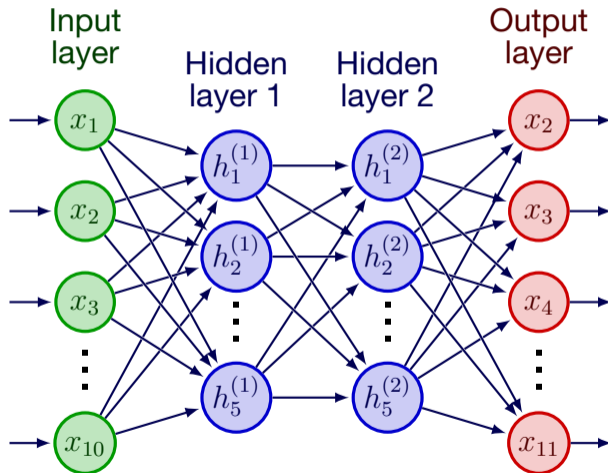


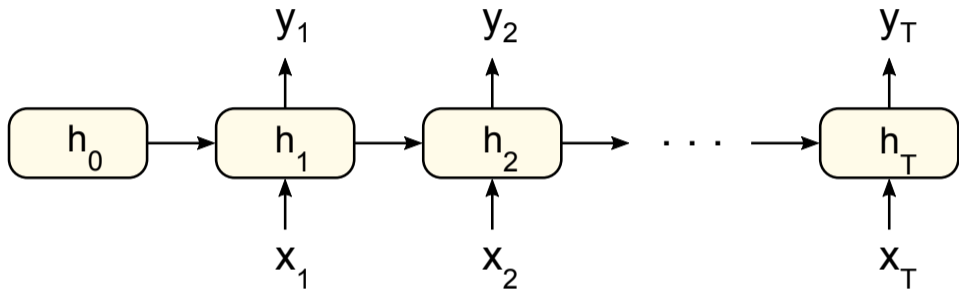
Recurrent Neural Network

One-step forecast



Problem: Past values depend on future values (e.g. x_2 depends on x_{10})

Recurrent Neural Network

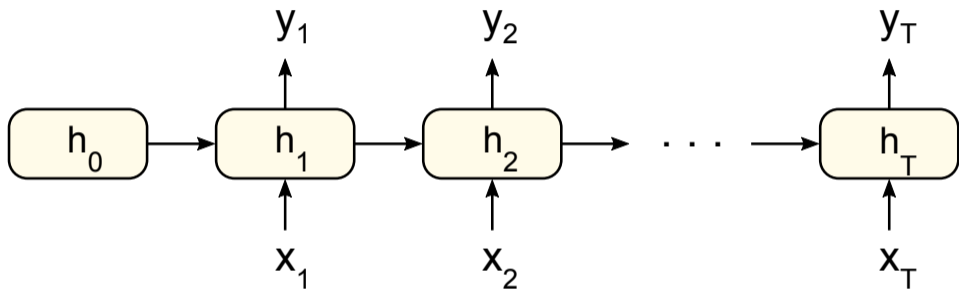


Inputs: x_1, x_2, \dots, x_T

Hidden states: h_1, h_2, \dots, h_T

Outputs: y_1, y_2, \dots, y_T

Recurrent Neural Network

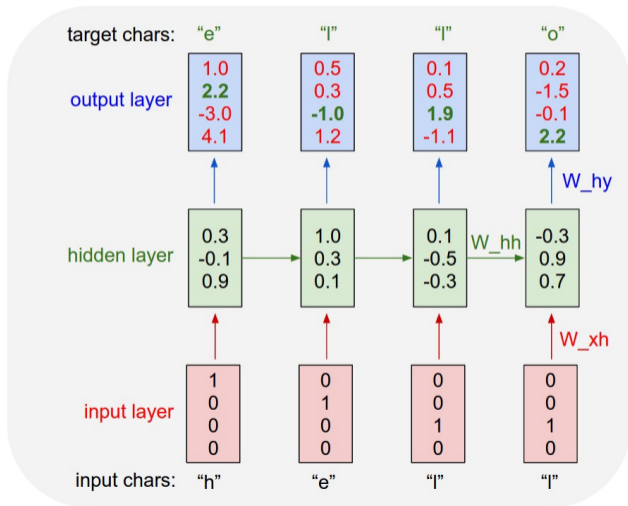


$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1})$$

$$y_t = W_{hy}h_t.$$

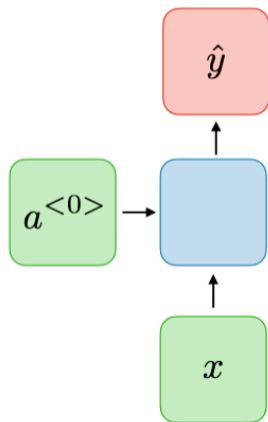
Character-level language model

- Goal: predicting the next character.
- Use the next characters as the target.



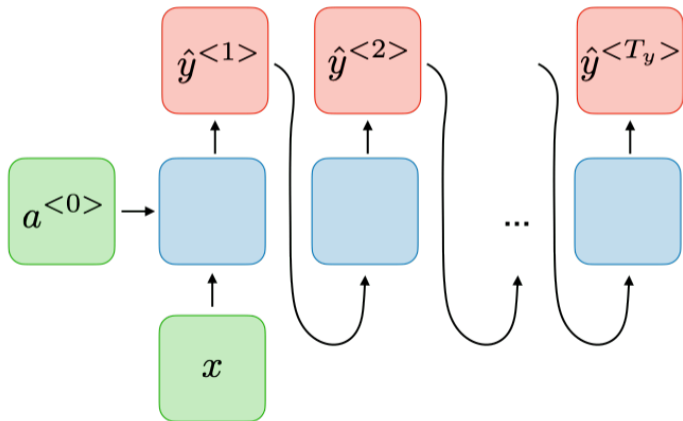
Types of RNN

One-to-one ($T_x = T_y = 1$)



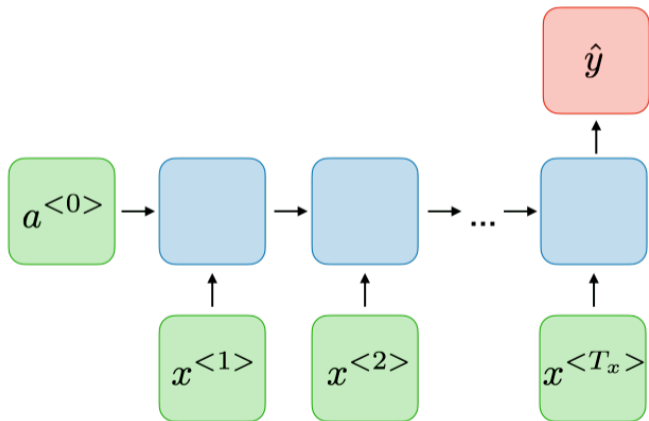
Types of RNN

One-to-many ($T_x = 1, T_y > 1$)



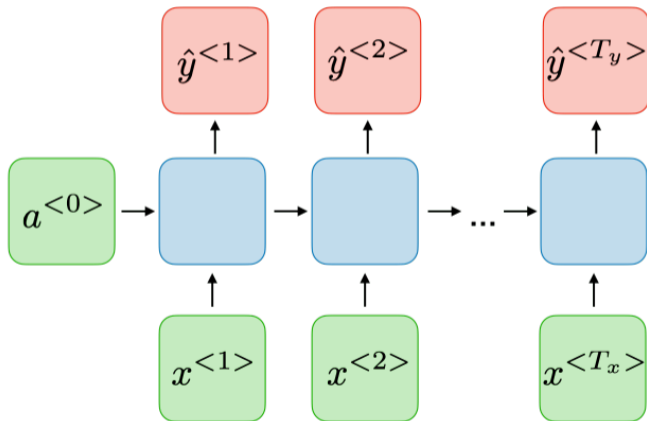
Types of RNN

Many-to-one ($T_x > 1, T_y = 1$)



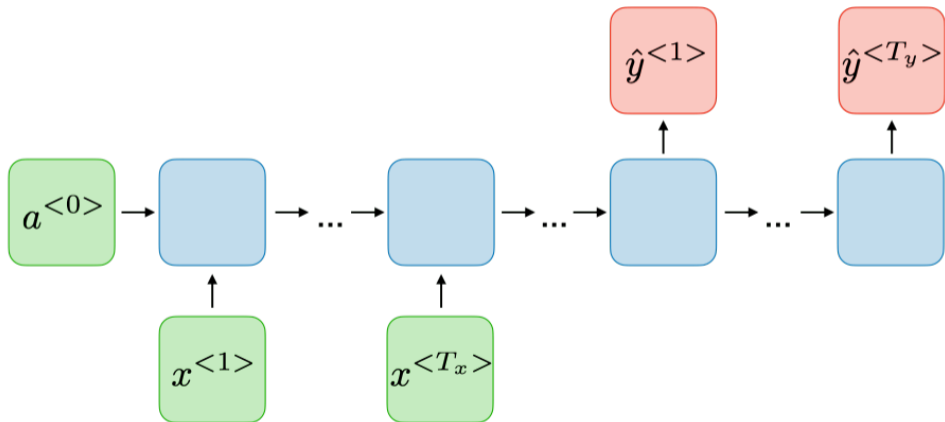
Types of RNN

Many-to-many ($T_x = T_y$)

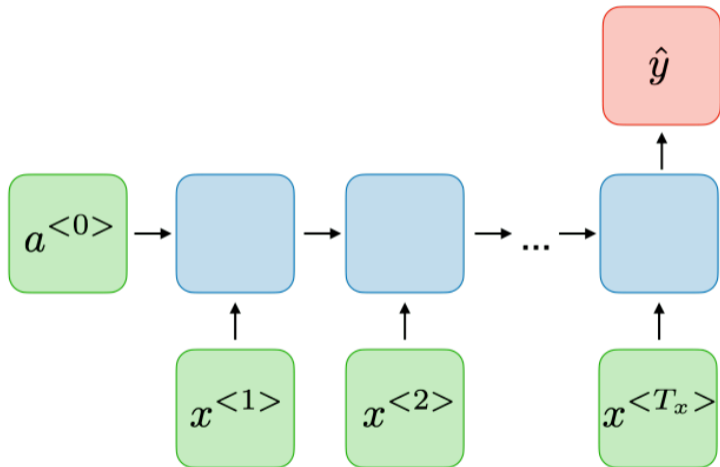


Types of RNN

Many-to-many ($T_x \neq T_y$)

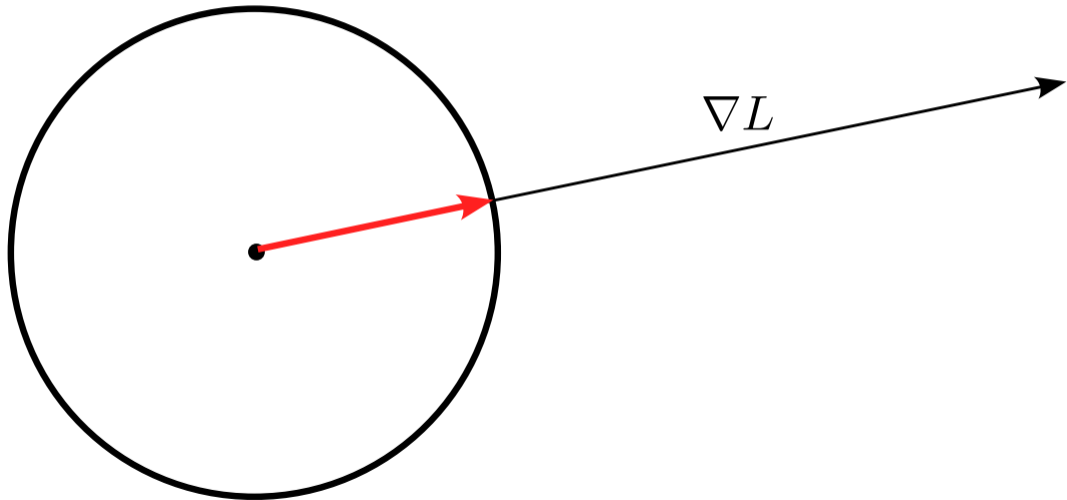


Recurrent Neural Network



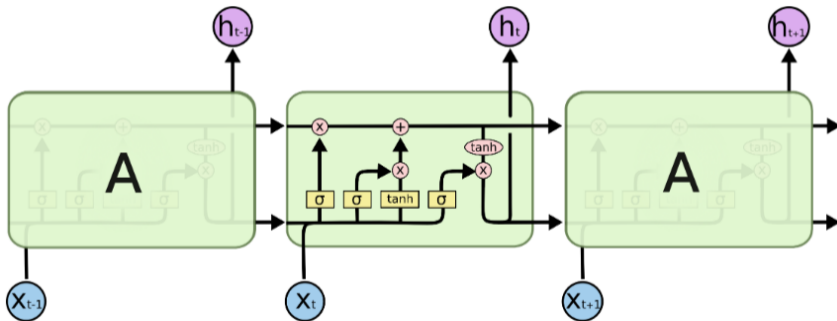
However, there's gradient vanishing/exploding problem.

Gradient clipping

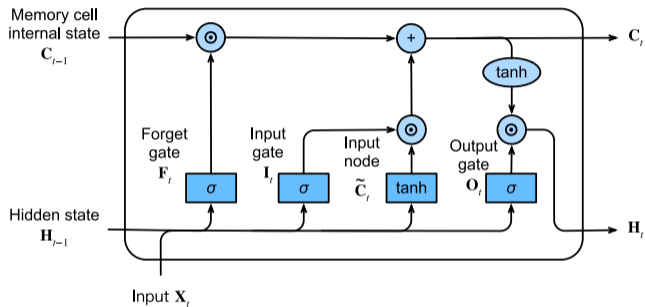


solves gradient exploding

Long-short term memory (LSTM)



Long-short term memory (LSTM)



FC layer with activation function



Elementwise operator



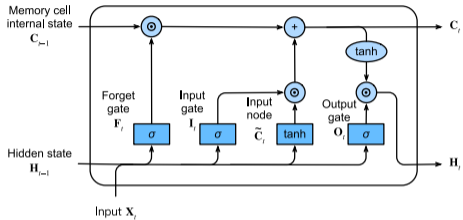
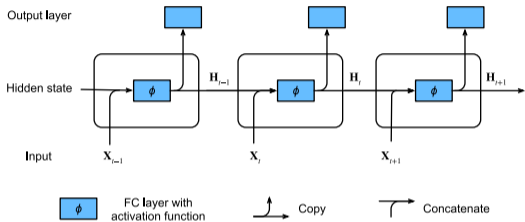
Copy



Concatenate

- **Hidden state** h_t and **Cell state** c_t
- h_t is also the output.

RNN vs LSTM



Text generation with LSTM

100 iterations

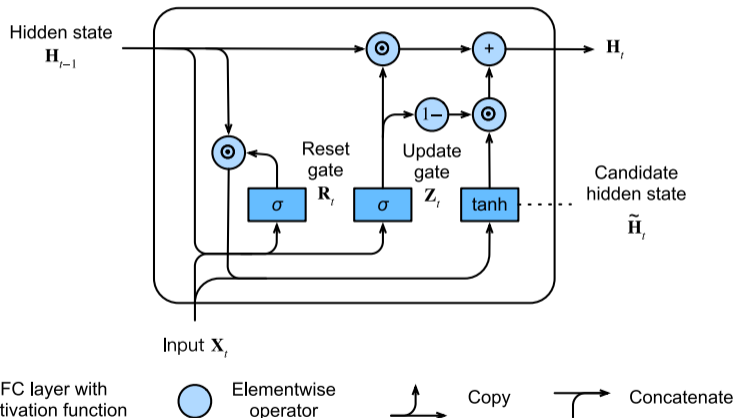
tyntd-iafhatawiaoihrdemot lytdws e ,tfti,
astai f ogoh eoase rrranbyne 'nhthnee
e plia tklrqd t o idoe ns,smtt h ne etie
h,hregtrs nigtike,aoaenns lng

Text generation

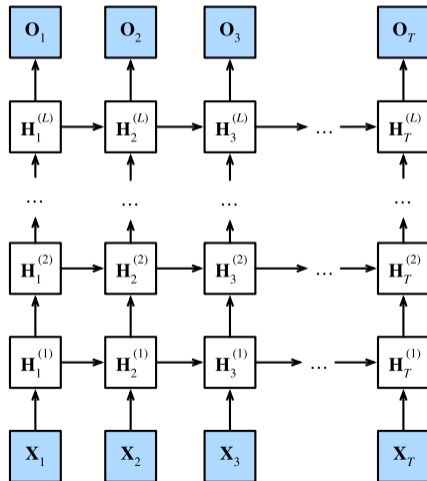
2000 iterations

“Why do what that day,” replied Natasha, and wishing to himself the fact the princess, Princess Mary was easier, fed in had oftended him. Pierre aking his soul came to the packs and drove up his father-in-law women.

Gated Recurrent Unit (GRU)



Deep RNN



Filling in the blank

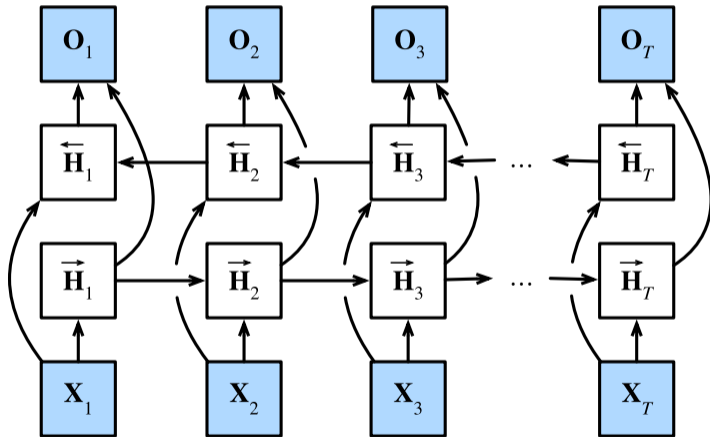
I am _____.

I am _____ hungry at all.

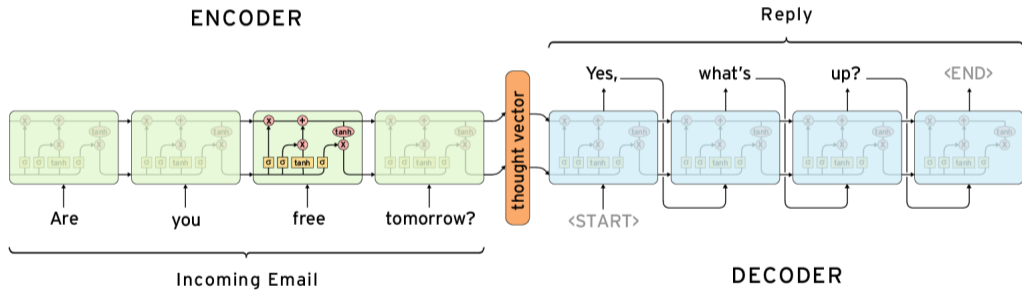
I am _____ hungry, and I can eat a horse.

The missing word heavily depends on the words that come after

Bidirectional RNN



Encoder-Decoder Seq2Seq



Sentence padding

Suppose RNN encoder has 7 hidden units, RNN decoder has 6 hidden units

How are we going to split the following sentences?

```
input = 'hello, how are you'  
output = 'i am fine'
```

Sentence padding

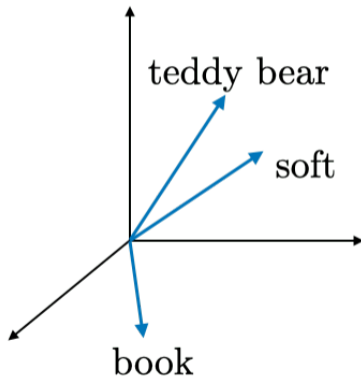
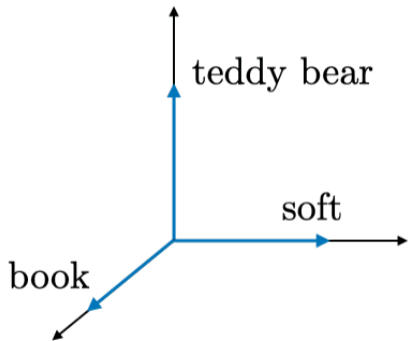
Suppose RNN encoder has 7 hidden units, RNN decoder has 6 hidden units

How are we going to split the following sentences?

```
input = 'hello, how are you'  
output = 'i am fine'
```

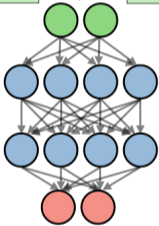
```
encoder_input = ['hello', 'how', 'are', 'you', '<EOS>', '<PAD>', '<PAD>']  
decoder_input = ['<START>', 'i', 'am', 'fine', '<EOS>', '<PAD>']  
output = ['i', 'am', 'fine', '<EOS>', '<PAD>', '<PAD>']
```


Word representation

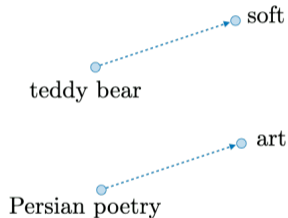
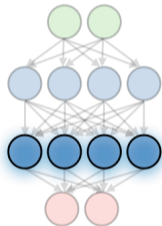


Word2vec (2013)

...A cute teddy bear is reading...



...A cute teddy bear is reading...



Train network on proxy task



Extract high-level representation



Compute word embeddings