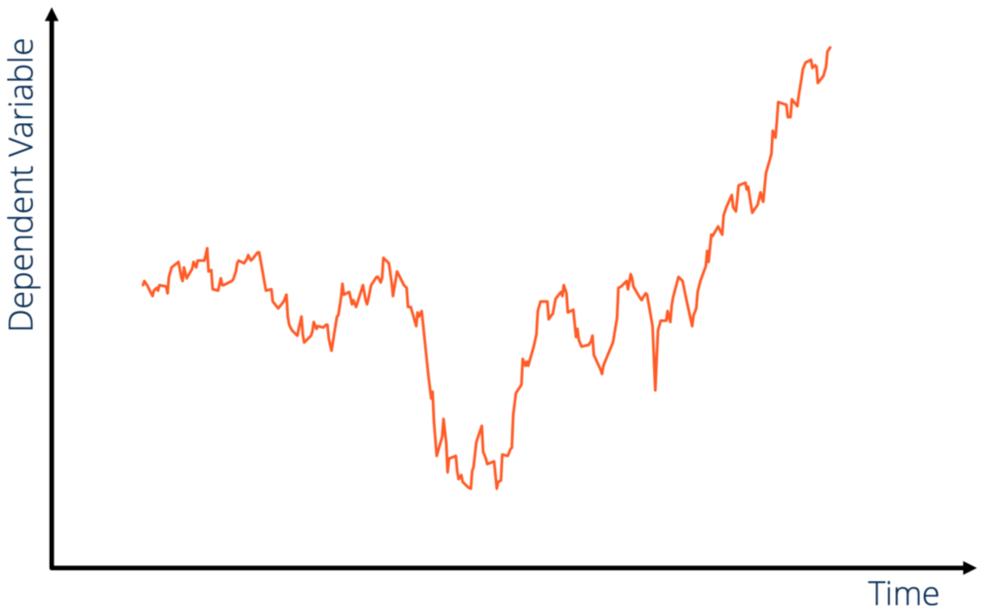
神经网络与深度学习

Recurrent Neural Networks
循环神经网络

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

什么是时间序列数据

Time-Series Analysis



	A	B	C	D	E	F
1	Date	Volume	Open	High	Low	Close
2	1-Apr	423454	12.12	12.89	12	12.18
3	2-Apr	534535	13.12	13.55	12.98	13.2
4	3-Apr	464255	11.16	12.04	11.11	11.3
5	4-Apr	462163	17.12	17.12	17.06	17.1
6	5-Apr	724552	16.24	16.25	15.95	16.24
7	6-Apr	452426	20.28	20.28	19.35	20.28
8	7-Apr	623562	14.12	14.53	13.75	14.2
9	8-Apr	245621	10.89	12.36	10.85	11.3
10	9-Apr	631531	20.1	20.1	18.69	20.04
11	10-Apr	222455	12.11	13.65	11.94	12.3

Lookback
5 days information

Next day,
Up? Down?

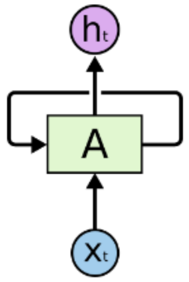
- 传统DNN只能利用当前时刻的信息（i.e., 5-Apr）
- RNN能够回溯更多时间步的信息去改善预测结果

循环神经网络

Humans **don't start their thinking from scratch** every second. As you read this essay, you understand each word **based on your understanding of previous words**. You don't throw everything away and start thinking from scratch again.

Traditional neural networks can't do this, and it seems like a major shortcoming.

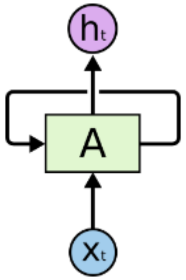
Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist.



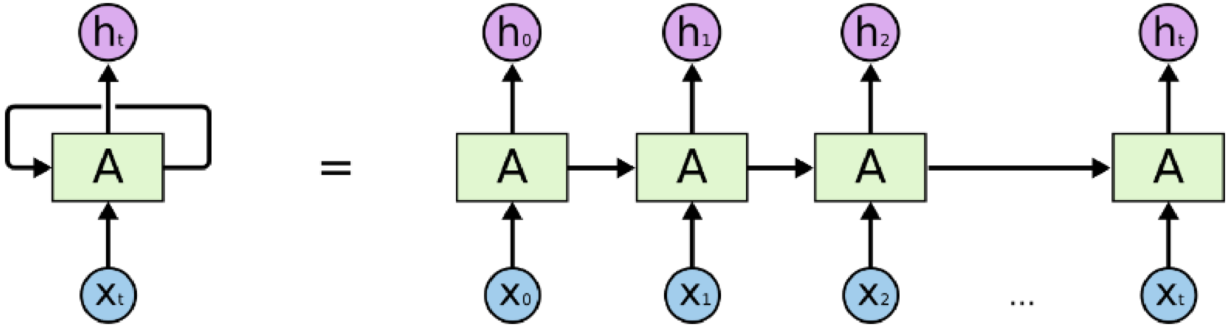
Recurrent Neural Networks have loops.

- h_t, x_t，具有时间下标，沿着时间维度信息流动
- A不变（网络参数），说明在不同的时间步下是共享参数的，回顾卷积核

沿着时间维度（时间步）将RNN展开



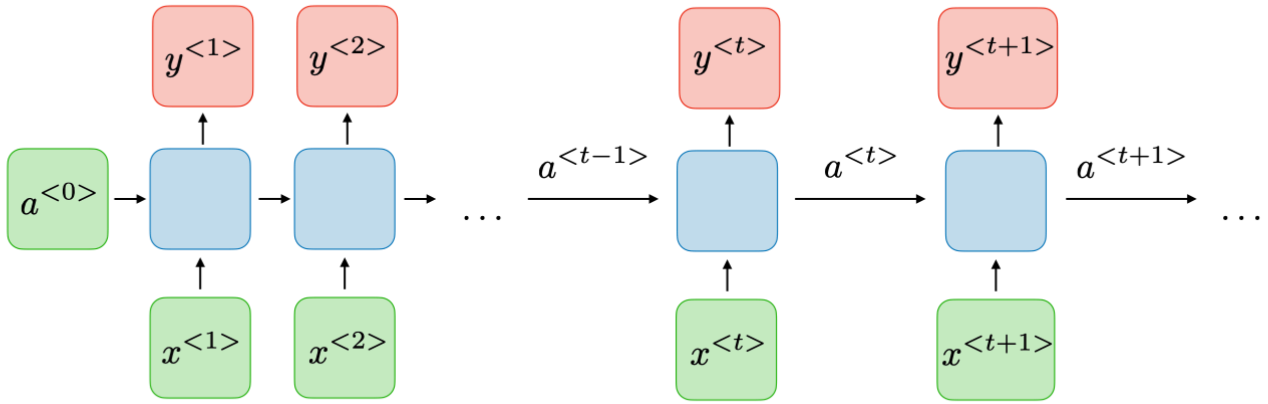
Recurrent Neural Networks have loops.



An unrolled recurrent neural network.

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states.

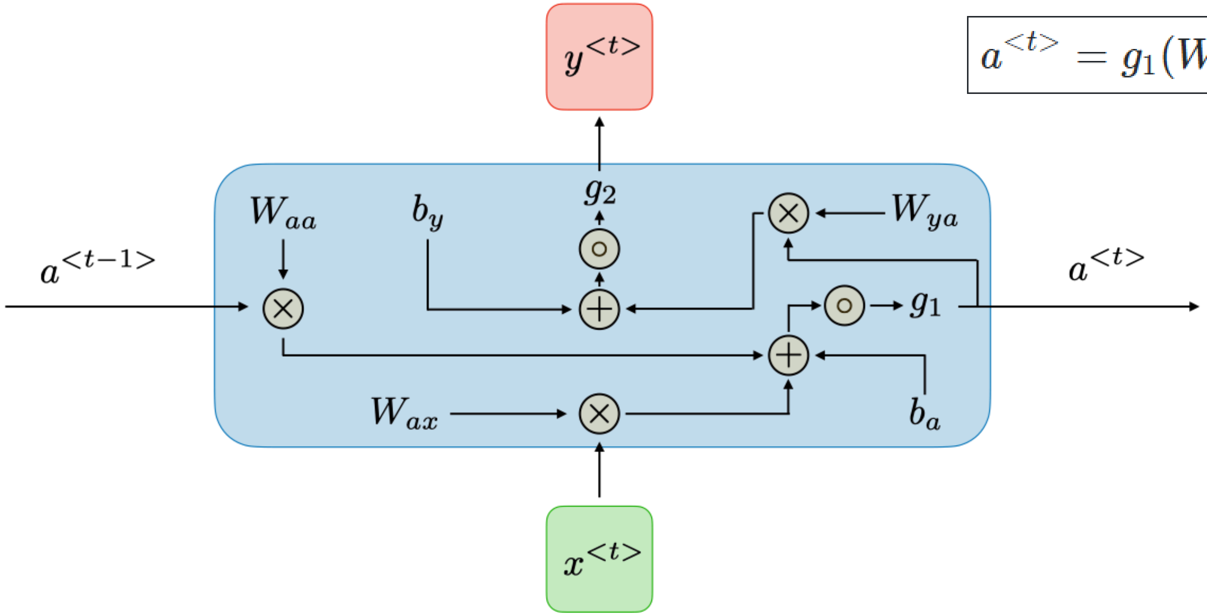


For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

本质上还是矩阵运算，只不过在时间维度上递归

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)} \quad \text{and} \quad \boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

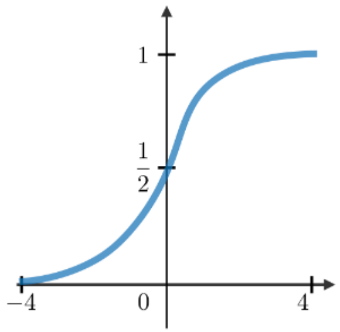
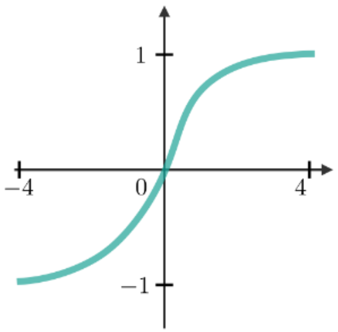
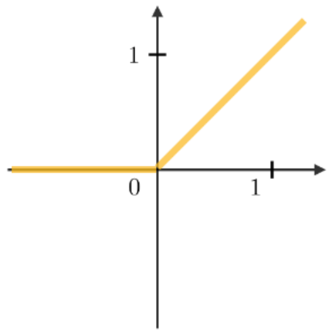
where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.



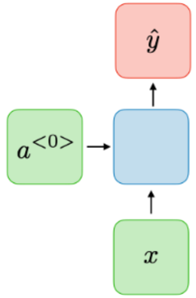
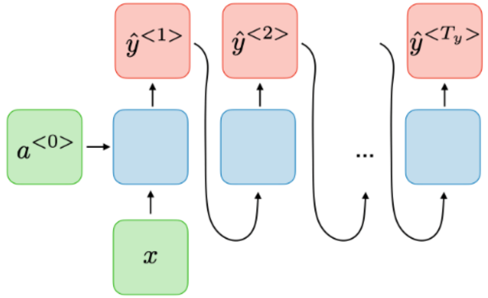
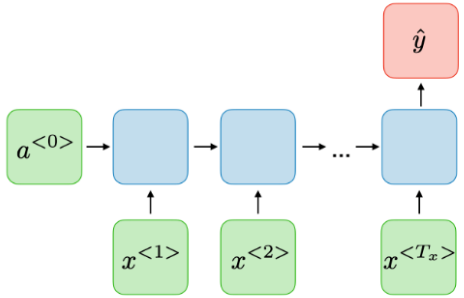
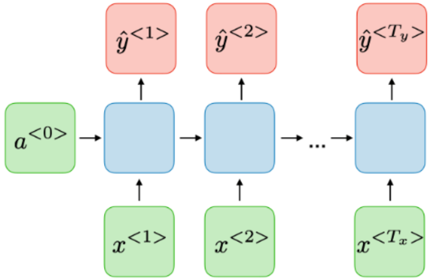
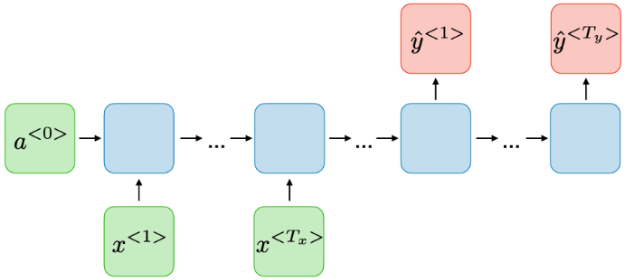
$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$ and $$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

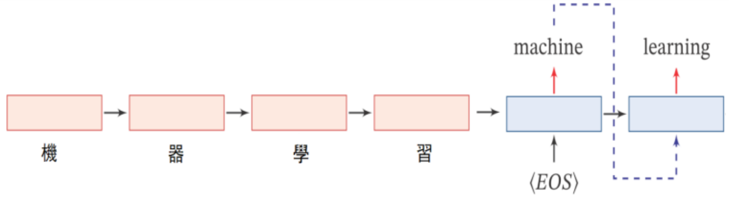
本质上还是矩阵运算，只不过在时间维度上递归，RNN中激活函数g可以有多种选择

❐ **Commonly used activation functions** — The most common activation functions used in RNN modules are described below:

Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

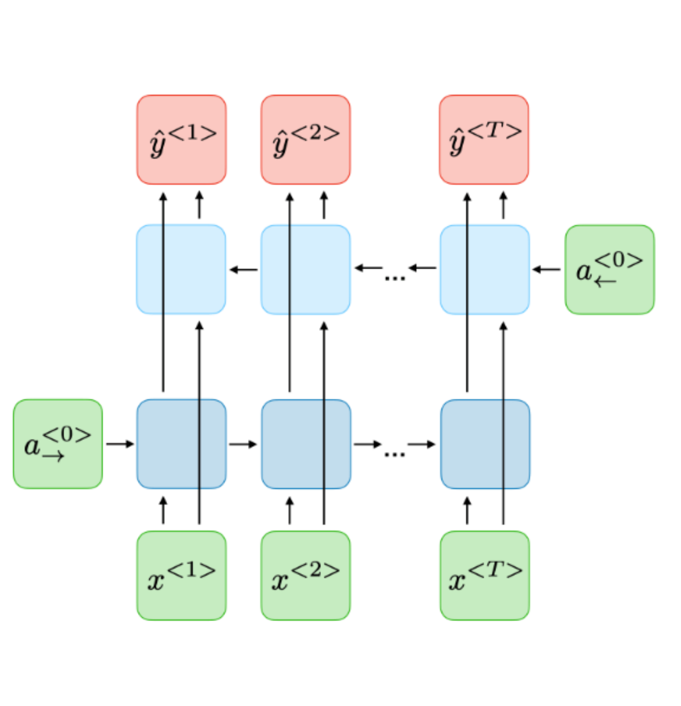
RNN Variants

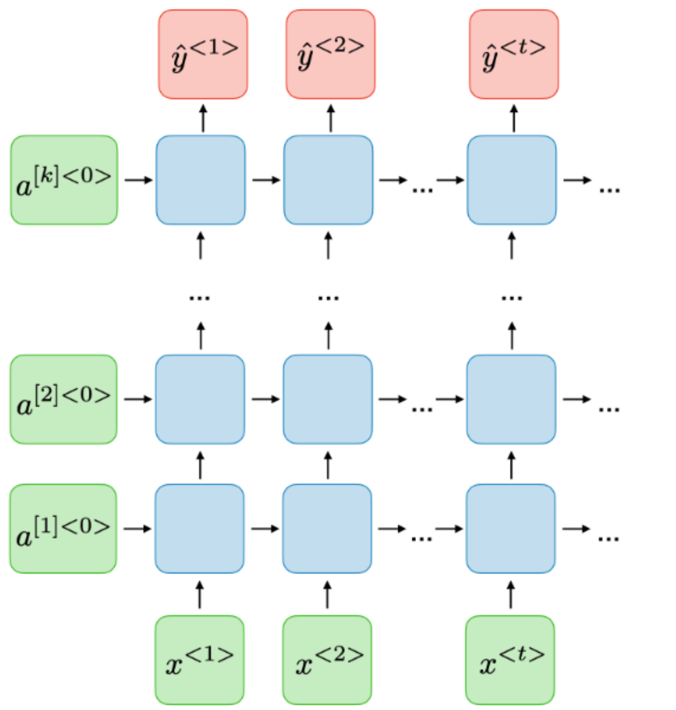
Type of RNN	Illustration	Example
One-to-one $T_x = T_y = 1$		Traditional neural network
One-to-many $T_x = 1, T_y > 1$		Music generation
Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition
Many-to-many $T_x \neq T_y$		Machine translation



RNN Variants

Bidirectional (BRNN)	**Deep** (DRNN)

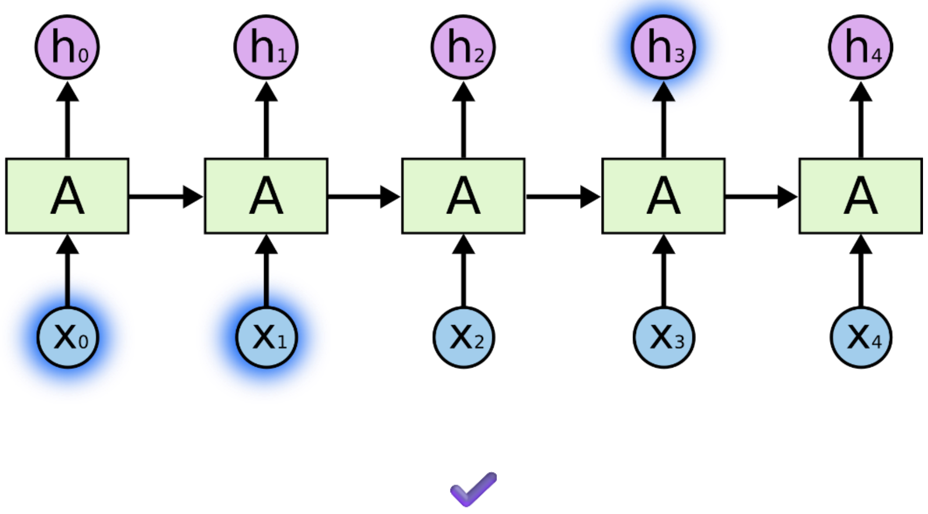


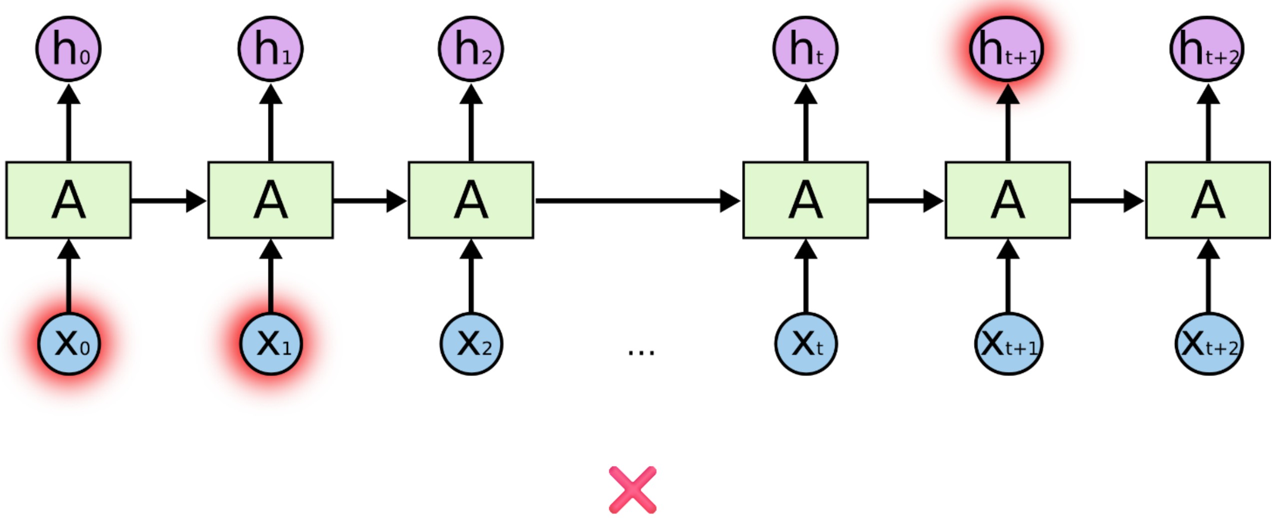


沿着时间维度正向、反向分别递归更新hidden state

某层（隐藏层）中是沿着时间维度递归更新，可以stack多层RNN

Advantages	Drawbacks
• Possibility of processing input of any length • Model size not increasing with size of input • Computation takes into account historical information • Weights are shared across time	• Computation being slow • Difficulty of accessing information from a long time ago • Cannot consider any future input for the current state

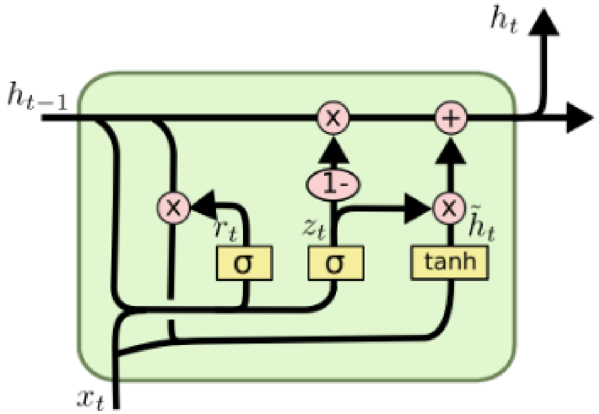




RNN Variants

- GRU
- LSTM
- RNN w/ attention (todo)

GRU



Sigmoid, element-wise gating

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

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

r_t控制h_t的计算是否/多大程度依赖上一时刻的状态 h_{t-1}。 这允许我们转换当前信息的时候一定程度上遗忘历史信息

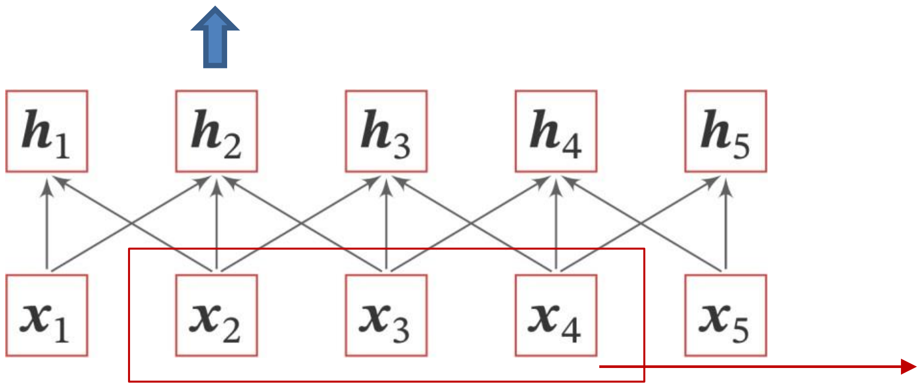
z_t 控制 h_t 的计算是如何融合当前信息与过去信息的。这允许我们在输出信息的时候一定程度上遗忘当前

GRU相比于LSTM去除了output gate，参数量更少，运算效率更高

CNN vs RNN

第二个时刻的hidden state不仅和x_1, x_2相关，还和x_3相关（未来的时间步信息）

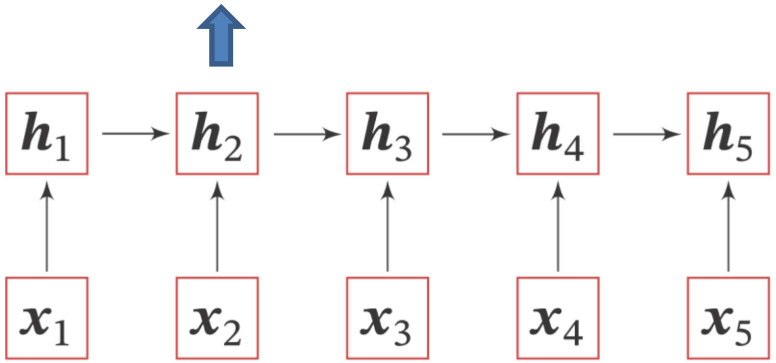
CNN



但是CNN能否对时间序列数据建模？

第二个时刻的hidden state仅和当前及过去时间步的信息相关

RNN



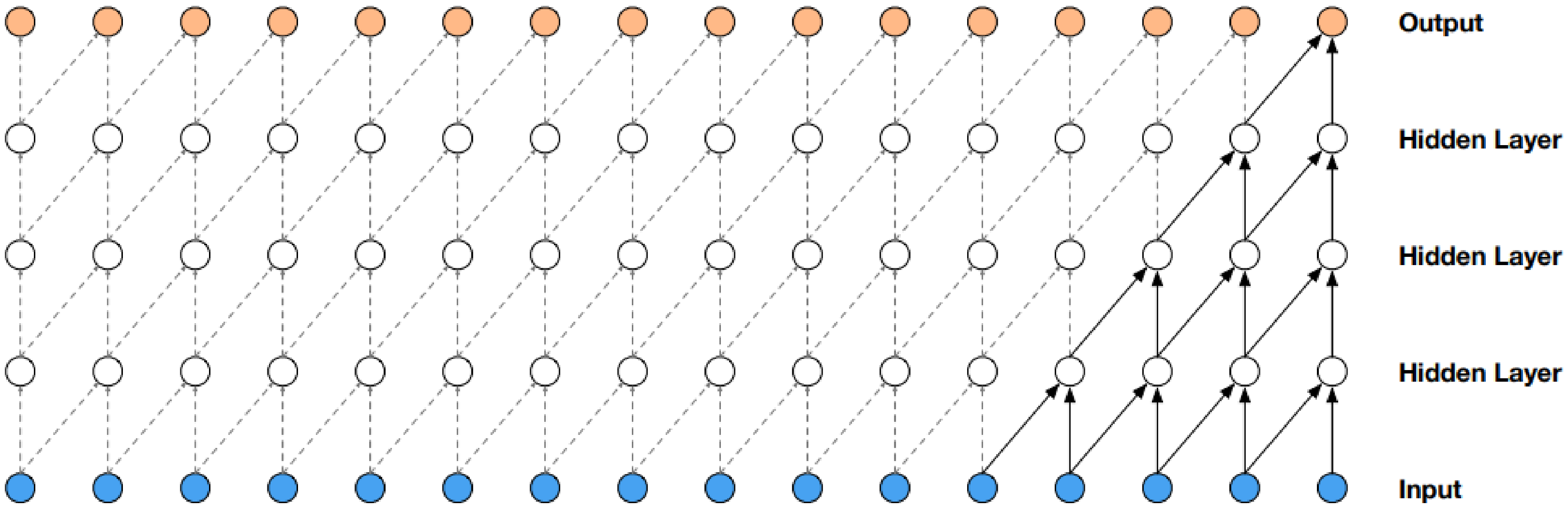


Figure 2: Visualization of a stack of causal convolutional layers.

[WAVENET: A GENERATIVE MODEL FOR RAW AUDIO]

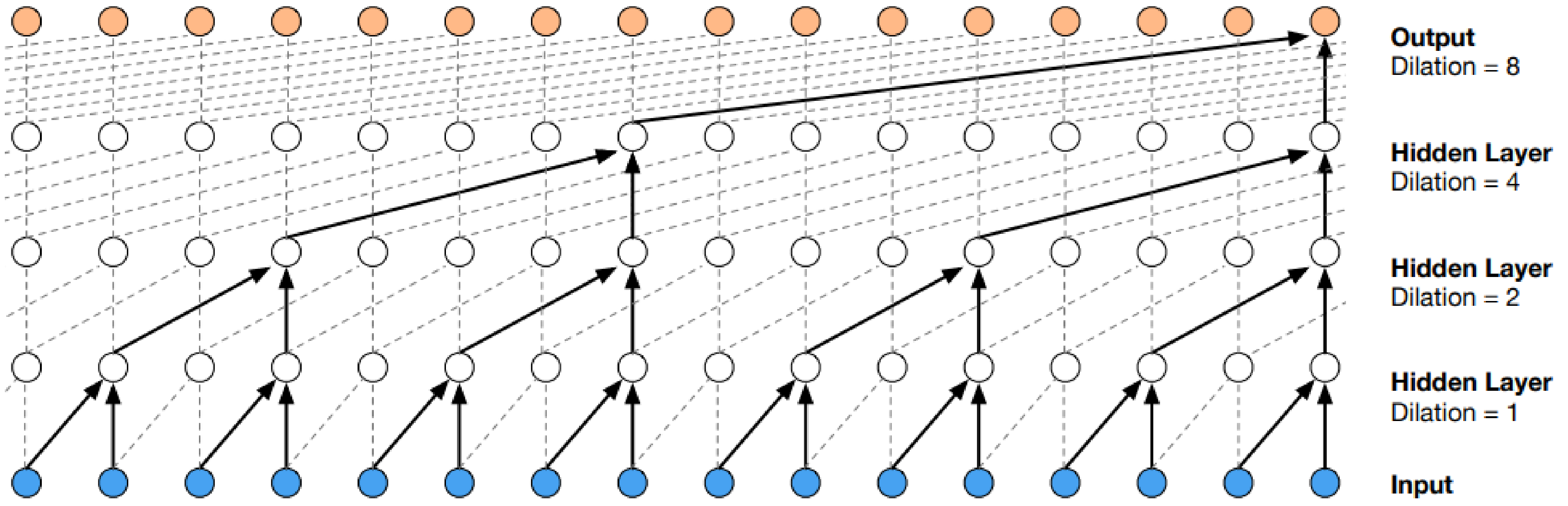
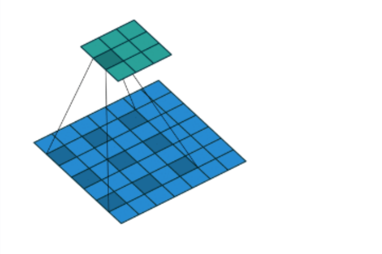


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

Dilated convolution animations

N.B.: Blue maps are inputs, and cyan maps are outputs.



No padding, no stride, dilation

可以在causal convolutional layers里面用空洞卷积（dilated convolution），在不显著增加计算成本的情况下增加网络的感受野范围。以这个layer构建的CNN model for time-series analysis称为Temporal Convolutional Networks (TCN)