Universität Potsdam

Deep Learning for NLP

More on Recurrence

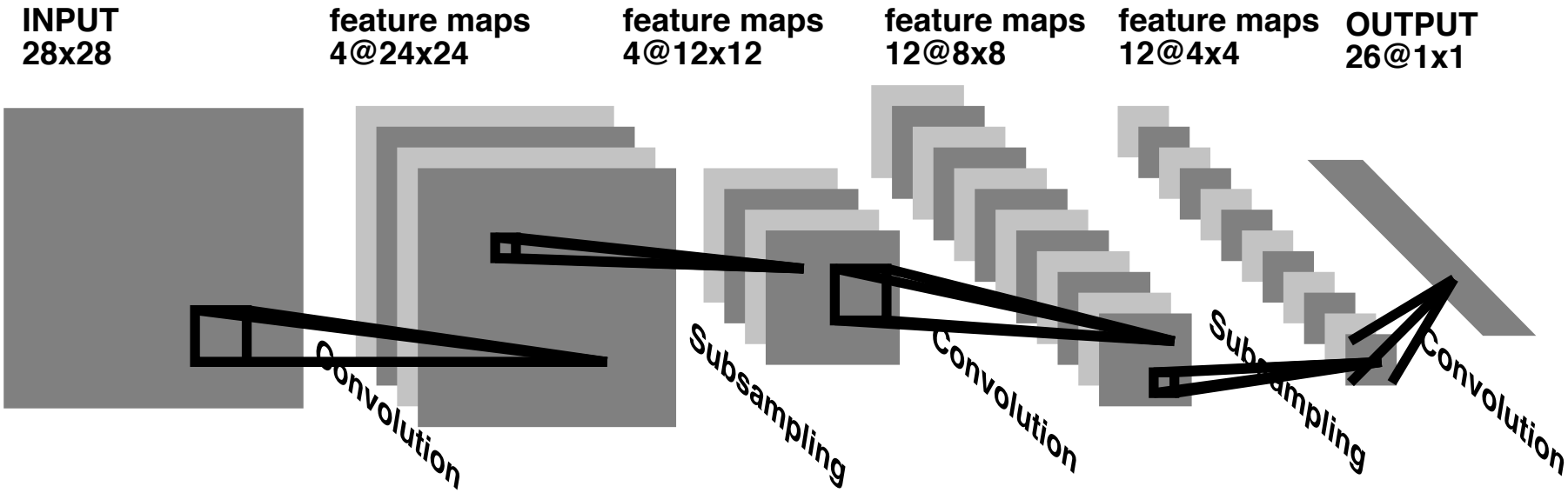
May 06, 2016

Alexander Koller

Sebastian Stober

*with figures from deeplearningbook.org

Convolution

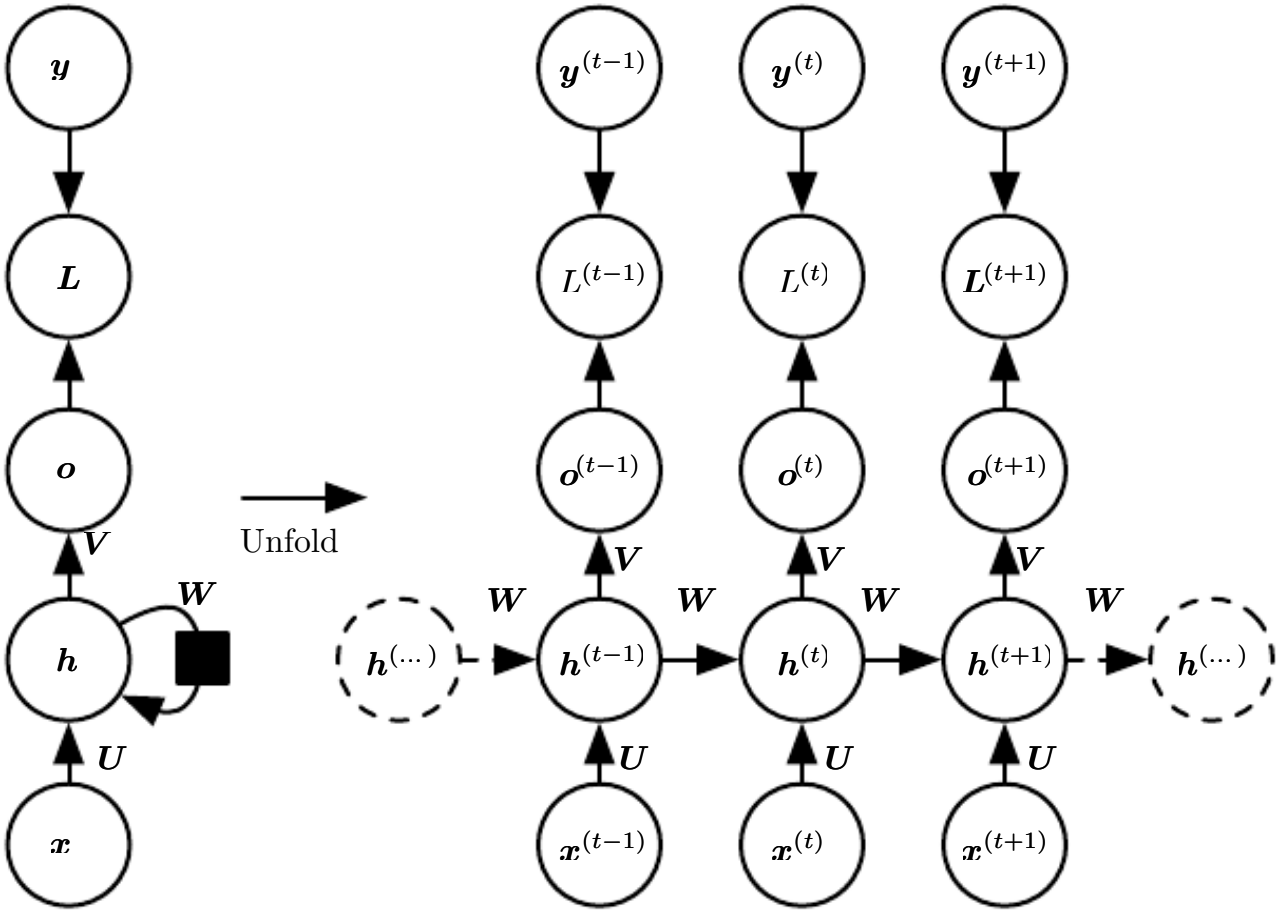


[Y. Bengio and Y. Lecun, 1995]

Recurrence: Motivation

- process sequential data
- capture history of inputs/states
- share parameters through a very deep computational graph
  - output is a function of the previous output
  - produced using the same update rule applied to the previous outputs.
- different from convolution across time steps

The general idea

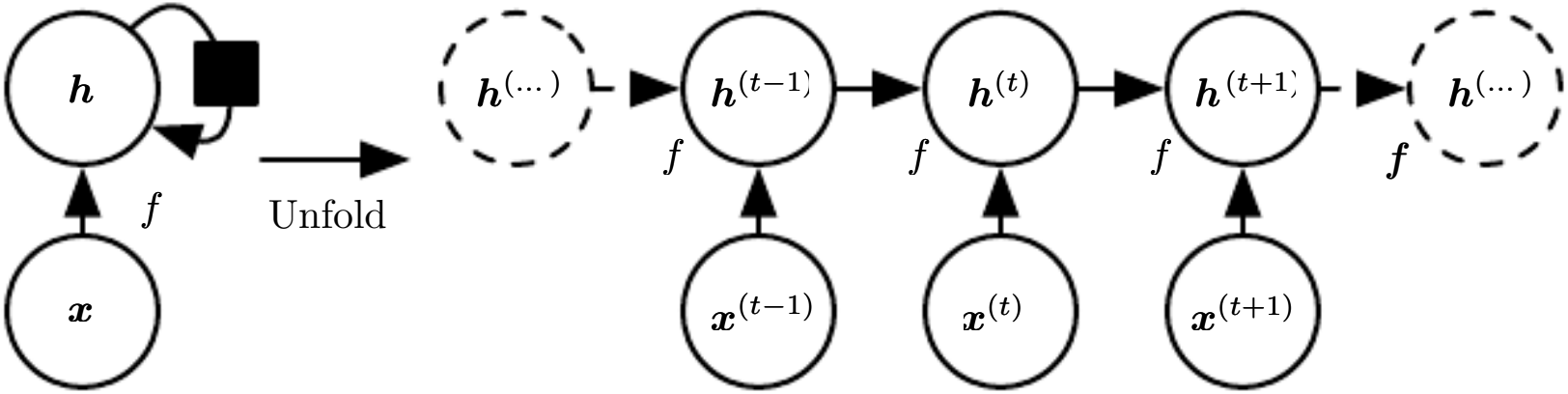


Cyclic Connections

- computational graph includes cycles (recursion)
- represent influence of the present value of a variable on its own value at future time steps

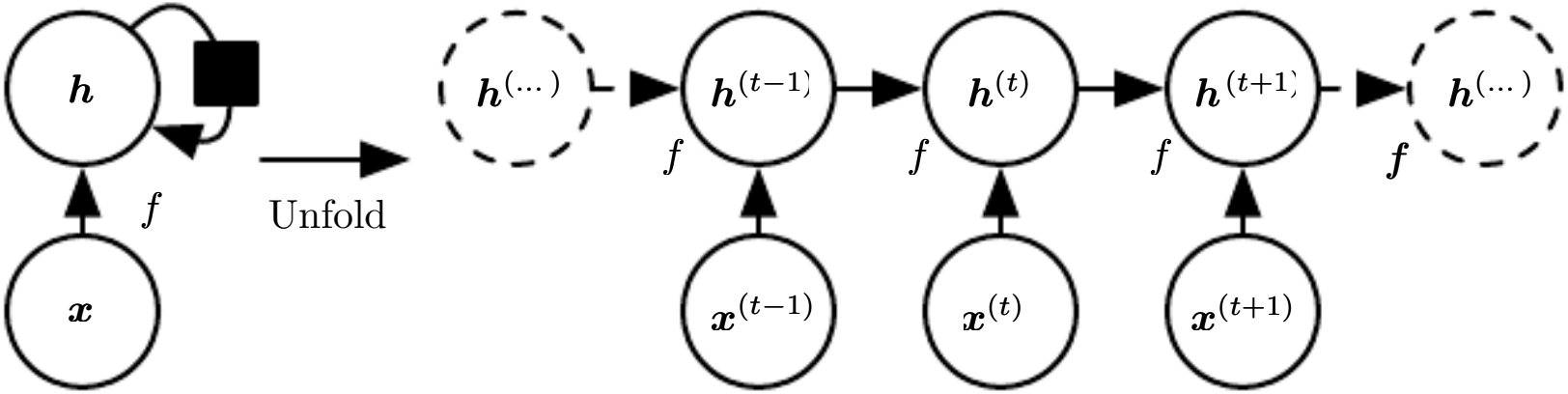
- unfolding to yield a graph that does not involve recurrence
  => gets very deep very quickly

Unfolding



1. regardless of the sequence length, the learned model always has the same input dimensionality
2. can use the same transition function f with the same parameters at every time step

Back-Propagation Through Time (BPTT)



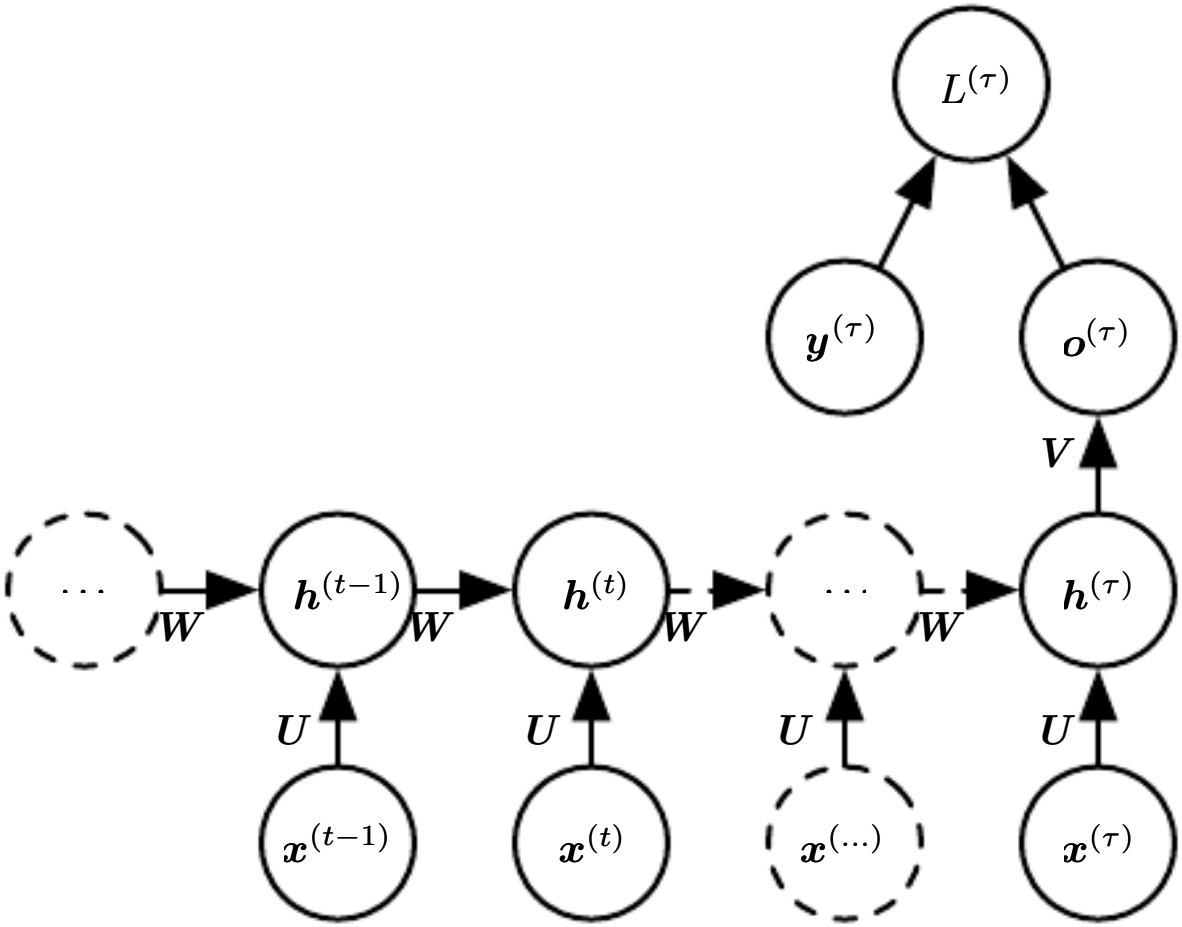
- gradient computation for unfolded loss function w.r.t parameters very expensive
- O(T) where T is history length
- no parallelization (sequential dependence)

Hidden State

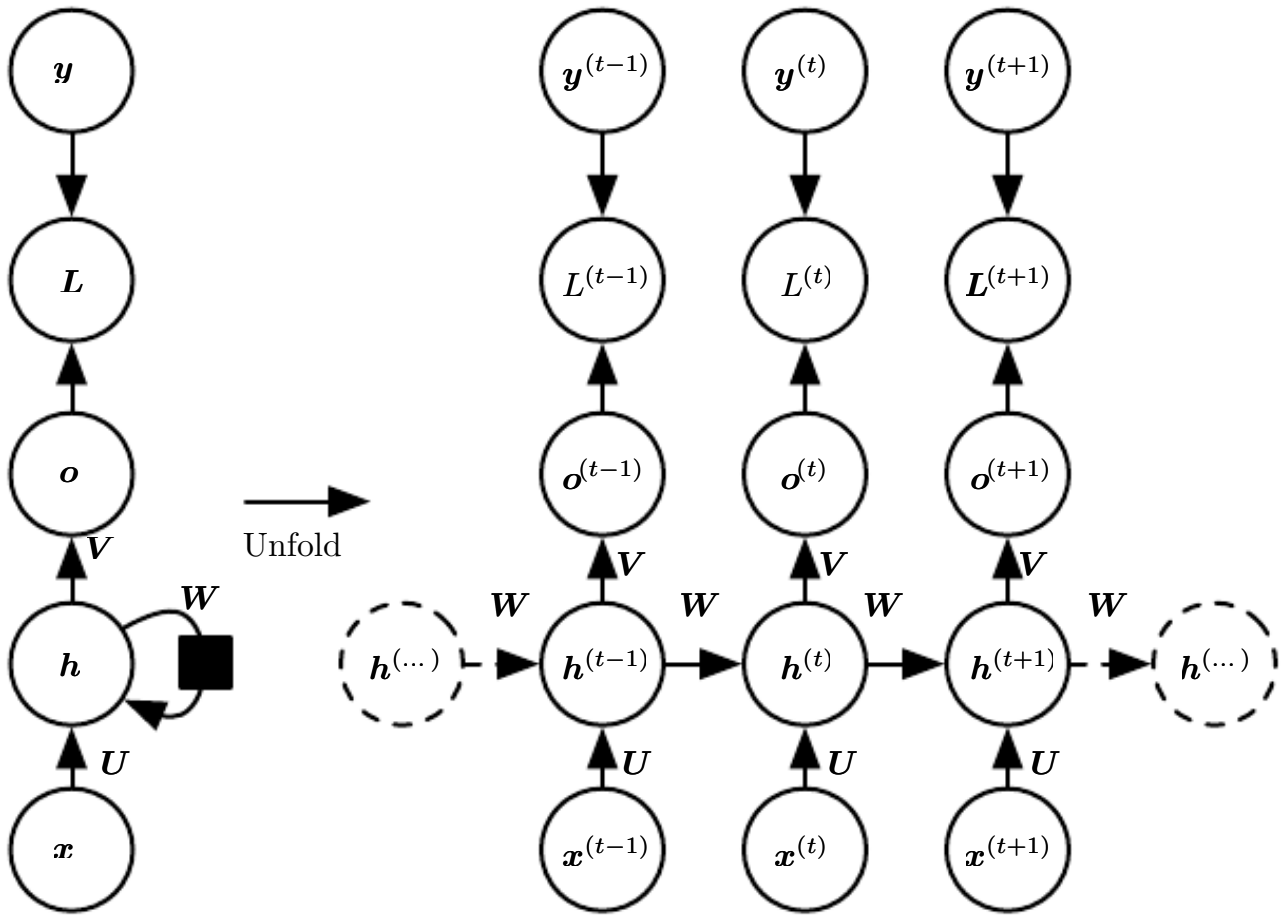
- h(t) as a kind of “lossy summary” of the task-relevant aspects of the history up to t
  - lossy compression necessary
  - selectivity based on training criterion (cost)

- most demanding situation: rich enough representation h(t) to allow approximate recovery of input sequences (autoencoder)

Output for entire sequence

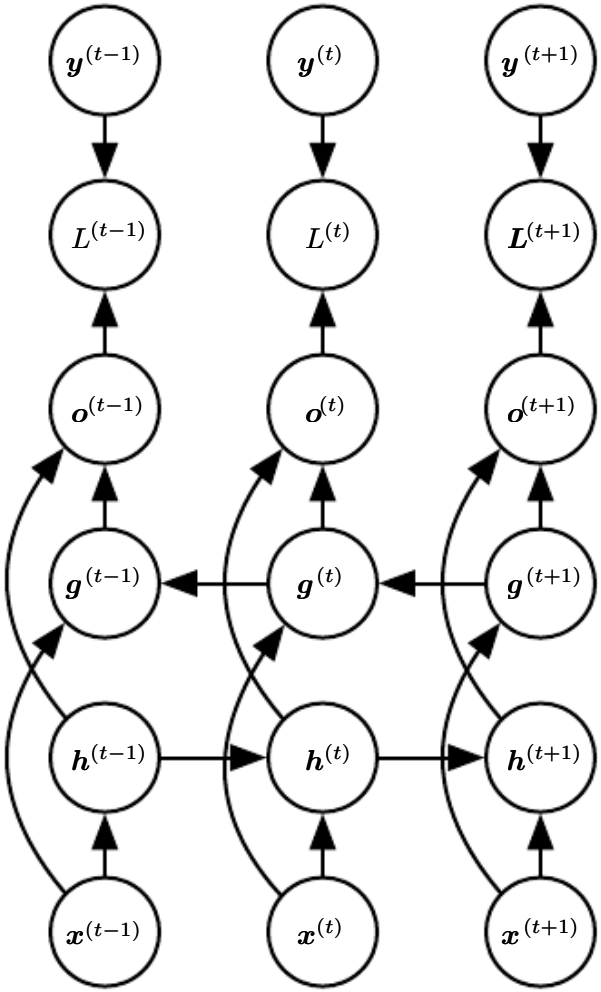


Output at every step



Such a recurrent network of with finite size can compute any function computable by a Turing machine.

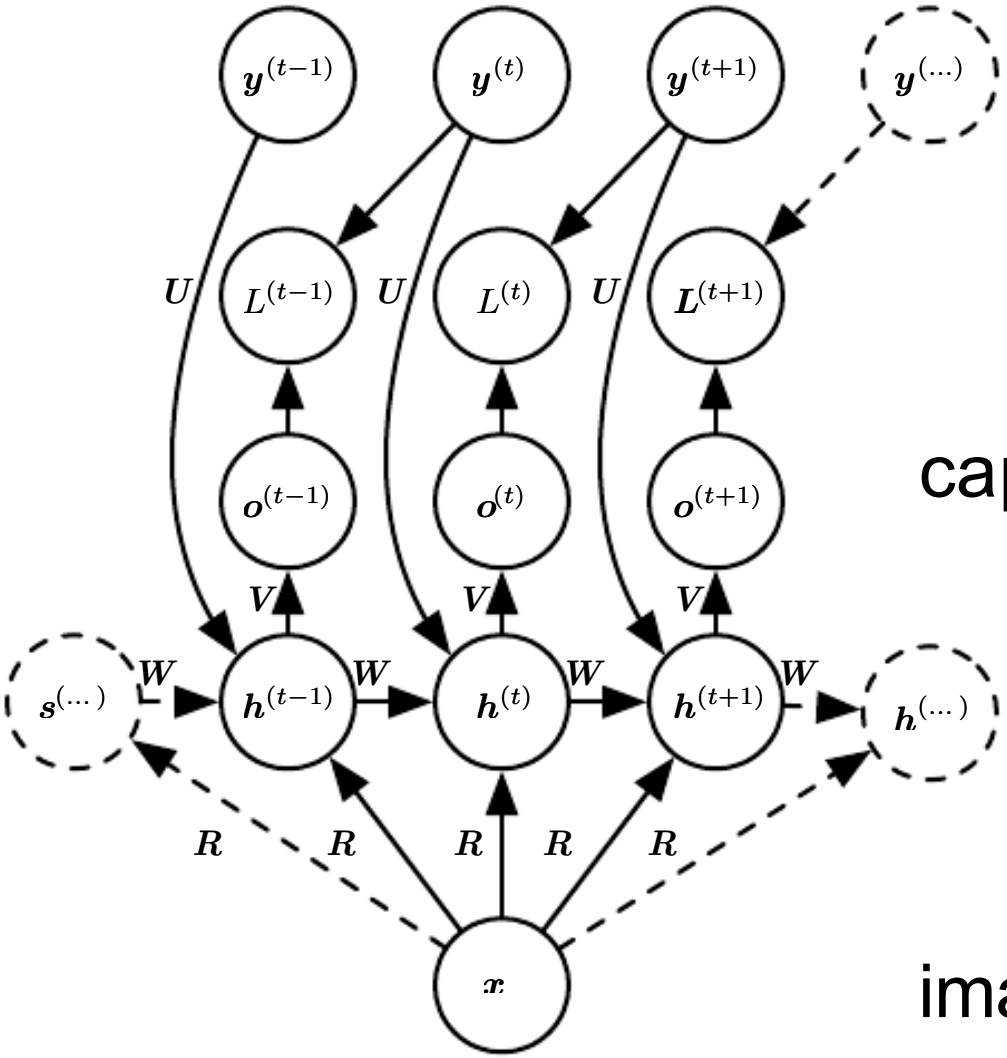
Bi-Directional RNN



- h(t) relevant summary of past (forward)
- g(t) relevant summary of future (backward)

- extendable to 2D inputs
- restriction: same-length sequences

Image labeling

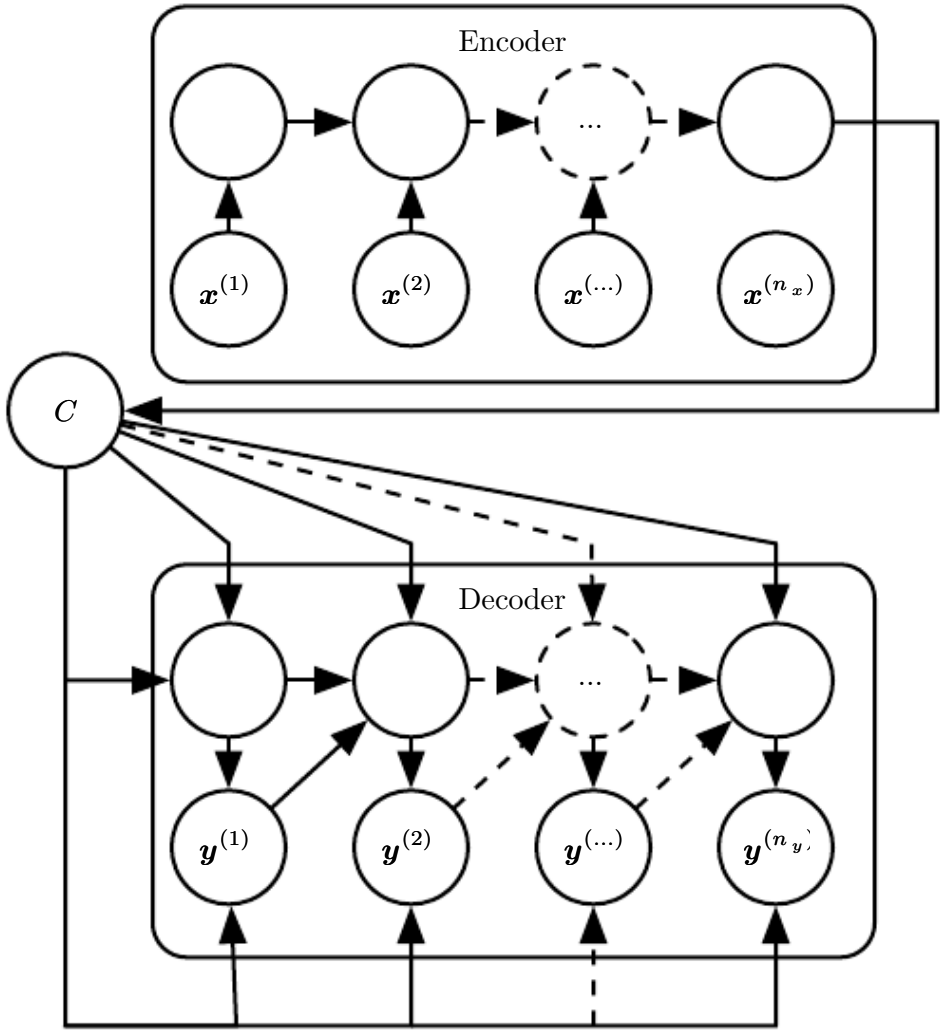


caption (word sequence)

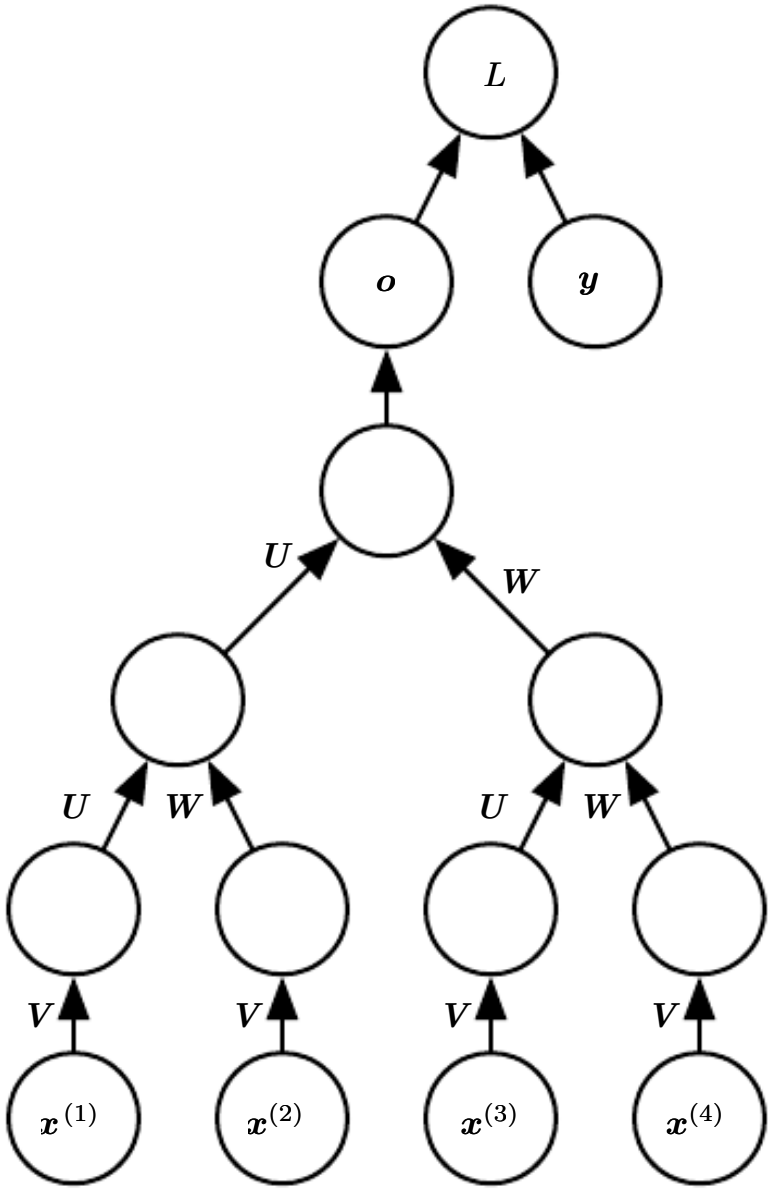
image (2D)

Encoder-Decoder Sequence-to-Sequence

- encoder (reader)
  - read input sequence
  - generate hidden state
- decoder (writer)
  - generate output sequence from hidden state
- variable length



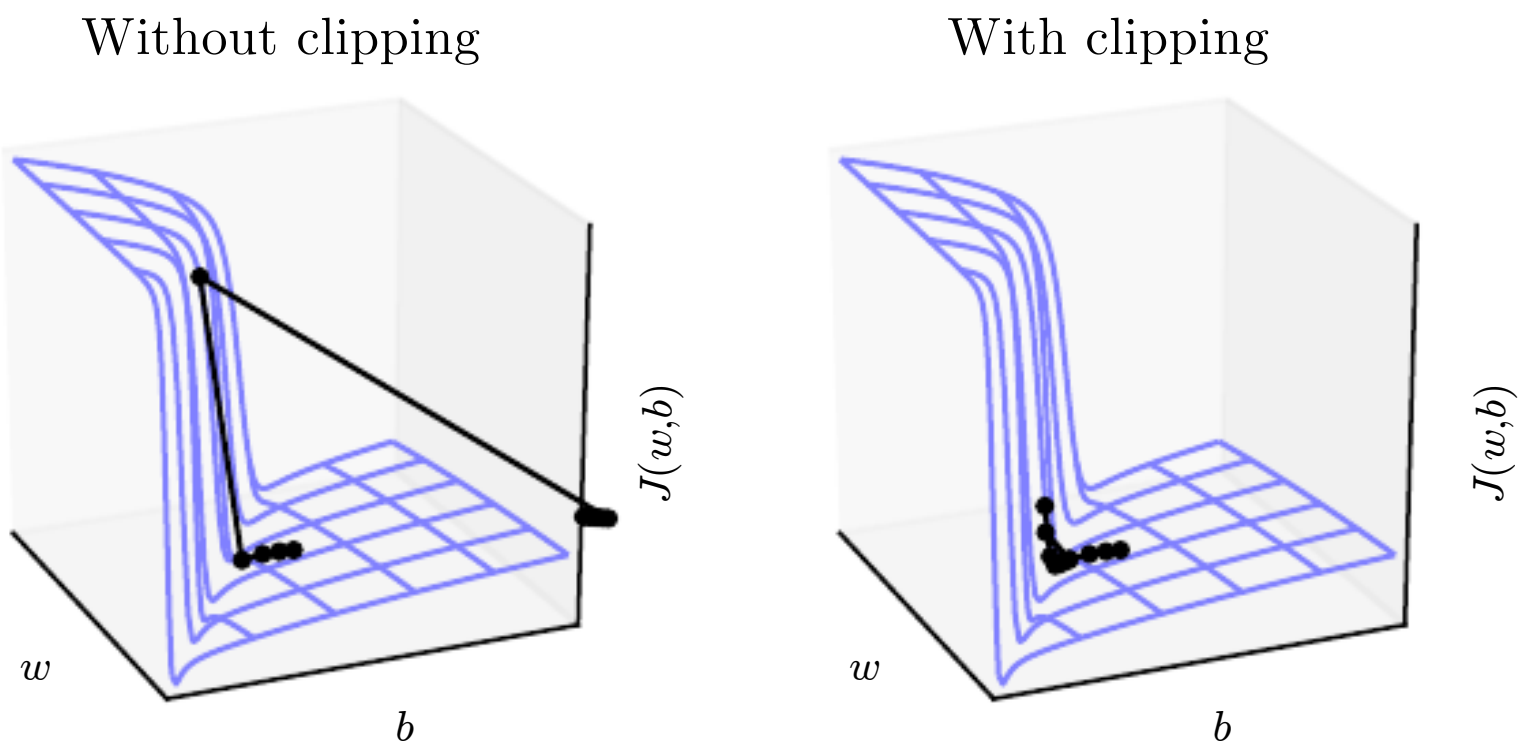
Recursive NNs



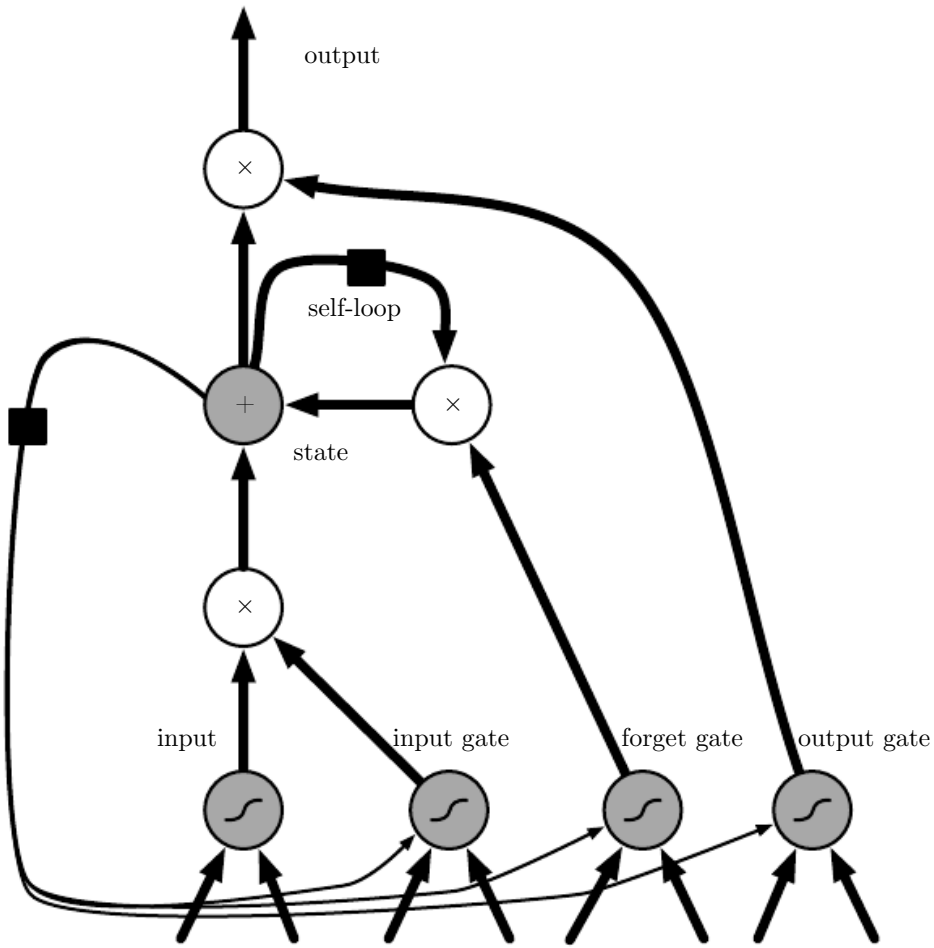
RNNs: Challenges

- Training can be slower than for feedforward or convolutional networks.
- Vanishing/exploding gradients.
- Difficulties in learning to use hidden state to remember information about the distant past.
  - Extremely important in NLP!

Gradient Clipping

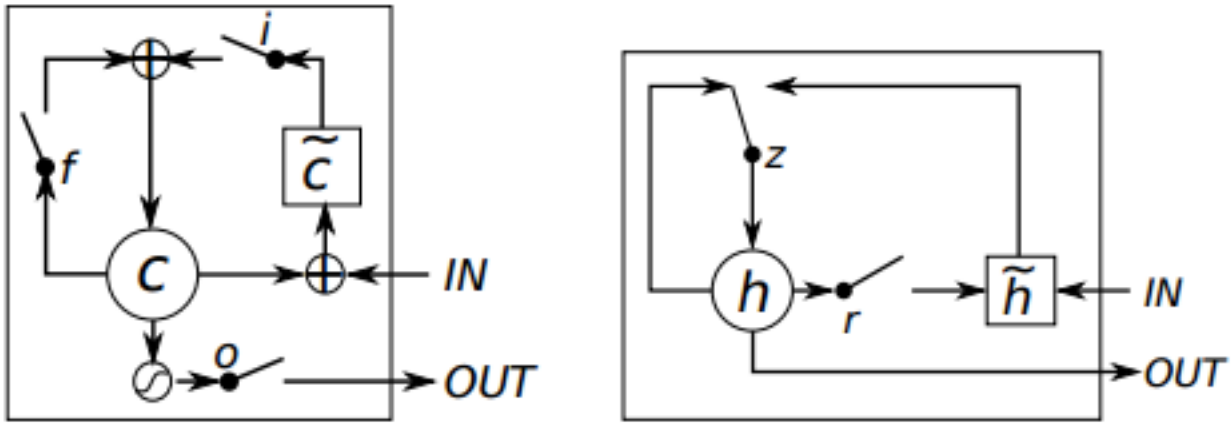


LSTMs



Long Short-Term Memory; Hochreiter & Schmidhuber 97

GRUs



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Figure 1: Illustration of (a) LSTM and (b) gated recurrent units. (a) i, f and o are the input, forget and output gates, respectively. c and $\tilde{c}$ denote the memory cell and the new memory cell content. (b) r and z are the reset and update gates, and h and $\tilde{h}$ are the activation and the candidate activation.

Gated Recurrent Units; Cho et al. 2014

Conclusion

- Recurrent neural networks extremely useful for modeling natural language.
- Special challenges that need to be overcome.
- Many possible solutions available; try them!