DSA5204 LEC5

Sequence Modelling → Today's topic

- classification
- regression
- seq2seq
  - Machine Translation
  - Health Monitor
- sequence generation (no ground-truth) → write poem / compose music (creative)
- sequence prediction (have ground-truth)

---

① Idea: Parameter-sharing across time

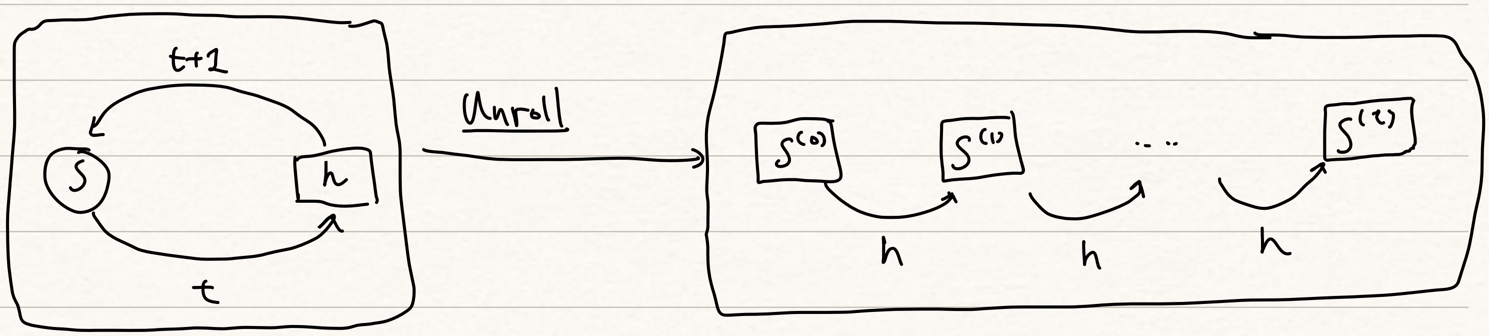
E.g.

- I went Nepal in 2009.
- In 2009 I went to Nepal.

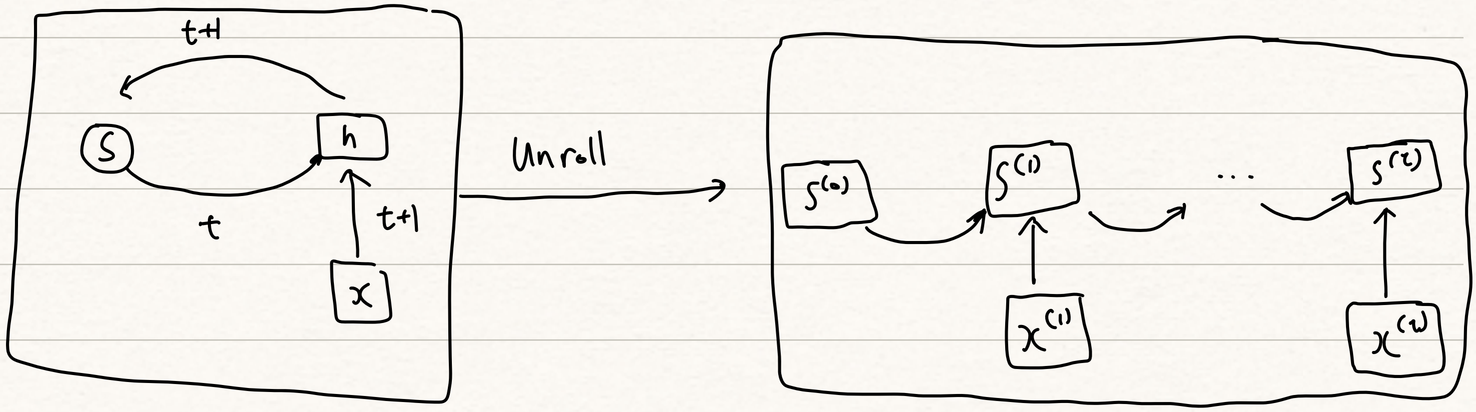
→ same meaning

② Dynamic System

a) $\boxed{s^{(t+1)} = h(s^{(t)} ; \theta)}$



b) $s^{(t+1)} = h(s^{(t)}, x^{(t+1)} ; \theta)$



③ Neural Network formulation

Recap :
$$\begin{cases} s^{(t+1)} = h(s^{(t)}, x^{(t+1)} ; \theta) \\ y^{(t)} = g(s^{(t)} ; \varphi) := o^{(t)} \end{cases}$$

NN

time-t memory

$$\begin{cases} s^{(t+1)} = \sigma_r(W \boxed{s^{(t)}} + U \boxed{x^{(t+1)}} + b) \\ y^{(t)} = \sigma_o(V s^{(t)} + c) \end{cases}$$

current (t+1) observation

Elman Variant

tanh usually (not ReLu)

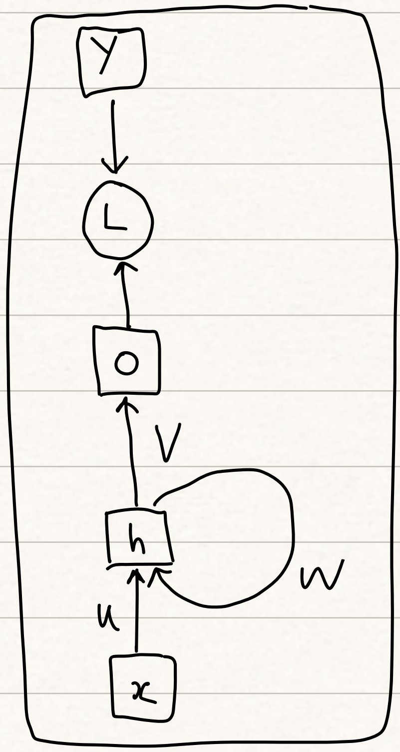
$$\boxed{\begin{cases} \theta = (W, U, b ; \sigma_r) \\ \varphi = (V, c ; \sigma_o) \end{cases}}$$

parametrization

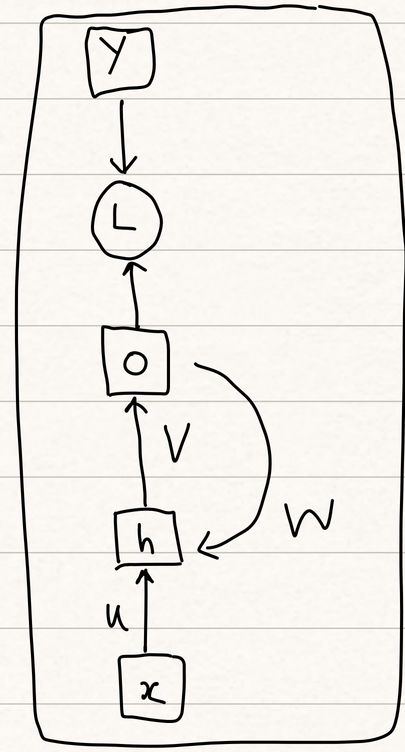
④ Loss Function :

- classification → cross-entropy
- sequence-prediction → sum across time

⑤ Jordan Variant of RNN



ELMAN



JORDAN

---

⑥ Example of [Purpose of Hidden Layer] !

Consider a scaler Time Series $\{ x^{(t)} : t = 1, 2, \dots \}$

output $\{ y^{(t)} : t = 1, 2, \dots \}$

Here, $y^{(t)} = x^{(t)} + x^{(t-1)} + x^{(t-2)}$ for $t \geq 1$

	t	1	2	3	4	5
Then. if	$x^{(t)}$	1	2	3	4	5
then	$y^{(t)}$	1	3	6	9	12

Goal: build model to predict $\boxed{\hat{y}^{(t)} = y^{(t)}}$ $\forall t$

Model: 1. $\hat{y}^{(t)} = FCNN(x^{(t)})$

Issue: No memory

2. $\hat{y}^{(t)} =$ General linear function on $\{x^{(1)}, \ldots, x^{(t)}\}$

$$= \sum_{s=1}^{t} a^{(s)} x^{(s)}$$

fit $a^{(s)}$ to data

Issue: cannot make inference (have memory)

3. (RNN)

3 unit of memory

$$\boxed{\begin{cases} h_1^{(t)} = x^{(t)} \\ h_2^{(t)} = x^{(t-1)} \\ h_3^{(t)} = x^{(t-2)} \end{cases}} \longrightarrow \hat{y}^{(t)} = h_1^{(t)} + h_2^{(t)} + h_3^{(t)}$$

$$\Longrightarrow \begin{cases} h^{(t)} = W h^{(t-1)} + U \cdot x^{(t)} + b \\ y^{(t)} = V h^{(t)} + c \end{cases}$$

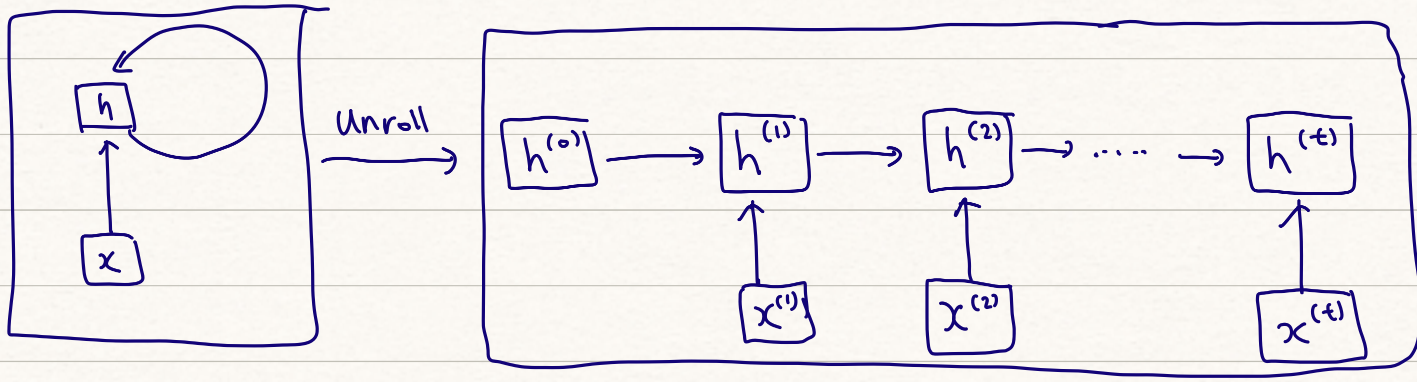
$\rightarrow$ identical activation

where $W = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ $\quad U = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

$V = \mathbb{1}^T$ $\quad b = c = 0$

⑦ How to train RNN ?

$\rightarrow$ Keypoint: Unroll the computational graph



Issue: Gradient
- Vanishing $\rightsquigarrow$ for far-away nodes
  - hard to learn long-term dependency
- Explosion

idea: $a^k \to +\infty$ $a>1$ as $k \to \infty$ (explosion)

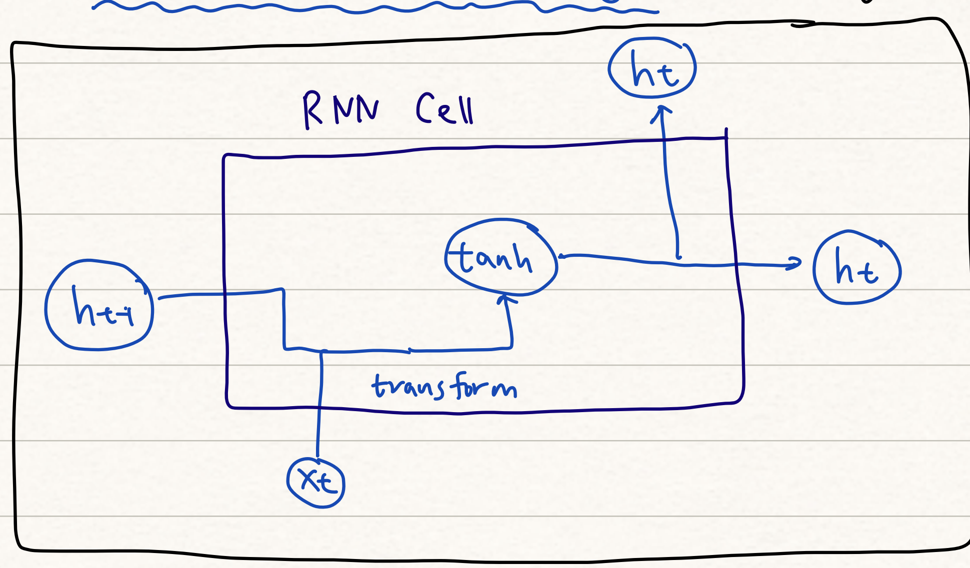
$a^k \to 0$ $a<1$ as $k \to \infty$ (vanish)

Modification → GRU (LSTM)

idea: use gates to control the accumulation of knowledge

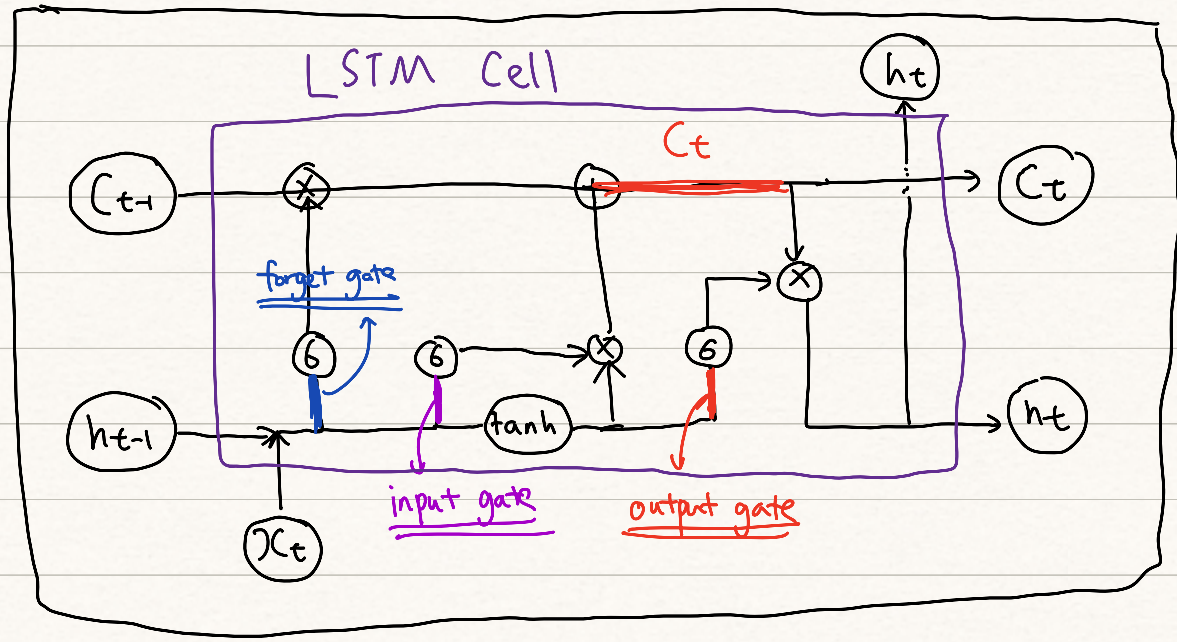
Diagram

RNN Cell



LSTM Cell

Diagram



☆

Self-understanding:

→ Gate Unit is trying to learn (adaptive) from $\begin{cases} h_{t-1} \\ x_t \end{cases}$

in order to determine
1. how much memory should be left
2. how much memory should be updated
3. how much hidden unit should be transformed

⑧ Deep RNN

→ Shallow RNN

$$\begin{cases} h^{(t)} = \sigma_r\left(W h^{(t-1)} + U x^{(t)} + b\right) \\ \hat{y}^{(t)} = \sigma_o\left(V h^{(t)} + c\right) \end{cases}$$

→ Deep RNN

$$\begin{cases} h^{(t)} = \sigma_r\left(W_1 h^{(t-1)} + U_1 x^{(t)} + b_1\right) \\ z^{(t)} = \sigma_r\left(W_2 z^{(t-1)} + U_2 h^{(t)} + b_2\right) \\ \hat{y}^{(t)} = \sigma_o\left(V z^{(t)} + c\right) \end{cases}$$

9 Other Variants

→ 1. **Bi-directional RNN** (Translation)

→ 2. Seq 2 Seq (Encoder - Decoder Architecture)

⇓

MODEL { input : sequence
output : sequence