EI320A(3) 深度學習使用 Python

Instructors

Tipajin Thaipisutikul (t.greentip@gmail.com)

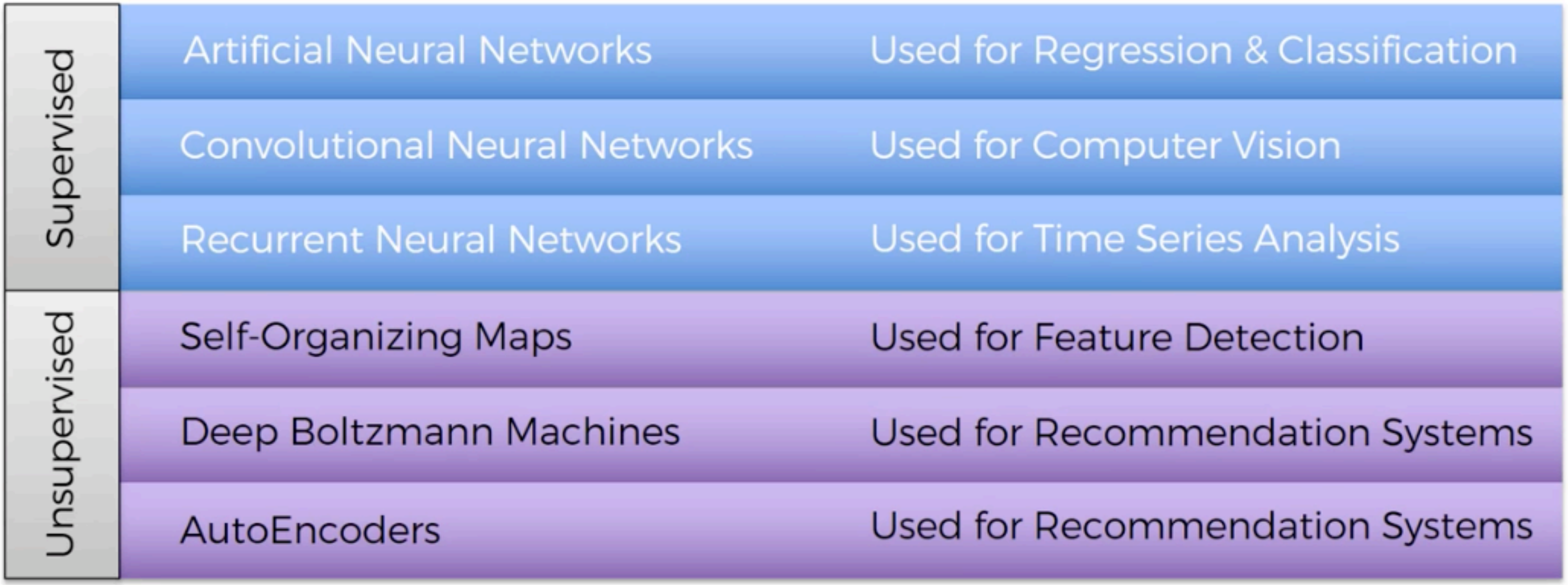
Prof. Huang-Chia Shih (hcshih@Saturn.yzu.edu.tw)

Week	Date	Content	Note	Total
1	2/26	Welcome to the course	Homework (1)	1
2	3/5	Crash Course of Python, NumPy, Pandas, and Matplotlib	In class hands-on (4)	5
3	3/12	Get to know about Data, ML: Classification Models	In class hands-on (5)	10
4	3/19	ML: Regression Models	In class hands-on (5)	15
5	3/26	ML: Clustering/Apriori Models	In class hands-on (5)	20
6	4/2	Holiday		
7	4/9	Introduction to Deep Learning (ANN)		
8	4/16	ANN Labs, Introduction to Convolutional Neural Network (CNN)	In class hands-on (10)	30
9	4/23	Convolutional Neural Network (CNN) & CNN Labs	In class hands-on (5)	35
10	**4/30**	**Introduction to Recurrent Neural Network (RNN)**	**In class hands-on (5)**	**40**
11	5/7	Recurrent Neural Network (RNN) & RNN Labs	In class hands-on (5)	45
12	5/14	Project Proposal Presentation	Proposal Presentation (10)	55
13	5/21	Wrap Up all ANN, CNN, RNN	In class hands-on (5)	60
14	5/28	NLP & Attention Neural Network	In class hands-on (5)	65
15	6/4	Generative Adversarial Network (GAN)	In class hands-on (5)	70
16	6/11	Reinforcement Learning (RL)	In class hands-on (5)	75
17	6/18	Final Project Presentation	Final Presentation (30)	105

RNNs Plan of Attack

- The idea behind Recurrent Neural Networks
- The Vanishing Gradient Problem for RNNs
- Long-Short Term Memory
- Practical Intuition

What are Recurrent Neural Networks?



What are Recurrent Neural Networks?

Frontal lope: Personal Behavior, working memory

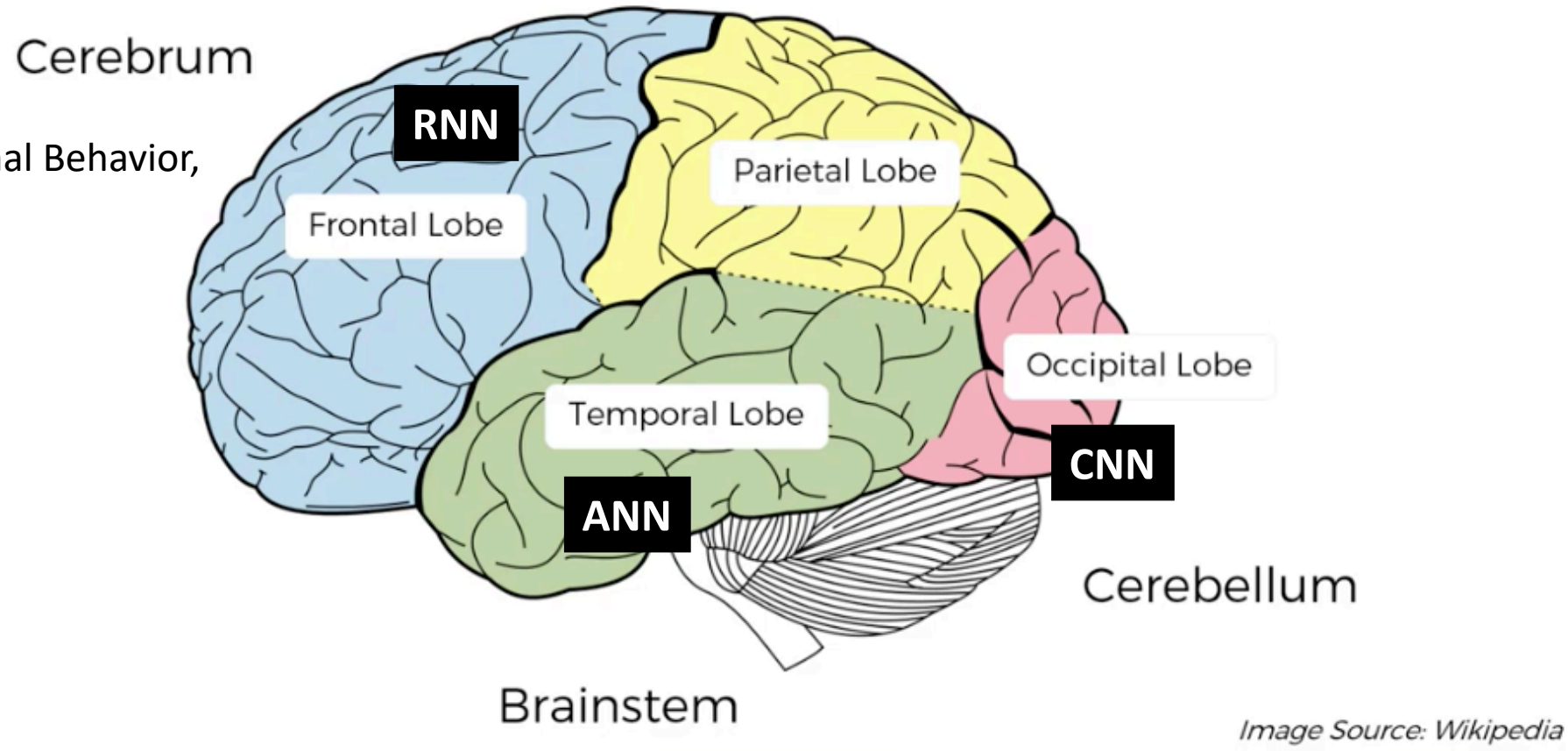
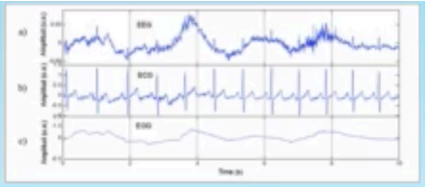


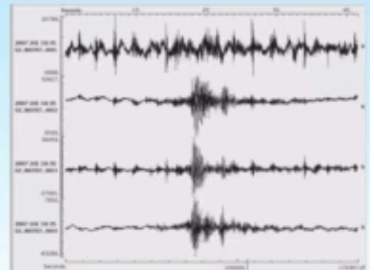
Image Source: Wikipedia

Time Series & Sequence Problems

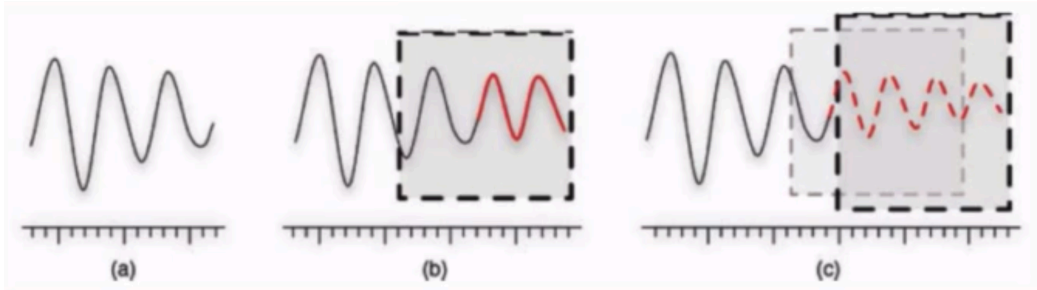
- Stock market
- Music
- Video Games
- Website monitoring and analytics
- Biosensors, biosignals (EEG, EKG, ECG, Wearables, etc…)



- Energy Monitoring
- IoT
- Earthquakes
- Weather
- Sunspots

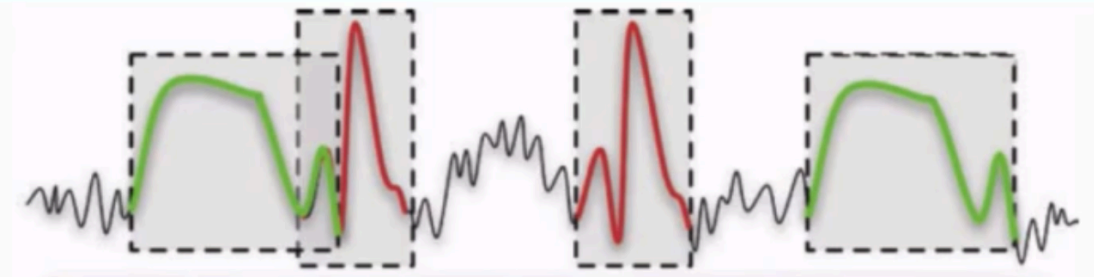


Use Past information to predict the future value



Anomaly Detection (Detect deviation from standard behvior)

Find Recurring Patterns (Can be classification task)

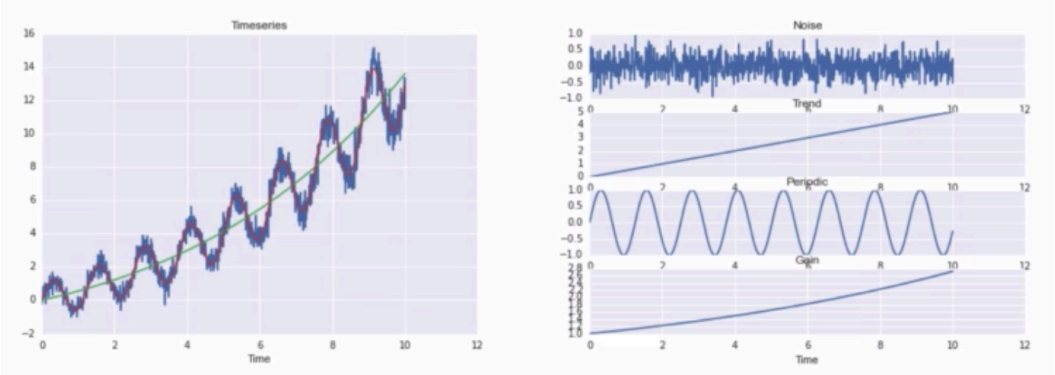


Time Series & Sequence Problems

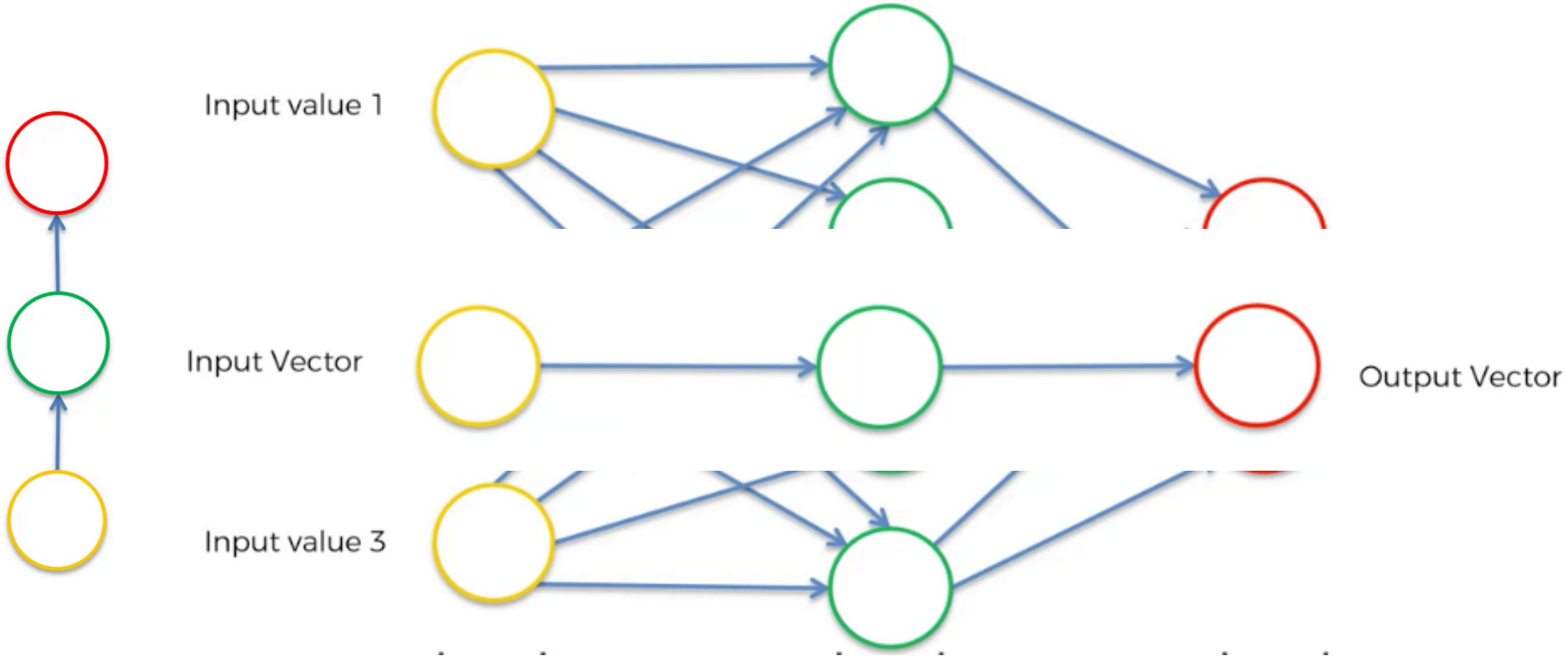
Caution Required

- Ordered Events
- Correlation in Time
- Periodicity
- Past….Future…..

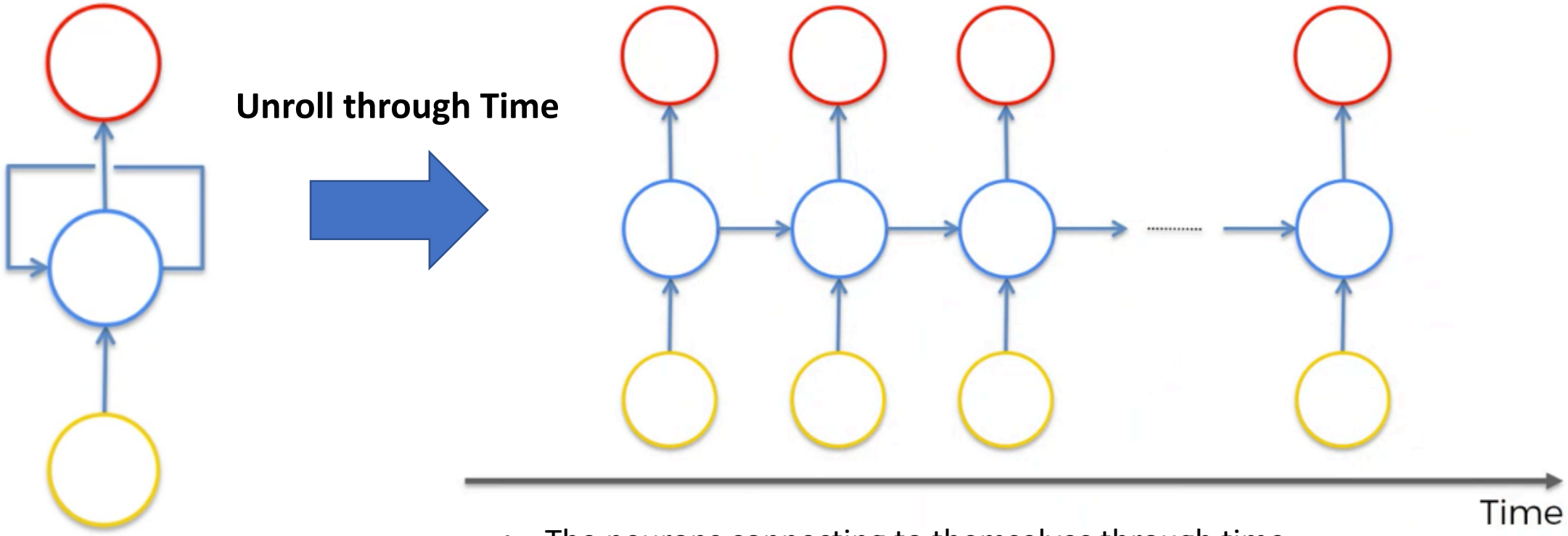
Trend and Seasonality (DOW, TOD)



What are Recurrent Neural Networks (RNN)?

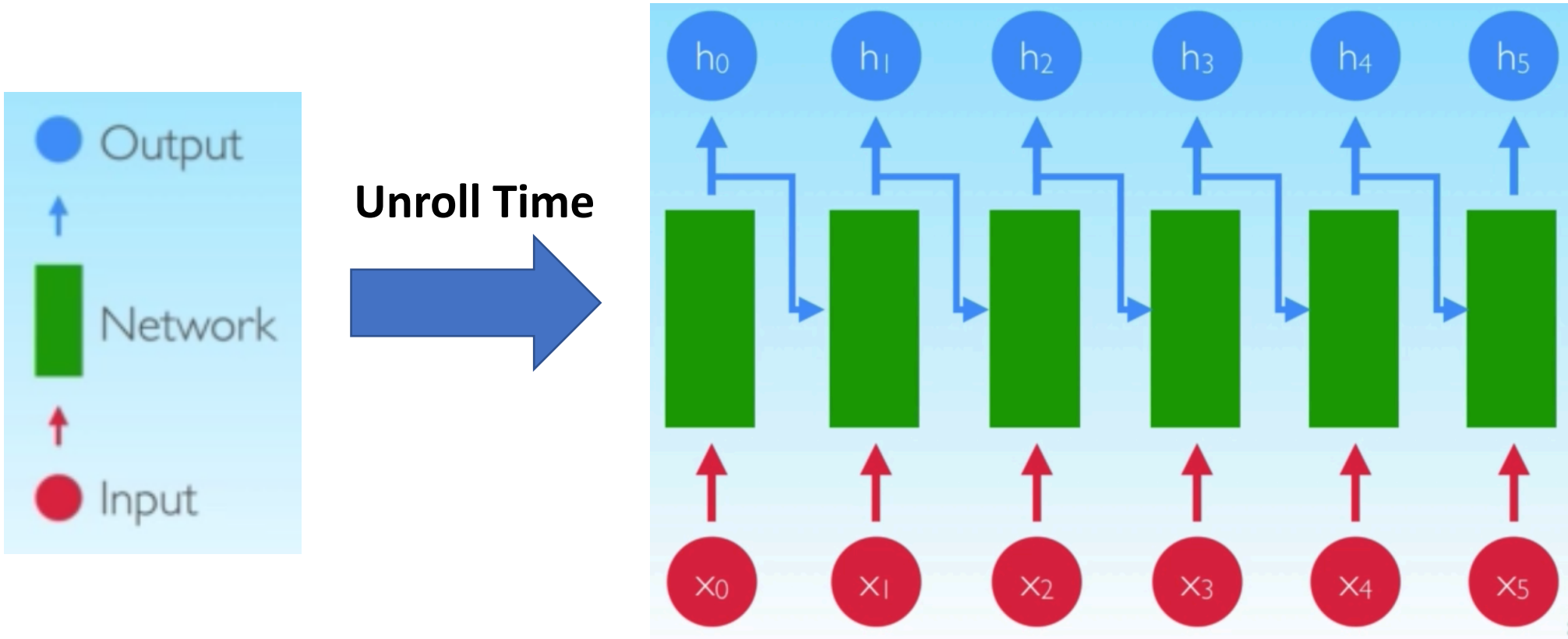


What are Recurrent Neural Networks (RNN)?

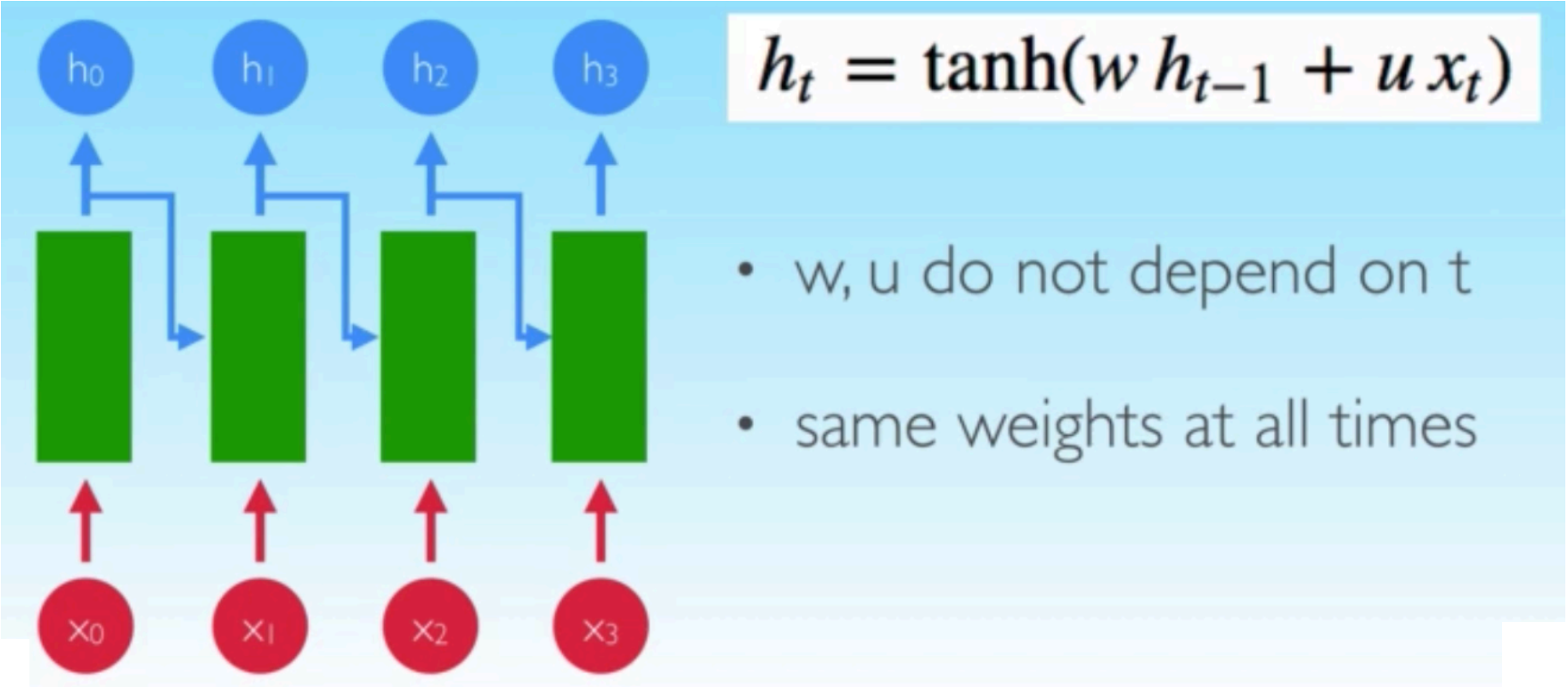


- The neurons connecting to themselves through time
- Connections between units form a directed cycle

What are Recurrent Neural Networks (RNN)?

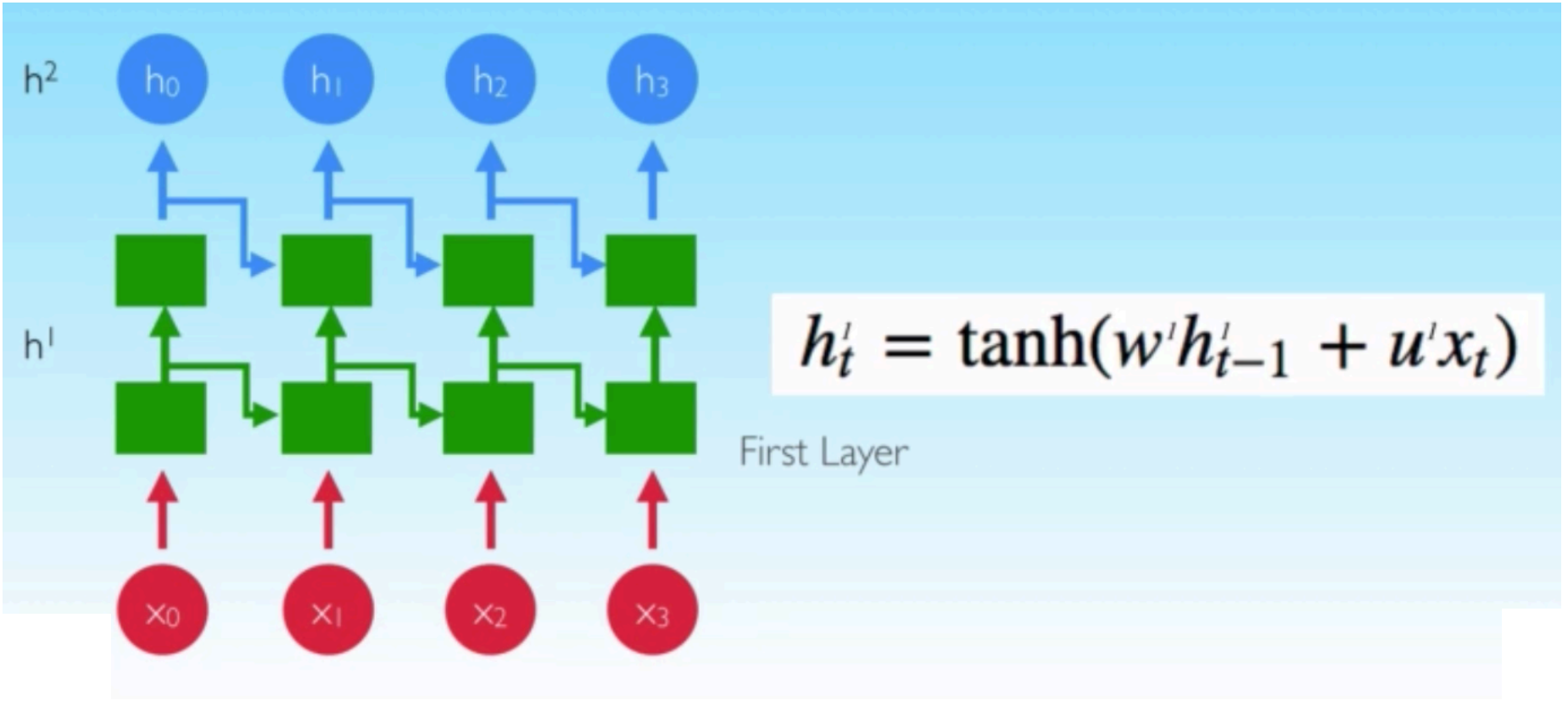


What are Recurrent Neural Networks (RNN)?



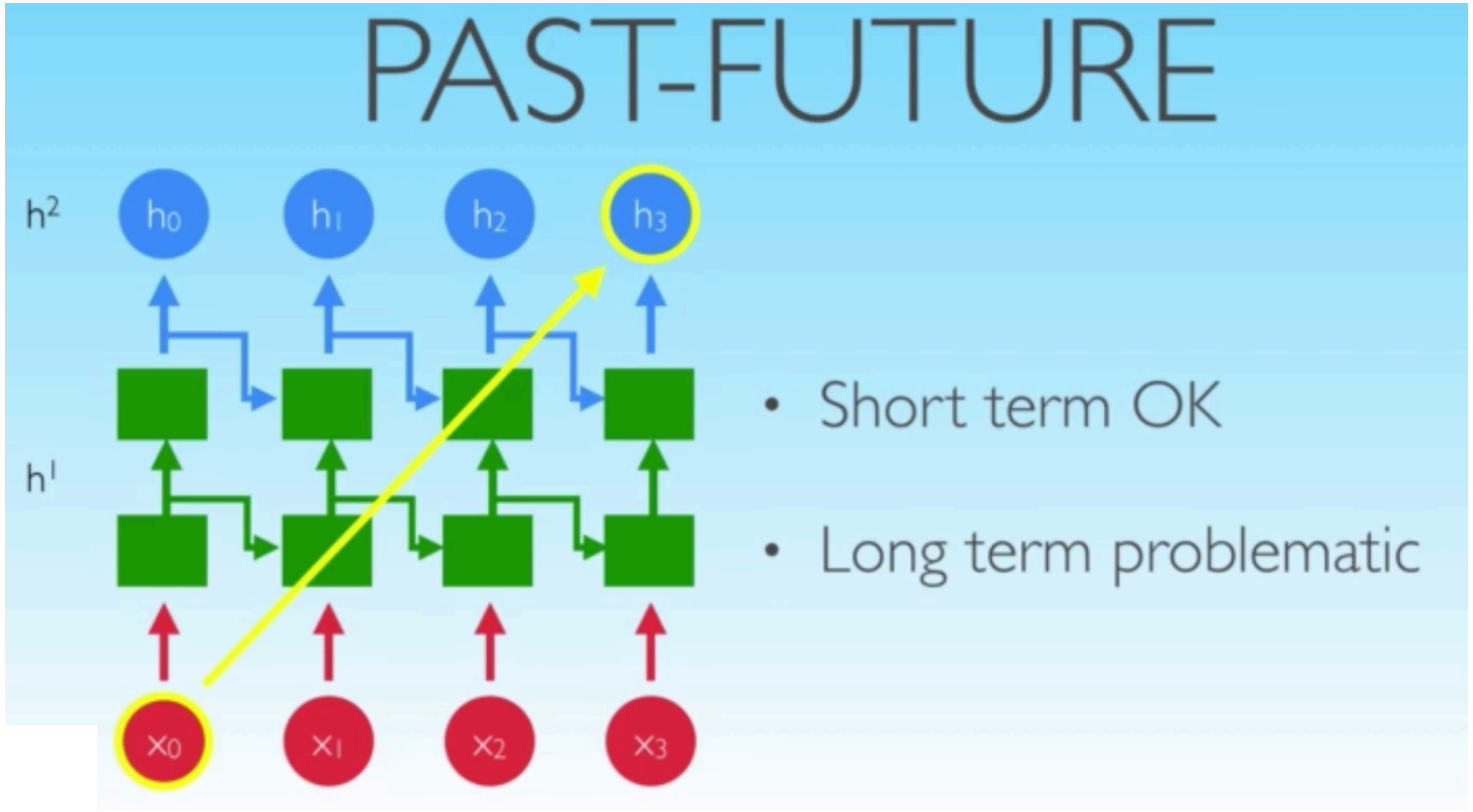
What are Recurrent Neural Networks (RNN)?

Deep RNN

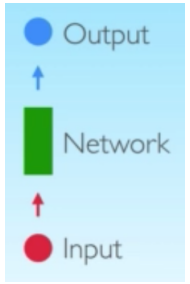


What are Recurrent Neural Networks (RNN)?

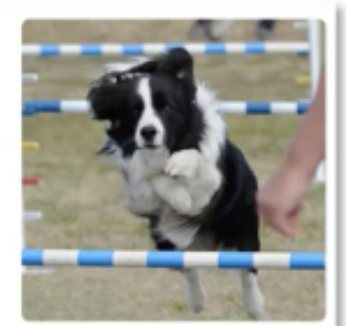
Problem with RNN



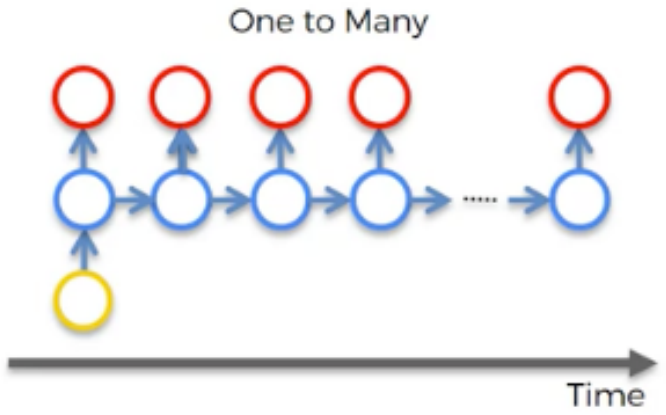
Sequence Problem

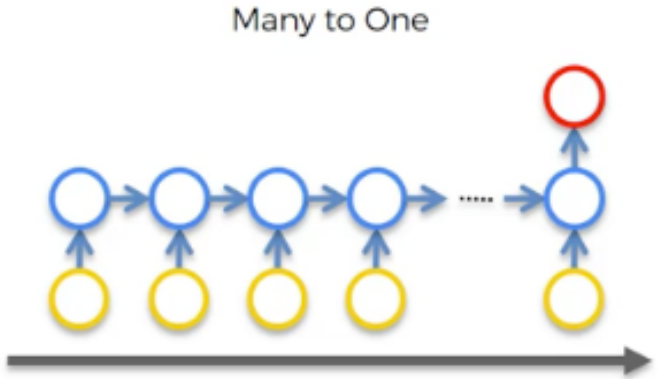


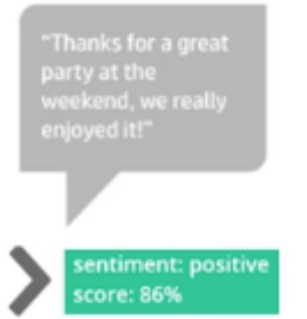
1-1: Point-Wise Regression/Classification



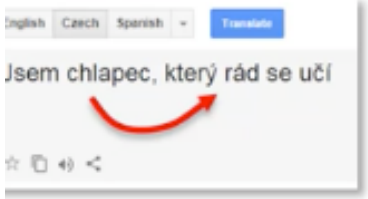
karpathy.github.io

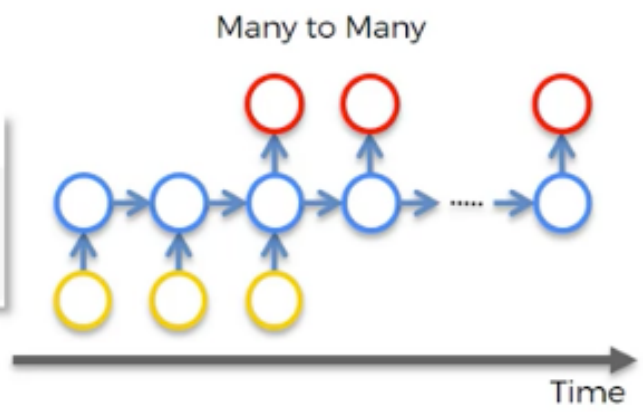


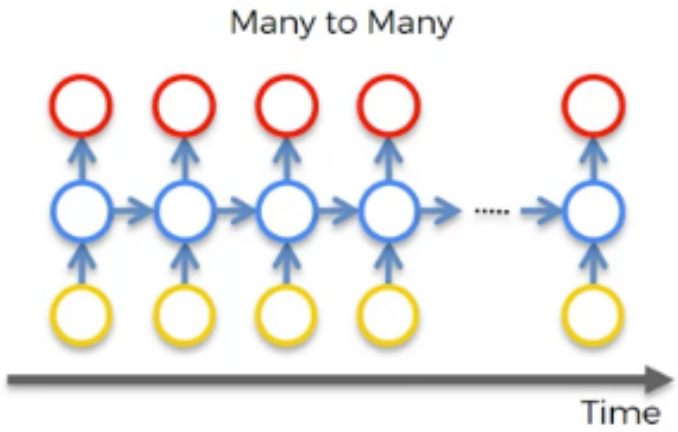




dev.havenondemand.com





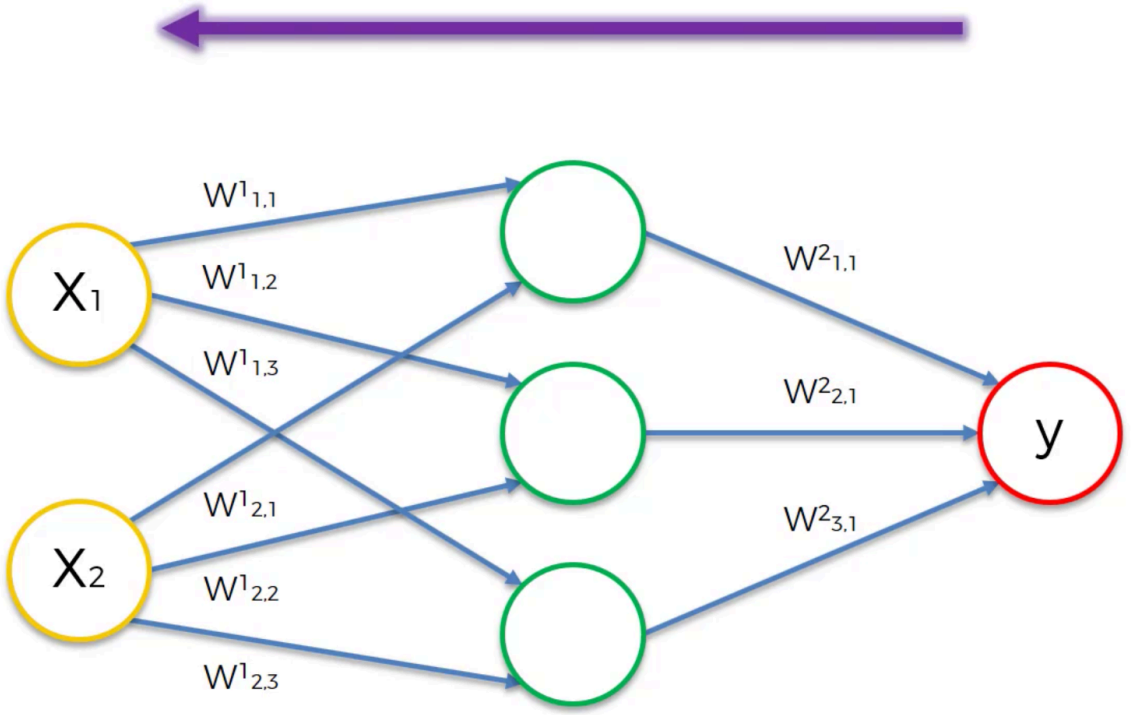


Subtitle movie (describe frame to text)

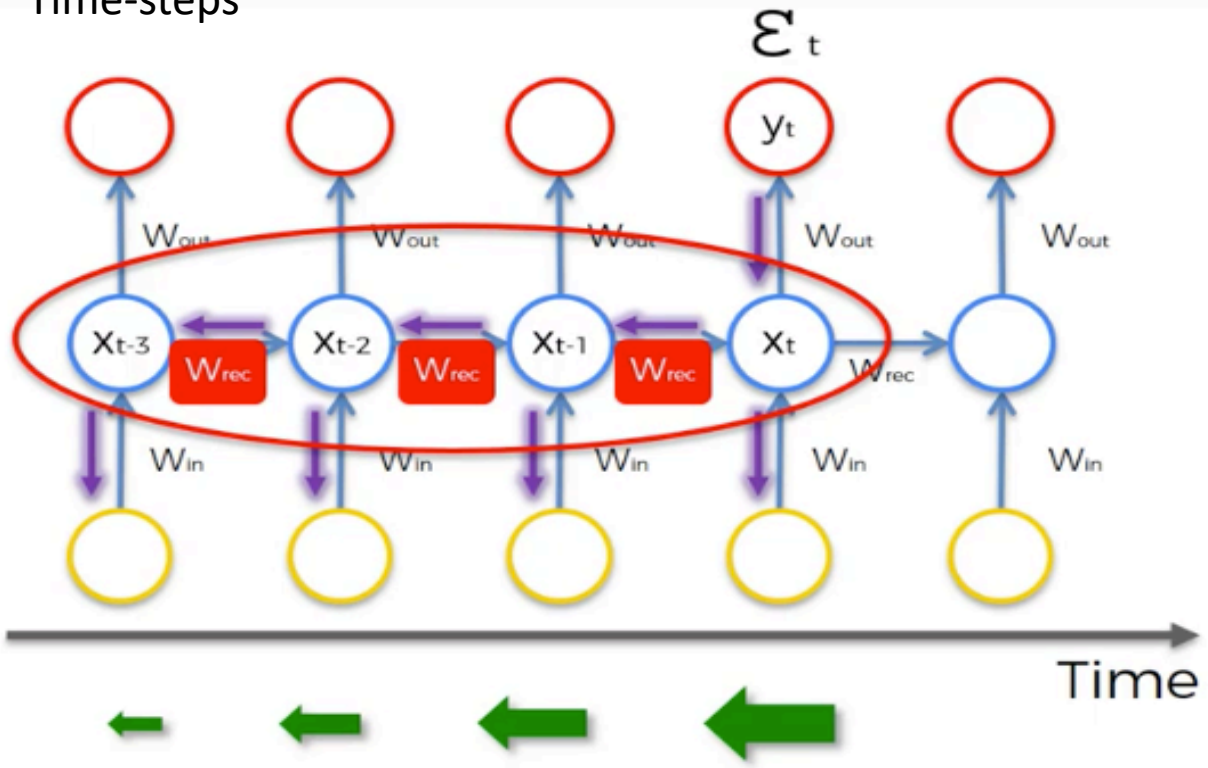
RNNs Plan of Attack

- The idea behind Recurrent Neural Networks
- The Vanishing Gradient Problem for RNNs
- Long-Short Term Memory
- Practical Intuition

The Vanishing Gradient Problem for RNNs



The Vanishing Gradient Problem for RNNs



$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \mathcal{E}_t}{\partial \theta} \quad (3)$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right) \quad (4)$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \ge i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \ge i > k} \mathbf{W}_{rec}^T diag(\sigma'(\mathbf{x}_{i-1})) \quad (5)$$

Weight Recurrent

W_{rec} ~ small ⇨ Vanishing

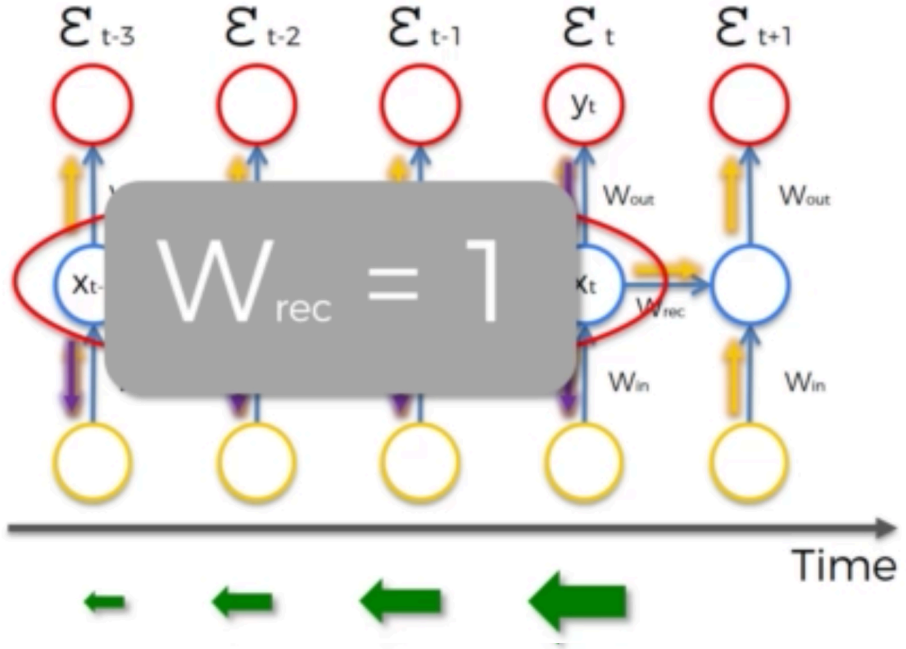
W_{rec} ~ large ⇨ Exploding

RNNs Plan of Attack

- The idea behind Recurrent Neural Networks
- The Vanishing Gradient Problem for RNNs
- Long-Short Term Memory
- Practical Intuition

Long-Short Term Memory (LSTM)

- A bit of history
- LSTM Architecture
- Example walkthrough



$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \mathcal{E}_t}{\partial \theta} \quad (3)$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right) \quad (4)$$

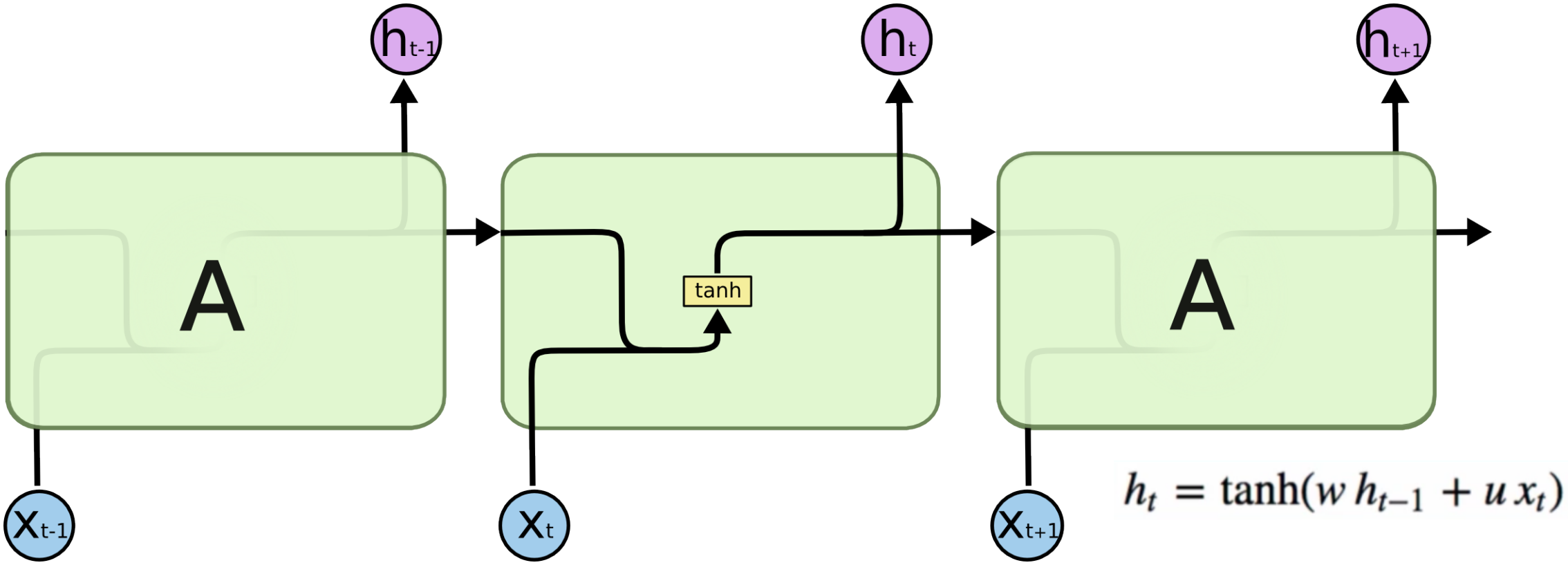
$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \ge i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \ge i > k} \mathbf{W}_{rec}^T diag(\sigma'(\mathbf{x}_{i-1})) \quad (5)$$

W_{rec} ~ small ⇨ Vanishing

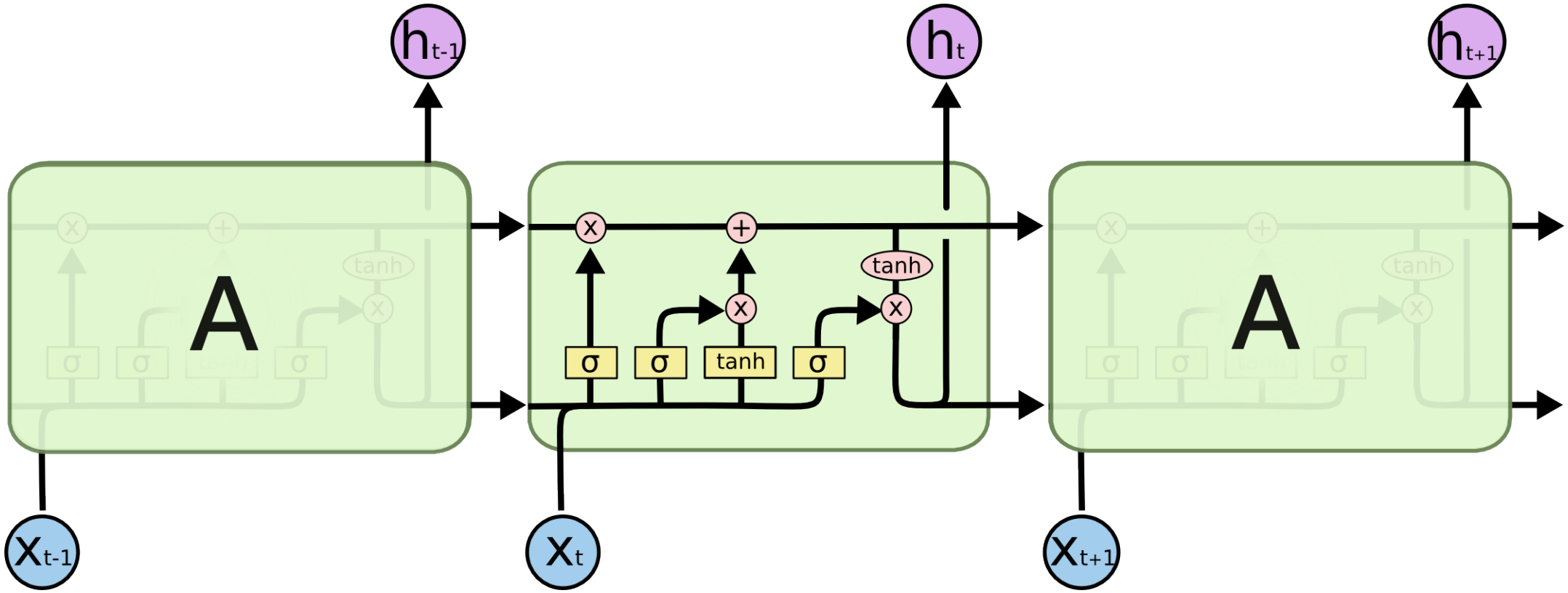
W_{rec} ~ large ⇨ Exploding

Recurrent Neural Network (RNN)

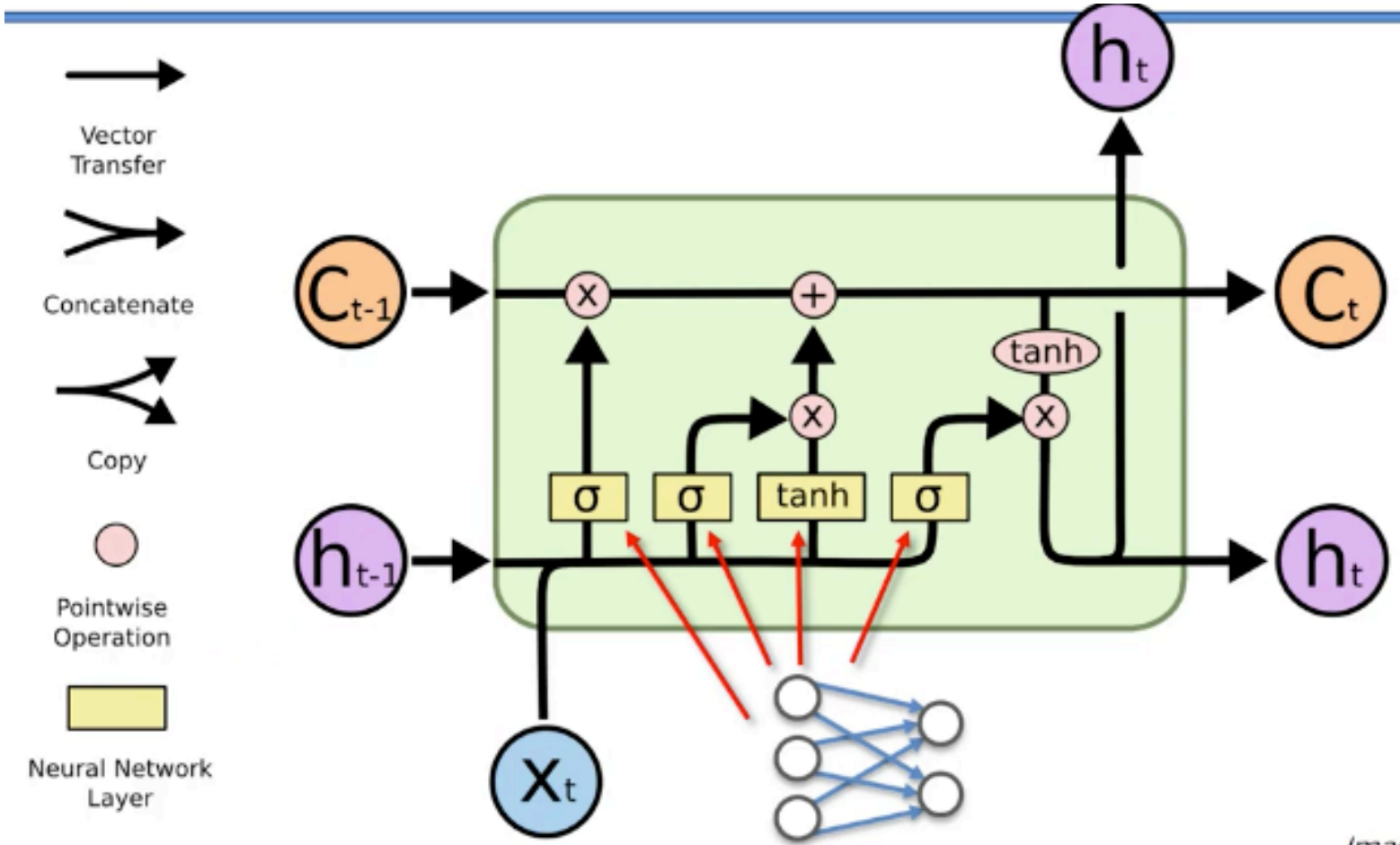
Ref: http://colah.github.io/posts/2015-08-Understanding-LSTMs/



Long-Short Term Memory (LSTM)



Long-Short Term Memory (LSTM)

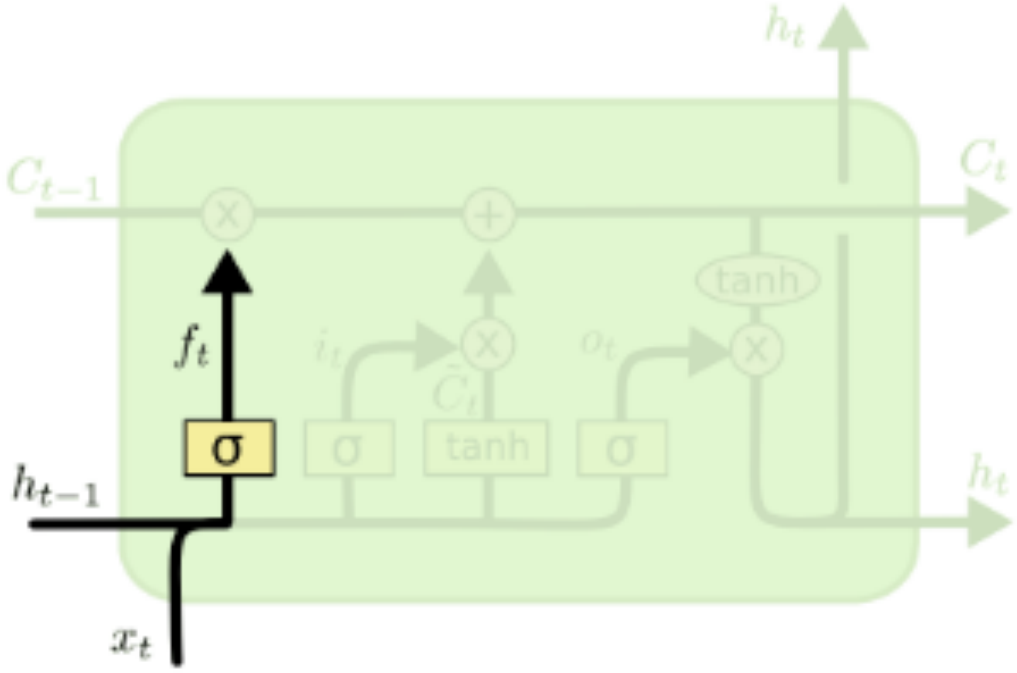


valve

T-shape

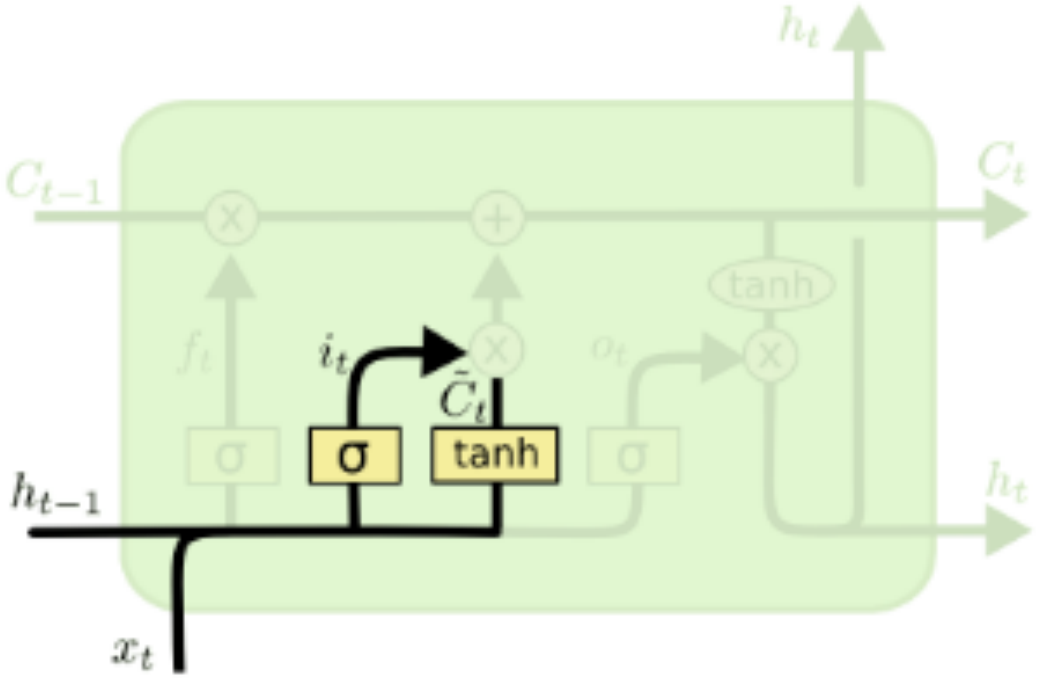
Image Source: colah.github.io

Long-Short Term Memory (LSTM)



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

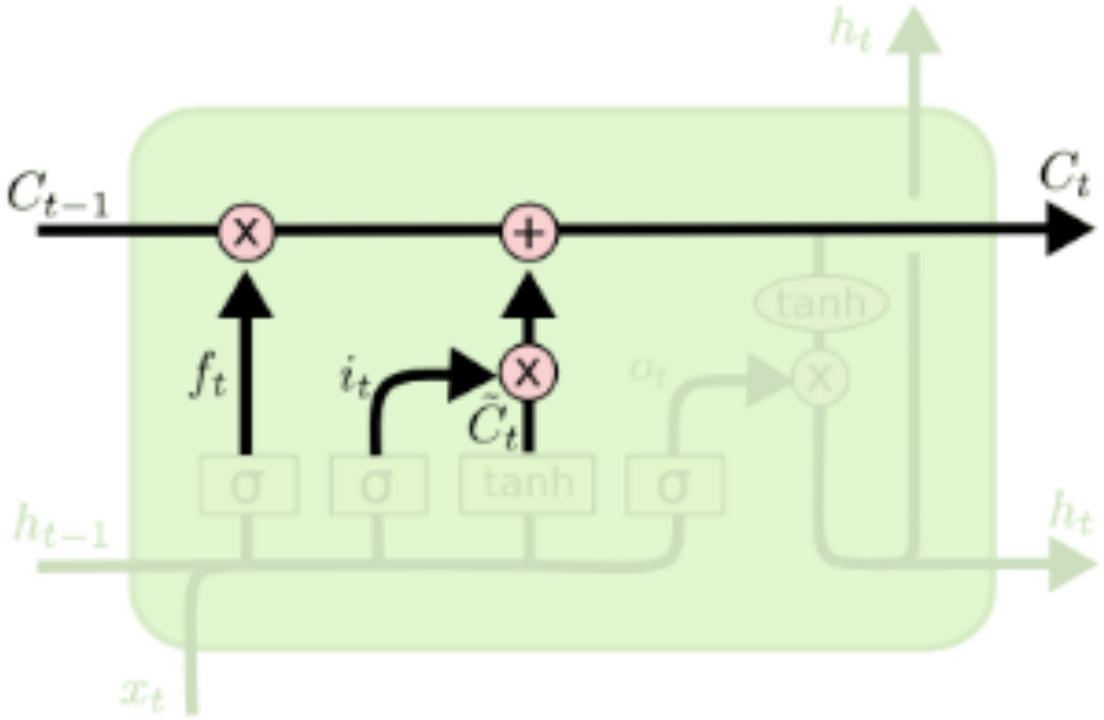
Long-Short Term Memory (LSTM)



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

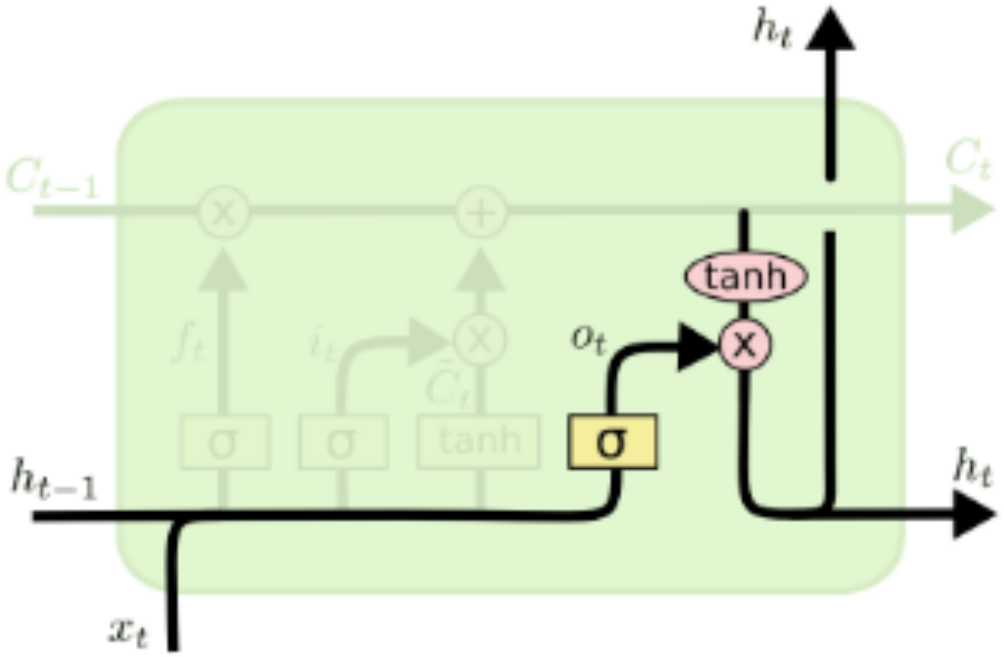
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Long-Short Term Memory (LSTM)



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Long-Short Term Memory (LSTM)



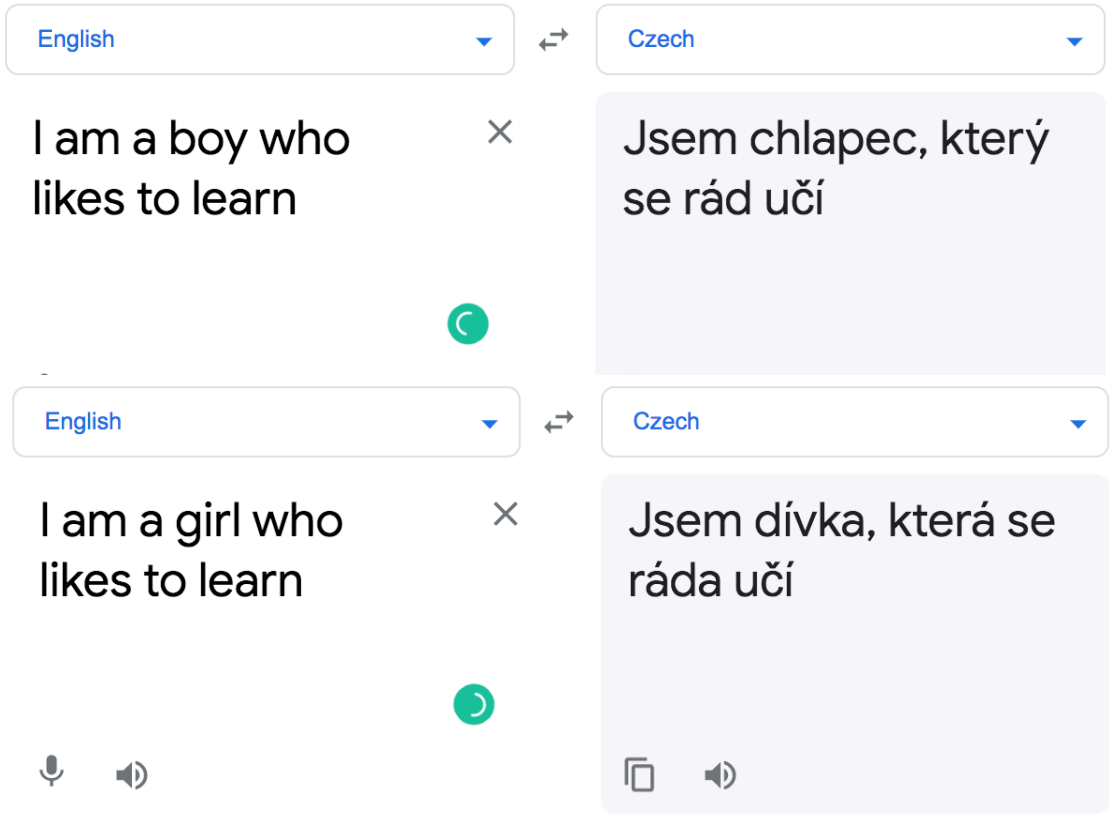
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

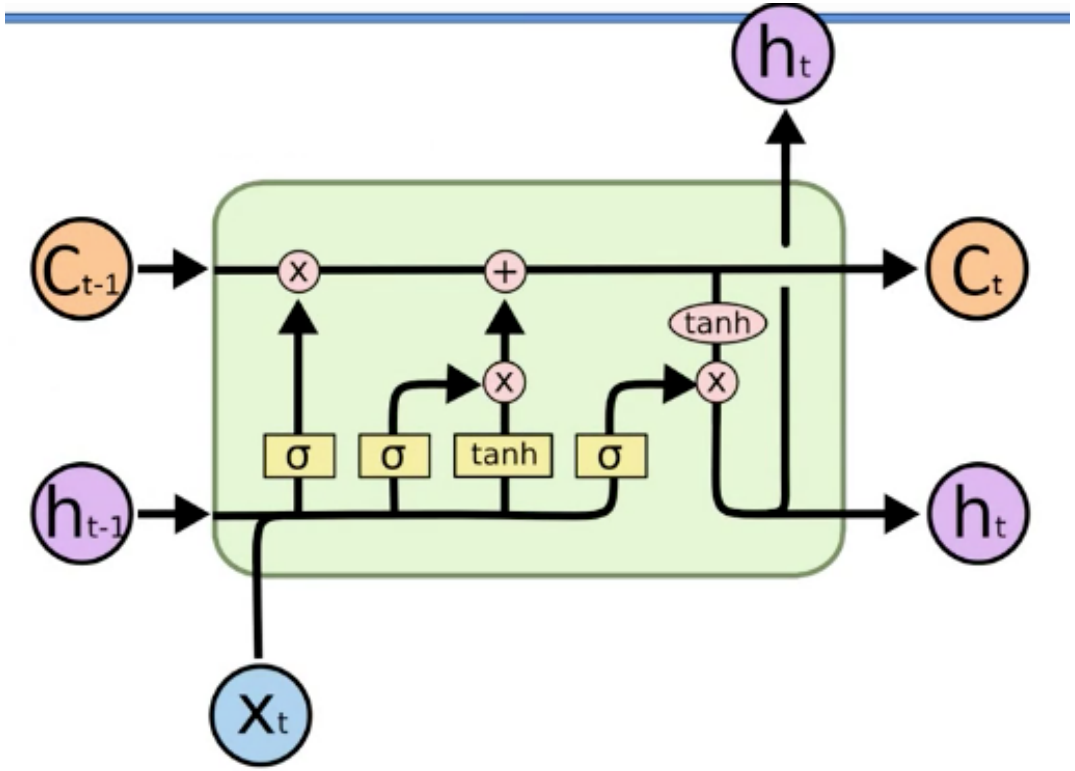
$$h_t = o_t * \tanh\left(C_t\right)$$

RNNs Plan of Attack

- The idea behind Recurrent Neural Networks
- The Vanishing Gradient Problem for RNNs
- Long-Short Term Memory
- Practical Intuition

Long-Short Term Memory (LSTM)





If we have a new subject, then we forget the old subject and we put the new info as much as we can in the memory cell

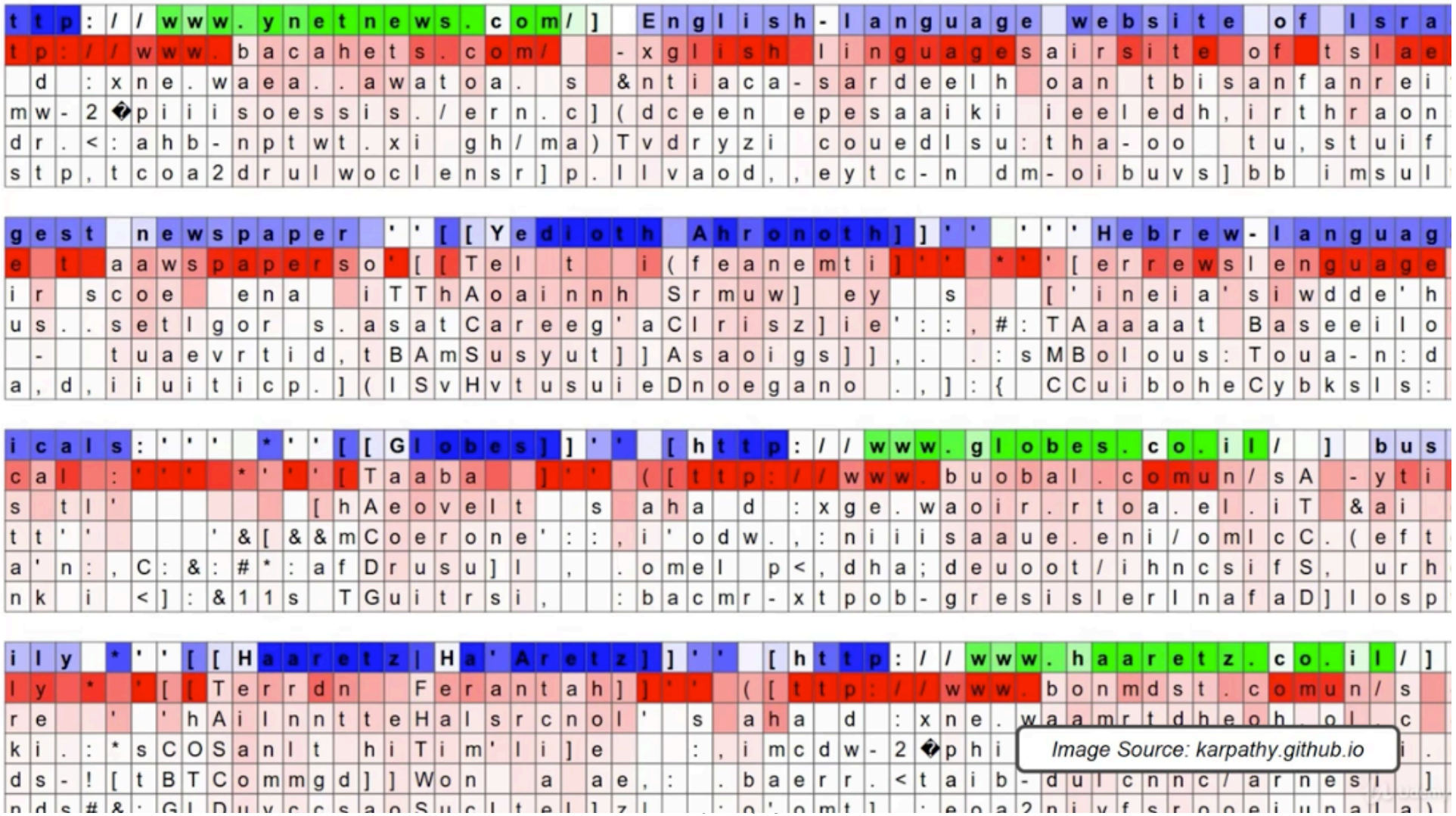
RNN that reads and predict the next character.

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Practical Intuition

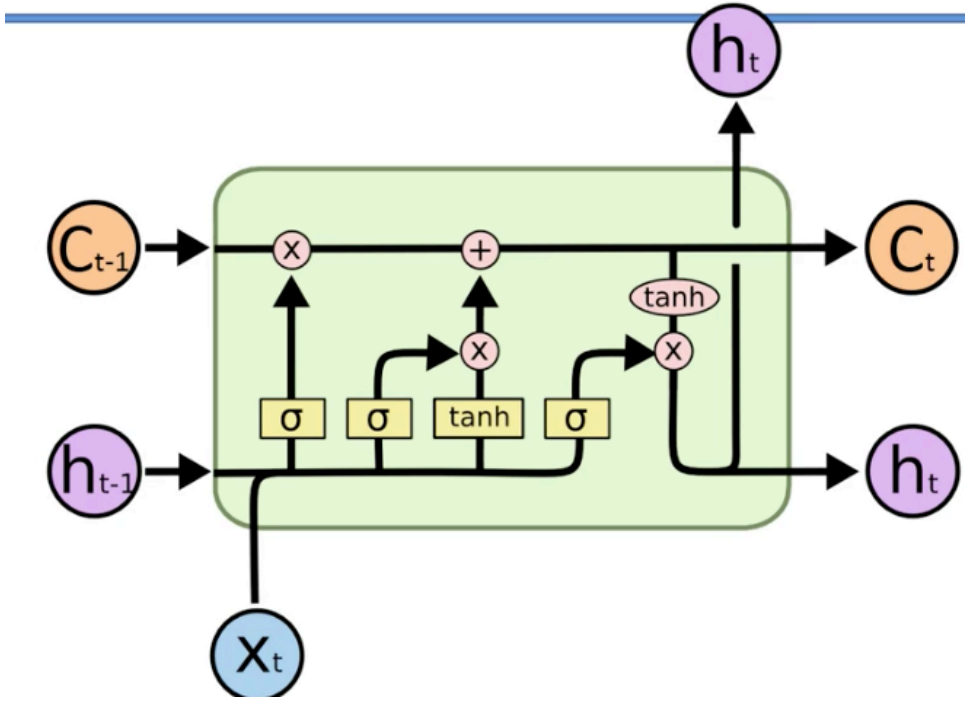
Green = activated
Blue = not-activated

Red = High confidence with the prediction



LSTM Intuition Explanation

Classification: In Relationship/Break up



$h_{t-1} = broke\ up$

x_1

x_2

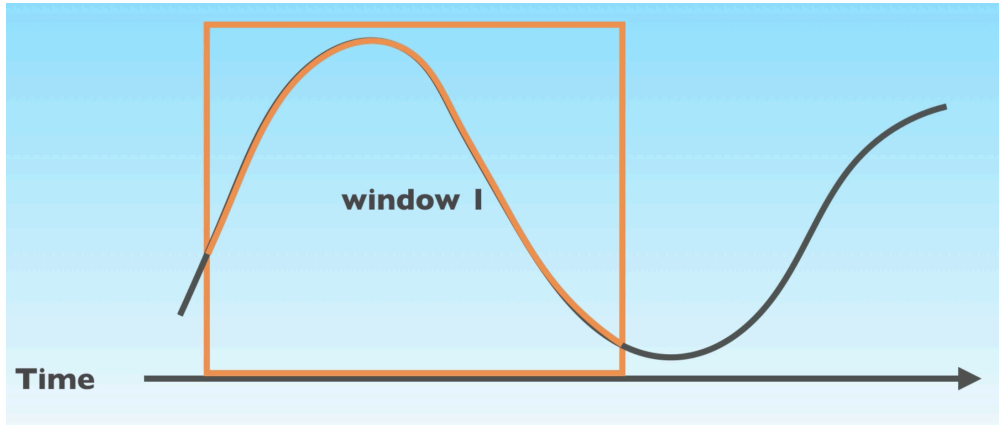
x_3

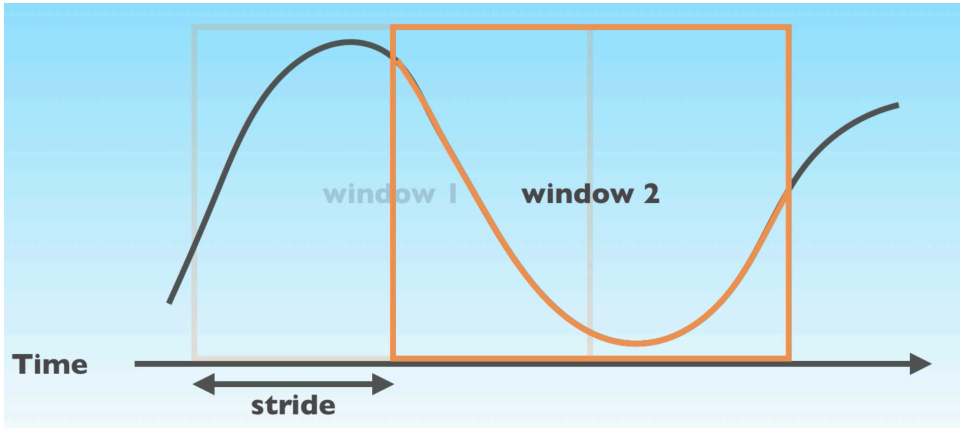
Code Demo

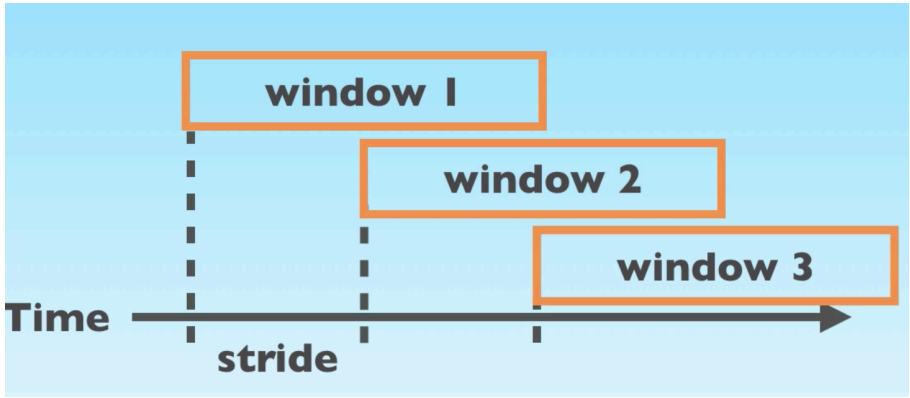
1. Fully Connected Predictor
2. Recurrent Predictor (LSTM)

Rolling Windows

To extract the features from time series

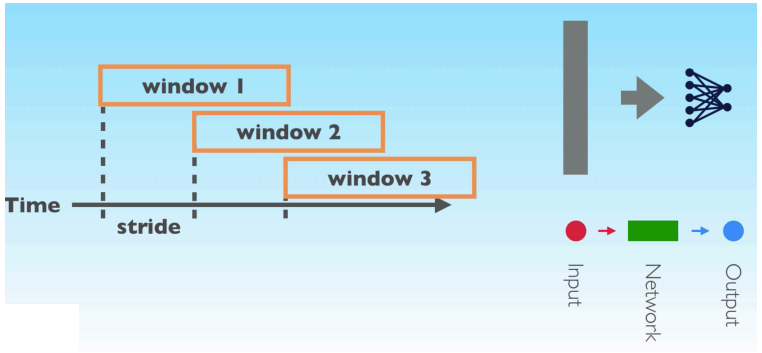






Code Demo

3. Fully Connected on Windows
4. LSTM on Windows with 1 timestep 12 features



5. LSTM on Windows with 12 timesteps 1 feature