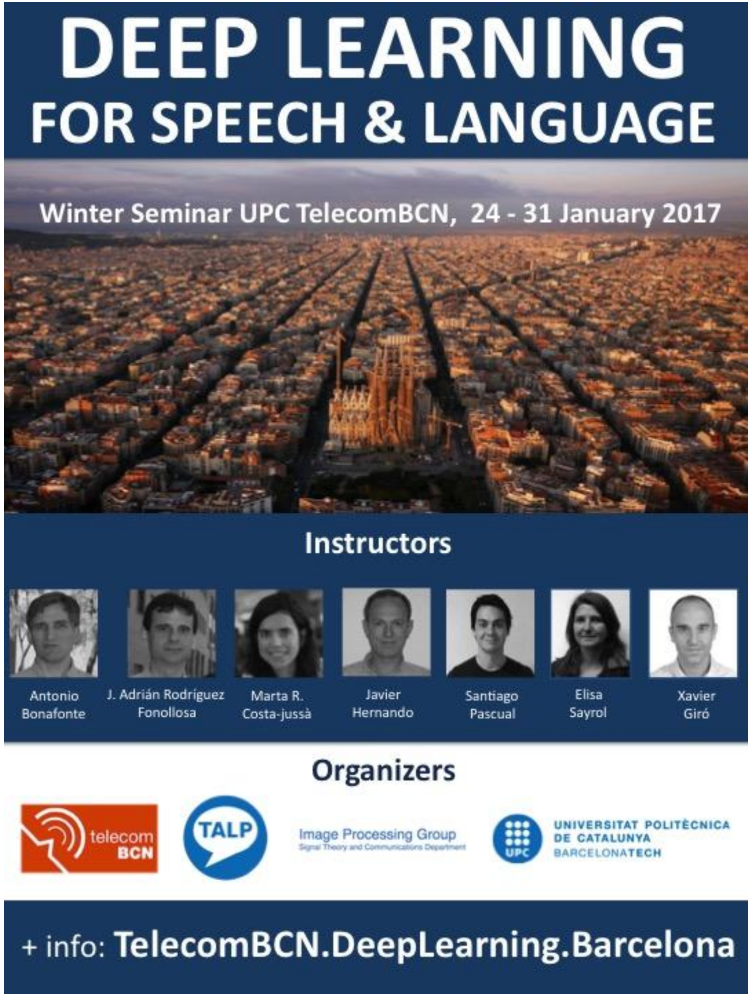


[course site]

Day 3 Lecture 4

Neural Machine Translation

Marta R. Costa-jussà

Acknowledgments

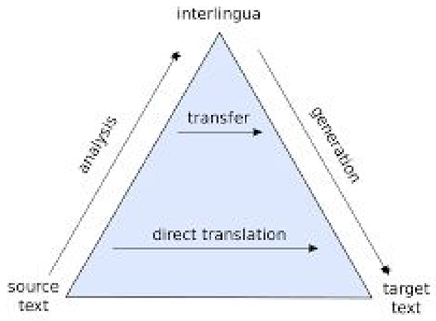
Kyunghyun Cho, NVIDIA BLOGS:
https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-with-gpus/

TALP Research Center

Previous concepts from this course

- Recurrent neural network (LSTM and GRU) (handle variable-length sequences)
- Word embeddings
- Language Modeling (assign a probability to a sentence)

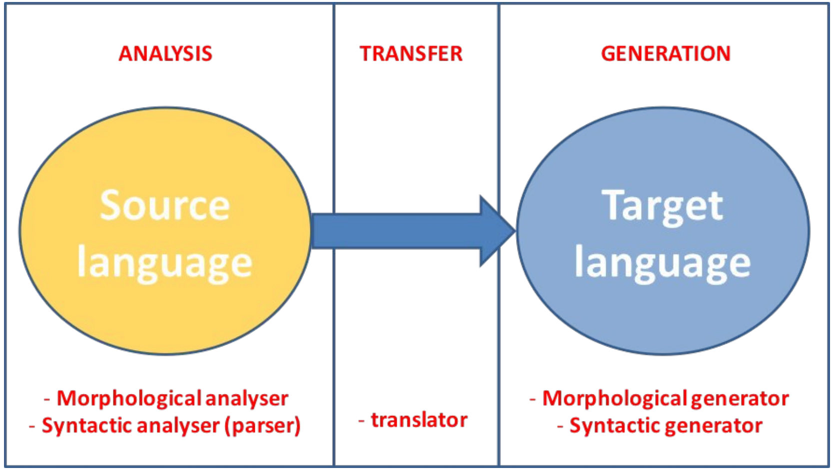
Machine Translation background



Machine Translation is the application that is able to automatically translate from source (S) to target (T).

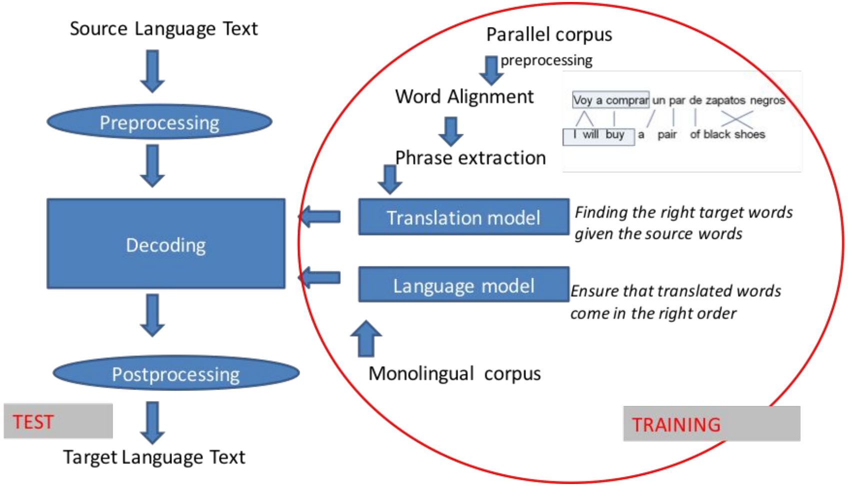
Rule-based approach

Main approaches have been either rule-based or statistical-based



Statistical-based approach

Main approaches have been either rule-based or statistical-based



Why a new approach?

We need years to develop a nice rule-based approach

Regarding statistical systems:

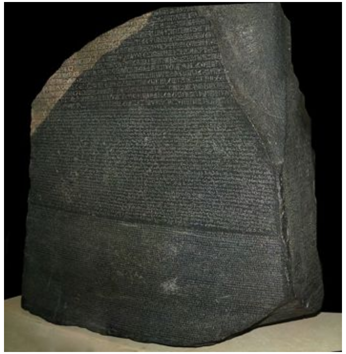
(1) Word alignment and Translation are optimized separately
(2) Translation at the level of words, but difficulties with high variations in morphology (e.g. translation English-to-Finnish)
(3) Translation by language pairs
    (a) difficult to think of an automatic interlingua
    (b) bad performance with low resourced-languages

Why Neural Machine Translation?

- Integrated MT paradigm
- Trainable at the subword/character level
- Multilingual advantages

What do we need?

- Parallel Corpus



English	Russian
This course is a thorough introduction to machine translation technology	Этот курс представляет собой интенсивное введение в технологию машинного перевода
We will describe all aspects of building a statistical machine translation system, from both formal and practical perspectives	Мы рассмотрим все аспекты построения системы статистического машинного перевода с теоретической и практической точки зрения

Same requirement than phrase-based systems

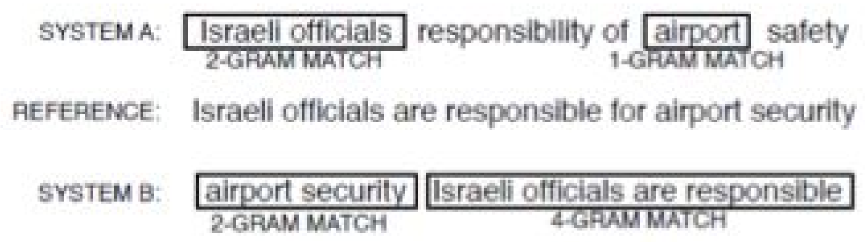
Sources of parallel corpus

- European Plenary Parlament Speeches (EPPS) transcriptions
- Canadian Handsards
- United Nations
- CommonCrawl
- ...

→ International evaluation campaigns:
Conference on Machine Translation (WMT)
International Workshop on Spoken Language Translation (IWSLT)

What else do we need?

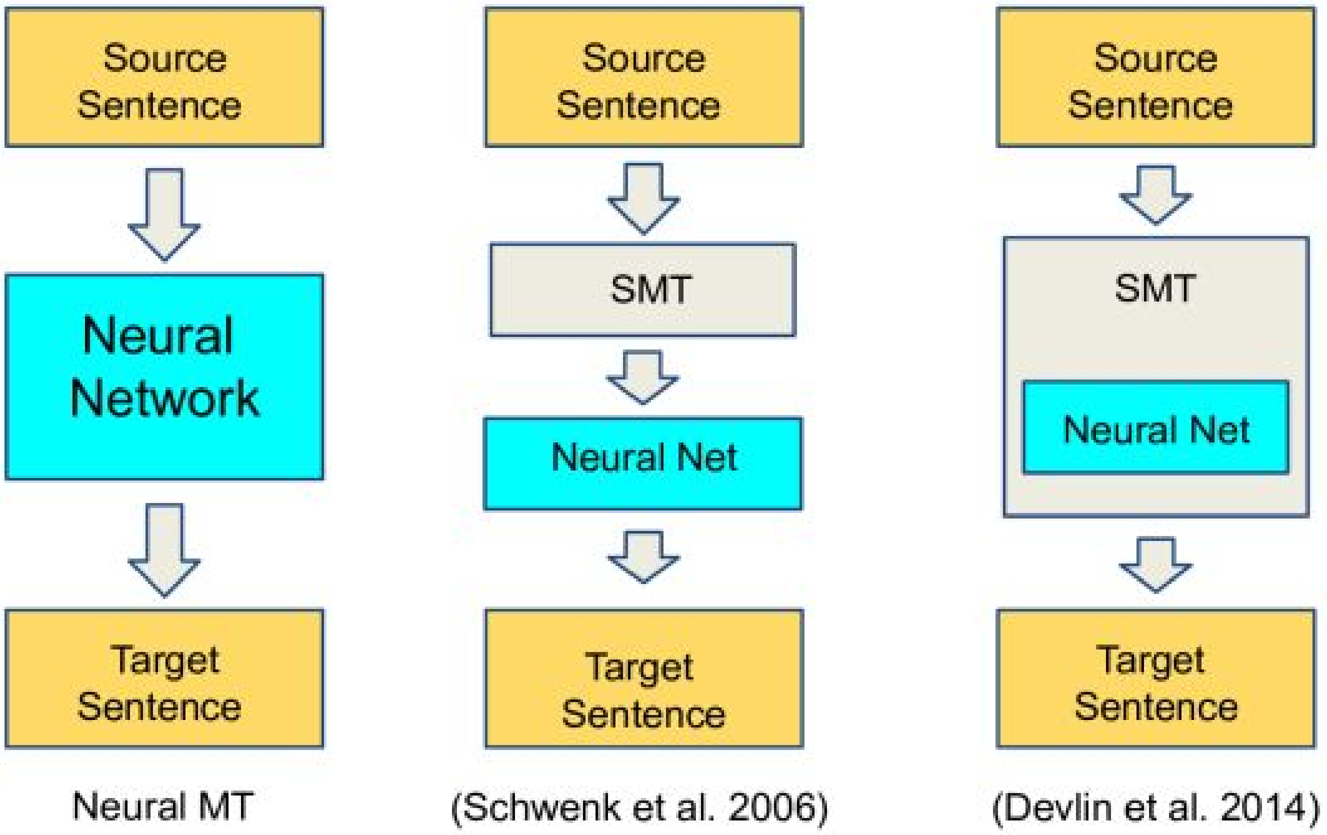
Automatic measure



Metric	System A	System B
precision (1gram)	3/6	6/6
precision (2gram)	1/5	4/5
precision (3gram)	0/4	2/4
precision (4gram)	0/3	1/3
brevity penalty	6/7	6/7
BLEU	0%	52%

Same requirements than phrase-based systems

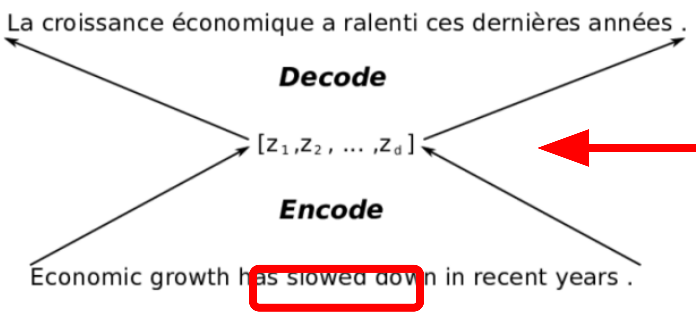
Towards Neural Machine Translation



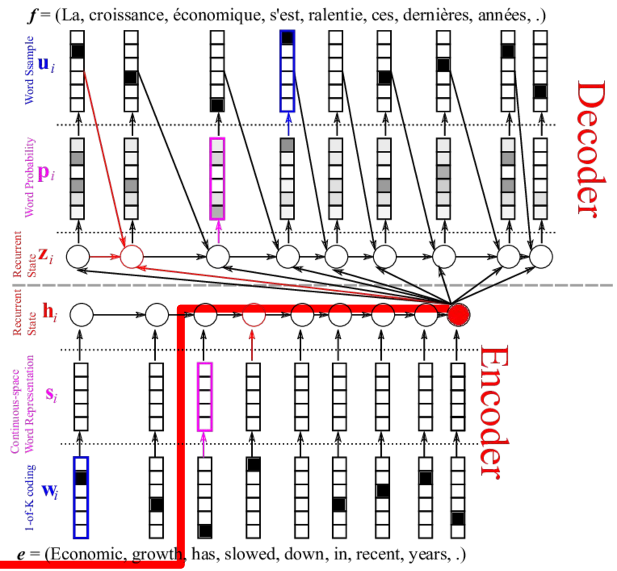
Kyunghyun Cho, "Introduction to Neural Machine Translation with GPUs" (2015)

Encoder-Decoder

Front View



Side View

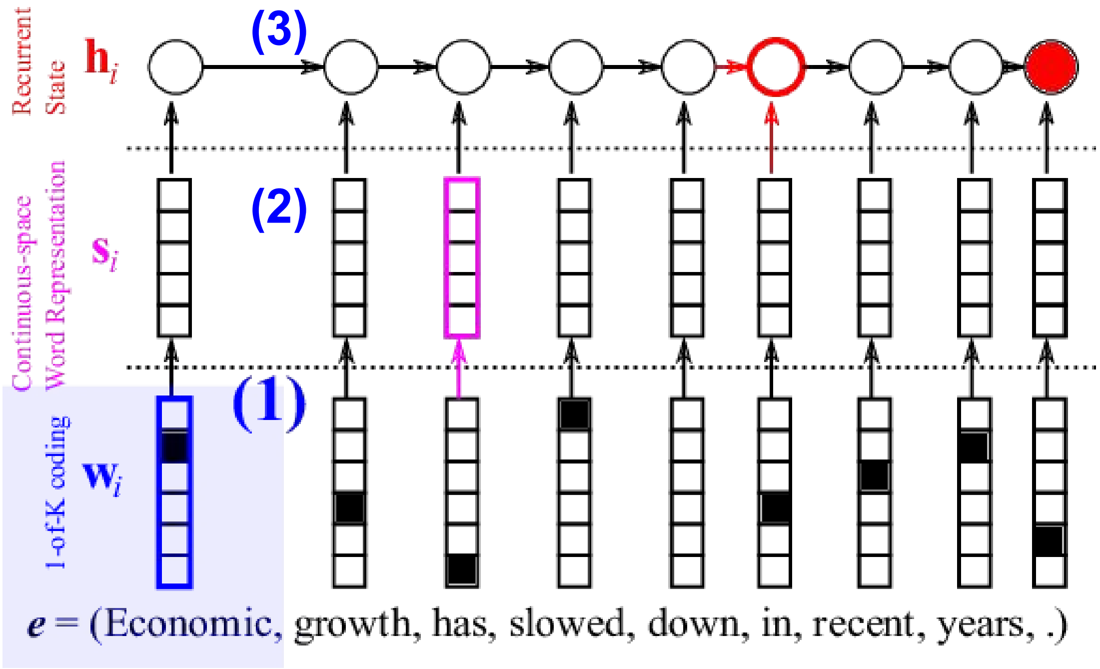


Representation of the sentence

Kyunghyun Cho, “Introduction to Neural Machine Translation with GPUs” (2015)
Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

Encoder

Encoder in three steps



(1) One hot encoding
(2) Continuous space representation
(3) Sequence summarization

Kyunghyun Cho, “Introduction to Neural Machine Translation with GPUs” (2015)

Step 1: One-hot encoding

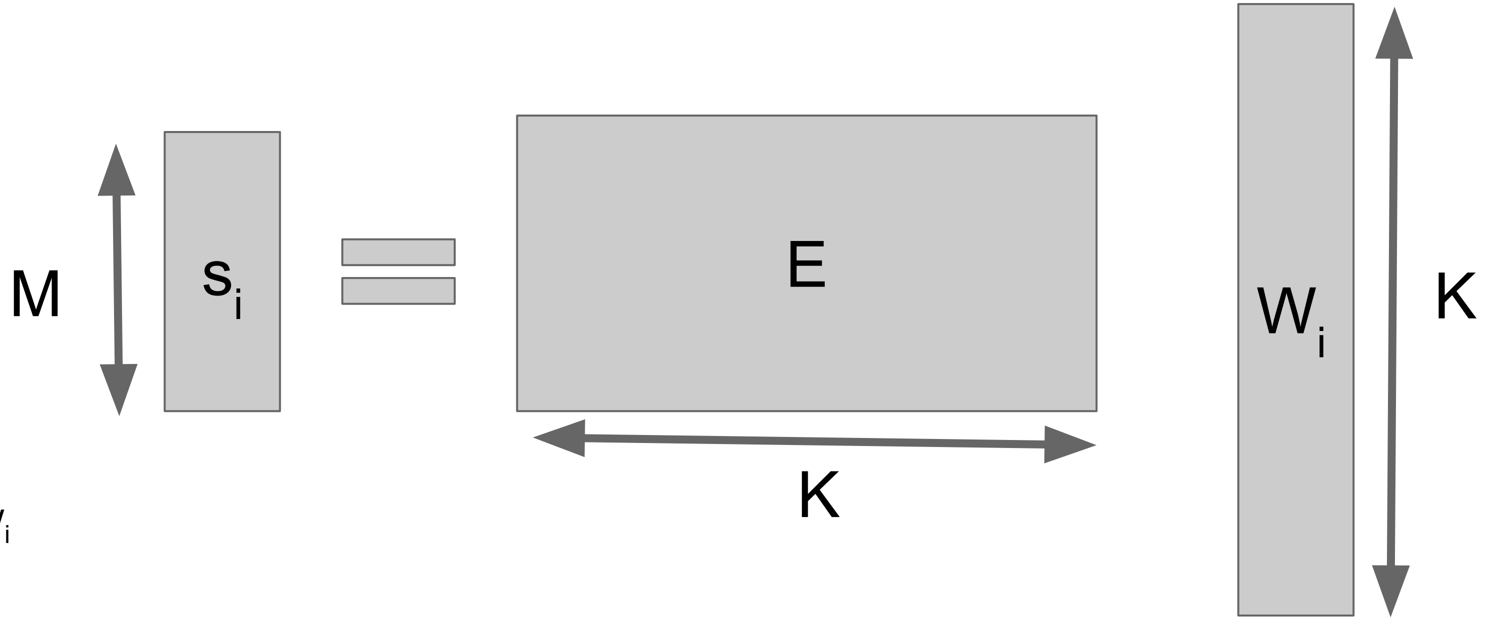
From previous lecture on language modeling

Natural language words can also be one-hot encoded on a vector of dimensionality equal to the size of the dictionary (K).

Word	One-hot encoding
economic	000010...
growth	001000...
has	100000...
slowed	000001...

Step 2: Projection to continuous space

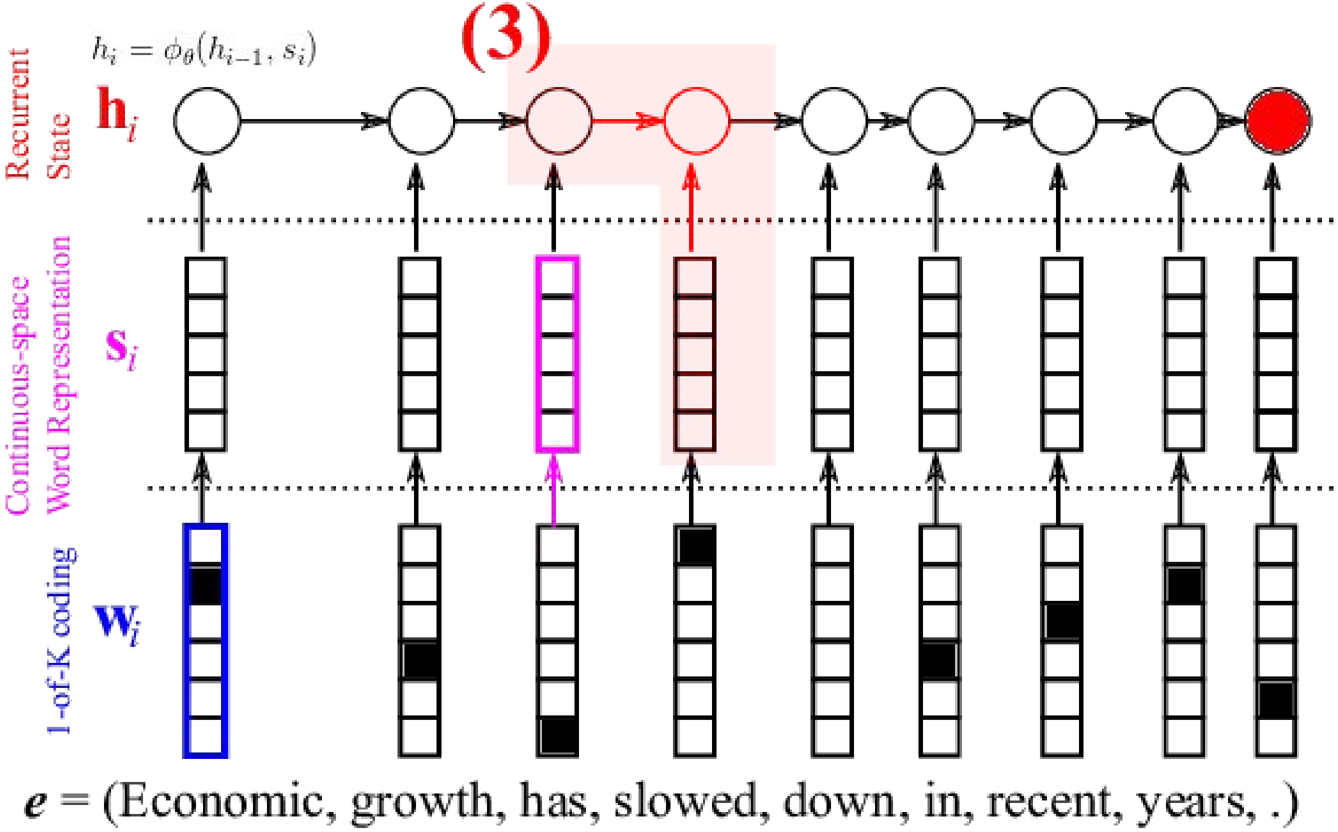
The one-hot is linearly projected to a space of lower dimension (typically 100-500) with matrix E for learned weights.



$s_i = Ew_i$

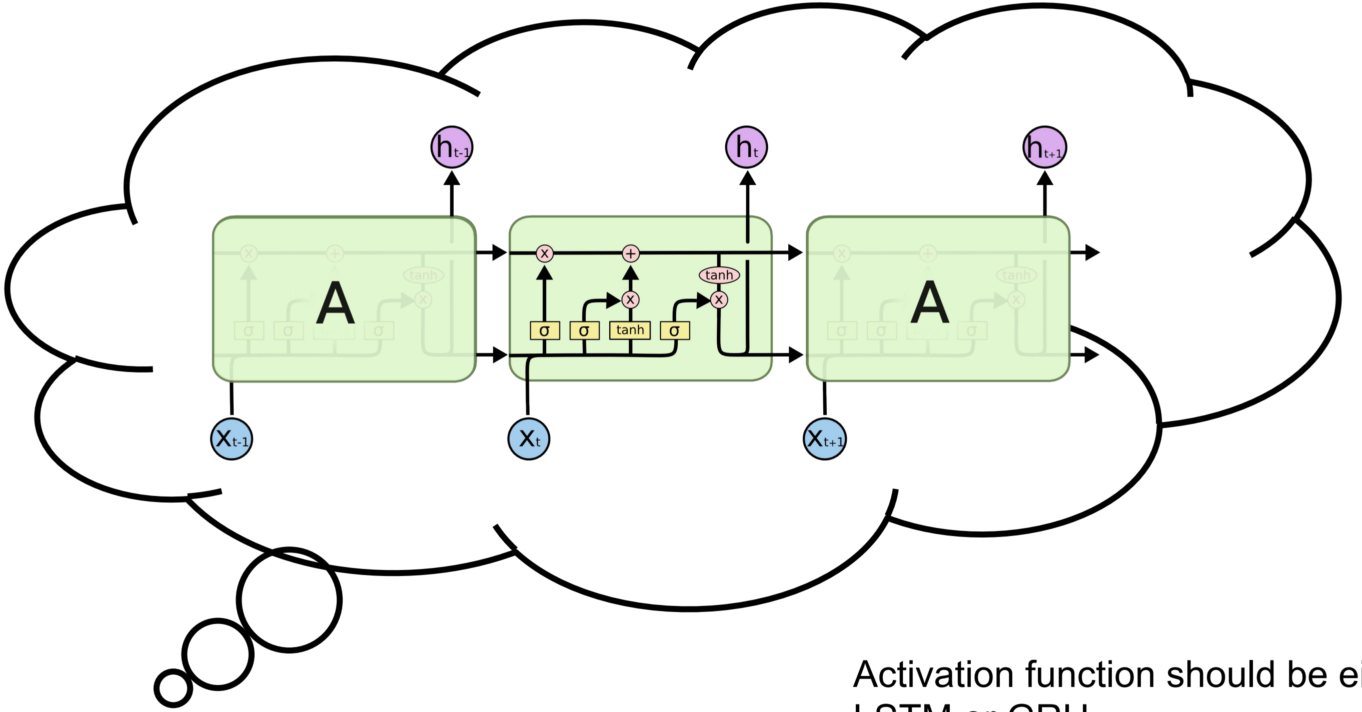
Kyunghyun Cho, [“Introduction to Neural Machine Translation with GPUs”]() (2015)

Step 3: Recurrence



Kyunghyun Cho, [“Introduction to Neural Machine Translation with GPUs”]() (2015)

Step 3: Recurrence



Sequence

Activation function should be either LSTM or GRU

Figure: Cristopher Olah, [“Understanding LSTM Networks”]() (2015)

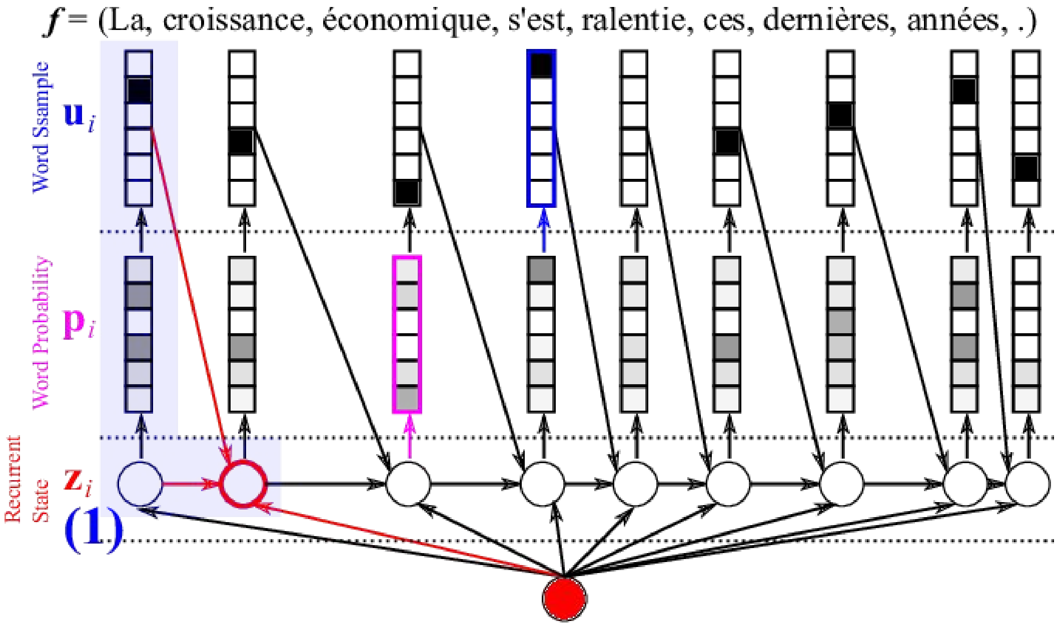
Decoder

Decoder

RNN's internal state z_i depends on: summary vector $h_{t'}$, previous output word u_{i-1} and previous internal state z_{i-1}.

NEW INTERNAL STATE

$$z_i = \phi_{\theta'}(h_T, u_{i-1}, z_{i-1}).$$



Kyunghyun Cho, “Introduction to Neural Machine Translation with GPUs” (2015)

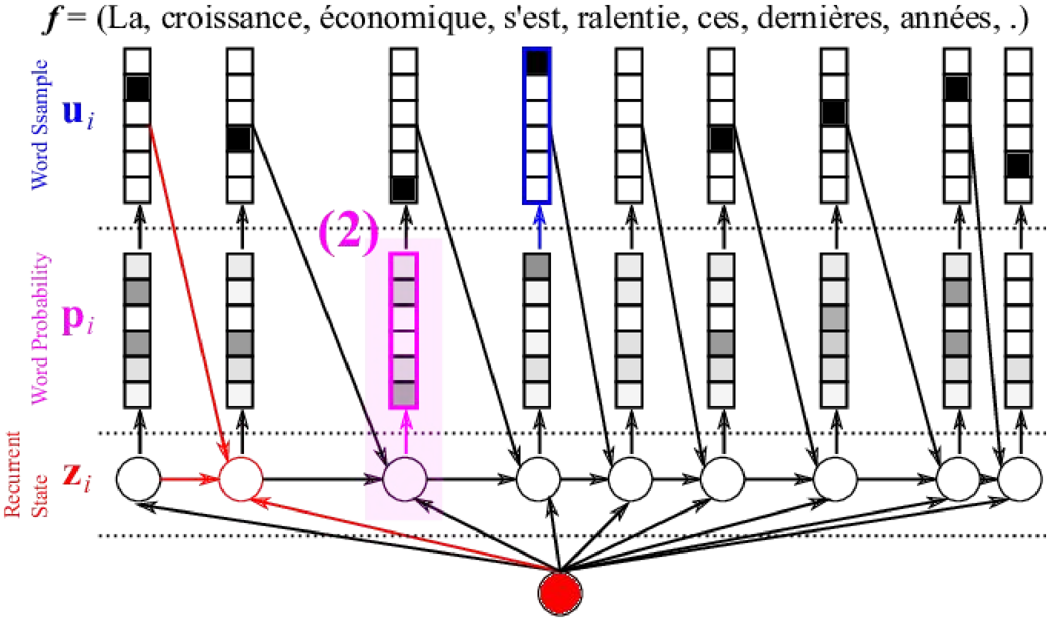
Decoder

With z_i ready, we can score each word *k* in the vocabulary with a dot product given this hidden state...

$$e(k) = w_k^\top z_i + b_k,$$

Neuron weights for word k

RNN internal state



Kyunghyun Cho, [“Introduction to Neural Machine Translation with GPUs”]() (2015)

Decoder

Given the score for word k

$$e(k) = w_k^\top z_i + b_k,$$

...we can finally normalize to word probabilities with a softmax.

Probability that the ith word is word k

$$p(w_i = k | \underbrace{w_1, w_2, \ldots, w_{i-1}}_{\text{Previous words}}, \underbrace{h_T}_{\text{Hidden state}}) = \frac{\exp(e(k))}{\sum_j \exp(e(j))}.$$

Bridle, John S. "Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters." NIPS 1989

Decoder

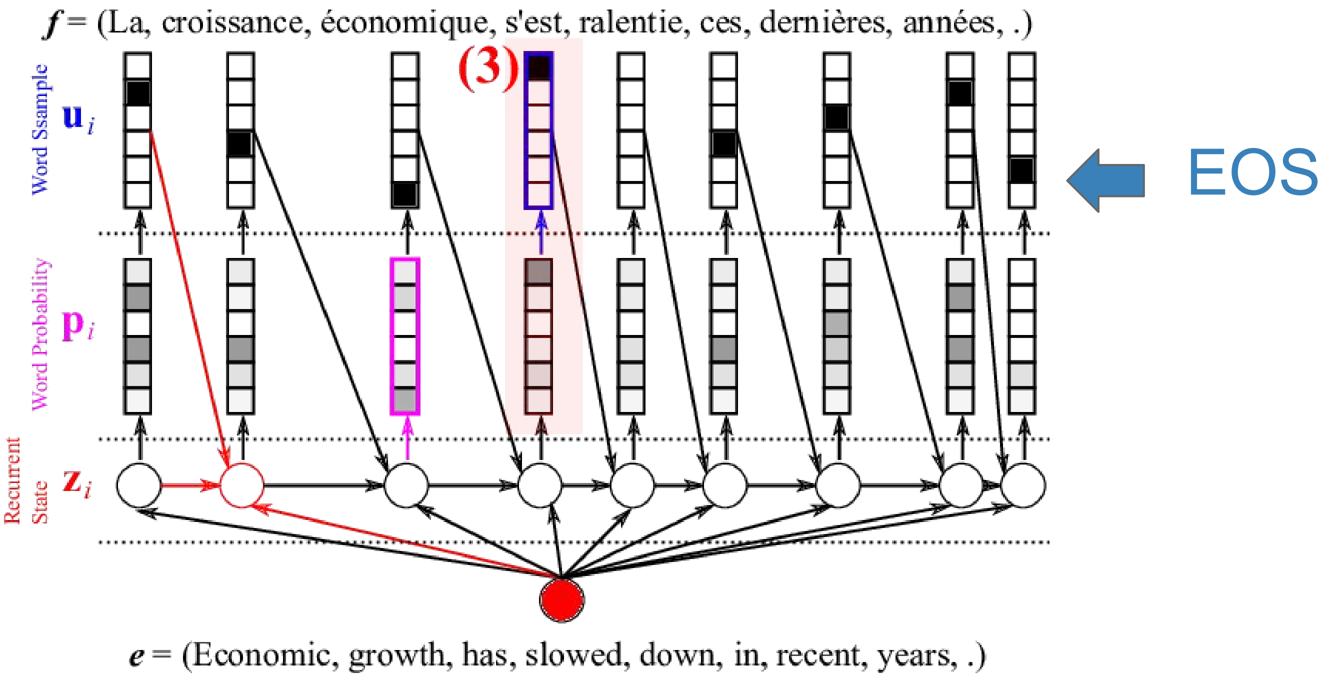
go back to the 1st step…

(1) computing the decoder's internal state
(2) score and normalize target words
(3) select the next word

Kyunghyun Cho, "Introduction to Neural Machine Translation with GPUs" (2015)

Decoder

More words for the decoded sentence are generated until a <EOS> (End Of Sentence) “word” is predicted.



Kyunghyun Cho, [“Introduction to Neural Machine Translation with GPUs”](#) (2015)

Training

Training: Maximum Likelihood Estimation

(1) Prepare the parallel corpus, each sample in the corpus is a pair (X^n, Y^n) of source and target

(2) Given any pair from the corpus, the NMT model can compute the conditional log-probability

$\log P(Y^n|X^n, \theta)$, and the log-likelihood of the whole training corpus:

$$\mathcal{L}(D, \theta) = \frac{1}{N} \sum_{n=1}^{N} \log P(Y^n|X^n, \theta)$$

(3) Maximize this log likelihood function, e.g. using stochastic gradient descent (SGD), Adam, Adadelta, Adagrad.. By using backpropagation

theano.tensor.grad (-loglikelihood, parameters)

Kyunghyun Cho, “Introduction to Neural Machine Translation with GPUs” (2015)

Computational Complexity

1. Source word embeddings: $T \times |V|$ (T source words, $|V|$ unique words)
2. Source embeddings to the encoder: $T \times n_e \times (3 \times n_r)$ (n_e-dim embedding, n_r recurrent units; two gates and one unit for GRU)
3. h_{t-1} to h_t: $T \times n_r \times (3 \times n_r)$
4. Context vector to the decoder: $T \times n_r \times (3 \times n_r)$
5. z_{t-1} to z_t: $T \times n_r \times (3 \times n_r)$
6. The decoder to the target word embeddings: $T' \times n_r \times n_{e'}$ (T' target words, $n_{e'}$-dim target embedding)
7. Target embeddings to the output: $T' \times n_{e'} \times |V'|$ ($|V'|$ target words)
8. Softmax normalization of the output: $T' \times |V'|$

Kyunghyun Cho, "Introduction to Neural Machine Translation with GPUs" (2015)

Why this may not work?

Why this may not work?

We are encoding the entire source sentence into a single context vector

How to solve this?

With the attention-based mechanism…
more details tomorrow

Summary

- Machine Translation is faced as a **sequence-to-sequence** problem
- The source sentence is **encoded** into a fixed length vector and this fixed length vector is **decoded** into the final most probable target sentence
- Only **parallel corpus** and **automatic evaluation** measures are required to train a neural machine translation system

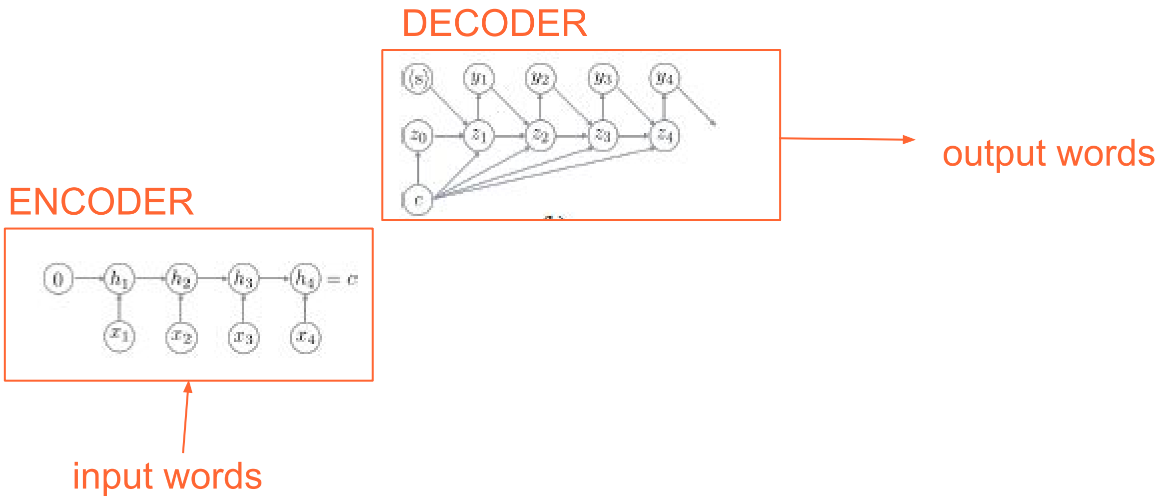
Learn more

Natural Language Understanding with Distributed Representation, Kyunghyun Cho, Chapter 6, 2015 (available in github)

Thanks ! Q&A ?

https://www.costa-jussa.com
marta.ruiz@upc.edu

Another useful image for encoding-decoding



Kyunghyun Cho, “Natural Language Understanding with Distributed Representations” (2015)