Machine Translation
WiSe 2016/2017

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Neural Machine Translation

Dr. Mariana Neves

January 30th, 2017

Overview

- Introduction
- Neural networks
- Neural language models
- Attentional encoder-decoder
- Google NMT

Overview

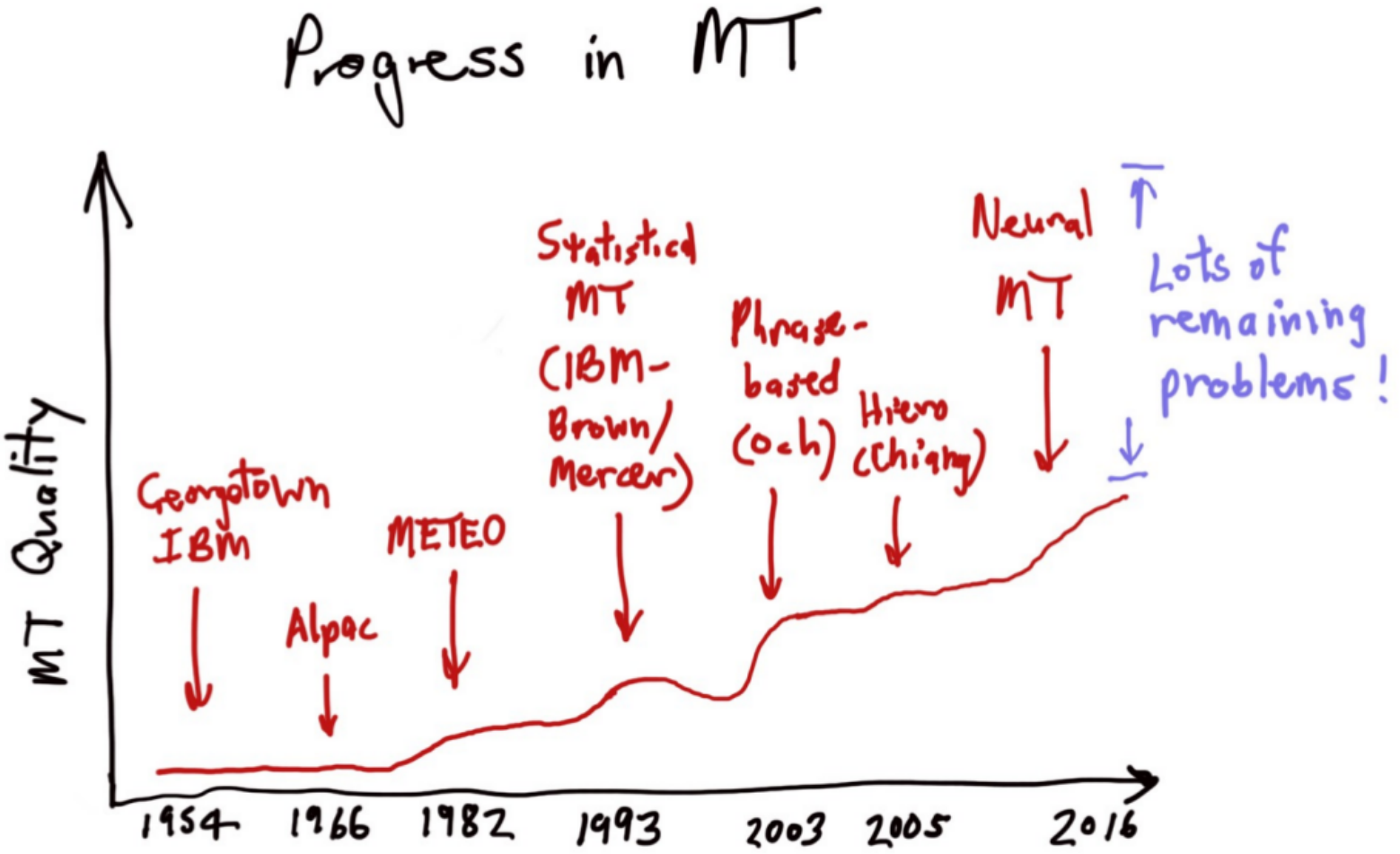
- Introduction
- Neural networks
- Neural language models
- Attentional encoder-decoder
- Google NMT

Neural MT

- „Neural MT went from a fringe research activity in 2014 to the widely-adopted leading way to do MT in 2016.“ (NMT ACL‘16)

- Google Scholar
  - Since 2012: 28,600
  - Since 2015: 22,500
  - Since 2016: 16,100

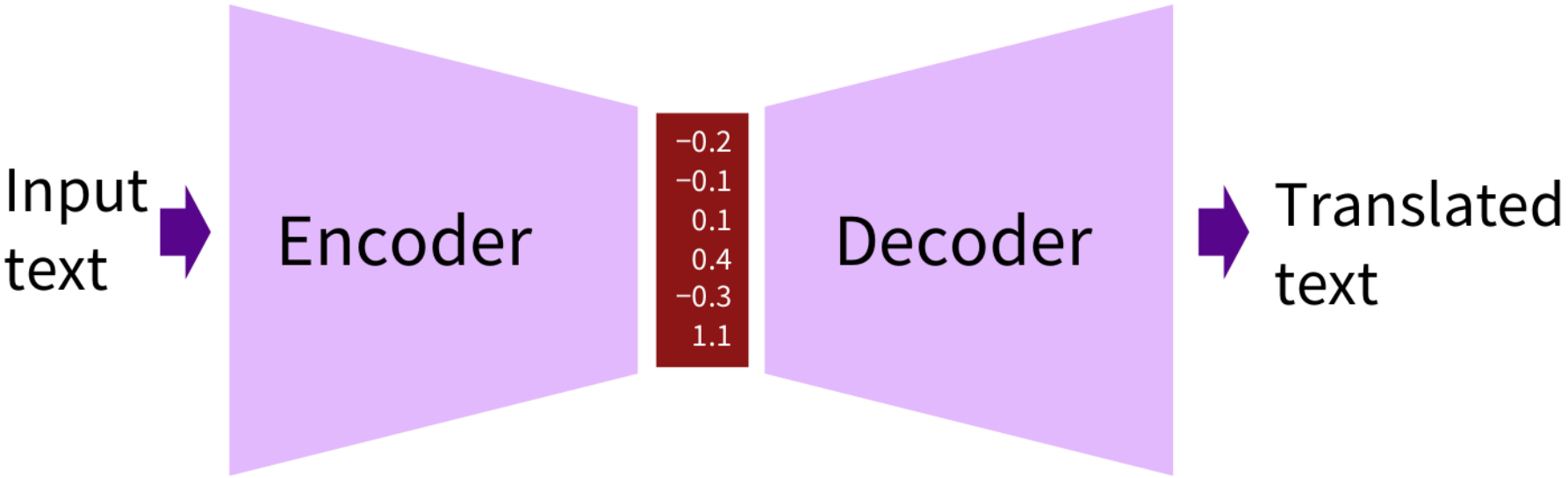
Neural MT



[Picture from NMT ACL16 slides]

Neural MT

- „Neural Machine Translation is the approach of modeling the entire MT process via one big artificial neural network“ (NMT ACL‘16)



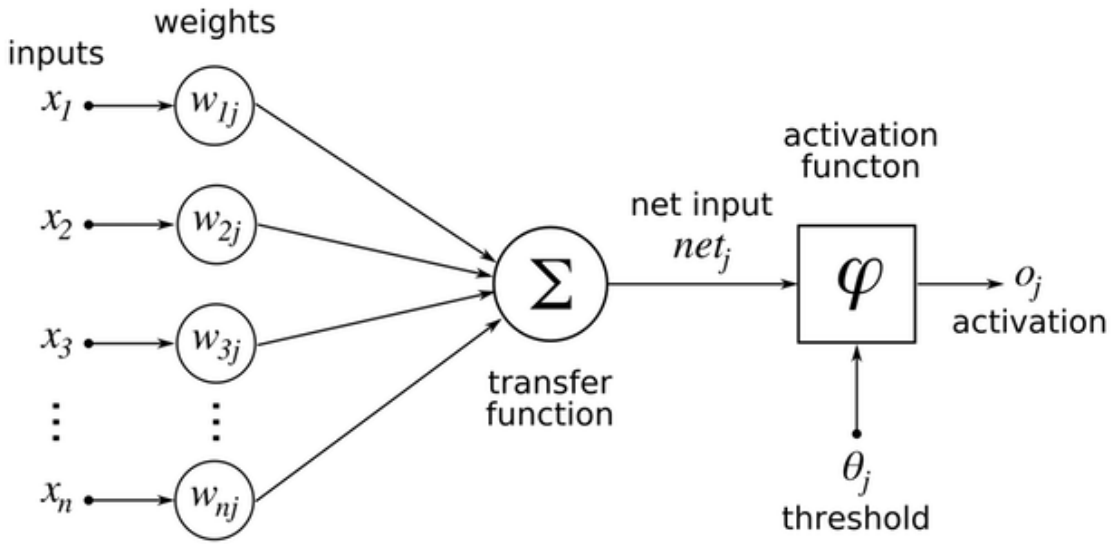
[Picture from NMT ACL16 slides]]

Overview

- Introduction
- Neural networks
- Neural language models
- Attentional encoder-decoder
- Google NMT

Artificial neuron

- Input are the dendrites; Output are the axons
- Activation occurs if the sum of the weighted inputs is higher than a threshold (message is passed)



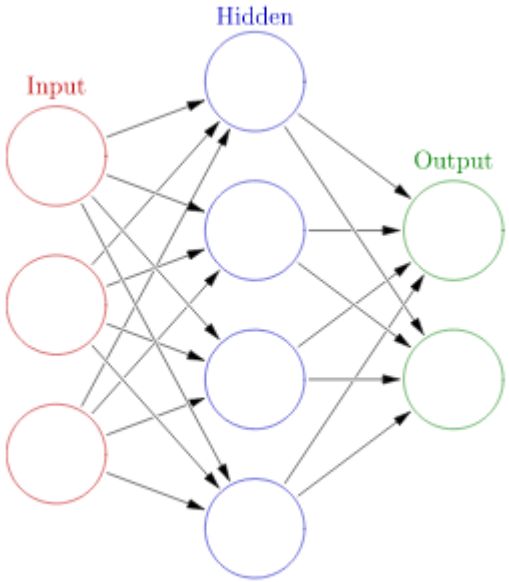
(http://www.innoarchitech.com/artificial-intelligence-deep-learning-neural-networks-explained/)

Artificial neural networks (ANN)

- Statistical models inspired on biological neural networks
- They model and process nonlinear relationships between input and output
- They are based on adptative **weights** and a **cost function**
- Based on optimization techniques, e.g., gradient descent and stochastic gradient descent

Basic architecture of ANNs

- Layers of artificial neurons
- Input layer, hidden layer, output layer
- Overfitting can occur with increasing model complexity



(http://www.innoarchitech.com/artificial-intelligence-deep-learning-neural-networks-explained/)

Deep learning

- Certain types of NN that consume very raw input data

- Data is processed through many layers of nonlinear transformations

Deep learning – feature learning

- Deep learning excels in unspervised feature extraction, i.e., automatic derivation of meaningful features from the input data
- They learn which features are important for a task
- As opposed to feature selection and engineering, usual tasks in machine learning approaches

Deep learning - architectures

- Feed-forward neural networks
- Recurrent neural network
- Multi-layer perceptrons
- Convolutional neural networks
- Recursive neural networks
- Deep belief networks
- Convolutional deep belief networks
- Self-Organizing Maps
- Deep Boltzmann machines
- Stacked de-noising auto-encoders

Overview

- Introduction
- Neural networks
- Neural language models
- Attentional encoder-decoder
- Google NMT

Language Models (LM) for MT

LM for MT

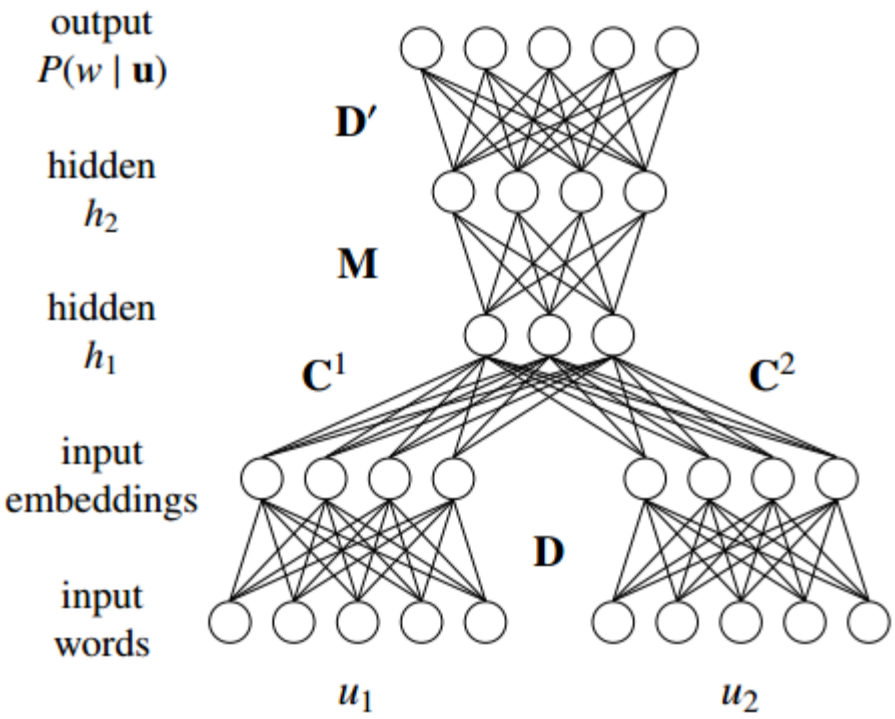
Help with reordering

$$p_{\text{LM}}(\text{the house is small}) > p_{\text{LM}}(\text{small the is house})$$

Help with word choice

$$p_{\text{LM}}(\text{I am going home}) > p_{\text{LM}}(\text{I am going house})$$

N-gram Neural LM with feed-forward NN



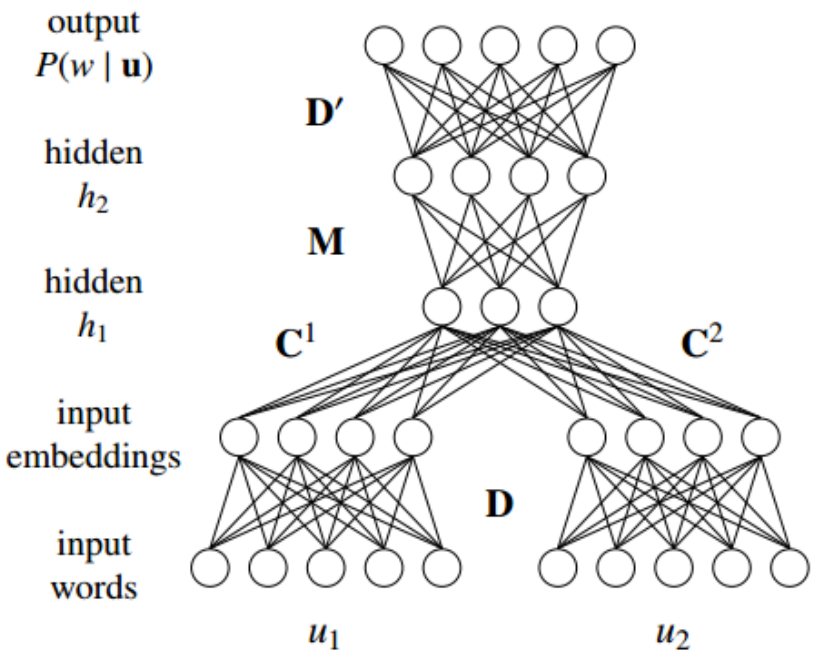
Input as **one-hot representations** of the words in context u (n-1),

where n is the order of the language model

(https://www3.nd.edu/~dchiang/papers/vaswani-emnlp13.pdf)

N-gram Neural LM with feed-forward NN

- Input: context of n-1 previous words
- Output: probability distribution for next word
- Size of input/output: vocabulary size
- One or many hidden layers
- Embedding layer is lower dimensional and dense
  - Smaller weight matrices
  - Learns to map similar words to similar points in the vector space



(https://www3.nd.edu/~dchiang/papers/vaswani-emnlp13.pdf)

One-hot representation

- Corpus: „the man runs.“
- Vocabulary = {man,runs,the,.}
- Input/output for p(runs|the man)

$$x_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad x_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad y_{true} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Softmax function

- It normalize the output vectors to probability distribution (sum=1)
- Its computational cost is linear to vocabulary size
- When combined with stochastic gradient descend, it minimizes cross-entropy (perpexity)

$$p(y=j|x)=\frac{e^{x^T w_j}}{\sum_{k=1}^{K} e^{x^T w_k}}$$

$x^T w$ is the inner product of x (sample vector) and w (weight vector)

Softmax function

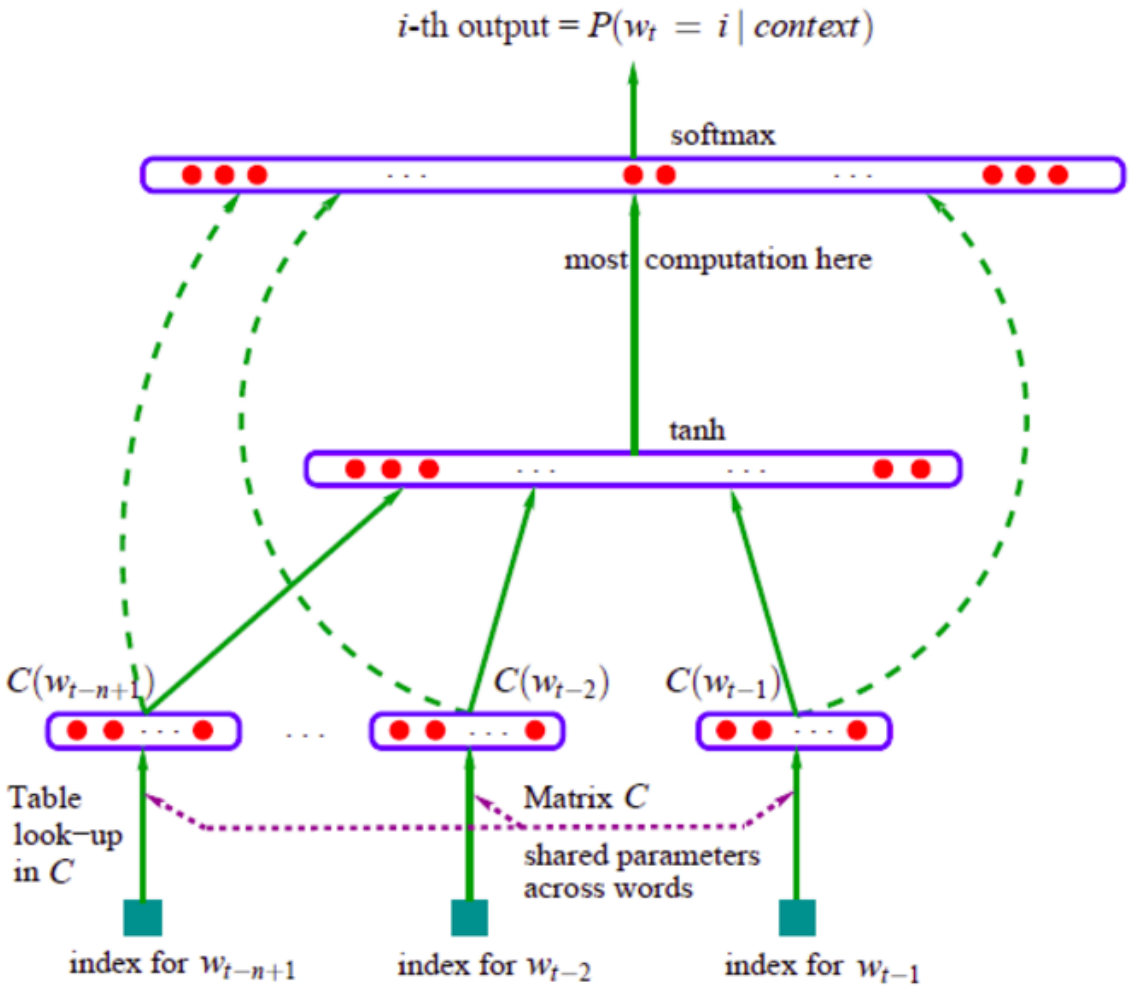
- Example:
  - input = [1,2,3,4,1,2,3]
  - softmax = [0.024, 0.064, 0.175, 0.475, 0.024, 0.064, 0.175]

- The output has most of its weight where the '4' was in the original input.
- The function highlights the largest values and suppress values which are significantly below the maximum value.

(https://en.wikipedia.org/wiki/Softmax_function)

(http://sebastianruder.com/word-embeddings-1/)

Classical neural language model (Bengio et al. 2003)



Feed-forward neural language model (FFNLM) in SMT

- One more feature in the log-linear phrase-based model

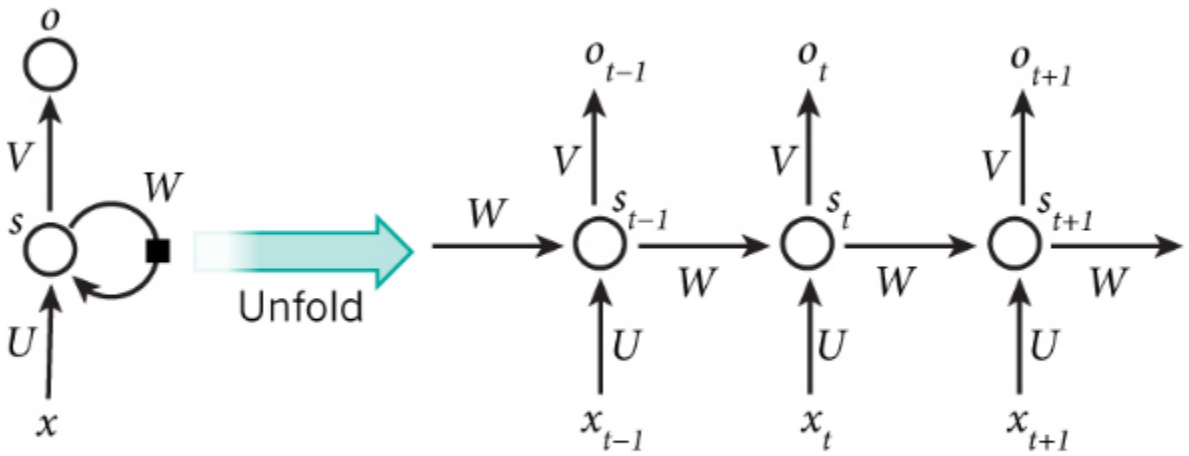
$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

$$p(e, a|f) = \exp(\lambda_\phi \sum_{i=1}^{I} \log \phi(\bar{f}_i|\bar{e}_i) +$$
$$\lambda_d \sum_{i=1}^{I} \log d(a_i - b_{i-1} - 1) +$$
$$\lambda_{LM} \sum_{i=1}^{|\mathbf{e}|} \log p_{LM}(e_i|e_1 \ldots e_{i-1}))$$

(http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/)

Recurrent neural networks language model (RNNLM)

- Recurrent neural networks (RNN) is a class of NN in which connections between the units form a directed cycle
- It makes use of sequential information
- It does not assume independence between input and output

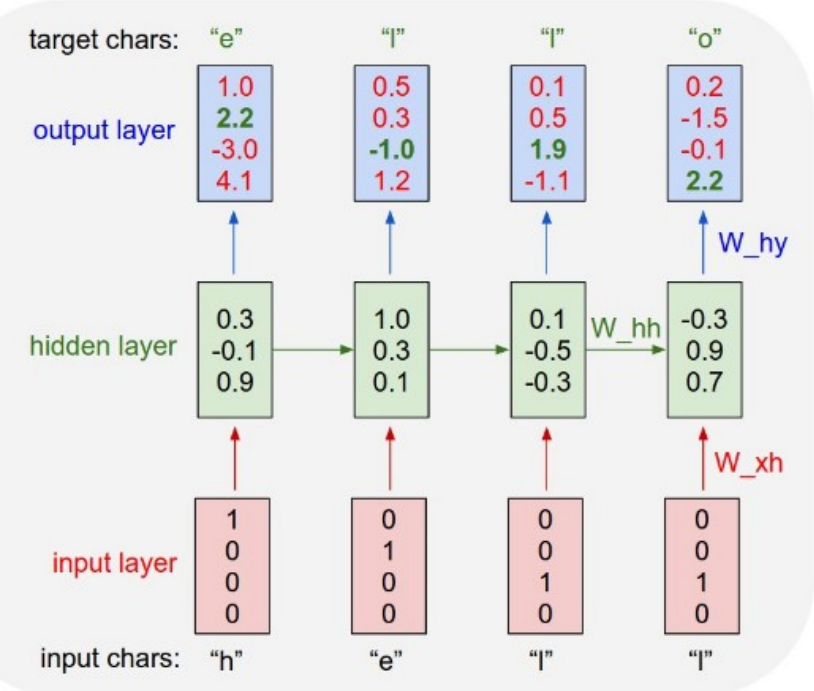


A recurrent neural network and the unfolding in time of the computation involved in its forward computation. Source: Nature

(http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

RNNLM

- Condition on arbitrarly long contexts
- No Markov assumption
- It reads one word at a time, updates network incrementally



Overview

- Introduction
- Neural networks
- Neural language models
- Attentional encoder-decoder
- Google NMT

Translation modelling

- Source sentence S of length m: $x_1, \ldots, x_m$
- Target sentence T of length n: $y_1, \ldots, y_n$

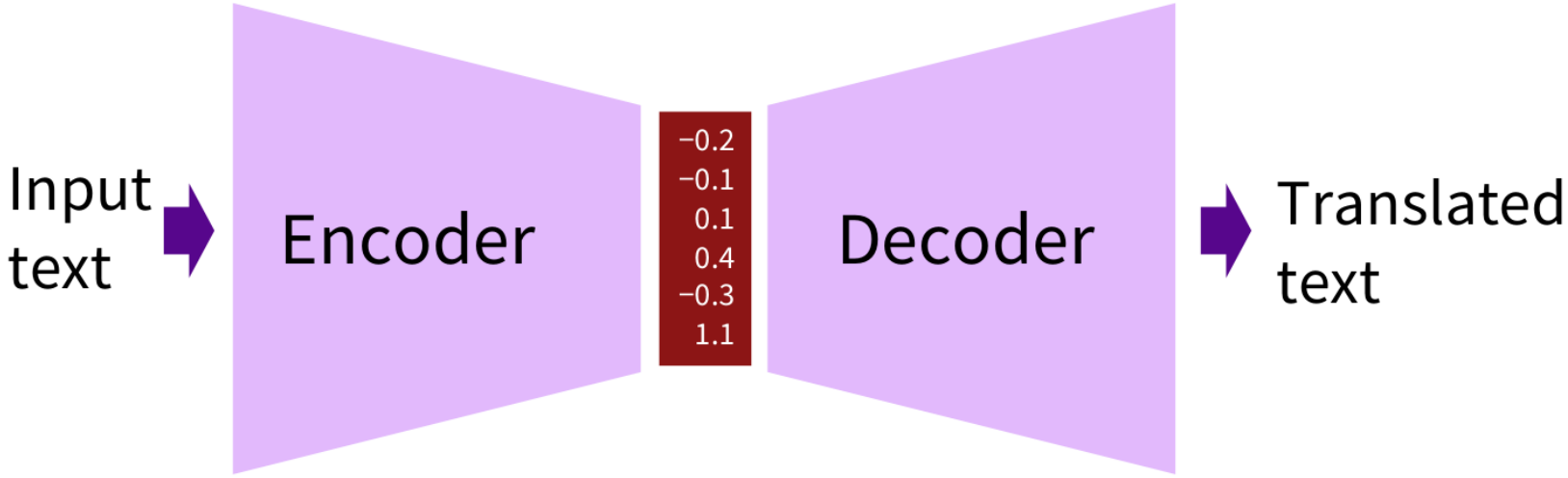
$$T^* = \arg\max_t P(T|S)$$

$$P(T|S) = P(y_1, \ldots, y_n | x_1, \ldots, x_m)$$

$$P(T|S) = \prod_{i=1}^{n} P(y_i | y_0, \ldots, y_{i-1}, x_1, \ldots, x_m)$$

Encoder-Decoder

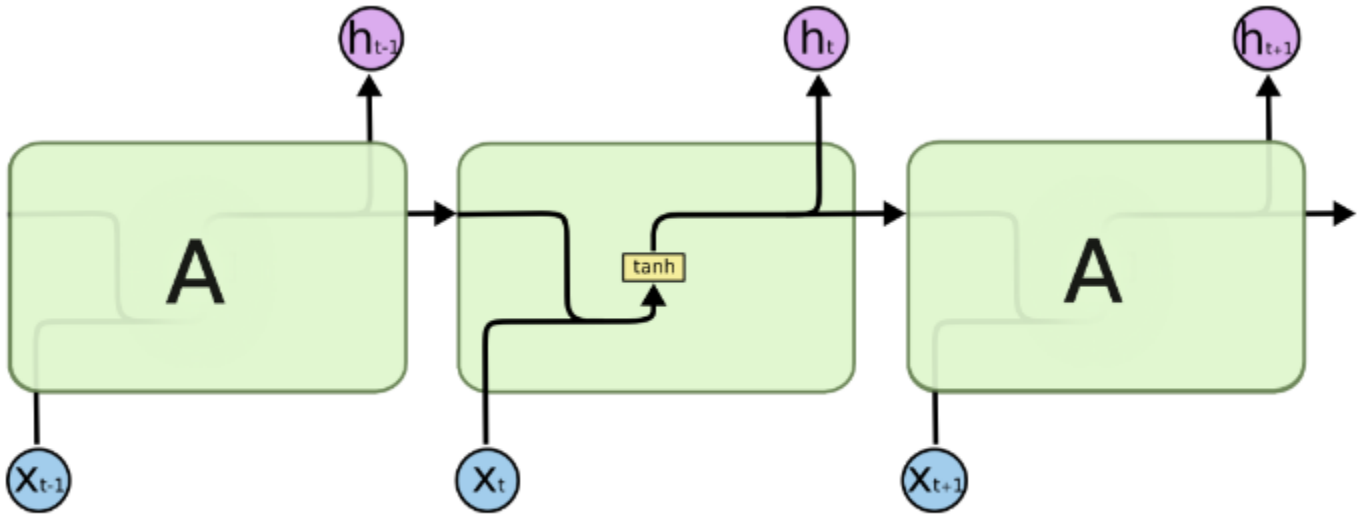
- Two RNNs (usually LSTM):
  - **encoder** reads input and produces hidden state representations
  - **decoder** produces output, based on last encoder hidden state



[Picture from NMT ACL16 slides]

Long short-term memory (LSTM)

- It is a special kind of RNN
- It connects previous information to the present task
- It is capable to learn long-term dependencies

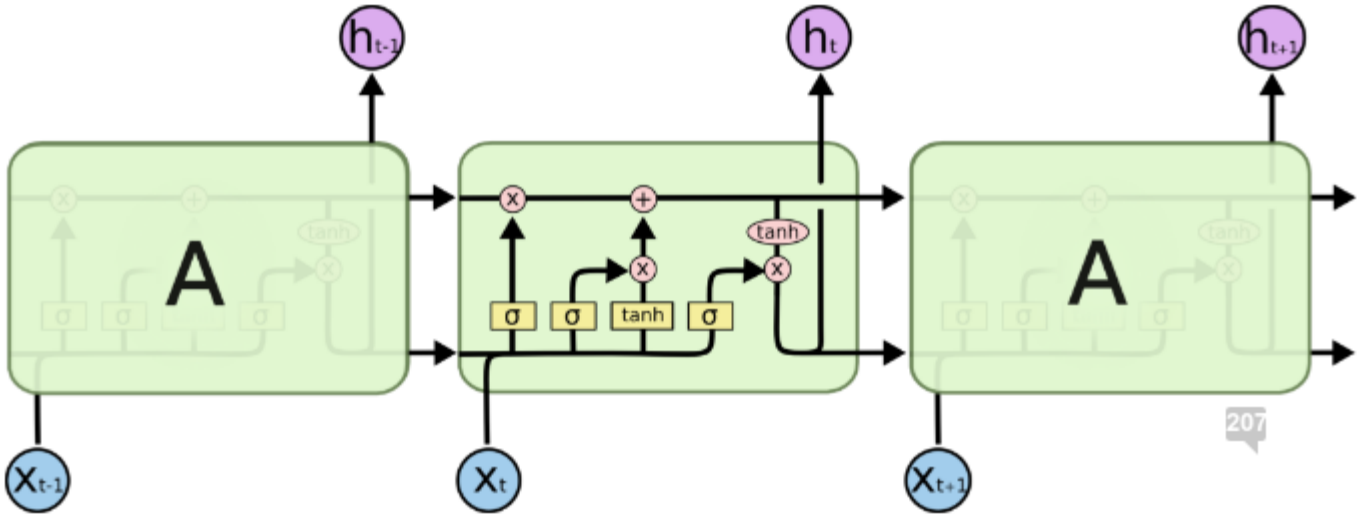


The repeating module in a standard RNN contains a single layer.

(http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Long short-term memory (LSTM)

- LSTMs have four interating layers
- But there are many variations of the architecture

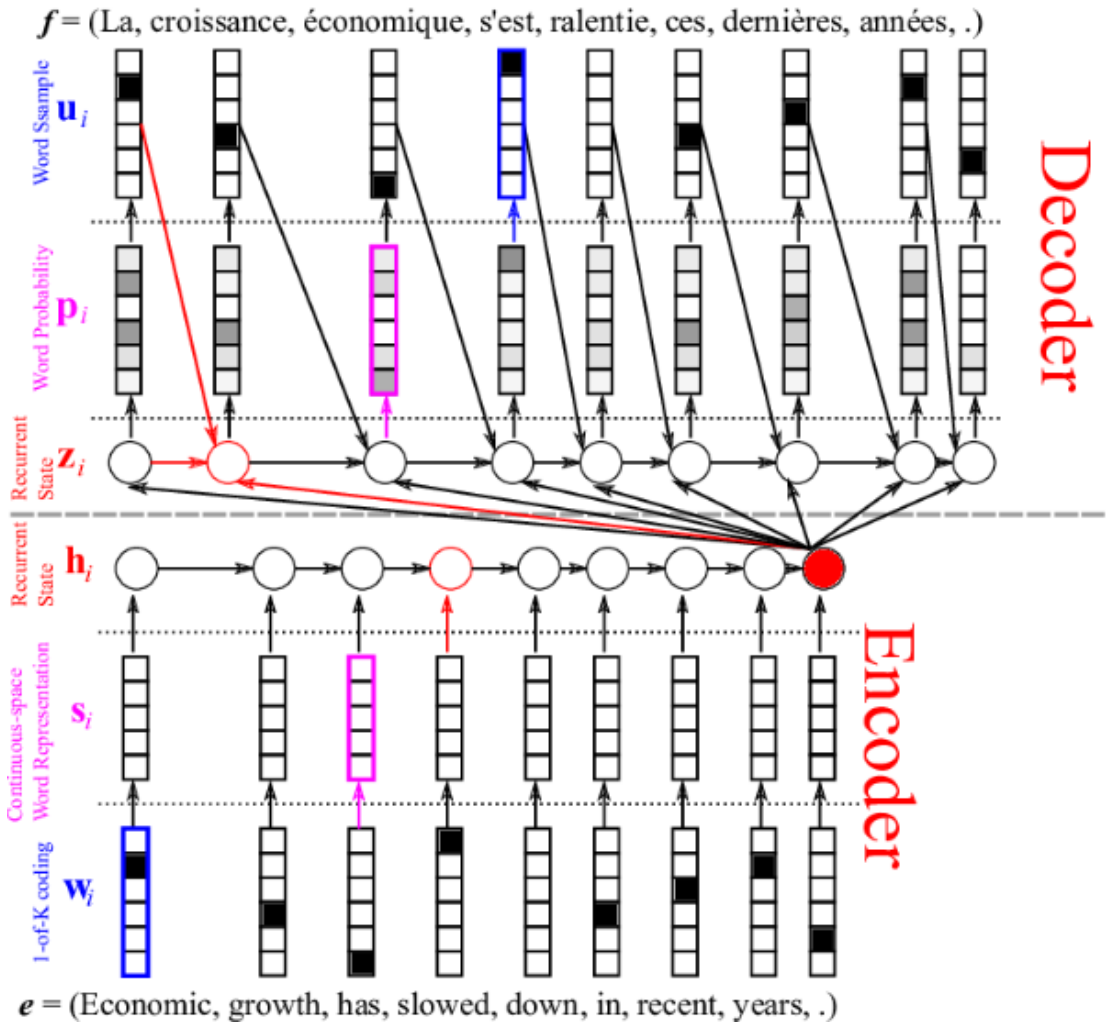


The repeating module in an LSTM contains four interacting layers.

(http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Encoder-Decoder

- Encoder-decoder are learned jointly
- Supervision signal from parallel corpora is backpropagated



(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2/)

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2/)

The Encoder (continuous-space representation)

- The encoder linearly projects the 1-of-K coded vector w_i with a matrix E which has as many columns as there are words in the source vocabulary and as many rows as you want (typically, 100 – 500.)

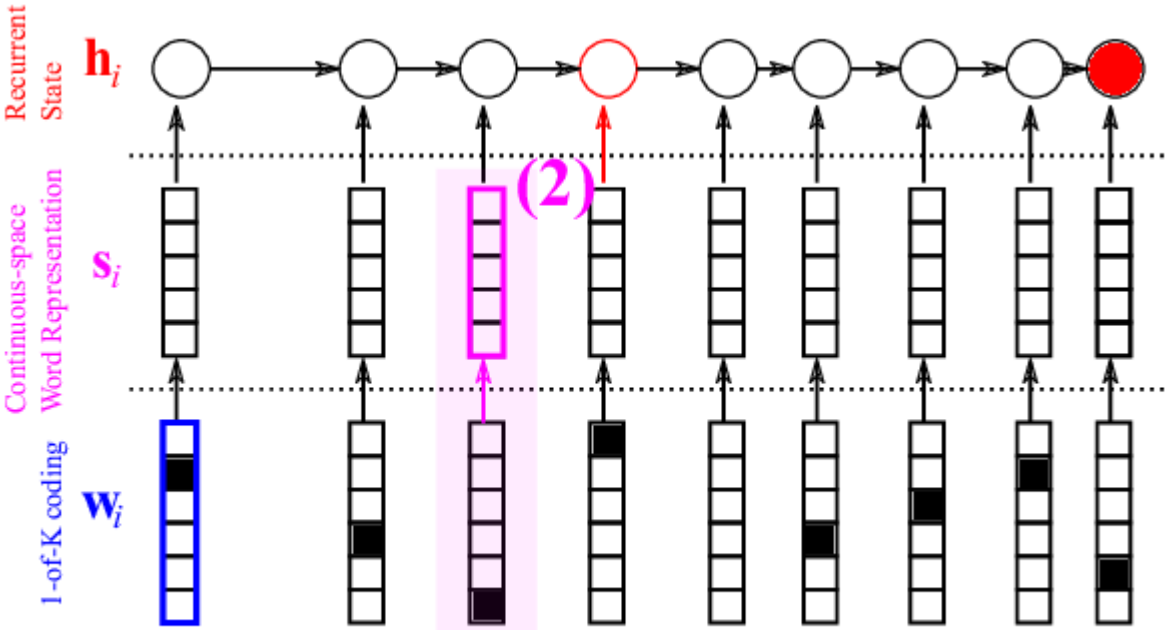
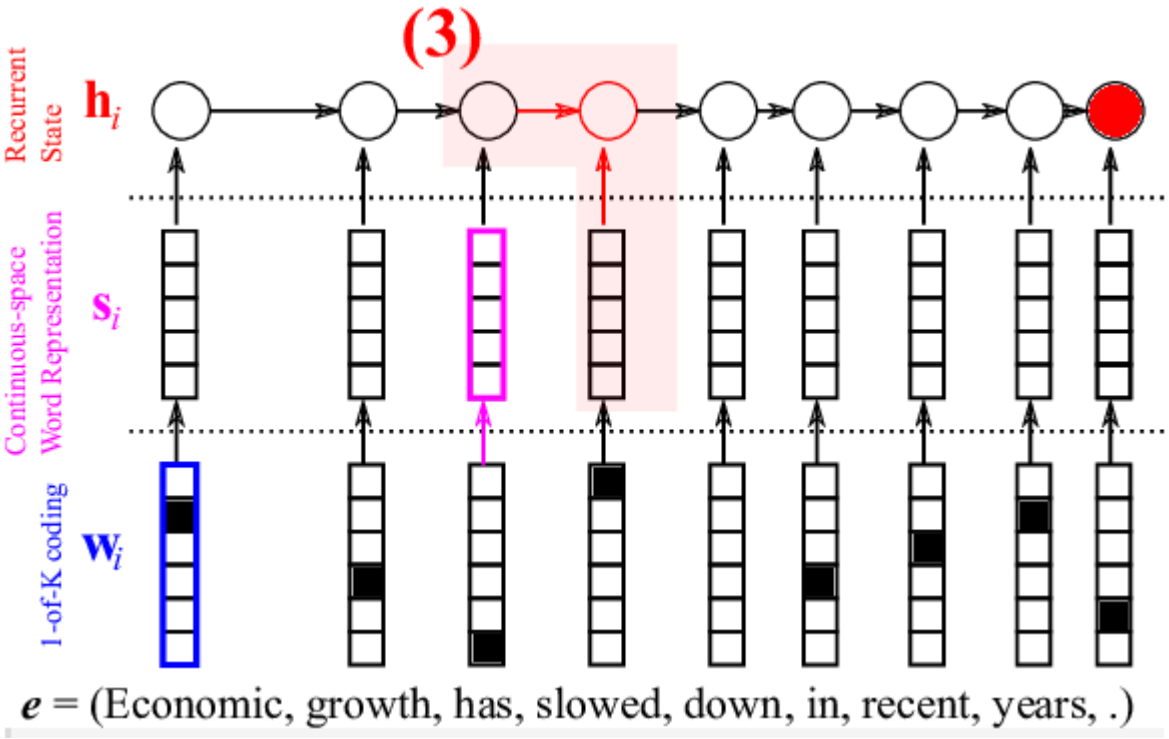


Figure 4. Step 2: A one-hot vector to a continuous-space representation.

The Encoder (summary vector)

- Last encoder hidden state summarizes source sentence
- But quality decreases for long sentences (fixed-size vector)



(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2/)

The Encoder (summary vector)

- Projection to 2D using Principal Componnet Analysis (PCA)

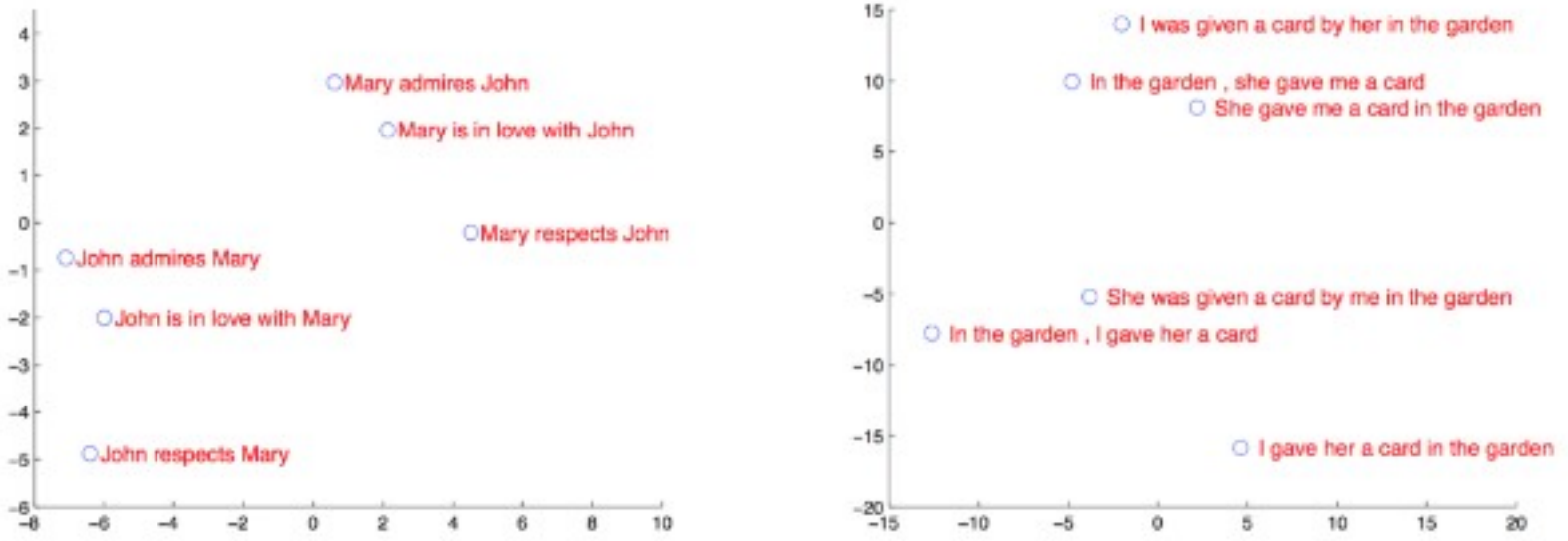


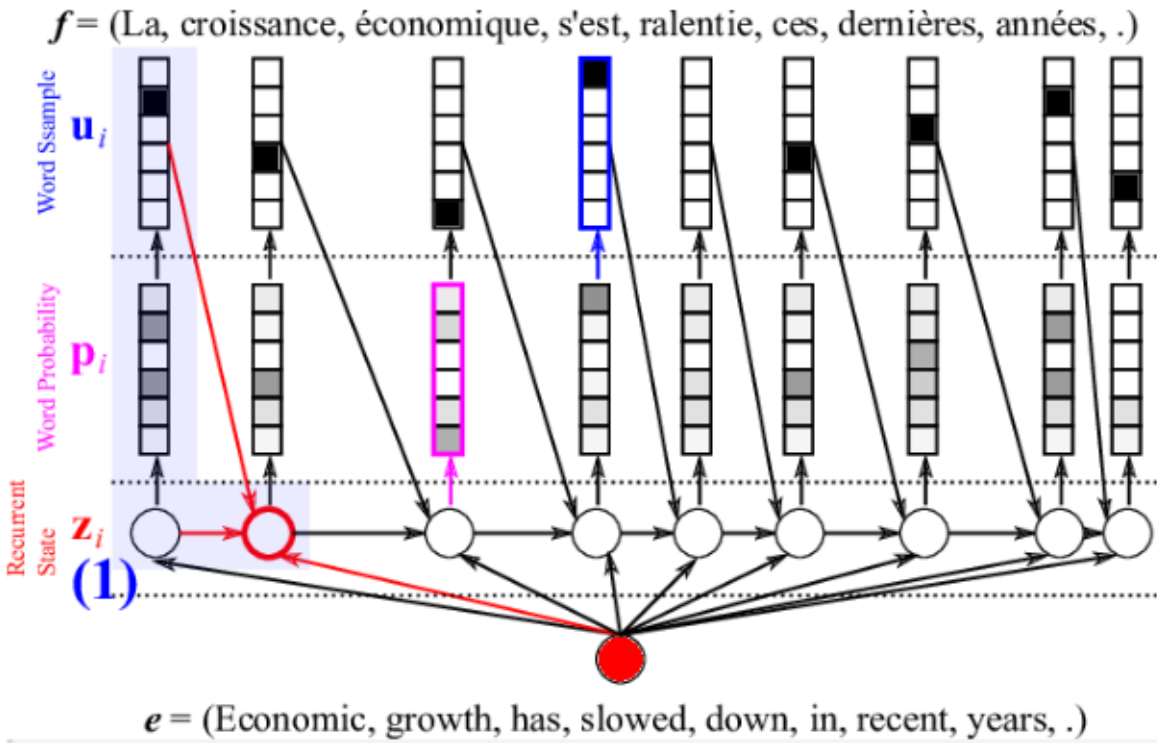
Figure 6. 2-D Visualization of Sentence Representations from [Sutskever et al., 2014]. Similar sentences are close together in summary-vector space.

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2/)

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2/)

The Decoder

- The inverse of the encoder
- Based on the softmax function



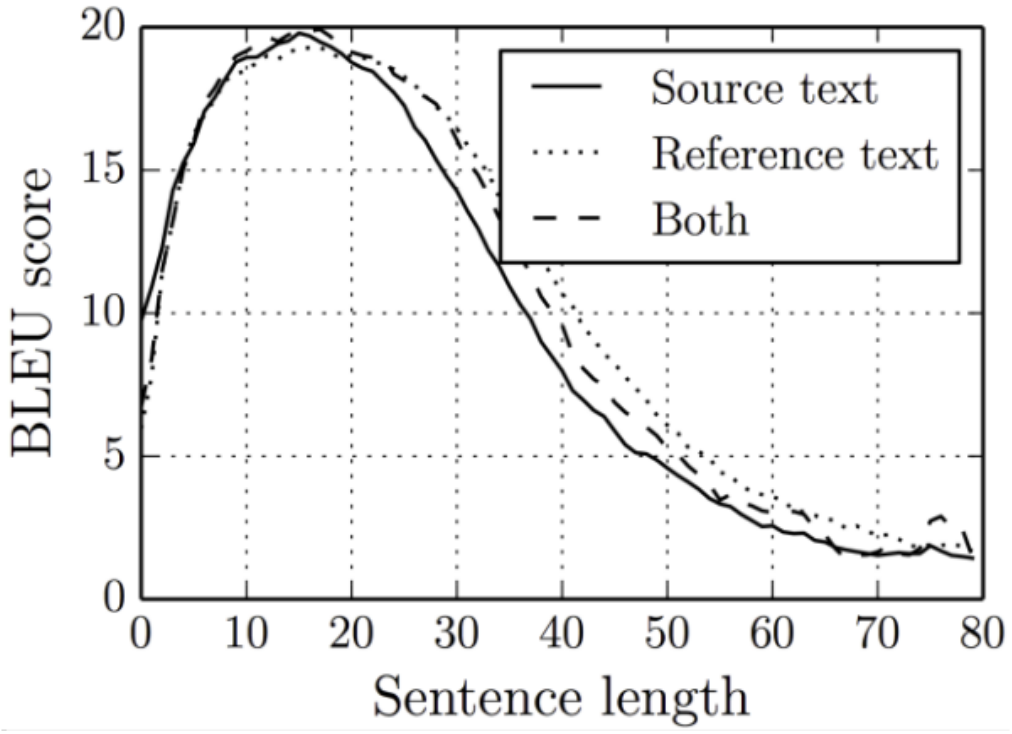
Problem with simple E-D architectures

- Fixed-size vector from which the decoder needs to generate a full translation
- The context vector must contain every signgle detail of the source sentence
- The dimensionality of the contect vector must be large enough that a sentence of any length can be compressed

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

Problem with simple E-D architectures

- Large models are necessary to cope with large sentences



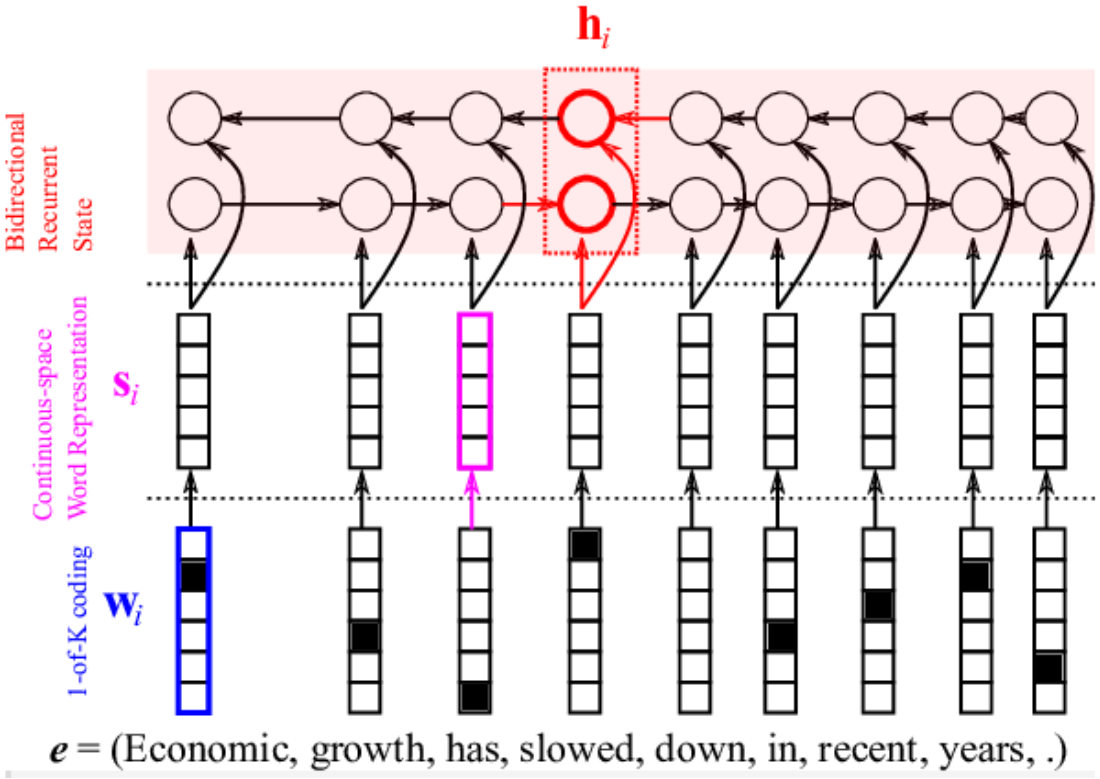
(experiments with small models)

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2/)

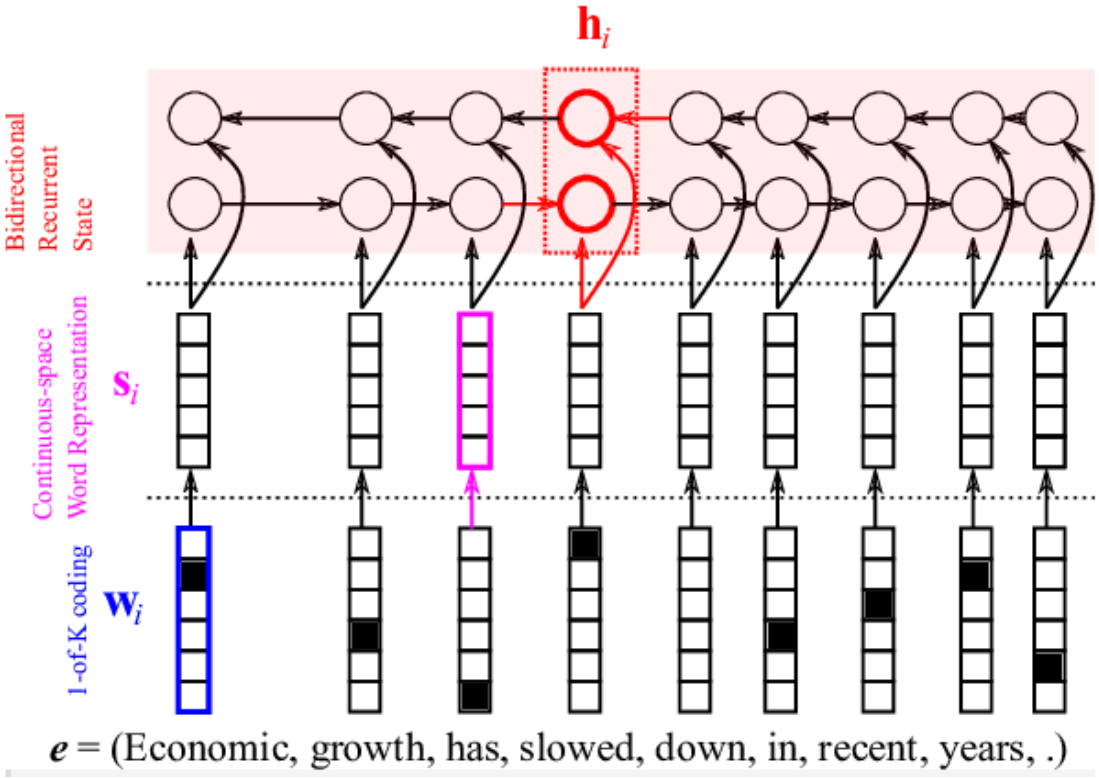
Bidirectional recurrent neural network (BRNN)

- Use a memory with as many banks as source words, instead of a fixed-size context vector
- BRNN = forward RNN + backwards RNN



Bidirectional recurrent neural network (BRNN)

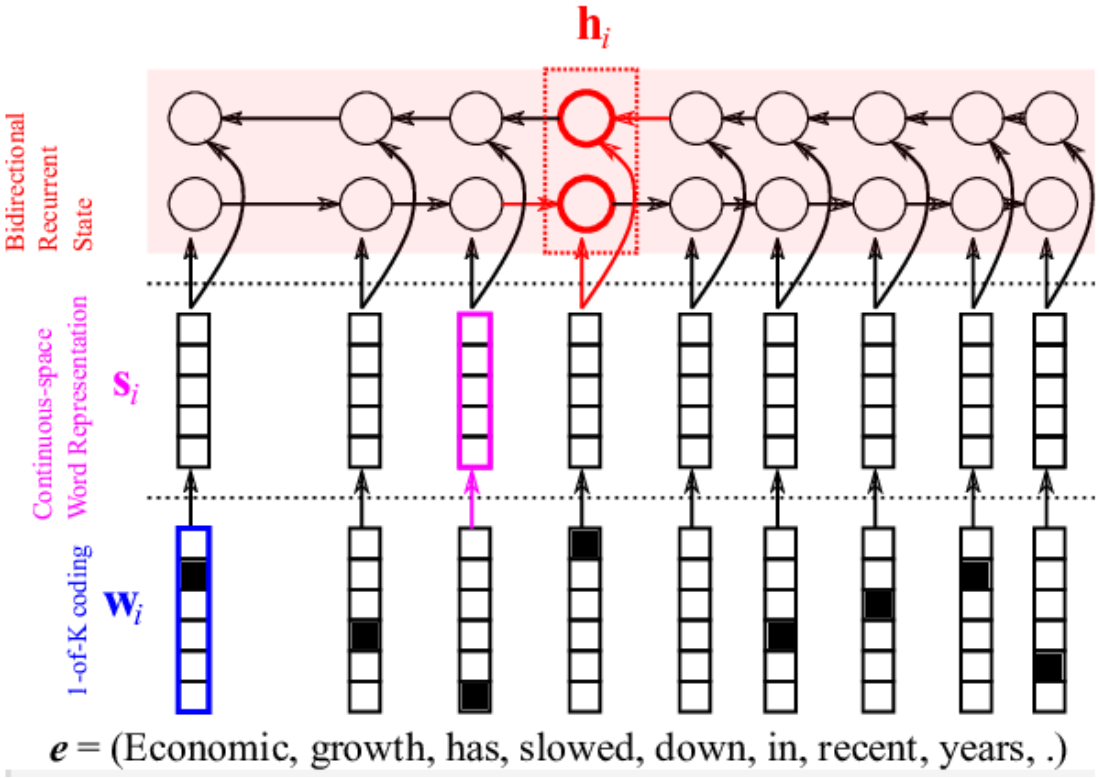
- At any point, the forward and backward vectors summarizes a whole input sentece



(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

Bidirectional recurrent neural network (BRNN)

- This mechanism allows storage of a source sentence as a variable-length representation



(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

Soft Attention mechanism

- It is a small NN that takes as input the previous decoder‘s hidden state (what has been translated)

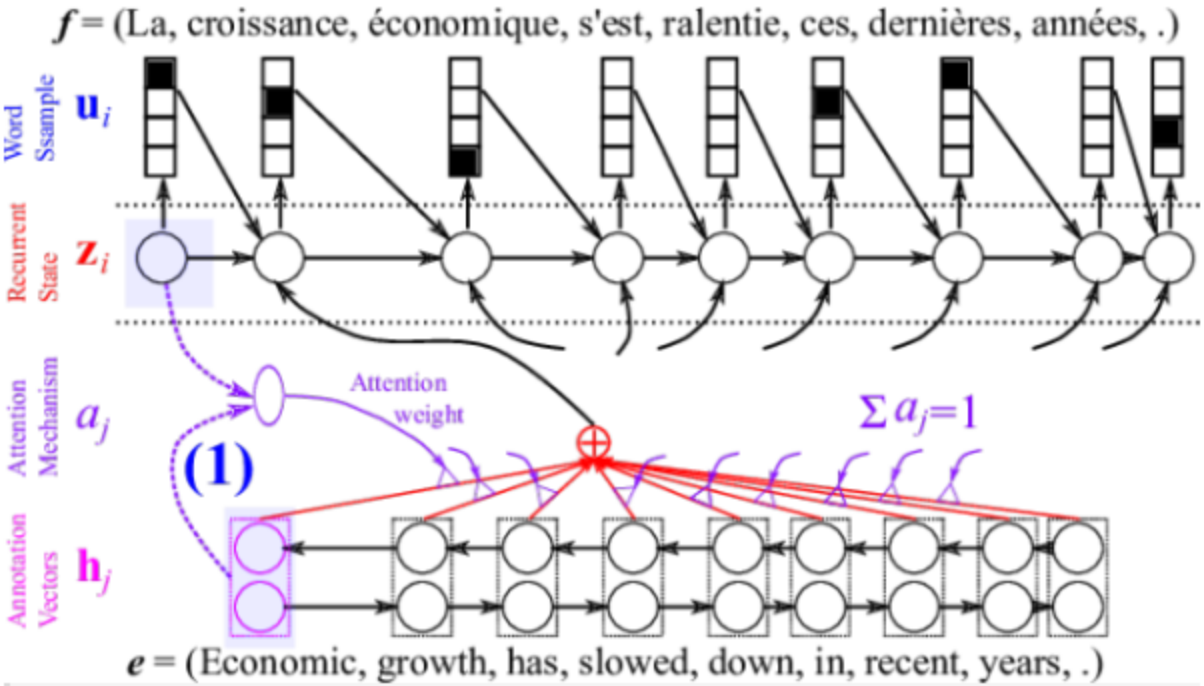


Figure 3. Attention Mechanism takes into consideration what has been translated and one of the source words.

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

Soft Attention mechanism

- It contains one hidden layer and outputs a scalar
- Normalization (to sum up to 1) is done with softmax function

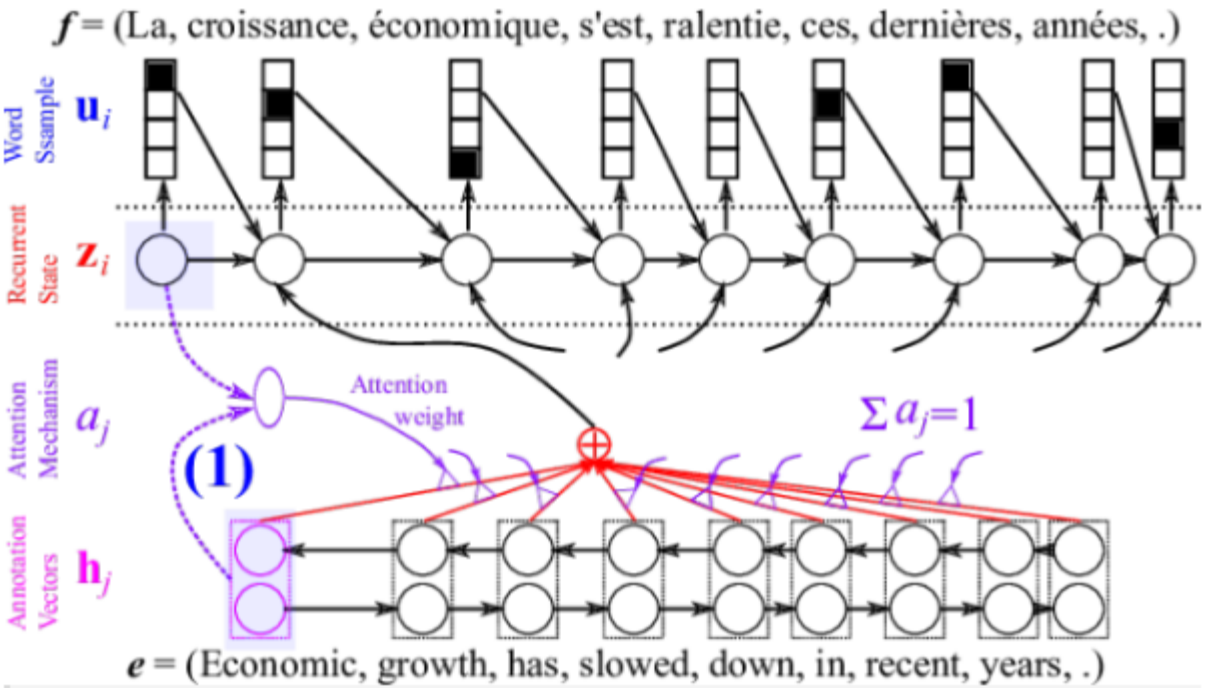


Figure 3. Attention Mechanism takes into consideration what has been translated and one of the source words.

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

Soft Attention mechanism

- The model learn attention (alignment) between two languages

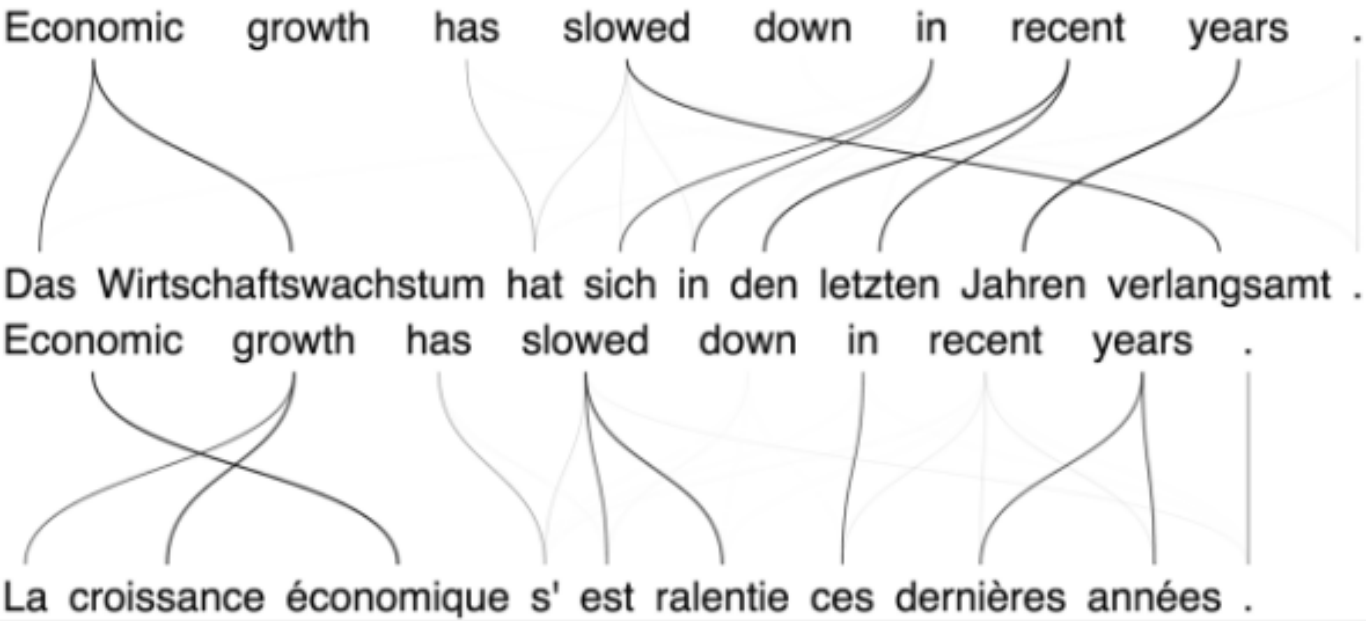


Figure 6. Sample translations made by the neural machine translation model with the soft-attention mechanism. Edge thicknesses represent the attention weights found by the attention model.

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

Soft Attention mechanism

- With this mechanism, the quality of the translation does not drop as the sentence length increases

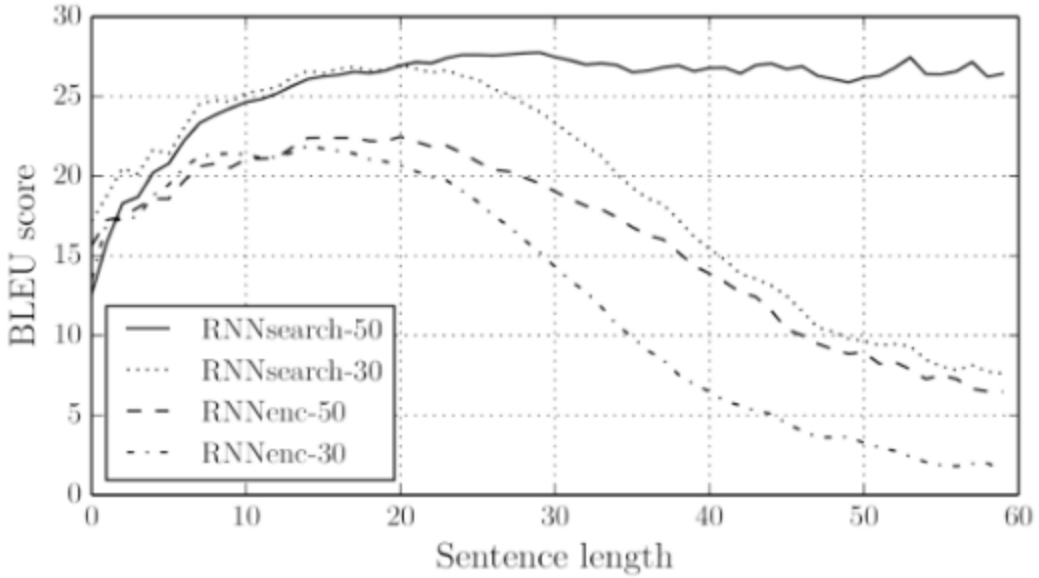


Figure 7. RNNsearch-50 is a neural machine translation model with the attention mechanism trained on all the sentence pairs of length at most 50.

(https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/)

Overview

- Introduction
- Neural networks
- Neural language models
- Attentional encoder-decoder
- Google NMT

Google Multilingual NMT system (Nov/16)

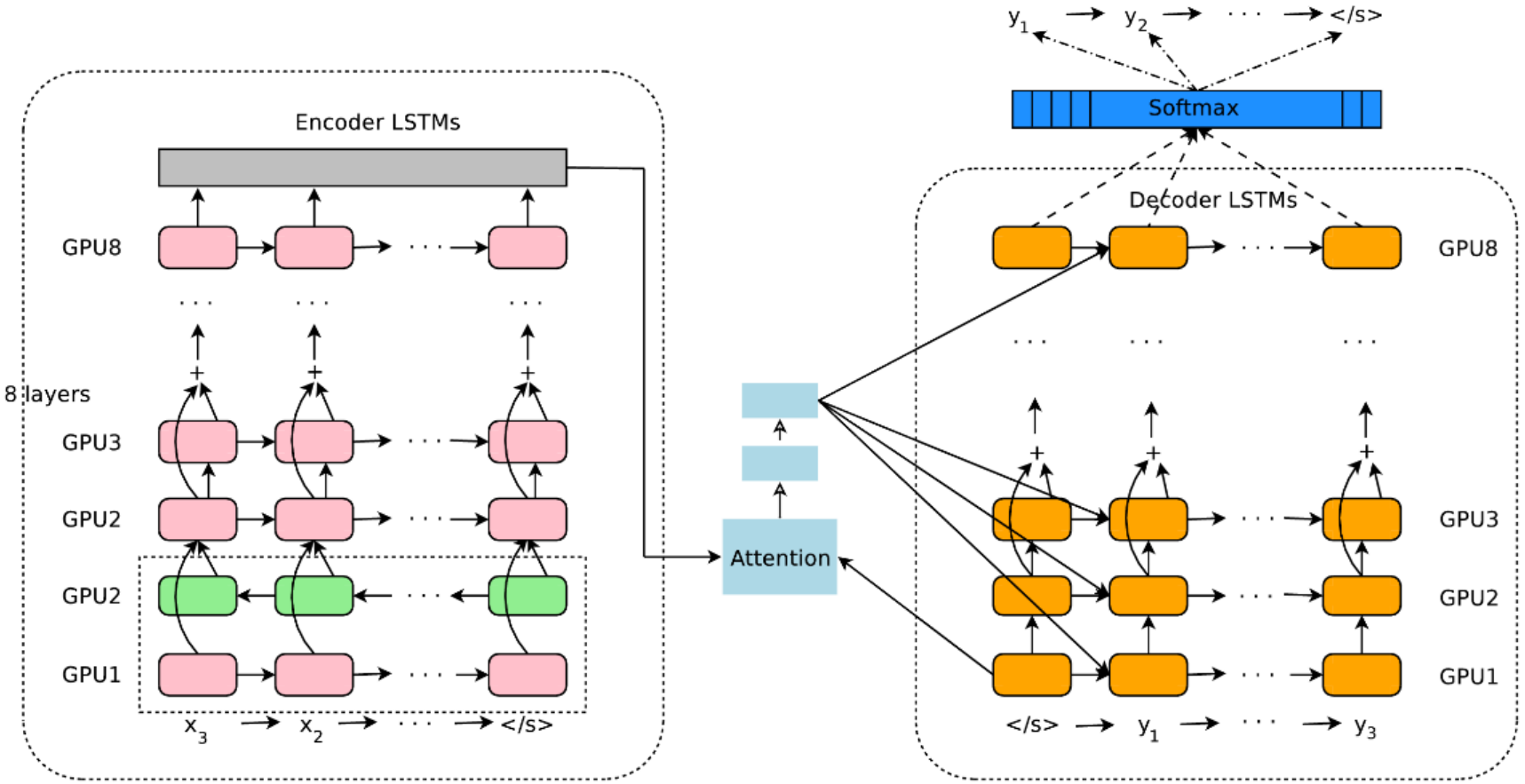
- Simplicity:
  - Single NMT model to translate between multiple languages, instead of many models (100^2)
- Low-resource language improvement:
  - Improve low-resource language pair by mixing with high-resource languages into a single model
- Zero-shot translation:
  - It learns to perform implicit bridging between language pairs never seen explicitly during training

(https://arxiv.org/abs/1611.04558)

(https://arxiv.org/abs/1609.08144)

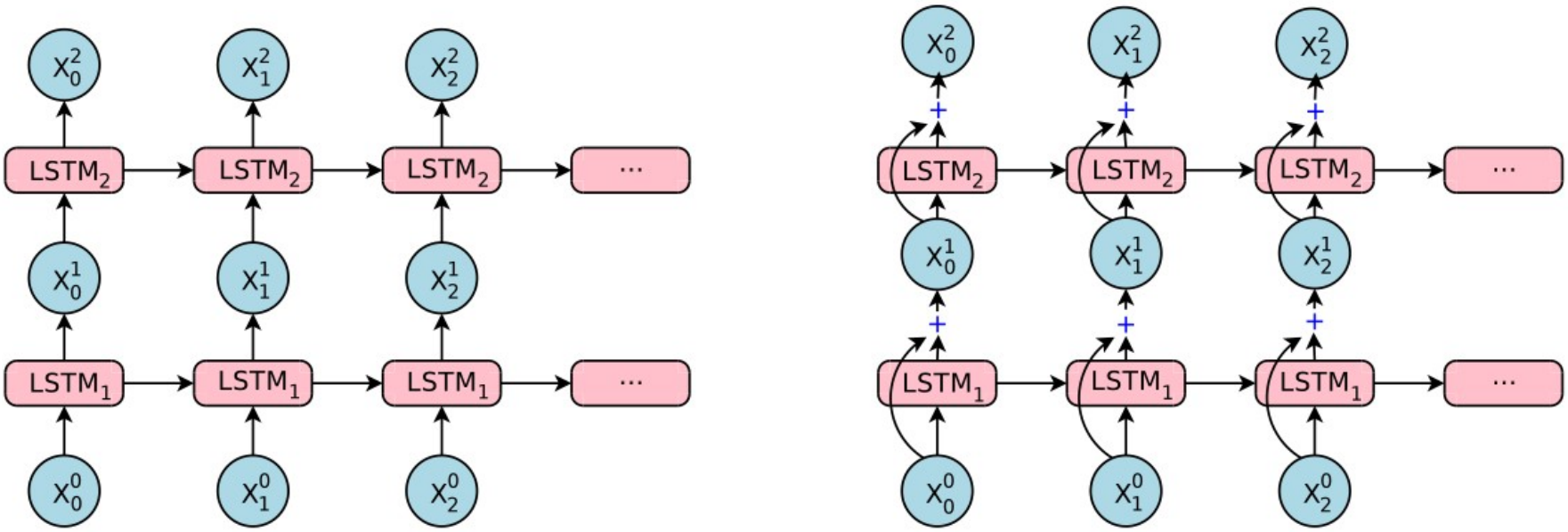
Google NMT system (Sep-Oct/16)

- Deep LSTM network with 8 encoder and 8 decoder layers



Google NMT system (Sep-Oct/16)

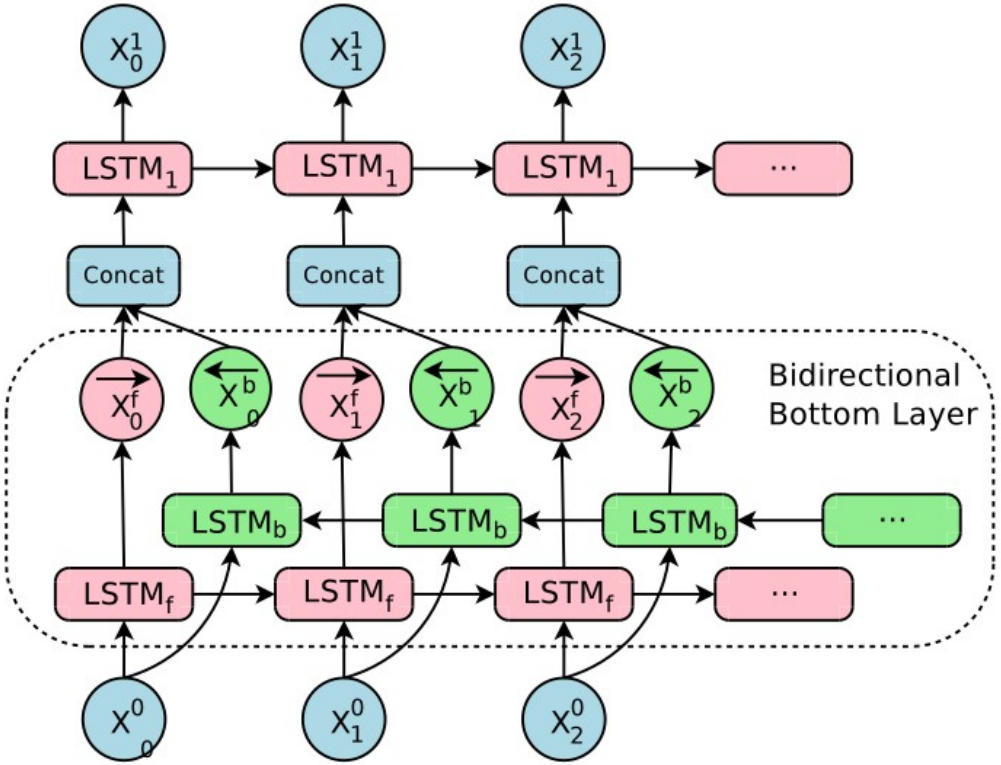
- Normal LSTM (left) vs. stacked LSTM (right) with residual connections



(https://arxiv.org/abs/1609.08144)

Google NMT system (Sep-Oct/16)

- Output from $LSTM_f$ and $LSTM_b$ are first concatenated and then fed to the next LSTM layer $LSTM_1$



(https://arxiv.org/abs/1609.08144)

Google NMT system (Sep-Oct/16)

- Wordpiece model (WPM) implementation initially developed to solve a Japanese/Korean segmentation problem

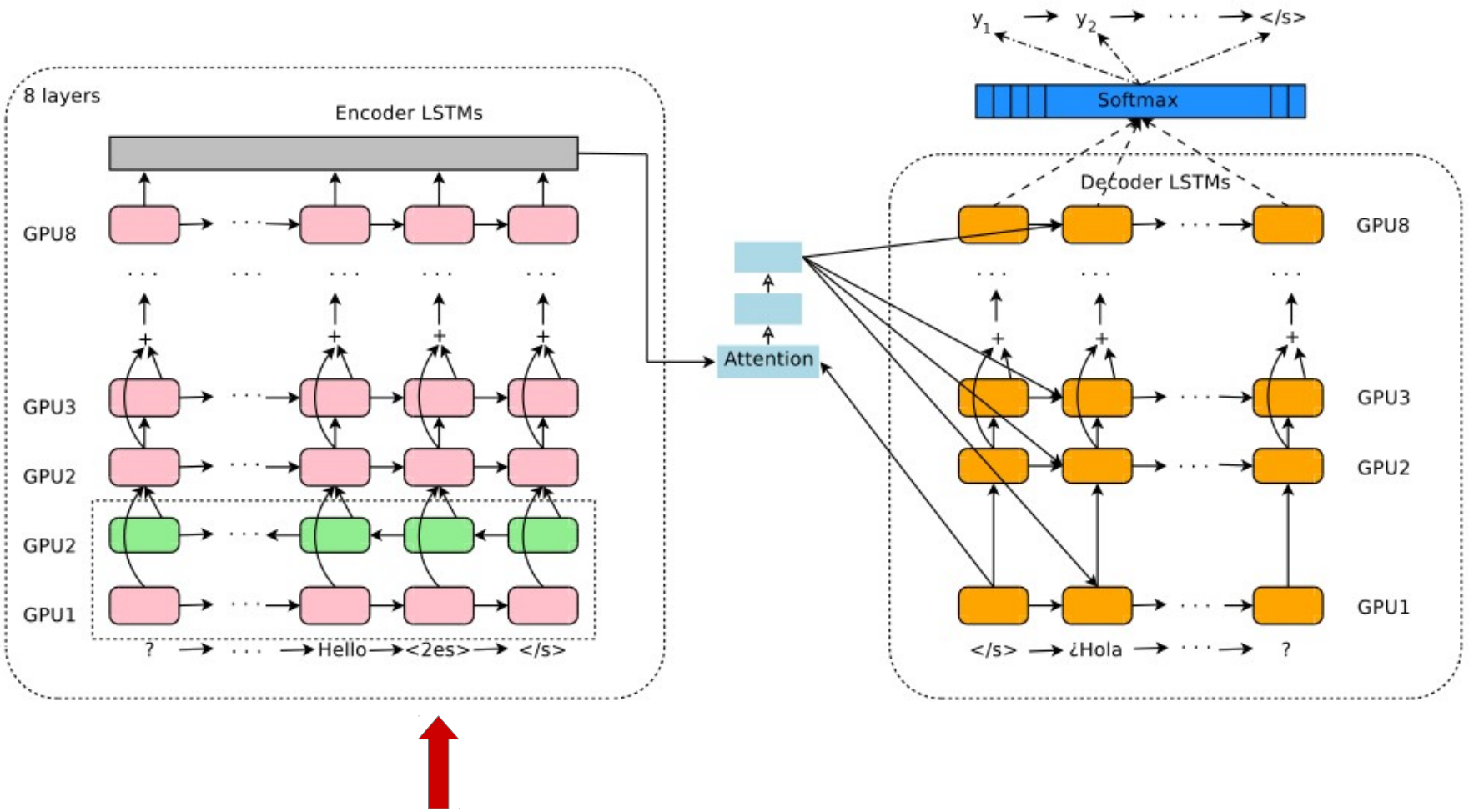
- Data-driven approach to maximize the language-model likelihood of the training data

- **Word**: Jet makers feud over seat width with big orders at stake
- **wordpieces**: _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

("_" is a special character added to mark the beginning of a word.)

(https://arxiv.org/abs/1609.08144)

Google Multilingual NMT system (Nov/16)



(https://arxiv.org/abs/1611.04558)

Google Multilingual NMT system (Nov/16)

- Introduction of an artificial token at the beginning of the input sentence to indicate the target language the model should translate to.

Hello, how are you? -> ¿Hola como estás?

<2es> Hello, how are you? -> ¿Hola como estás?

(https://arxiv.org/abs/1611.04558)

Google Multilingual NMT system (Nov/16)

- Experiments: Many to one

Table 1: Many to One: BLEU scores on various data sets for single language pair and multilingual models.

Model	Single	Multi	Diff
WMT German→English (oversampling)	30.43	30.59	+0.16
WMT French→English (oversampling)	35.50	35.73	+0.23
WMT German→English (no oversampling)	30.43	30.54	+0.11
WMT French→English (no oversampling)	35.50	36.77	+0.27
Prod Japanese→English	23.41	23.87	+0.46
Prod Korean→English	25.42	25.47	+0.05
Prod Spanish→English	38.00	38.73	+0.73
Prod Portuguese→English	44.40	45.19	+0.79

(https://arxiv.org/abs/1611.04558)

Google Multilingual NMT system (Nov/16)

- Experiments: One to many

Table 2: One to Many: BLEU scores on various data sets for single language pair and multilingual models.

Model	Single	Multi	Diff
WMT English→German (oversampling)	24.67	24.97	+0.30
WMT English→French (oversampling)	38.95	36.84	-2.11
WMT English→German (no oversampling)	24.67	22.61	-2.06
WMT English→French (no oversampling)	38.95	38.16	-0.79
Prod English→Japanese	23.66	23.73	+0.07
Prod English→Korean	19.75	19.58	-0.17
Prod English→Spanish	34.50	35.40	+0.90
Prod English→Portuguese	38.40	38.63	+0.23

(https://arxiv.org/abs/1611.04558)

Google Multilingual NMT system (Nov/16)

- Experiments: Many to many

Table 3: Many to Many: BLEU scores on various data sets for single language pair and multilingual models.

Model	Single	Multi	Diff
WMT English→German (oversampling)	24.67	24.49	-0.18
WMT English→French (oversampling)	38.95	36.23	-2.72
WMT German→English (oversampling)	30.43	29.84	-0.59
WMT French→English (oversampling)	35.50	34.89	-0.61
WMT English→German (no oversampling)	24.67	22.61	-2.06
WMT English→French (no oversampling)	38.95	38.16	-0.79
WMT German→English (no oversampling)	30.43	29.22	-1.21
WMT French→English (no oversampling)	35.50	35.93	+0.43
Prod English→Japanese	23.66	23.12	-0.54
Prod English→Korean	19.75	19.73	-0.02
Prod Japanese→English	23.41	22.86	-0.55
Prod Korean→English	25.42	24.76	-0.66
Prod English→Spanish	34.50	34.69	+0.19
Prod English→Portuguese	38.40	37.25	-1.15
Prod Portuguese→English	44.40	44.02	-0.38
Prod Spanish→English	38.00	37.65	-0.35

(https://arxiv.org/abs/1611.04558)

Google Multilingual NMT system (Nov/16)

- Experiments: Zero-Shot translation

Table 5: Portuguese→Spanish BLEU scores using various models.

	Model	BLEU
(a)	PBMT bridged	28.99
(b)	NMT bridged	30.91
(c)	NMT Pt→Es	31.50
(d)	Model 1 (Pt→En, En→Es)	21.62
(e)	Model 2 (En↔{Es, Pt})	24.75
(f)	Model 2 + incremental training	31.77

(https://arxiv.org/abs/1611.04558)

Summary

- Very brief introduction to neural networks
- Neural language models
  - One-hot representations (1-of-K coded vector)
  - Softmax function
- Neural machine translation
  - Recurrent NN; LSTM
  - Encoder and Decoder
  - Soft attention mechanism (BRNN)
- Google MT
  - Architecture and multilingual experiments

Suggested reading

- Artificial Intelligence, Deep Learning, and Neural Networks Explained: http://www.innoarchitech.com/artificial-intelligence-deep-learning-neural-networks-explained/
- Introduction to Neural Machine Translation with GPUs: https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-with-gpus/

- Neural Machine Translation slides, ACL‘2016: https://sites.google.com/site/acl16nmt/
- Neural Machine Translation slides (Univ. Edinburgh) http://statmt.org/mtma16/uploads/mtma16-neural.pdf