Neural Machine Translation

Radek Bartyzal
rbartyzal1@gmail.com

31.8.2016

Let’s talk ML in Prague

History

- ~ WW2: Word by word + language rules
  - Requires linguists
  - Does not work well in real world

History

~ 1980: Statistical Machine Translation (SMT)

- Requires *parallel corpora = Rosetta stones*
- Must be tweaked for every pair of languages

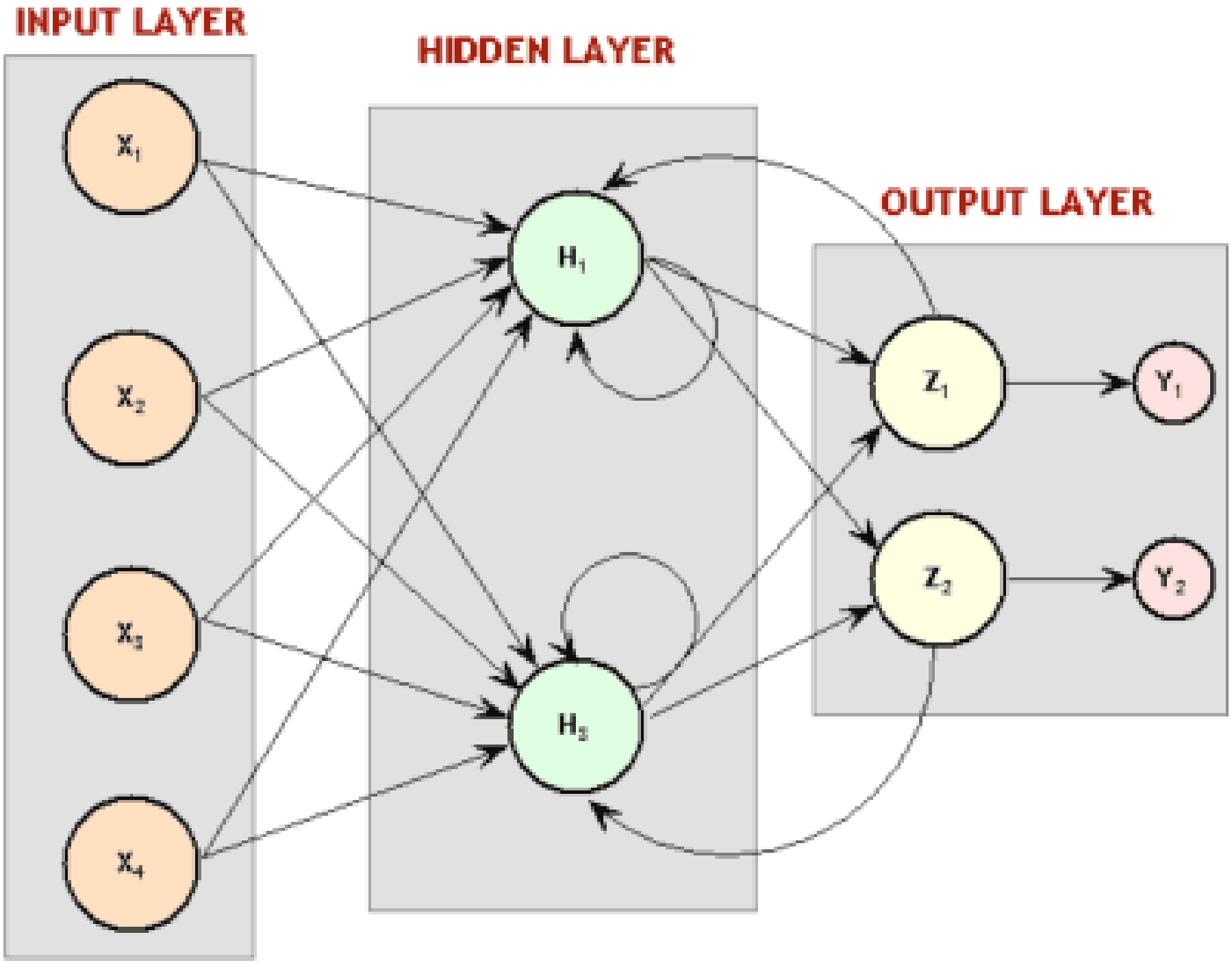
1) Find all translations of each word

2) Generate all possible sentences from these translations including all possible orderings of words

3) Pick the sentence that has the most likely word translations while also being the most similar overall to real sentences

Recurrent Neural Network

- Next state = function of previous state and input
- Used for sequential processing


INPUT LAYER
HIDDEN LAYER
OUTPUT LAYER
X1
X2
X3
X4
H1
H2
Z1
Z2
Y1
Y2

RNN Encoder - Decoder

- Variable length in

= input sequence

=> **fixed size encoding**

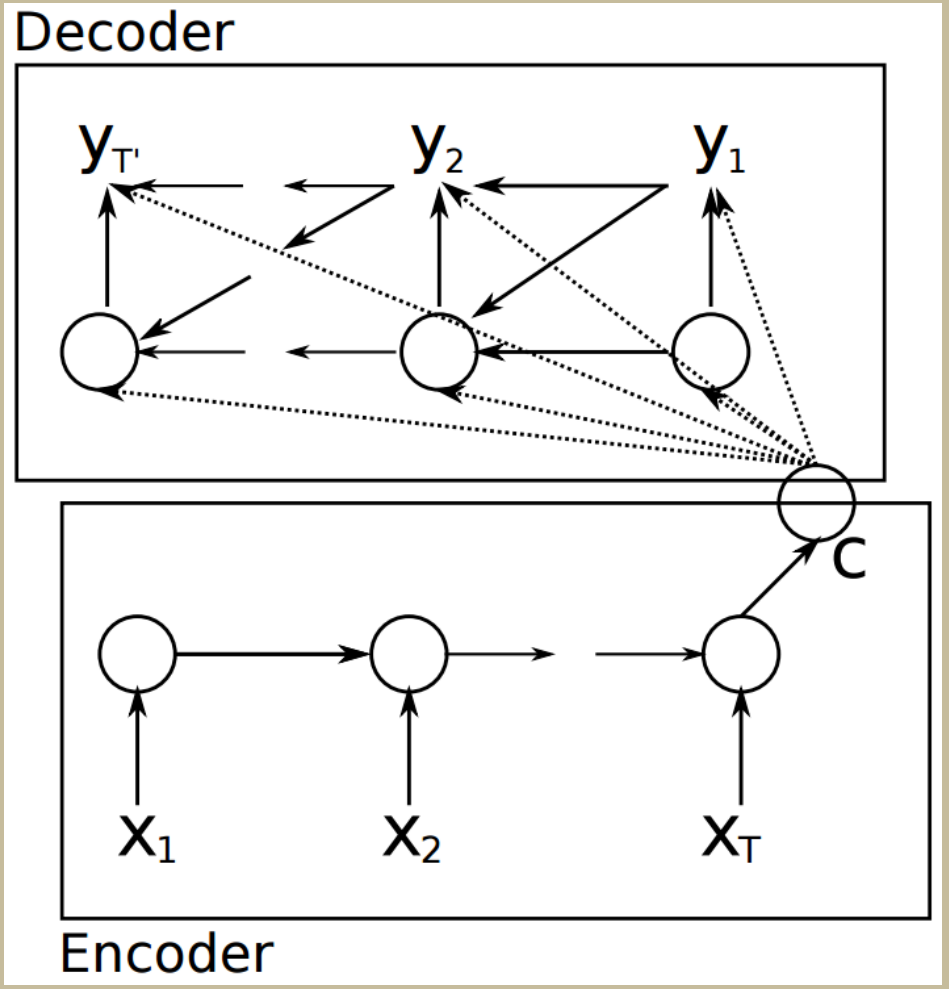
= vector of size N = number of hidden units

=> **decoded output**

= function of:

  - prev decoded output
  - prev state
  - encoding

- Variable length out



Training

- Train = maximize the conditional probability of the target sequence given a source sequence

- Encoder & Decoder trained jointly

Usage

- Model output

  = translation probability of an English phrase to a corresponding French phrase

- Used in good old phrase-based SMT for scoring each phrase pair in the phrase table

Hidden units

- Simple recurrent unit

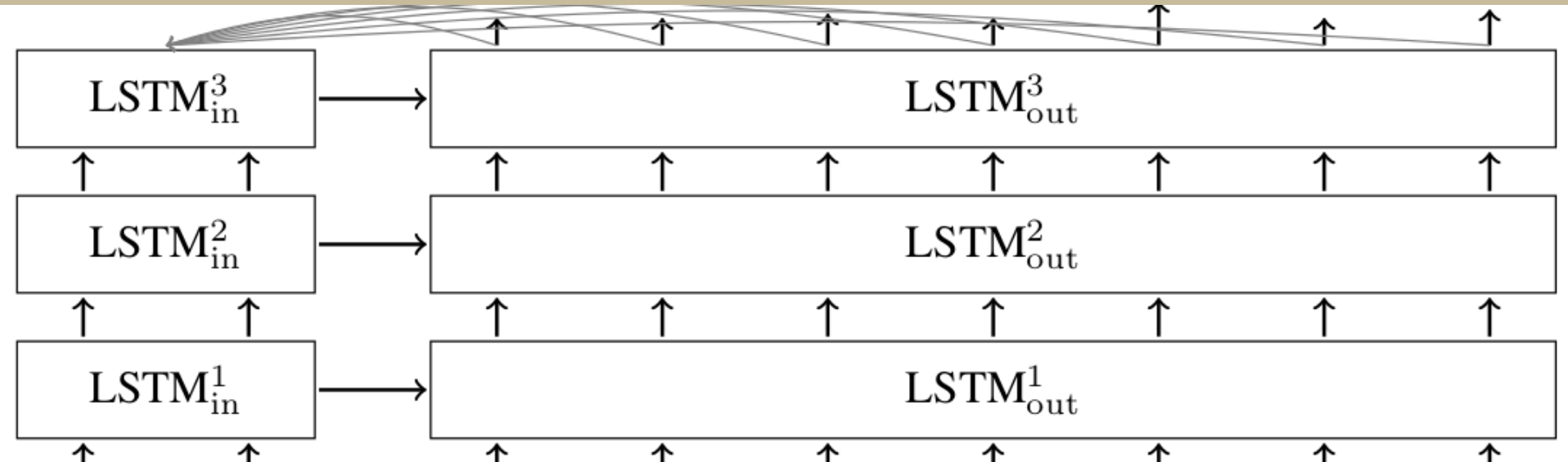
- Gated recurrent unit = GRU
- Long Short-Term Memory unit = LSTM

Point?

How much information is kept from the previous state

Current research

- Attention based NMT
  - peek into the input at every decoding step
  - **local** = look only at a group of words
  - **global** = look at all the input words



Sources

- RNN Encoder – Decoder
- Easy to read explanation of seq2seq
- Effective Approaches to Attention-based Neural Machine Translation
- Tensor Flow seq2seq