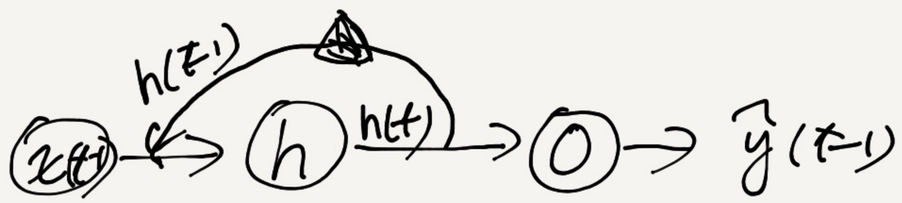
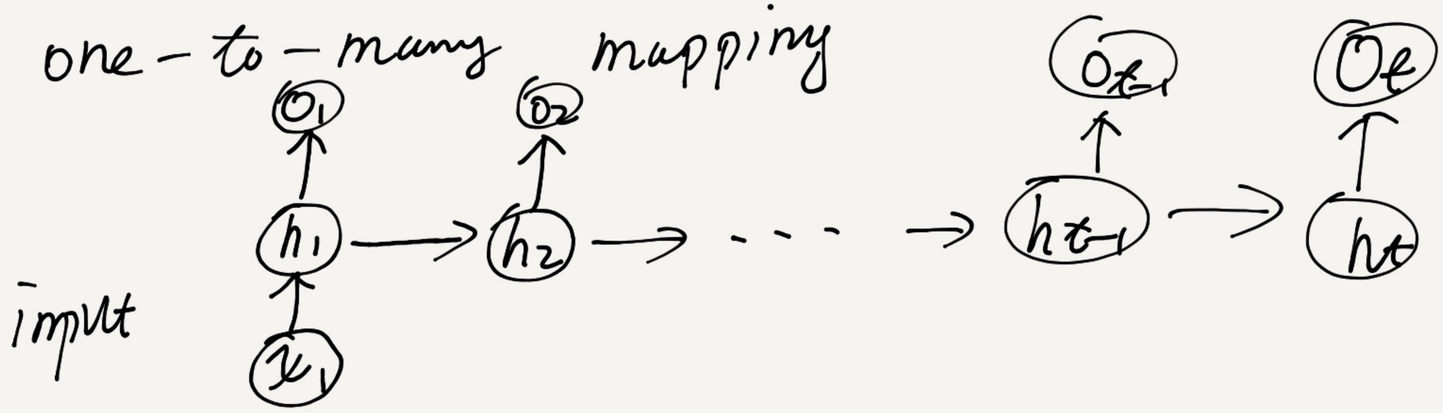
RNN architectures.

1. Basic architectures.



① one-to-many mapping

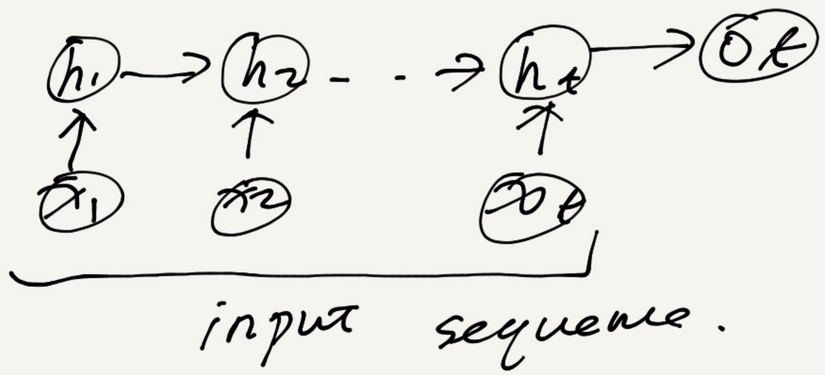


Application: image captioning.

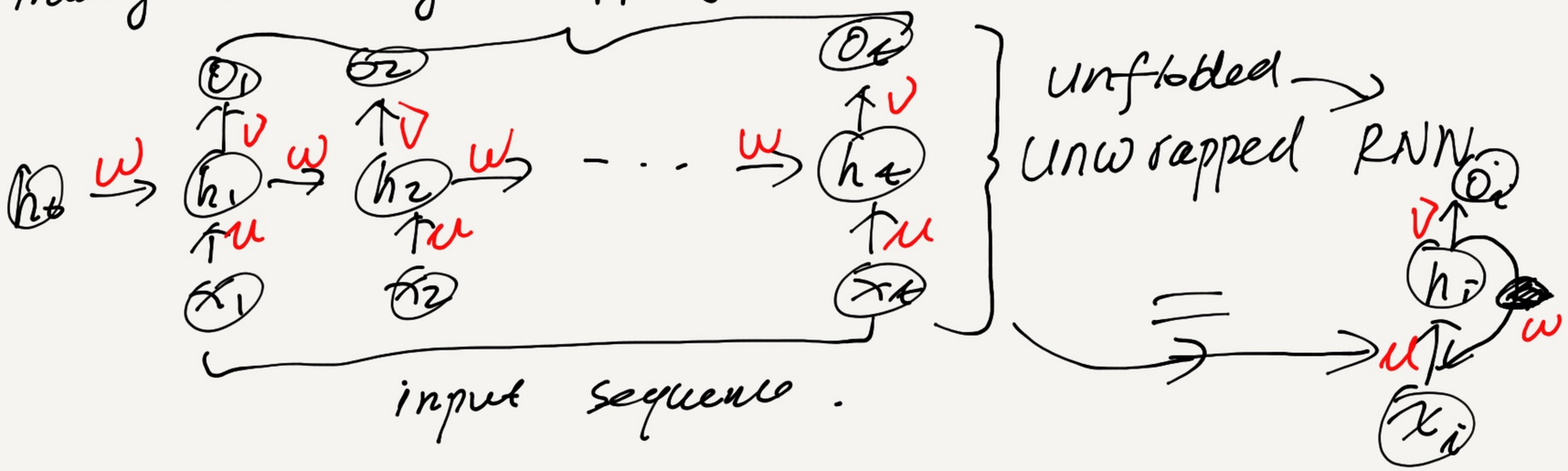
x_1: an image

$O_1 \cdots O_t$: sequence of words.

② many-to-one mapping



③ many-to-many mapping



input : $x_1 \longrightarrow h_1$ (hidden state)

$x_2, h_1 \longrightarrow h_2$

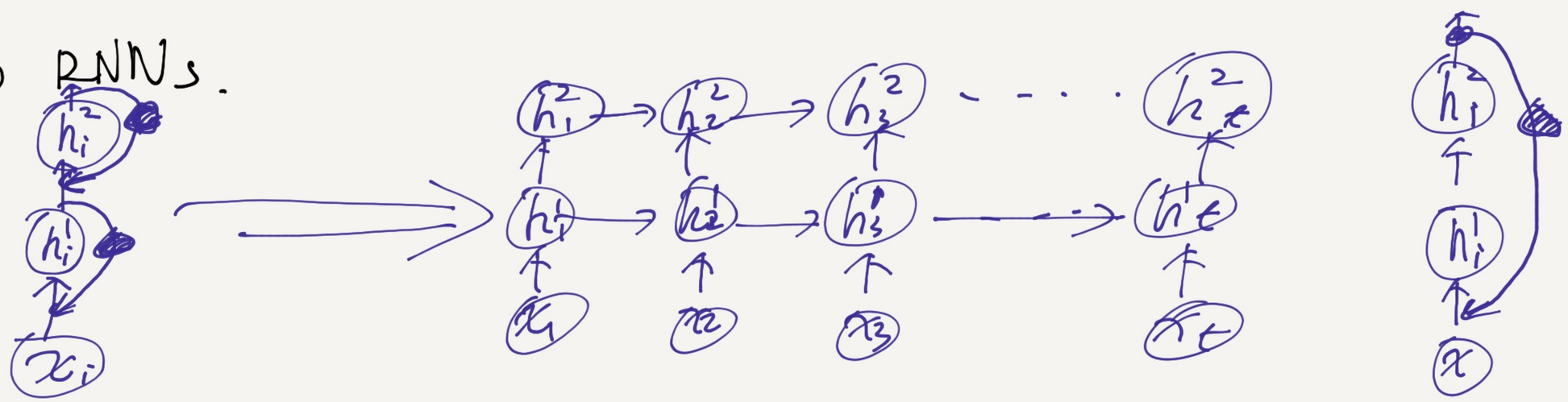
$x_3, h_2 \longrightarrow h_3$

△ RNNs share weight across time.

△ carry over past information to (h_t) future by using the feedback connection.

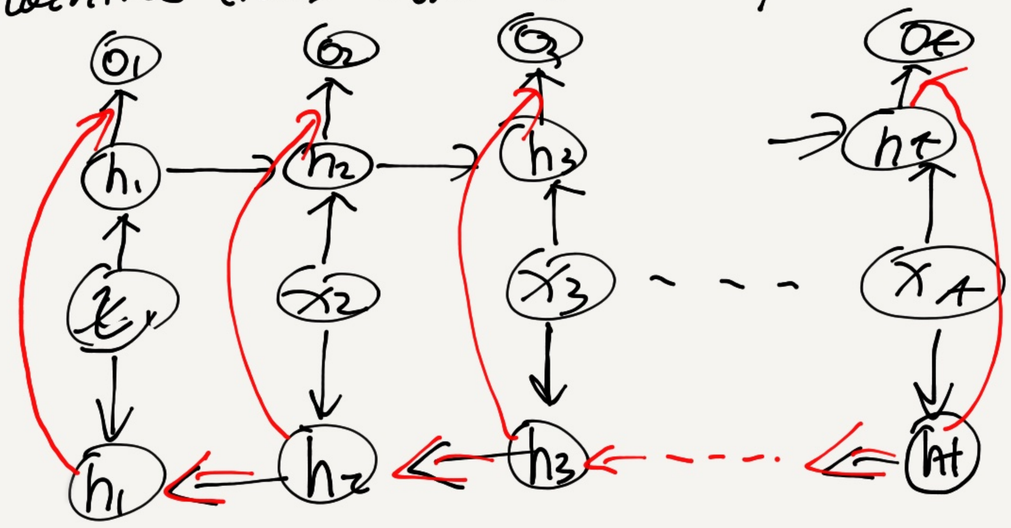
Machine translation:

2 Deep RNNs.



3 Bidirectional RNNs use both past and future values in a sequence.

machine translation and speech recognition.



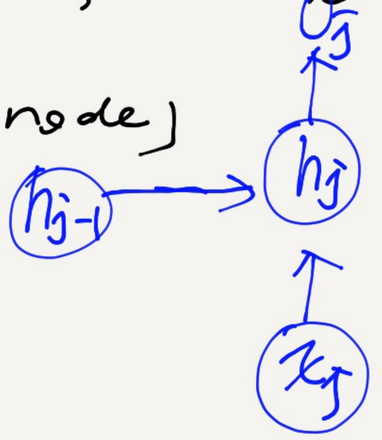
BERT (LLM).

4. Optimize RNNs: (BP through time)

$$L_{seq} = \sum_{j=1}^{e} -y_j \cdot \log o_j \quad \text{(cross-entropy between two sequences)}$$

$$o_j = g_o(V h_j + b_o) \quad \text{(output node)} \qquad \begin{cases} y_1, y_2 \cdots y_e \text{ (target)} \\ o_1, o_2, \cdots o_e \text{ (pred.)} \end{cases}$$

$$h_j = g_h(U x_j + W h_{j-1} + b_h) \quad \text{(hidden node)}$$

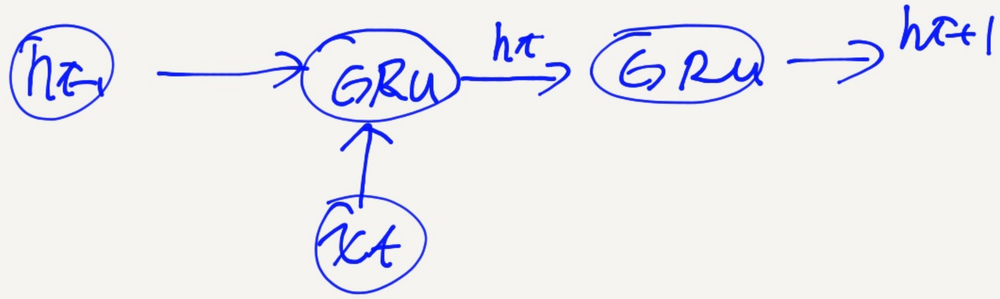


5. Two popular RNNs

1) LSTM (long short-term memory) 1997.

2) Gated Recurrent Unit (GRU)

① overall architecture.

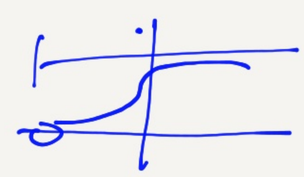


② 3- component in GRU.

Update gate: z_t : controls how much information from the past

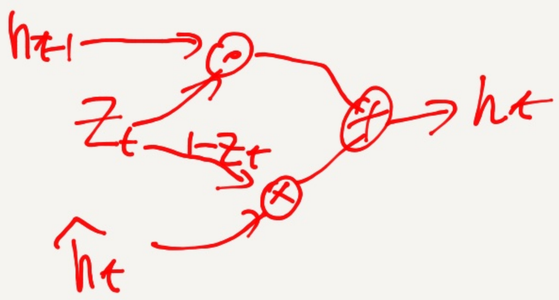
Reset gate: r_t controls how much information to ignore from the past.

z_t: $z_t = g_z(W_z \cdot h_{t-1} + U_z x_t)$ g_z: sigmoid $\to [0,1]$



update gate.

$h_t = z_t \odot h_{t-1} + (1 - z_t)\, \hat{h}_t$



$\odot$: element-wise multiplication.

Candidate: $\hat{h}_t = g_h(W_h(r_t \odot h_{t-1}) + U_h \cdot x_t)$

r_t: $r_t = g_r(W_r \cdot h_{t-1} + U_r \cdot x_t)$: g_r: sigmoid $\to \in [0,1]$

Reset gate