Natural Language Processing

- Kathirmani Sukumar

Course Outline

- Session1: NLP Introduction and Basics of Text Pre-processing (& web scraping)
- Session2: Text Processing : Vectorisation & Word Embedding
- Session 3 : Sequence to Sequence (Seq2seq) Learning
- Session 4 : Transformer
- Session 5: Generative AI and Large Language Models (LLMs)
- Session 6 : Case Study

Session 1: NLP Introduction

- Introduction to Natural Language Processing
  - Need for NLP
  - Applications of NLP
- Introduction to Text Pre-processing
  - Challenges With Text Data
  - Importance of Text Processing
- Common Text Pre-processing /Cleaning Methods
- Brief Introduction to Web Scraping (Beautiful Soup)
- Name Entity Recognition
- Word Cloud
- NLP Essential Libraries : NLTK/Spacy

Session 2:

- Conceptual Text Processing Terminologies
- Elementary text to numeric conversion techniques
  - Count vectorizer
  - TFIDF
- Introduction to Word Embeddings
  - Word2vec -Skip Gram, CBOW
  - Global vector (GloVe)
  - FastText
  - Pre-Trained Word Embeddings
- Keras packages/methods for Text Processing
- Use Case : Sentiment Analysis
- Libraries : Genism, Glove, Keras

NLP Applications

Text classification	Intent Identification	Entity Extraction
Sentiment Analysis	Text generation	Information Retrieval
Text Summarization	Virtual Assistants	Machine Translation
Text Summarization	Speech Processing	Topic Modeling

Text Preprocessing

- Case conversations
- Tokenization
- Stop word removal
- Root word identification
    - Stemming
    - Lemmatization
- Special characters removal
- …

TF-IDF

- Standardization technique
- Advantages
  - Helps to compare smaller documents with larger documents
  - Reduces the weightage of those terms which appear almost in the documents
  - Attempts to give higher relevance scores to words that occur in fewer documents within the corpus

TF-IDF Transformation

- Term Frequency (TF)

$$tf(t, D) = No.of\ times\ the\ term\ t, appeared\ in\ the\ document\ D$$

- Inverse Document Frequency (IDF)

$$idf(t) = \log\left(\frac{Total\ no.of\ documents\ in\ the\ corpus}{No.of\ documents\ in\ which\ the\ terms\ appears}\right)$$

  - Terms which appears in almost all the documents will have IDF close to zero

TF-IDF Transformation

- Term Frequency – Inverse Document Frequency (TF-IDF)

$$tfidf(t, D) = \frac{1}{D_N} * tf(t, D) * idf(t)$$

Where D_N is the no. of terms in the document D

- Advantages
  - Less importance to most frequent words appearing in all the documents (but not part of common stop words list)
  - Larger documents (i.e. high document length) can be compared with smaller documents

TF-IDF

DTM	T1	T2	T3	T4	T5
D1	1	0	2	1	2
D2	1	1	1	1	1
D3	0	0	3	1	0
D4	0	1	2	1	2
D5	5	0	0	0	1
D6	5	5	2	1	0

$$tf(t1, D1) = No.of\ times\ the\ term\ t1, appeared\ in\ the\ document\ D1 = 1$$

$$idf(t1) = \log\left(\frac{Total\ no.of\ documents\ in\ the\ corpus}{No.of\ documents\ in\ which\ the\ terms\ t\ appears}\right) = log_2\left(\frac{6}{4}\right) = 0.58$$

TF-IDF

DTM	T1	T2	T3	T4	T5
D1	1/6*0.58	0	2	1	2
D2	1	1	1	1	1
D3	0	0	3	1	0
D4	0	1	2	1	2
D5	5	0	1	0	1
D6	5	5	2	1	0

$$tf(t1, D1) = No.of\ times\ the\ term\ t1, appeared\ in\ the\ document\ D1 = 1$$

$$idf(t1) = \log\left(\frac{Total\ no.of\ documents\ in\ the\ corpus}{No.of\ documents\ in\ which\ the\ terms\ t\ appears}\right) = log_2\left(\frac{6}{4}\right) = 0.58$$

TF-IDF

DTM	T1	T2	T3	T4	T5
D1	1/6*0.58	0	2	1	2
D2	1/5*0.58	1	1	1	1
D3	0/4*0.58	0	3	1	0
D4	0/6*0.58	1	2	1	2
D5	5/7*0.58	0	1	0	1
D6	5/13*0.58	5	2	1	0

$$tf(t1, D1) = No.of\ times\ the\ term\ t1, appeared\ in\ the\ document\ D1 = 1$$

$$idf(t1) = \log\left(\frac{Total\ no.of\ documents\ in\ the\ corpus}{No.of\ documents\ in\ which\ the\ terms\ t\ appears}\right) = log_2\left(\frac{6}{4}\right) = 0.58$$

TF-IDF

DTM	T1	T2	T3	T4	T5
D1	1/6*0.58	0	2	1	2
D2	1/5*0.58	1	1	1	1
D3	0/4*0.58	0	3	1	0
D4	0/6*0.58	1	2	1	2
D5	5/7*0.58	0	1	0	1
D6	5/13*0.58	5	2	1	0

$$tf(t3, D1) = No.of\ times\ the\ term\ t3, appeared\ in\ the\ document\ D1 = 2$$

$$idf(t3) = \log\left(\frac{Total\ no.of\ documents\ in\ the\ corpus}{No.of\ documents\ in\ which\ the\ terms\ t\ appears}\right) = log_2\left(\frac{6}{6}\right) = 0$$

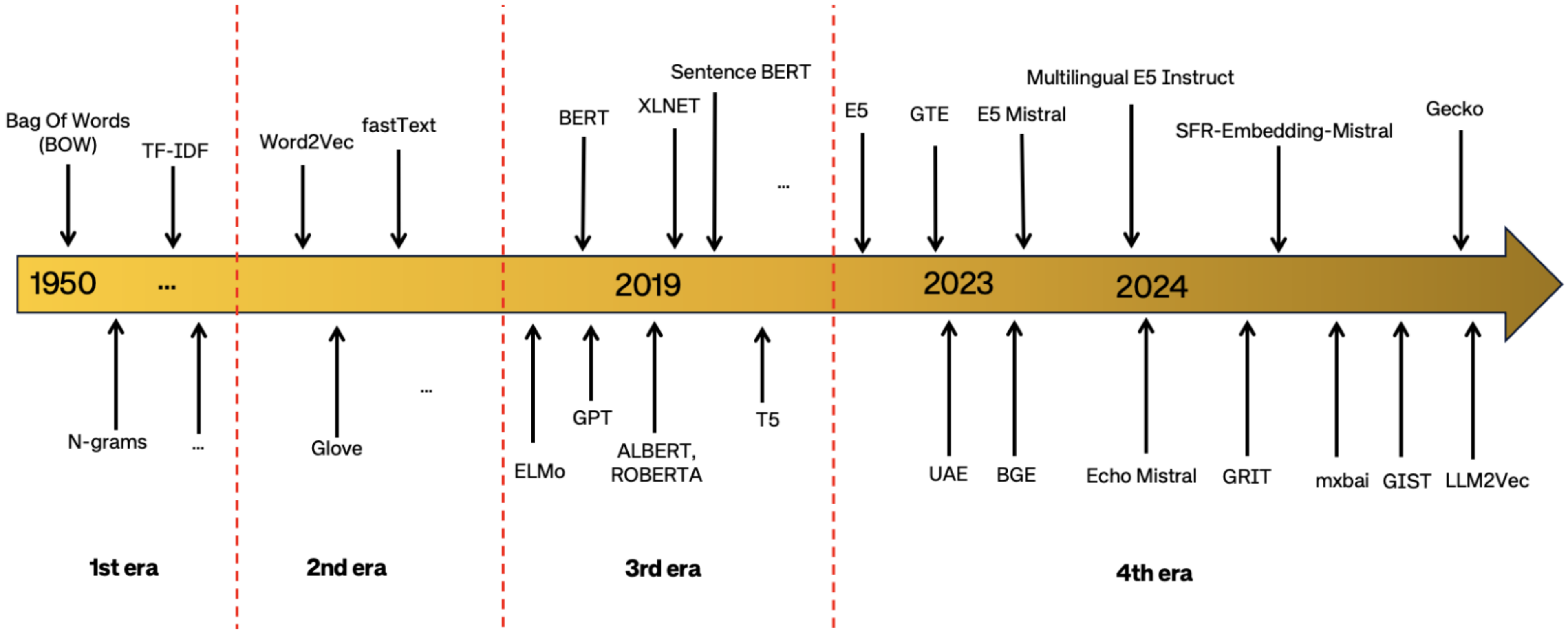
TF-IDF

DTM	T1	T2	T3	T4	T5
D1	1/6*0.58	0	2/6 * 0	1	2
D2	1/5*0.58	1	1/5 * 0	1	1
D3	0/4*0.58	0	3/4 * 0	1	0
D4	0/6*0.58	1	2/6 * 0	1	2
D5	5/7*0.58	0	1/7 * 0	0	1
D6	5/13*0.58	5	2/13 * 0	1	0

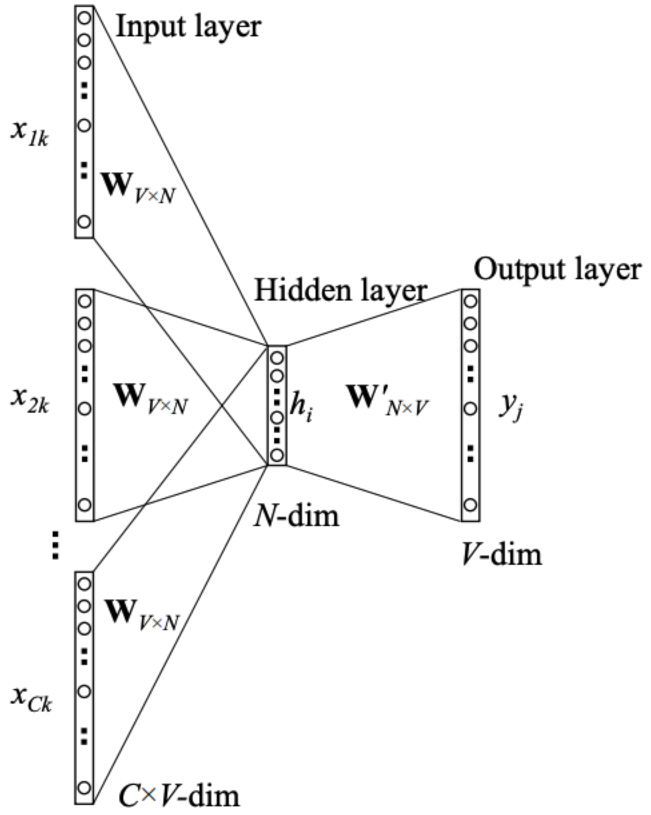
$$tf(t3, D1) = No.of\ times\ the\ term\ t3, appeared\ in\ the\ document\ D1 = 2$$

$$idf(t3) = \log\left(\frac{Total\ no.of\ documents\ in\ the\ corpus}{No.of\ documents\ in\ which\ the\ terms\ t\ appears}\right) = log_2\left(\frac{6}{6}\right) = 0$$

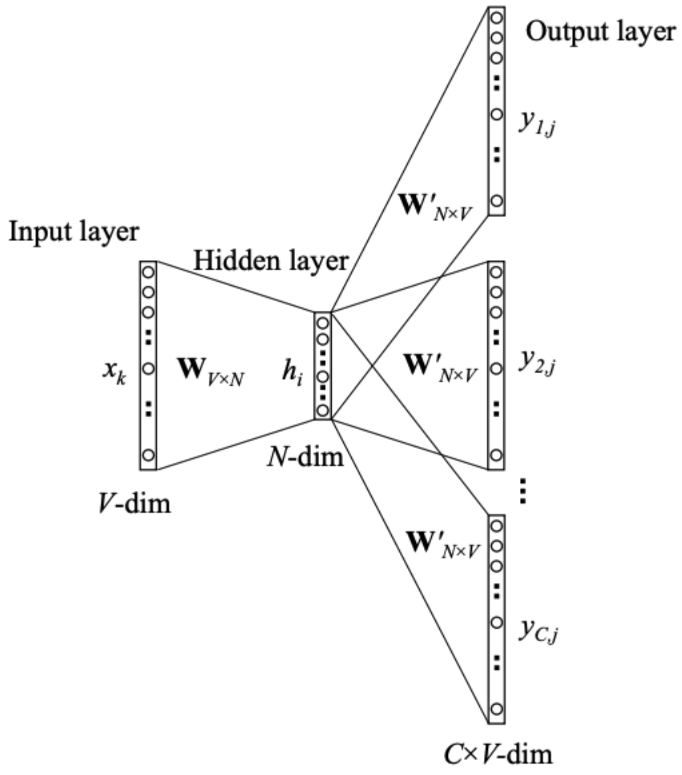
Text Embeddings Evolution



Word2vec Architectures

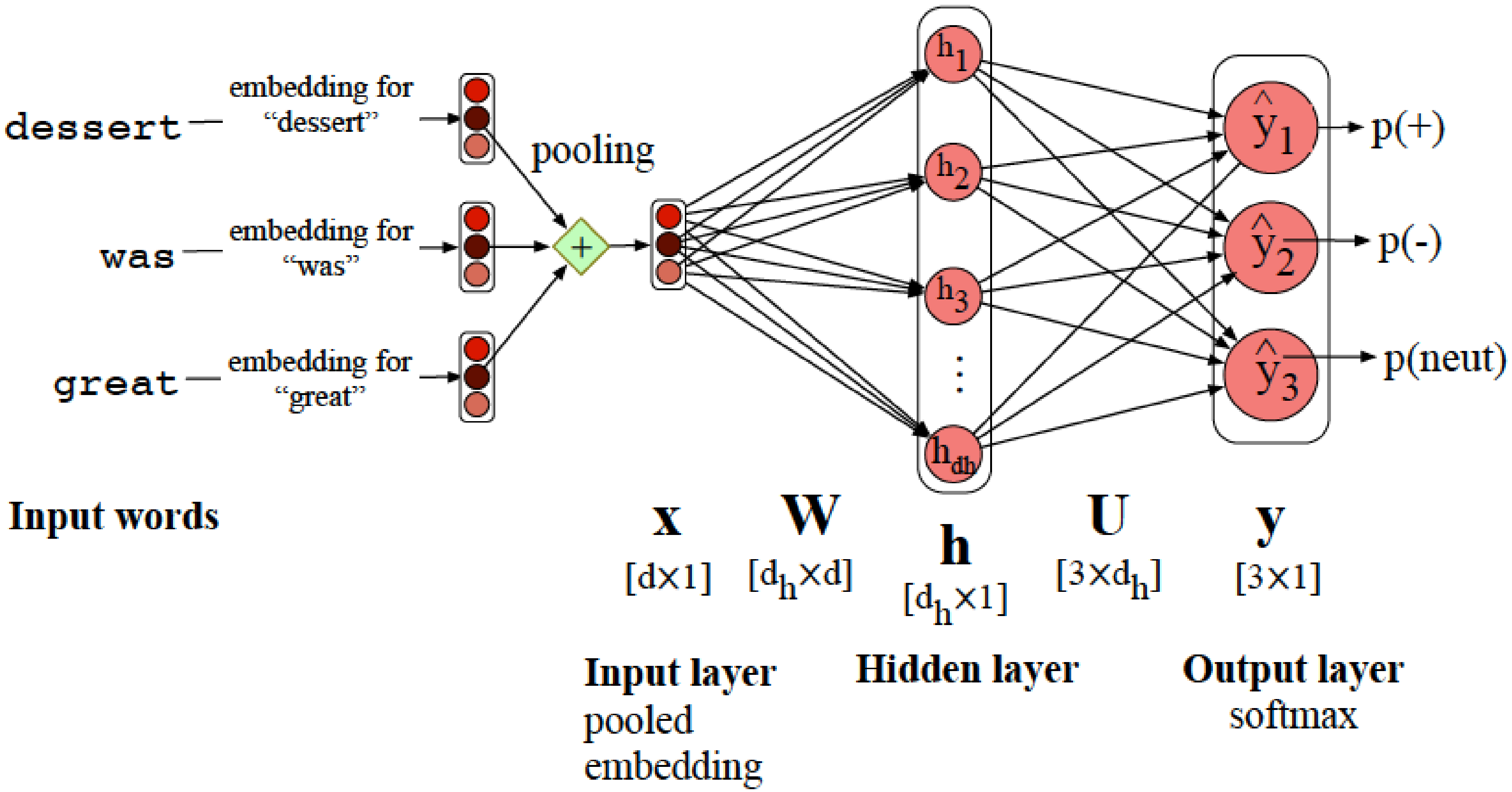


Continuous Bag of Words (CBOW)



Skip Gram

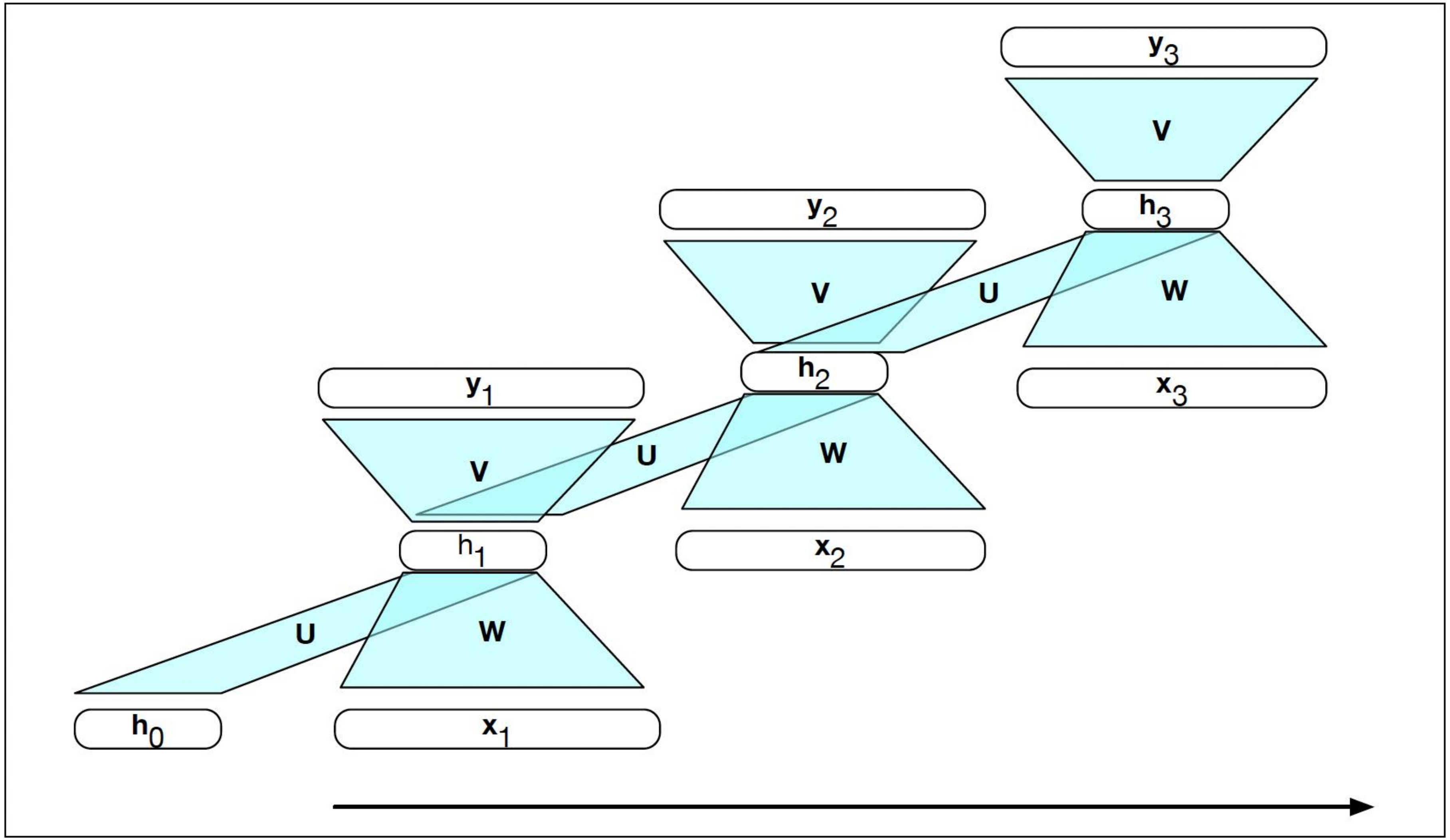
Text classification using embeddings



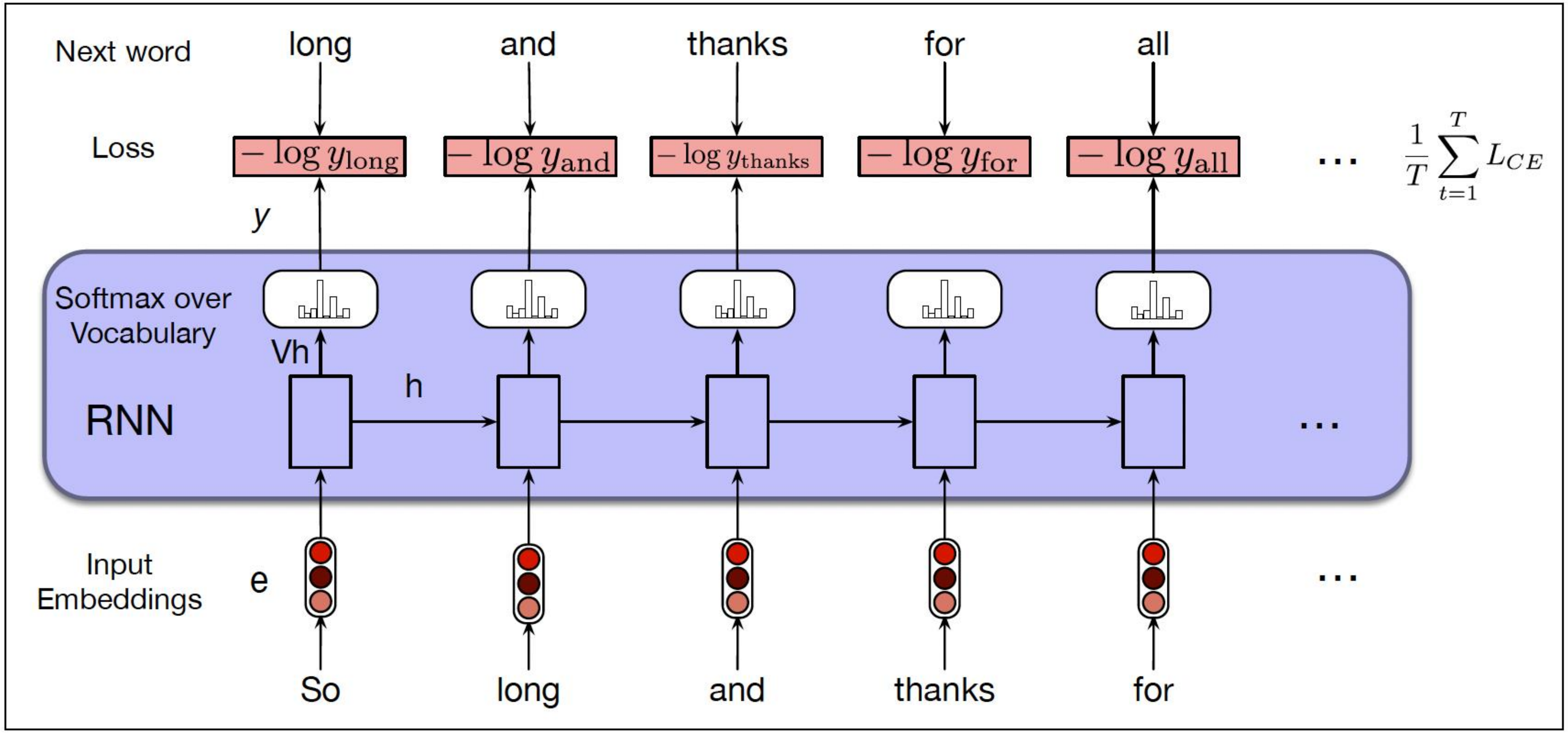
$$\mathbf{H} = \sigma(\mathbf{X}\mathbf{W}^\intercal + \mathbf{b})$$
$$\mathbf{Z} = \mathbf{H}\mathbf{U}^\intercal$$
$$\hat{\mathbf{Y}} = \mathrm{softmax}(\mathbf{Z})$$

Source: Speech and Language Processing book

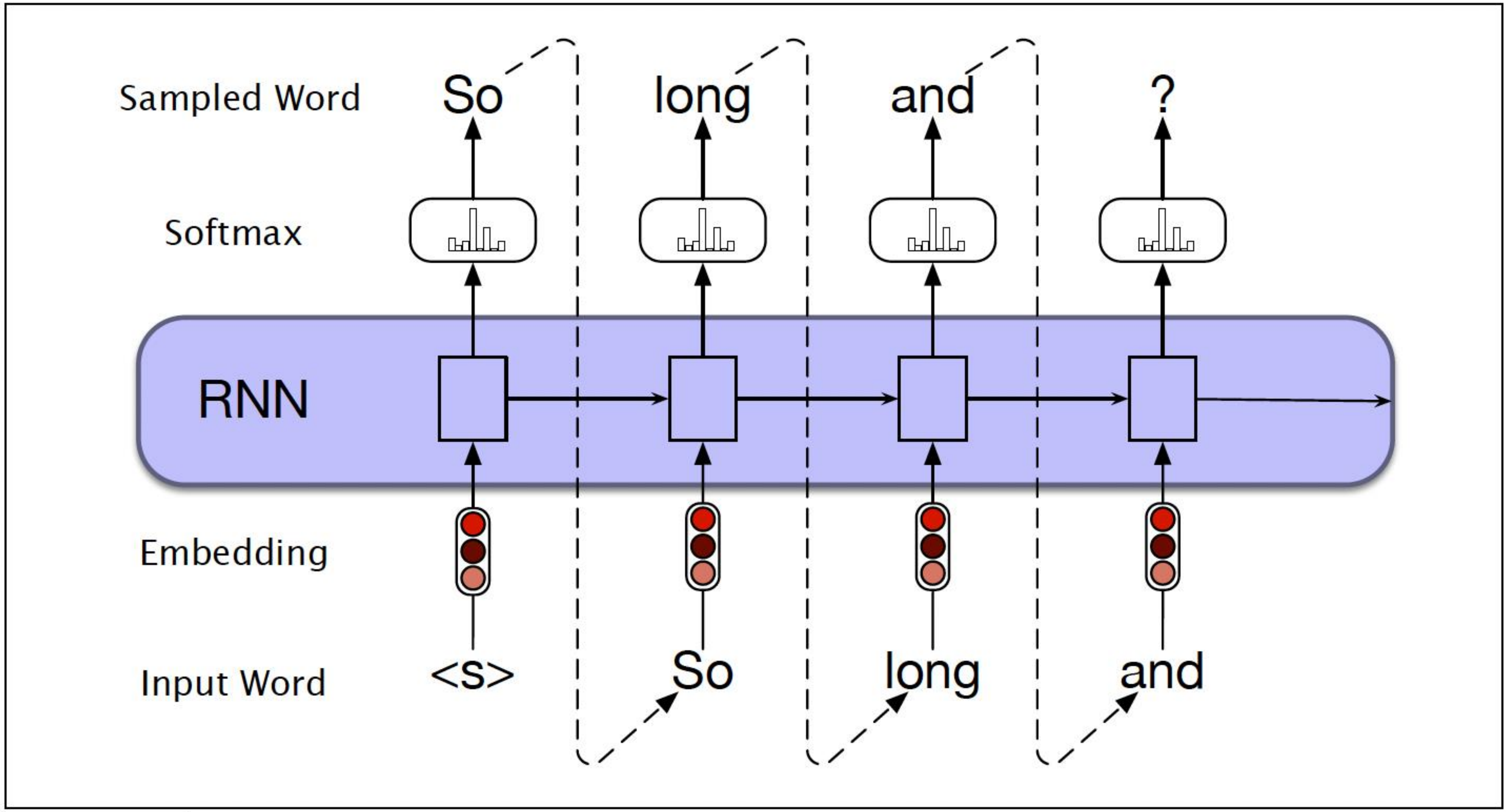
RNN



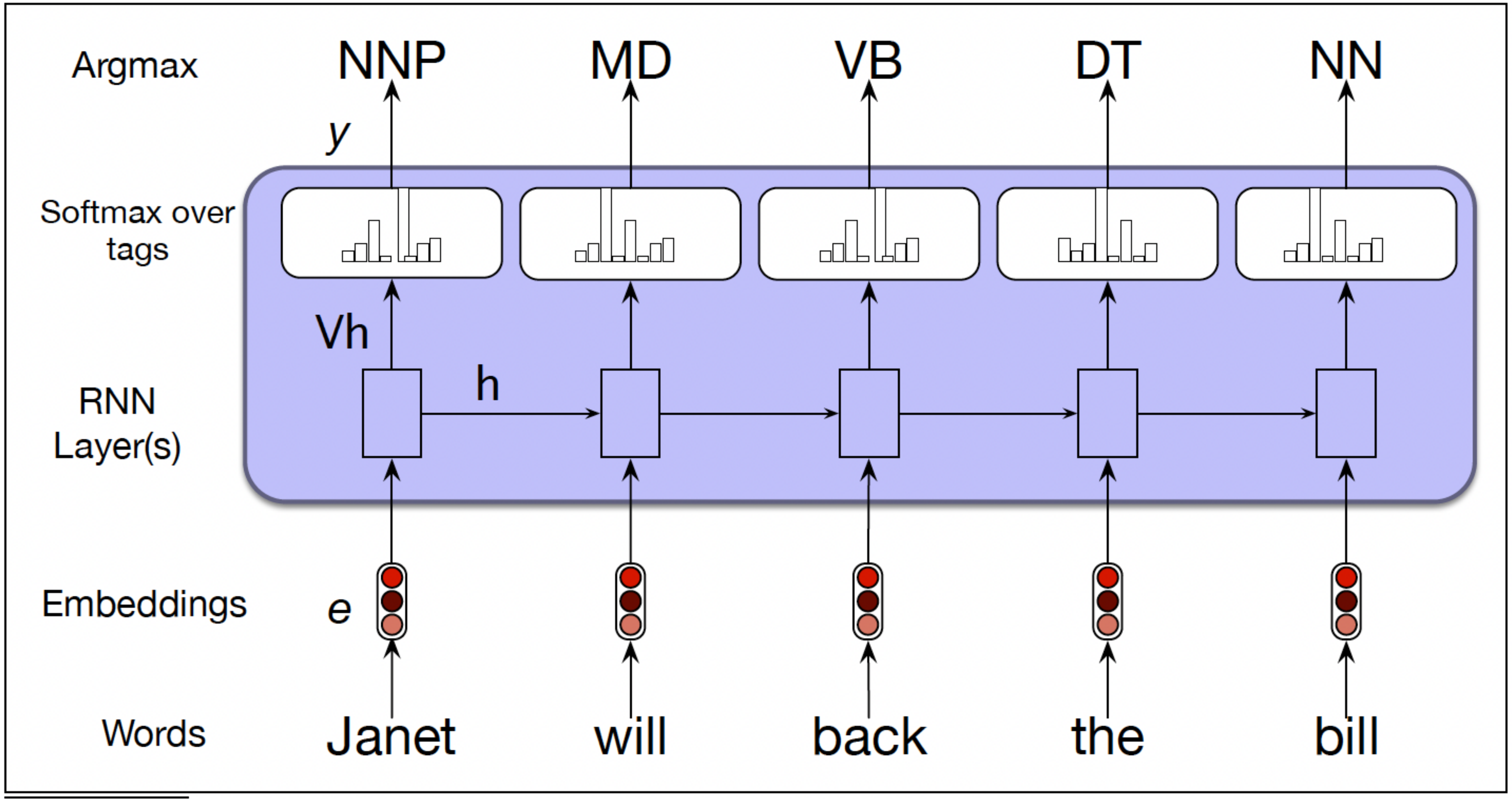
RNN – Language Modeling



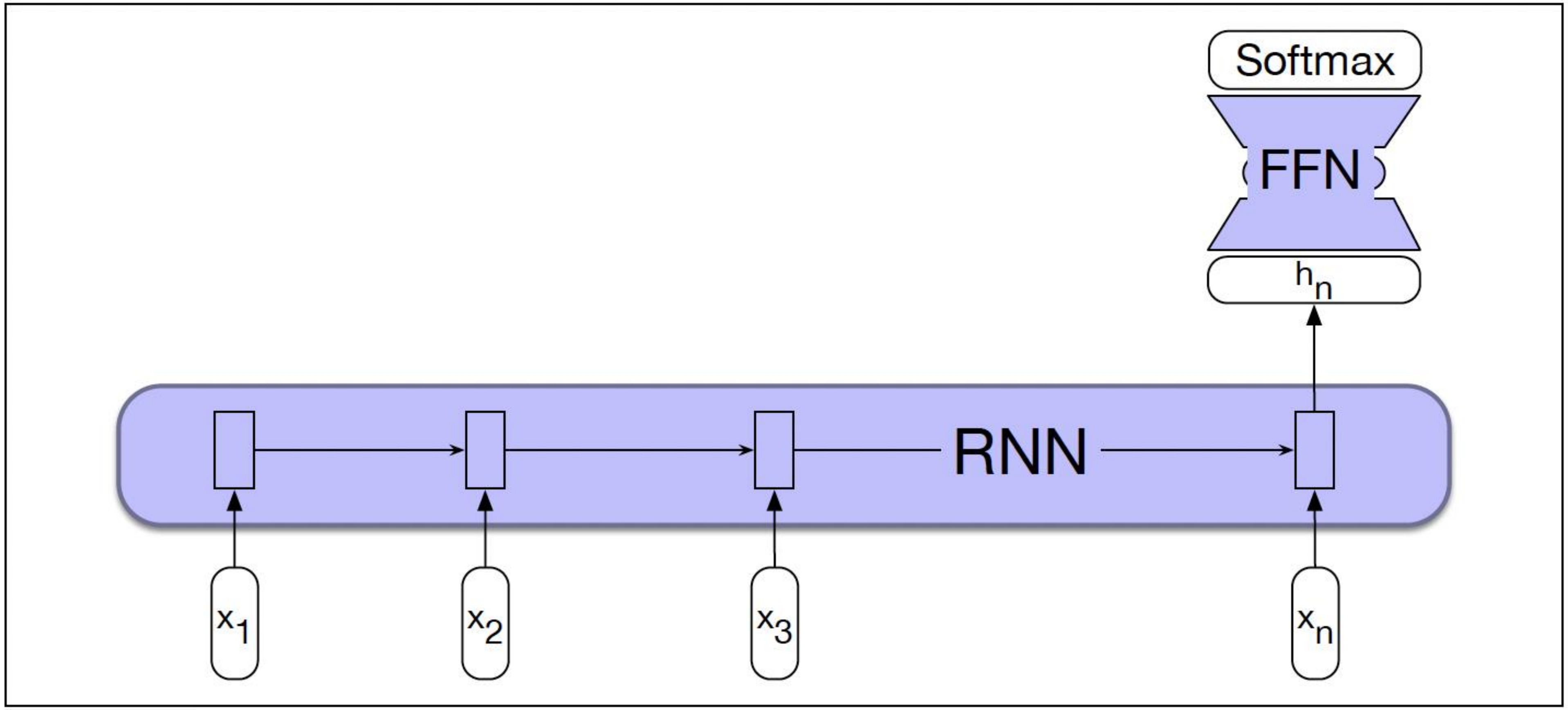
RNN – Language Modeling



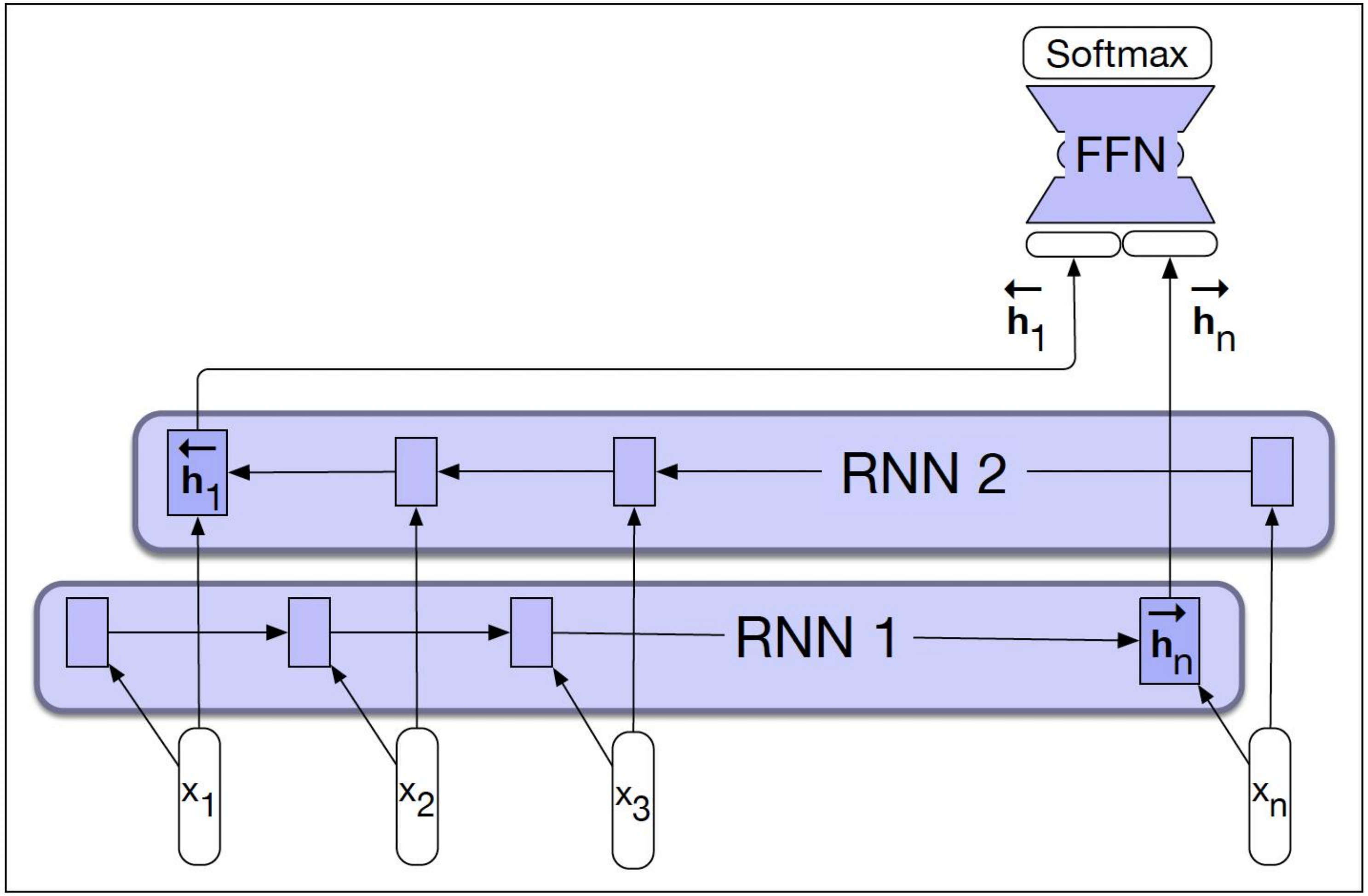
RNN – Sequence Labeling



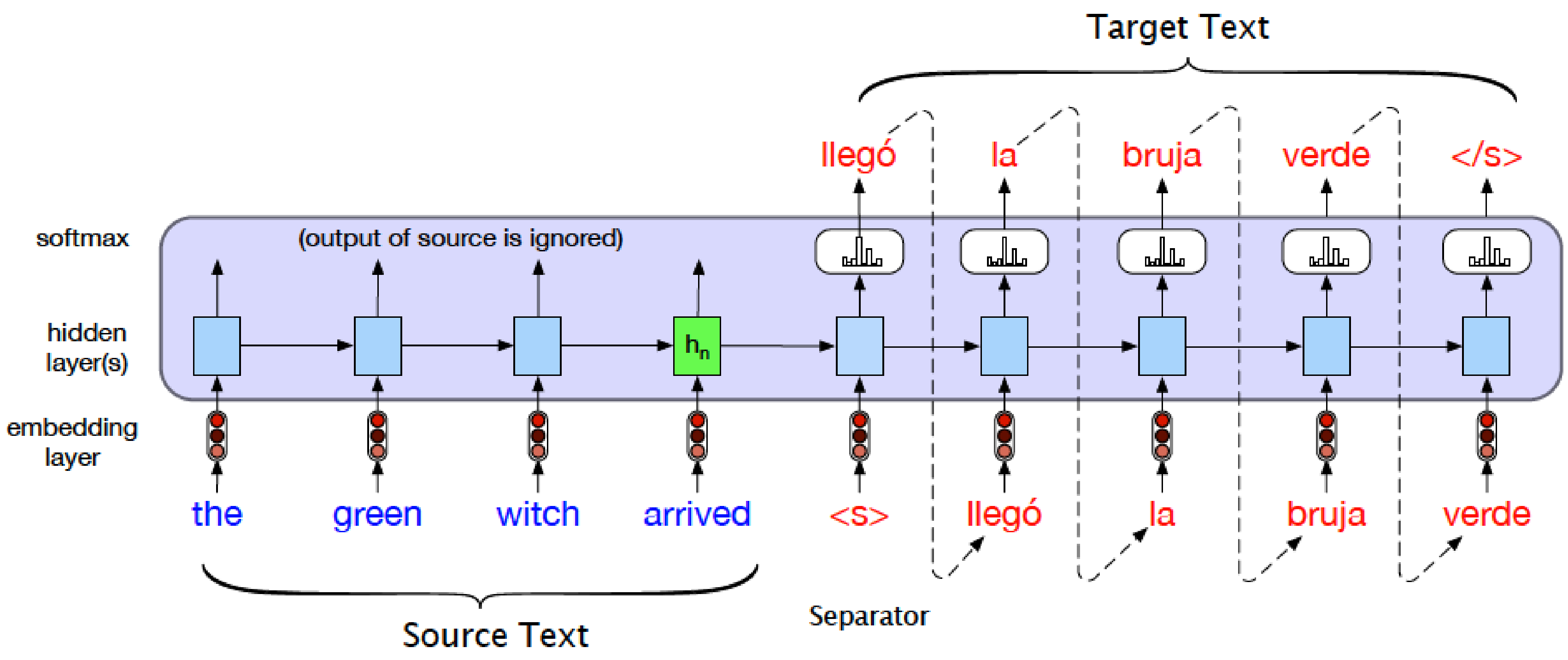
RNN – Document Classification



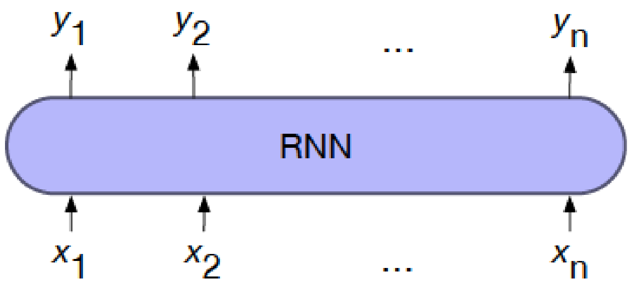
Bi-Directional RNN



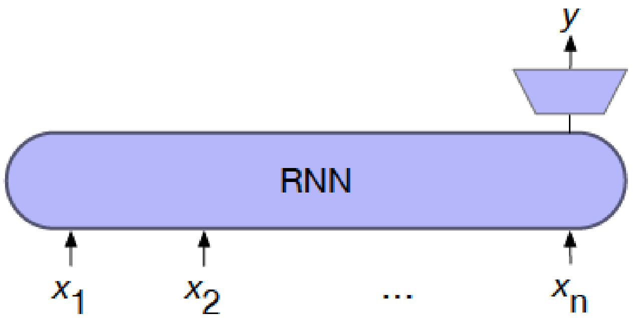
Encoder - Decoder



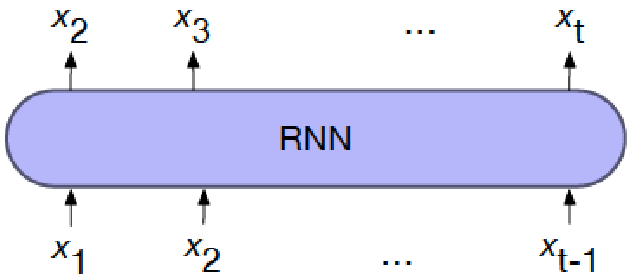
RNN Architectures



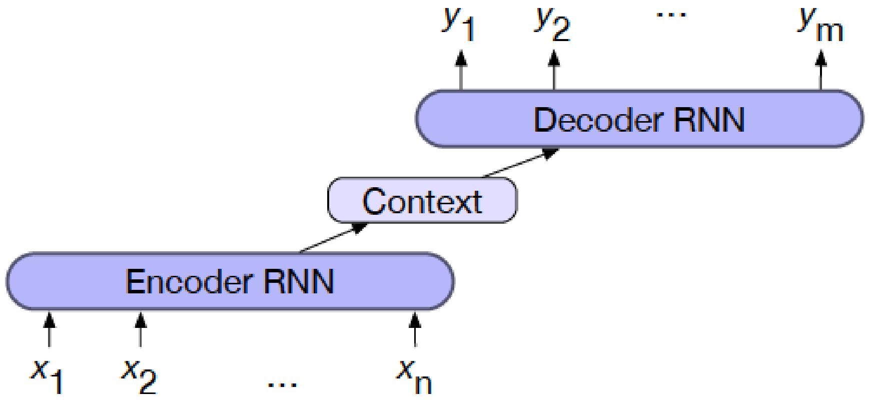
a) sequence labeling



b) sequence classification

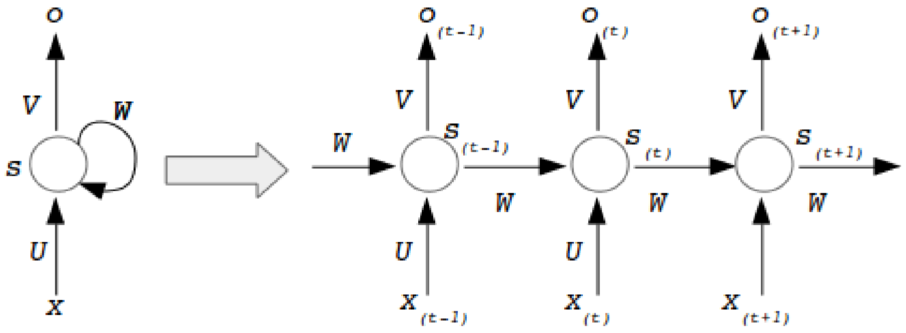


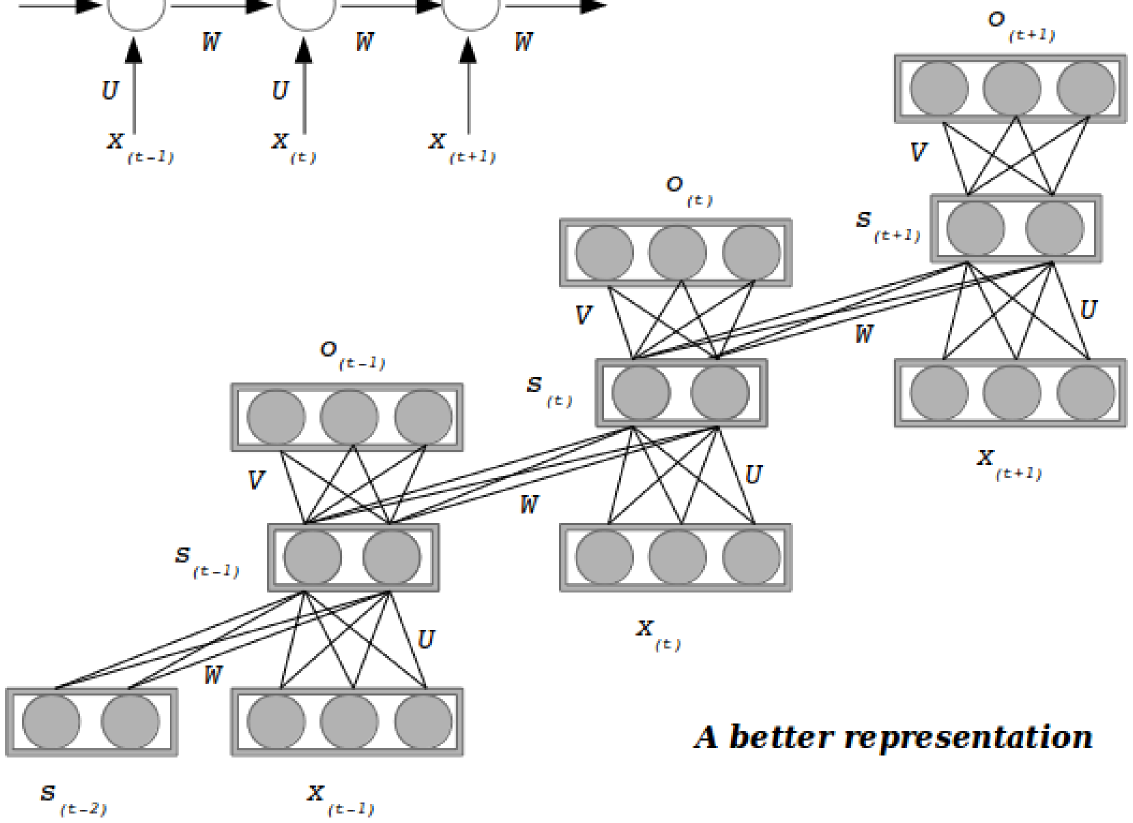
c) language modeling



d) encoder-decoder

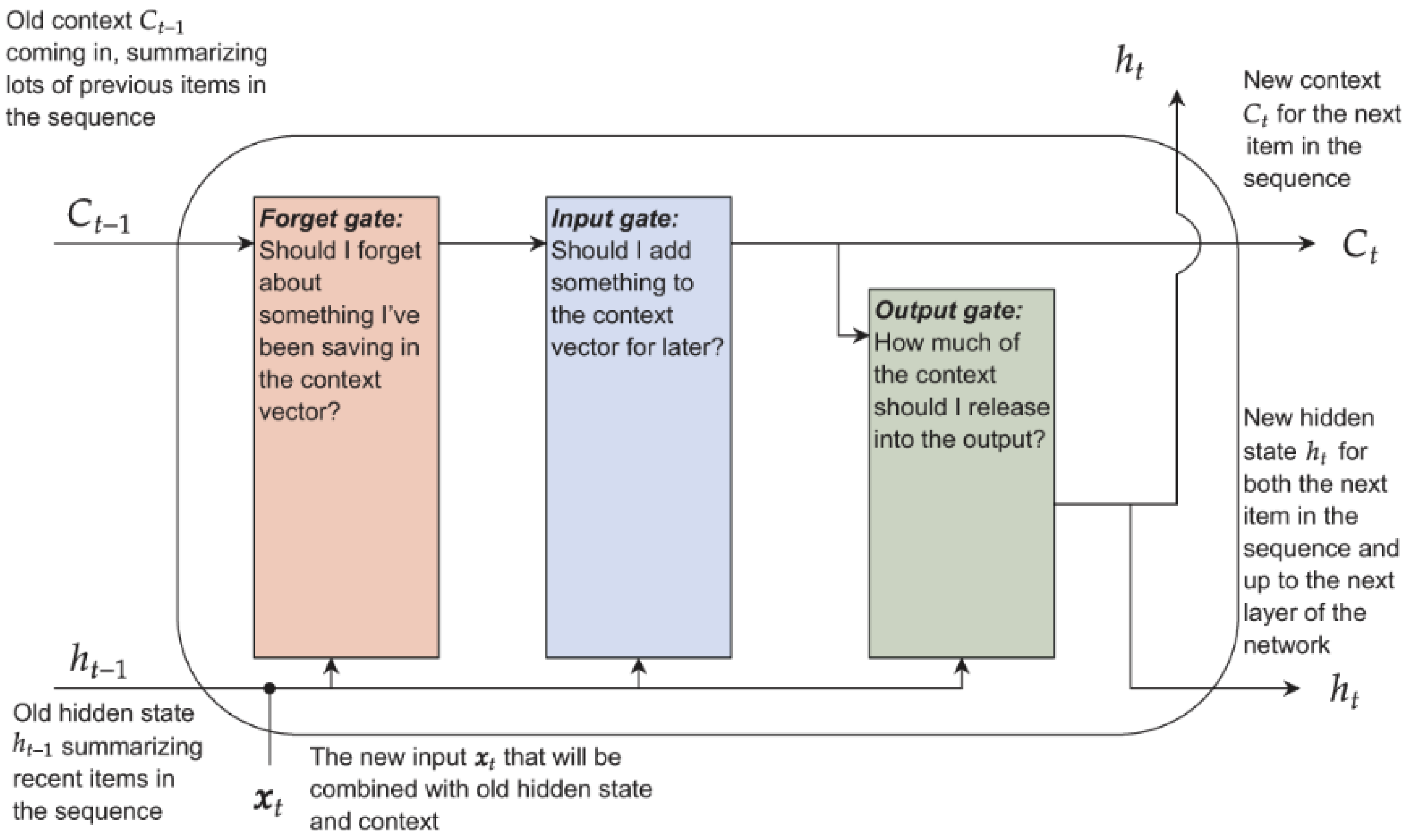
Unrolling a RNN



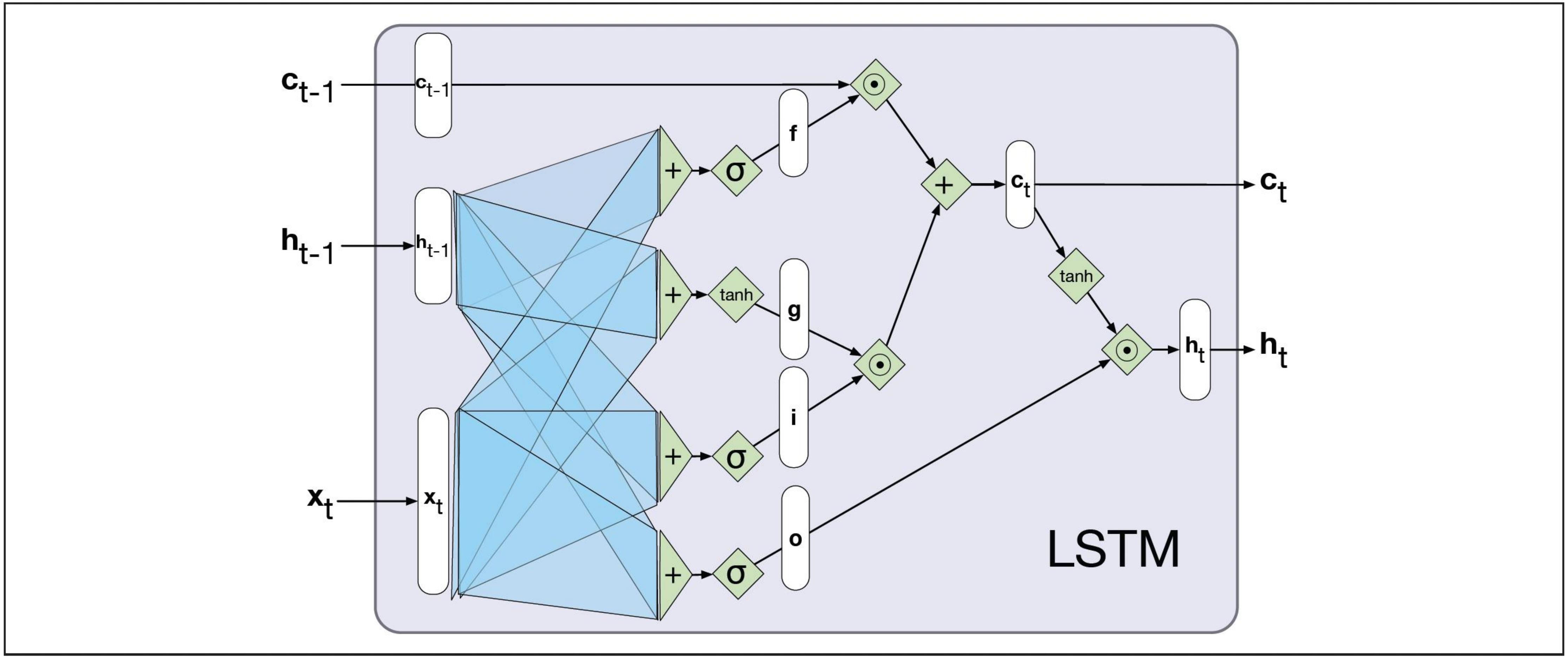


A better representation

LSTM


Old context C_{t-1} coming in, summarizing lots of previous items in the sequence
C_{t-1}
Forget gate: Should I forget about something I've been saving in the context vector?
Input gate: Should I add something to the context vector for later?
Output gate: How much of the context should I release into the output?
h_t
New context C_t for the next item in the sequence
C_t
New hidden state h_t for both the next item in the sequence and up to the next layer of the network
h_t
h_{t-1}
Old hidden state h_{t-1} summarizing recent items in the sequence
x_t
The new input x_t that will be combined with old hidden state and context

LSTM Gates



Simple RNN vs LSTM