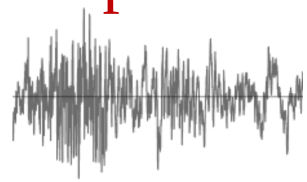
Sequence Models

Recurrent Neural Networks

CS8004: Deep Learning and Applications

Slides based on Andrew Ng’s Coursera lecture course: Sequence models

Sequence Data

	Input	Output
Speech Signal processing		Wow, it is so nice!
Name Entity Recognition	GH Hardy said, his main contribution was discovery of Ramanujan.	GH Hardy said, his main contribution was discovery of Ramanujan.
Machine Translation	Wow, it is so nice!	वाह, यह कितना अच्छा है !
Activity recognition		→ Dancing

Sequence Data

	Input	Output
Sentiment classification	Wow, it is so nice!	
Music generation	∅	
DNA sequence analysis	A5ASC3.1 B4F917.1 A9S1V2.1 B9GSN7.1 Q8H056.1 Q0D4Z3.2	SIKLWPPSQTTRLLLVERMANNLST..PSIFTRK. SIKLWPPSESTRIMLVDRMTNNLST..ESIFSRK. VFKLWPPSQGTREAVRQKMALKLSS..ACFESQS. SVKLWPPGQSTRLMLVERMTKNFIT..PSFISRK. SFSIWPPTQRTRDAVVRRLVDTLGG..DTILCKR. SLSIWPPSQRTRDAVVRRLVQTLVA..PSILSQR.

Andrew Ng Coursera lecture Slides

Example: Name Entity Recognition

- Example Input Sequence: *GH Hardy said his main contribution was discovery of Ramanujan.*
- Each word \ comma \ full stop is considered as an input. There are a total of 11 inputs in the sentence.
    - Therefore input sequence length = 11.
    - The corresponding word labels are 1 or 0 if the word is a name or not a name, respectively.
    - For example : discovery is labeled as 0 and Ramanujan is labeled as 1.
- Input : The entire sentence with labels.
- Output: The entire sentence with probability for each word to be a name entity.
- Note that in this example, the input and output lengths are the same.

Andrew Ng Coursera lecture Slides

Sequence Data Input and Outputs

- There can be sequence data with different input and output lengths.

	Input length Tx	Output length Ty
Sentiment classification	Wow, it is so nice! Tx= 7	★★★★☆ Ty= 1 (0 to 5 stars)
Music generation	∅ Tx= 1	 Ty= (more than 1)2
Machine Translation	Today it is raining. Tx= 5	आज वर्षा हो रही है । Ty = 6

Limitations of General ANN

- Inputs, outputs can be of different lengths in different examples. This cannot be handled in conventional ANNs.
- Words learned or approximated at a later position may change the approximation of a previous word.
  - Example :
    - Red roses are sold at higher prices.
    - Red rose is sold at a higher price.
  - Here the word 'roses' or 'rose' may be revisited for better approximation accuracy later, depending on the next word 'are' or 'is'.
- Parameter sharing is not done in conventional ANNs.

Different Input\Output Length Requirement

- One to one
- One to many
- Many to one
- Many to many (same number)
- Many to many (possibly different output number)

Recurrent Neural Networks

- A recurrent neural network (RNN) is a type of advanced ANN that involves directed cycles in memory for sharing parameters across different parts of the network.

- It can accept inputs and outputs of varying lengths.

- It is designed to recognize sequential characteristics of data and uses patterns to predict the next likely scenario.

When to Use RNN ?

- “Whenever there is a sequence of data and the temporal dynamics that connects the data is more important than the spatial content of each individual frame.”

– Lex Fridman (MIT)

History

- Recurrent neural networks are based on David Rumelhart's work in 1986.
- Hopfield networks introduced in 1982. A **Hopfield network** is a form of recurrent neural **network** popularized by John **Hopfield** in **1982**, but was described earlier by Little in 1974.
- In 1993, a neural history compressor system solved a "Very Deep Learning" task that required more than 1000 subsequent layers in an RNN unfolded in time.

History…

- Long short-term memory (LSTM) by Hochreiter and Schmidhuber in 1997.
- LSTM made a revolution by its excellent performance in speech recognition.
- LSTM also improved text-to-speech synthesis and is used in Google Android.
- LSTM broke records for improved machine translation, Language Modeling, and Multilingual Language Processing.
- LSTM combined with convolutional neural networks (CNNs) improved automatic image captioning.

- Google assistant and Apple Siri use RNNs.

Recurrent Neural Networks

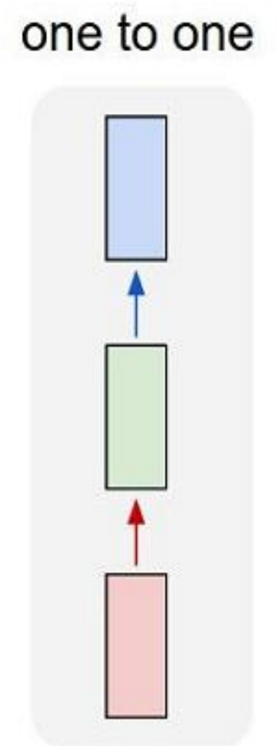
- Example:
- Input sequence is –

  A magnitude 7.8 earthquake struck Nepal in 2015.
- Or

  In 2015, A magnitude 7.8 earthquake struck Nepal.

- In both the sentences the year ‘2015’ and ‘Nepal’ are crucial for information extraction on ‘earthquake’.

Recurrent Neural Network Architectures

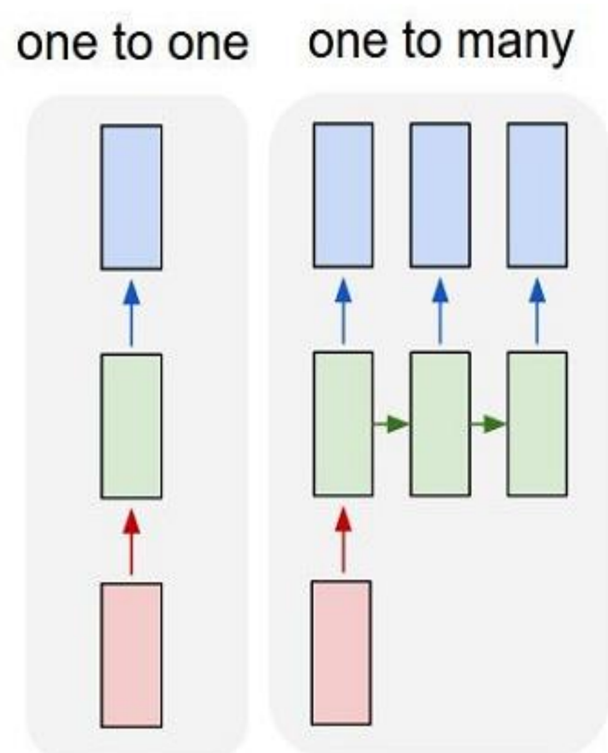


Vanilla NN

Recurrent Neural Networks accommodate input and output sequences of different lengths.

Source: cs231n Stanford University course lecture slides

Recurrent Neural Network Architectures…

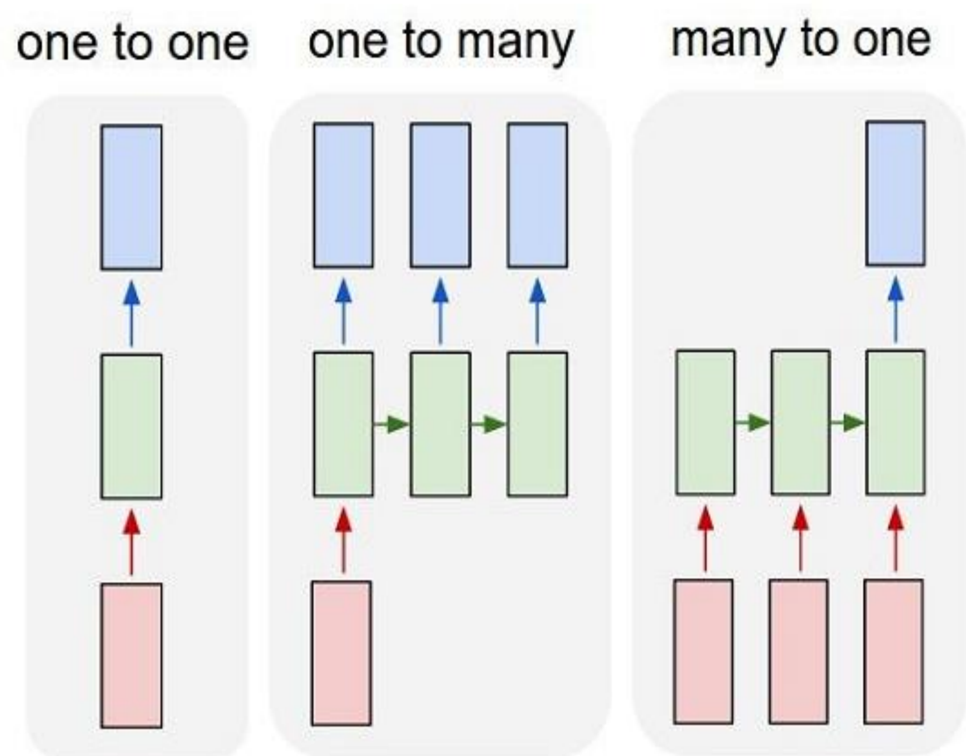


Vanilla NN

Image caption

Source: cs231n Stanford University course lecture slides

Recurrent Neural Network Architectures…



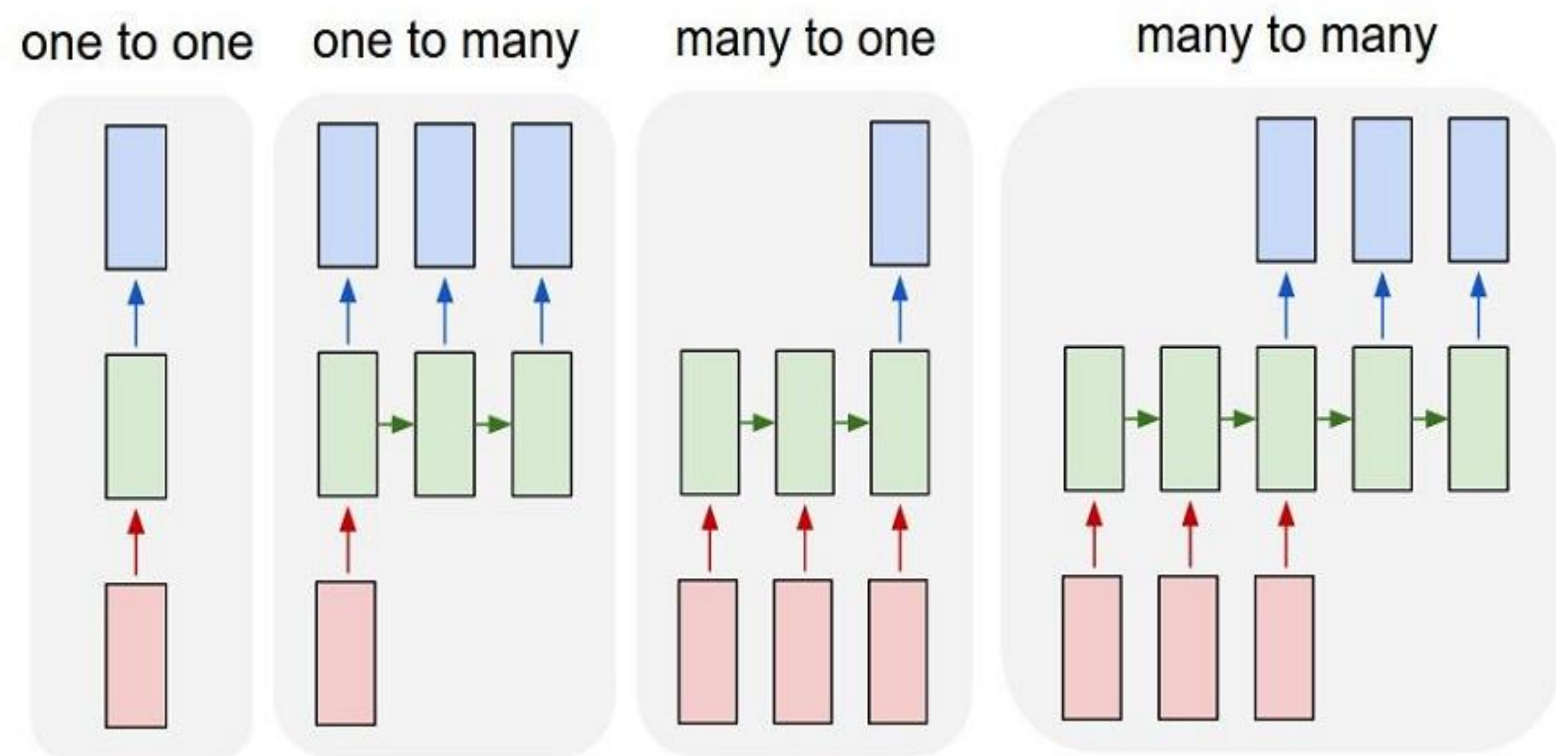
Vanilla NN

Image caption

Sentiment classification

Source: cs231n Stanford University course lecture slides

Recurrent Neural Network Architectures…



Vanilla NN

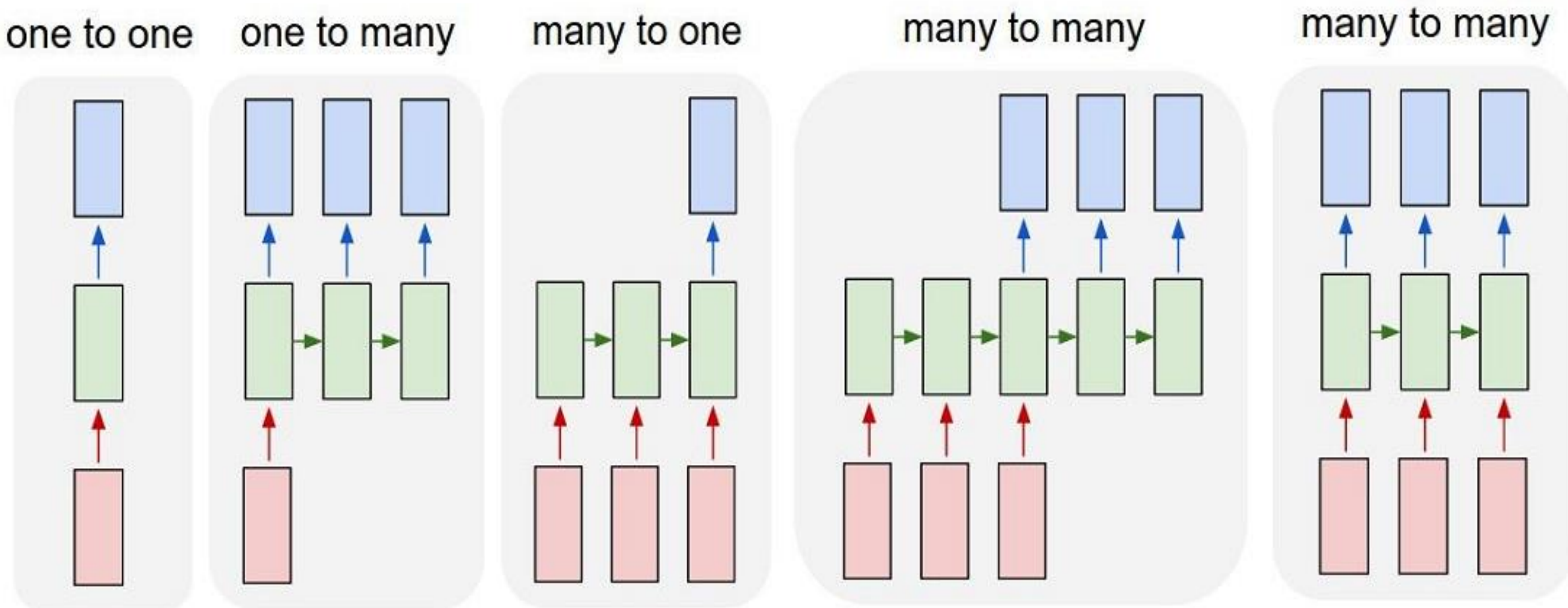
Image caption

Sentiment classification

Machine translation

Source: cs231n Stanford University course lecture slides

Recurrent Neural Network Architectures…



Vanilla NN | Image caption | Sentiment classification | Machine translation | Name Entity Recognition

Source: cs231n Stanford University course lecture slides

Example from NLP

- Vocabulary and word representation:
- Suppose the vocabulary consists of 20,000 words.
- Each of the words in the dictionary and puncutations are represented by a 1-hot vector.
- In the following input sentence x –

  "Hardy said, his main contribution was discovery of Ramanujan."
- Each word is replaced by its corresponding 1-hot vector.
- Every word in the input sequence is assigned a label in the dictionary, from 1 to 20, 000.
- Any word not in the dictionary is assigned a value <UNK>

Example from NLP…

x = {Hardy said, his main contribution was discovery of Ramanujan.}

$$x^1, \quad x^2, \quad x^3, \quad x^4 \quad \cdots \quad x^{11}$$

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ . \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \cdot \\ 1 \\ . \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ . \\ 1 \\ 0 \\ 0 \\ \cdot \\ . \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ . \\ 0 \\ 0 \\ 1 \\ \cdot \\ . \\ 0 \end{bmatrix}$$

Example from NLP...

x = {Hardy said, his contribution was discovery of Ramanujan.}

$$x^1, \quad x^2, \quad x^3, \quad x^4 \quad \quad x^{11}$$

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ . \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ . \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ . \\ 1 \\ 0 \\ 0 \\ \vdots \\ . \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ . \\ 0 \\ 0 \\ 1 \\ \vdots \\ . \\ 0 \end{bmatrix}$$

If the problem is of name entity recognition, input and output sequence will be of the same size : $T_x = T_y$.

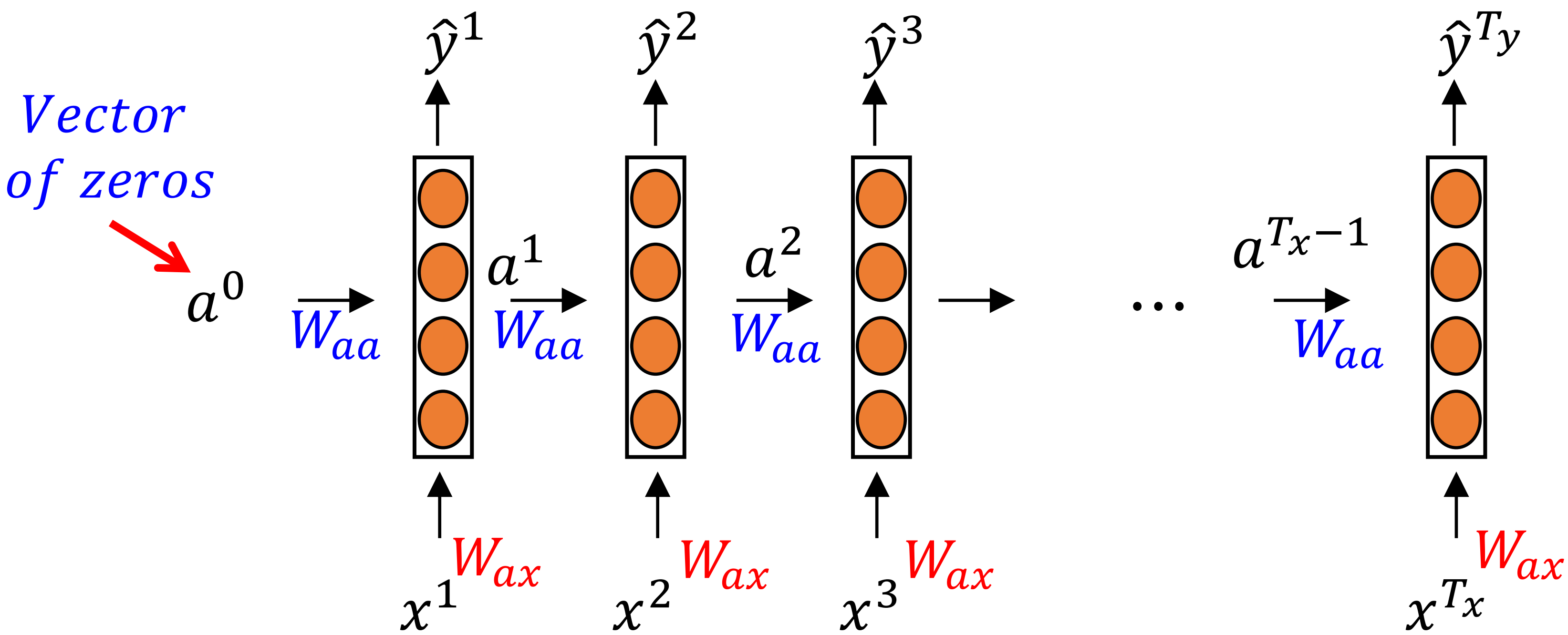
Recurrent Neural Network

- A sequence of vectors is processed by applying a recurrence formula at each time step.
- By time step we don't mean the actual time step, but the order in which each unit of the sample is fed \processed.
- The same function and same set of parameters are used in each time step.
    - $a^t = f_W(a^{t-1}, x^t)$
- Example
    - $a^t = tanh(W_{aa}a^{t-1} + W_{ax}x^t)$

Forward Propagation

- $a^t = g_1(W_{aa}a^{t-1} + W_{ax}x^t + b_a)$ $g_1 = relu$ / tanh

- $\hat{y}^t = g_2(W_{ay}a^t + b_y)$ $g_2 = softmax\ /sigmoid$



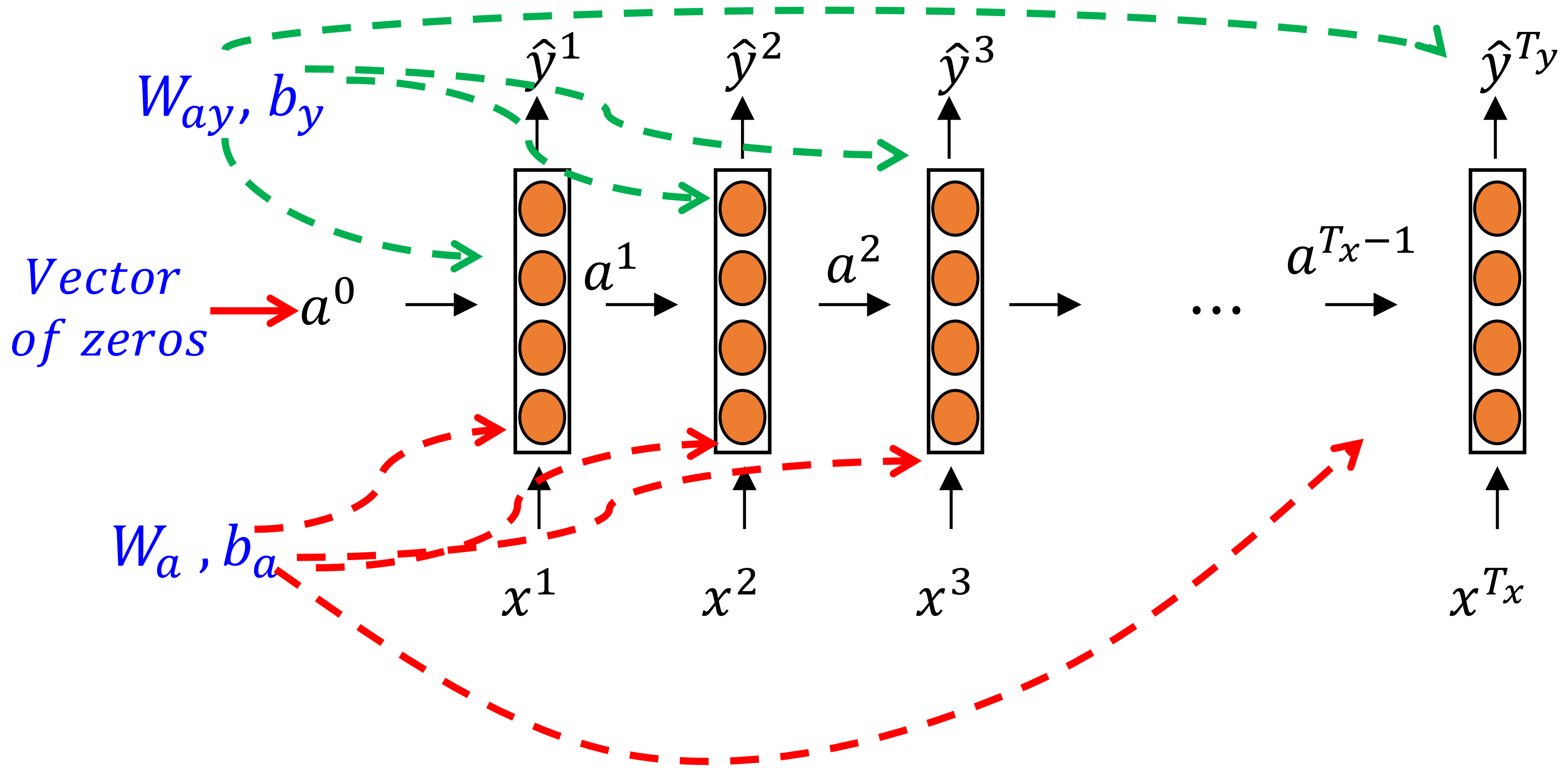
Forward Propagation

- $a^t = g_1(W_{aa}a^{t-1} + W_{ax}x^t + b_a)$ $g_1 = relu$ / tanh
- $\hat{y}^t = g_2(W_{ay}a^t + b_y)$ $g_2 = softmax\ or\ sigmoid$
- Simplified notation :
  - $W_a = [\ W_{aa} : W_{ax}\ ]$ concatenate the two parameter matrices.
  - Also concatenate a^{t-1} and x^t

    $$[a^{t-1}, x^t] = \begin{bmatrix} a^{t-1} \\ x^t \end{bmatrix}$$
  - $W_{aa}a^{t-1} + W_{ax}x^t = W_a\ [a^{t-1}, x^t]$

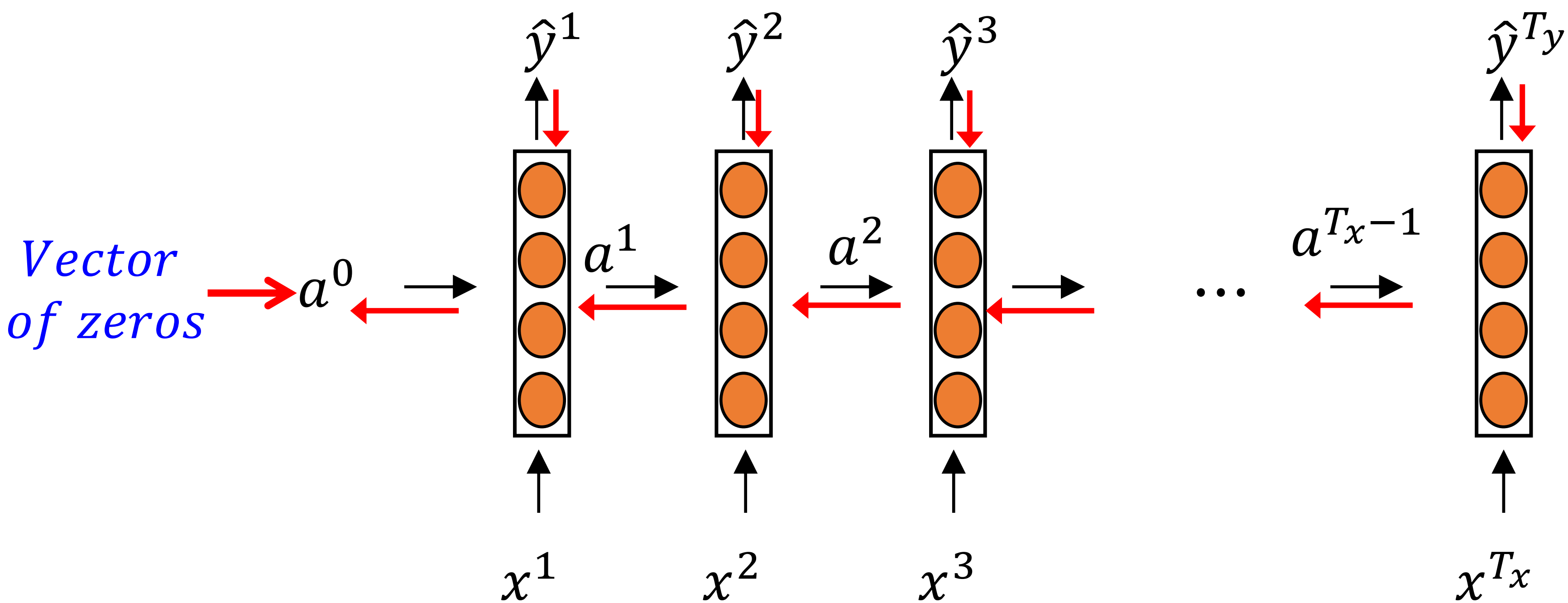
Forward Propagation

- Parameter sharing throughout the input sequence.

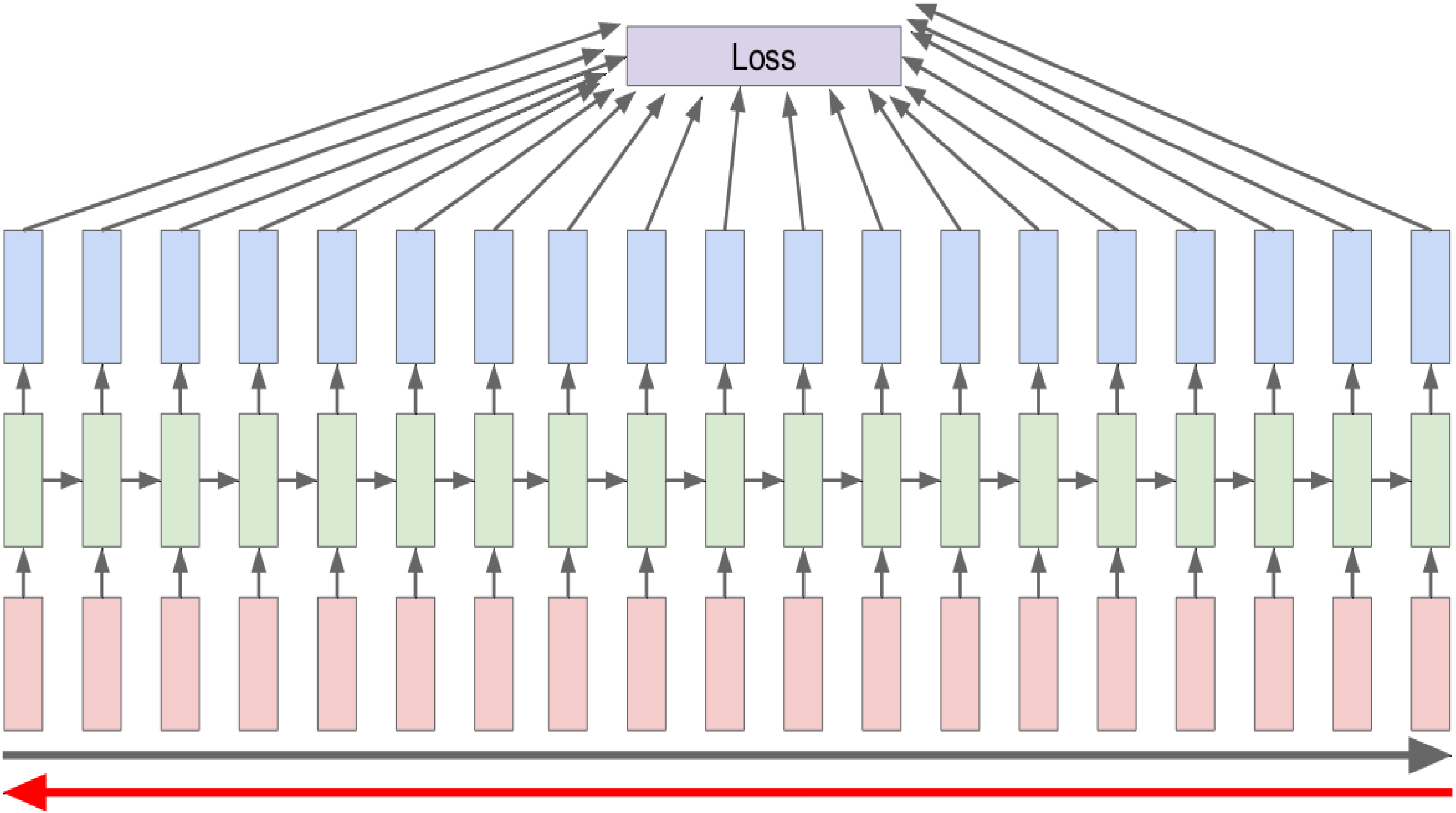


Forward and Backward Propagation

- Forward propagation to compute loss.
- Backward propagation to compute gradient.

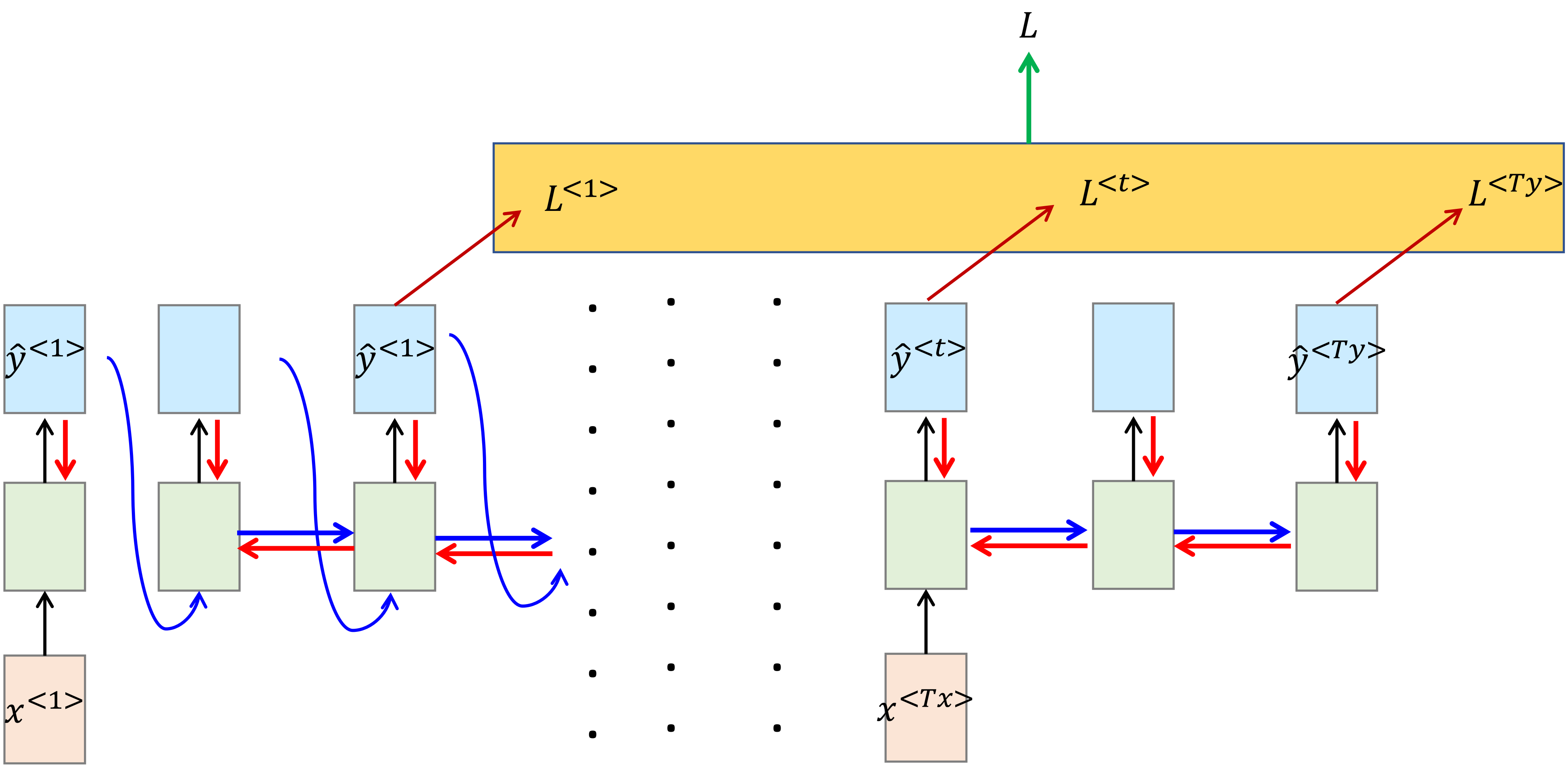


Forward and Backward Propagation : Loss function

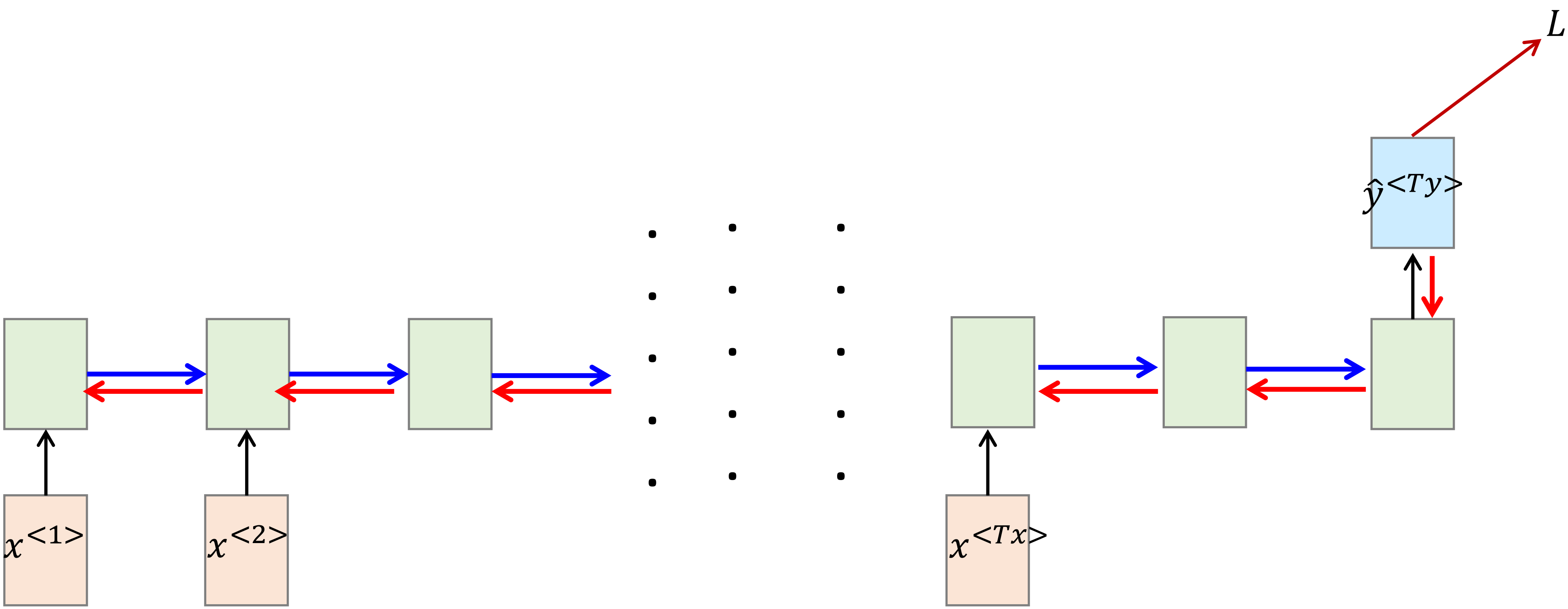


Source: cs231n Stanford University course lecture slides

Many to Many RNN : Loss Function



Many to One RNN : Loss Function

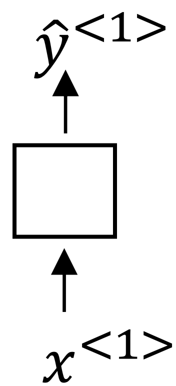


Loss Function

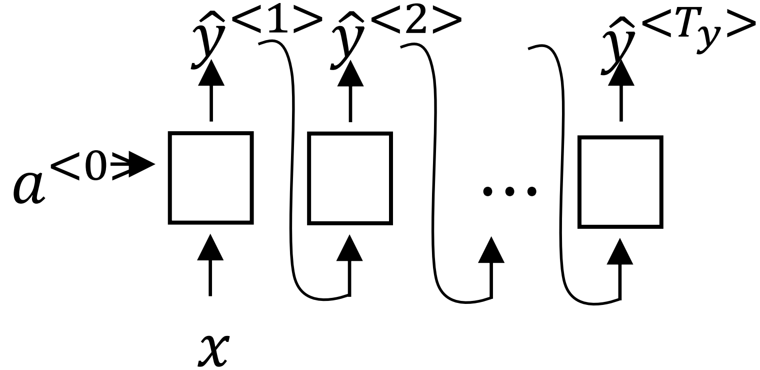
- $\mathcal{L}^t(\hat{y}^t, y^t) = \sum_{t=1}^{Ty} L^t(\hat{y}^t, y^t)$

- $L^t(\hat{y}^t, y^t) = -y^t \log \hat{y}^t \; - (1 - y^t)\log(1 - \hat{y}^t)$

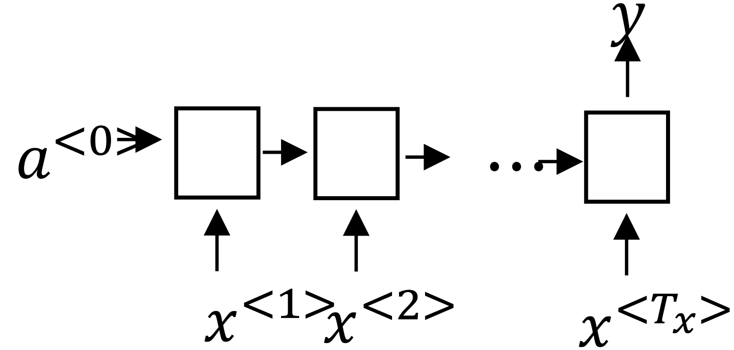
Summary of RNN types



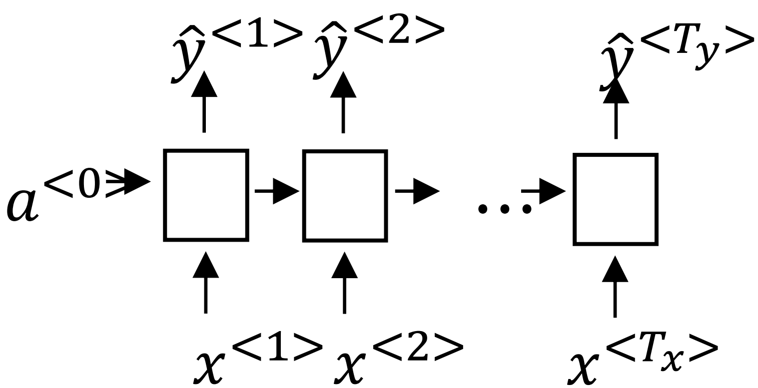
One to one



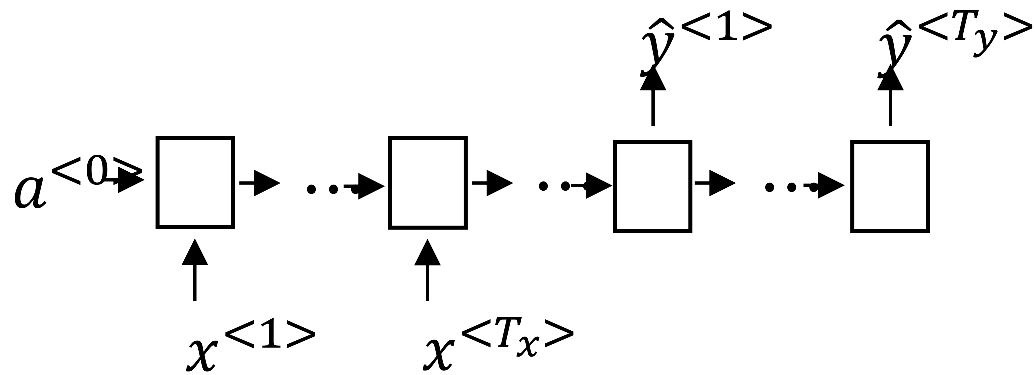
One to many



Many to one



Many to many

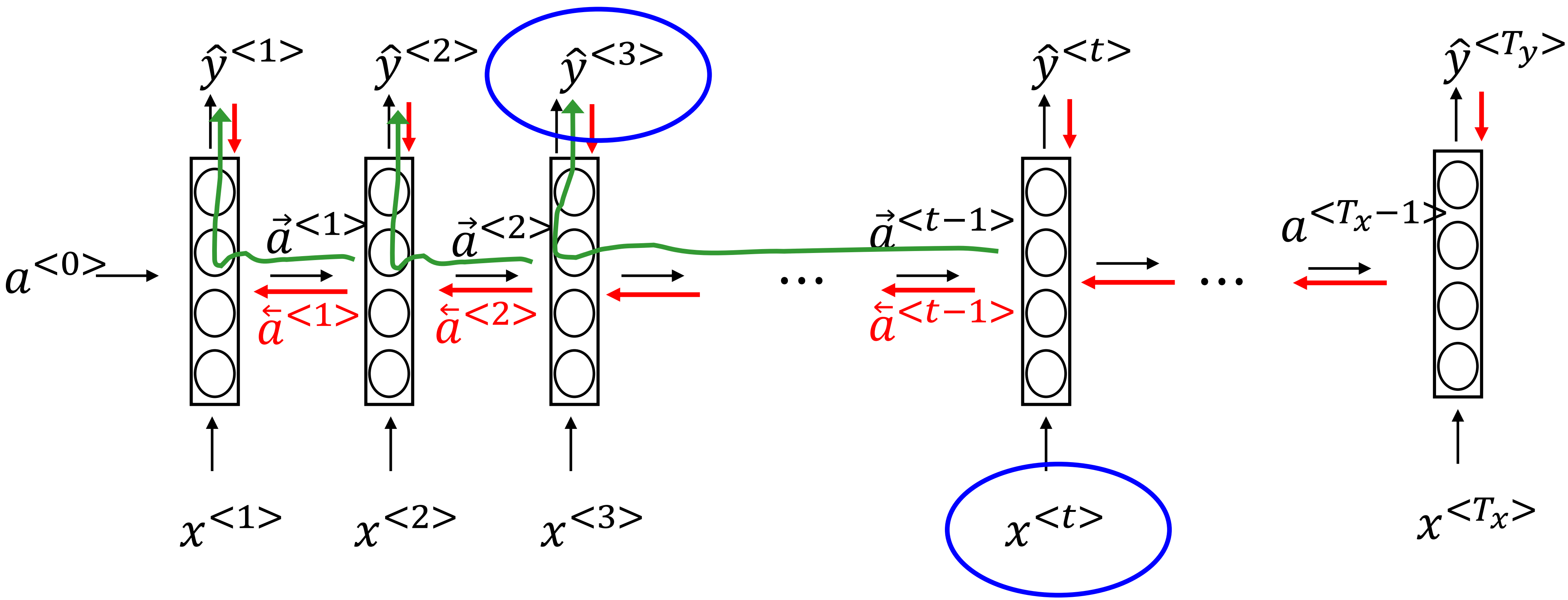


Many to many

Andrew Ng Coursera lecture Slides

Bidirectional RNN

- Some RNN send back the processed variables to improve the output.
- Example: He said, Teddy bears are on sale!
- He said, Teddy Roosvelt was an American president.



$$\hat{y}^{<t>} = g(W_y\ [\vec{a}^{<1>}, \overleftarrow{a}^{<2>}]\ ,\ b_y\ )$$

Andrew Ng Coursera lecture Slides

What is language modelling?

Speech recognition

I enjoyed my trip to Andhra Pradesh.
I also learned to speak some sentences from Telugu Language.

I enjoyed my trip to Andhra Pradesh.
I also learned to speak some sentences from Tamil Language.

$$Probabiltiy\ of\ \text{(I also …from Telugu Language} = 2.3 \times 10^{-12}$$

$$Probabiltiy\ of\ (\text{I also ...from Tamil Language.}) = 8.1 \times 10^{-14}$$

Language modelling with an RNN

Training set: large corpus of English text.
Each word is tokenized.
<EOS> and other punctuations are also tokenized in some models.

Language modelling with an RNN

Training set: large corpus of english text.
Each word is tokenized.
<EOS> and other punctuations are also tokenized in some models
An unknown word is labeled as <UNK>

Language modelling with an RNN

Training set: large corpus of english text.
Each word is tokenized.
<EOS> and other punctuations are also tokenized in some models
An unknown word is labeled as <UNK>

Then the model is built with $x^{<0>} = 0, x^{<t>} = y^{<t-1>}$

Language modelling with an RNN

Training set: large corpus of english text.
Each word is tokenized.
<EOS> and other punctuations are also tokenized in some models
An unknown word is labeled as <UNK>

Then the model is built with $x^{<0>} = 0, x^{<t>} = y^{<t-1>}$

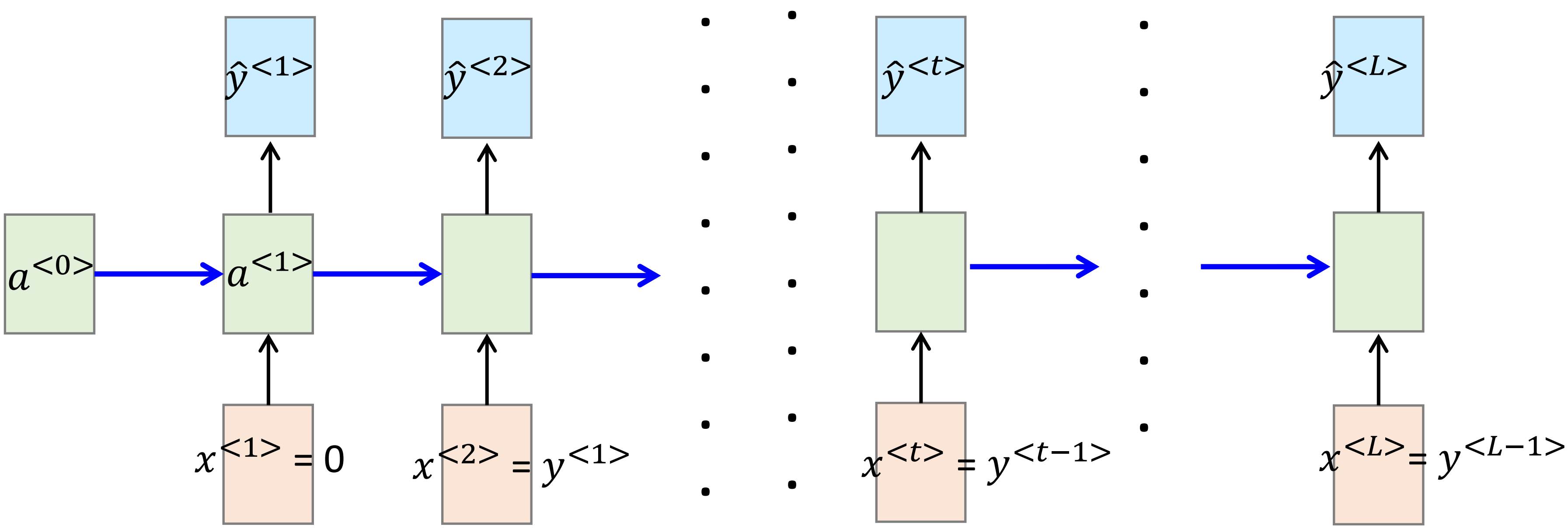
Example: Dogs sleep more in winter.

Word tokenization: [‘Dogs’, ‘sleep’, ‘more’, ‘in’, ‘winter’, ‘.’]

Language modelling with an RNN

Dogs sleep more in winter.

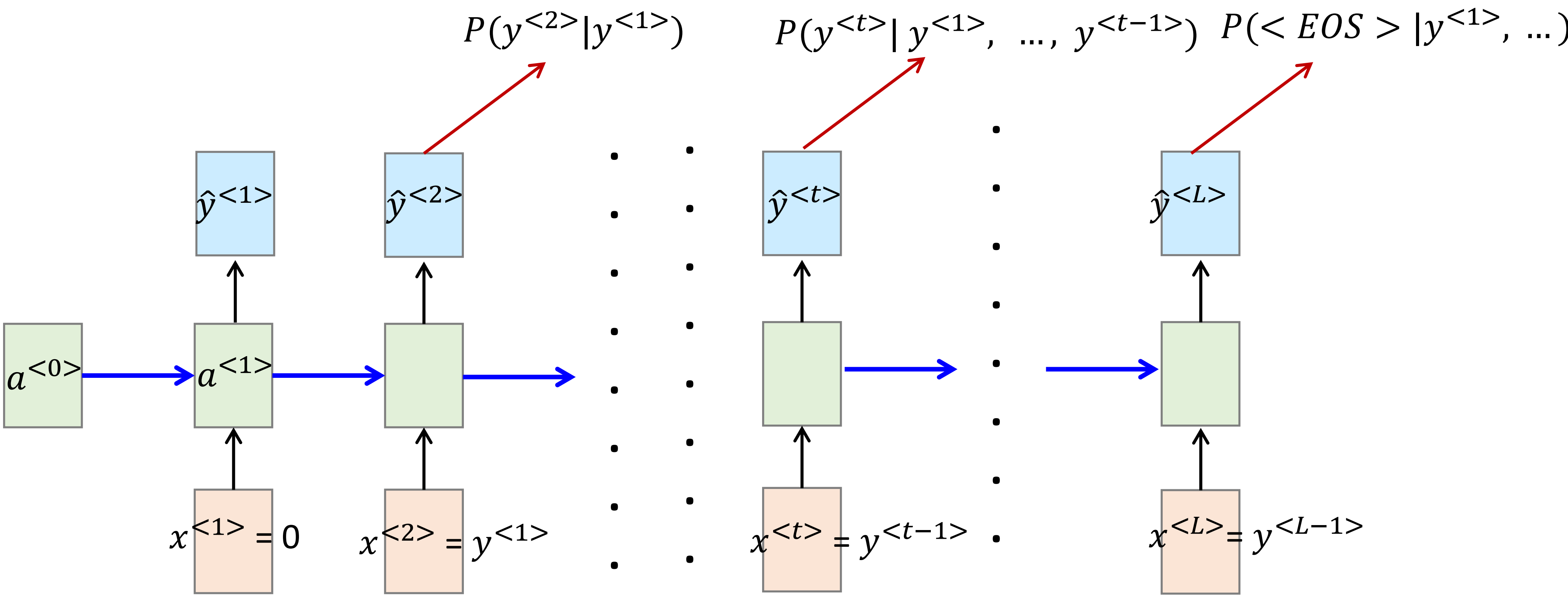
$y^{<1>}, y^{<2>}, y^{<3>}, y^{<4>}, y^{<5>}$, <EOS>



Language modelling with an RNN

Dogs sleep more in winter.

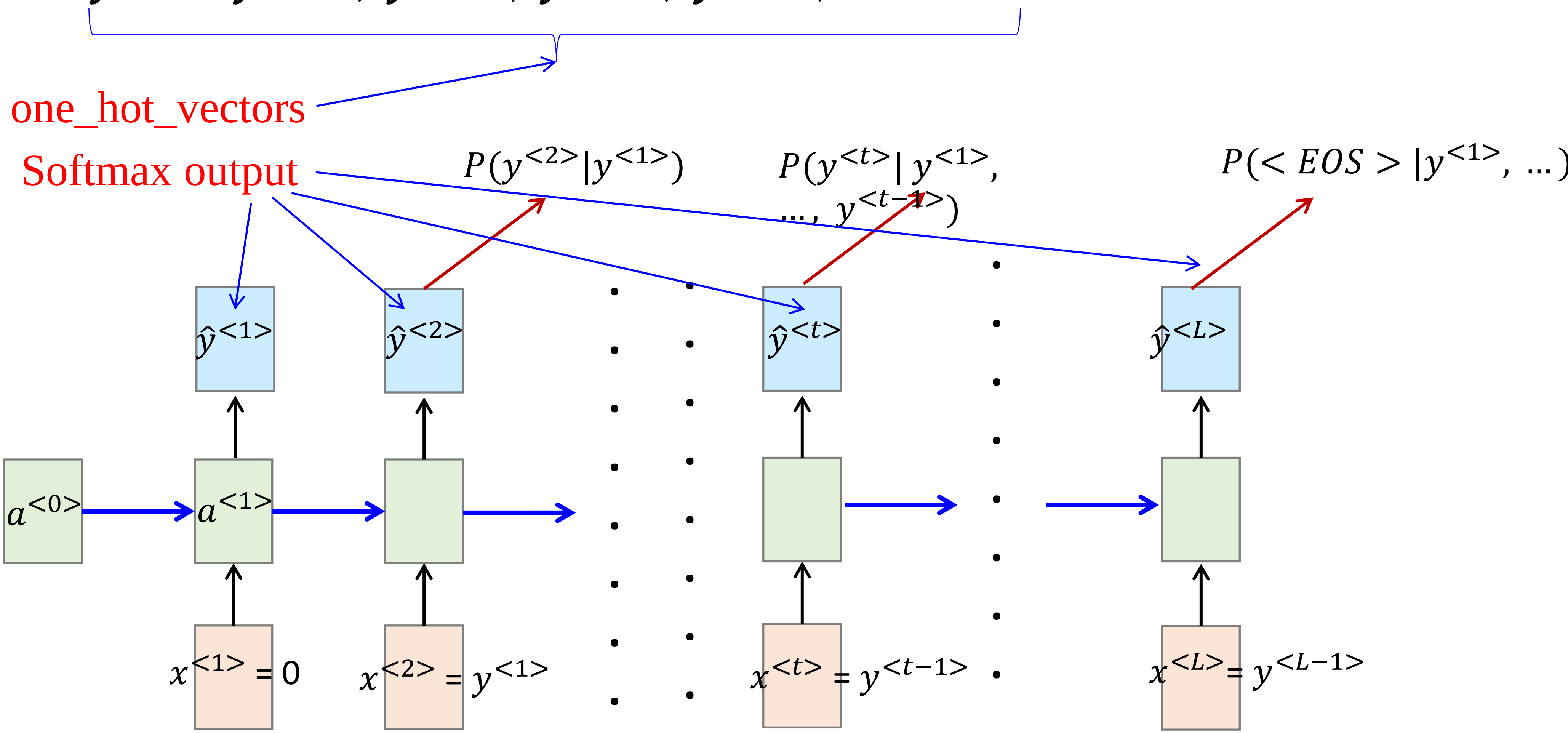
$y^{<1>}, y^{<2>}, y^{<3>}, y^{<4>}, y^{<5>}$, <EOS>



Language modelling with an RNN

Dogs sleep more in winter.

$y^{<1>}, y^{<2>}, y^{<3>}, y^{<4>}, y^{<5>}$, <EOS>



Language modelling with RNN

Loss function

$$\mathcal{L}(\hat{y}^{<t>}, y^{<t>}) = -\sum_i y_i^{<t>} \log \hat{y}_i^{<t>}$$

$$\mathcal{L} = \sum_t \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

Summary

- Sequence models are variants of NN that are used when the input data is sequential in nature or when the input data may be of different length in the sequence.
- Need to share features learned from the previous instances.
- Types of RNN: one-to-one, many-to-one, many-to-many (same size), many-to-many (different size of input and output sequence), one-to-many.
- Bidirectional RNNs are used for example, when the words coming later in a sequence may have a different interpretation of a previous word.