Introduction to Neural Machine Translation

Alexander Fraser

Slides from **Minh-Thang Luong, Stanford**

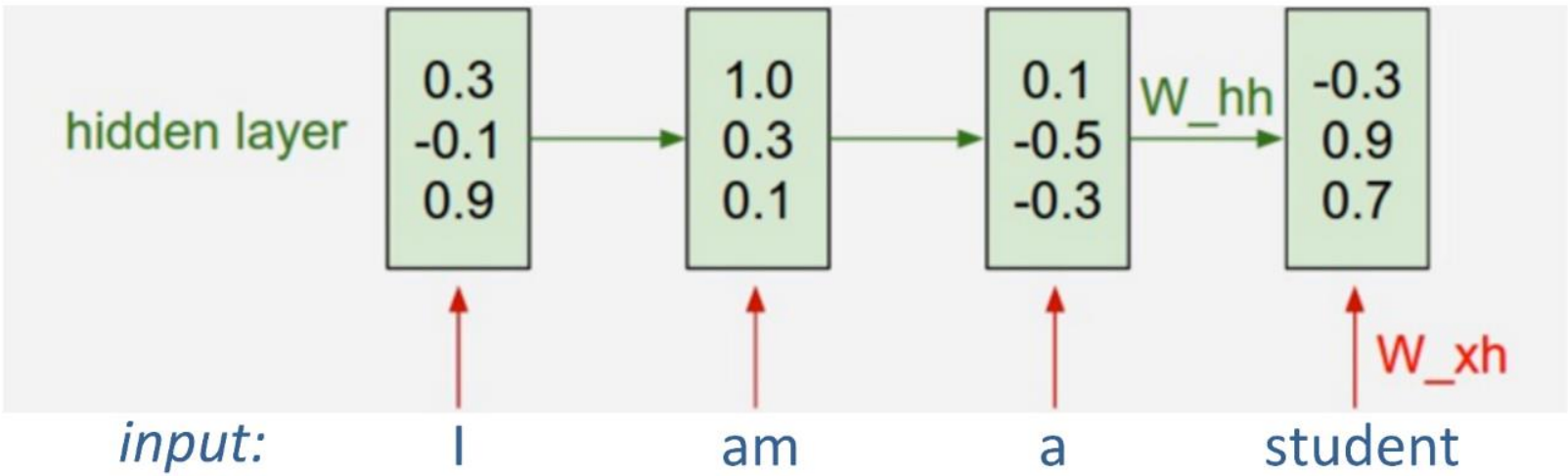
CS224d Lecture 15, 2016

(The longer lecture these slides are from can be found online, it is very worth reading!)

Neural Machine Translation (NMT)

- Neural machine translation is a relatively new technology which was developed from neural language modeling
- N-gram language models try to predict the next word given n-1 previous words
  - Often estimated using simple frequencies over a corpus
  - Example: if we have seen „he said“, could the next word be „no“?
  - Conditional probability: p(no|he said)
  - Large N does not work well because of sparsity
- Neural language models use recurrent neural networks
  - By using continuous valued representations we can condition on the whole sentence

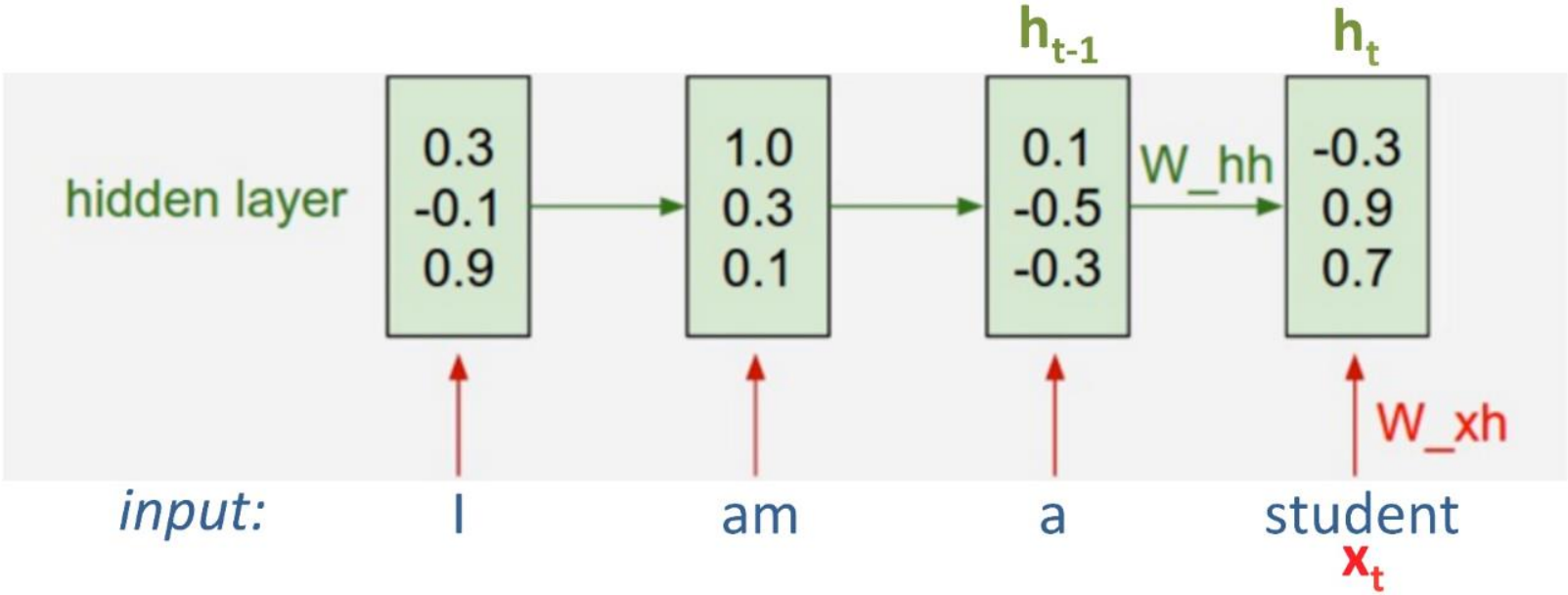
Recurrent Neural Networks (RNNs)



(Picture adapted from Andrej Karparthy)

Recurrent Neural Networks (RNNs)

$$\boldsymbol{h}_t = \sigma\left(\boldsymbol{W}_{\boldsymbol{xh}}\boldsymbol{x}_t + \boldsymbol{W}_{\boldsymbol{hh}}\boldsymbol{h}_{t-1}\right)$$



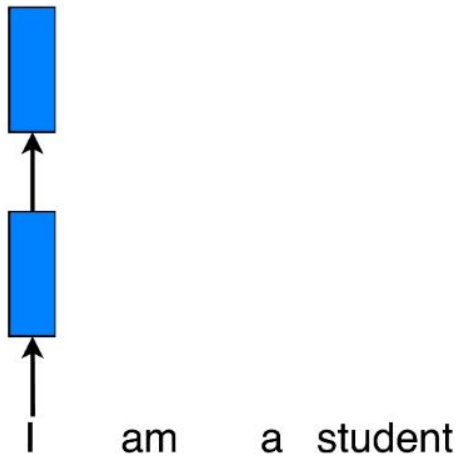
RNNs to represent sequences!

(Picture adapted from Andrej Karparthy)

Neural Machine Translation (NMT)

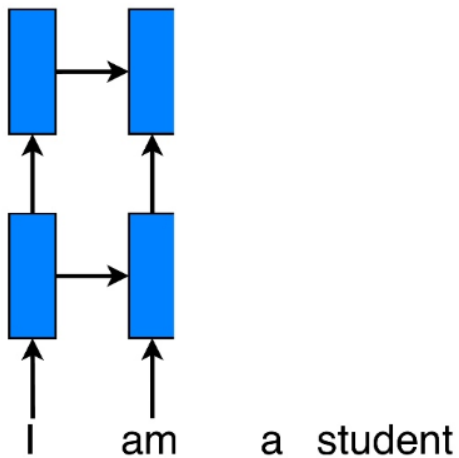
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



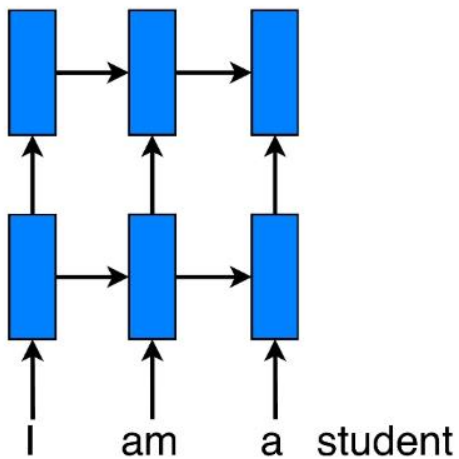
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



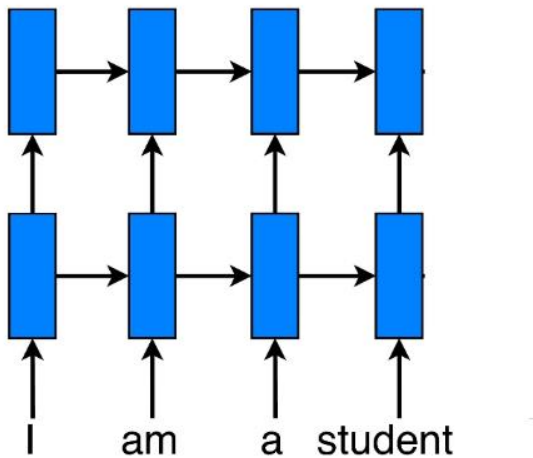
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



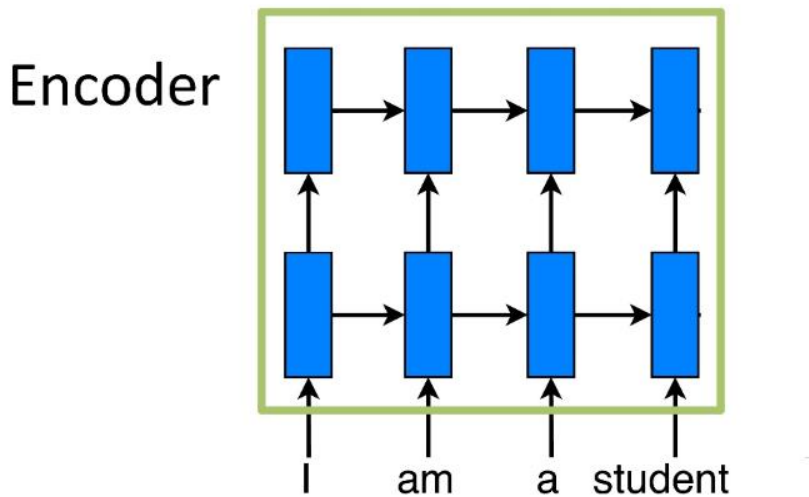
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



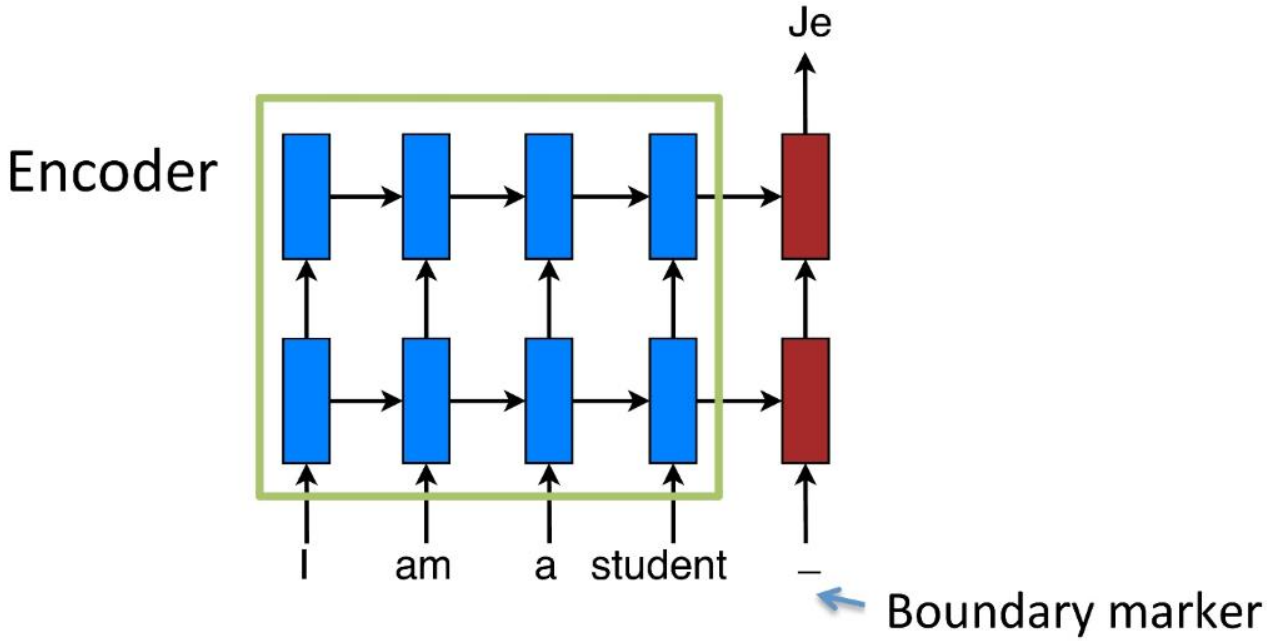
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



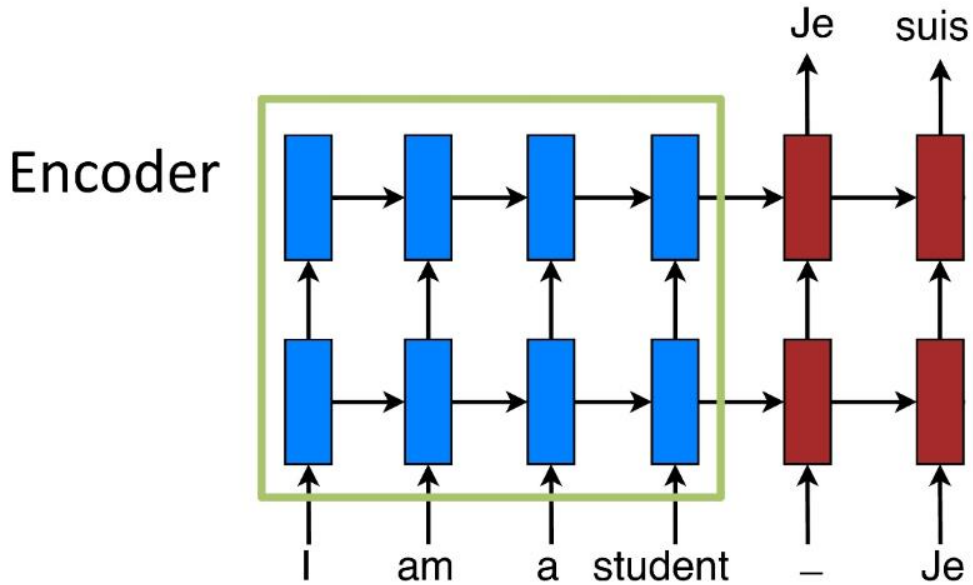
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



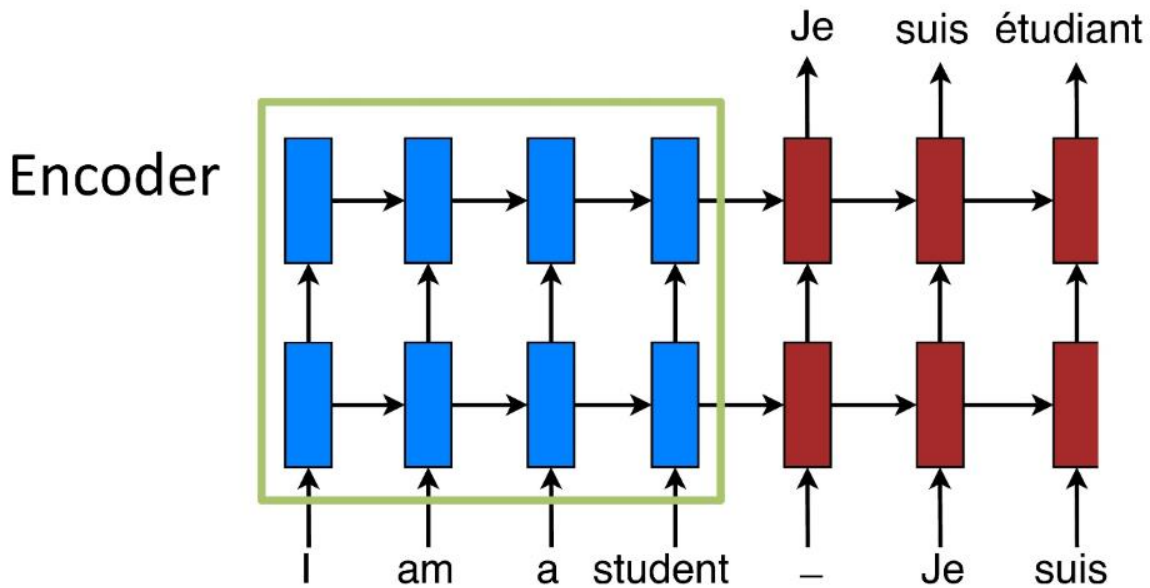
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



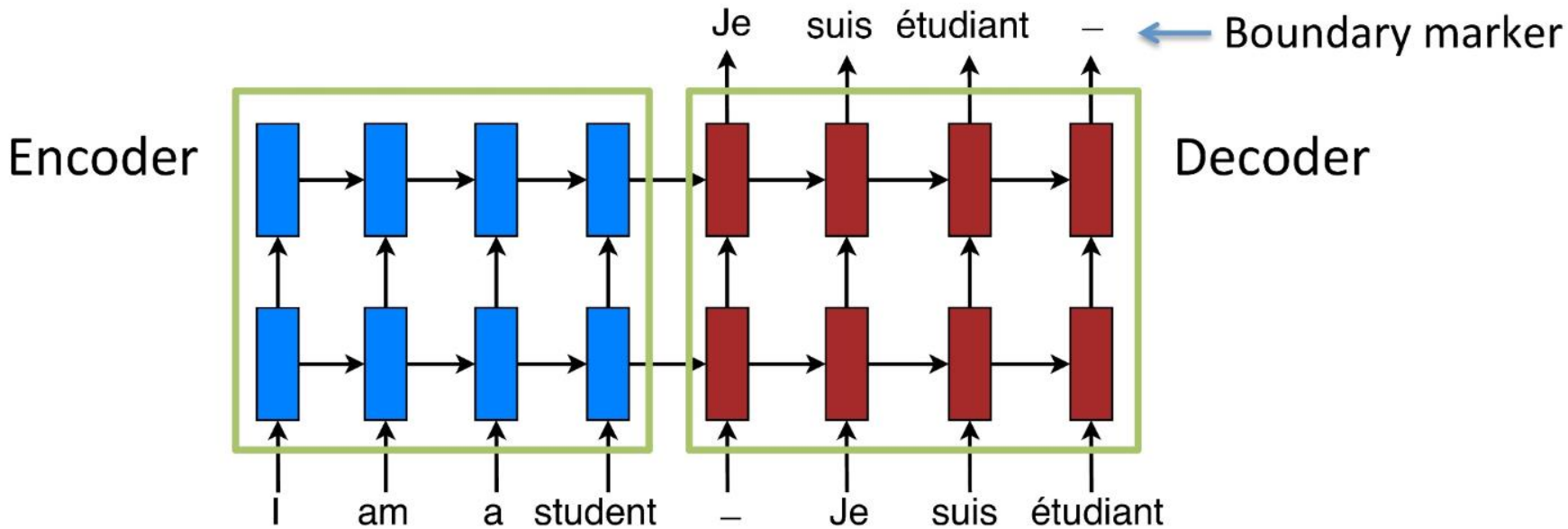
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



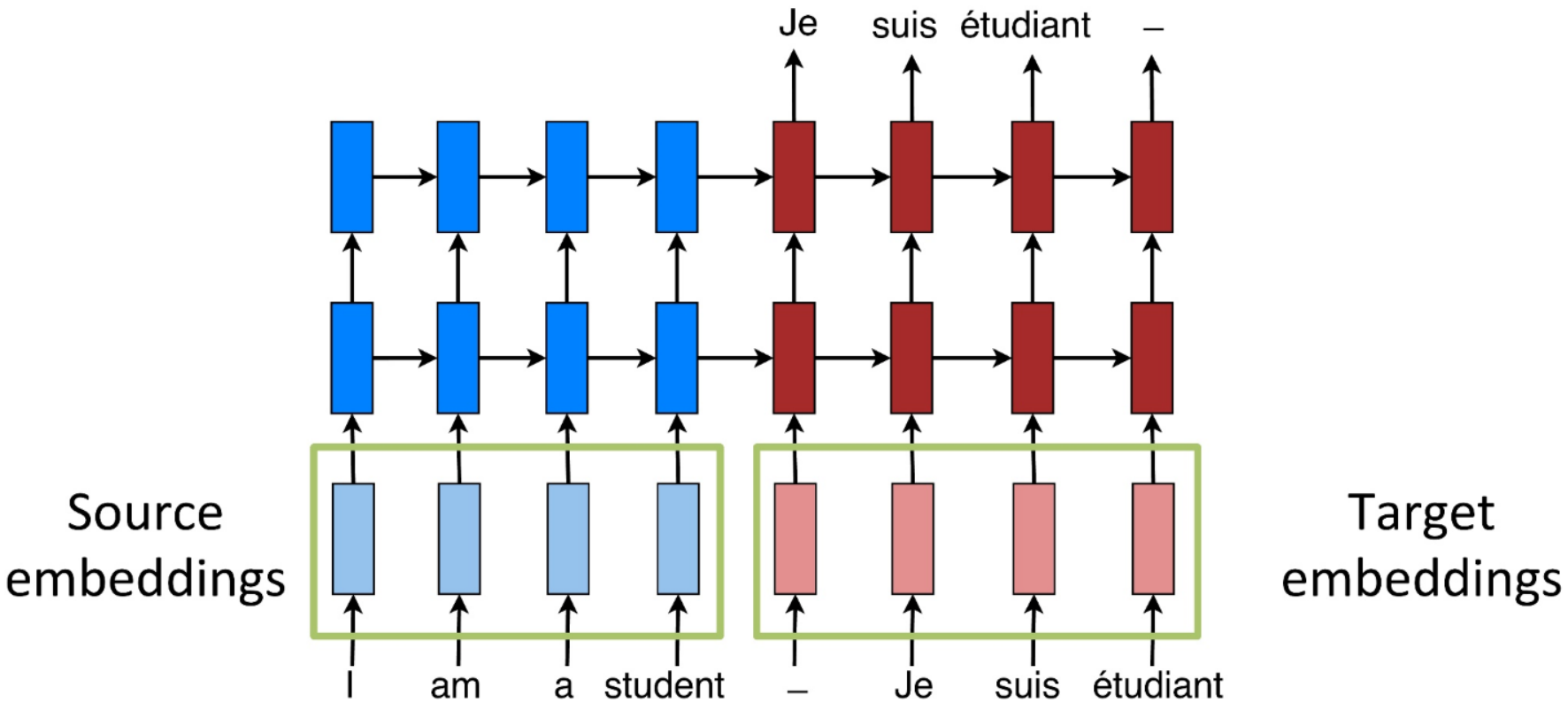
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Neural Machine Translation (NMT)



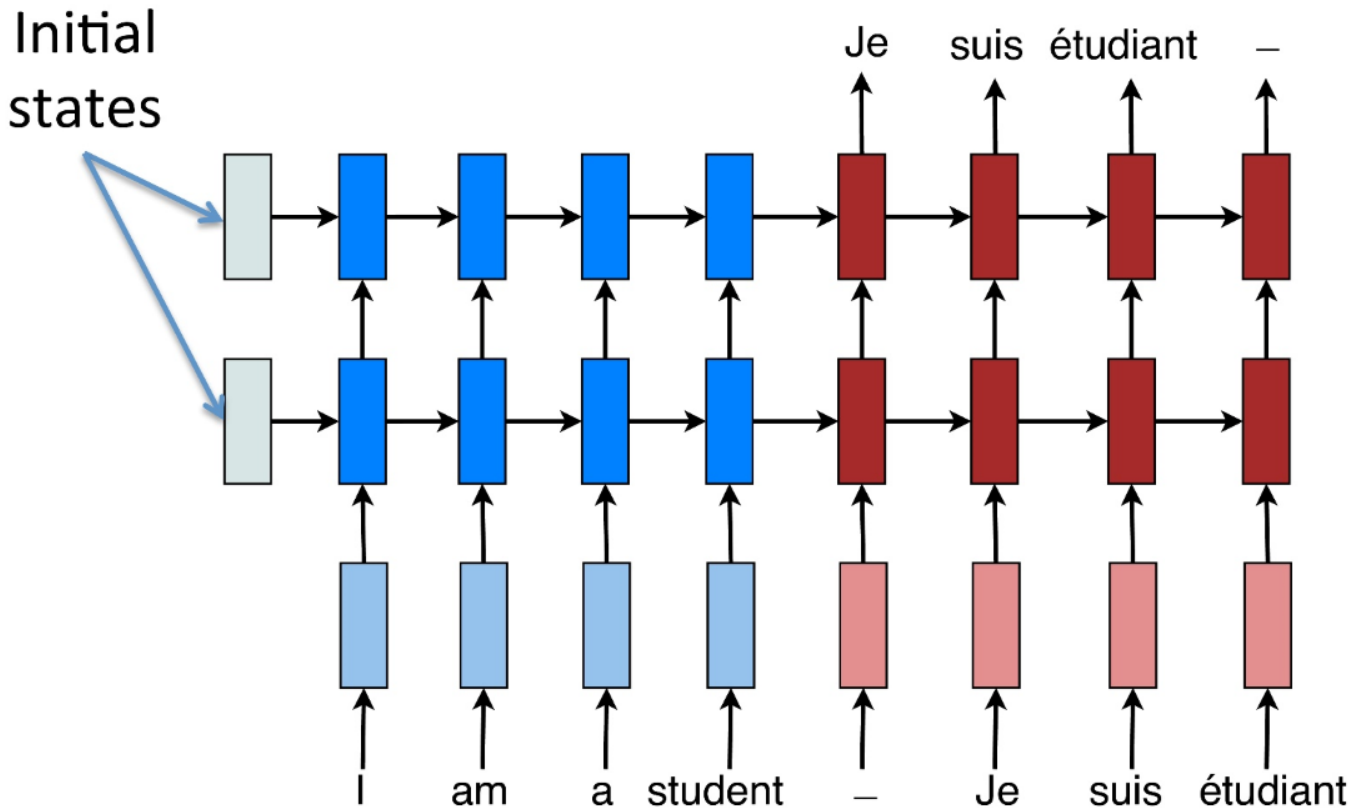
- Recurrent Neural Networks:
  - Model P(target | source) directly.
  - Can be trained end-to-end.

Word Embeddings



- One for each language: can learn from scratch.

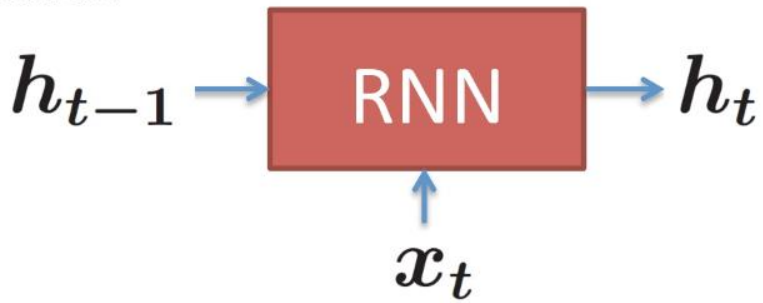
Recurrent Connections



- Often set to 0.

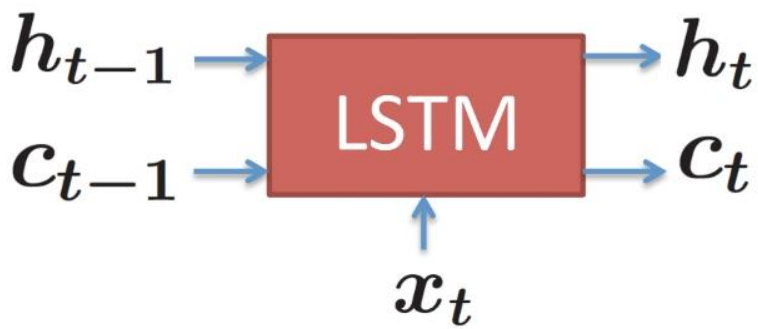
Recurrent Units

- Vanilla:



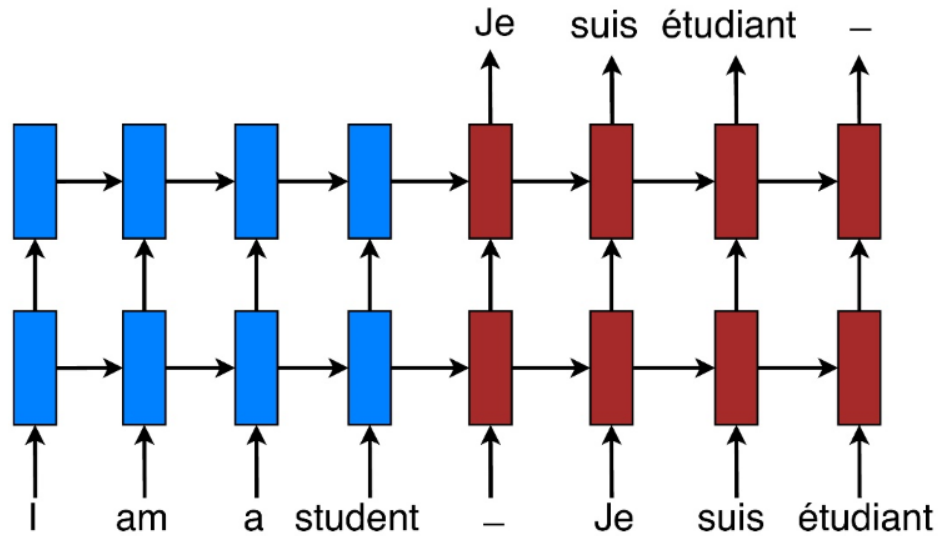
Vanishing gradient problem!

- LSTM:

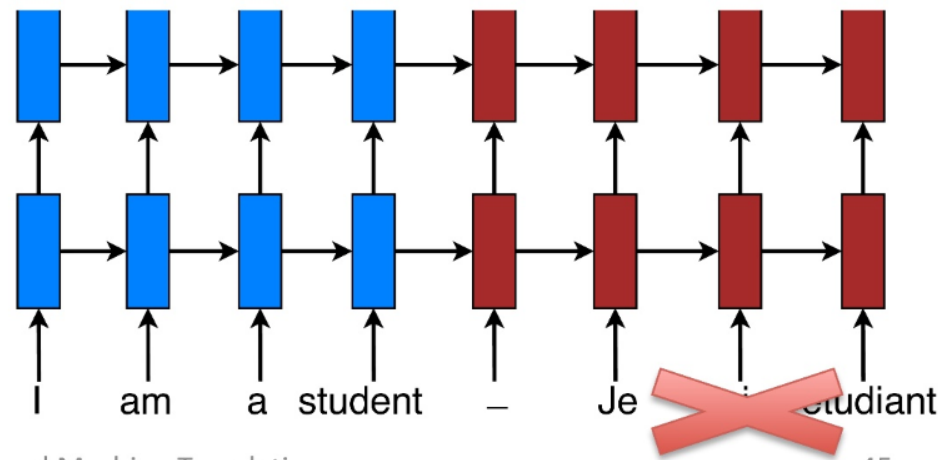


Training vs. Testing

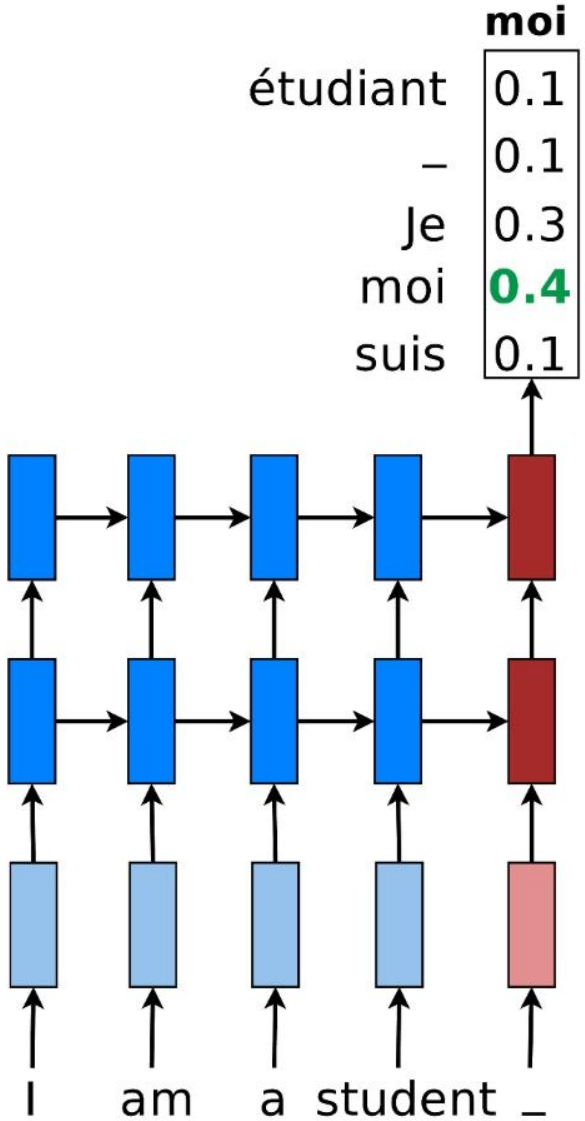
- *Training*
  - Correct translations are available.



- *Testing*
  - Only source sentences are given.

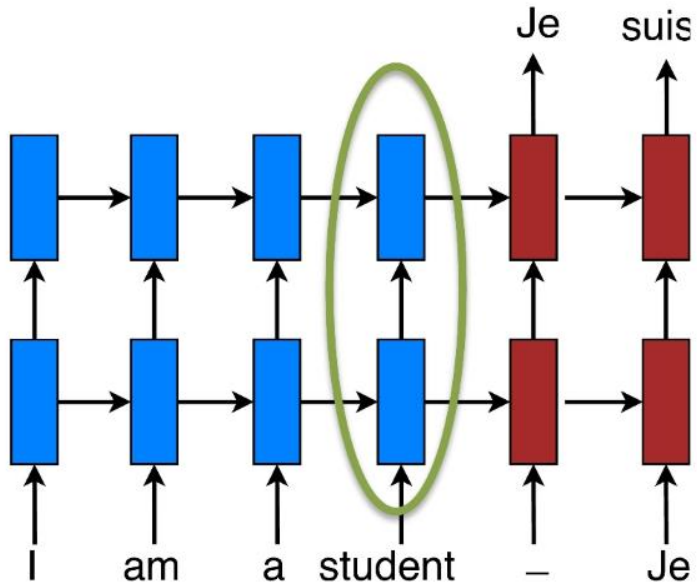


Testing



- Feed the most likely word

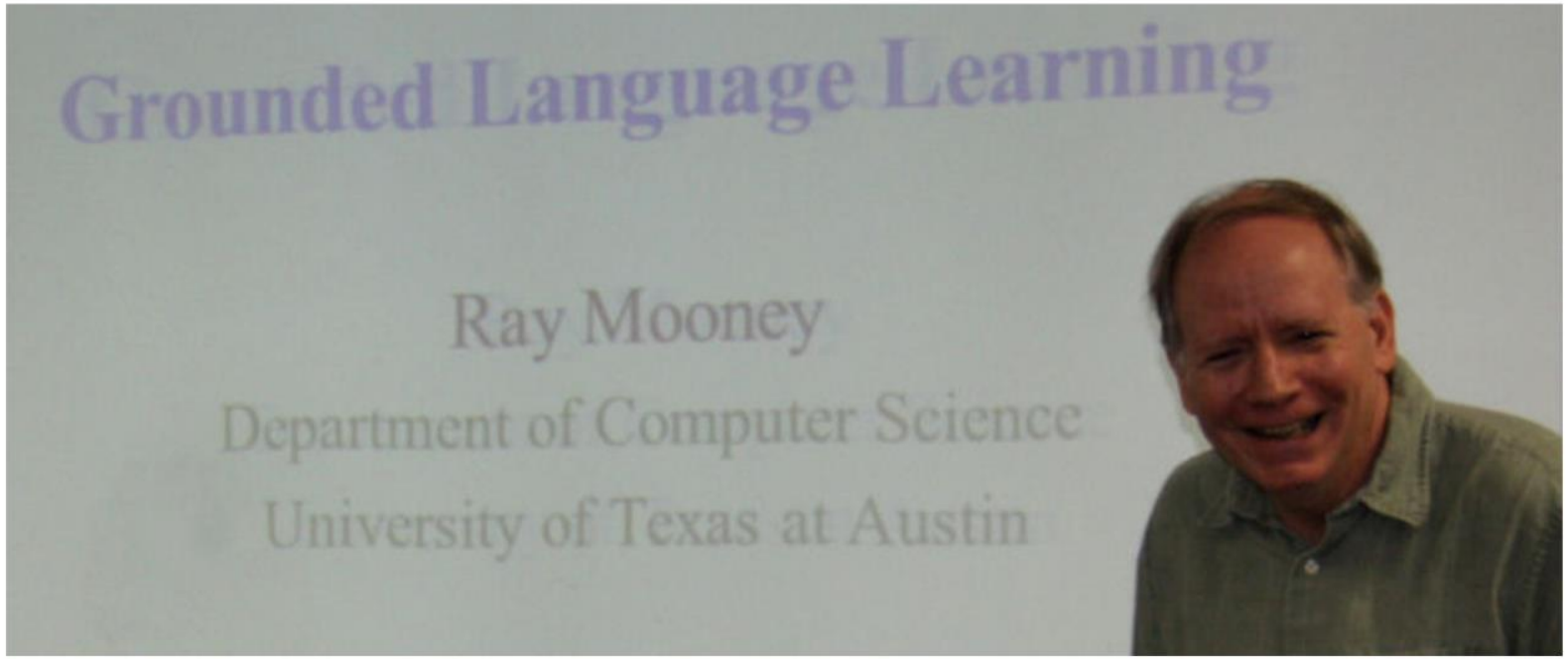
#2 The Sentence Length Problem



- Translation quality degrades with long sentences.

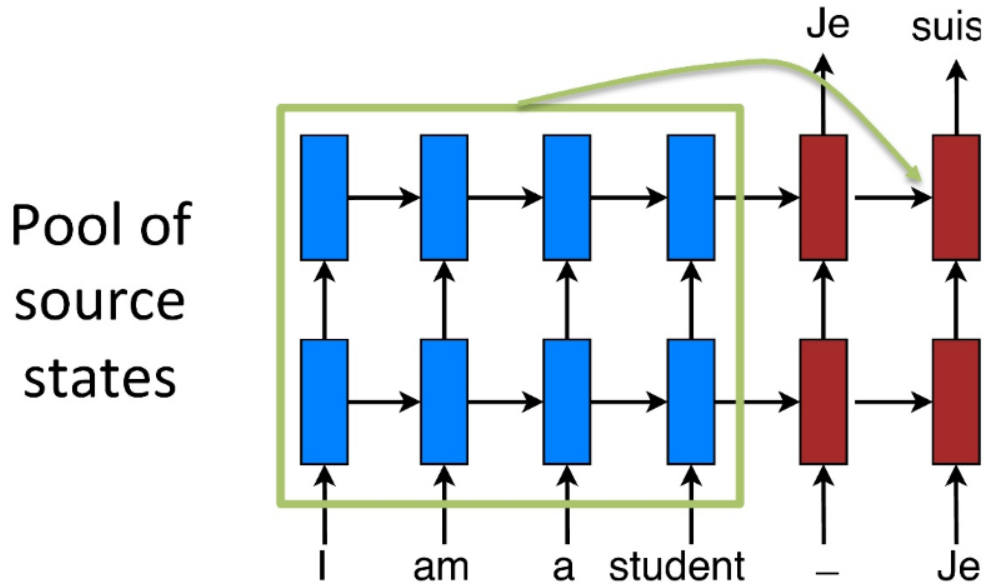
Problem: sentence meaning is represented by a fixed-dimensional vector.

You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!



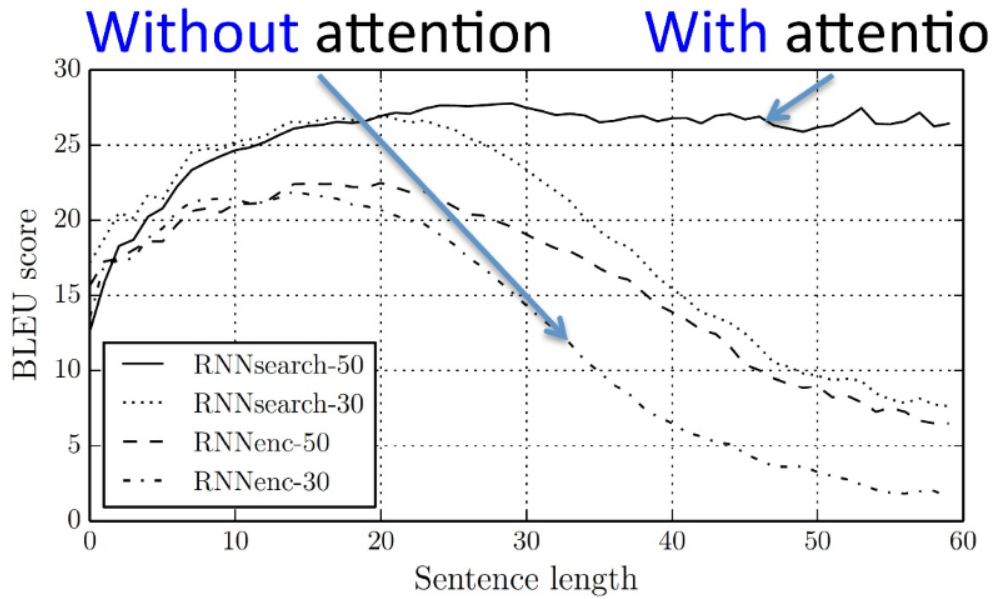
(Adapted from KyungHuyn Cho' talk)

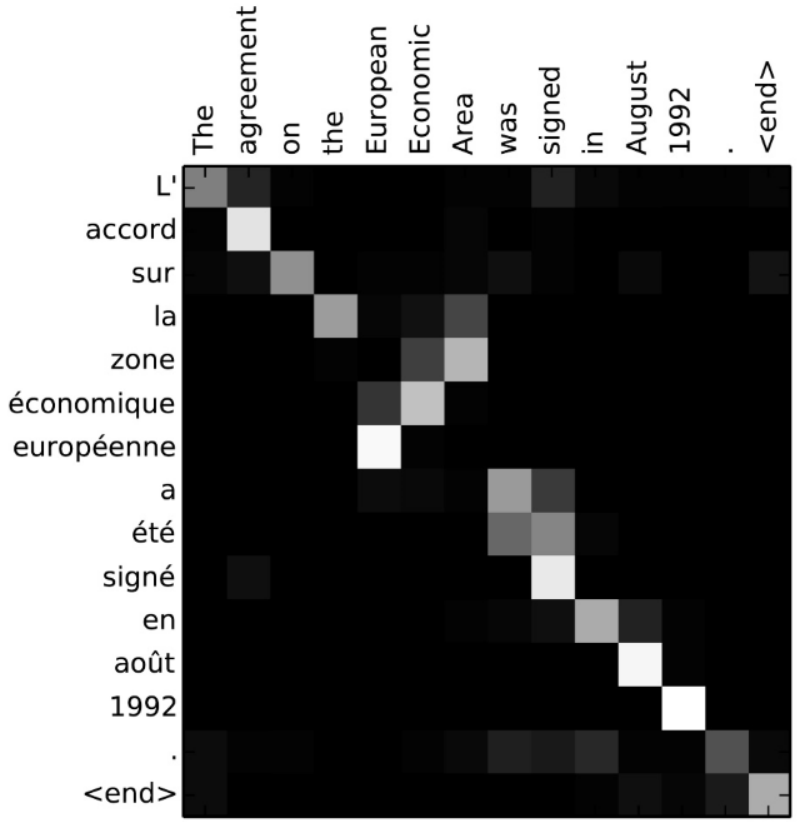
Attention Mechanism



- Solution: random access memory
  - Retrieve as needed.

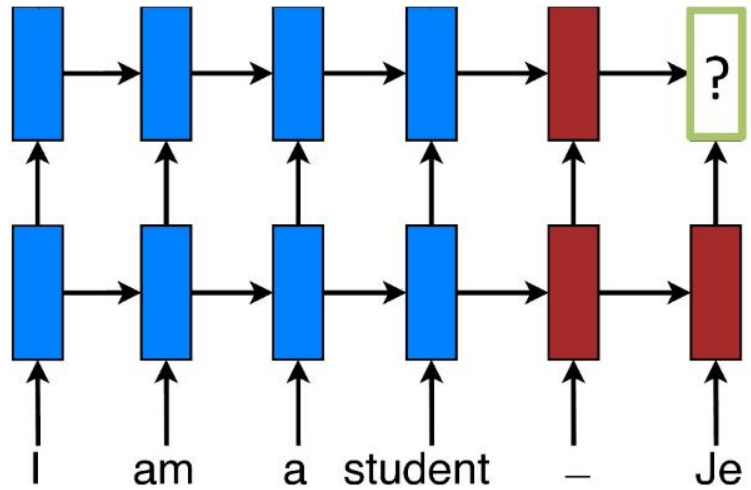
Université de Montréal





Dzmitry Bahdanau, KyungHuyn Cho, and Yoshua Bengio. ***Neural Machine Translation by Jointly Learning to Translate and Align****. ICLR 2015.*

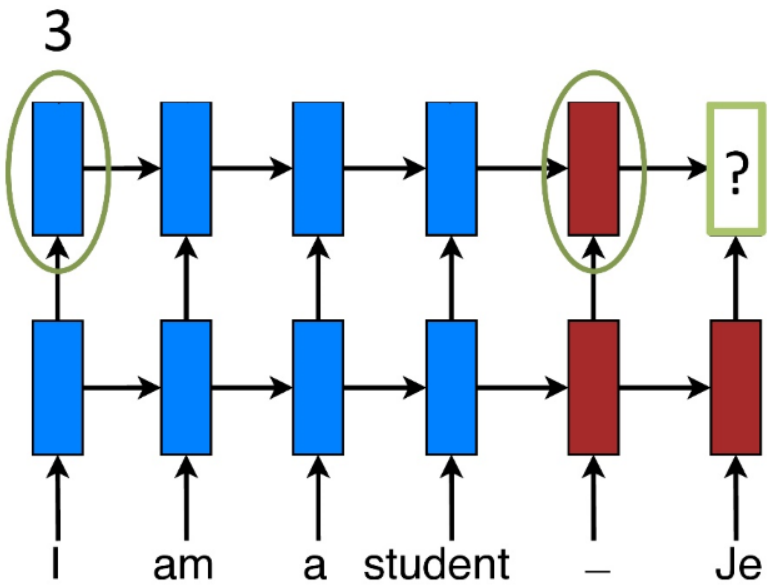
Attention Mechanism



A simplified version of (Bahdanau et al., 2015)

Attention Mechanism – *Scoring*

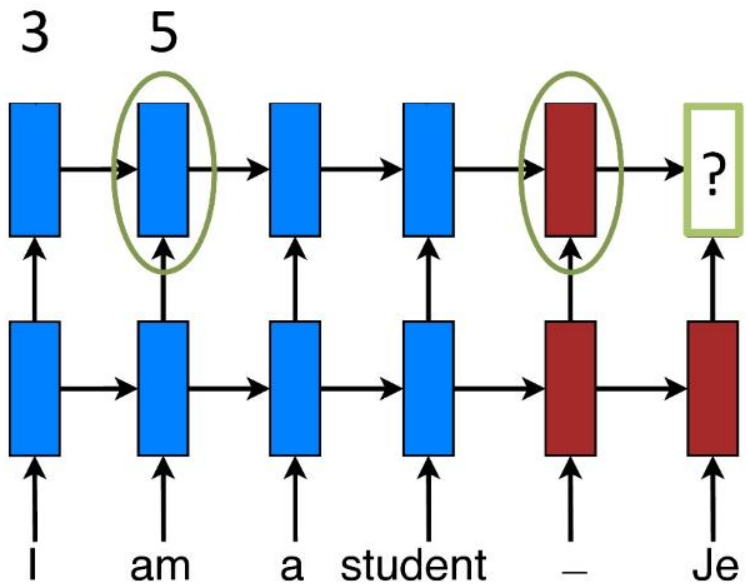
$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)$$



- Compare target and source hidden states.

Attention Mechanism – *Scoring*

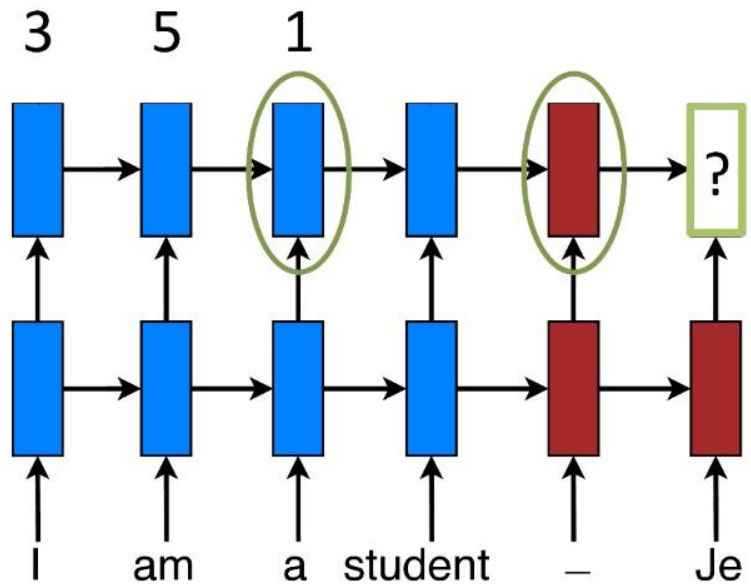
$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)$$



- Compare target and source hidden states.

Attention Mechanism – *Scoring*

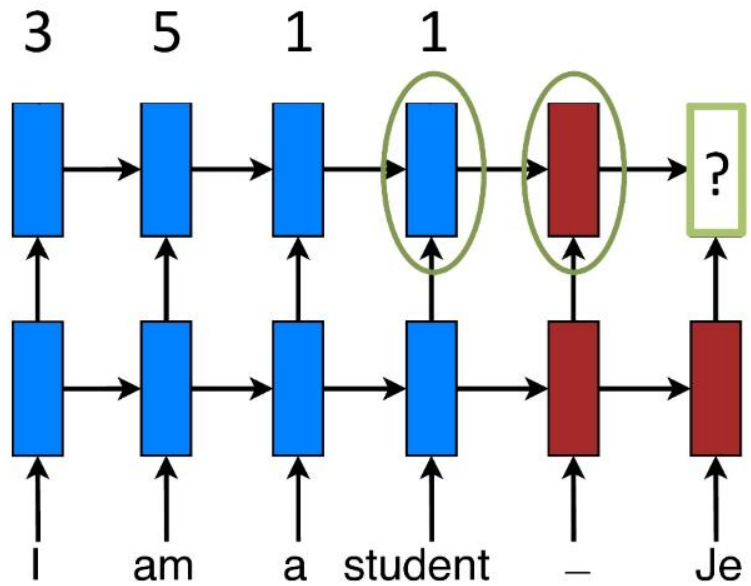
$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)$$



- Compare target and source hidden states.

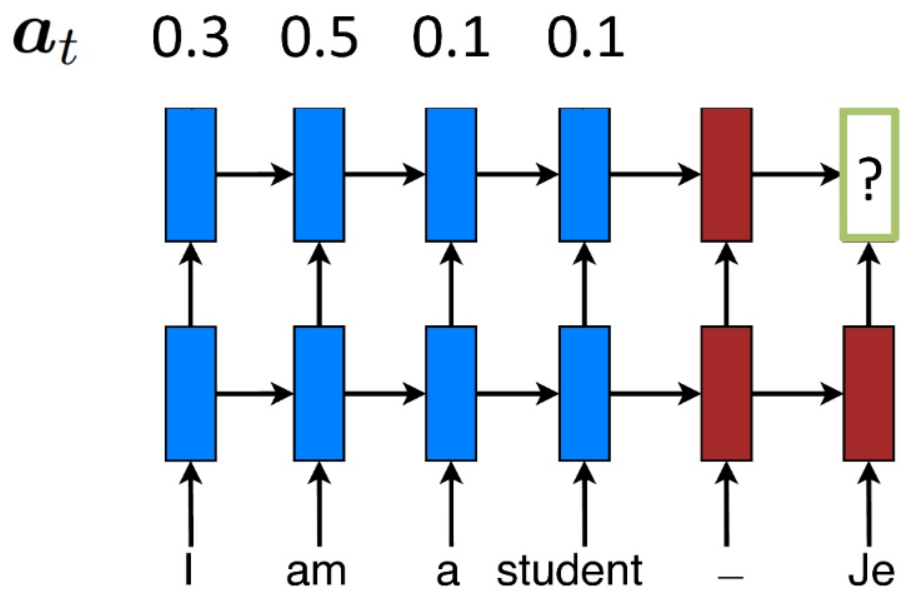
Attention Mechanism – *Scoring*

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)$$



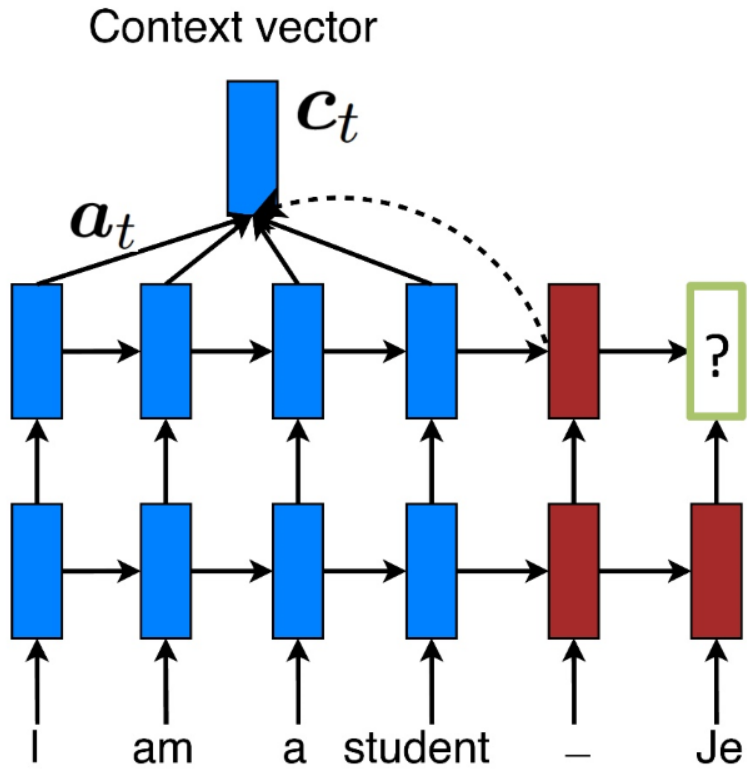
- Compare target and source hidden states.

Attention Mechanism – *Normalization*



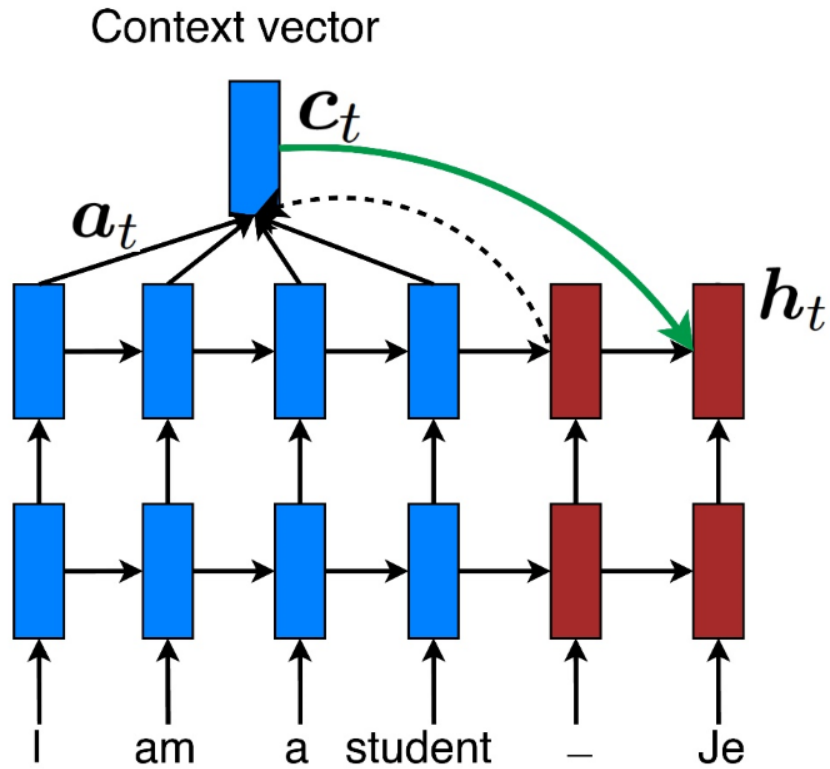
- Convert into alignment weights.

Attention Mechanism – *Context vector*



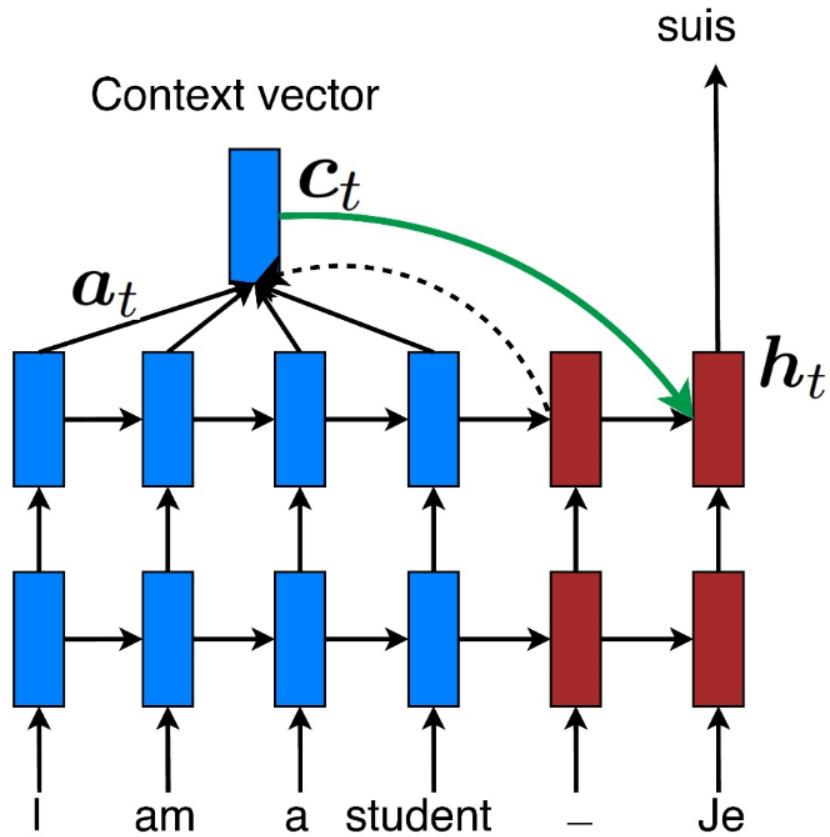
- Build context vector: weighted average.

Attention Mechanism – *Hidden state*



- Compute the next hidden state.

Attention Mechanism – *Predict*



- Predict the next word.