Neural Networks and Deep Learning

Recurrent Neural Network

Shumin Wu

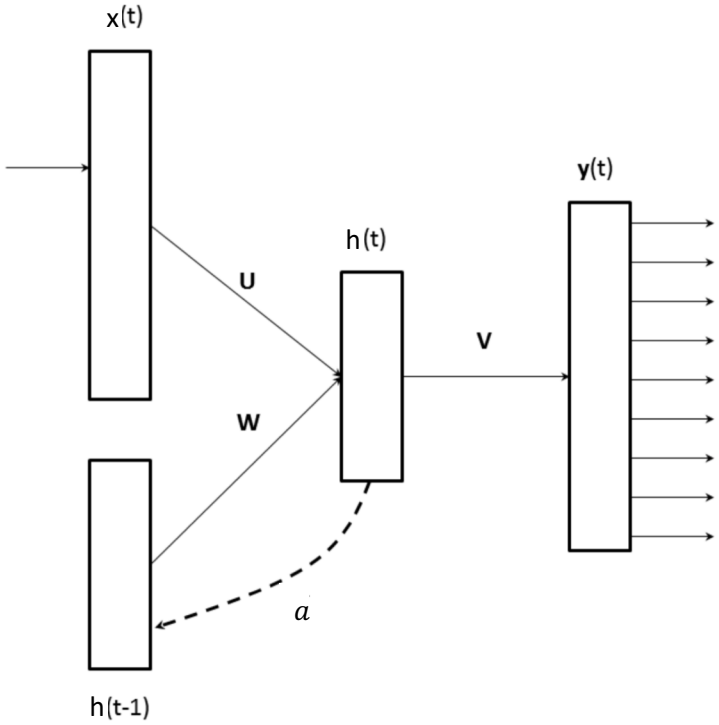
Department of Computer Science

shumin.wu@colorado.edu

February 17, 2020

Previous Lecture: DL in NLP

- Word representation
  - Latent semantic analysis
  - Word2vec
- Language model
  - RNN architecture



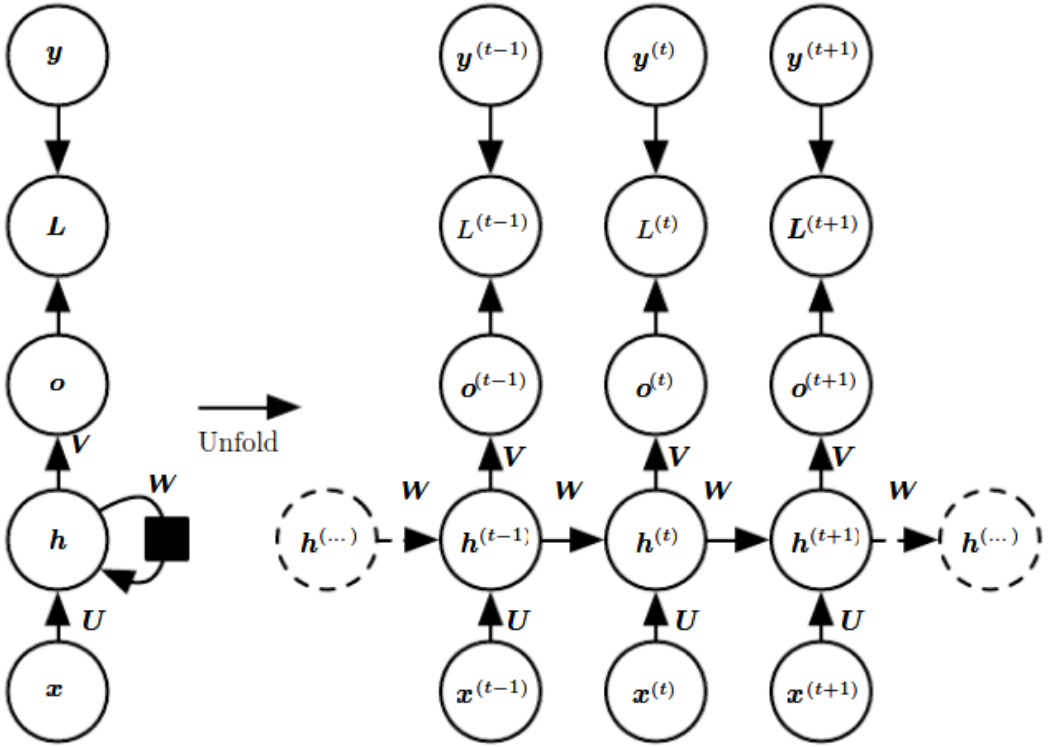
Mikolov et al. 2010

$$h(t) = \sigma(Ux^t + Wh^{t-1})$$
$$y^t = \mathrm{softmax}(Vh(t))$$

Recurrent Neural Network

RNN with multiple outputs

- Left: circuit diagram of recurrent network.
- Right: the same graph unfolded over time, so each node is associated with a particular time step *t*



source: http://deeplearningbook.org

Recurrent Neural Network

$$\alpha^t = Ux^t + Wh^{t-1} + b_\alpha$$
$$h^t = \tanh(\alpha^t)$$
$$o^t = Vh^t + b_o$$
$$\hat{y}^t = \text{softmax}(o^t)$$

U	Input-to-hidden weights
V	Hidden-to-output weights
W	Hidden-to-hidden weights (recurrent)
b_α	Input bias
b_o	Output bias
h^0	Initial state

RNN compared with CNN

CNN: $$S(t) = (I * K)(t) = \sum_m I(t-m)K(m)$$

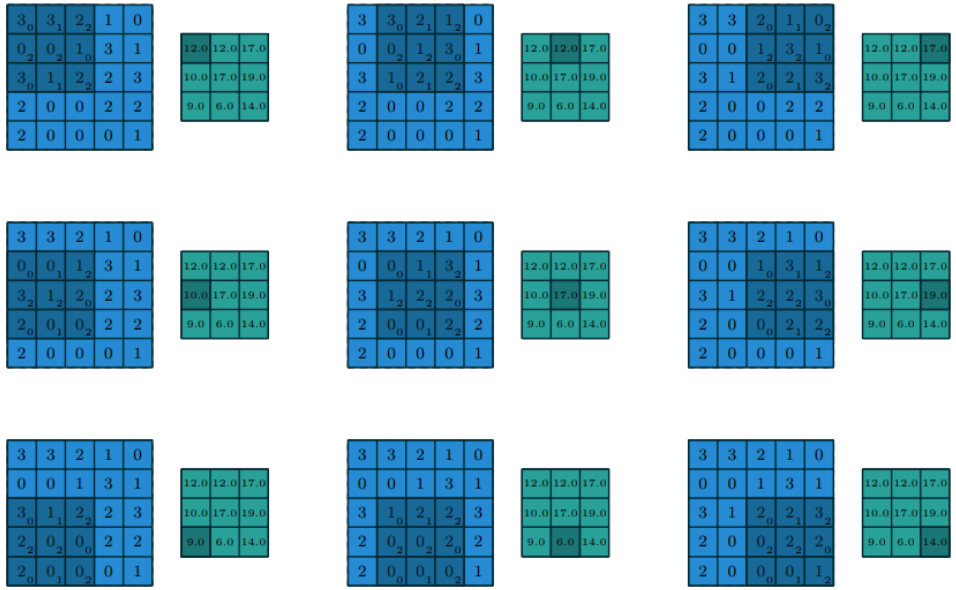
Parameter sharing across a window of m steps
(Larger effective window size thru multiple CNN layers)

RNN:
$$\begin{aligned} h^t &= \tanh(\alpha^t) \\ &= \tanh(Wh^{t-1} + Ux^t + b_\alpha) \\ &= \tanh(W\tanh(Wh^{t-2} + Ux^{t-1} + b_\alpha) + Ux^t + b_\alpha) \\ &= \dots \end{aligned}$$

Parameter sharing across all previous time steps

RNN compared with CNN

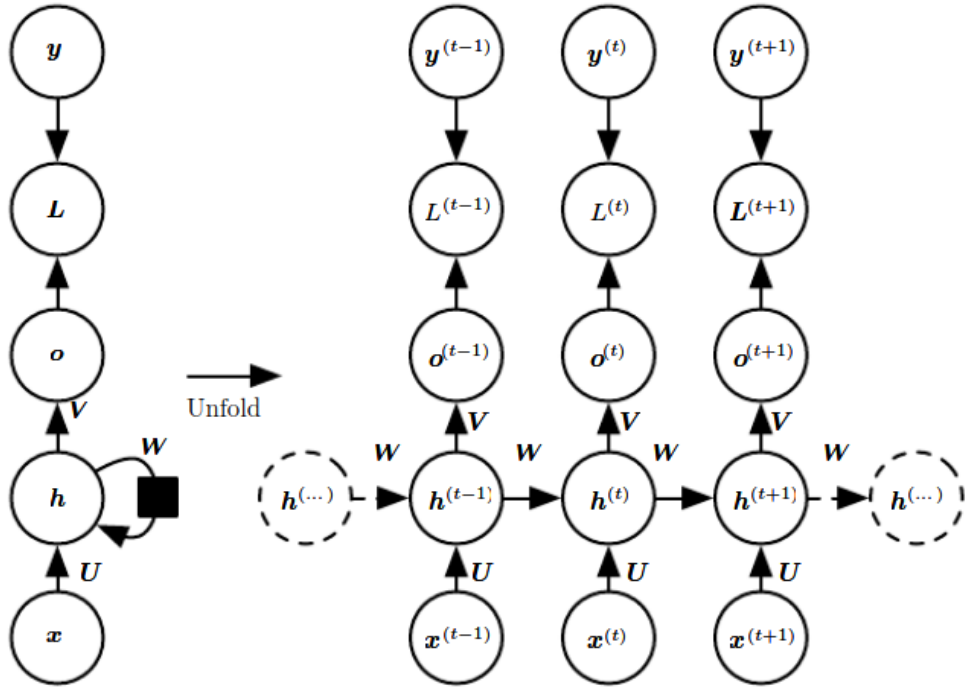
CNN:



source: A guide to convolution arithmetic for deep learning

Easily parallelizable

RNN:



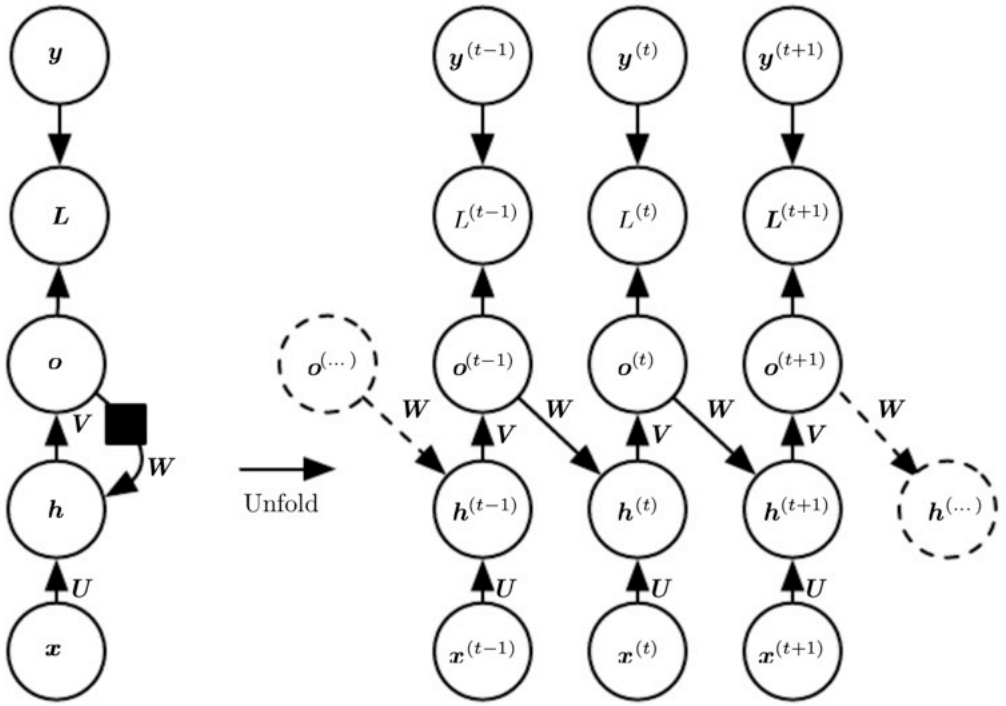
Not parallelizable

RNN Output-to-Hidden Recurrence

Input to the hidden state $h^{(t)}$ is the output $o^{(t-1)}$:

$$\alpha^t = Ux^t + Wo^{t-1} + b_\alpha$$

$$h^t = \tanh(\alpha^t)$$



source: http://deeplearningbook.org

Why is this less powerful than the previous RNN?

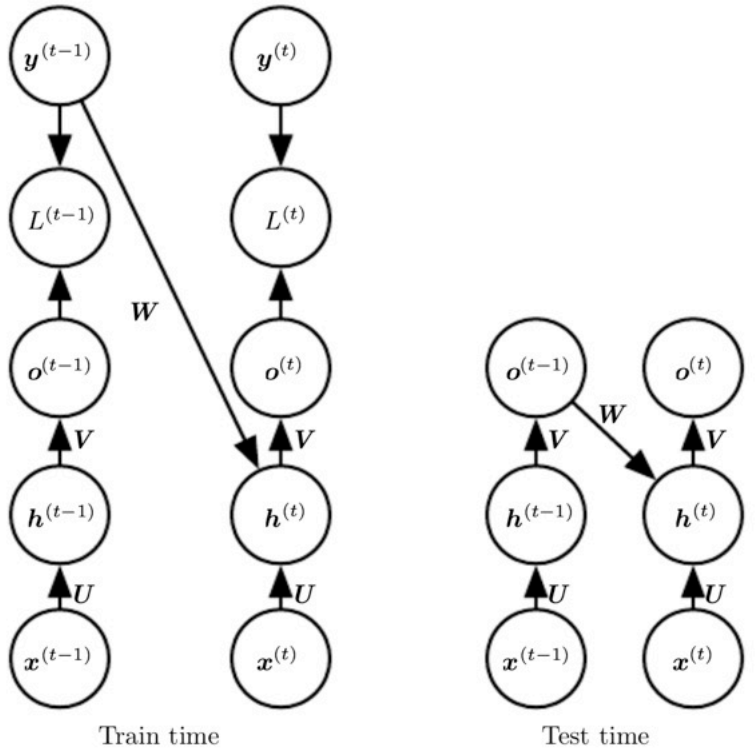
RNN Teacher Forcing

Allows one to directly input the target $y^{(t-1)}$ to the hidden state h^t at training time.

At inference time, the output of the network can be input to the hidden state.

Makes (training) parallelism possible.

What are the risks?



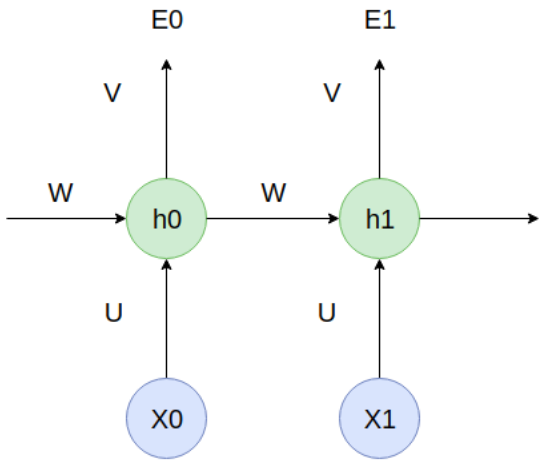
source: http://deeplearningbook.org

RNN Backpropagation Through Time (BPTT)

Backpropagating through an unrolled RNN requires:

$$\frac{\partial E^1}{\partial V} = \frac{\partial E^1}{\partial \hat{y}^1}\frac{\partial \hat{y}^1}{\partial V}$$

$$\frac{\partial E^1}{\partial W} = \frac{\partial E^1}{\partial \hat{y}^1}\frac{\partial \hat{y}^1}{\partial h^1}\frac{\partial h^1}{\partial W}$$



Note the dependency of h^1 on h^0:

$$\frac{\partial E^1}{\partial W} = \frac{\partial E^1}{\partial \hat{y}^1}\frac{\partial \hat{y}^1}{\partial h^1}\frac{\partial h^1}{\partial h^0}\frac{\partial h^0}{\partial W}$$

RNN Truncated BPTT

We can limit the computational cost of BPTT procedure by starting with $k = t - a$, where a is a constant and $a < t$.

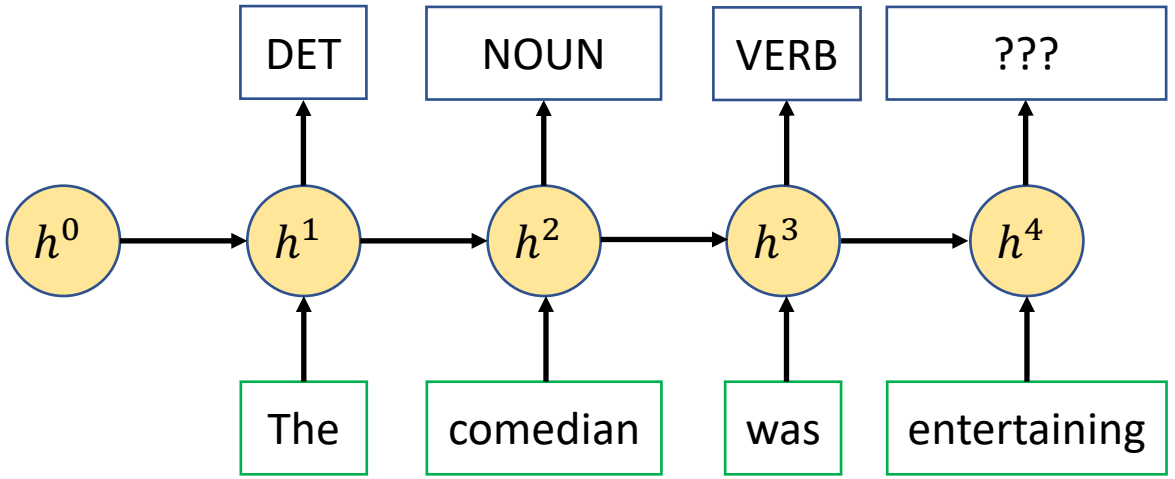
$$\frac{\partial E}{\partial W} = \sum_{t} \frac{\partial E^t}{\partial W}$$

$$\frac{\partial E^t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E^t}{\partial \hat{y}^t} \frac{\partial \hat{y}^t}{\partial h^t} \frac{\partial h^t}{\partial h^k} \frac{\partial h^k}{\partial W}$$

RNN Part of Speech Tagging

DET	NOUN	VERB	VERB	DET	NOUN
The	comedian	was	entertaining	the	audience
The	comedian	was	entertaining	and	informative
DET	NOUN	VERB	ADJ	CONJ	ADJ

RNN:

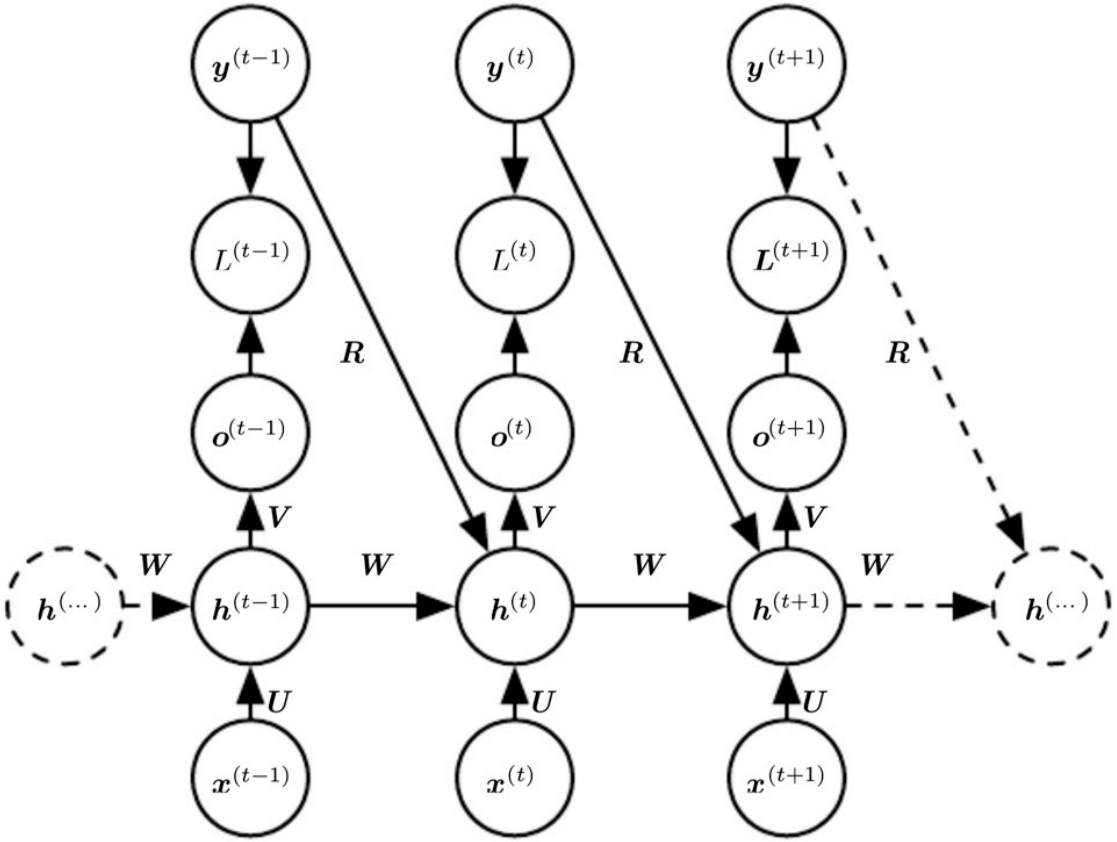


With the same input sequence, this RNN architecture cannot disambiguate the outputs.

RNN Conditioned on Previous Outputs

$h^{(t)}$ depends: on $y^{(t-1)}$:

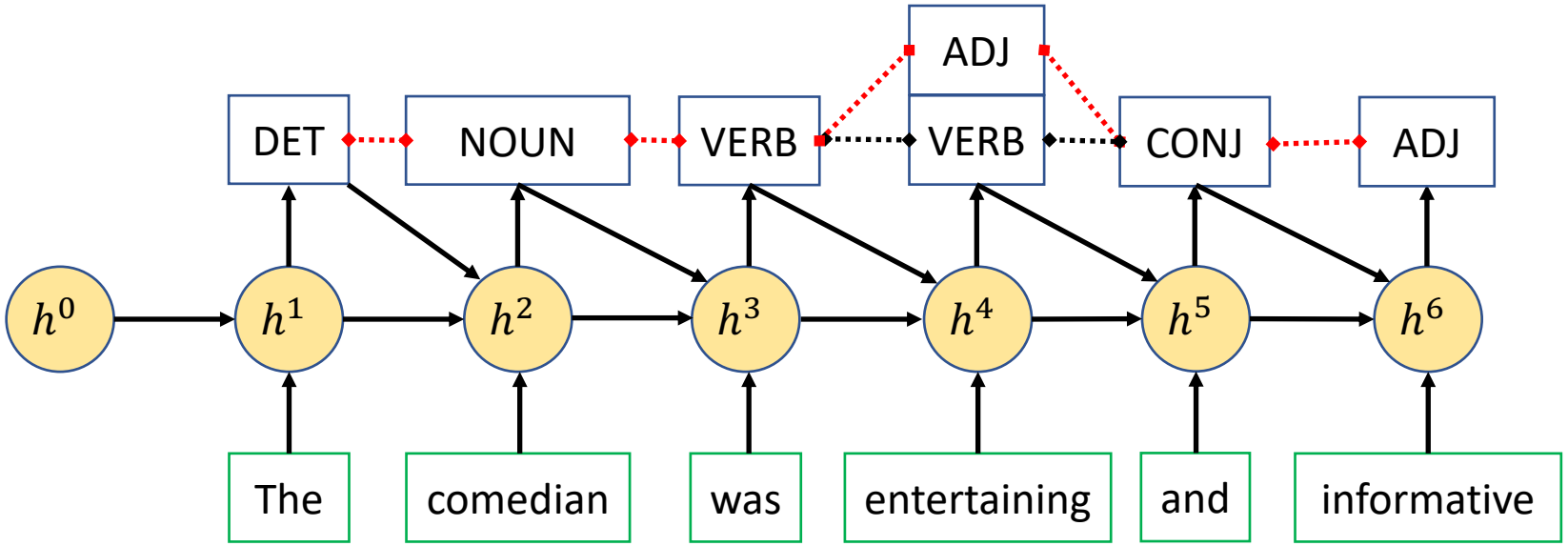
$$\alpha^t = Ux^t + Wh^{t-1} + \boxed{Ry^{t-1}} + b_\alpha$$
$$h^t = \tanh(\alpha^t)$$



source: http://deeplearningbook.org

But how does this help with the POS task?

RNN Beam Search



Perform inference on alternative output label inputs and keep track of the most probable m (beam width) sequences at any time.

Bidirectional RNNs

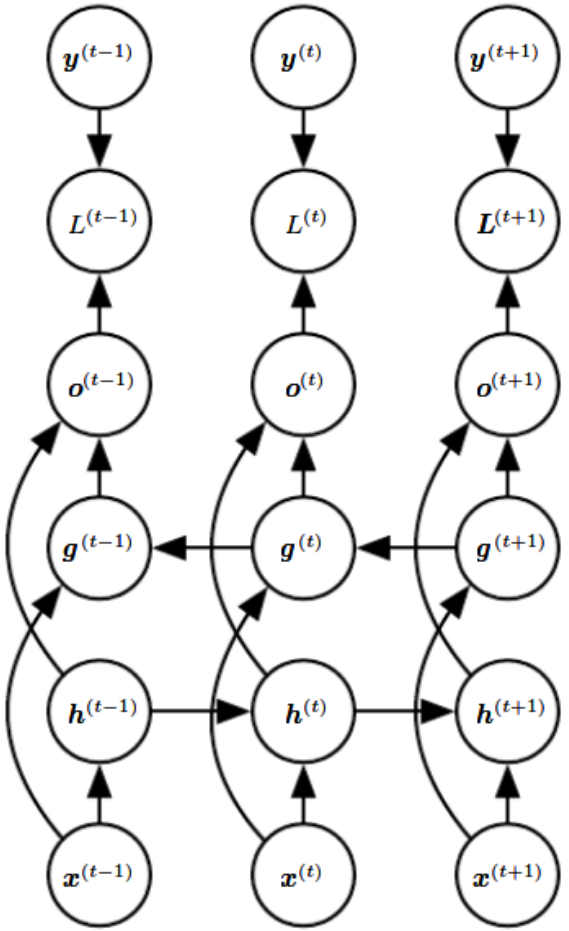
A bidirectional RNN (Schuster and Paliwal, 1997) has hidden state for each direction of the input:

h carries information forward in time

g carries it backward

Higher layers of the network have access to representations of each time step conditioned on both its left and right context.

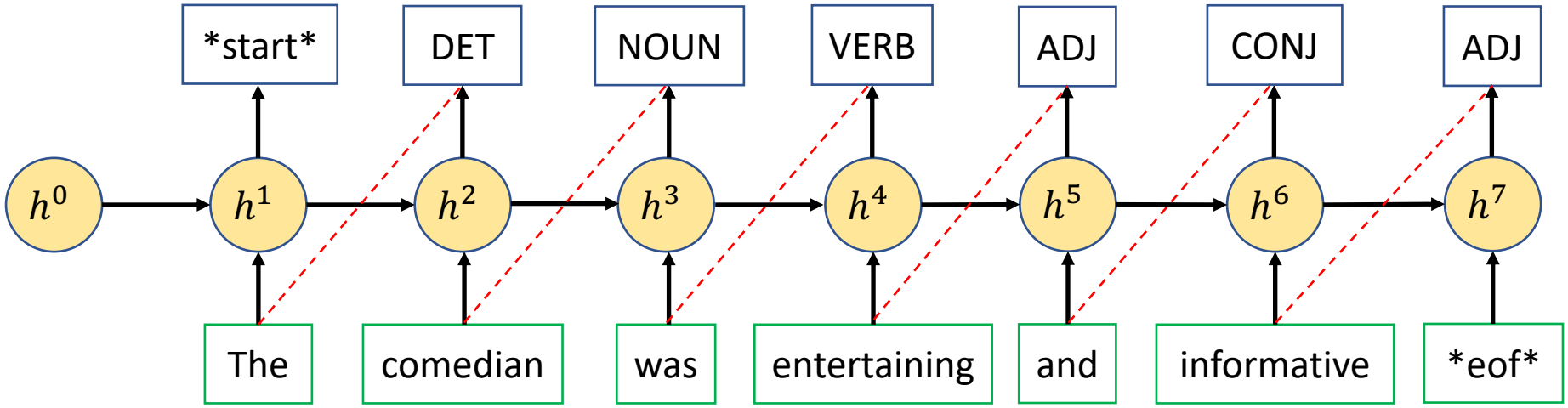
Works when the entire input sequence is available.



source: http://deeplearningbook.org

RNN Encoding Sequence

POS example can be disambiguated if we “look ahead” one input token (or delay output by one token):



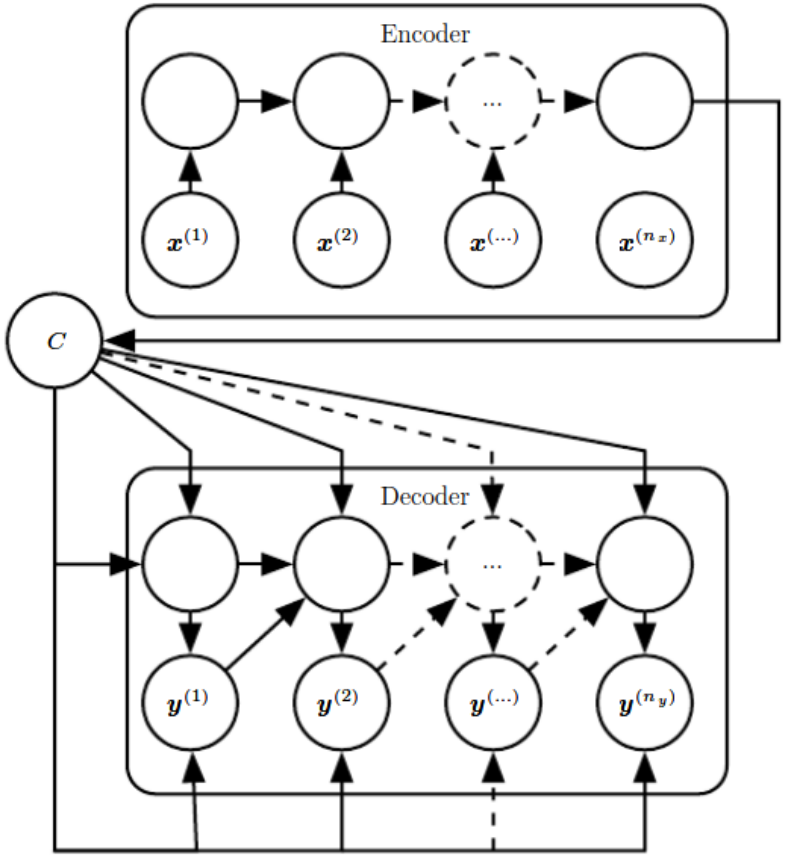
What if we read the entire input sequence before output?

RNN Encoder-decoder

An encoder-decoder RNN (Cho et al, 2014 and Sutskever et al, 2014) decouples input and output lengths by having an encoder map the input to a context C and a decoder map C to the target output.

Well suited for tasks where input and output lengths differ, like machine translation.

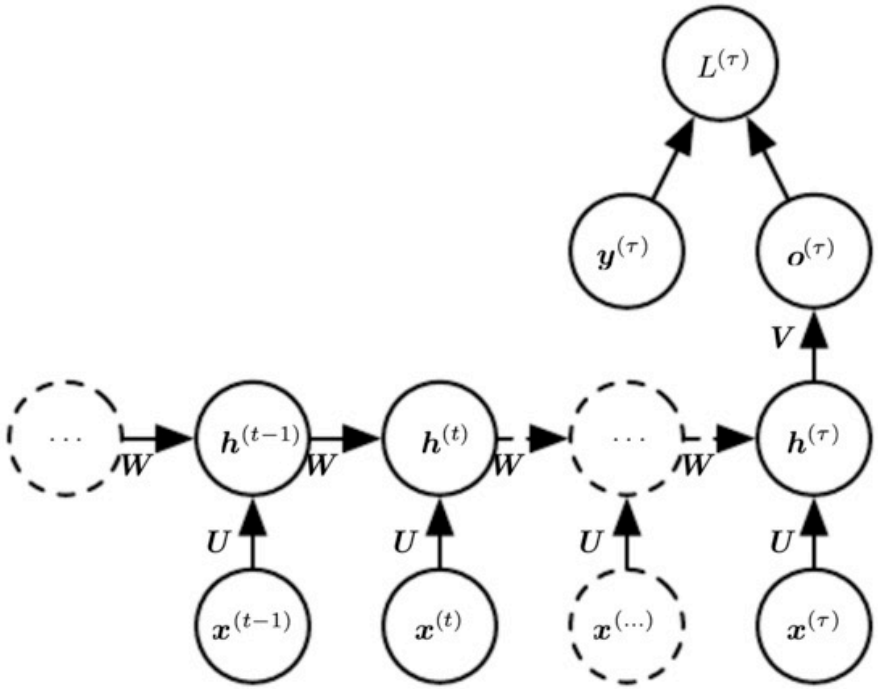
(How big should we pick C, especially when input sequence length can vary a lot?)



Example RNN encoder-decoder architecture
source: http://deeplearningbook.org

RNN Many Inputs to One Output

Application: sentence classification



source: http://deeplearningbook.org

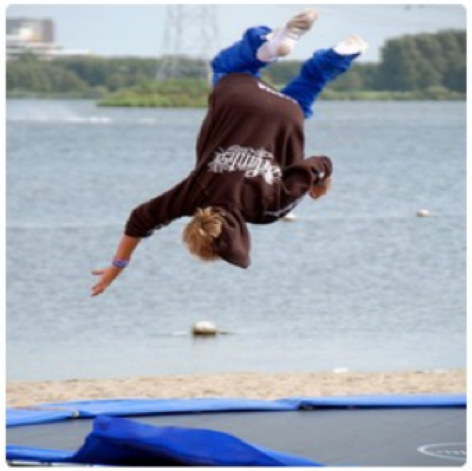
RNN One Input to Many Outputs

Image Captioning:

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."



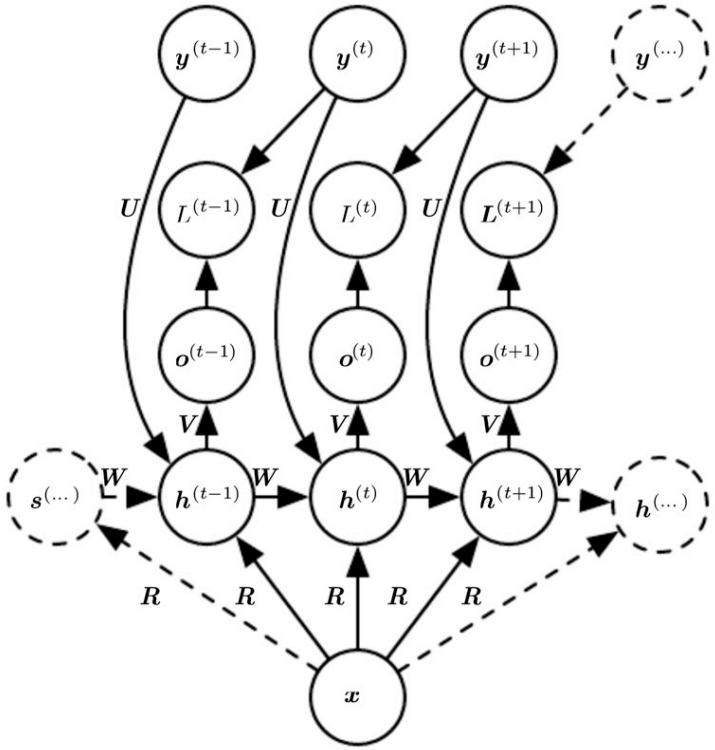
"boy is doing backflip on wakeboard."

source: [Andrey Karpathy, Fei-Fei Li]

Output typically conditioned on previous outputs. Why?

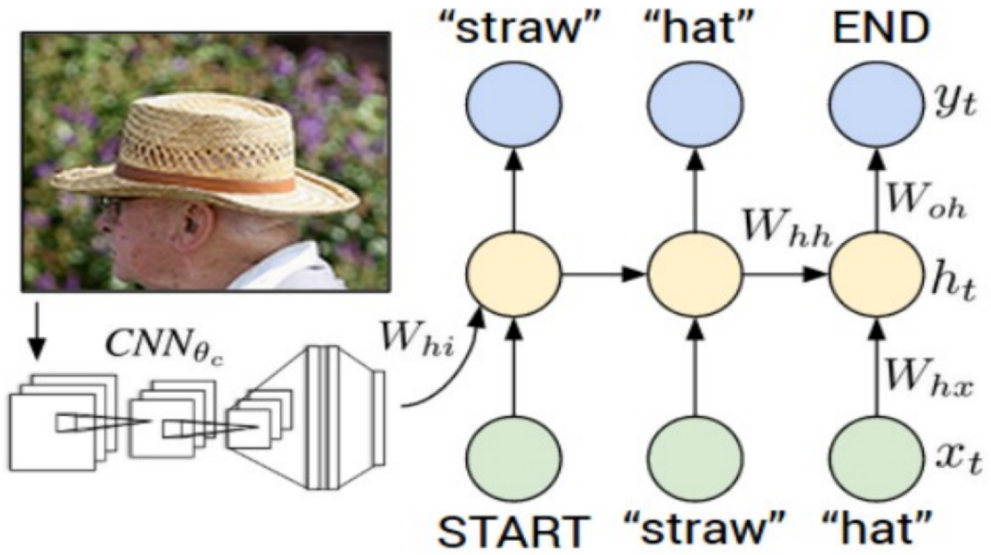
RNN One Input to Many Outputs

Same input at each time step:



source: http://deeplearningbook.org

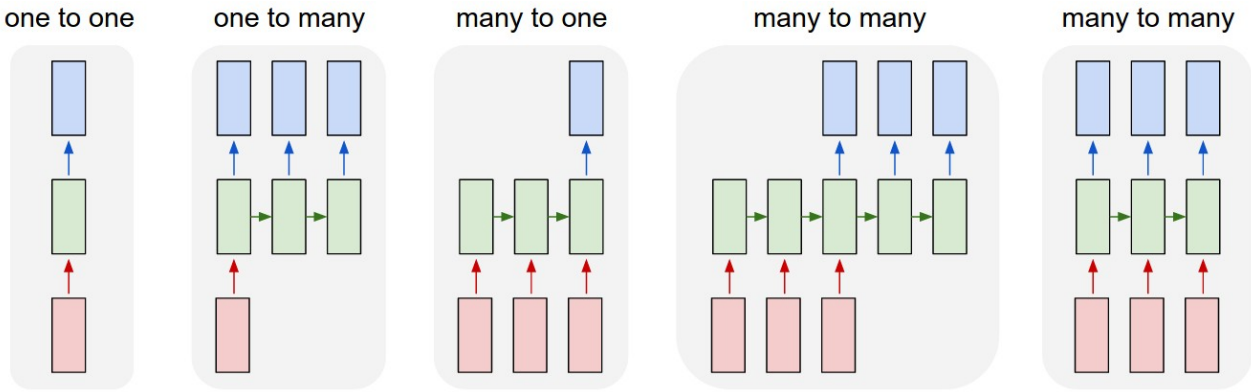
Input fed into initial state:



source: [Andrey Karpathy, Fei-Fei Li]

RNN Design Patterns

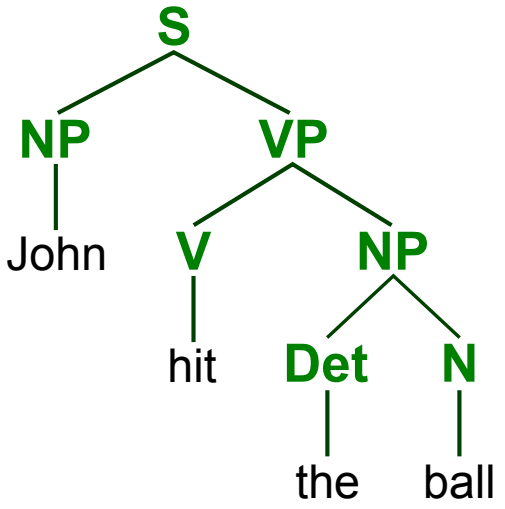
- One to one → no recurrence, a feed-forward network
- One to many → image captioning
- Many to one → sentence classification
- Many to many (different size) → machine translation
- Many to many (same size) → word/video frame tagging/labeling



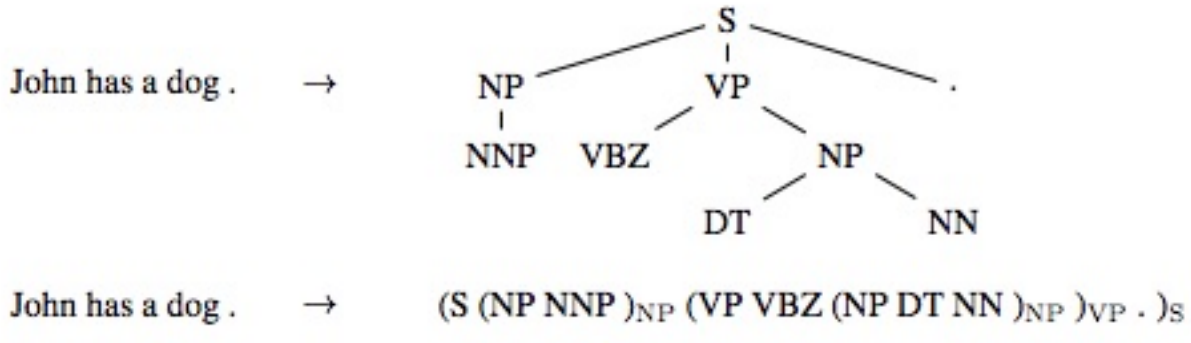
source: https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Recursive Neural Network (RecNN)

Some interesting sequential data can be represented hierarchically, and the hierarchical representations can be represented sequentially:



Example of a constituent parse tree



Linearization of a parse tree

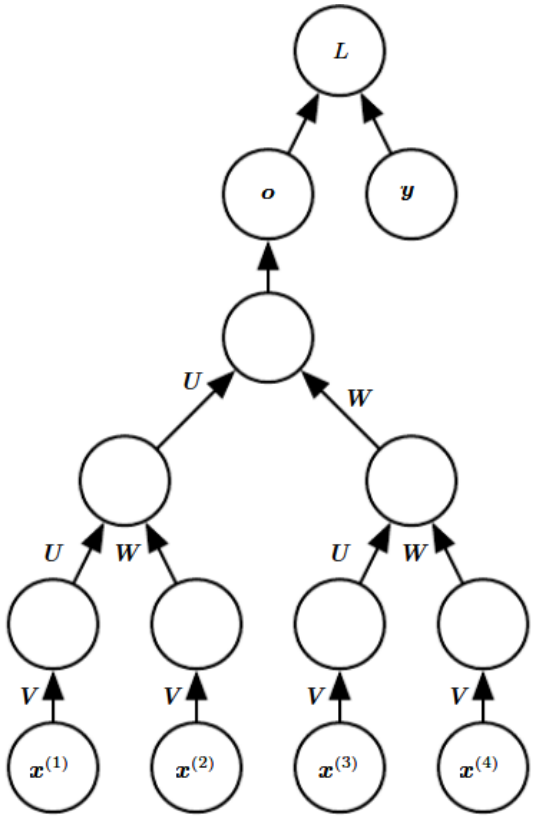
Recurrent NN is special (linear) recursive NN.

Recursive Neural Network (RecNN)

The computational graph of a recursive network is a tree

Instead of backpropagation through time, we have backpropagation through structure (BPTS).

What are V, U, and W?



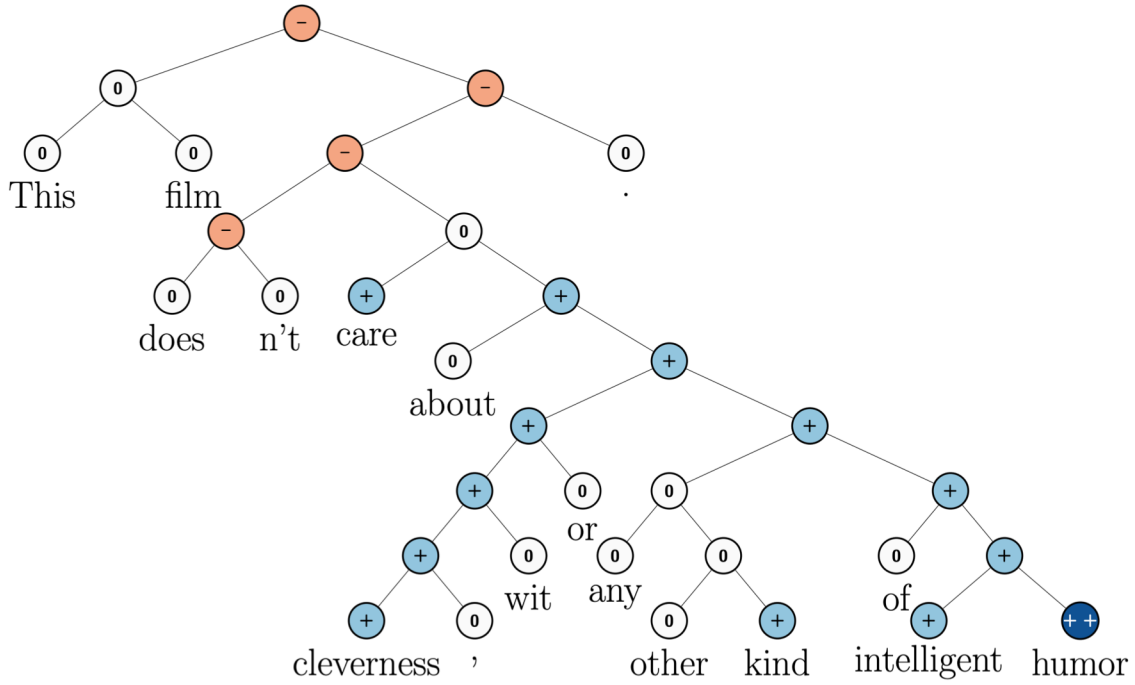
source: http://deeplearningbook.org

RecNN Fine-grain Sentiment Analysis

Use tree structure to discover long-term sentiment dependency (Socher et al, 2013).

Requires explicit representation of the parse tree (what if the parse tree is wrong?)

RecNN enjoyed some success, but as of 2020, not as widely used.



source: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank