5-1 Introduction

Zhonglei Wang

WISE and SOE, XMU, 2025

Contents

Motivation

1. It is a common task for natural language processing (NLP)
    - Machine translation
    - Sentiment analysis
    - ChatGPT (Chat Generative Pre-trained Transformer)
2. For example, find the destination in the following sentences
    - I arrive at Beijing from Xiamen
    - I leave Beijing to Xiamen
3. Problems
    - How to convert sentences into features for calculation?
    - What are appropriate models for NLP?

Motivation

1. We usually use the following steps for NLP
   - Tokenization
   - Embedding

     (For feature extraction)
   - Modeling (For analysis)

Tokenization

1. Difficulties for Tokenization
   - Some words, including names or rarely used ones, may not be in the vocabulary
   - It is not clear how to handle punctuations, including “,.;:?!”
   - Different tokens may be needed for the same word with different suffixes
     - ▷ For example, walk, walks, walked...
2. Possible solution: a vocabulary includes
   - Commonly used words
   - Word fragments from which larger or less frequent words can be formed

Tokenization

1. For English, tokenization is straightforward
2. For example, consider "I arrive at Beijing from Xiamen."
3. We can tokenize the above sentence into the following seven parts

   I / arrive / at / Beijing / from / Xiamen / .

4. Then, a vector is assigned to each token according to a vocabulary

One-hot encoding

1. In practice, we have a vocabulary of size $N(\approx 30,000)$
2. One-hot encoding can be used to represent each token in this vocabulary
   - A vector of length N
   - Contains 0 but only a single 1, indicating the position of that token in the vocabulary

Remarks on one-hot encoding

1. Disadvantages
   - High dimensionality and sparsity: most information is redundant, especially when the vocabulary is large
   - Low generality to new words
   - Overfitting due to the large dimension of the one-hot encoding
   - Cannot model relationship between (among) words, such as “Xiamen University”
   - Sensitive to small changes in the vocabulary: adding or deleting a word may change all encoding system
   - ……

Embedding

1. We want a new token representation (Embedding)
   - Achieving low dimensionality ($\approx 1,024$)
   - Reflecting relationship between (among) words.
2. Possible solutions
   - Word2Vec: CBOW+Skip-gram
   - GloVe: generalizes Word2Vec
   - N-Gram: a probability model
   - TF-IDF
   - BERT
   - ......

Embedding

1. Embedding is a fundamental task for NLP
   - Embeddings are learned from data using various algorithms
   - For example, with embeddings, OpenAI's GPT models can generate more coherent and contextually relevant responses to user prompts and questions.
2. Nevertheless, we may not discuss those techniques, and check by yourselves
3. In the following analysis, we assume that embedding is done for each word

Models

1. Find the destination of the following sentence

	I	arrive	at	Beijing	from	Xiamen
Embedding:	$\boldsymbol{x}^{<1>}$	$\boldsymbol{x}^{<2>}$	$\boldsymbol{x}^{<3>}$	$\boldsymbol{x}^{<4>}$	$\boldsymbol{x}^{<5>}$	$\boldsymbol{x}^{<6>}$
Label:	$y^{<1>} = 0$	$y^{<2>} = 0$	$y^{<3>} = 0$	$y^{<4>} = 1$	$y^{<5>} = 0$	$y^{<6>} = 0$

2. What are appropriate models for NLP?

Models

1. Why not using FNN or CNN?
   - The dimension of embedding may be extremely large
   - Inputs may have different length
   - Specifically, standard NN cannot deal with dependence

Recurrent Neural Network

1. A similar concept was proposed in 1980s
2. We focus on finding the destination in a sentence
   - Essentially, this is a binary classification task for each word in a sentence
3. We have a training dataset, consisting of different labeled sentences
4. Different tasks are discussed later

Recurrent Neural Network

1. Suppose we have a sentence

	I	arrive	at	Beijing	from	Xiamen
Embedding:	$\boldsymbol{x}^{<1>}$	$\boldsymbol{x}^{<2>}$	$\boldsymbol{x}^{<3>}$	$\boldsymbol{x}^{<4>}$	$\boldsymbol{x}^{<5>}$	$\boldsymbol{x}^{<6>}$
Label:	$y^{<1>} = 0$	$y^{<2>} = 0$	$y^{<3>} = 0$	$y^{<4>} = 1$	$y^{<5>} = 0$	$y^{<6>} = 0$

2. We are interested in estimating probabilities $\{\hat{y}^{<i>} : i = 1, \ldots, 6\}$ for each word in this sentence

Building block

1. Initialize $\boldsymbol{a}^{<0>} = 0$
2. Given model parameters $\{\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a, \boldsymbol{W}_{ya}, b_y\}$, obtain

$$\boldsymbol{a}^{<i>} = \sigma_{ax}(\boldsymbol{W}_{ax}\boldsymbol{x}^{<i>} + \boldsymbol{W}_{aa}\boldsymbol{a}^{<i-1>} + \boldsymbol{b}_a) \quad (i = 1, \ldots, 6)$$

$$\hat{y}^{<i>} = \sigma_{ya}(\boldsymbol{W}_{ya}\boldsymbol{a}^{<i>} + b_y) \quad (i = 1, \ldots, 6)$$

Building block

$$\hat{y}^{<i>} = \sigma_{ya}(\boldsymbol{W}_{ya}\boldsymbol{a}^{<i>} + b_y) \quad (i = 1, \ldots, 6)$$

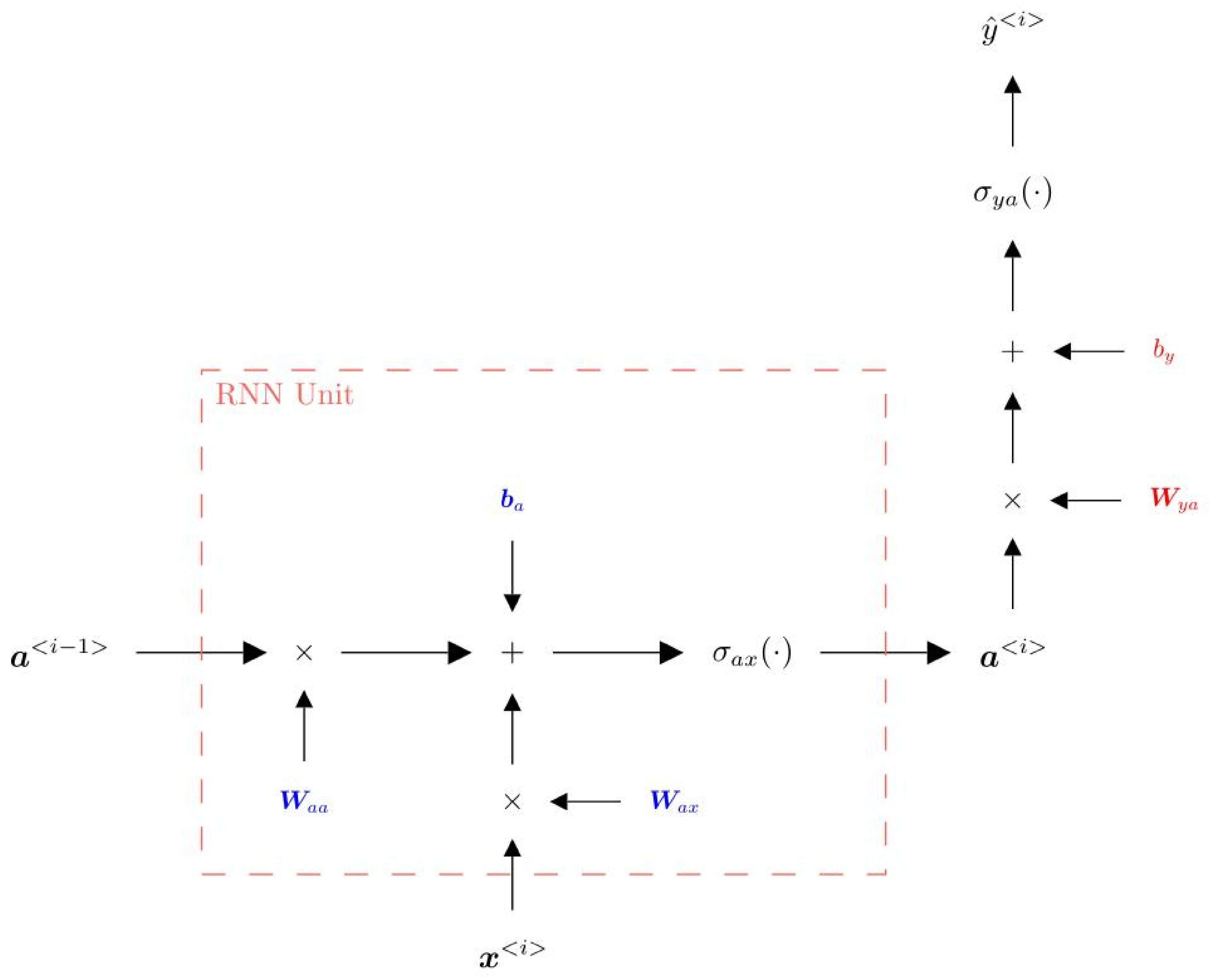
$$\begin{tikzcd}
 & \hat{y}^{<1>} & \hat{y}^{<2>} & \hat{y}^{<3>} & \hat{y}^{<4>} & \hat{y}^{<5>} & \hat{y}^{<6>} \\
\boldsymbol{a}^{<0>} \arrow[r] & \boldsymbol{a}^{<1>} \arrow[r] \arrow[u, "\boldsymbol{W}_{ya}, b_y"'] & \boldsymbol{a}^{<2>} \arrow[r] \arrow[u, "\boldsymbol{W}_{ya}, b_y"'] & \boldsymbol{a}^{<3>} \arrow[r] \arrow[u, "\boldsymbol{W}_{ya}, b_y"'] & \boldsymbol{a}^{<4>} \arrow[r] \arrow[u, "\boldsymbol{W}_{ya}, b_y"'] & \boldsymbol{a}^{<5>} \arrow[r] \arrow[u, "\boldsymbol{W}_{ya}, b_y"'] & \boldsymbol{a}^{<6>} \arrow[u, "\boldsymbol{W}_{ya}, b_y"'] \\
 & \boldsymbol{x}^{<1>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<2>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<3>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<4>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<5>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<6>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"']
\end{tikzcd}$$

$$\boldsymbol{a}^{<i>} = \sigma_{ax}(\boldsymbol{W}_{ax}\boldsymbol{x}^{<i>} + \boldsymbol{W}_{aa}\boldsymbol{a}^{<i-1>} + \boldsymbol{b}_a) \quad (i = 1, \ldots, 6)$$

Remark

1. Since we handle a binary classification problem, cross entropy is used as the loss function
2. Backpropagation can be derived based on the forward propagation in the previous slide
   - Remember: sum up all derivatives involving information about the model parameter

Flowchart


$\hat{y}^{<i>}$
$\sigma_{ya}(\cdot)$
$+$
b_y
$\times$
$\boldsymbol{W}_{ya}$
RNN Unit
$\boldsymbol{b}_a$
$\boldsymbol{a}^{<i-1>}$
$\times$
$+$
$\sigma_{ax}(\cdot)$
$\boldsymbol{a}^{<i>}$
$\boldsymbol{W}_{aa}$
$\times$
$\boldsymbol{W}_{ax}$
$\boldsymbol{x}^{<i>}$

Other examples

1. Sentiment analysis (Many to one)
   - Feature: a sentence like “I like this movie very much”
   - Label: score like “5”
2. Translation (many to many)
   - Feature: a sentence
   - Label: translated sentence
3. Text generation (? to many)
   - Feature: a start of a sentence like “I like ”
   - Label (finish this sentence)
4. ······

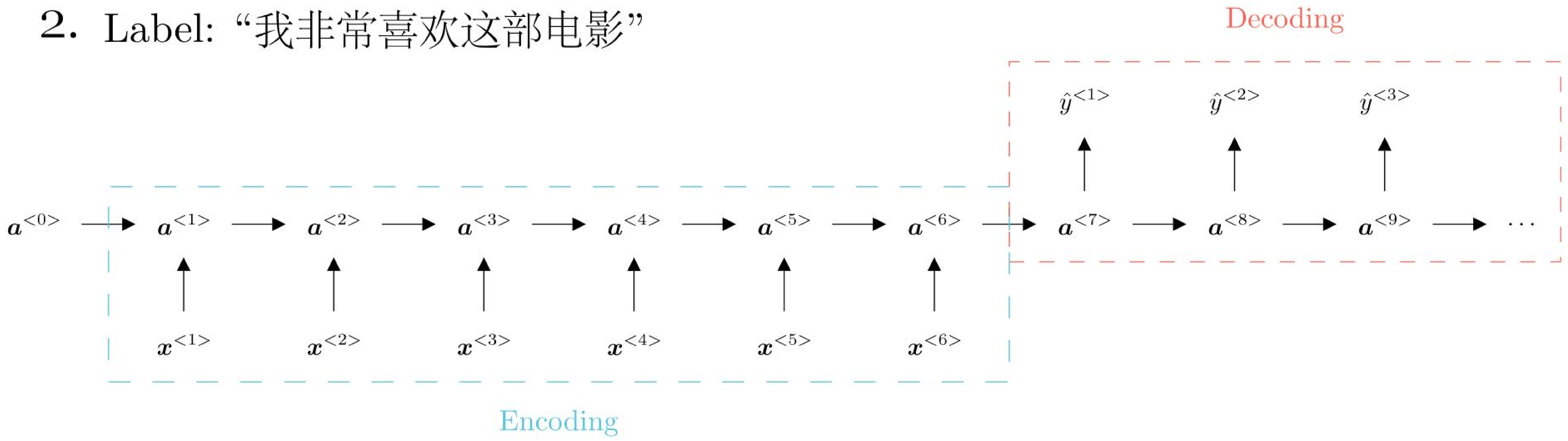
Sentiment analysis

1. Feature: a sentence like “I like this movie very much”
2. Label: score like “5”

$$\begin{tikzcd}
 & & & & & & \hat{y} \\
\boldsymbol{a}^{<0>} \arrow[r] & \boldsymbol{a}^{<1>} \arrow[r] & \boldsymbol{a}^{<2>} \arrow[r] & \boldsymbol{a}^{<3>} \arrow[r] & \boldsymbol{a}^{<4>} \arrow[r] & \boldsymbol{a}^{<5>} \arrow[r] & \boldsymbol{a}^{<6>} \arrow[u, "\boldsymbol{W}_{ya}, b_y"'] \\
 & \boldsymbol{x}^{<1>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<2>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<3>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<4>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<5>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"'] & \boldsymbol{x}^{<6>} \arrow[u, "\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{b}_a"']
\end{tikzcd}$$

Translation

1. Feature: a sentence like “I like this movie very much”
2. Label: “我非常喜欢这部电影”



Text generation

1. Feature: a sentence like "I like"
2. Task: complete this sentence

$$\begin{tikzcd}
& & \hat{y}^{<1>} \arrow[rd, dotted] & \hat{y}^{<2>} \arrow[rd, dotted] & \hat{y}^{<3>} & \\
\boldsymbol{a}^{<0>} \arrow[r] & \boldsymbol{a}^{<1>} \arrow[r] & \boldsymbol{a}^{<2>} \arrow[r] \arrow[u] & \boldsymbol{a}^{<3>} \arrow[r] \arrow[u] & \boldsymbol{a}^{<4>} \arrow[r] \arrow[u] & \cdots \\
& \boldsymbol{x}^{<1>} \arrow[u] & \boldsymbol{x}^{<2>} \arrow[u] & & &
\end{tikzcd}$$

Remarks

1. Bengio et al. (1994) pointed out that an RNN cannot capture long-term dependencies since the gradients may either vanish (most of the time) or explode (rarely, but with severe effects).

2. "This makes gradient-based optimization method struggle, not just because of the variations in gradient magnitudes but because of the effect of long-term dependenci is hidden (being exponentially smaller with respect to sequence length) by the effect of short-term dependencies" (Chung et al., 2014)