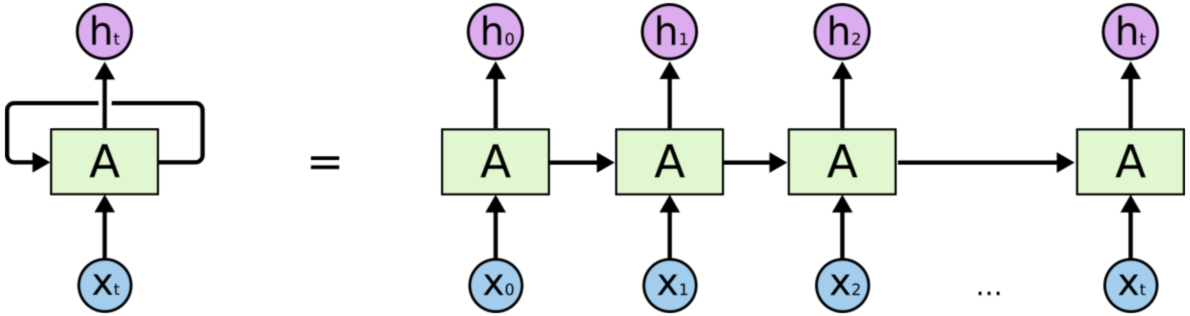
Deep Learning Workshop

Recurrent Neural Networks

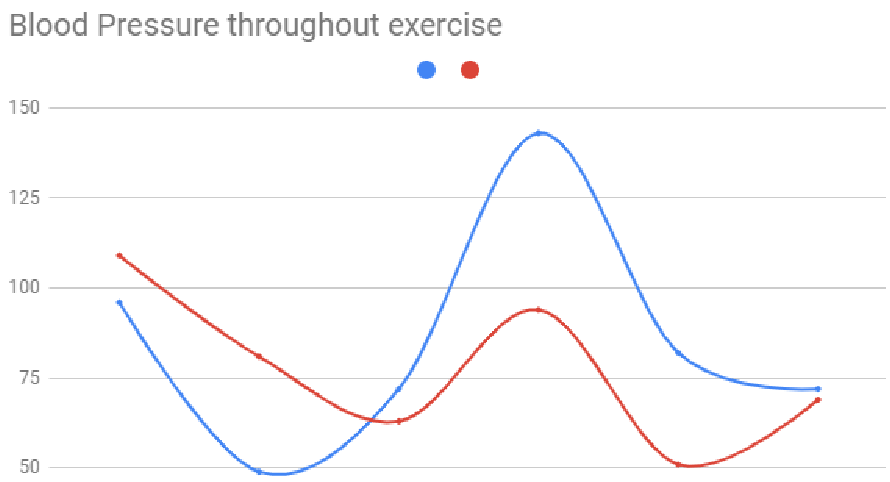


Instructor: Aaron Low

HELP University, Faculty of Computing and Digital Technology

Sequential Data

Stock Market

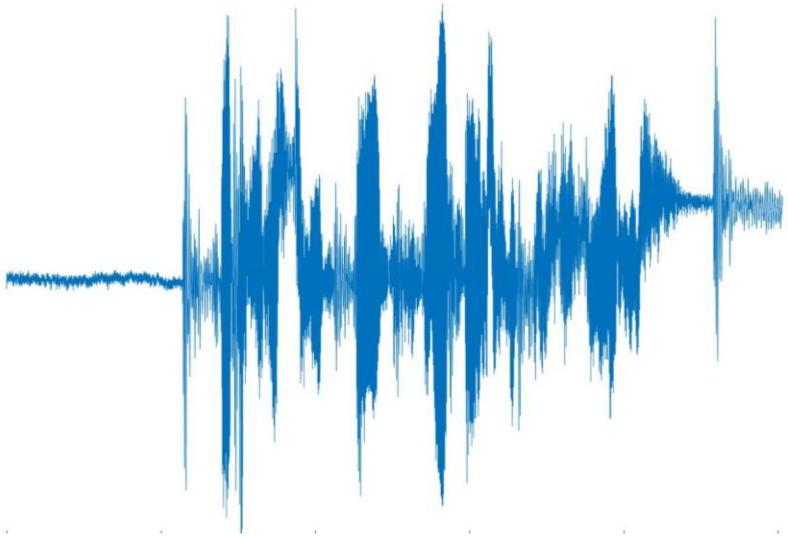


Blood Pressure

Sequential Data

"The quick brown fox jumps over the lazy dog" is an English-language pangram—a sentence that contains all of the letters of the alphabet. It is commonly used for touch-typing practice, testing typewriters and computer keyboards, displaying examples of fonts, and other applications involving text where the use of all letters in the alphabet is desired. Owing to its brevity and coherence, it has become widely known.

Text



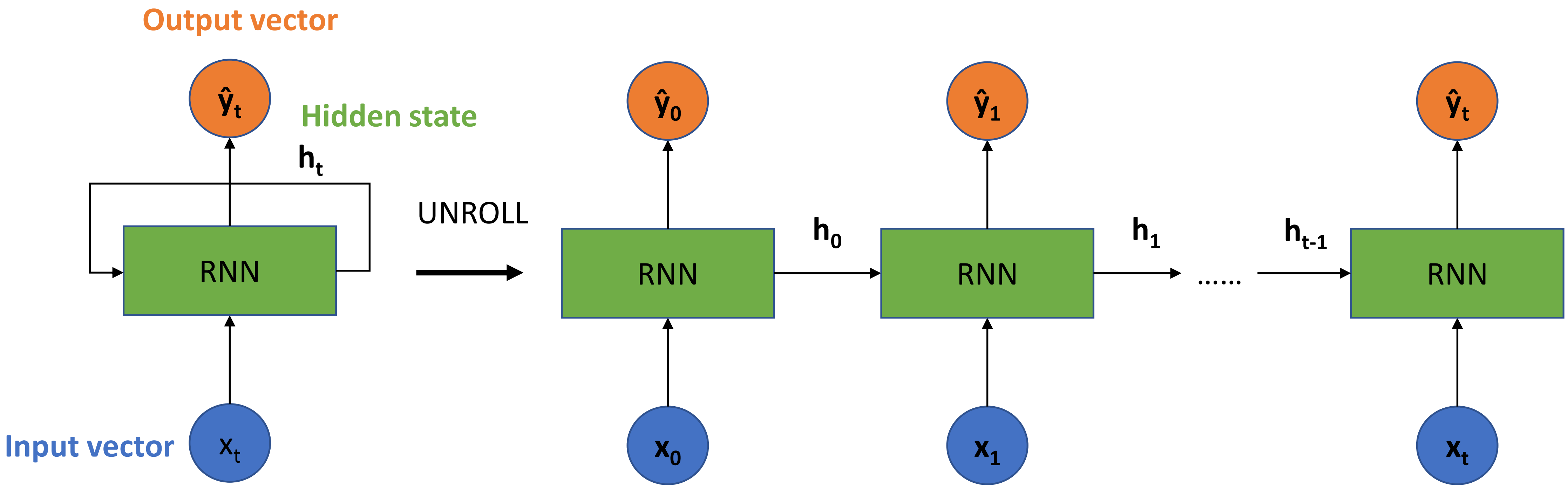
Audio

Sequential Data

Video Frames

Sequential Modelling Tasks: Trajectory Prediction

Recurrent Neural Network



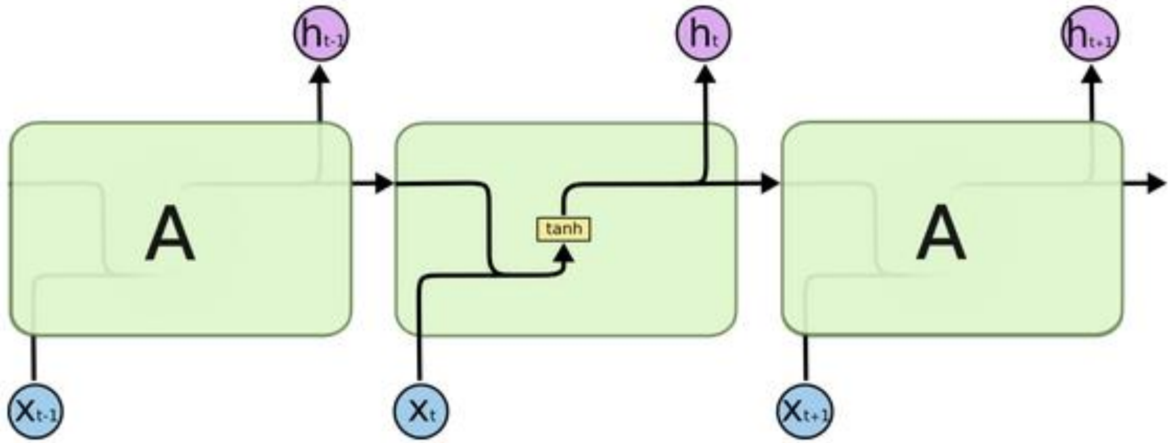
Recurrent Neural Network

- Update hidden state

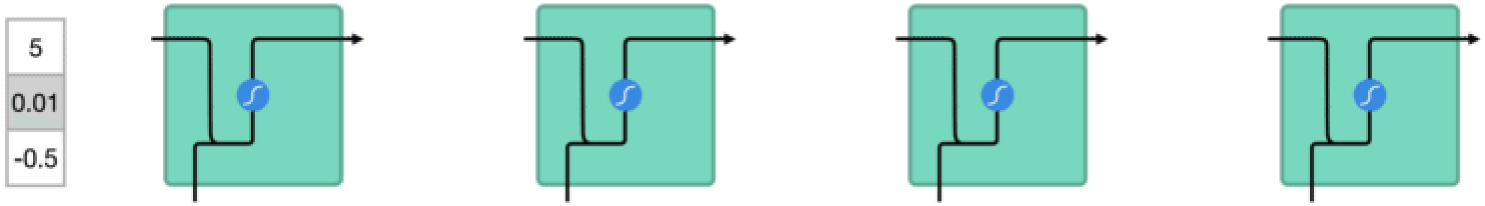
$$h_t = \tanh(\boldsymbol{W_{hh}^{t}} h_{t\,\_\,1} + \boldsymbol{W_{xh}^{t}} x_t)$$

- Generate output

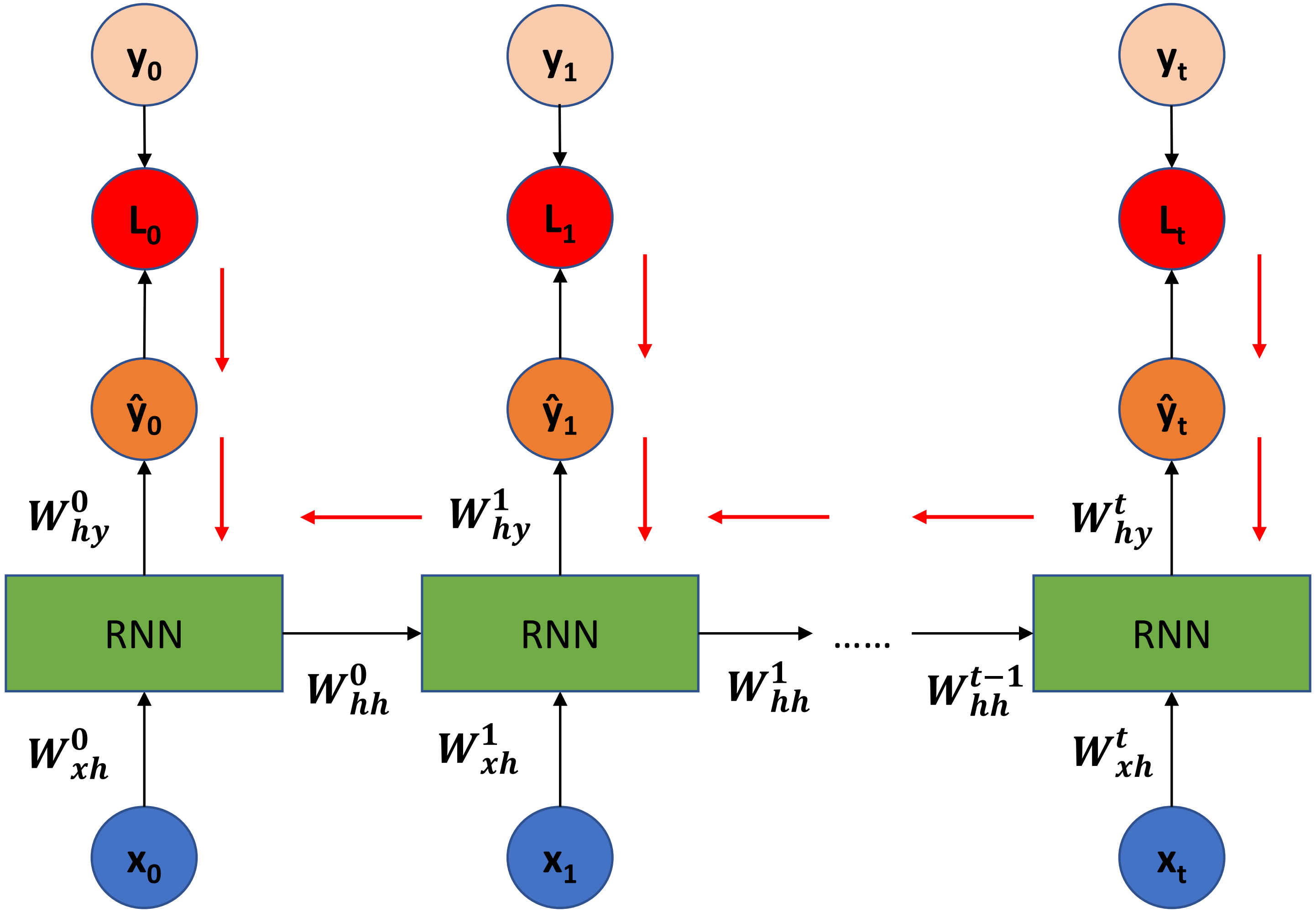
$$\hat{y}_t = \boldsymbol{W_{hy}^{t}} h_t$$



Recurrent Neural Network

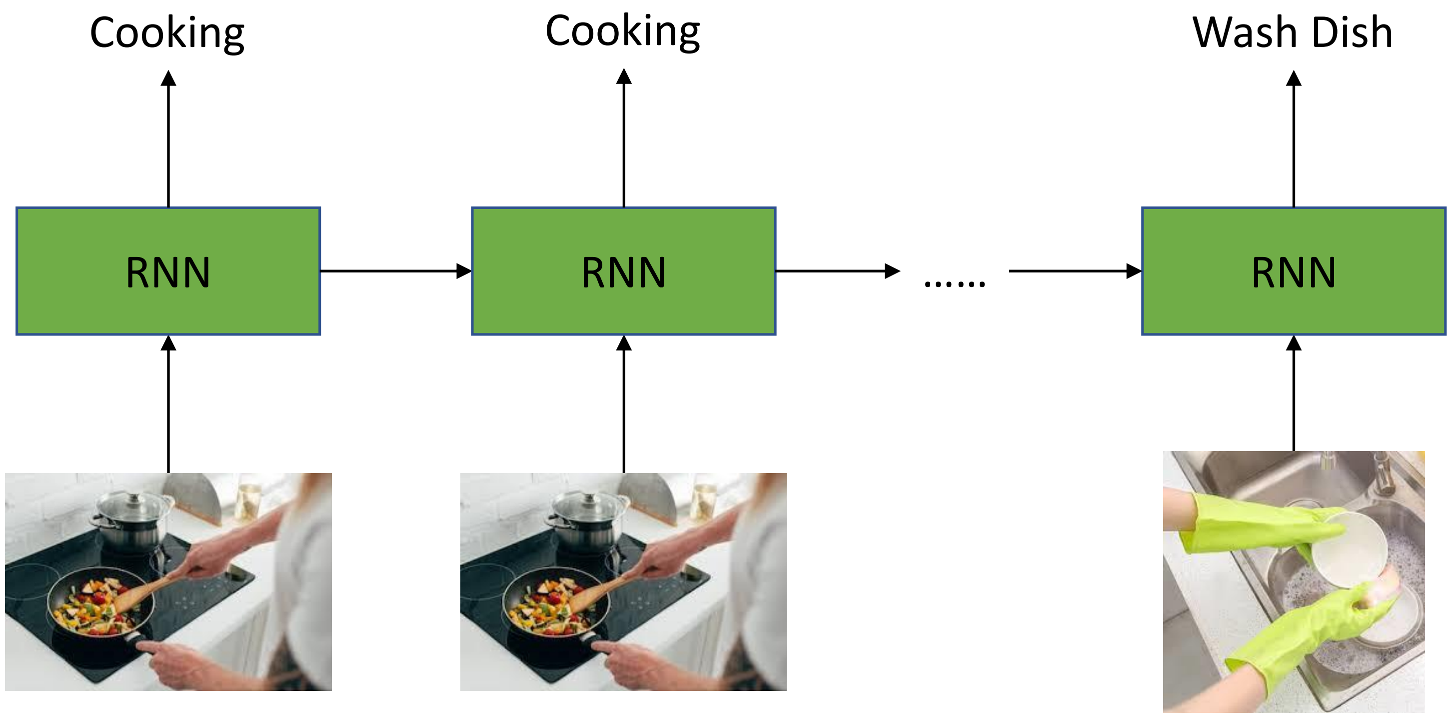


Backpropagation Through Time



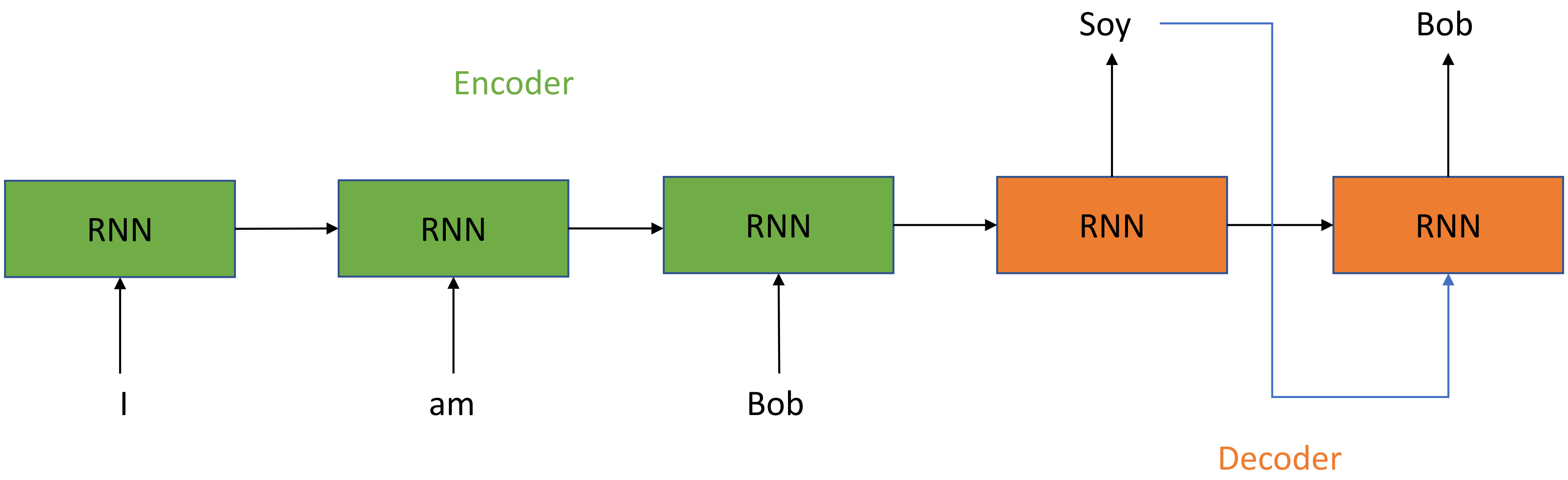
RNN Types: Many-to-Many

- **Example:** Video Frame Classification
- **Input:** Video frames
- **Output:** Corresponding frame classification



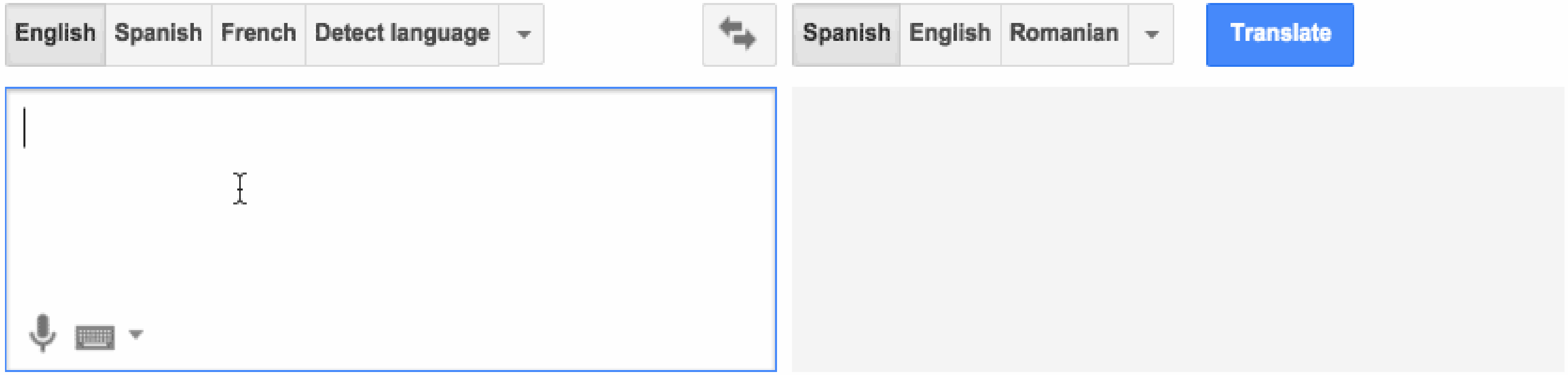
RNN Types: Many-to-Many (input and output different length)

- **Example:** Machine Translation
- **Input:** Original sentence (English)
- **Output:** Translated sentence (Spanish)



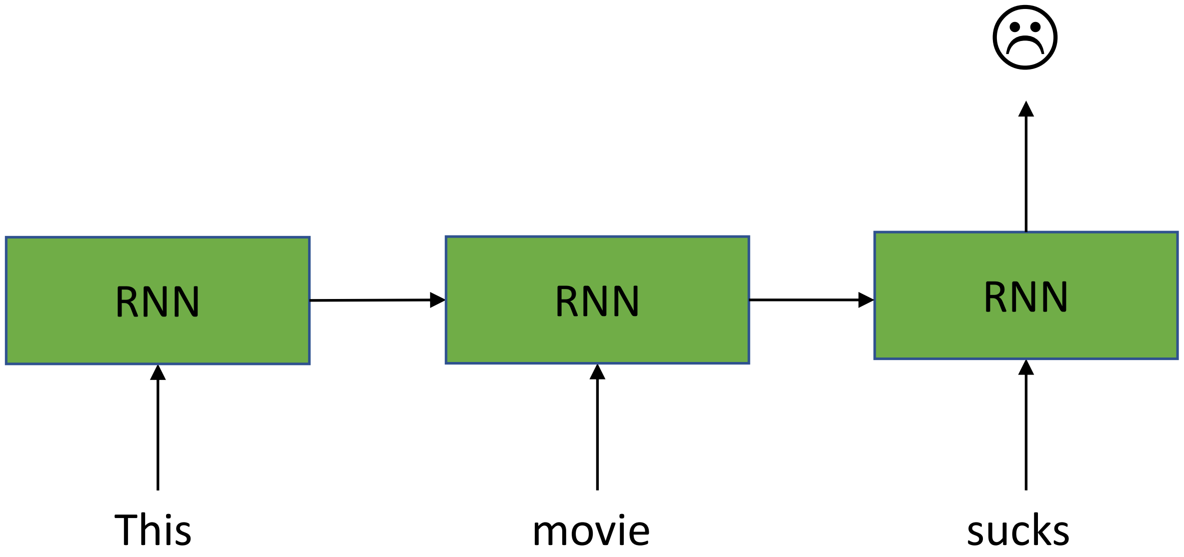
RNN Types: Many-to-Many (input and output different length)

- **Example:** Machine Translation
- **Input:** Original sentence (English)
- **Output:** Translated sentence (Spanish)



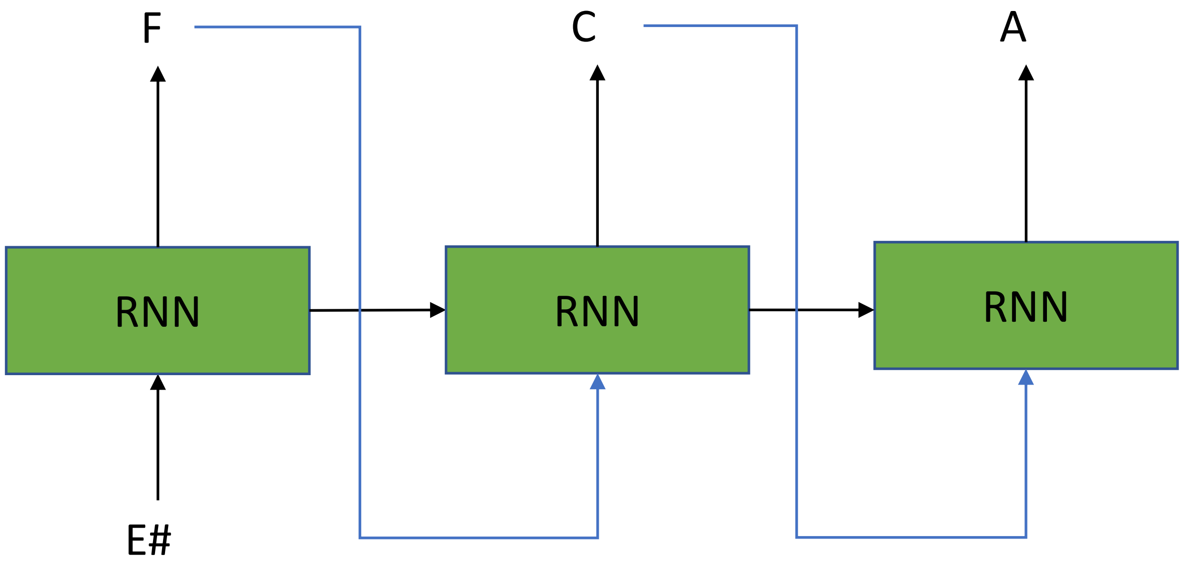
RNN Types: Many-to-One

- **Example:** Sentiment Classification
- **Input:** Text
- **Output:** Sentiment



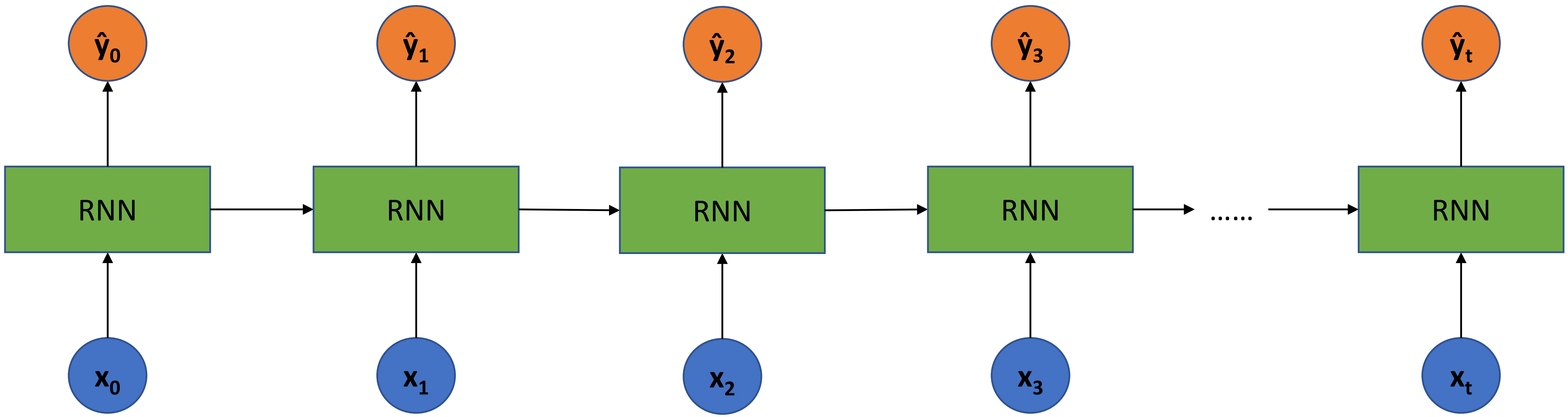
RNN Types: One-to-Many

- **Example:** Music Generation
- **Input:** Music Note
- **Output:** Music Sequence



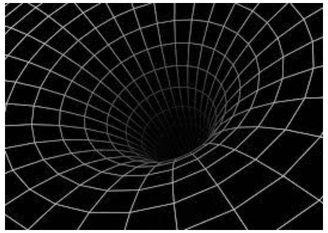
Exploding and Vanishing Gradients

- **As neural network becomes deeper, gradient propagation can result in gradients becoming vanishing-ly small or exploding-ly large**
- **Long sequence RNNs suffer similarly**



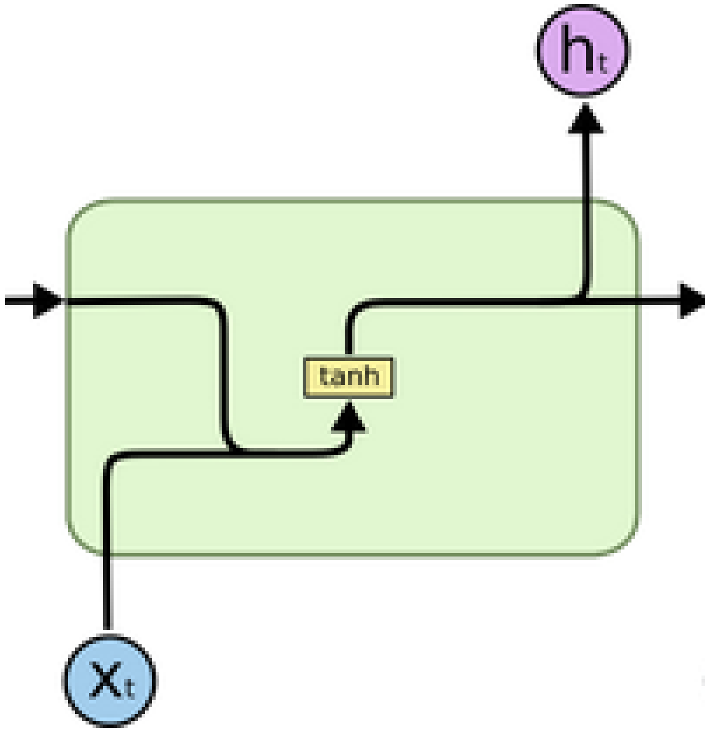
Exploding and Vanishing Gradients: Solutions

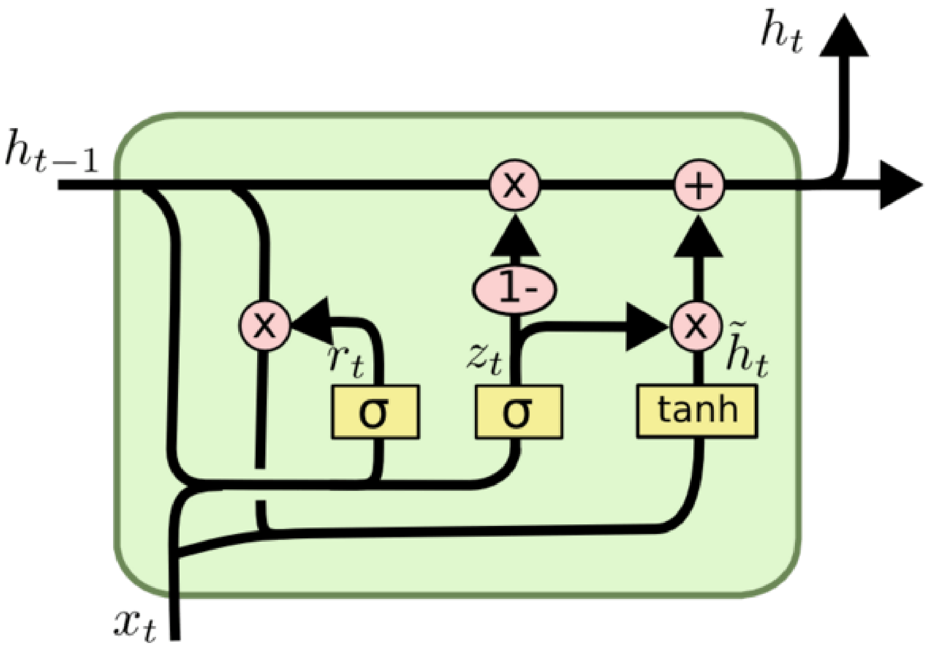
- Exploding gradients can be dealt with **gradient clipping**
- Vanishing gradients can be dealt with
  - Weight Initialization
  - Activation Function
  - Weight Regularization
  - **Gated cells**



Gated Cells

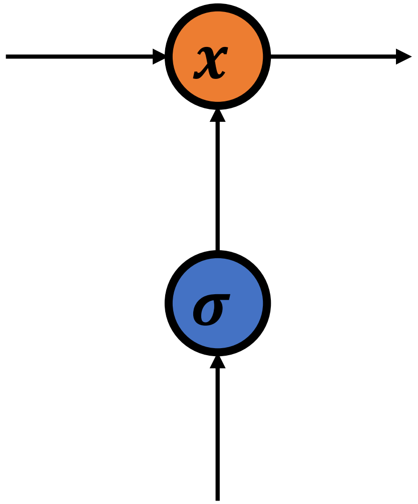
- Add gates to the recurrent unit to control what information is passed through
- Example: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)





Gated Cells

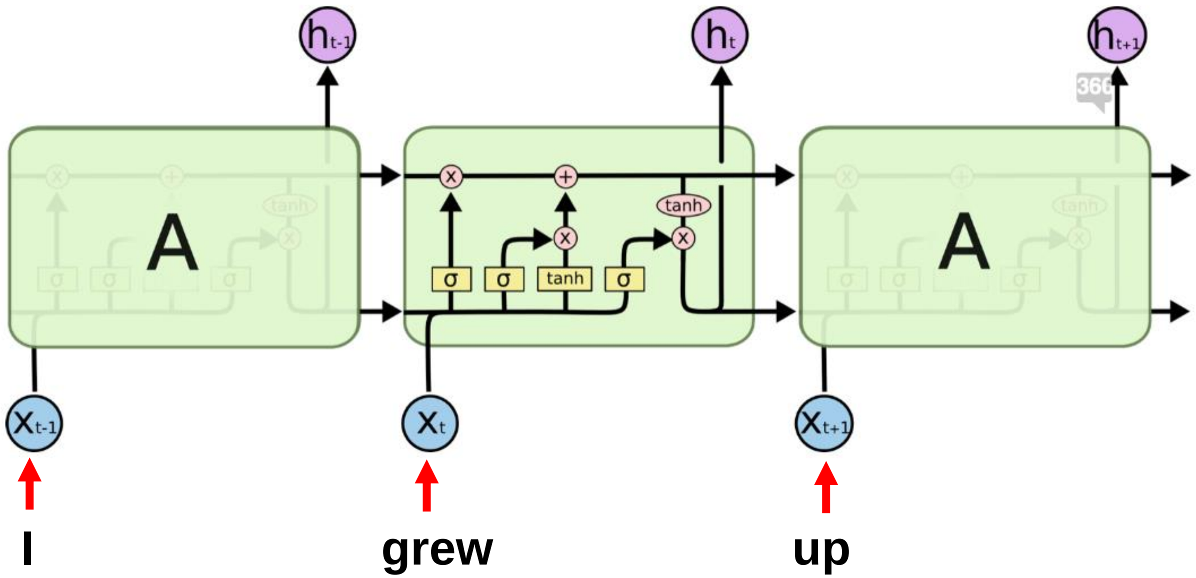
- Information is **added** or **removed** optionally through **gates**
- Example: Sigmoid activation function and multiplication (choose the amount of information to pass through)



Long Short-Term Memory (LSTM)

- Learn what to **remember** and what to **forget**

“I grew up in France. My name is Teddy. I live in Malaysia. I speak fluent French”



Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

Long Short-Term Memory (LSTM)

- **Cell state**
  - Contains information flow through the LSTM
  - Keeps track of relevant information

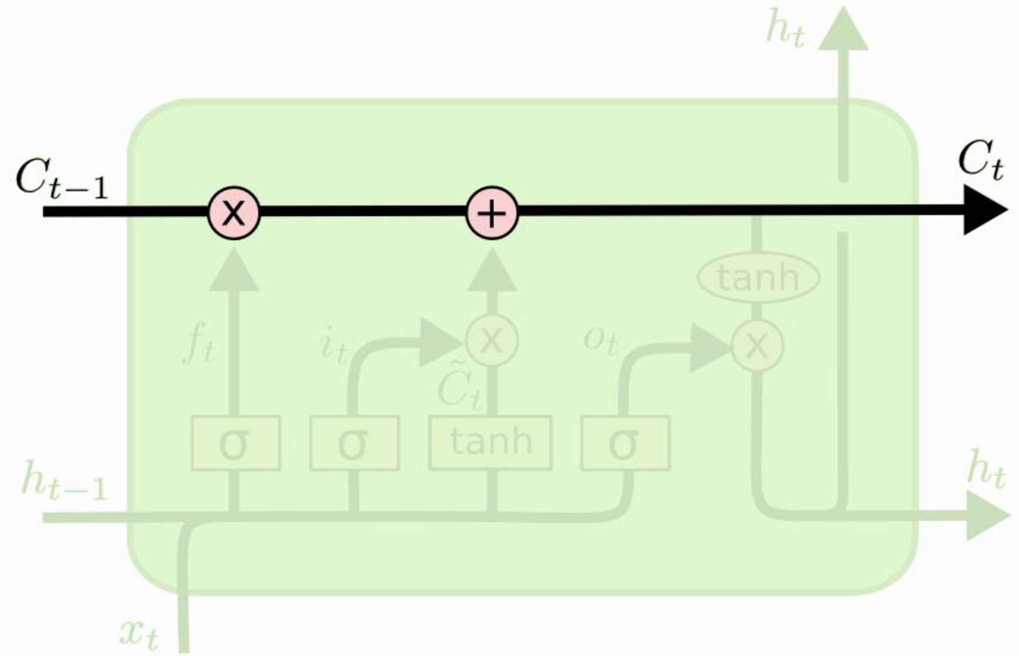
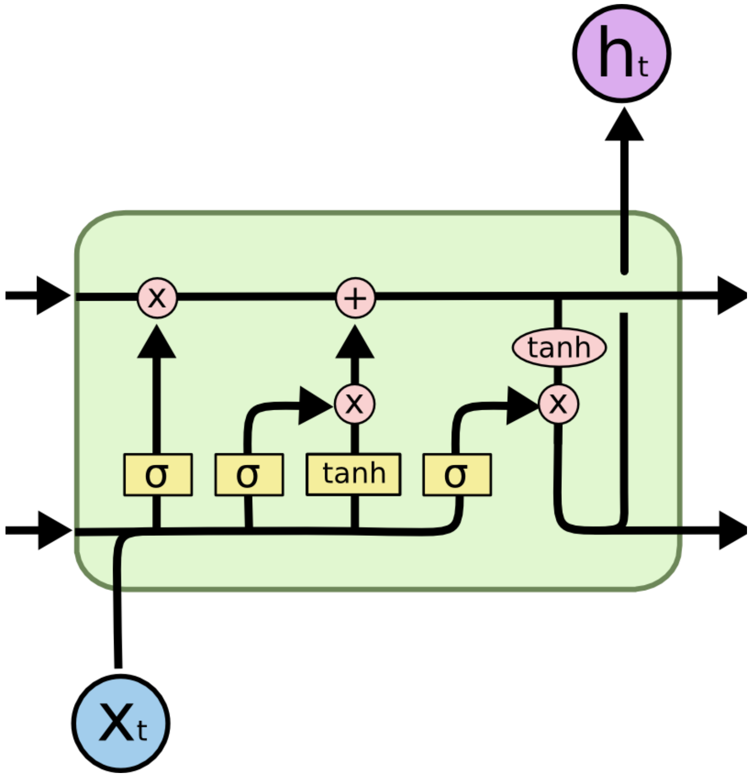


Image from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

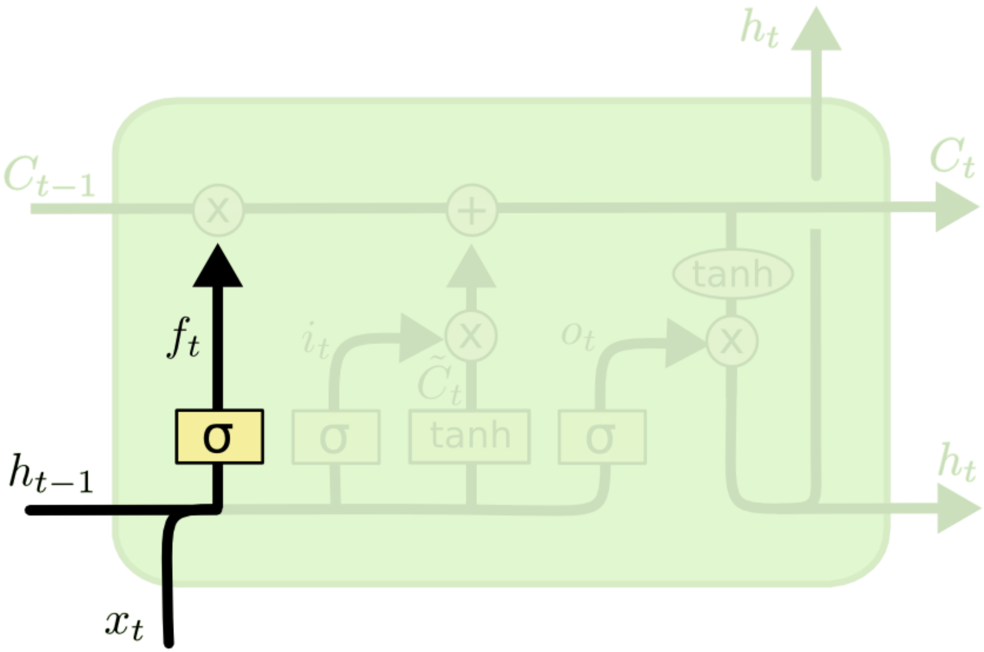
Long Short-Term Memory (LSTM)

- **Steps**
  1. Forget
  2. Store
  3. Update
  4. Output
- **Gates**
  - Forget
  - Input
  - Output



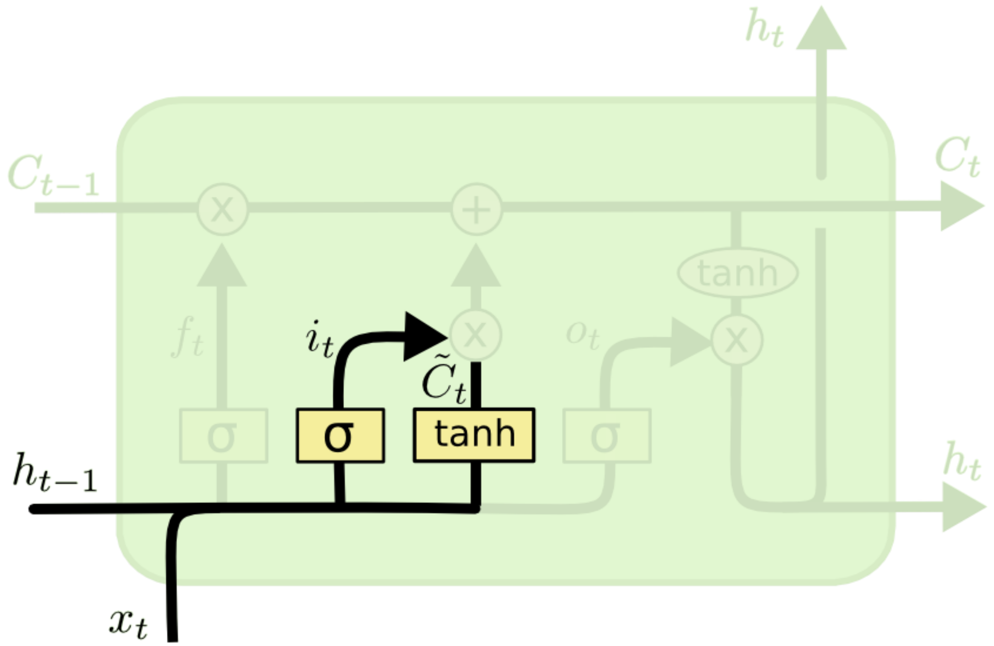
Long Short-Term Memory (LSTM)

- **Step 1: Forget**
  - Forget irrelevant parts of previous state
  - Forget gate



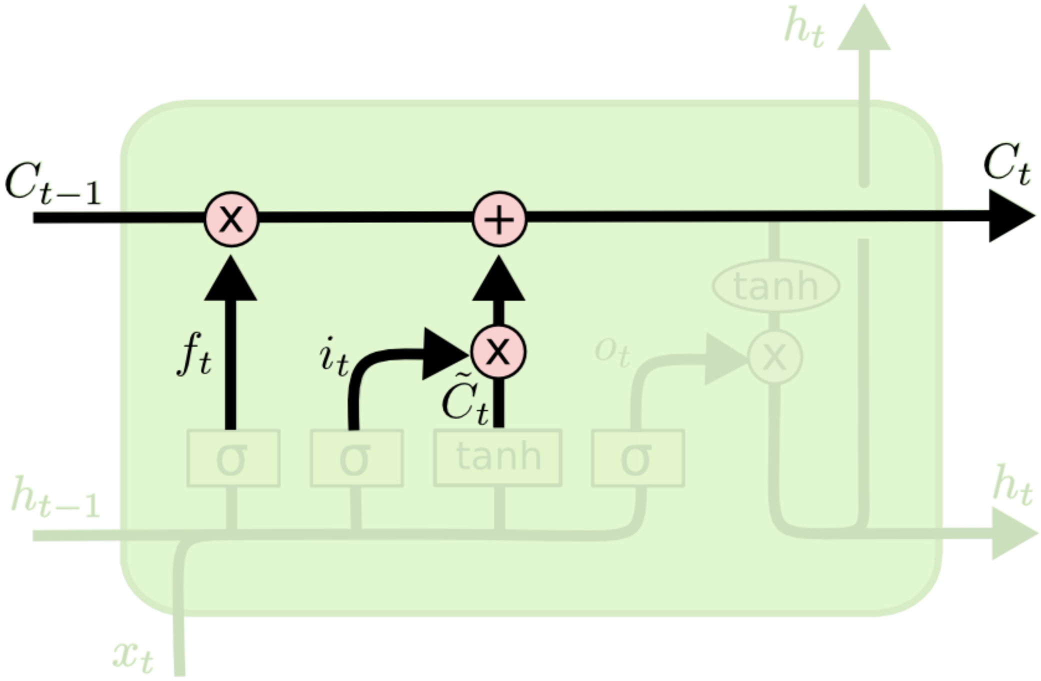
Long Short-Term Memory (LSTM)

- **Step 2: Store**
  - Store relevant new information to output state
  - Input gate



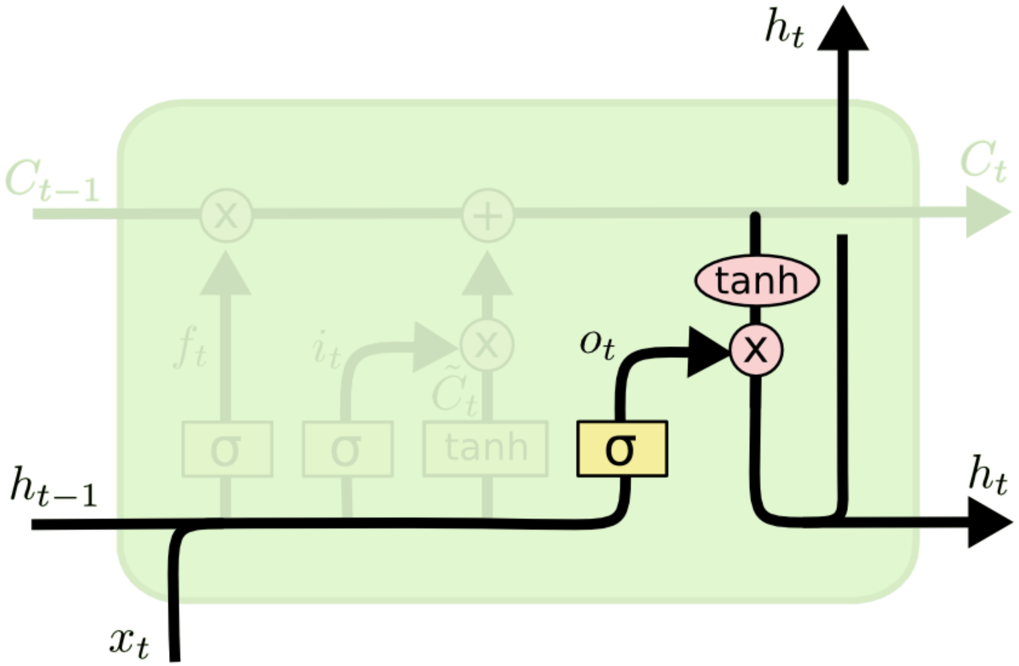
Long Short-Term Memory (LSTM)

- **Step 3: Update**
  - Update the old cell state to the new cell state



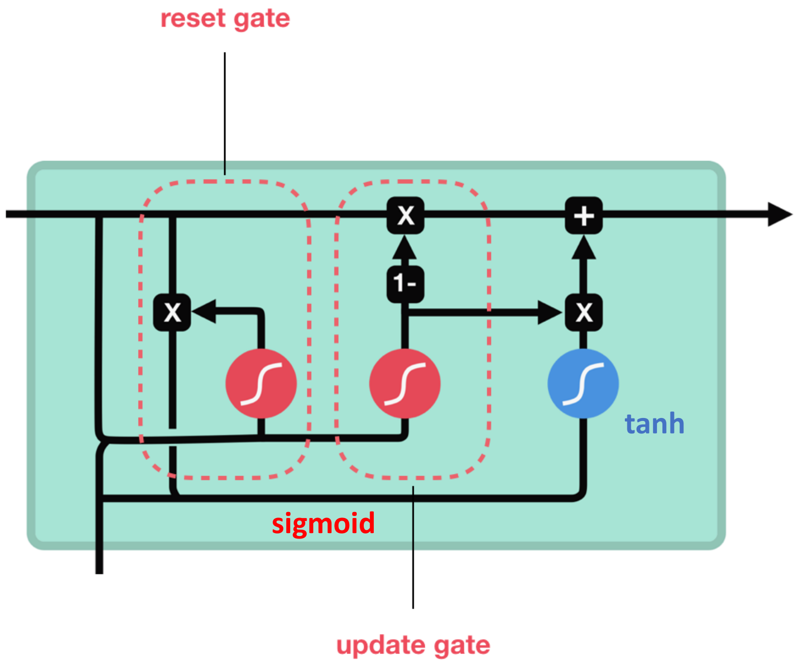
Long Short-Term Memory (LSTM)

- **Step 4: Output**
  - Choose what information is sent to the next time step
  - Output gate



Gated Recurrent Units (GRU)

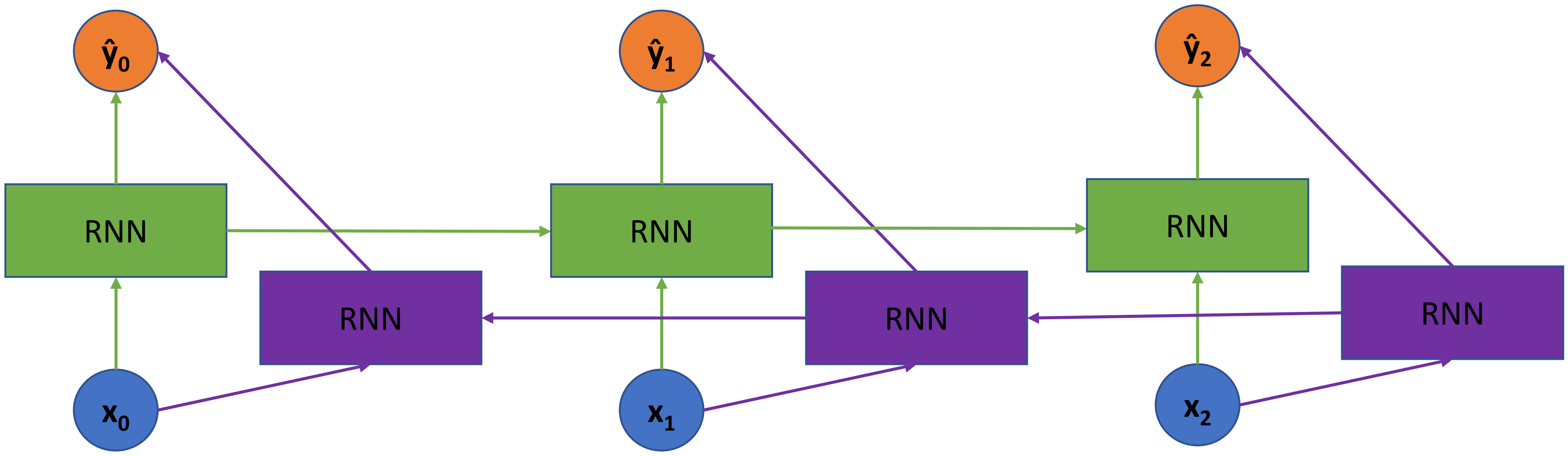
- **Remove cell state** and use hidden state to transfer information instead
- **Fewer** parameters compared to LSTM
- **Update Gate**
    - Acts like both Forget and Input gate of LSTM
    - Decides what information needs to be passed along and what information needs to be thrown away
- **Reset Gate**
    - Decide how much past information to forget



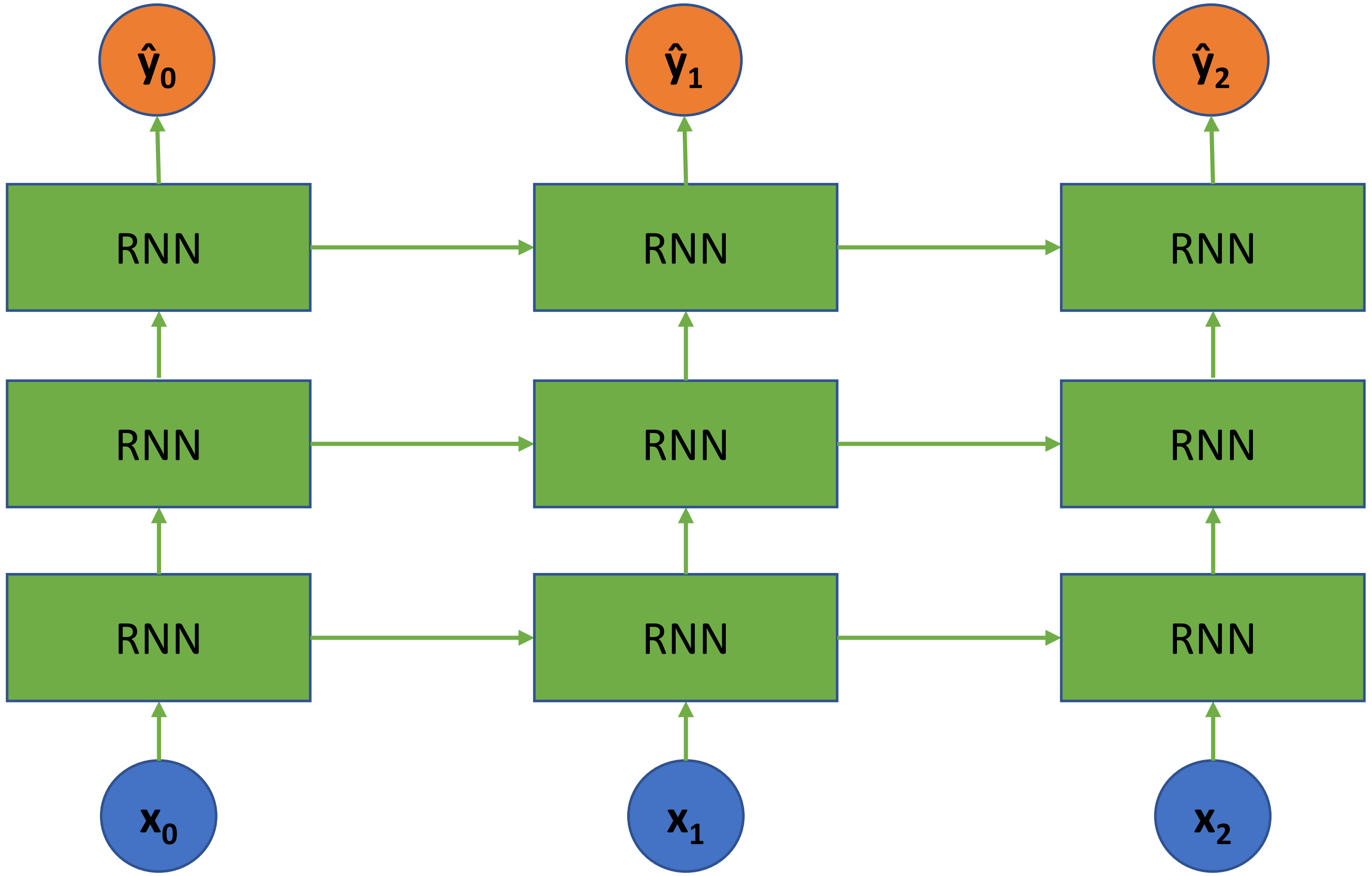
Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).

Bidirectional RNN

- **He said, “Teddy** bears are on sale!”
- **He said, “Teddy** Roosevelt was a great President!”
- Need to take information from the **future**

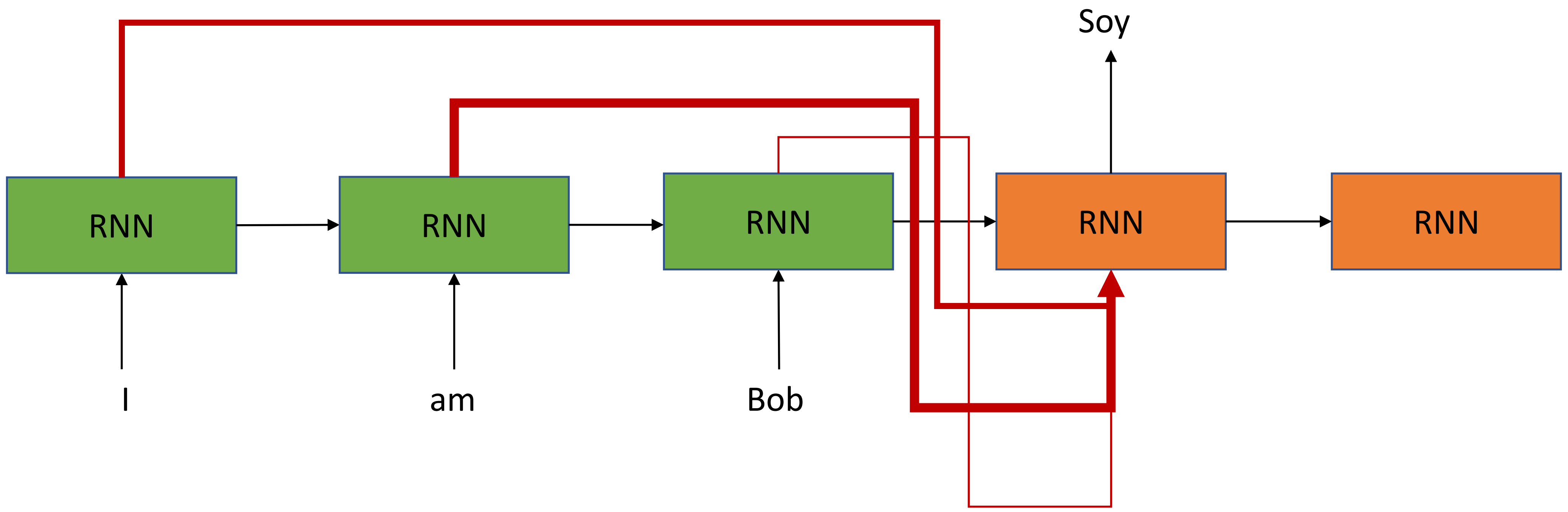


Deep RNN



Attention Mechanism

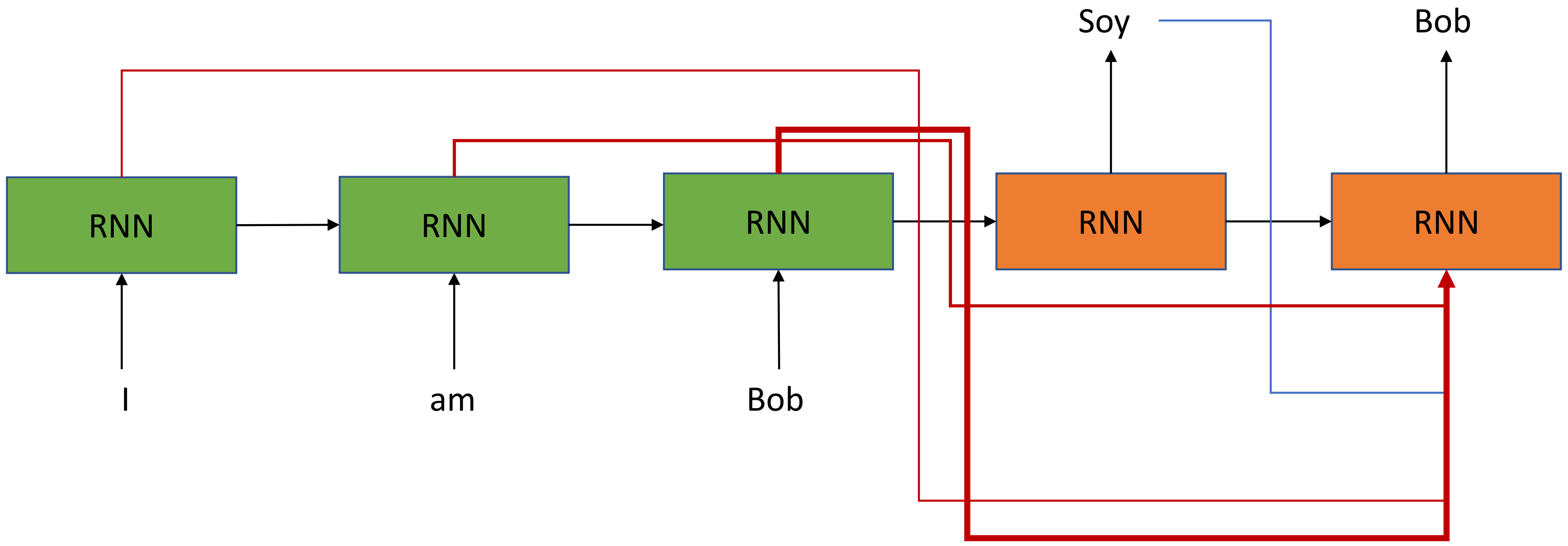
- Learn how much "attention" to give to specific input
- Reduce the necessity to memorize long sequences



Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

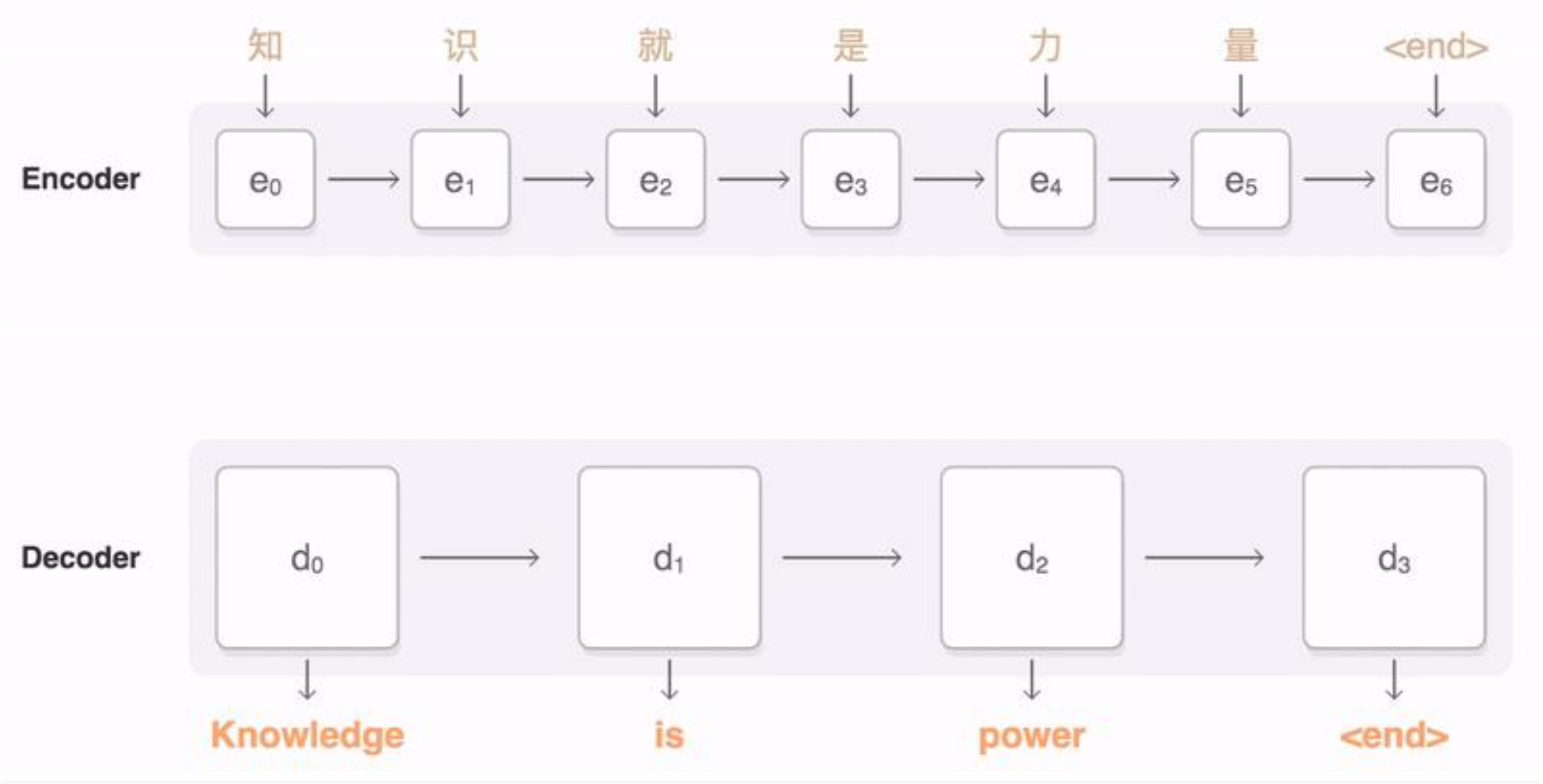
Attention Mechanism

- Learn how much "attention" to give to specific input
- Reduce the necessity to memorize long sequences



Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Attention Mechanism



Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Natural Language Processing: One-hot Vector

How to represent words for our network?

Vocabulary: $V = [a, aaron, apple, \ldots., zoo, < UNKNOWN >]$

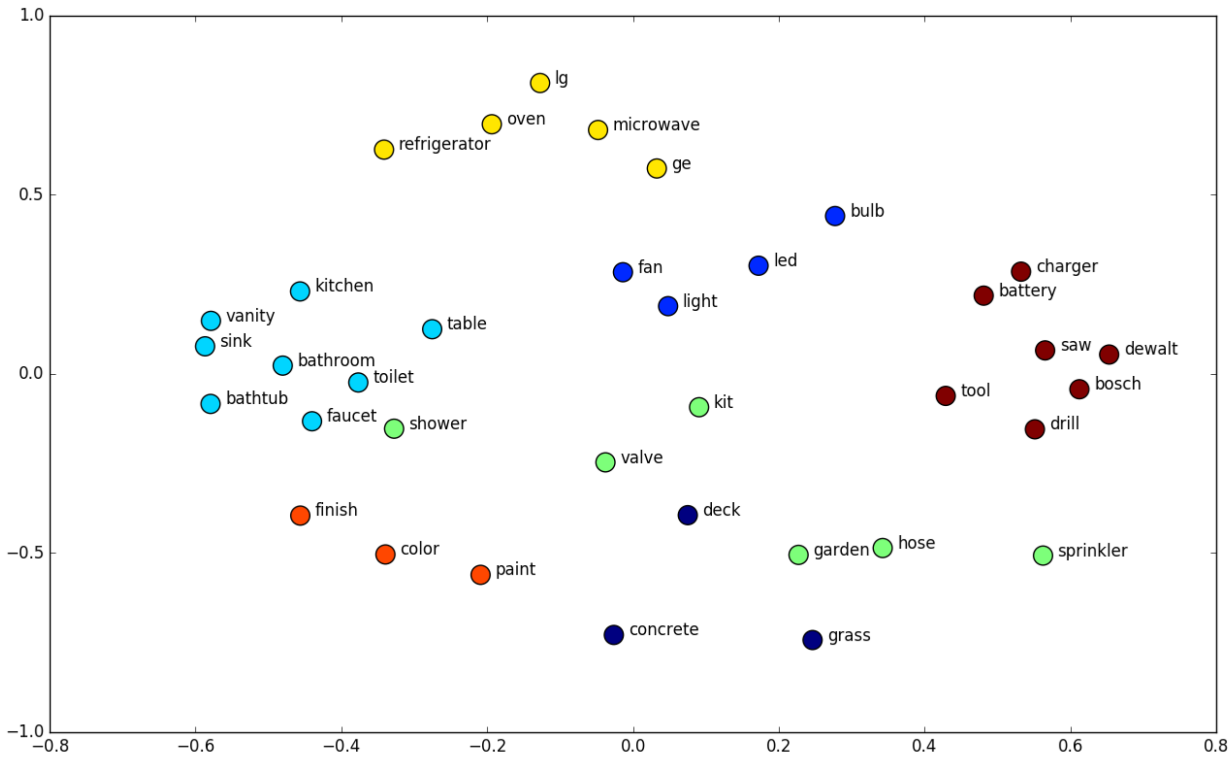
Weakness:

- Treats each word as independent entities with no relation
- If vocabulary is large, the length of vector with be large

$$\text{a} \rightarrow \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \\ 0 \end{bmatrix} \qquad \text{aaron} \rightarrow \begin{bmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 0 \\ 0 \end{bmatrix} \qquad \text{zoo} \rightarrow \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 0 \\ 1 \end{bmatrix}$$

Natural Language Processing: Word Embedding

- Find embeddings for each word such that similar words are "closer"
- For example we want words like "apple" and "orange" or "height" and "width" to be represented as more similar compared to other non-related words



Natural Language Processing Evaluation: BLEU score

- **B**i**L**ingual **E**valuation **U**nderstudy

Example on unigrams:

Reference 1: The cat is on the mat

Reference 2: There is a cat on the mat

Machine Translation Output: The cat the cat on the mat

Unigrams	Output Count	Max Ref Count
The	3	2
Cat	2	1
On	1	1
Mat	1	1
SUM	7	5

$$P_1 = \frac{\sum_{unigrams} Max\ Ref\ Count}{\sum_{unigrams} Output\ Count}$$

$$P_1 = \frac{5}{7}$$

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

Natural Language Processing Evaluation: BLEU score

Example on bigrams:

Reference 1: The cat is on the mat

Reference 2: There is a cat on the mat

Machine Translation Output: The cat the cat on the mat

$$P_2 = \frac{\sum_{bigrams} Max\ Ref\ Count}{\sum_{bigrams} Output\ Count}$$

Bigrams	Output Count	Max Ref Count
The cat	2	1
Cat the	1	0
Cat on	1	1
On The	1	1
The Mat	1	1
SUM	6	4

$$P_2 = \frac{4}{6}$$

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

Natural Language Processing Evaluation: BLEU score

Complete BLEU score

$$P_n = \frac{\sum_{n-grams} Max\ Ref\ Count}{\sum_{n-grams} Output\ Count}$$

$$\mathrm{BLEU} = \mathrm{BP} * \exp\left(\frac{1}{4}\sum_{n=1}^{4} P_n\right)$$

$$\mathrm{BP, Brevity\ Penalty} = \begin{cases} 1\ if\ len(Machine\ Translation\ Output) > len(Reference) \\ \exp\left(1 - \frac{len(Machine\ Translation\ Output)}{len(Reference)}\right) \quad otherwise \end{cases}$$

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

Questions?