Neural Machine Translation

Arul Selvam Periyasamy

July 1, 2017

Rheinische Friedrich-Wilhelms-Universität Bonn
Seminar: Natural Language Processing

Agenda

- Introduction to Machine Translation

Agenda

- Introduction to Machine Translation
- Statistical Phrase-Based Translation

Agenda

- Introduction to Machine Translation
- Statistical Phrase-Based Translation
- Introduction to Deep Learning

Agenda

- Introduction to Machine Translation
- Statistical Phrase-Based Translation
- Introduction to Deep Learning
- Neural Machine Translation

Motivation

- Translation: The process of translating words or text from one language into another (OED).

Motivation

- Translation: The process of translating words or text from one language into another (OED).
- Machine Translation: Translation carried out by a computer (OED).

Motivation

- Translation: The process of translating words or text from one language into another (OED).
- Machine Translation: Translation carried out by a computer (OED).
- Why do we need it?

Motivation

- Translation: The process of translating words or text from one language into another (OED).
- Machine Translation: Translation carried out by a computer (OED).
- Why do we need it?
- Do I need to convince that we need machine translation?

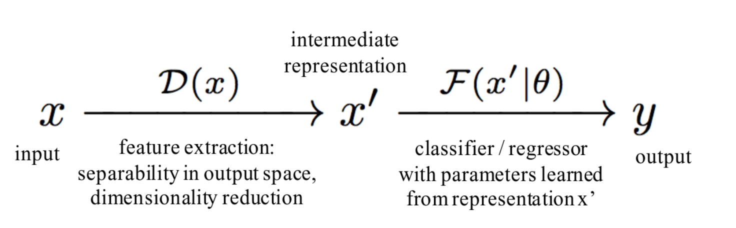
Machine Translation

In a probabilistic perspective, machine translation can be formulated the problem of finding a target sentence y that maximizes the conditional probability of y from a given source sentence x.

$$arg\, max_y\ p(x|y)$$

Introduction to Deep Learning

Machine Learning (Supervised)



Traditional Supervised learning

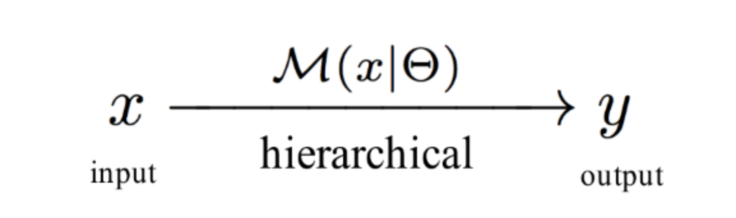
Deep Learning (Supervised)

$$\underset{\text{input}}{x} \xrightarrow[\text{hierarchical}]{\mathcal{M}(x|\Theta)} \underset{\text{output}}{y}$$

Deep learning

- Hierarchical representations of features.

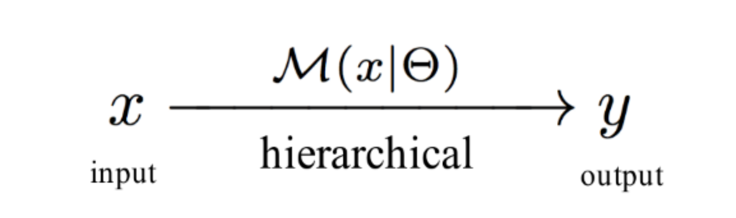
Deep Learning (Supervised)



Deep learning

- Hierarchical representations of features.
- Joint learning of representation.

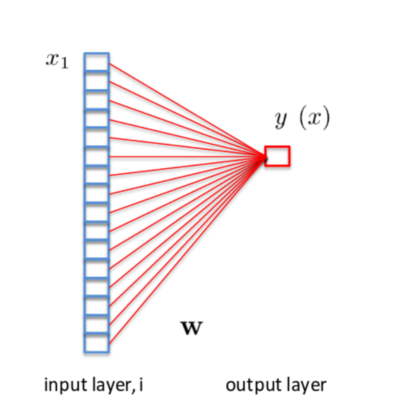
Deep Learning (Supervised)



Deep learning

- Hierarchical representations of features.
- Joint learning of representation.
- Increased levels of abstraction.

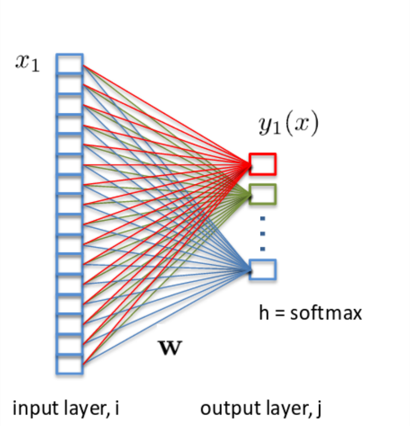
Perceptron



A perceptron (close to a biological neuron)

$$y(x) = f(W^T x)$$

Logistic Regression



A perceptron (A collection of perceptrons)

Logistic Regression

Binary classification:

$$P(y = 1|x) = h_w(x) = \frac{1}{1 + exp(-W^T x)}$$

$$P(y = 0|x) = 1 - h_w(x) = 1 - P(y = 1|x)$$

Logistic Regression

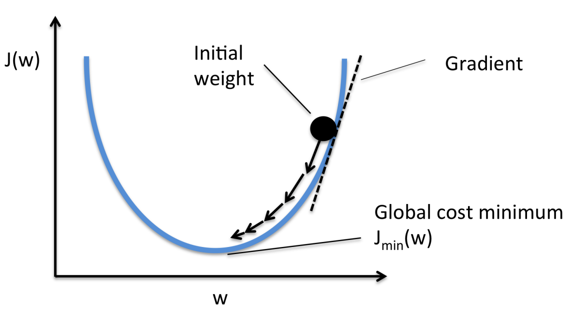
Cost function:

$$J(w) = -\sum_i (y^i log(h_w(x^i)) + (1 - y^i) log(1 - h_w(x^i)))$$

Learning Weights : Gradient Descent

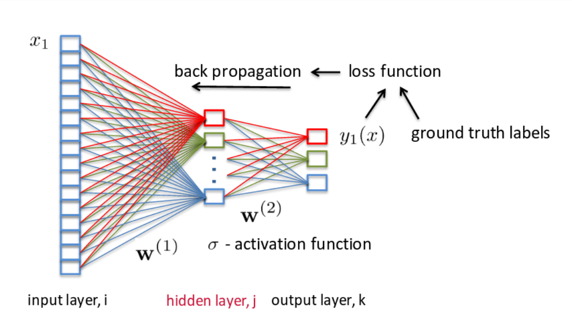
$$\nabla_w J(w) = \frac{\partial J(w)}{\partial w_j} = \sum_i x_j^i (h_w(x^i) - y^i)$$

Gradient Descent



Update weights in the direction of negative gradient.

Multi Layer Perceptron

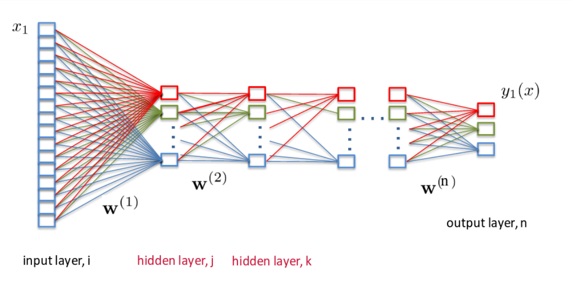


Multiple layers of perceptron

Learning weights: Same as before but apply chain rule.

$$\frac{\partial x}{\partial y} = \frac{\partial x}{\partial z} * \frac{\partial z}{\partial y}$$

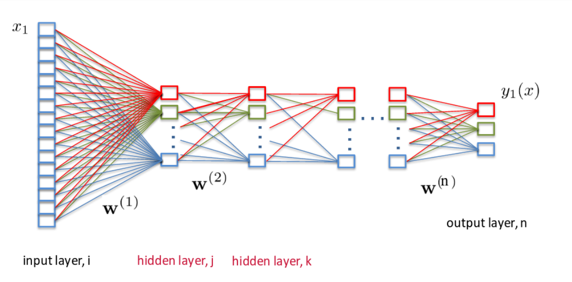
Deep Neural Networks



Deep Neural Networks

- Simply adding layers won't work.

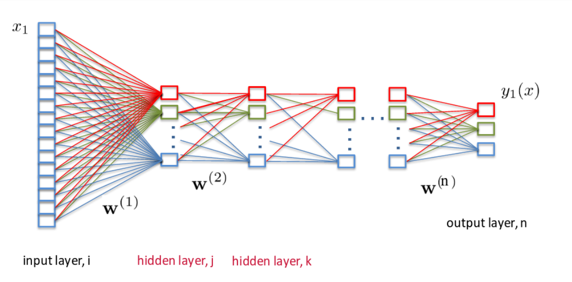
Deep Neural Networks



Deep Neural Networks

- Simply adding layers won't work.
- Too many parameters to train.

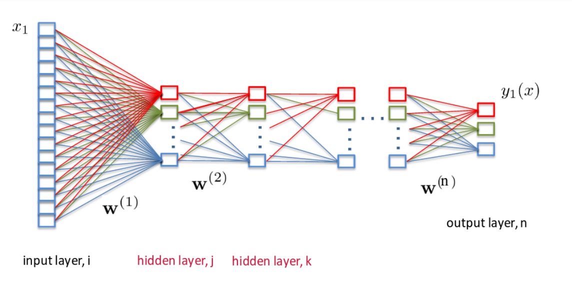
Deep Neural Networks



Deep Neural Networks

- Simply adding layers won't work.
- Too many parameters to train.
- Need smart architectures to capture additional priors.

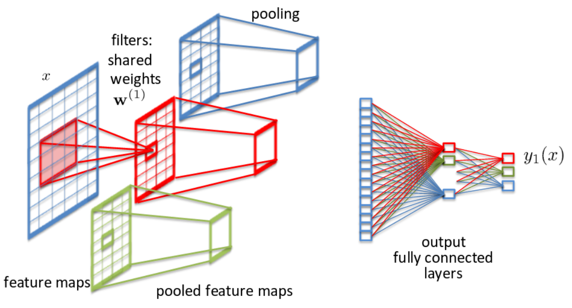
Deep Neural Networks



Deep Neural Networks

- Simply adding layers won't work.
- Too many parameters to train.
- Need smart architectures to capture additional priors.
- Two most commonly used architectures are CNNs and RNNs.

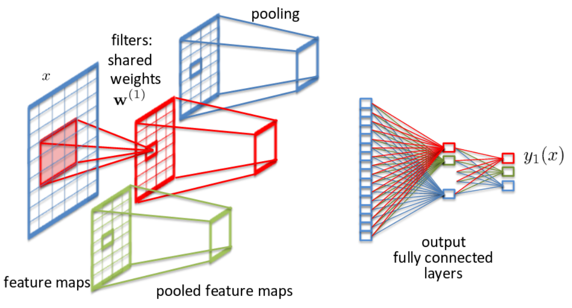
Convolutional Neural Networks



Convolutional Neural Networks

- Each layers learns a set of convolution kernels.

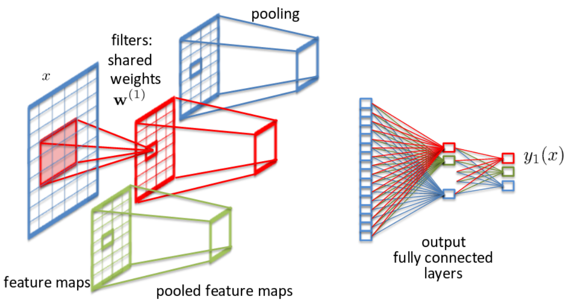
Convolutional Neural Networks



Convolutional Neural Networks

- Each layers learns a set of convolution kernels.
- Captures a very important prior –smoothness prior– known to computer vision community for a very long time.

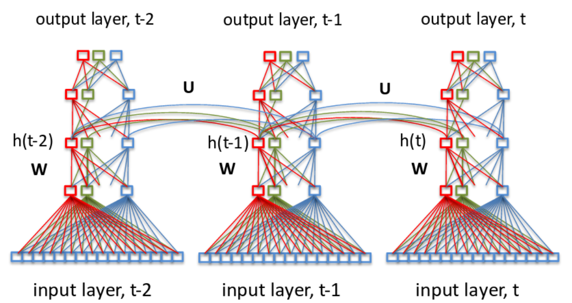
Convolutional Neural Networks



Convolutional Neural Networks

- Each layers learns a set of convolution kernels.
- Captures a very important prior –smoothness prior– known to computer vision community for a very long time.
- Much less number of parameters.

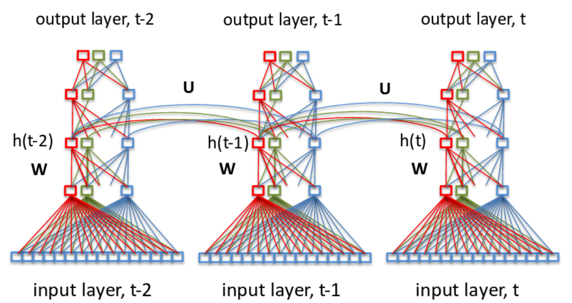
Recurrent Neural Networks



Recurrent Neural Networks

- Used for predicting sequential data

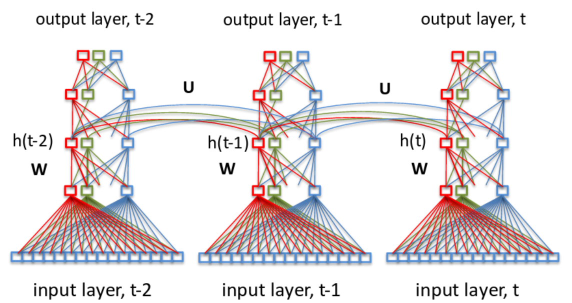
Recurrent Neural Networks



Recurrent Neural Networks

- Used for predicting sequential data
- Captures dependences across time frames.

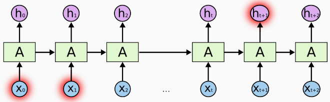
Recurrent Neural Networks



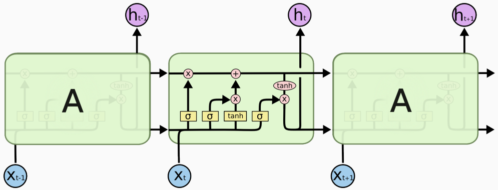
Recurrent Neural Networks

- Used for predicting sequential data
- Captures dependences across time frames.
- Usually harder to train (Vanishing Gradients).

LSTM

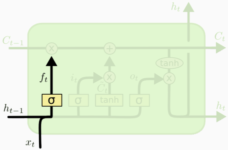


Recurrent Neural Networks



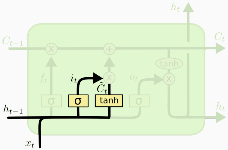
Long Short Term Memory

LSTM



$$f_t = \sigma \left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f \right)$$

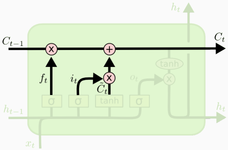
LSTM Forget gate



$$i_t = \sigma \left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i \right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

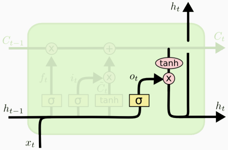
LSTM new content

LSTM



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM Add gate



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

LSTM Output Gate

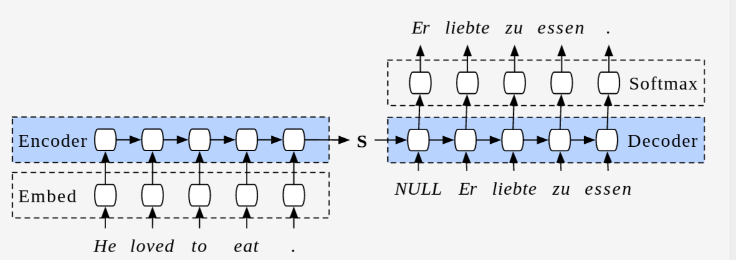
Neural Machine Translation

Neural Machine Translation

$$arg\, max_y\ p(x|y)$$

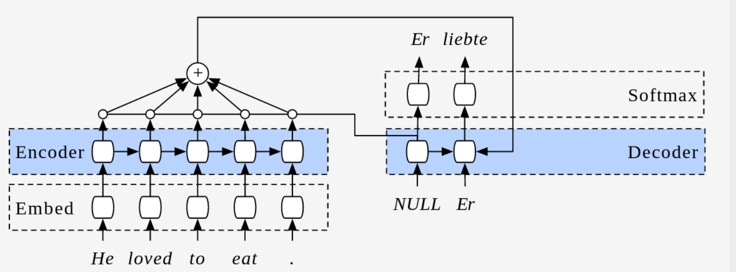
In Neural Machine Translation, a parameterized model (a neural network) is trained to maximize the conditional probability of the sentence pairs given parallel training corpus.

NMT - A historic perspective



Encoder-Decoder model for Machine Translation

- Fixed size encodings.
- Each language typically required an Encoder and Decoder.



Encoder-Decoder model with context

Jointly learning to Align and translate

For a input sentence, $X = (x_1, \cdots, x_{T_x})$. The NMT[1] system consists of,

- Encoder and Decoder are multi-layer recurrent neural networks (RNNs).
- Encoder RNN, at each input step t, generates hidden state, $h_t = f(x_t, h_{t-1})$.
- Context vector encodes the input sequence as, $c = q(\{h_1, \cdots, h_{T_x}\})$.

[1] **NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE. Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio, ICLR, 2015.**

Jointly learning to Align and translate

- The decoder is trained to predict the next work y_t given the context vector c and all previously predicted words $\{y_1, \cdots, y_{t-1}\}$

$$p(y) = \prod_{t=1}^{T} p(y_t | \{y_1, \cdots, y_{t-1}\}, c)$$

- With RNN, each conditional probability is modeled as,

$$p(y_t | \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c)$$

where s_t is the hidden state of the RNN.

Context Vector

The context vector for a input sentence i, is computed as a weighted sum of hidden states of the encoder (also known as **annotations**)

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})}$$

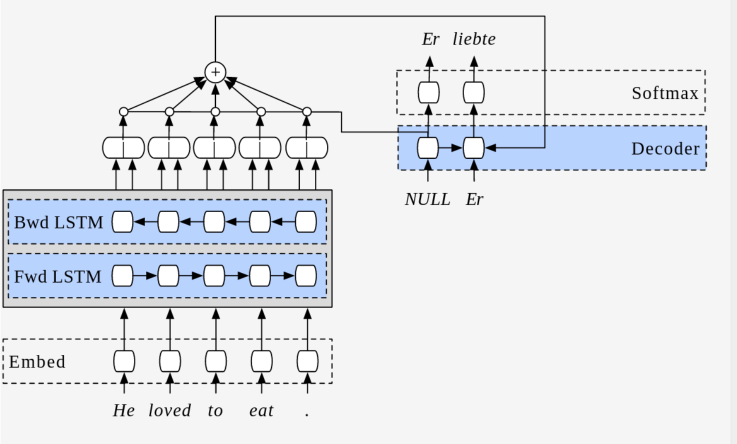
where,

$$e_{ij} = a(s_{i-1}, h_j)$$

is the alignment model that scores how well the inputs around the j and the output at the position i match.

A feedforward neural network is used as the alignment model and is **jointly trained** with all the NMT system as a whole.

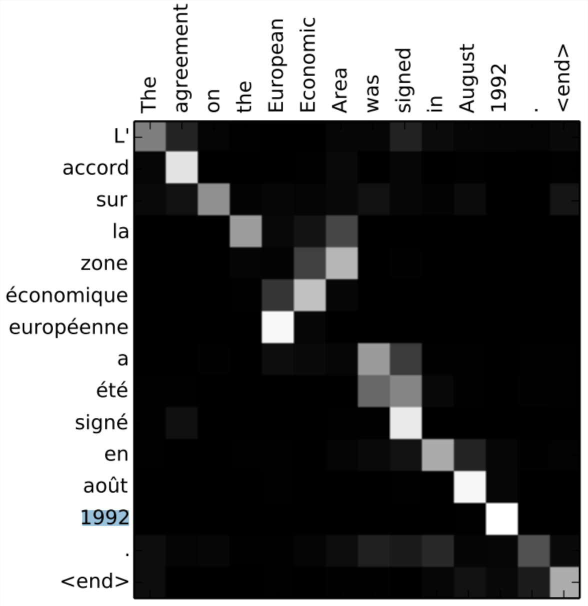
Bi-directional Encoder



Bi-directional Encoder

- Recurrent connection in both directions.
- Two independent states, updated independently.

Visualization of the context

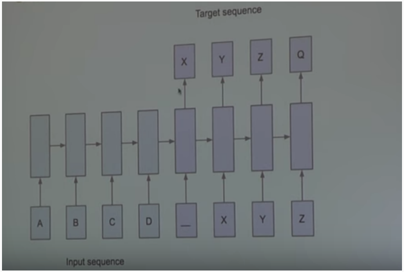


Seq2Seq Learning

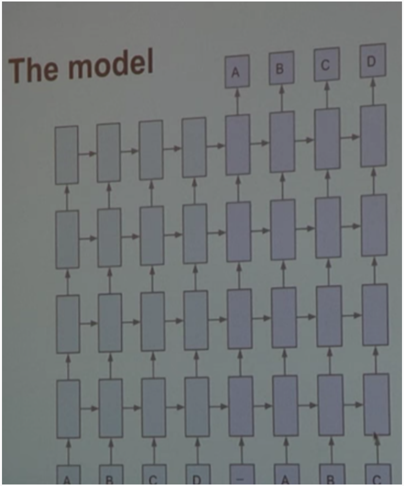
Machine Translation is can treated as a special case of a more generic sequence to sequence modeling.

1. Idea is simple: Throw more power at the network.
2. Deep LSTM layers.
3. No special handling for Machine translation.
4. Trained with SGD.

$$p(y_1, \cdots, y_T | x_1, \cdots, x_T) = \prod_{t=1}^{T} p(y_t | v, y_1, \cdots, y_{t-1})$$

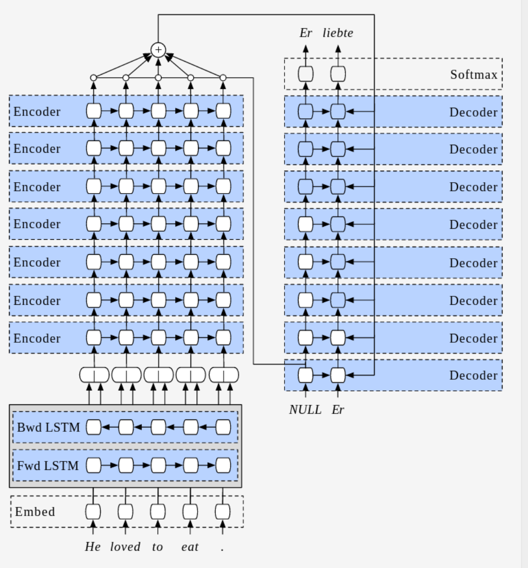


(a) Seq2Seq model



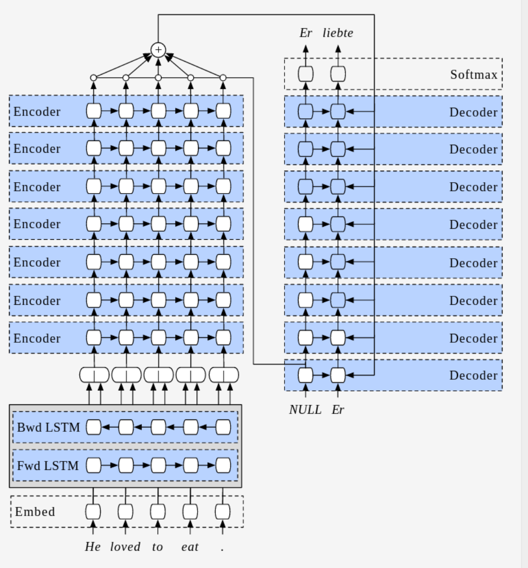
(b) More powerful model

Google NMT



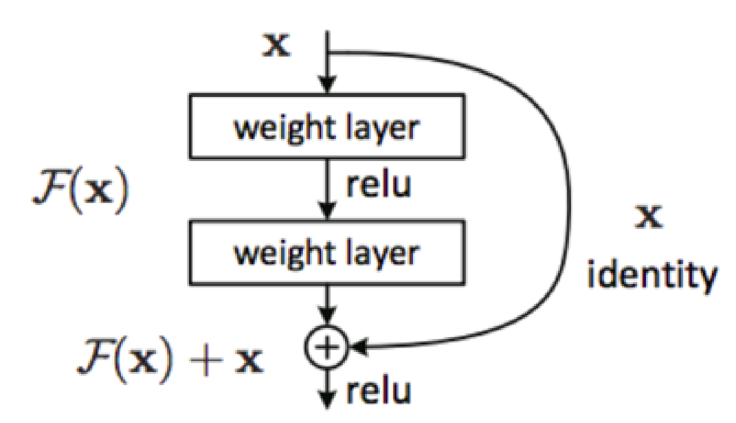
Simple Encoder-Decoder but more deeper as in Seq2Seq, and Context

Google NMT



Simple Encoder-Decoder but more deeper as in Seq2Seq and Context

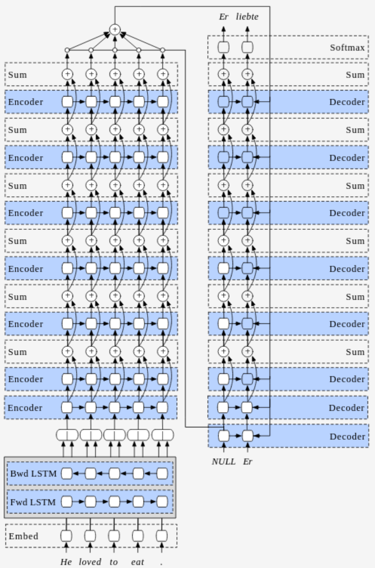
Residual learning



Residual connections enables training of very deep networks.[2]

[2] Deep Residual Learning for Image Recognition. Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. CVPR, 2016.

Google NMT with Residual connection



GNMT with residual connections.

Conclusion

Neural Machine Translation systems,

- Are State of the Art in Machine translation.
- Greatly benefited from the Neural Network research by other communities.
- In production by companies like Google, Microsoft, Facebook, etc.

Referencee

http://liris.cnrs.fr/natalia.neverova/nslides/presentation$_{s}oftshake_{1}51022_{n}ovide$