Recurrent Neural Networks (RNNs)

1. Models predict the future from the past.

Input: sequence: $x(1), x(2), x(3), \cdots, x(t), \cdots, x(\tau)$

t: timepoint

$x(t)$: tth input value

$x(1), x(2), \cdots x(t-1)$ → [Model] → $x(t)$

↳ model is a dynamic model.

$f(x, t)$

2. Applications

① voice recognition / speech recognition

② machine translation.

③ weather prediction.

④ stock market prediction

3. models

1) Markov models (Markov chain, Hidden Markov model, (HMM)

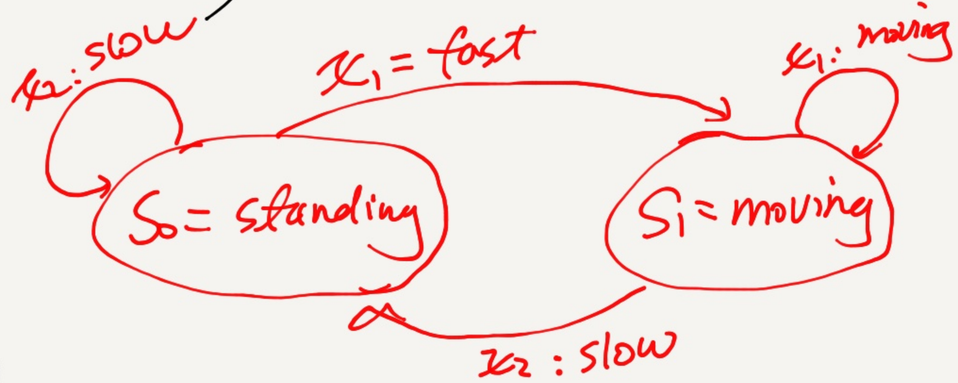
Markov random field (MRF))

Sequence: $x(1), x(2), \cdots, x(T)$

$$p(x(t) \mid x(1), x(2), \cdots, x(t-1)) \simeq p(x(t) \mid x(t-1)) \quad x(t-1) \longrightarrow x(t)$$

$$= \frac{\text{count}(x(t-1), x(t))}{\text{count}(x(t-1))}$$ (frequency of word in a sentence)

2) State machines



Set of states: $S = \{ S_0, S_1, \cdots, S_m \}$

Set of input: $X = \{ x_1, x_2, \cdots, x_n \}$

State transition: $f: S \times X \rightarrow S$. $f(S_{t-1}, x_t) = S_t$

output: $g: S \rightarrow y$: $y_t = g(S_t)$

RNNs are state machines:

4. How can we design a NN to predict the future?

input: $x(1), x(2), x(3), \cdots x(T)$

output: $y(1), y(2), y(3), \cdots, y(T)$

$\|$ $\|$ $\|$ $\|$

$x(2)$ $x(3)$ $x(4)$ $x(T+1)$

① use one timepoint to predict next timepoint

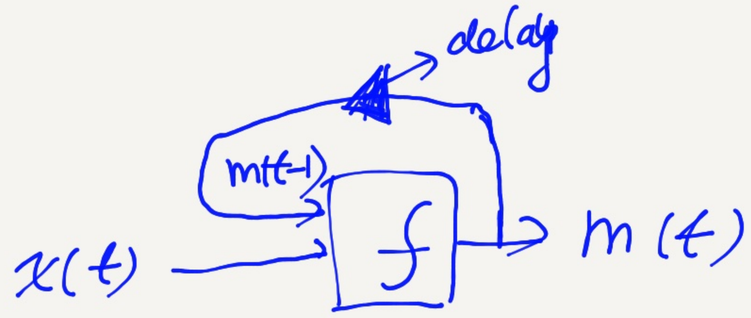
$$x(t-1) \rightarrow \boxed{f} \rightarrow y(t-1) = x(t)$$

✍: This only model local context; sensitive to noise

✍: This is static model. It has no long-range / long-term dependence.

② $x(1), x(2), \cdots, x(t-1) \rightarrow \boxed{f(x)} \rightarrow y(t-1)$

③ Design RNNs based on the idea of moving average.



m(t-1): encodes the past sequence.

input sequence. moving average

x_1 $\quad m_1 = x_1$

x_2 $\quad m_2 = \frac{x_1+x_2}{2} = \frac{m_1}{2} + \frac{x_2}{2}$

x_3 $\quad m_3 = \frac{2m_2+x_3}{3} = \frac{2}{3}m_2 + \frac{x_3}{3}$

$\vdots$

x_t

current state / state

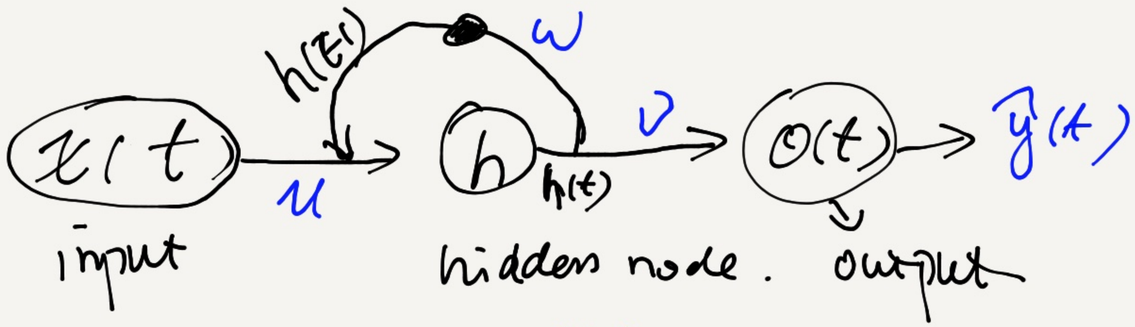
$$m_t = \frac{(t-1)\cdot m_{t-1}}{t} + \frac{x_t}{t}$$

5. RNN concepts.

1) RNN is a family of NNs for processing sequential data.

Both the input and output are sequences.

2) standard RNN.



$$h(t) = \underbrace{g_H}_{\text{activation}}(net_H), \quad net_H = \overbrace{u^T x(t) + w^T \cdot \underbrace{h(t-1)}_{0}}^{\text{input}} + \overbrace{b_h}^{\text{bias}}$$

$$\hat{y}(t) = \underbrace{g_O}_{\text{activation}}(v^T h(t) + b_O)$$

The feedback connection gives the network the ability to model long-term dependence.

3). ① $x(t) \to h \to o \to \hat{y}(t)$

② $x(t) \to h \to o \to \hat{y}(t)$ (use previous prediction)