Recurrent Neural Networks

Ling 282/482: Deep Learning for Computational Linguistics

C.M. Downey

Fall 2025

RNNs: high-level

RNNs: high-level

- Feed-forward networks: **fixed-size** input, fixed-size output
  - Feedforward LM: n-gram assumption (i.e. **fixed-size context** of word embeddings)

RNNs: high-level

- Feed-forward networks: **fixed-size** input, fixed-size output
  - Feedforward LM: n-gram assumption (i.e. **fixed-size context** of word embeddings)
- RNNs process (arbitrarily long) **sequences** of vectors
  - Maintaining a "hidden state" through time
  - Applying the **same operation at each step**

RNNs: high-level

- Feed-forward networks: **fixed-size** input, fixed-size output
  - Feedforward LM: n-gram assumption (i.e. **fixed-size context** of word embeddings)
- RNNs process (arbitrarily long) **sequences** of vectors
  - Maintaining a "hidden state" through time
  - Applying the **same operation at each step**
- Different RNNs
  - Different operations at each step
  - Operation also called “recurrent cell”
  - Other architectural considerations (e.g. depth, bidirectionally)

Long-distance dependencies: agreement

Long-distance dependencies: agreement

- Language modeling (fill-in-the-blank)
  - The keys ____
  - The keys on the table ____
  - The keys next to the book on top of the table ____

Long-distance dependencies: agreement

- Language modeling (fill-in-the-blank)
  - The keys ____
  - The keys on the table ____
  - The keys next to the book on top of the table ____
- The verb needs to **agree in number** with the subject, which can be **far away**
  - And number often **disagrees with linearly-close nouns**

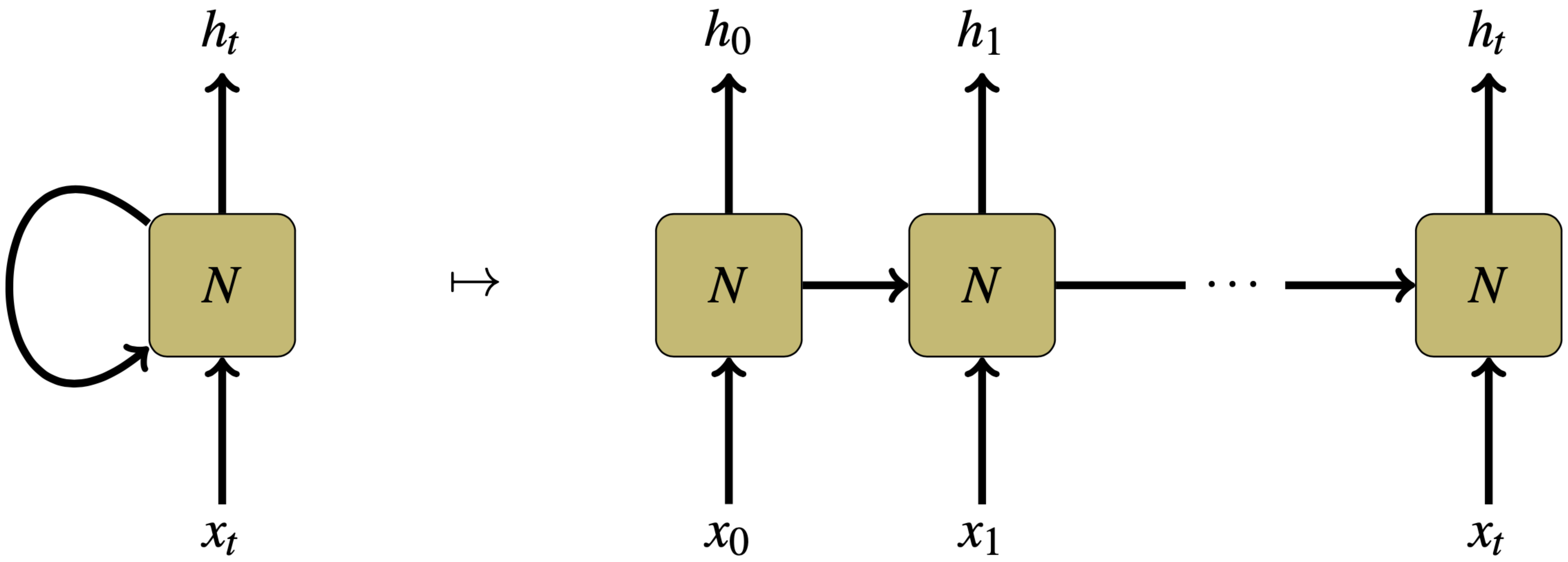
Selectional Restrictions

- "The **family** moved from the city because they wanted a larger ____"
- "The **team** moved from the city because they wanted a larger ____"

Selectional Restrictions

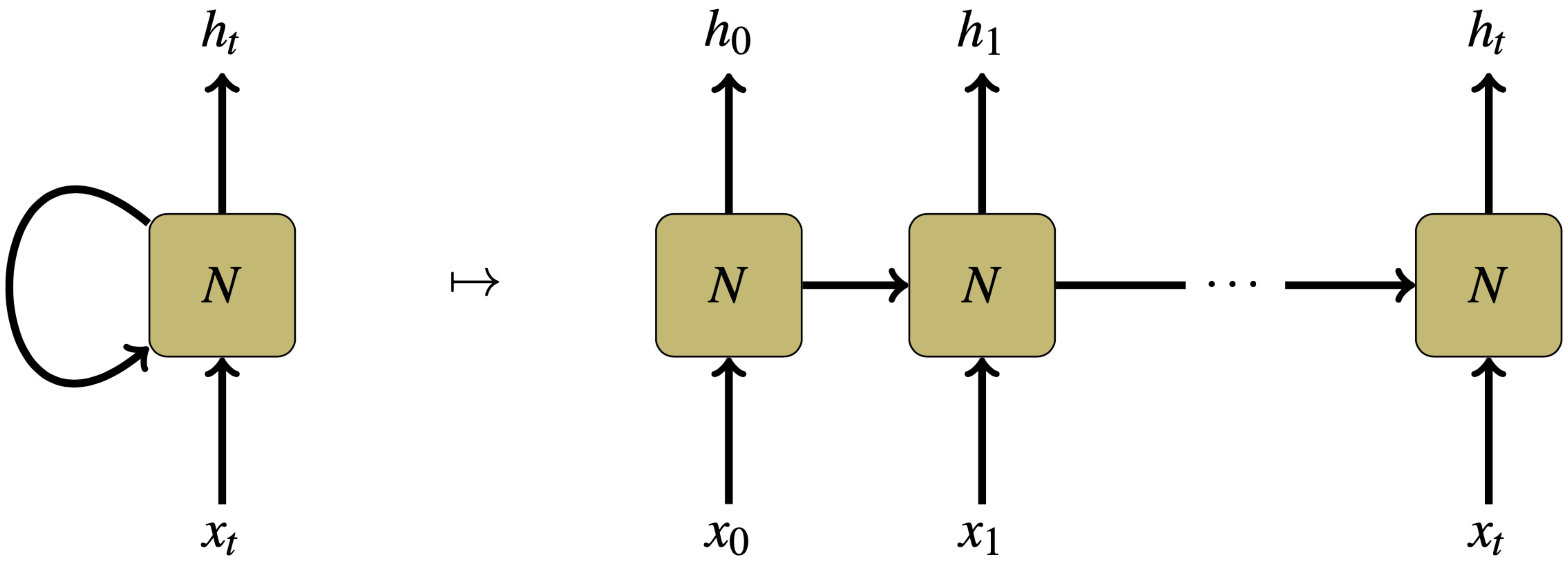
- "The **family** moved from the city because they wanted a larger **house"**
- "The **team** moved from the city because they wanted a larger **market"**
- Need models that can capture long-range dependencies like this.
- N-gram (whether count-based or neural) **cannot** (e.g. with n=4)
  - P(word | “they wanted a larger”)

RNNs



Steinert-Threlkeld and Szymanik 2019; Olah 2015

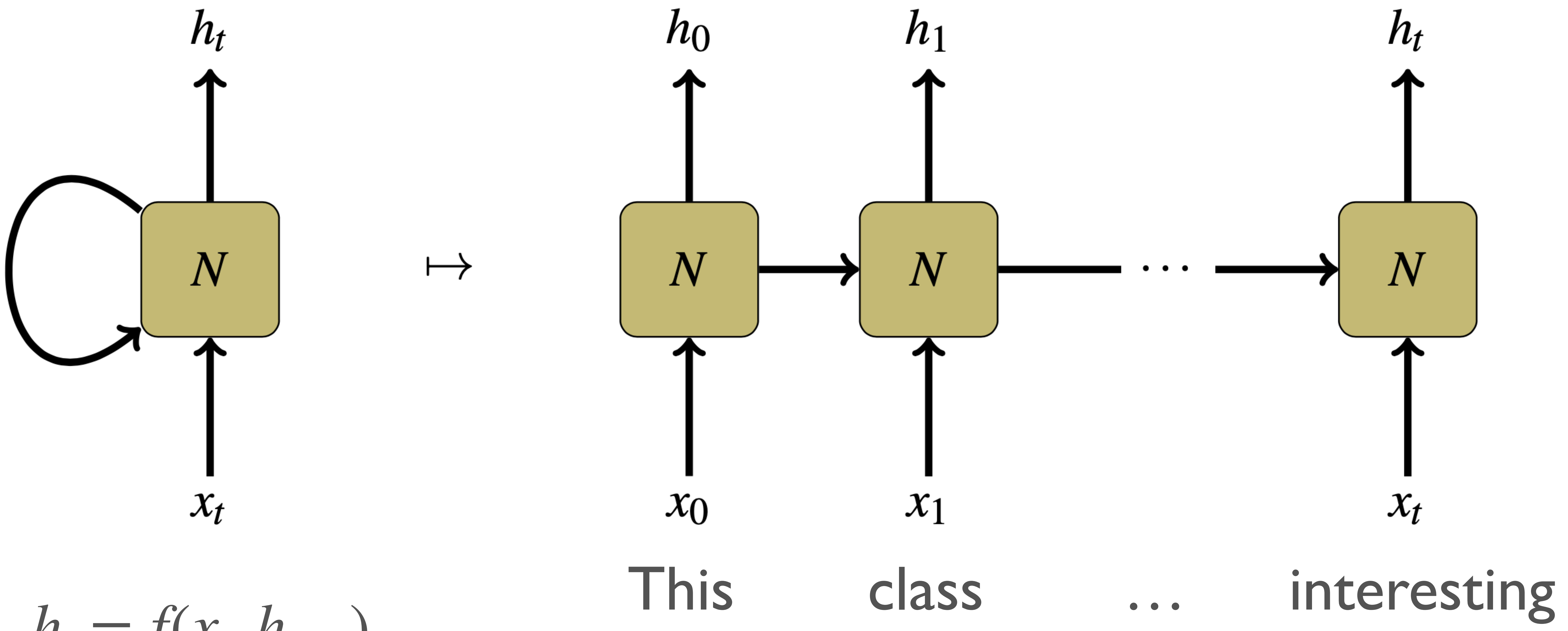
RNNs



$$h_t = f(x_t, h_{t-1})$$

Steinert-Threlkeld and Szymanik 2019; Olah 2015

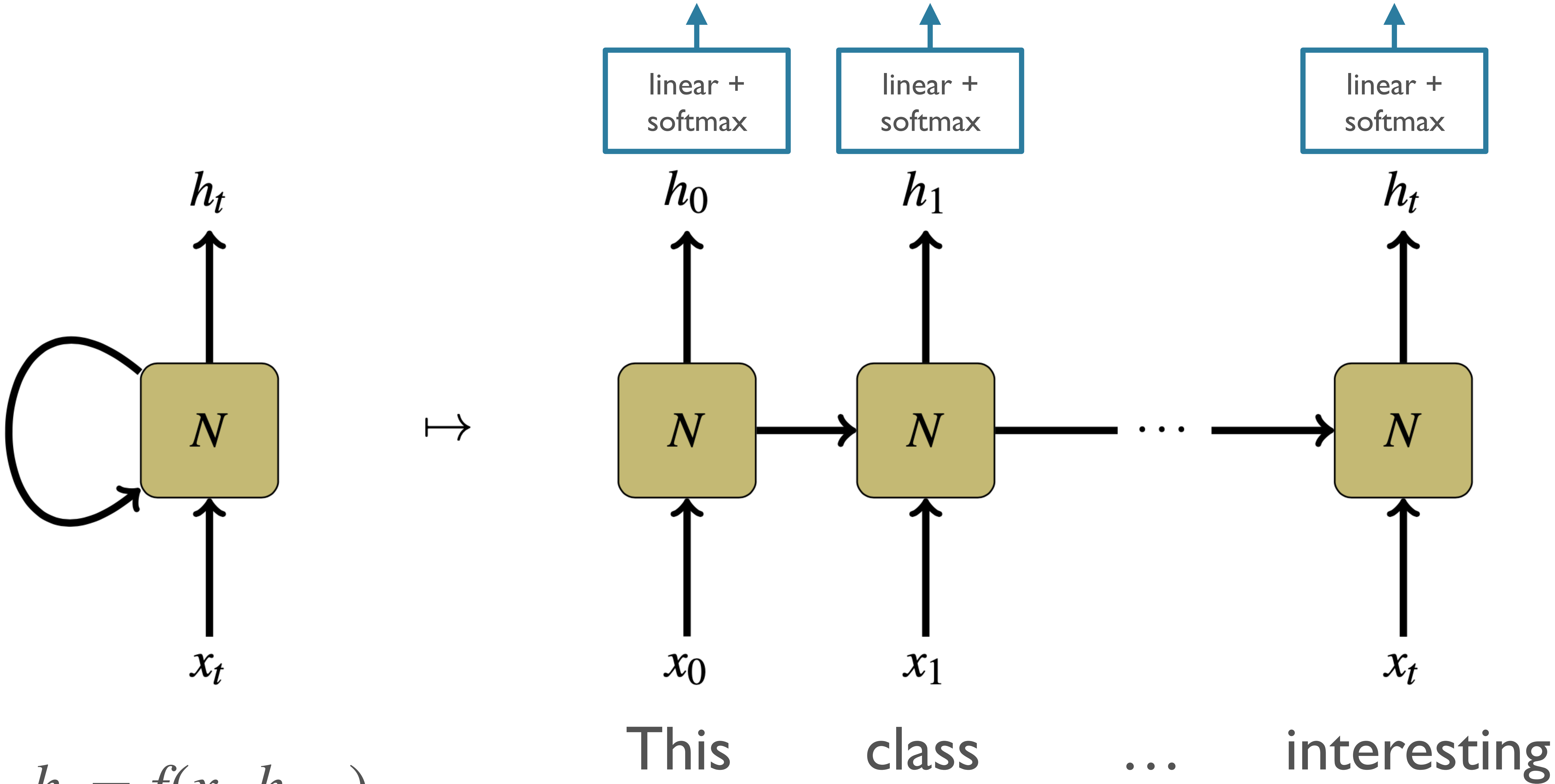
RNNs



$$h_t = f(x_t, h_{t-1})$$

Steinert-Threlkeld and Szymanik 2019; Olah 2015

RNNs



$$h_t = f(x_t, h_{t-1})$$

Steinert-Threlkeld and Szymanik 2019; Olah 2015

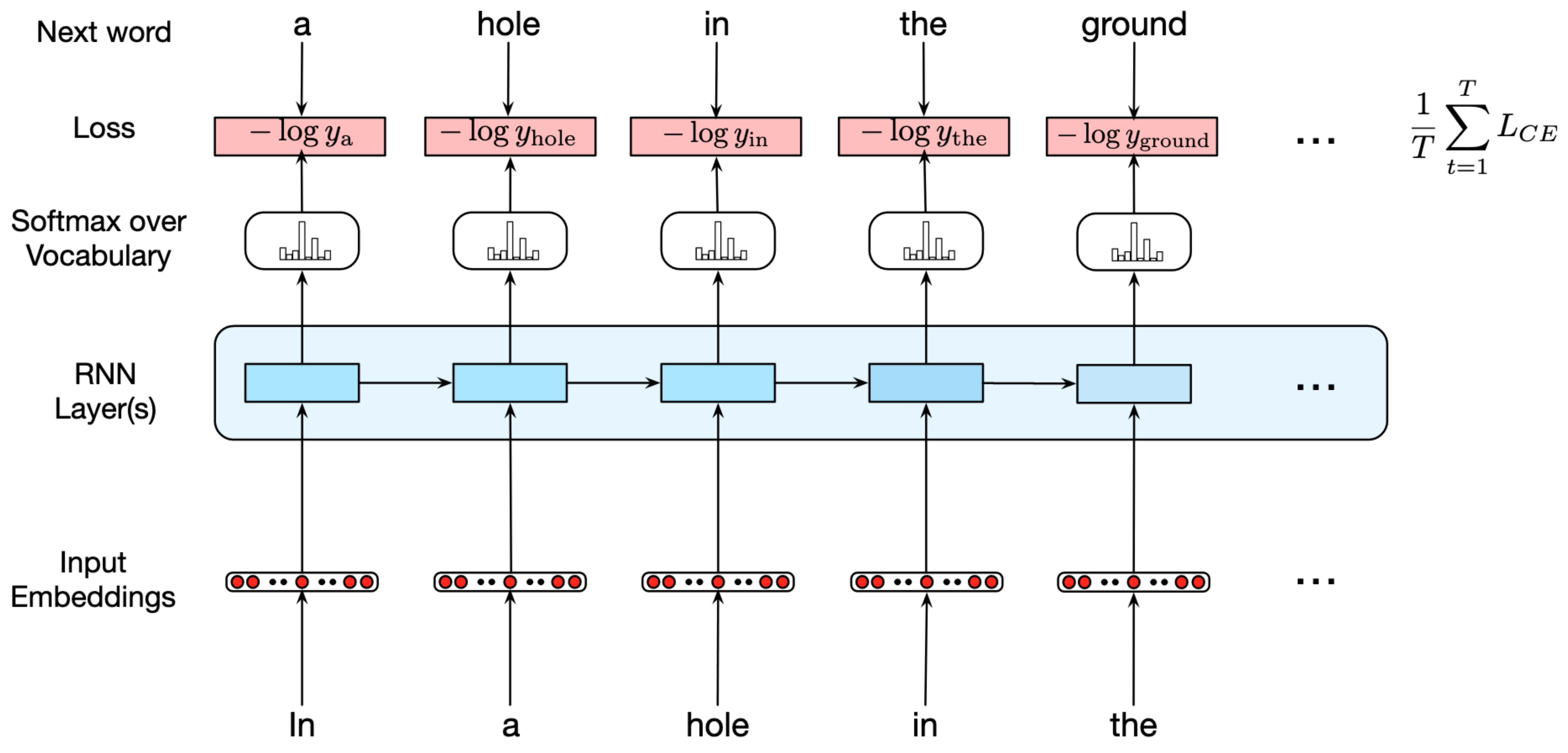
Simple / Vanilla / Elman RNNs

- Same kind of **feed-forward** computation we've been studying, but:
  - x_t: **sequence element** at time *t*
  - h_{t-1}: **hidden state of the model** at previous time *t-1*
- At each step, apply weight matrix to **both** current input and previous h
  - "Carry over" information from **preceding sequence**

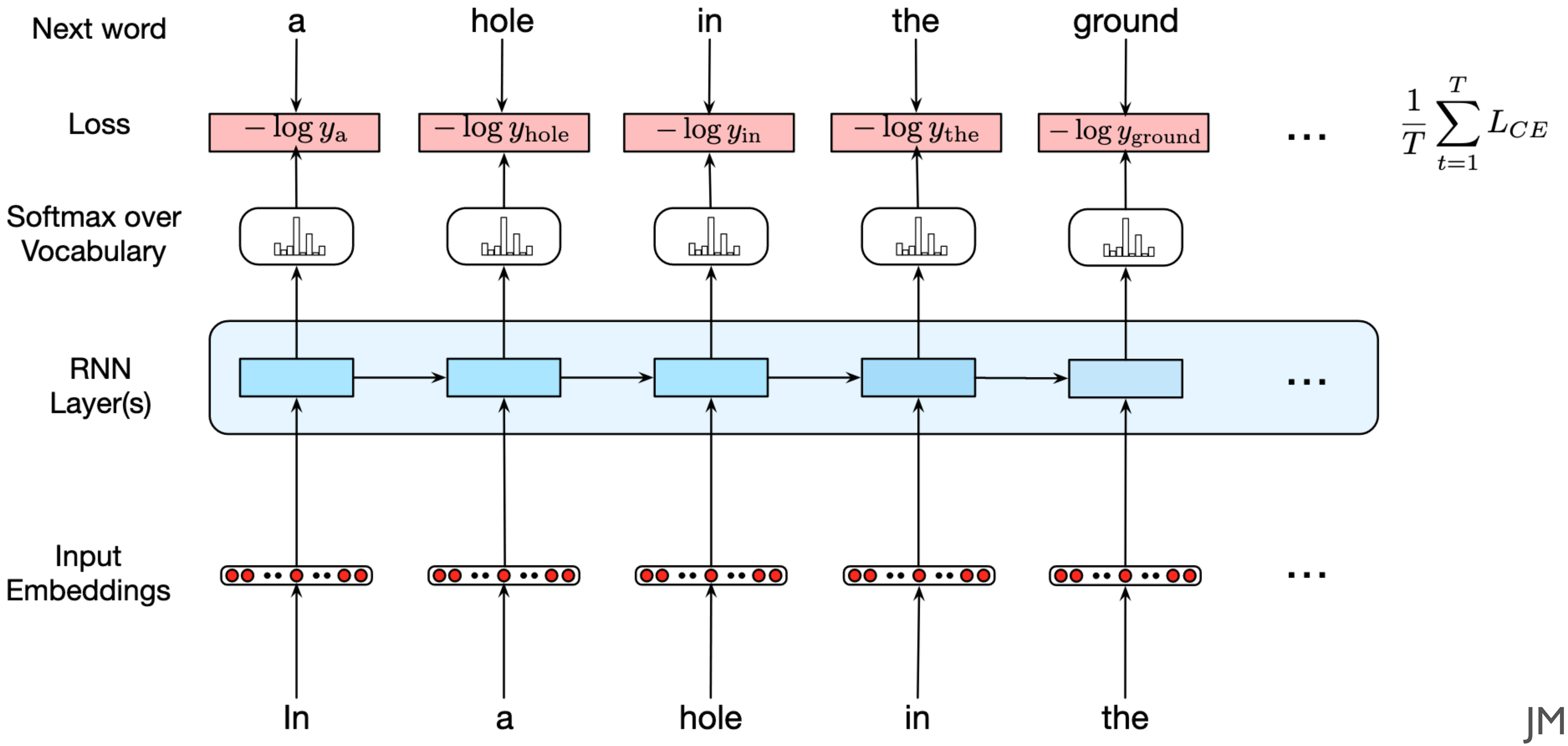
Simple/"Vanilla" RNN: $$h_t = \tanh(W_x x_t + W_h h_{t-1} + b)$$

Training: BPTT

- Algorithm called Backpropagation Through Time
- “Unroll” the network **across time-steps**
  - Easier to understand visually (next slide)
- Apply backpropagation to the “wide” network
  - Each cell has the **same parameters**
  - Gradients **sum across time-steps**
  - Multi-variable chain rule



"Unrolled" RNN

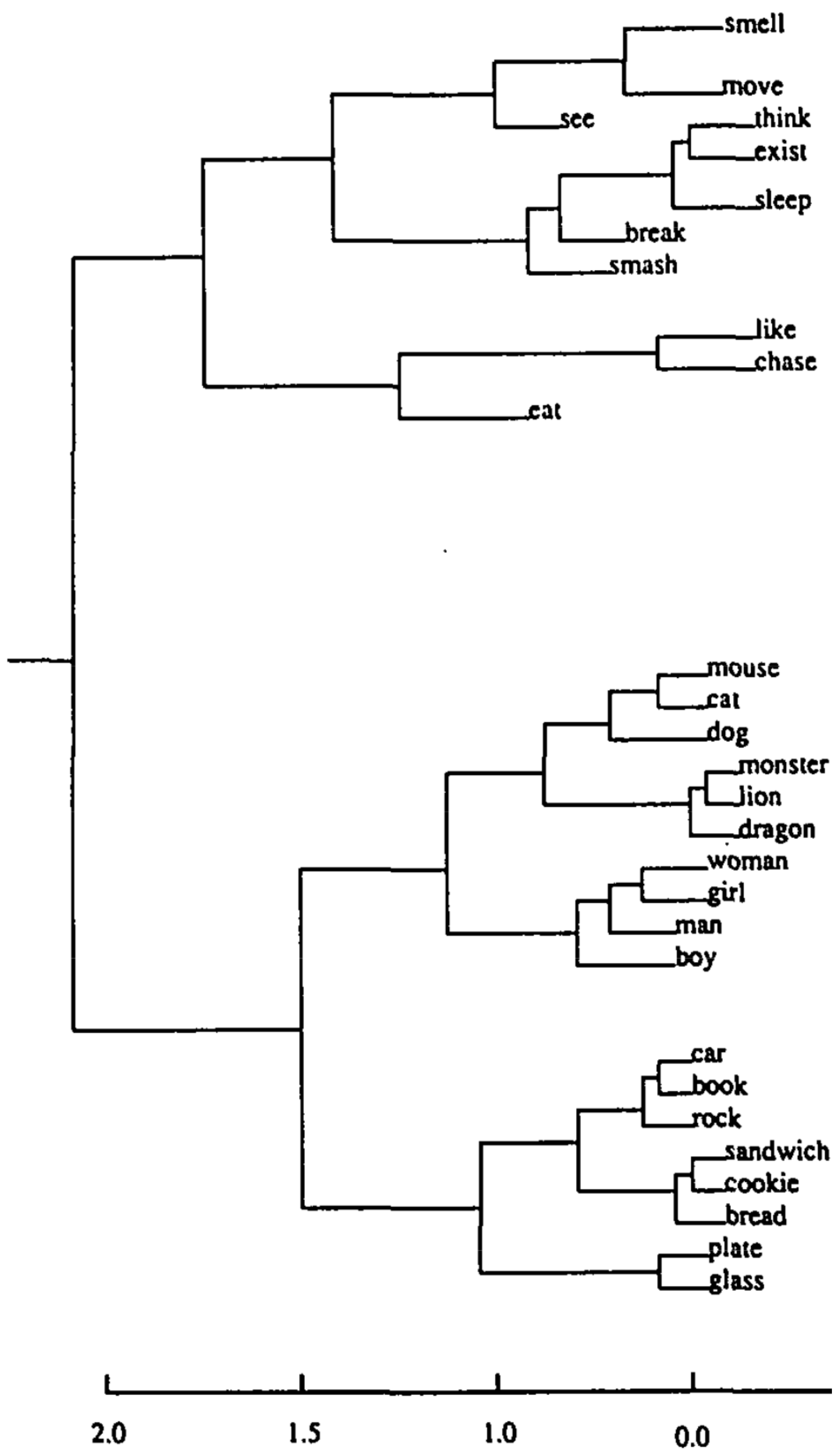


JM sec 9.2.3

Power of RNNs

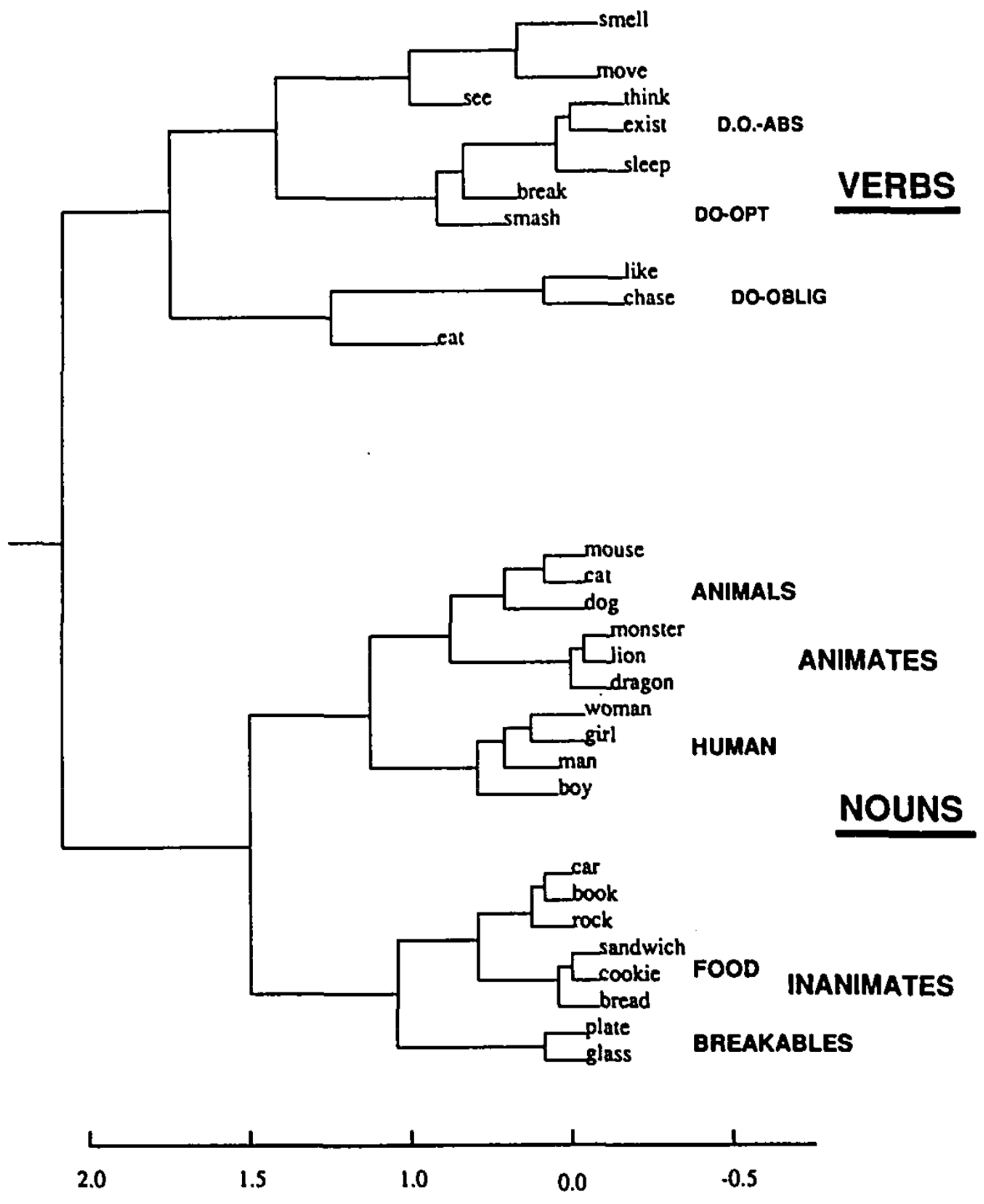
Hierarchical clustering of Vanilla RNN hidden states trained as LM on synthetic data

What trends do you notice?



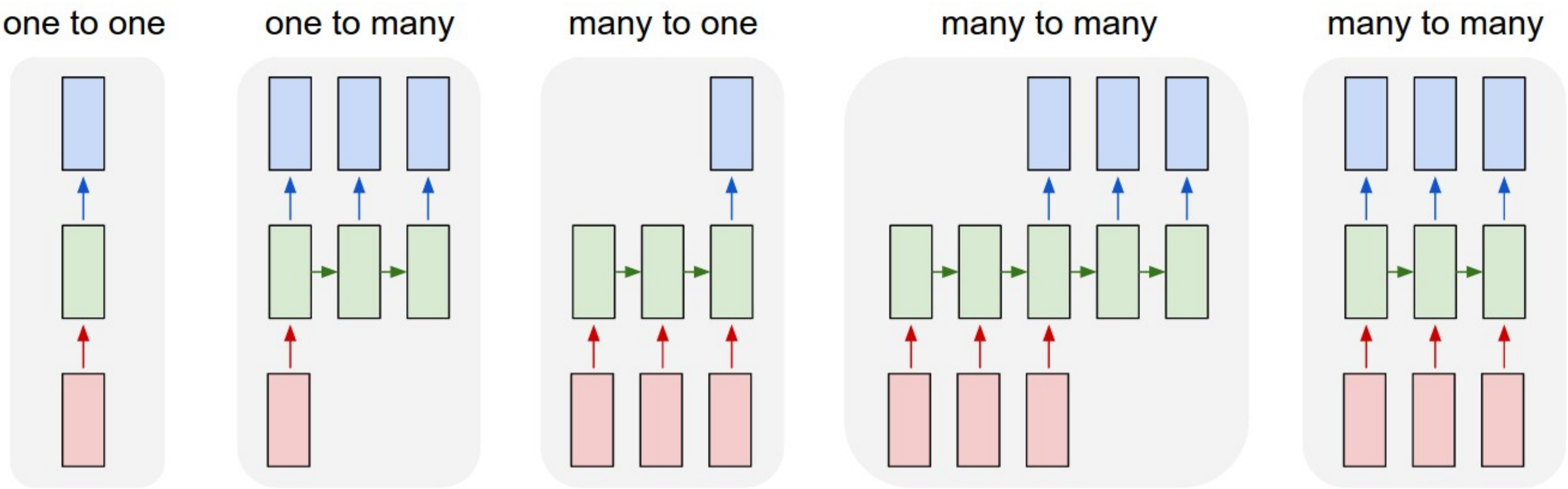
Elman 1990

Power of RNNs



[Elman 1990]

Using RNNs

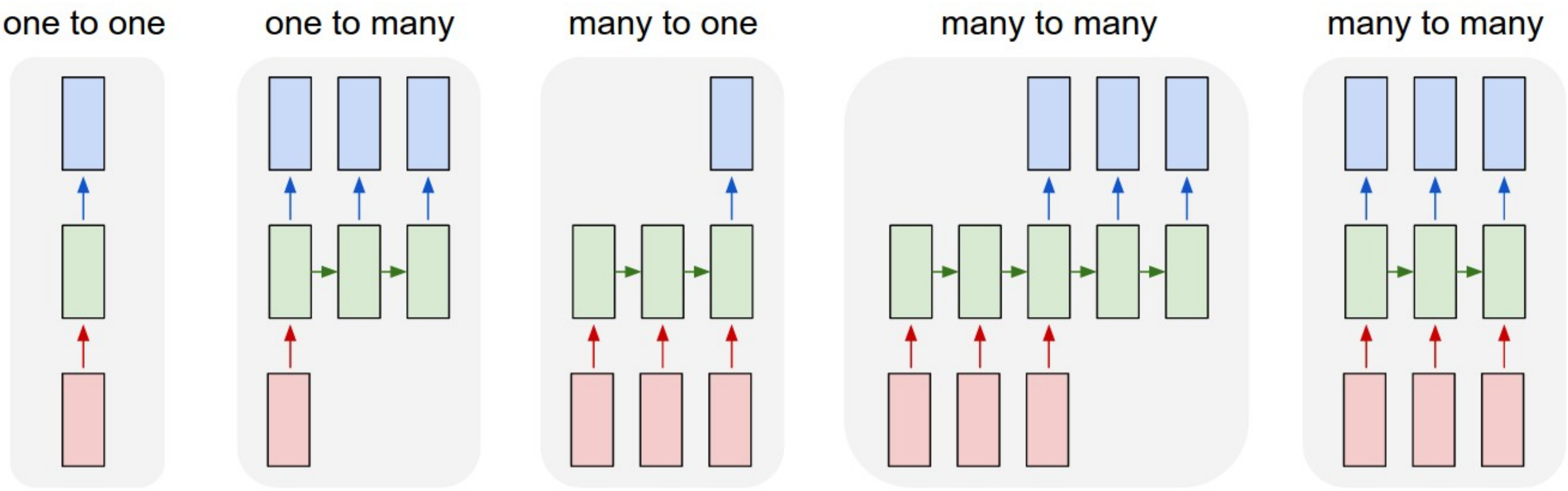


MLP

e.g. image captioning

e.g. text classification

Using RNNs



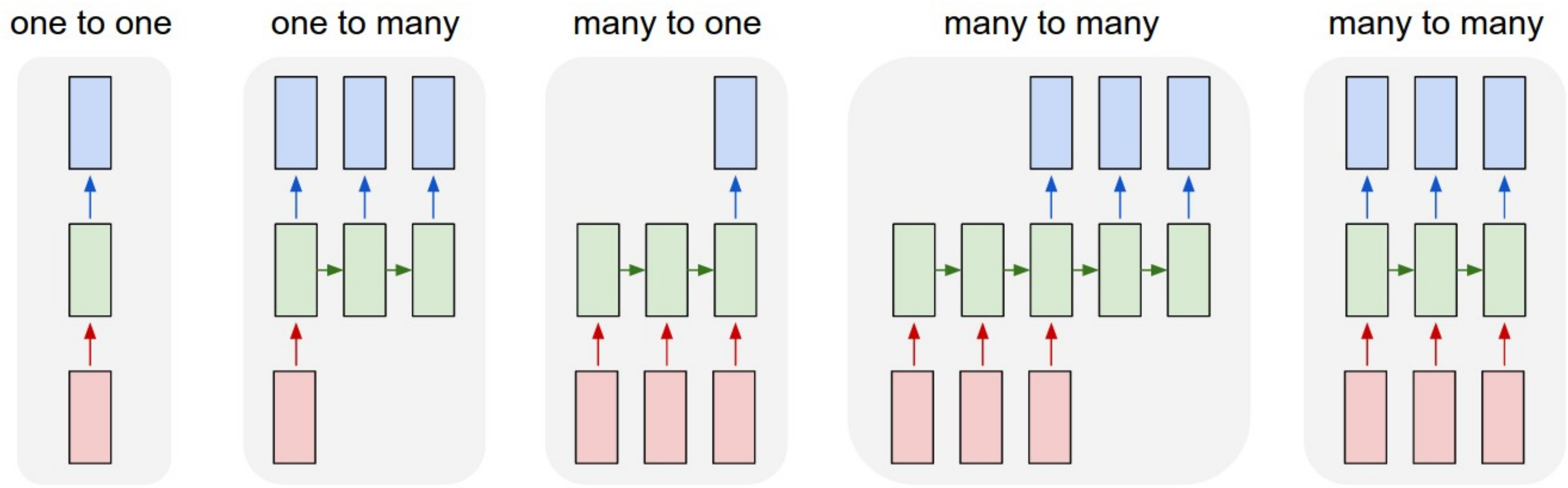
MLP

e.g. image captioning

e.g. text classification

e.g. POS tagging

Using RNNs



MLP

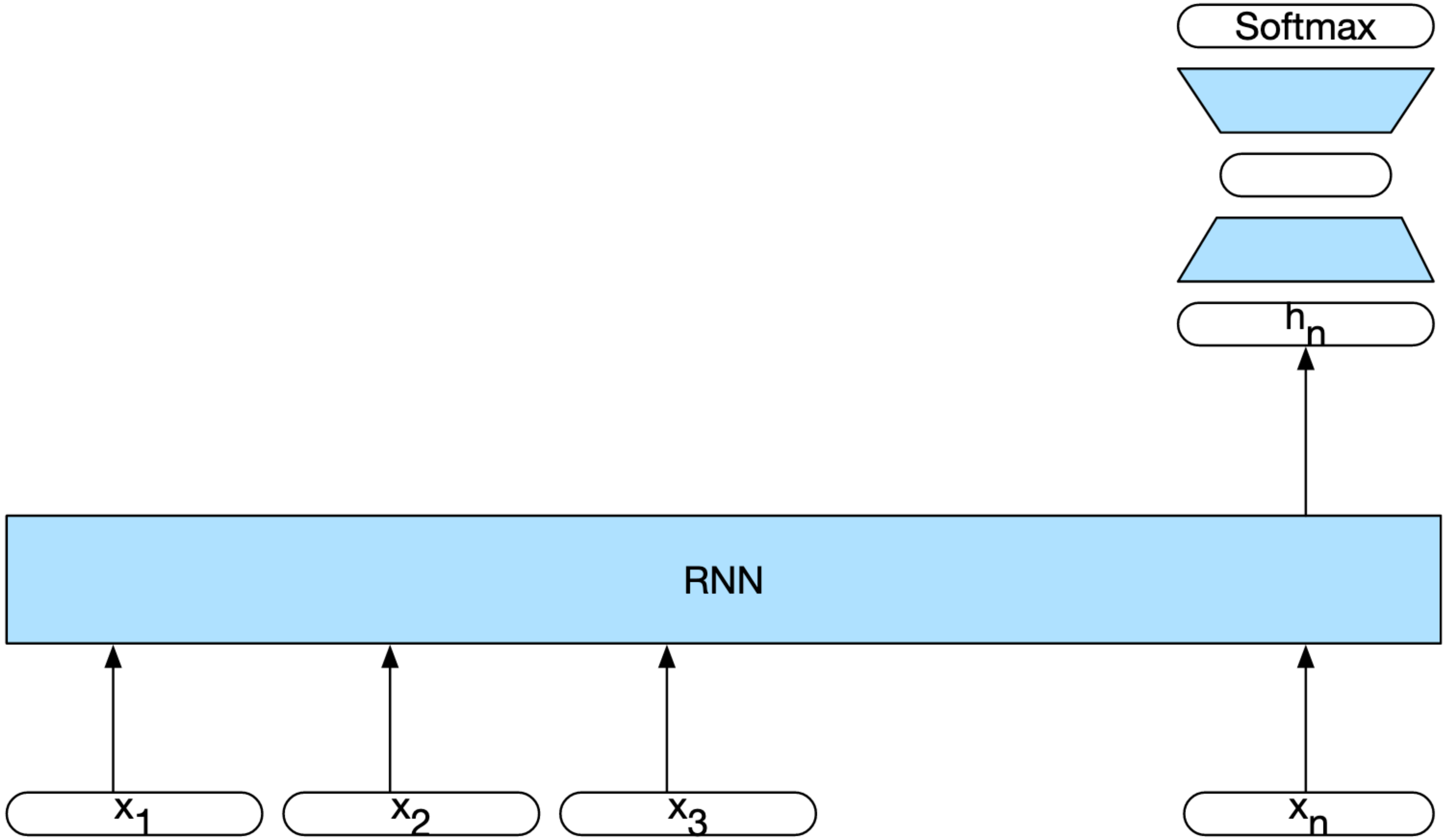
e.g. image captioning

e.g. text classification

seq2seq (later)

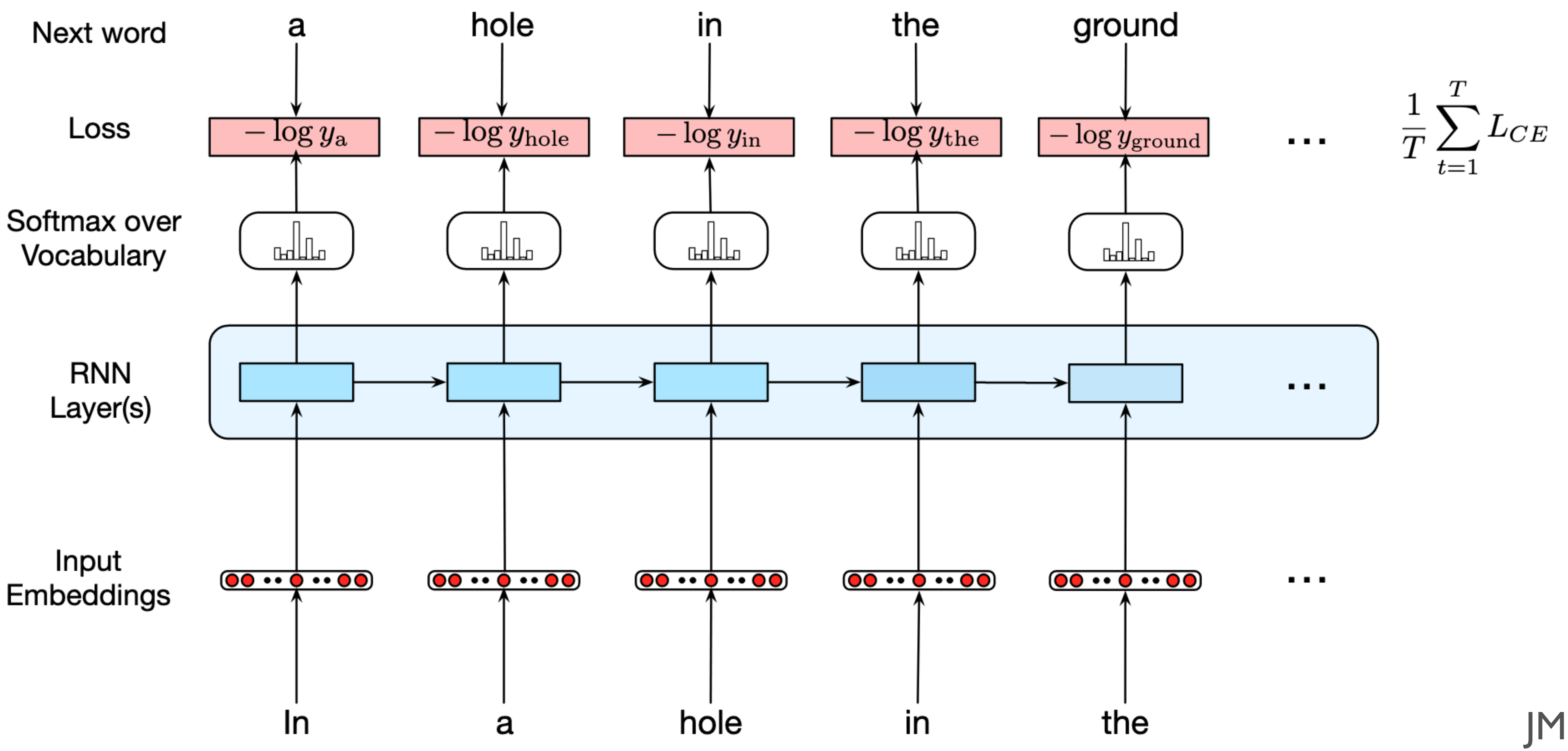
e.g. POS tagging

RNN for Text Classification



JM sec 9.2.5

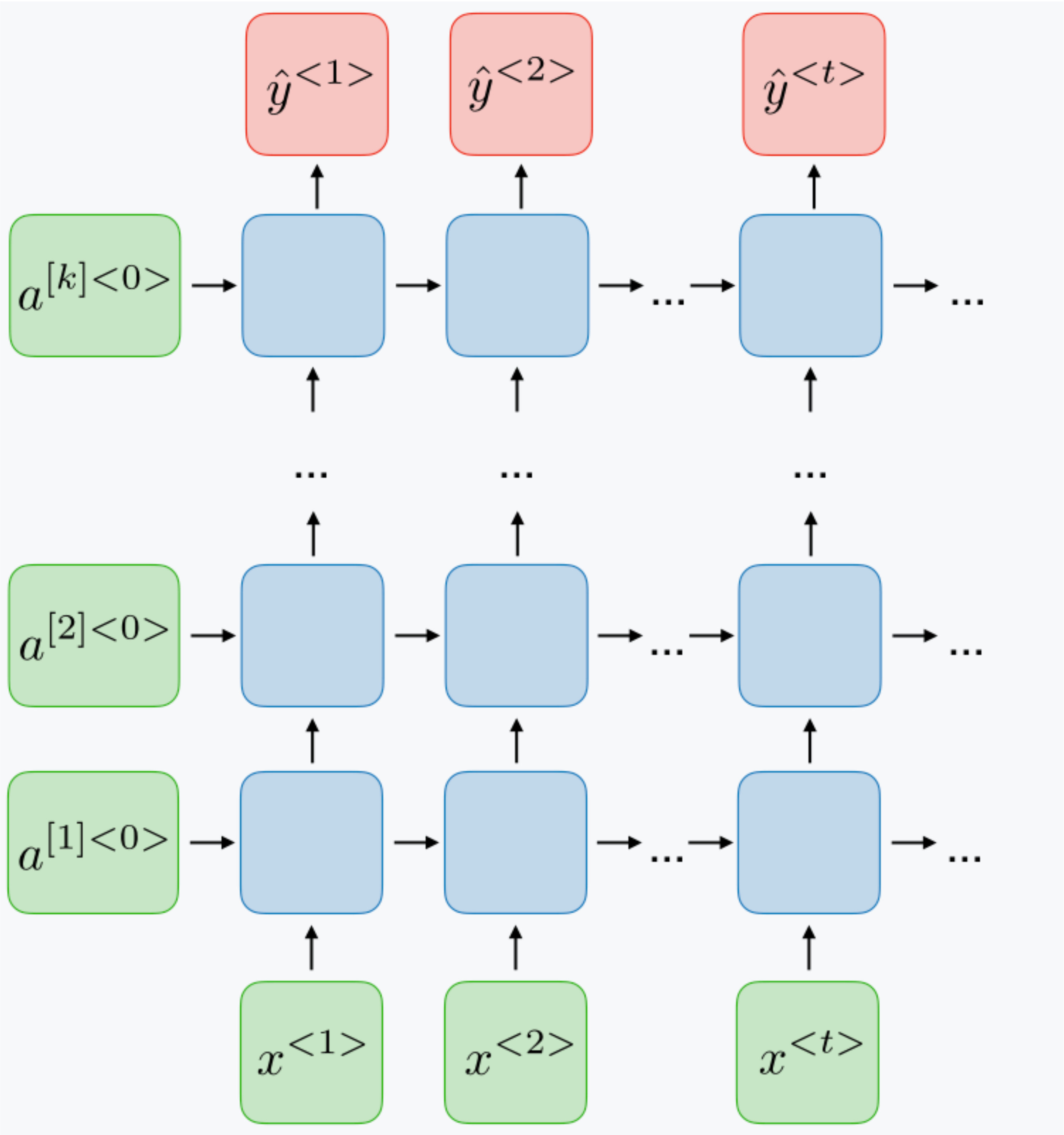
RNNs for Language Modeling



JM sec 9.2.3

Two Extensions

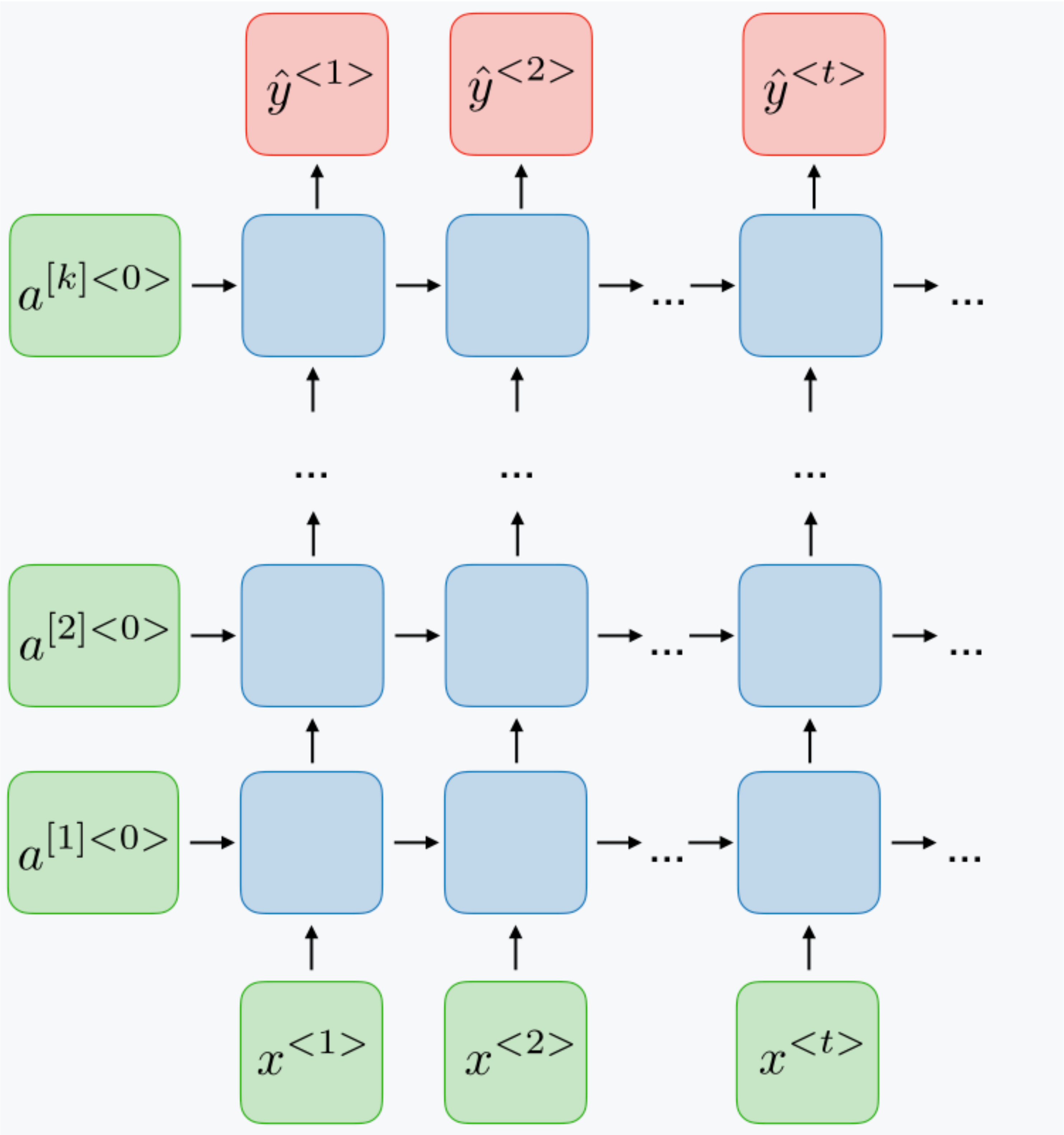
- Deep RNNs:



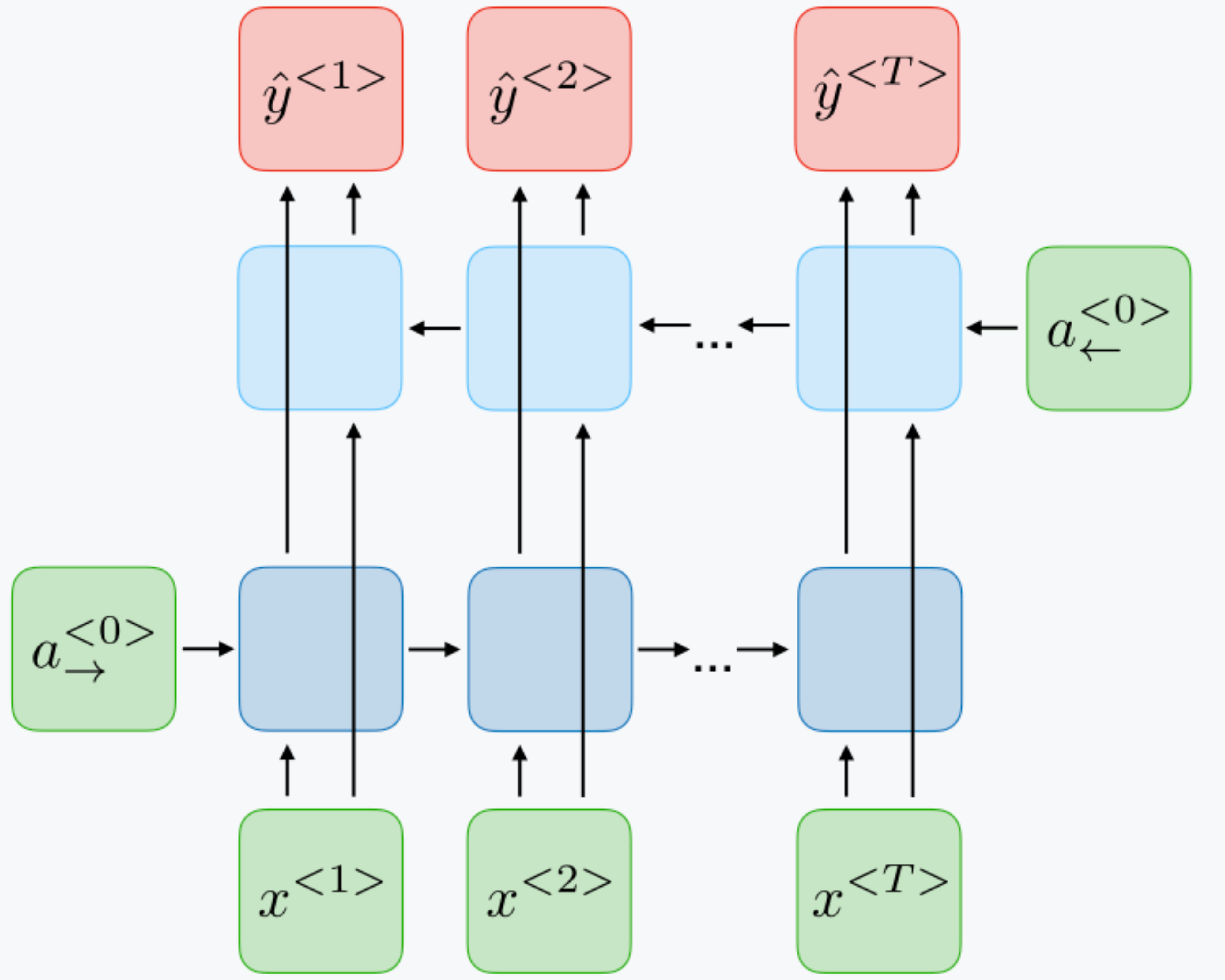
Source: RNN cheat sheet

Two Extensions

- Deep RNNs:



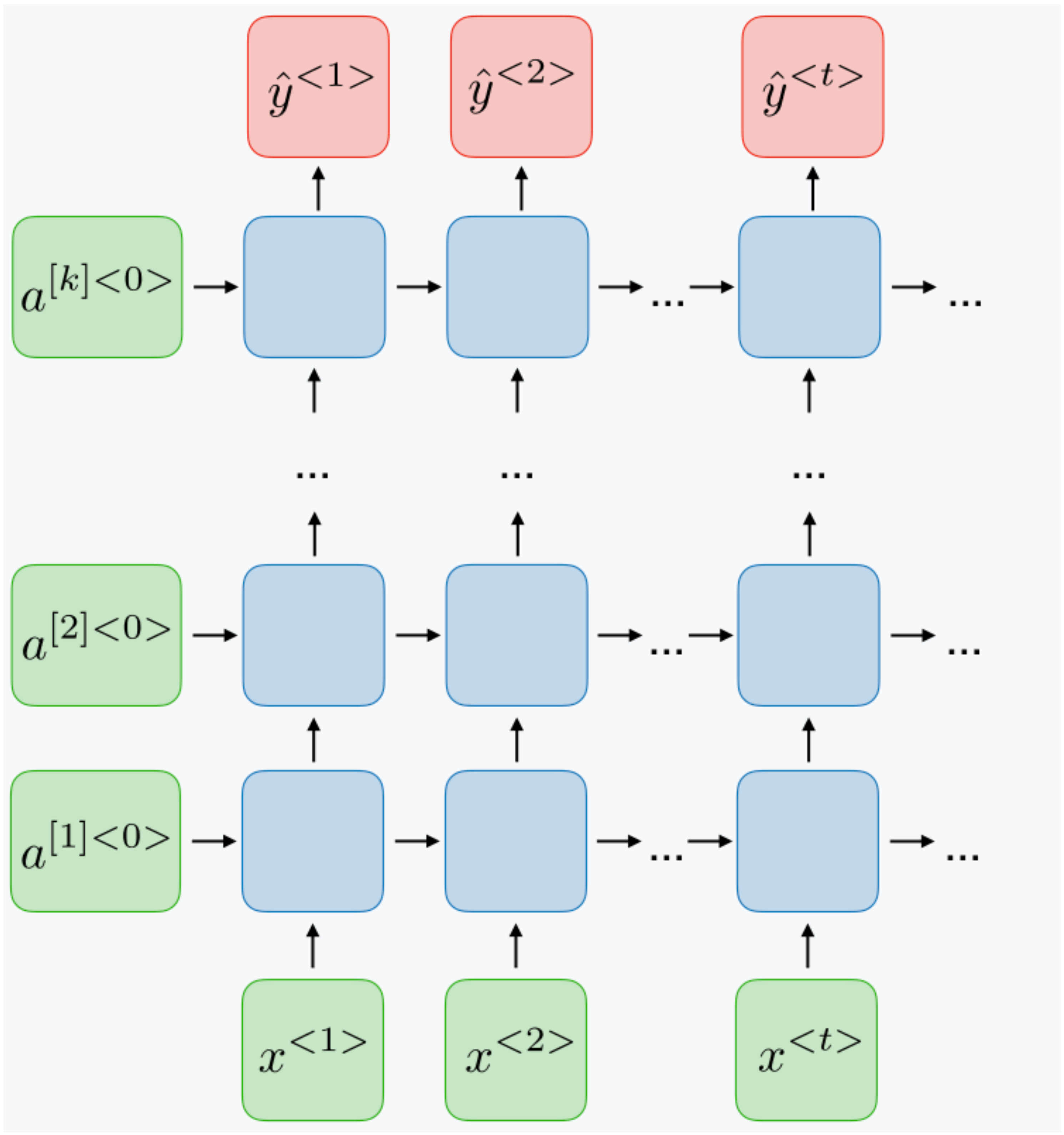
- Bidirectional RNNs:



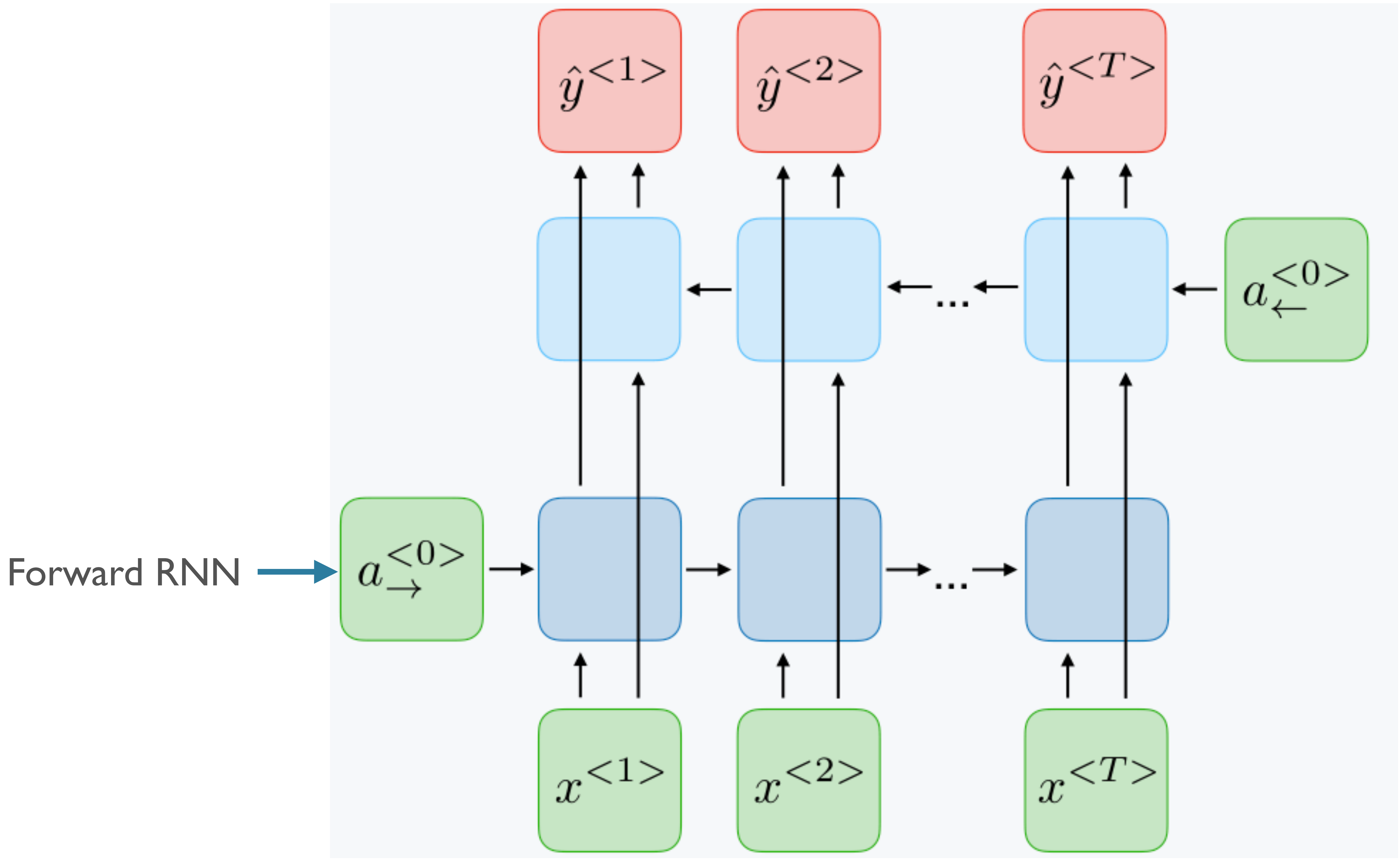
Source: RNN cheat sheet

Two Extensions

- Deep RNNs:



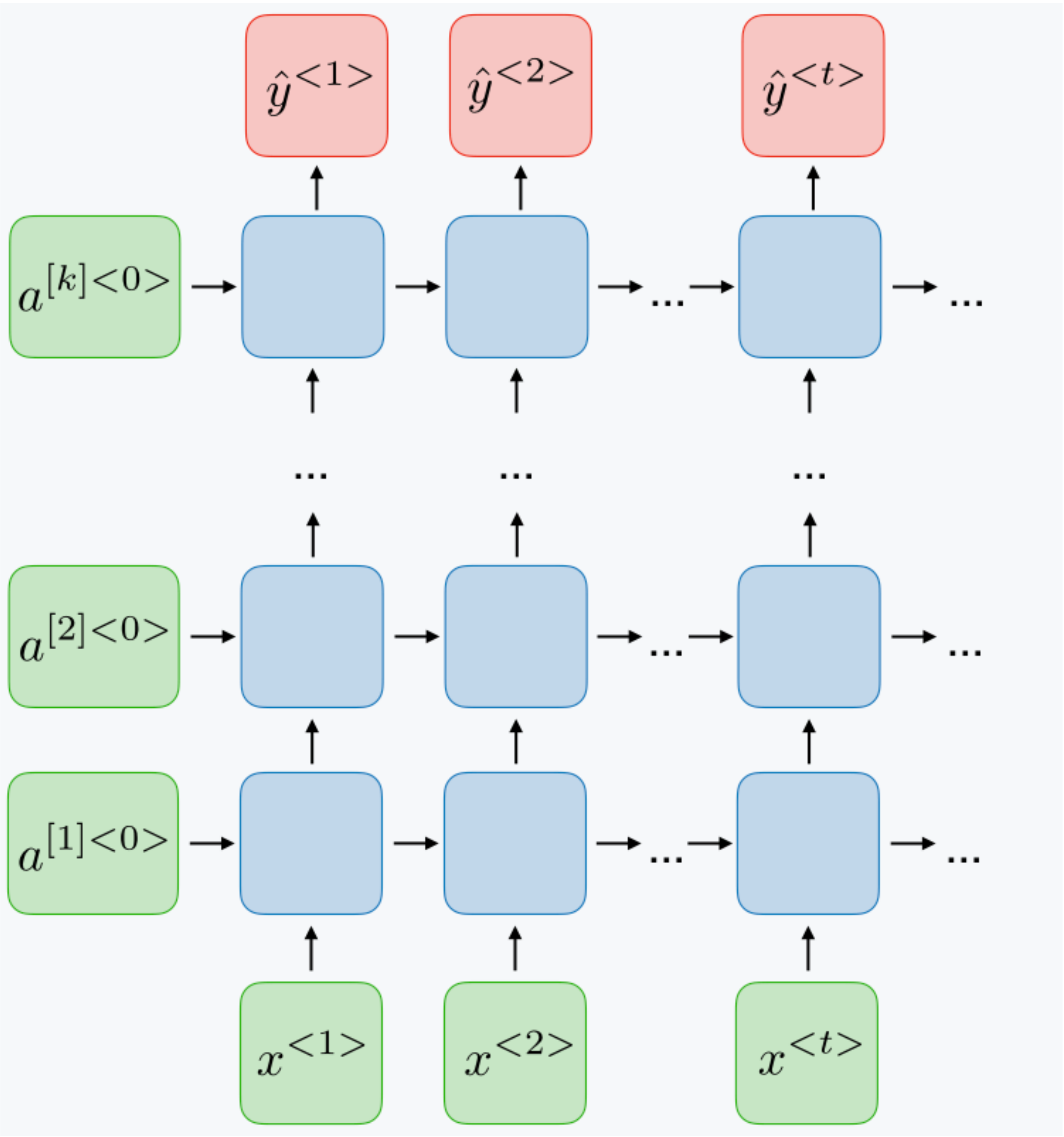
- Bidirectional RNNs:



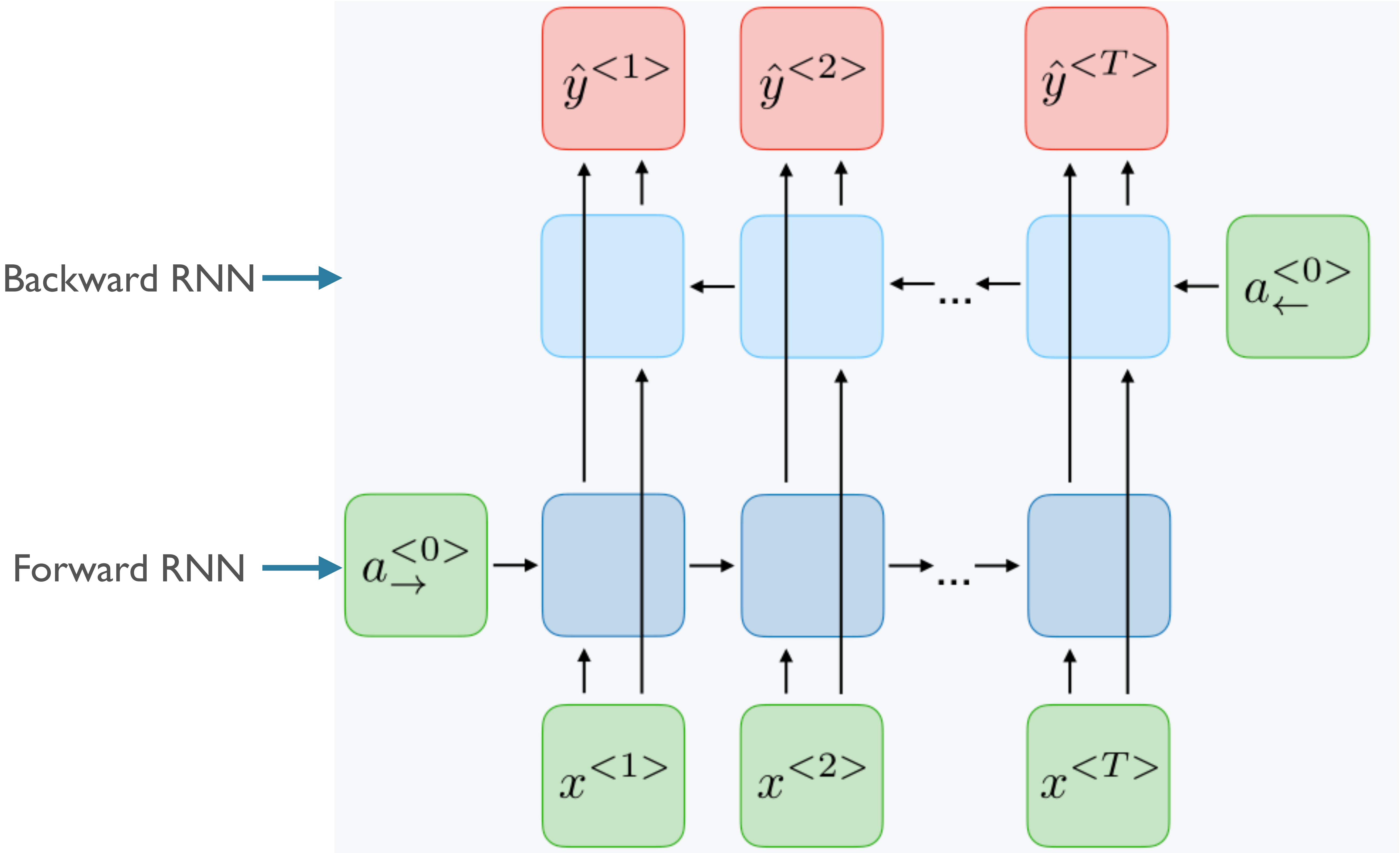
Source: RNN cheat sheet

Two Extensions

- Deep RNNs:



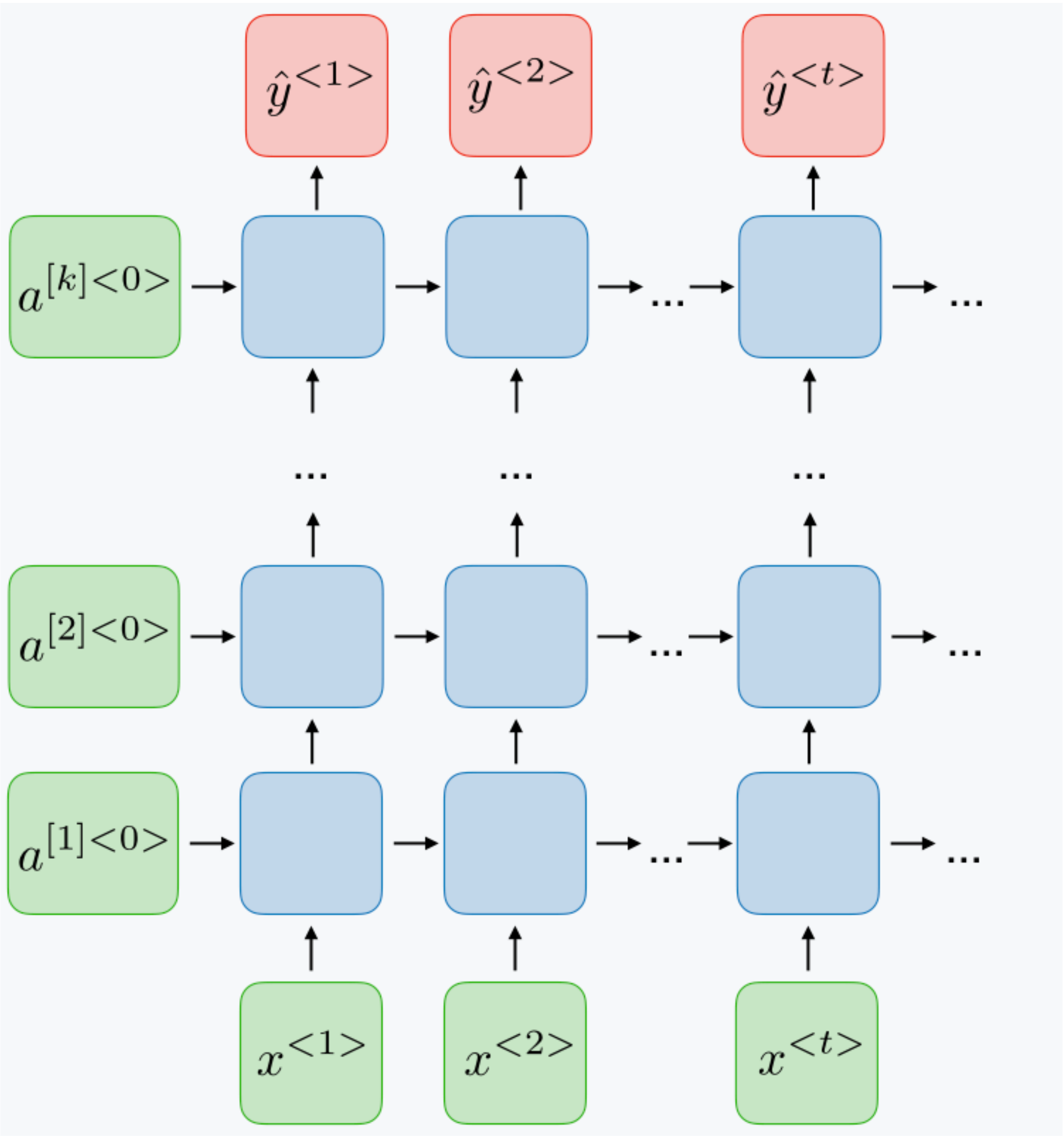
- Bidirectional RNNs:



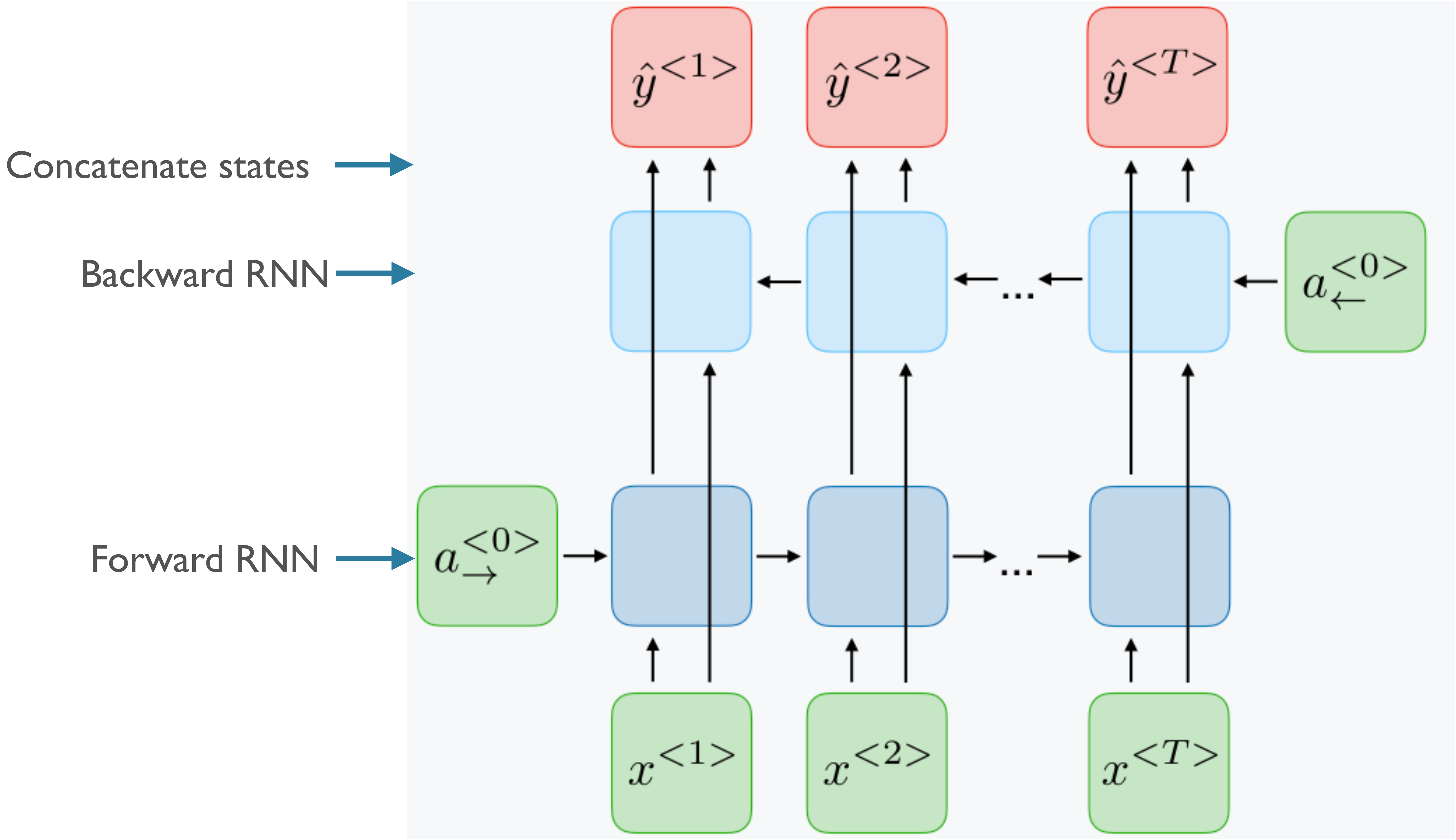
Source: RNN cheat sheet

Two Extensions

- Deep RNNs:



- Bidirectional RNNs:



Source: RNN cheat sheet

Batching in RNNs

- Intuitively, shape of inputs: [batch_size, seq_len, vocab_size]
- But what is sequence length??
  - “This is the first example </s>”: 6
  - “This is another </s>”: 4

Padding and Masking

- Step 1: **pad** all sequences in batch to be of the **same length**
  - "This is the first example </s>": 6
  - "This is another </s> PAD PAD": 6
- Step 2: build a **"mask"** (1 = True token, 0 = padding)

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

- Step 3: use mask to tell model **what to ignore**, either
  - Select correct final states (classification)
  - Multiply losses in tagging tasks (LM)

Summary

- RNNs allow for neural processing of **sequential data**
- In principle, should help models capture **long-distance dependencies** (e.g. number agreement, selectional preferences, …)
  - Maintain a state over time
  - **Repeatedly apply the same weights**
  - as opposed to n-gram models, which cannot build such dependencies
- Uses: classification, tagging
- Extensions: deep, bidirectional

Next Time

- Discuss a technical problem in training Vanilla RNNs
  - Vanishing gradients
- Introduce *gating-based* RNNs
  - LSTMs
  - GRUs
  - Strengths, weaknesses, differences