Recurrent Neural Networks

PHYS591000 2022.04.27

Warming up

- As usual, take 3 mins to introduce yourself to your teammate for this week!
  – “Are you done with your midterms?”
  – “Only 4 labs left! Let’s work together to get it through!”

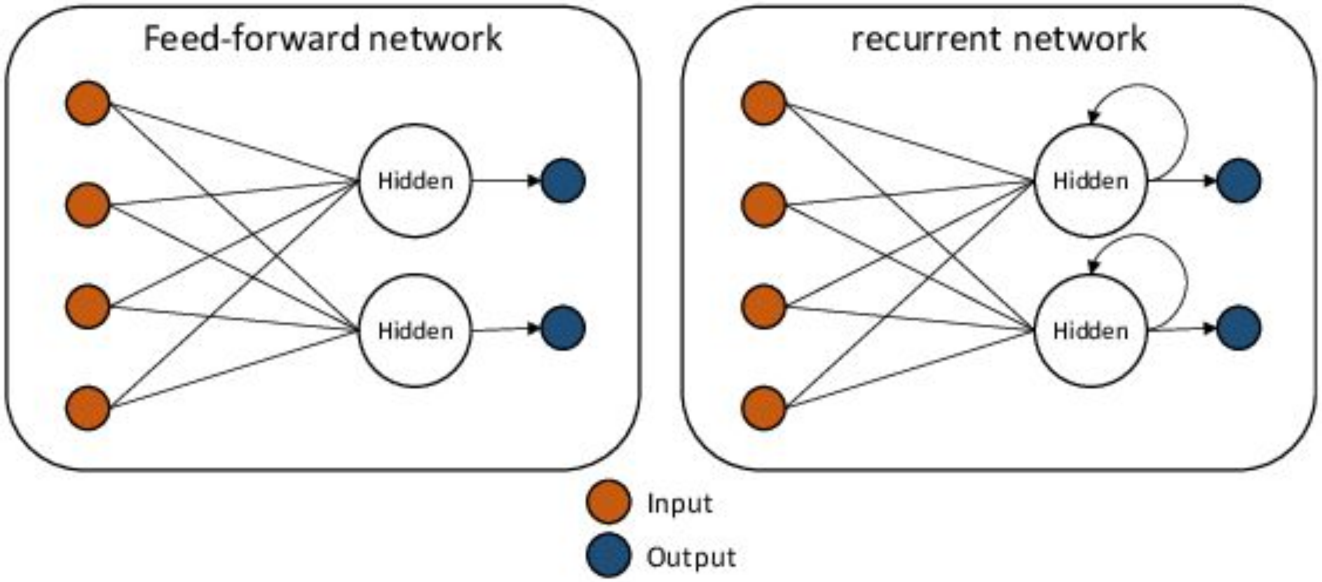
Outline

- Today we are going to talk about yet another kind of NN, the Recurrent Neural Networks (RNN).

Ref: Lecture 10 of CS231(2017) at Stanford (youtube)

RNN

- Recurrent = loops in processing information



RNN

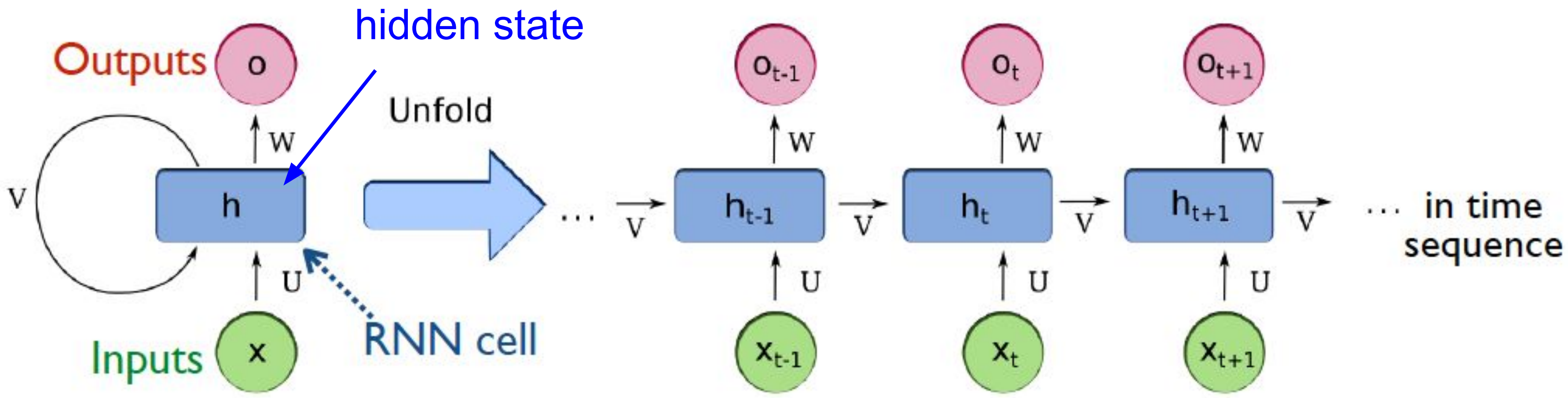
- RNN is useful for *sequential data*, data in which the order matters.
  - Time series: speech/language recognition
  - Weather/stock price prediction (from the history of data)

- The i-th output (O_i) depends on the input of this moment (t_i) and information from previous moment (t_i-1).

RNN

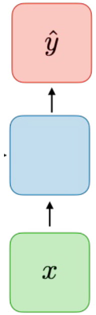
Courtesy of Prof. Kai-Feng Chen (NTU)

Classical ("Vanilla") RNN has a structure to connect the information from the previous time frame to the next, in addition to the regular inputs:



RNN Structure and Applications

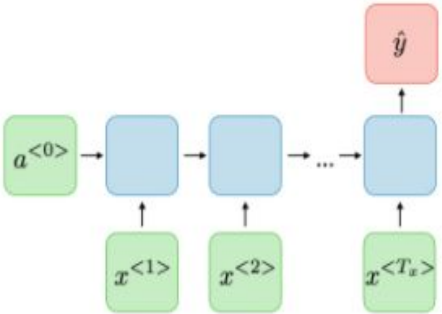
One-to-one



Binary Classification

Pass vs Fail

Many-to-One



Sentiment Classification

“There is nothing to like in this movie.”

One-to-Many

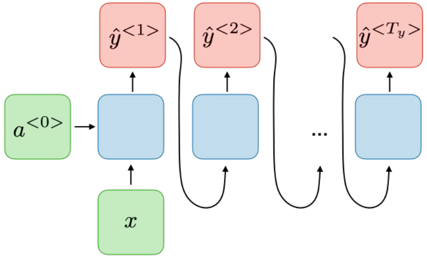
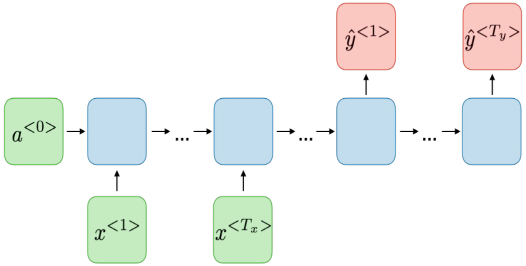


Image Captioning

A man is running.

Many-to-Many



Machine Translation

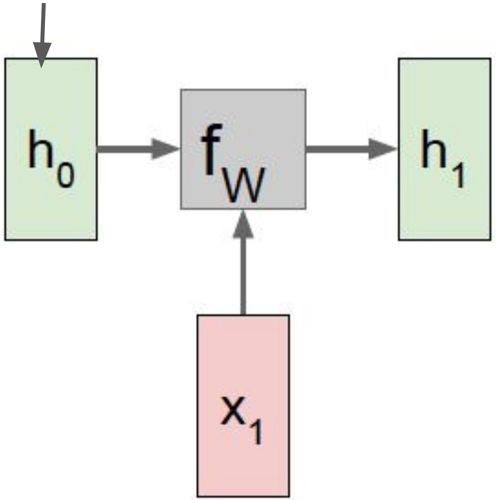
«Hey Siri, où puis-je acheter une Tesla?»

“Hey Siri,
where can I buy a Tesla?”

How does RNN work?

- The RNN cell applies a *recurrent* formula at each time step t

Initialized hidden state



$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

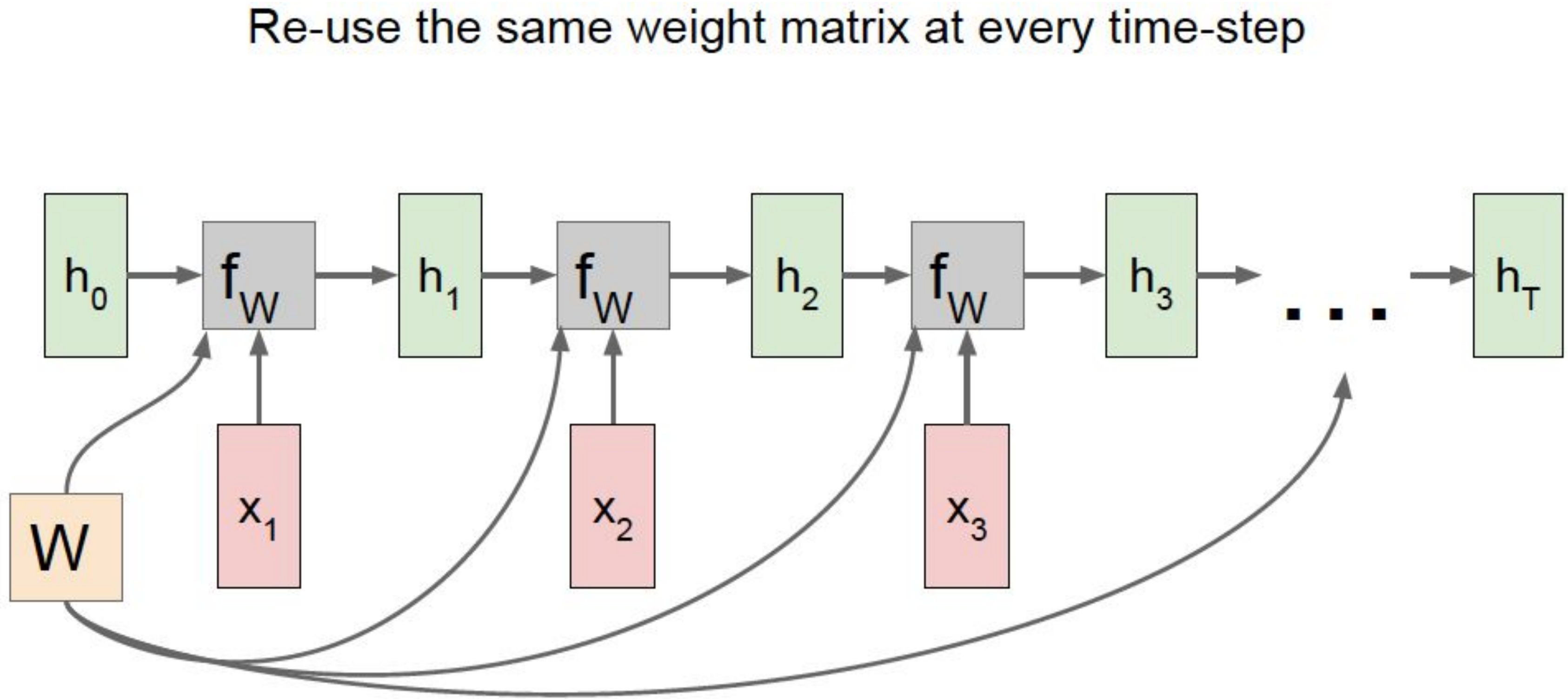
new state

some function
with parameters W

old state

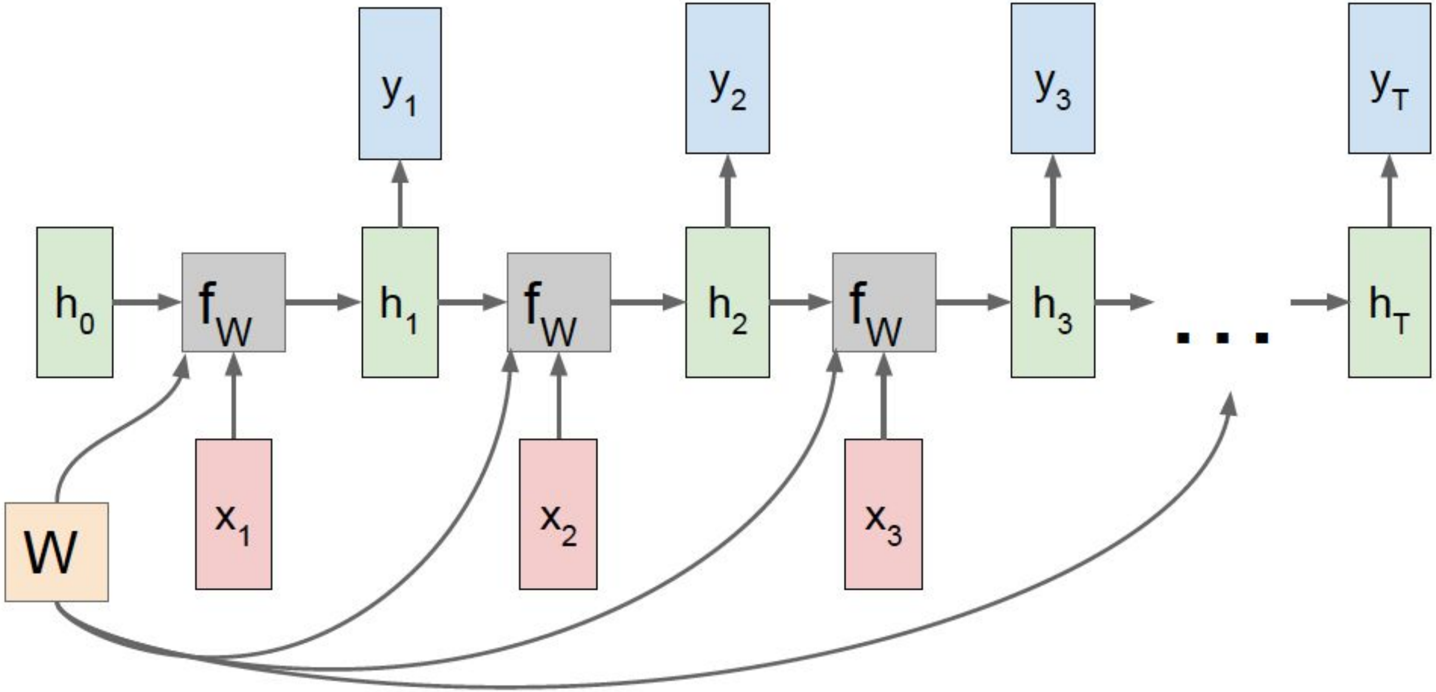
input vector at
some time step

How does RNN work?

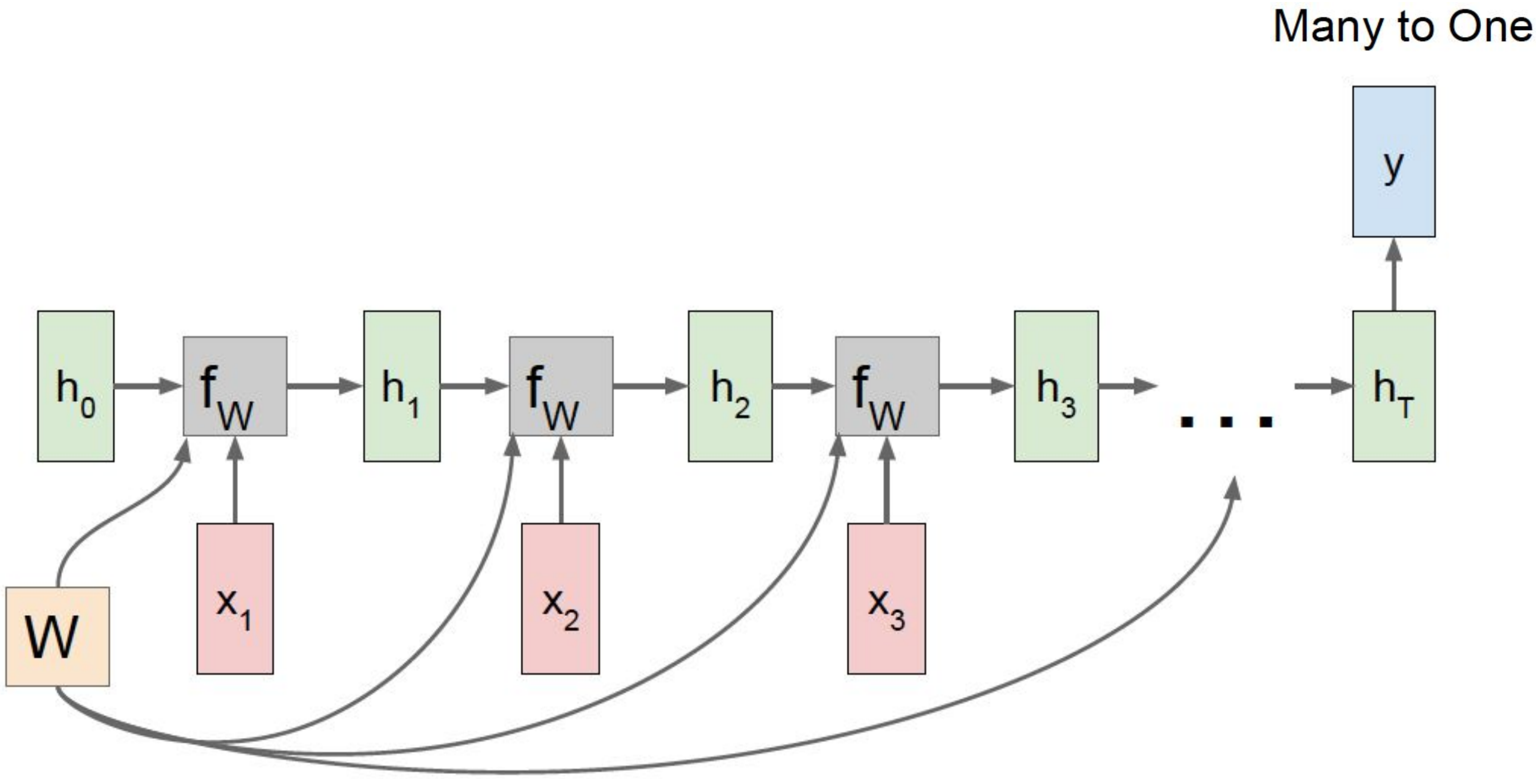


How does RNN work?

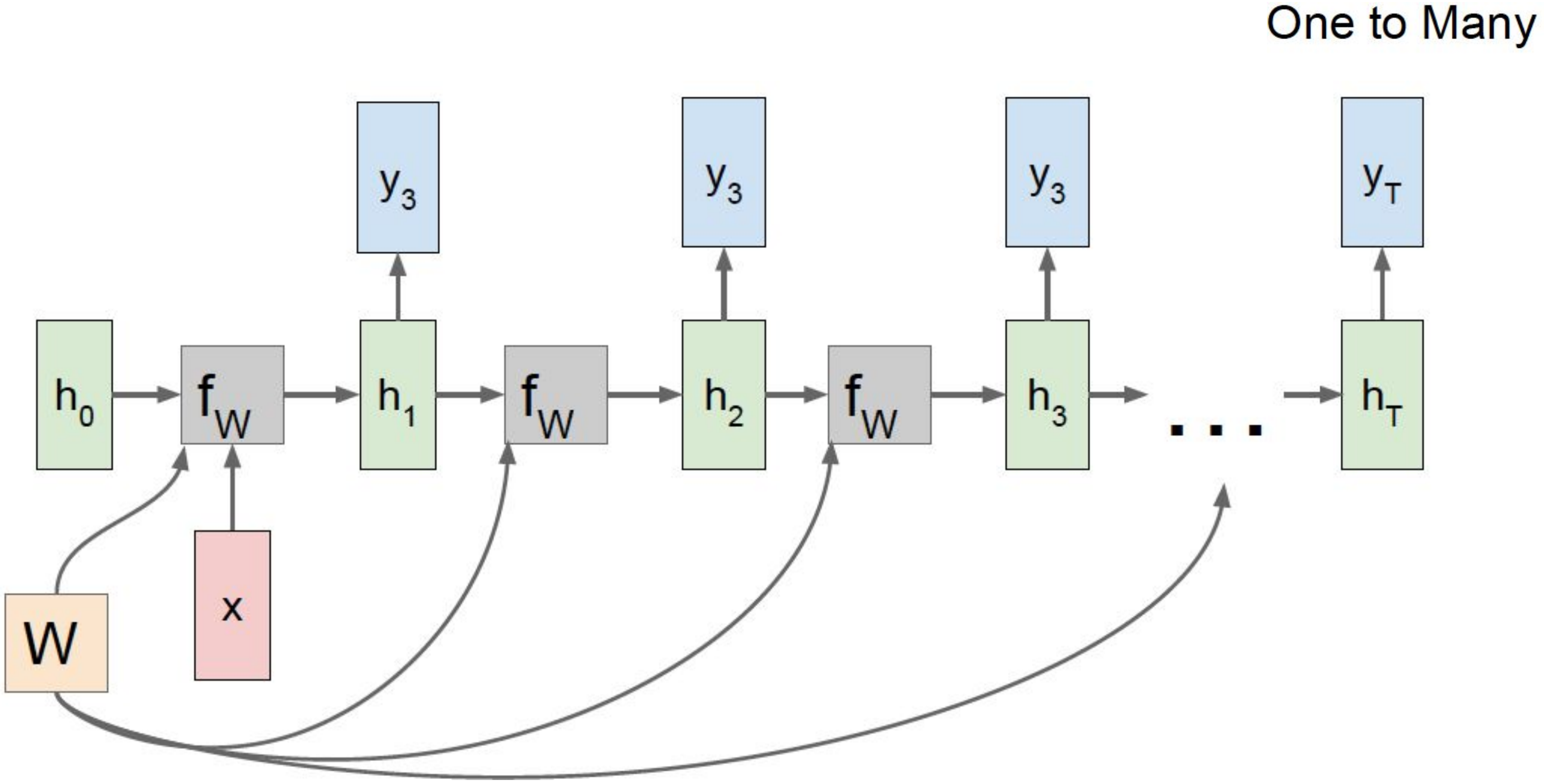
Many to Many



How does RNN work?



How does RNN work?

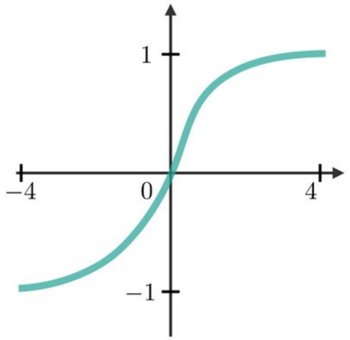


Multilayer RNN

- Hidden states (h) are a vector at at each time step, and W is now a matrix.

Tanh

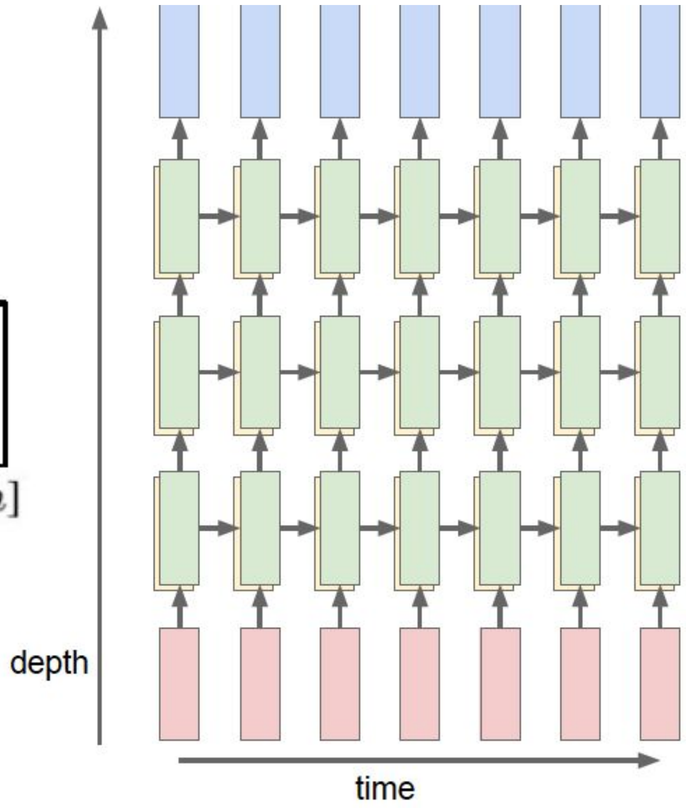
$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$



$$\boxed{h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}}$$

$h \in \mathbb{R}^n$ $\quad W^l \; [n \times 2n]$

tanh = activation function used in this example



Vanishing gradient problem (again!)

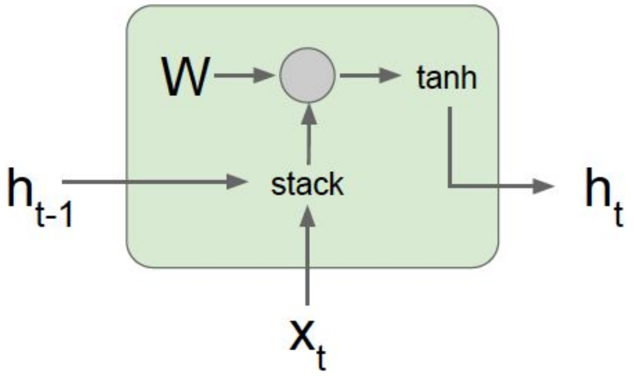
- RNN can ‘remember’ the information used before and thus make a prediction based on the previous information.
    - John grew up in **France**. Of course he can speak **French**.

Vanishing gradient problem (again!)

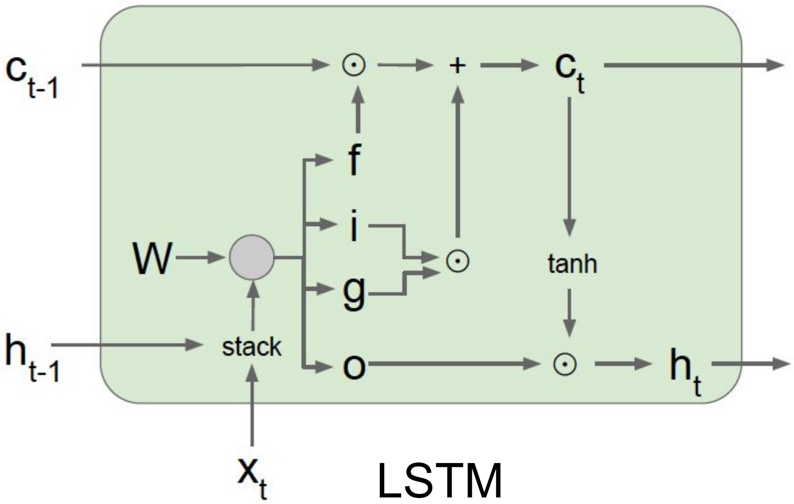
- But if the relevant input lies in much earlier time steps, the information fades away due to vanishing gradients at earlier stages (from backpropagation):
    - John grew up in France. When he was 30 he got married and …(blah blah)... Of course he can speak ______

Long Short-term Memory (LSTM)

- Long short-term memory (LSTM) is a kind of RNN:
  – Replace ‘vanilla’ RNN cell with an LSTM cell



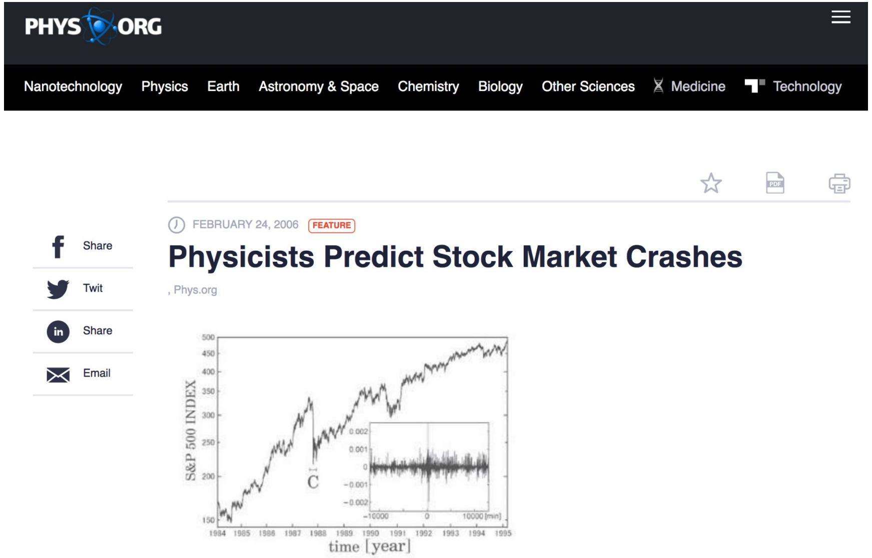
Vanilla RNN

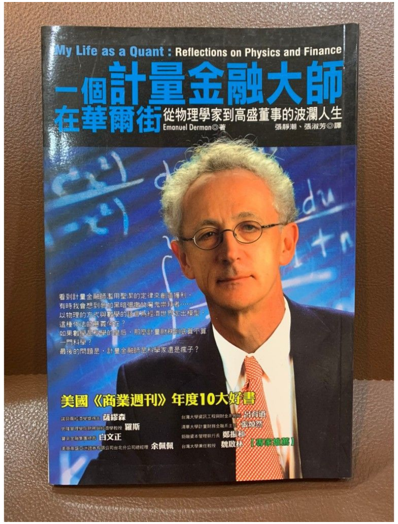


LSTM

In-class exercise for this week

- For the in-class exercise, we’re going to build an RNN to predict the price of a stock given the prices of the past 60 days.



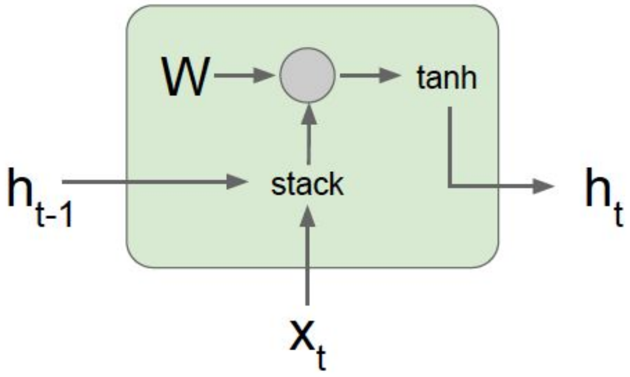


In-class exercise for this week

- Since it takes about 10 mins to train the model, let’s go to the in-class exercise first, then come back to the introduction of LSTM while we are waiting for the training.

Long Short-term Memory (LSTM)

- Long short-term memory (LSTM) is a kind of RNN:
  – Replace ‘vanilla’ RNN cell with an LSTM cell

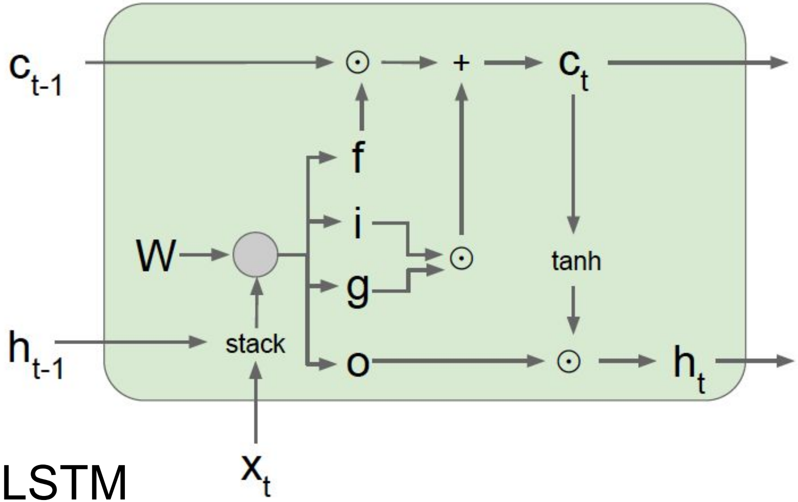


Vanilla RNN

$$
\begin{aligned}
h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\
&= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\
&= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)
\end{aligned}
$$

Long Short-term Memory (LSTM)

- Long short-term memory (LSTM) is a kind of RNN:
  – Replace ‘vanilla’ RNN cell with an LSTM cell



LSTM

Sigmoid

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Long Short-term Memory (LSTM)

- Forget gate (f): whether to erase the cell
- Input gate (i): whether to write to the cell
- Update gate (g): how much to write
- Output gate (o): how much to reveal
- Hidden state (h_t) now depends on the cell state (c_t).

Sigmoid (σ) ranges from 0~1
→ acts like a switch (on-off)

tanh: ranges from -1 to 1

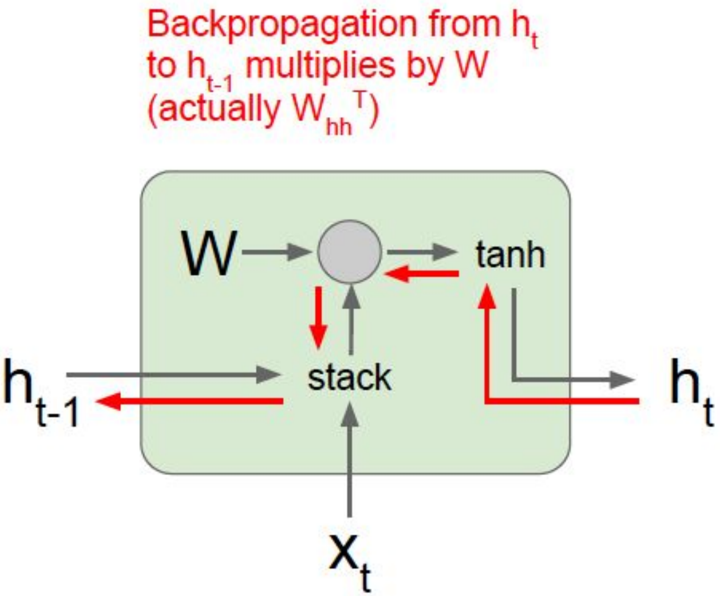
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

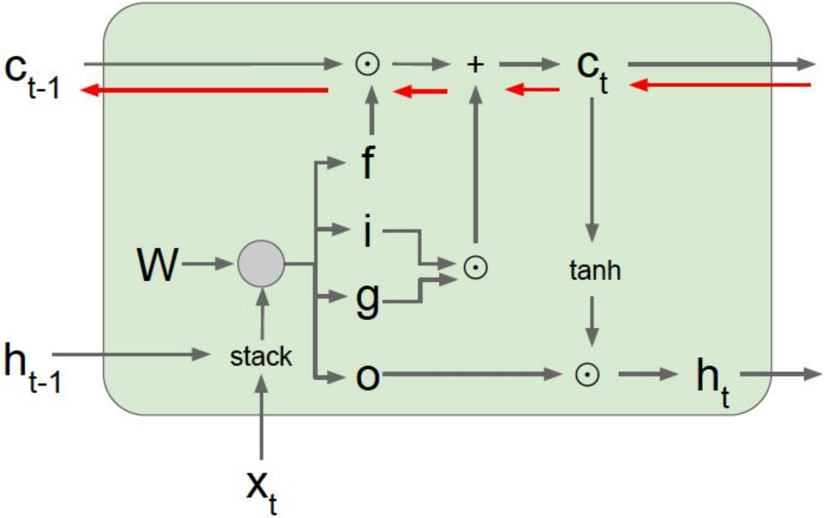
$$h_t = o \odot \tanh(c_t)$$

Long Short-term Memory (LSTM)

Vanilla RNN is prone to vanishing gradients

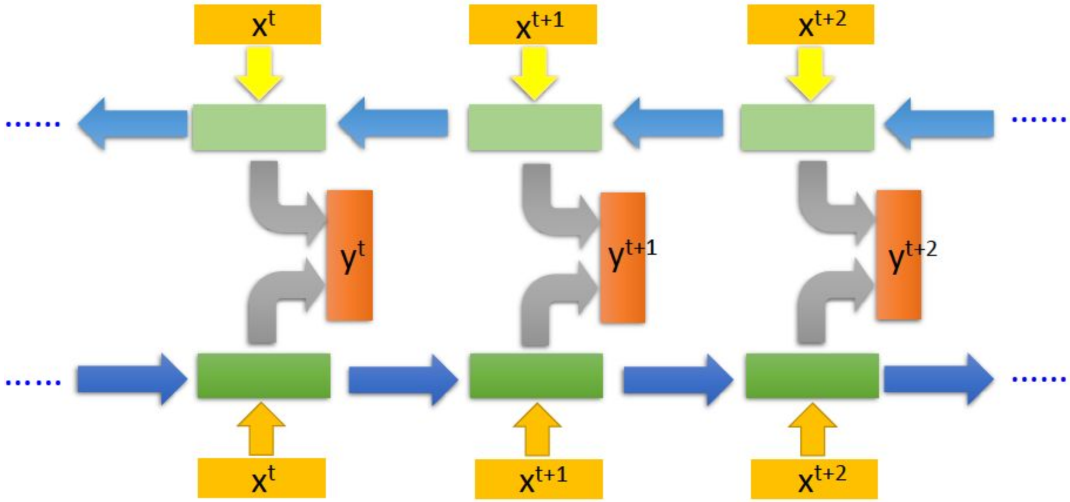


LSTM: Backpropagation from c_t to c_{t-1} only elementwise multiplication by f, no matrix multiply by W



More about RNN

- Gated Recurrent Unit (GRU): ‘simplified’ LSTM (faster)
- Bidirectional RNN: Information can be process both forward and backwards

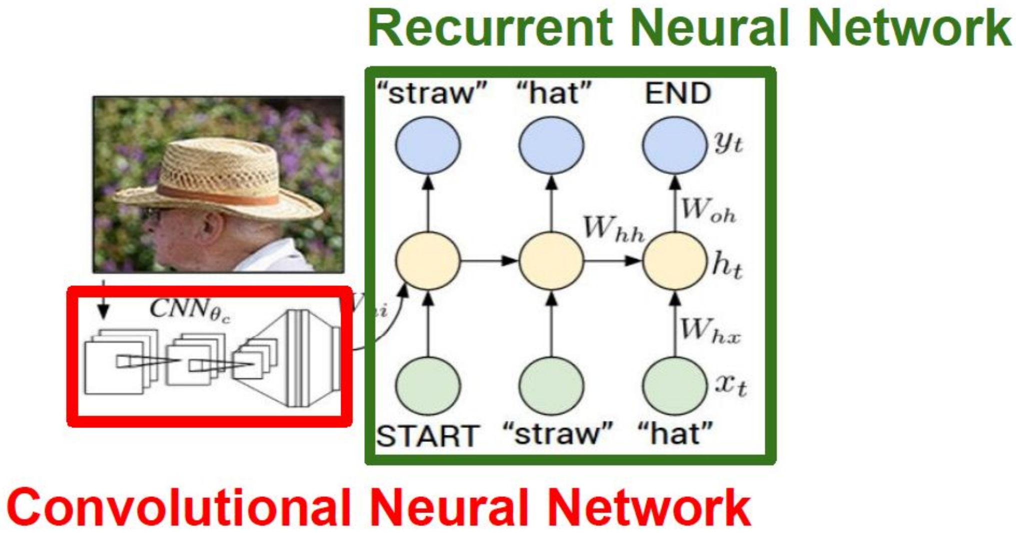


More about RNN

- Bidirectional RNN: Sometimes the relevant information comes later in the sequence
    - John speaks _______, because he grew up in France.

RNN + CNN/DNN

- Don't forget we can put things we've learned together!
    - Image captioning

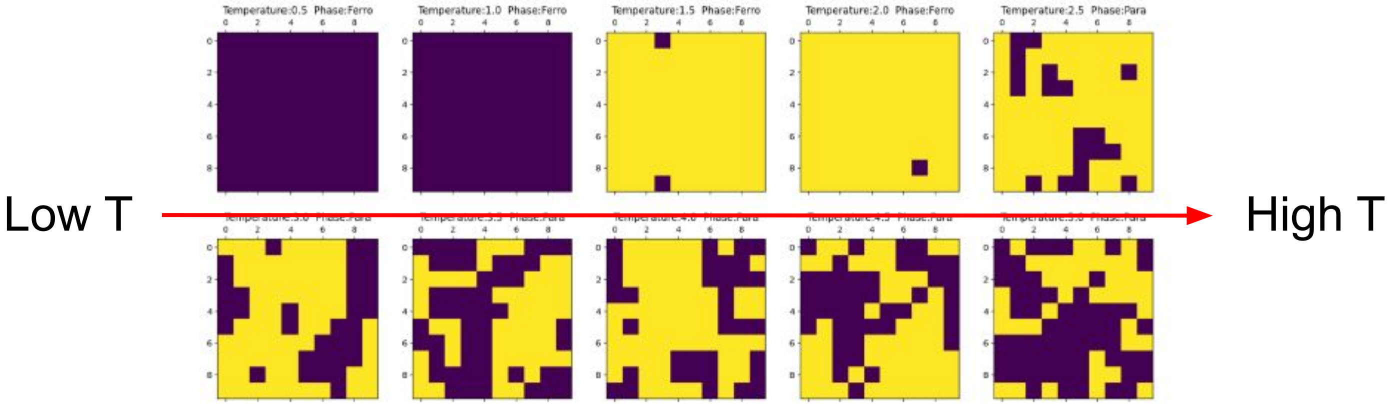


In-class exercise for this week (conti.)

- Let’s go back to the in-class exercise to see the fit results.

Lab for this week

- For Lab this week we will use a dataset of 2D Ising model:
    - Each system has 10x10 spin; spin is up or down.
    - Spin configurations change with respect to temperature



Backup

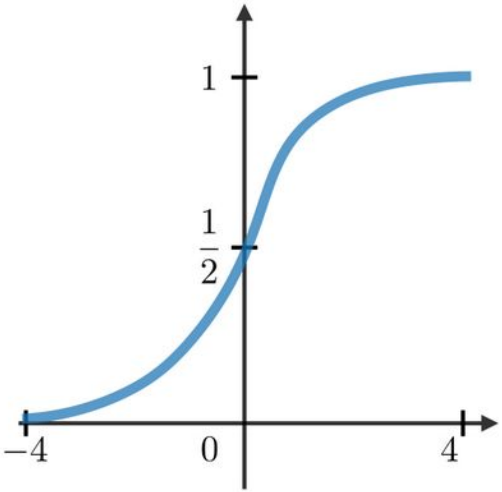
RNN Structure and Applications

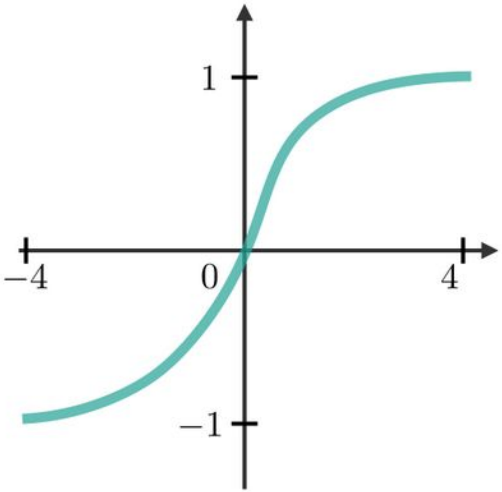
- Why need RNN? E.g. Language processing/recognition (google translate? NLP.), weather/stock prediction, → time-sequenced data

- One-to-one, many-to-one etc.

- Bidirectional RNN

RNN Activation Functions

<table>
<tr><th>Sigmoid</th><th>Tanh</th></tr>
<tr><td>$$g(z) = \frac{1}{1 + e^{-z}}$$</td><td>$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$</td></tr>
<tr><td>


</td><td>


</td></tr>
</table>