

Machine translation

sequence-text English



sequence-text in Spanish

Introduction to Deep Learning

22. Encoder-Decoder, Seq2seq

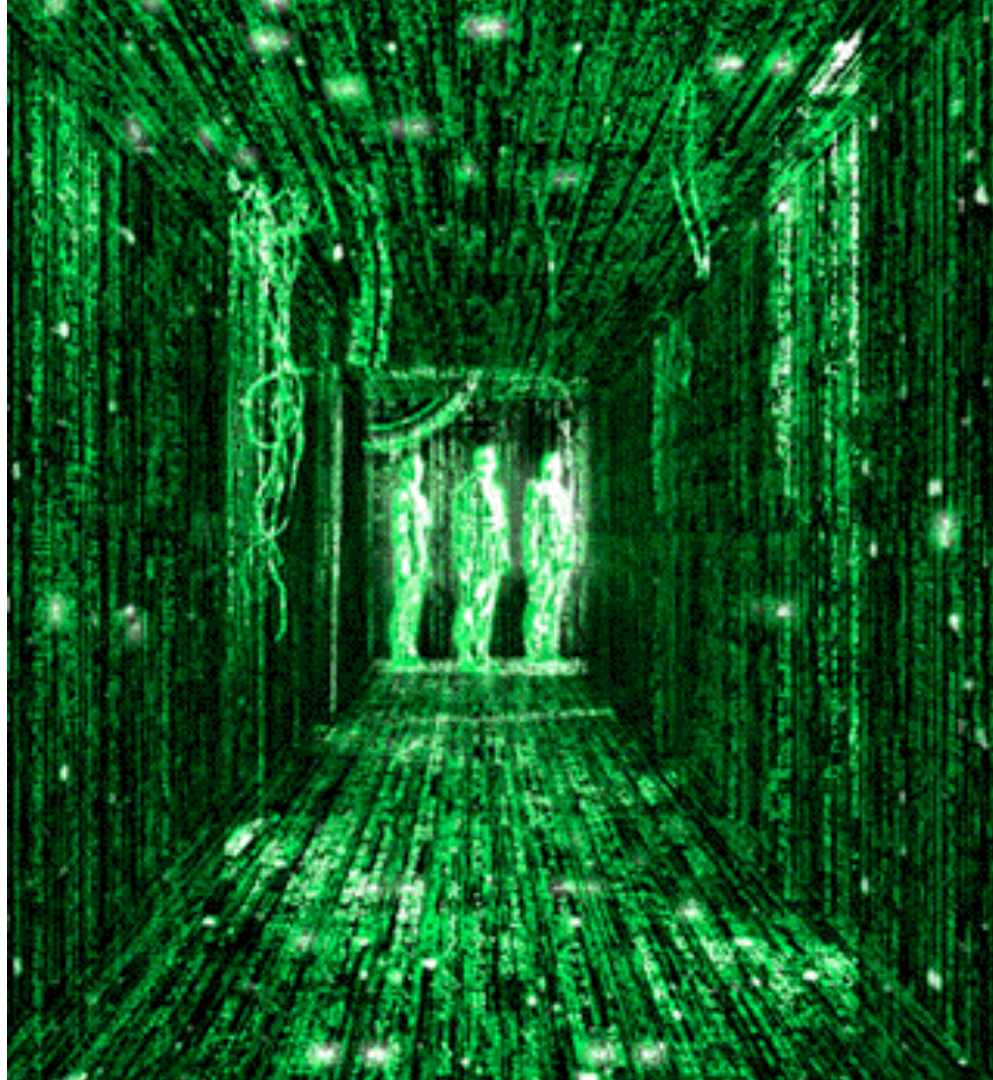
STAT 157, Spring 2019, UC Berkeley

Alex Smola and Mu Li

courses.d2l.ai/berkeley-stat-157

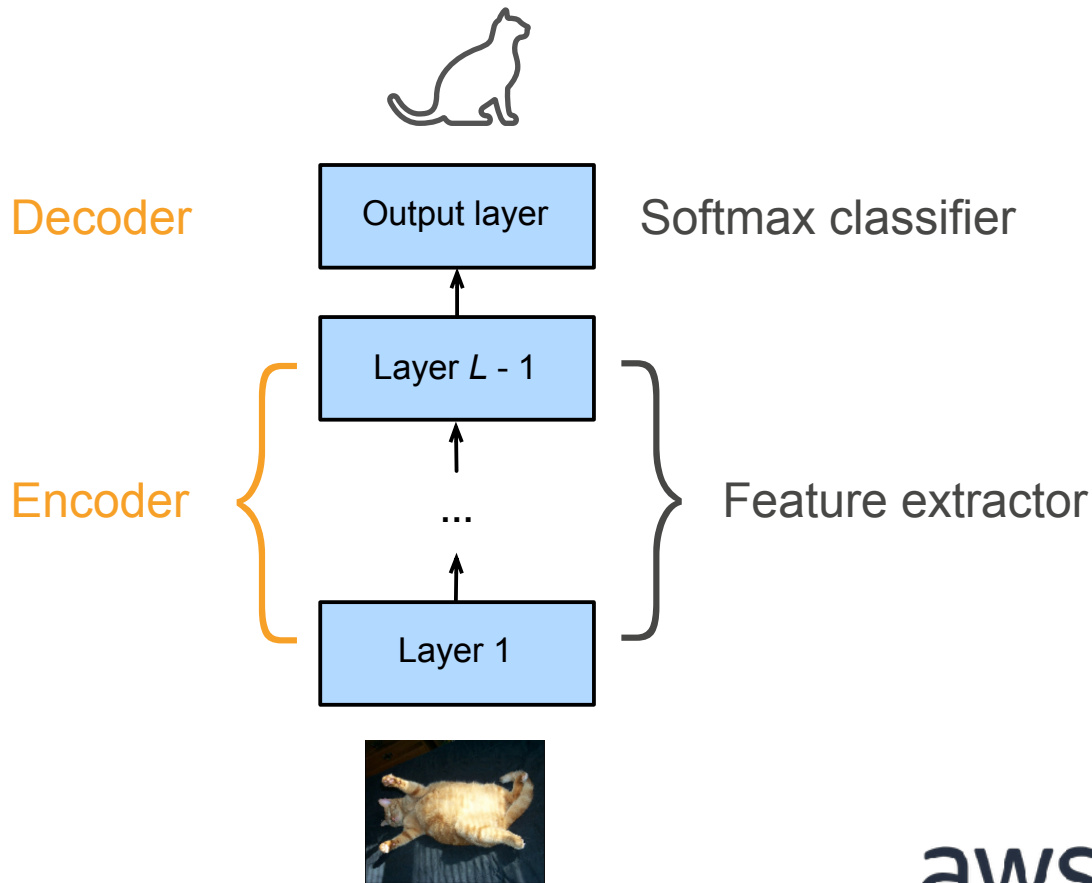


Encoder-Decoder



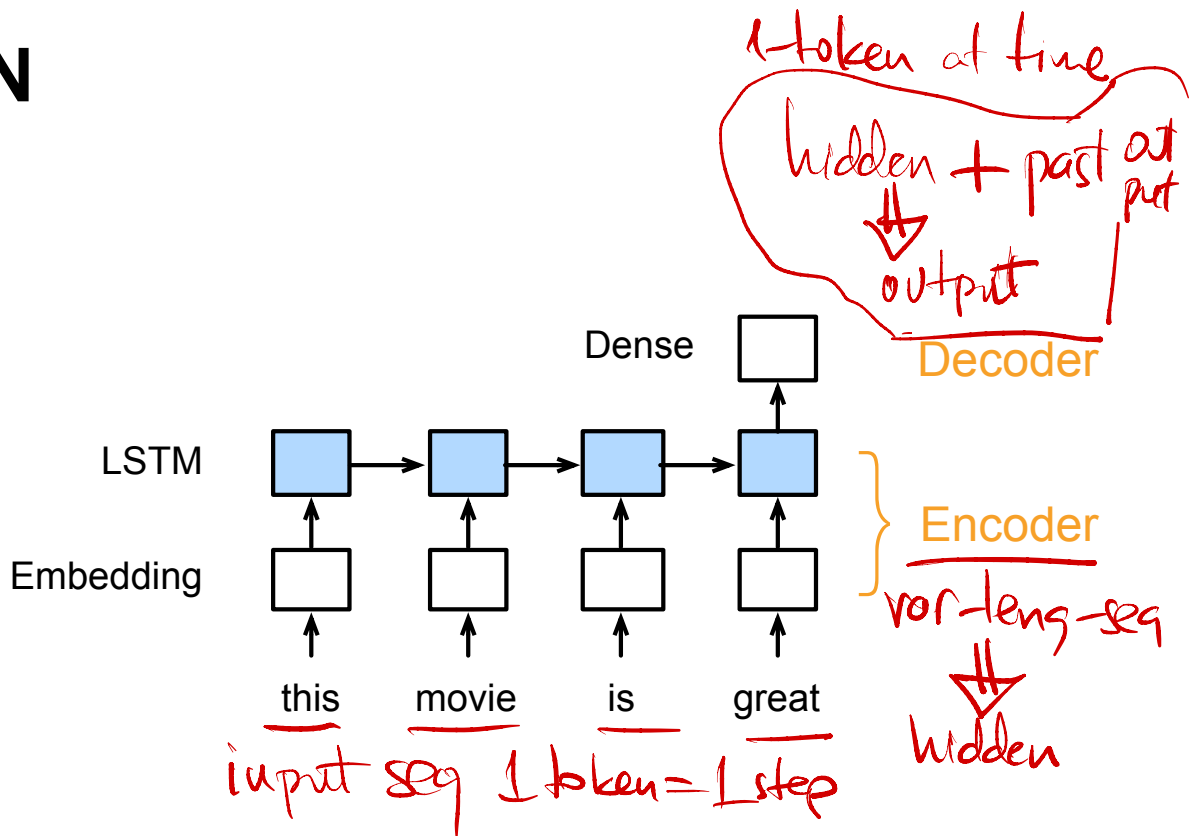
Rethink about CNN

- Encoder: encode inputs into intermediate presentation (features)
- Decoder: decode the presentation into outputs



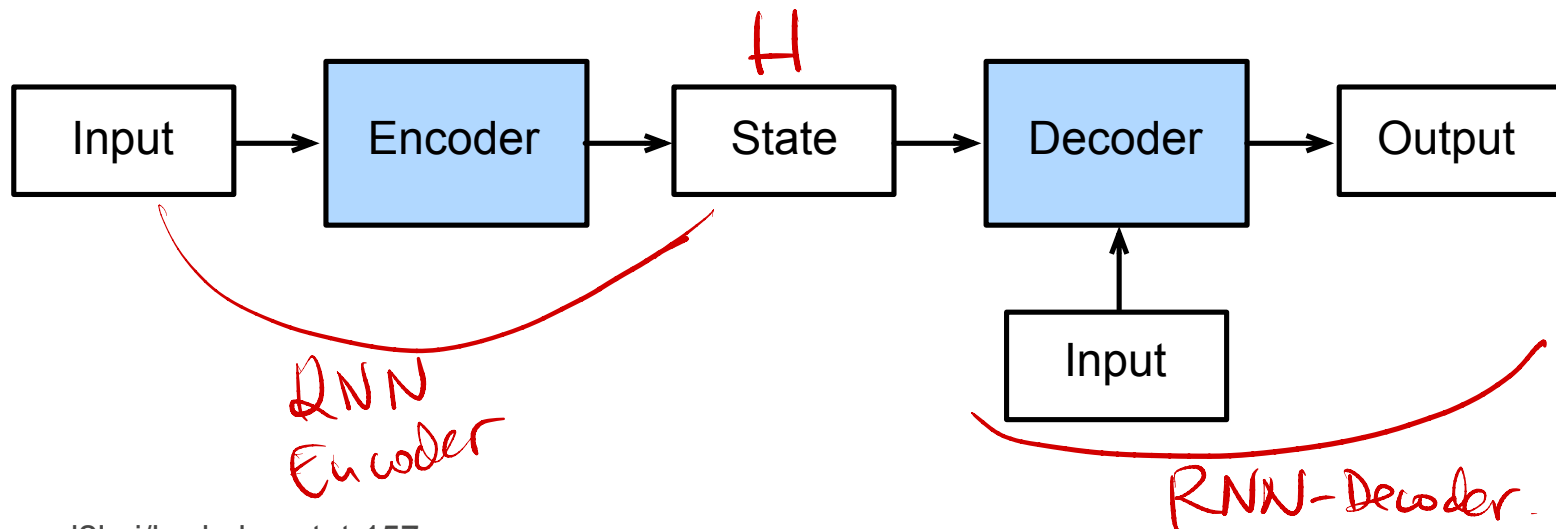
Rethink about RNN

- Encoder: present a piece of text as a vector
- Decoder: decode the presentation into outputs



The Encoder-decoder Architecture

- A model is partitioned into two parts
 - The encoder process inputs
 - The decoder generates outputs



The Base Class for an Encoder

```
class Encoder(nn.Block):  
    def __init__(self, **kwargs):  
        super(Encoder, self).__init__(**kwargs)  
  
    def forward(self, X):  
        raise NotImplementedError
```

The Base Class for a Decoder

- Create state with the encoder outputs and any other infos

```
class Decoder(nn.Block):  
    def __init__(self, **kwargs):  
        super(Decoder, self).__init__(**kwargs)  
  
    def init_state(self, enc_outputs, *args):  
        raise NotImplementedError  
  
    def forward(self, X, state):  
        raise NotImplementedError
```

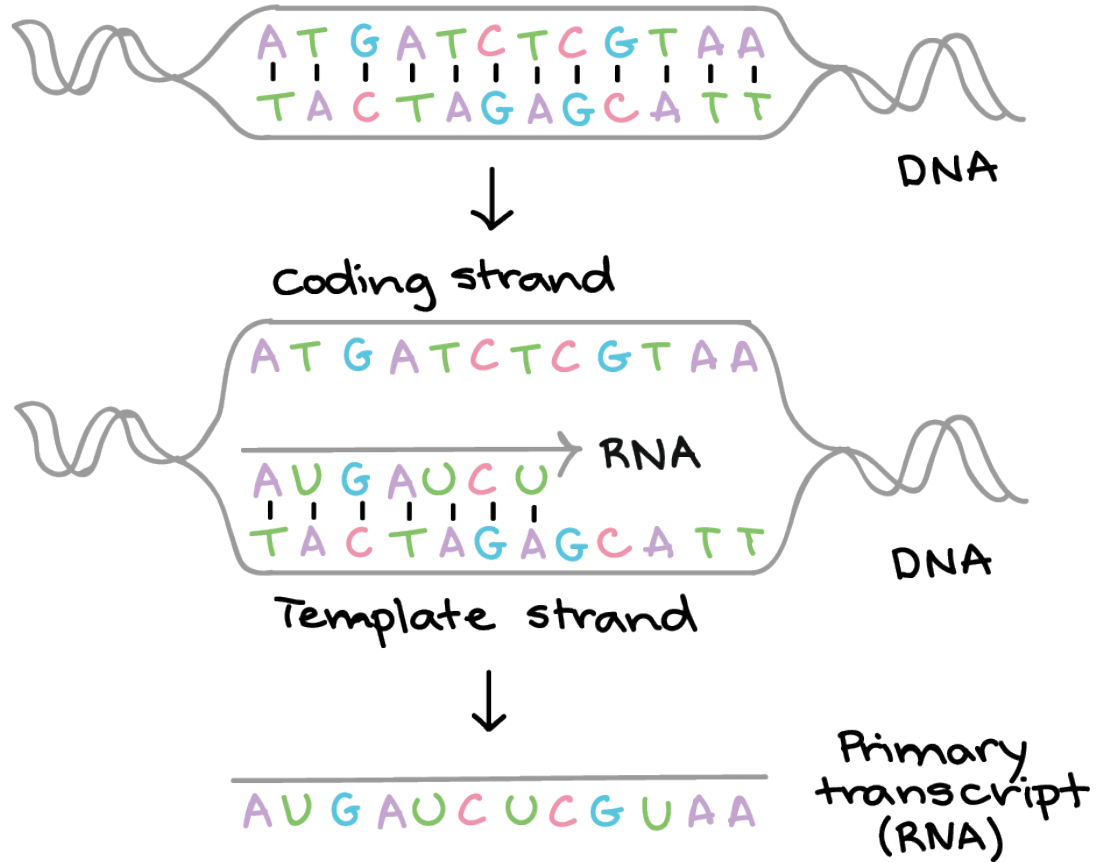
prev
state



The Base Class of the model

```
class EncoderDecoder(nn.Block):  
    def __init__(self, encoder, decoder, **kwargs):  
        super(EncoderDecoder, self).__init__(**kwargs)  
        self.encoder = encoder  
        self.decoder = decoder  
  
    def forward(self, enc_X, dec_X, *args):  
        enc_outputs = self.encoder(enc_X)  
        dec_state = self.decoder.init_state(enc_outputs, *args)  
        return self.decoder(dec_X, dec_state)
```

Seq2seq



Machine Translation

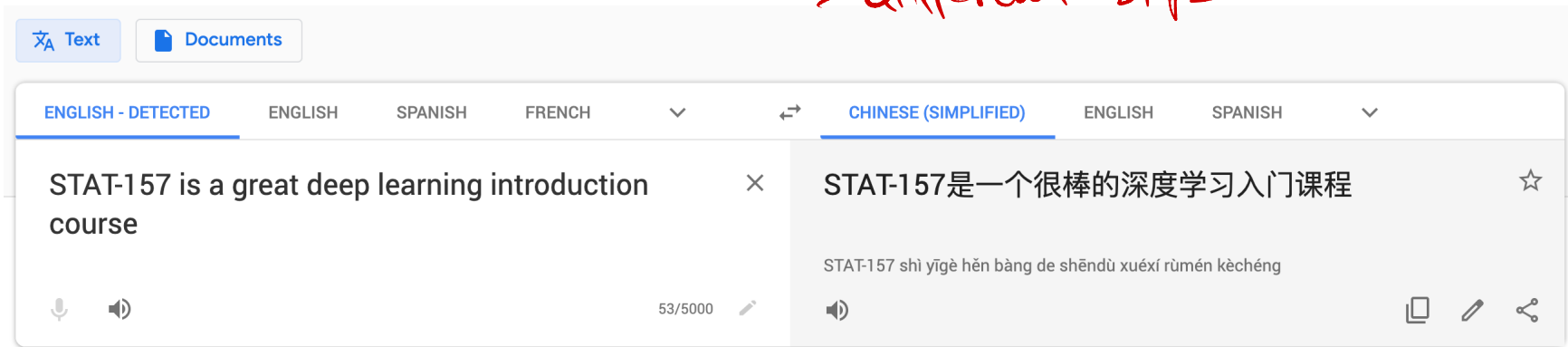
need processing of translation seq

- truncation
- padding
- tokenization
- vocabulary

HW6: done as given.

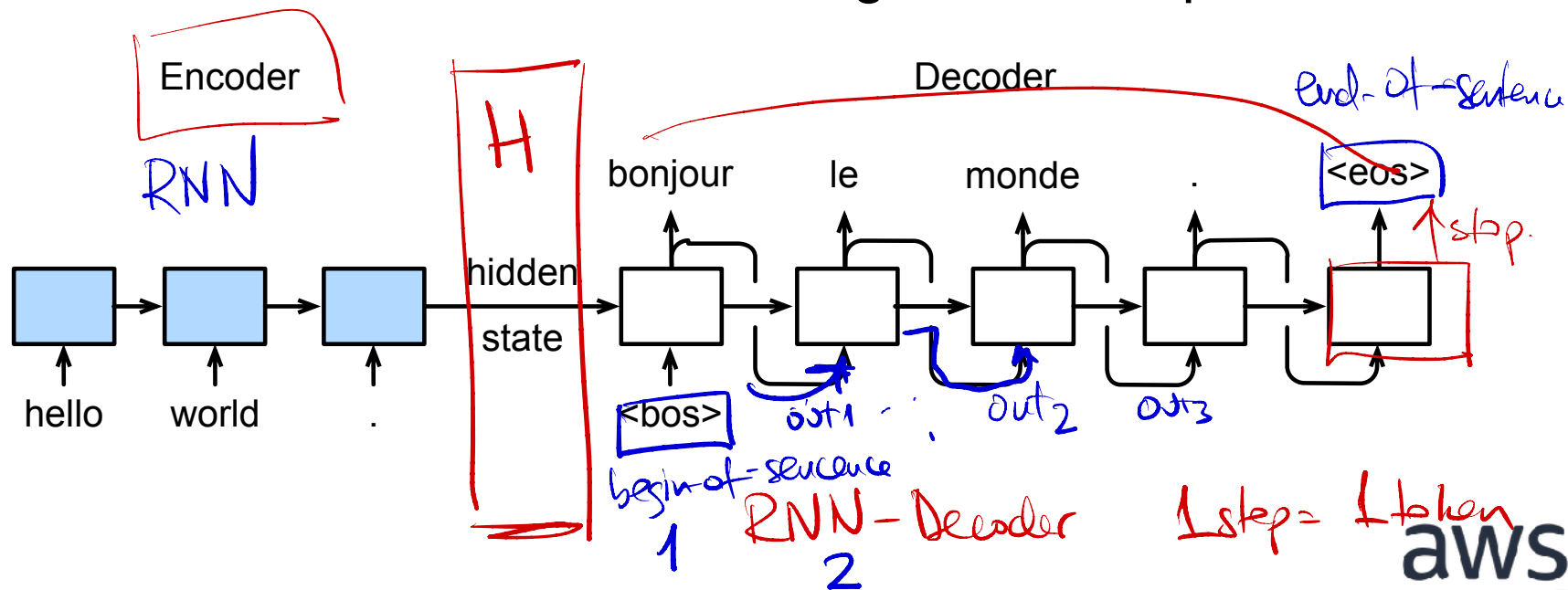
- Given a sentence in a source language, translate into a target language
- These two sequences may have different lengths

different steps



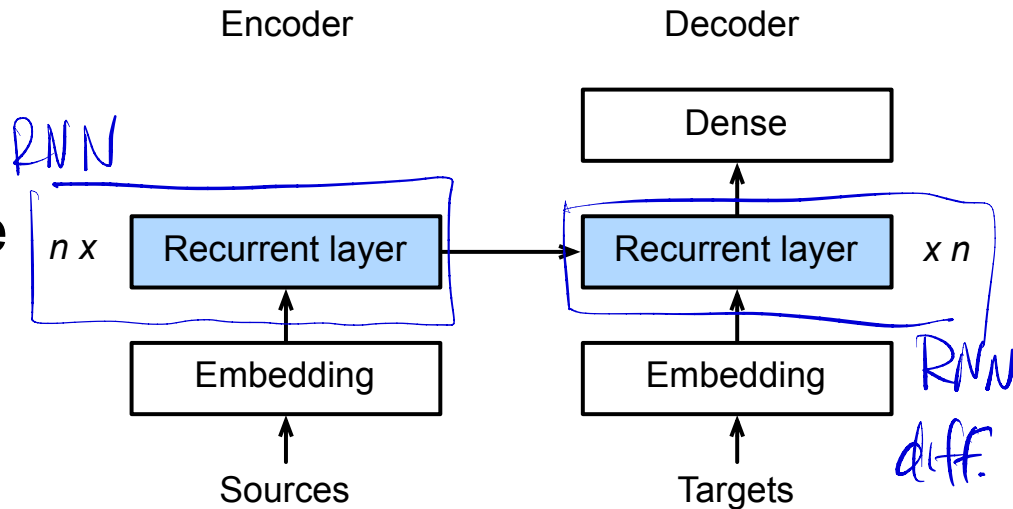
Seq2seq

- The encoder is a RNN to read input sequence
- The decoder uses another RNN to generate output



Encoder/Decoder Details

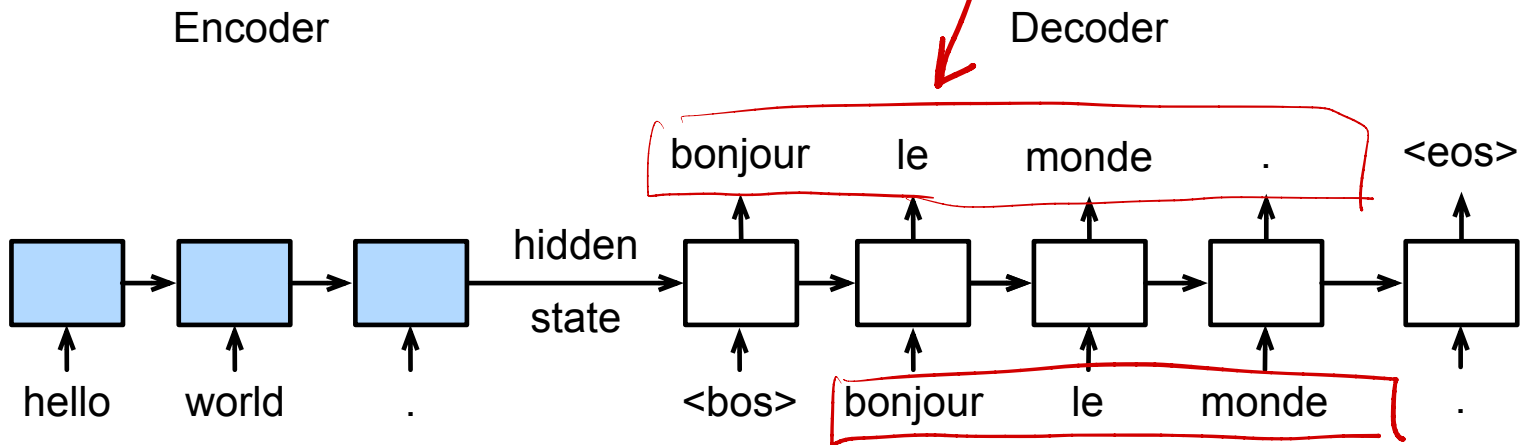
- The encoder is a standard RNN model without the output layer
- The encoder's hidden state in last time step is used as the decoder's initial hidden state



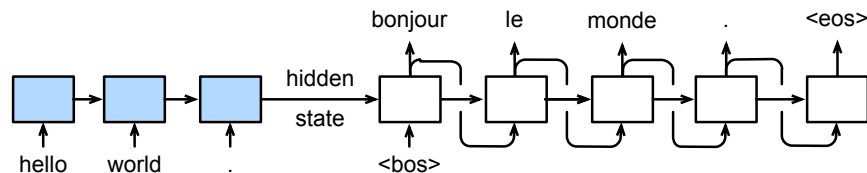
Training

"teacher forcing"

- The decoder is feed with the **targeted sentence** during training



Predict:



*shift by one
1 step = 1 token*