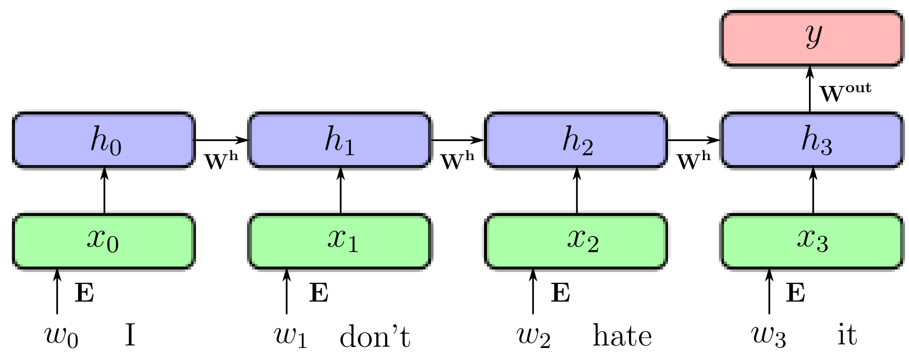
Deep Learning

Sequences, Attention and Transformers

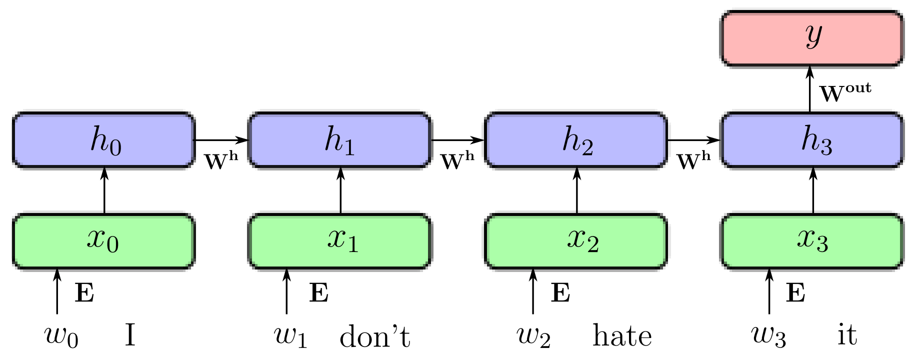
Alex Olson

Adapted from material by Charles Ollion & Olivier Grisel

Reminder: Recurrent Neural Networks



Reminder: Recurrent Neural Networks



takes a sequence as input

may output a single value, or a value for each time-step of the input

Outline

Encoder-decoder for machine translation

Outline

Encoder-decoder for machine translation

Attention mechanisms

Outline

Encoder-decoder for machine translation

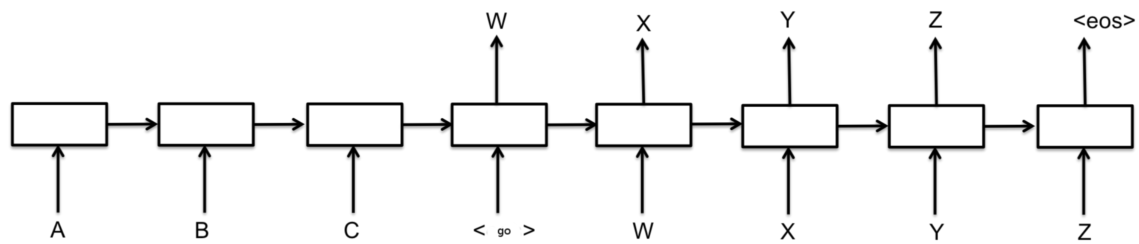
Attention mechanisms

Self-attention and Transformer

Encoder–Decoder for machine translation

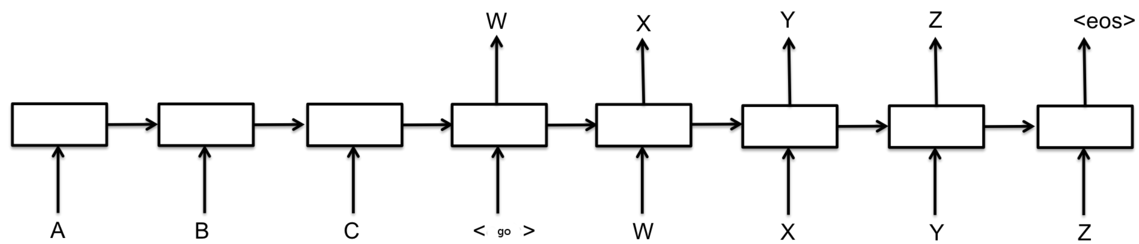
From sequence to sequence

Decoder-only Seq2Seq



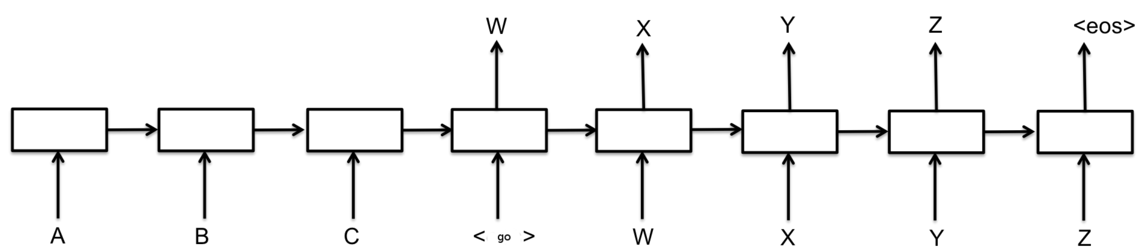
Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." NIPS 2014

Decoder-only Seq2Seq



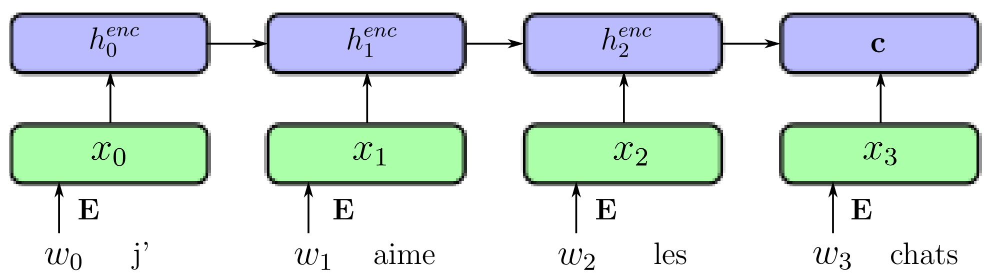
- Trained via **Teacher Forcing**:
  - Feed the ground-truth translation as input at each time-step
  - The model is trained to predict the next word in the target sequence

Decoder-only Seq2Seq



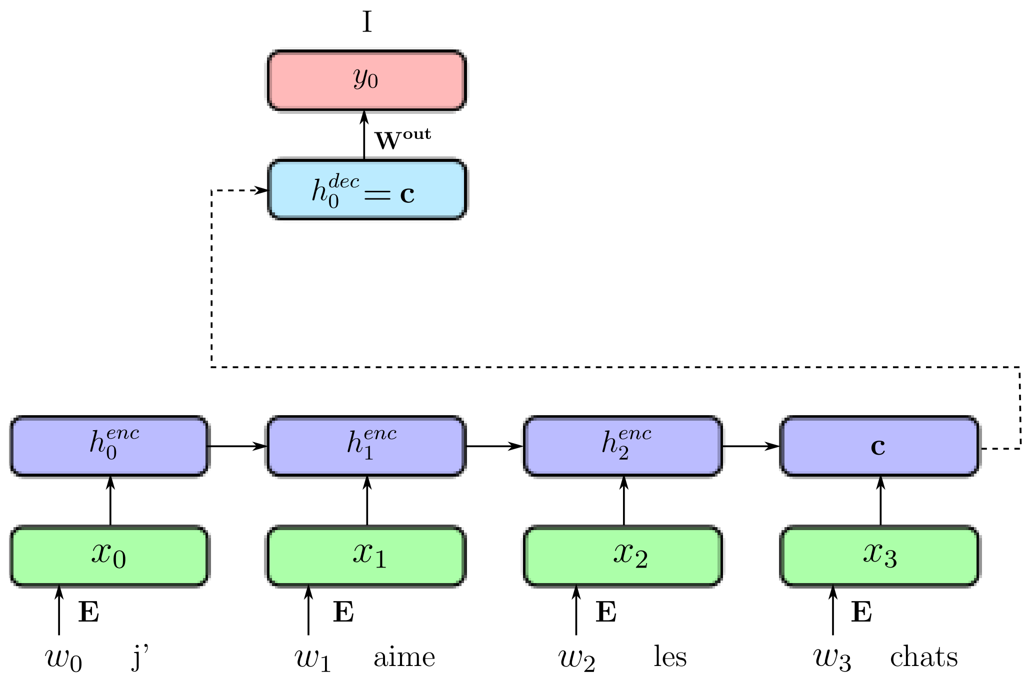
- Trained via **Teacher Forcing**:
    - Feed the ground-truth translation as input at each time-step
    - The model is trained to predict the next word in the target sequence
- Special symbols for starting decoding and end of sentence

Encoder-Decoder Seq2Seq



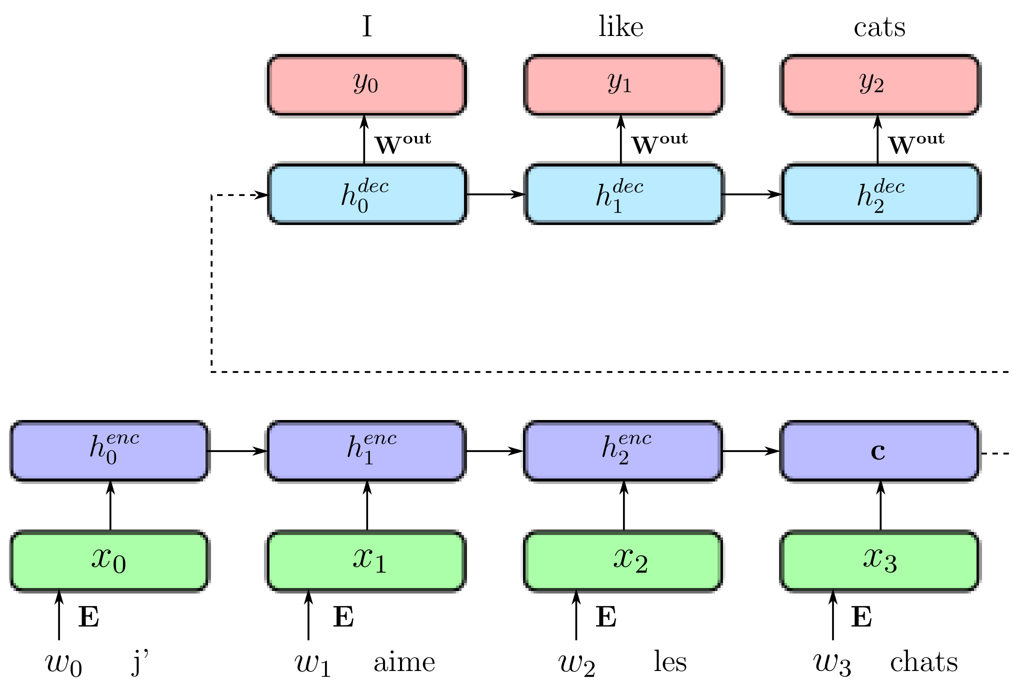
Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." 2014

Encoder-Decoder Seq2Seq



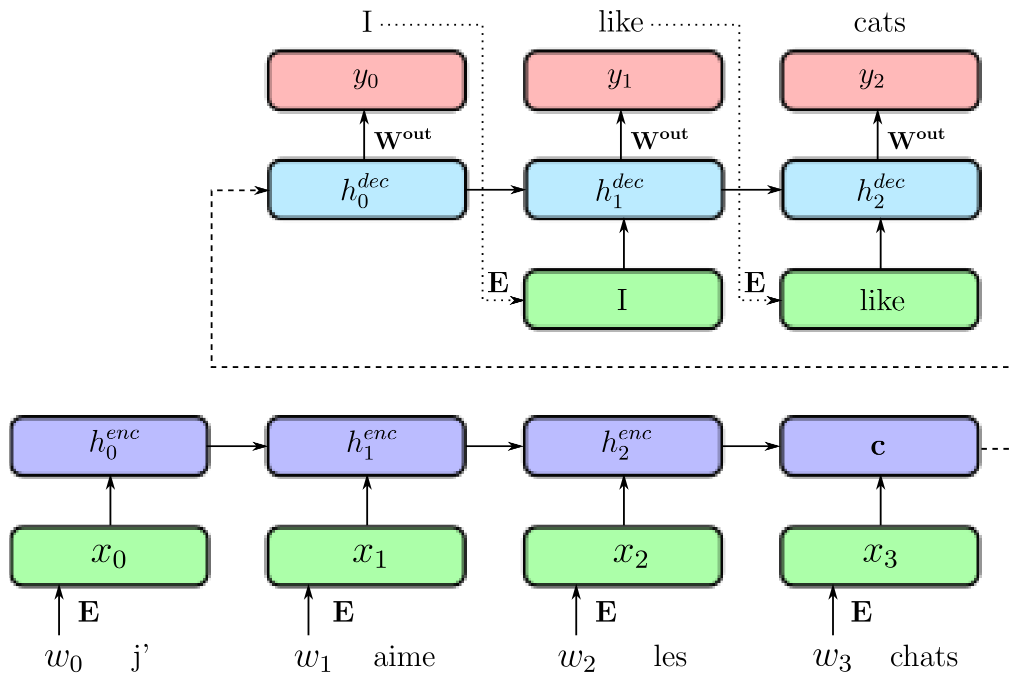
Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." 2014

Encoder-Decoder Seq2Seq



Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." 2014

Encoder-Decoder Seq2Seq



Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." 2014

Large Vocabulary Size

Softmax computation becomes **intractable** both at training and inference time (sum over $|V|$).

Large Vocabulary Size

Softmax computation becomes **intractable** both at training and inference time (sum over $|V|$).

Negative Sampling works well to learn word embeddings but is **not a good approximation** for **language modeling** and machine translation.

Large Vocabulary Size

Softmax computation becomes **intractable** both at training and inference time (sum over $|V|$).

Negative Sampling works well to learn word embeddings but is **not a good approximation** for **language modeling** and machine translation.

Approximate softmax with **sampled softmax** (a.k.a. bucketing):

- Accumulate train sequences in buckets $i \in B$ with $|V_i| = 50k$;
- Sample bucket i at random and train with regular softmax on V_i;
- Share softmax parameters for words in common across buckets;
- Iterate untill the end of the training set.

Alternative to Word Embeddings

Character-level Embedding (possibly with a CNN layer)

- (+) Much smaller vocabulary size (faster softmax)
- (+) No need for language specific segmentation (e.g. Chinese);
- (+) Robust to spelling mistakes and out-of-vocabulary words;
- (+) Can deal with mixed language contents.

Alternative to Word Embeddings

Character-level Embedding (possibly with a CNN layer)

- (+) Much smaller vocabulary size (faster softmax)
- (+) No need for language specific segmentation (e.g. Chinese);
- (+) Robust to spelling mistakes and out-of-vocabulary words;
- (+) Can deal with mixed language contents.

however

- (-) Need to learn word structure from data;
- (-) Decoding more complex and expensive.

Alternative to Word Embeddings

Character-level Embedding (possibly with a CNN layer)

- (+) Much smaller vocabulary size (faster softmax)
- (+) No need for language specific segmentation (e.g. Chinese);
- (+) Robust to spelling mistakes and out-of-vocabulary words;
- (+) Can deal with mixed language contents.

however

- (-) Need to learn word structure from data;
- (-) Decoding more complex and expensive.

Sub-word representations and **Byte Pair Encoding** (BPE) are better

Attention Mechanism

Attention Mechanism

Main problem with Encoder-Decoder:

- A sentence may have different parts with different concepts
- The **whole sentence** is represented as a **single vector**

I like cats but I don't like dogs

In depth explanation on https://blog.heuritech.com/2016/01/20/attention-mechanism/

Attention Mechanism

Main problem with Encoder-Decoder:

- A sentence may have different parts with different concepts
- The **whole sentence** is represented as a **single vector**

I like cats but I don't like dogs

Solution:

- Use all outputs of the encoder h_i to compute the outputs
- Build an **Attention Mechanism** to determine which output(s) to attend to

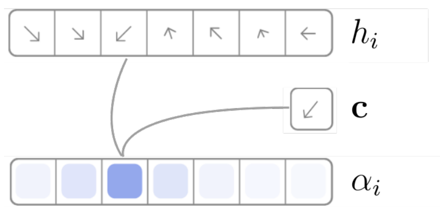
In depth explanation on https://blog.heuritech.com/2016/01/20/attention-mechanism/

Attention Mechanism

h_i vectors to attend to
$\mathbf{c}$ context

- Goal : select most relevant vector(s) given context $\mathbf{c}$

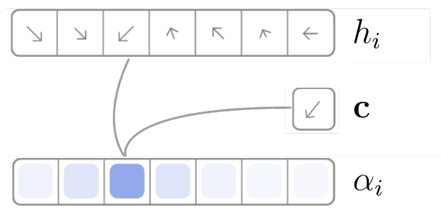
Attention Mechanism



$$e_i = f_{att}(\{h_i\}, \mathbf{c})$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$$

Attention Mechanism

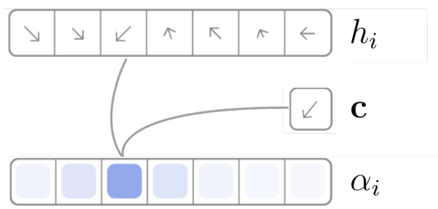


$$e_i = f_{att}(\{h_i\}, \mathbf{c})$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$$

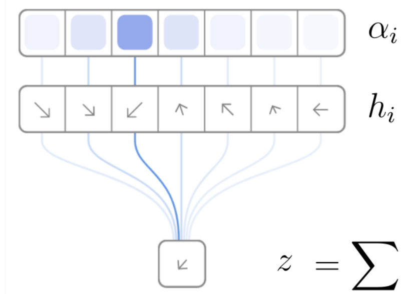
- f_{att} may be a cosine similarity, a deep network, etc.
- softmax enables to normalize and focus on very few items

Attention Mechanism



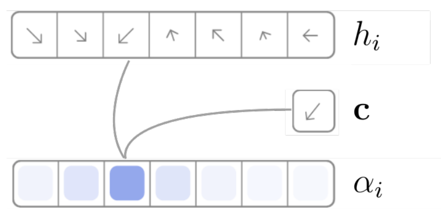
$$e_i = f_{att}(\{h_i\}, \mathbf{c})$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$$



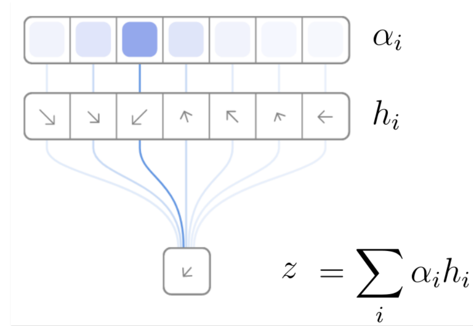
$$z = \sum_i \alpha_i h_i$$

Attention Mechanism



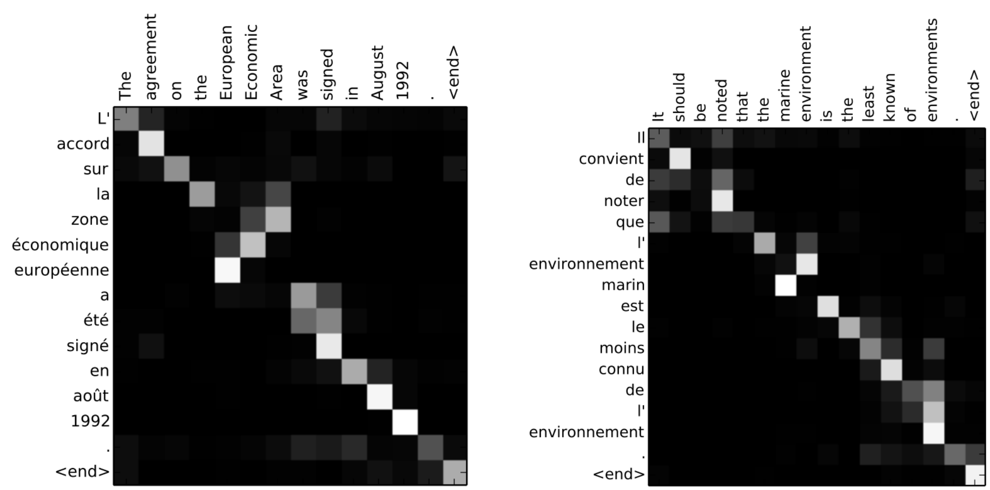
$$e_i = f_{att}(\{h_i\}, \mathbf{c})$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}$$



Compute a soft (differentiable) selection on a set of vectors

Visualizing Attention



Neural machine translation by jointly learning to align and translate, D Bahdanau, K Cho, Y Bengio 2014

The GNMT architecture (2016)

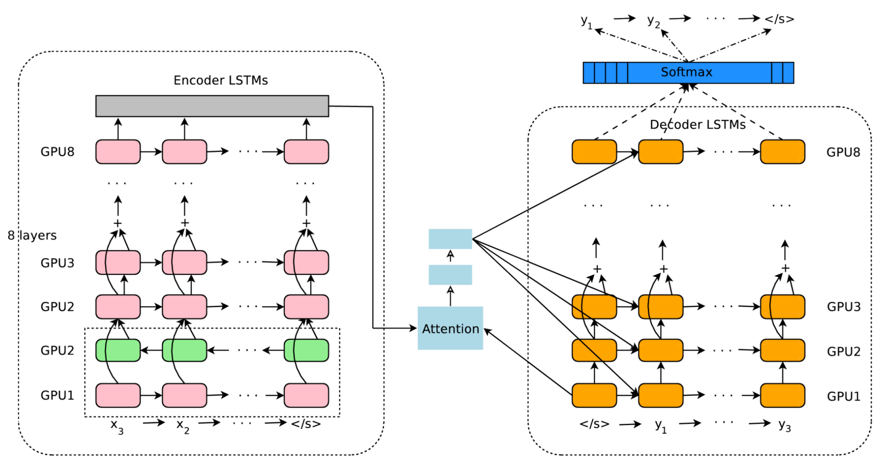
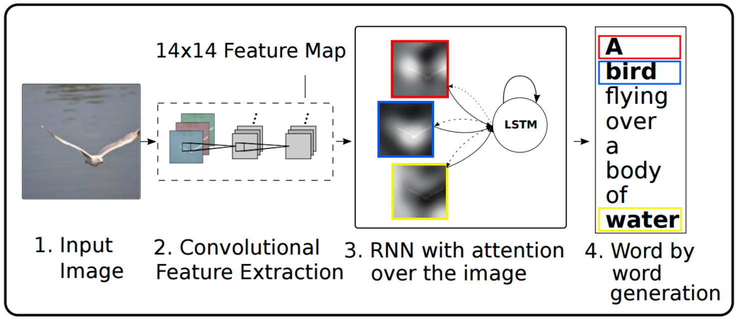


Figure 1: The model architecture of GNMT, Google's Neural Machine Translation system. On the left is the encoder network, on the right is the decoder network, in the middle is the attention module. The

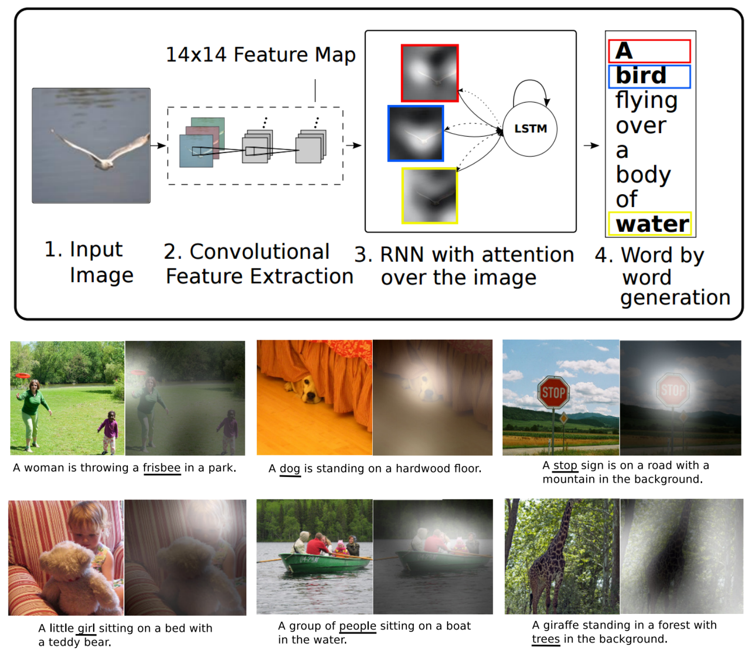
Yonghui Wu et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation"

Image Captioning



Xu, Kelvin, et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." ICML. 2015

Image Captioning

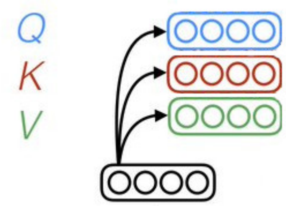


Xu, Kelvin, et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." ICML. 2015

Self-attention and transformer

Self-Attention

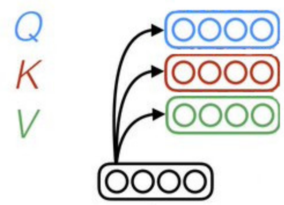
For each element of an input sequence X_i project into 3 vectors: query, key and value



Attention Is All You Need Ashish Vaswani et al. NIPS 2017

Self-Attention

For each element of an input sequence X_i project into 3 vectors: query, key and value

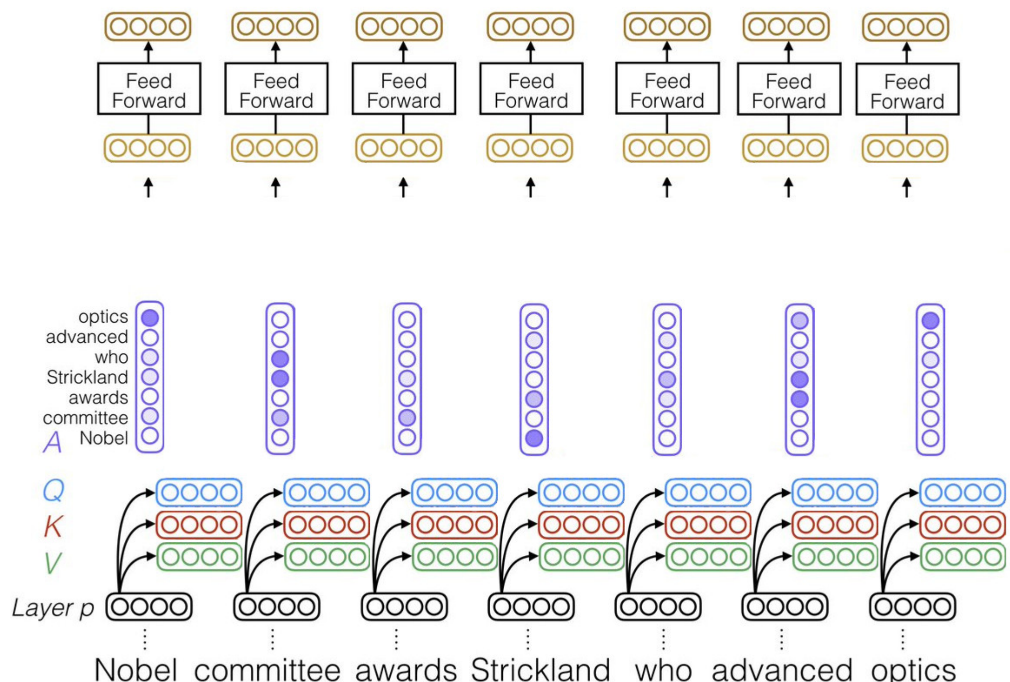


For each element, compute attention over all other vectors

$$SelfAttention(Q_i, \mathbf{K}, \mathbf{V}) = \sum_j softmax_j(\frac{Q_i \cdot \mathbf{K}^T}{\sqrt{d_k}})V_j$$

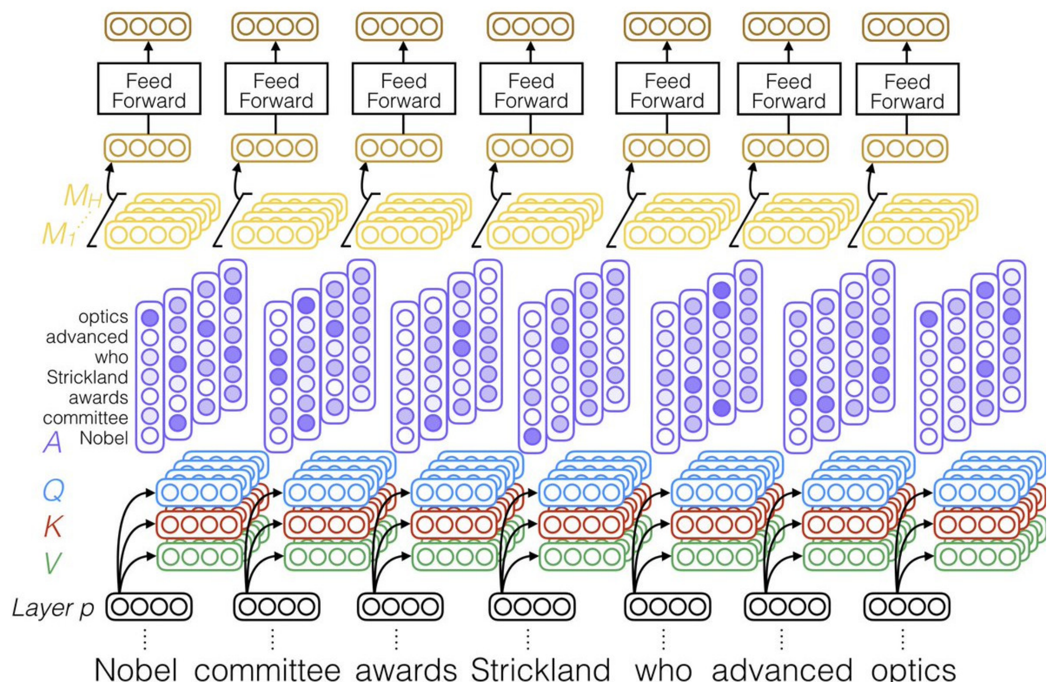
Attention Is All You Need Ashish Vaswani et al. NIPS 2017

Single-head self-attention



Adapted from Emma Strubell EMNLP 2018

Multi-head self-attention



Adapted from Emma Strubell EMNLP 2018

Transformer Architecture

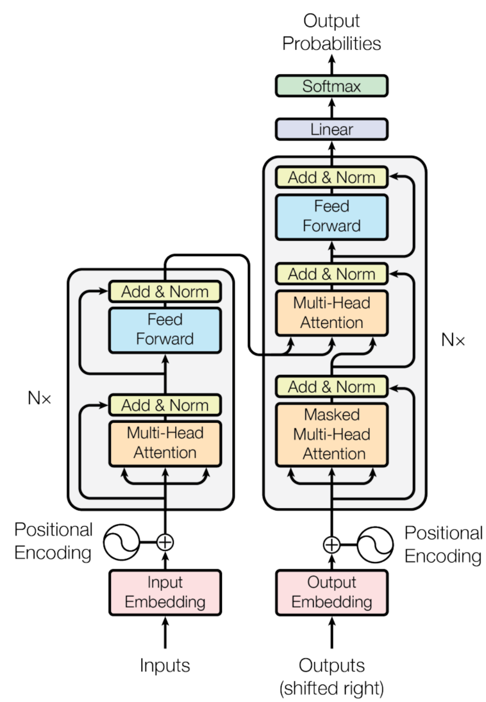
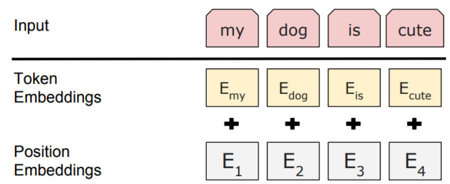


Figure 1: The Transformer - model architecture.

Attention Is All You Need Ashish Vaswani et al. NIPS 2017

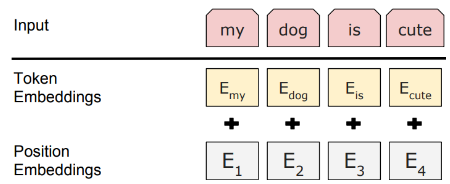
Transformer tricks

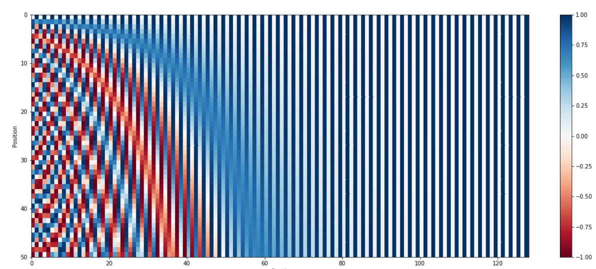
No notion of word order. Positional encoding need to be added:



Transformer tricks

No notion of word order. Positional encoding need to be added:





May also learn the embedding

Transformer based language models

Pretrained transformers for transfer learning, like "ImageNet-pretrained convnets" for NLP

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

GLUE : a multi-task benchmark and analysis platform for natural language processing, Alex Wang et al. ICLR 2019
BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin et al. 2018
GPT2 : https://openai.com/blog/better-language-models/

Transformer based language models

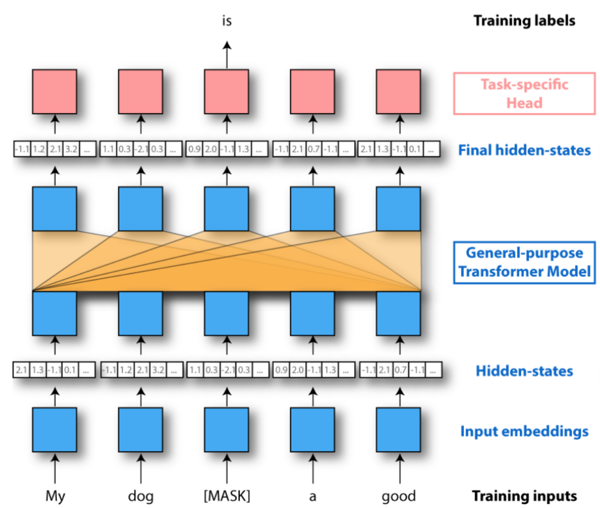
Pretrained transformers for transfer learning, like "ImageNet-pretrained convnets" for NLP

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Most recent models https://github.com/huggingface/transformers

GLUE : a multi-task benchmark and analysis platform for natural language processing, Alex Wang et al. ICLR 2019
BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin et al. 2018
GPT2 : https://openai.com/blog/better-language-models/

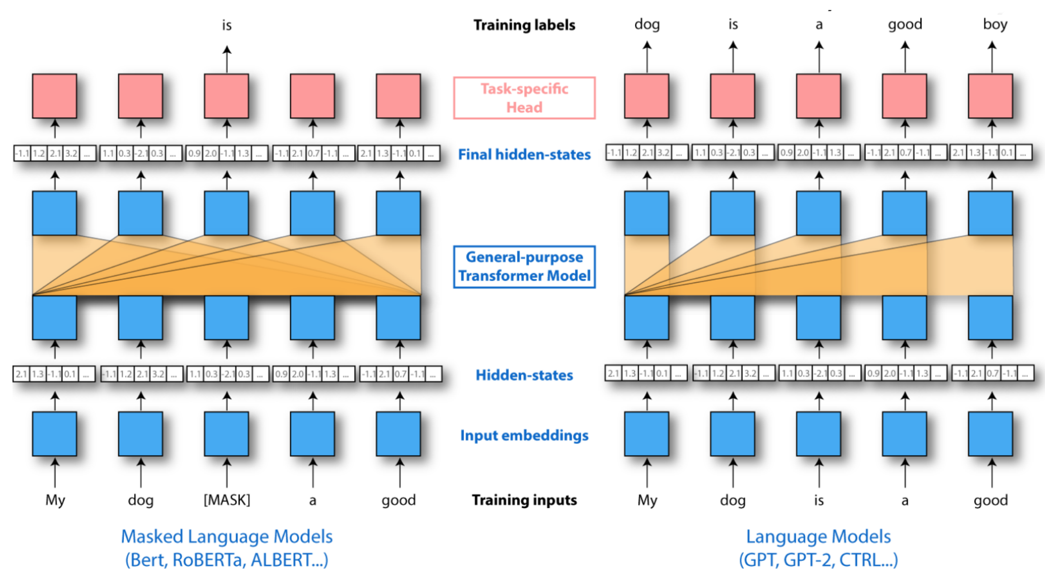
BERT



Masked Language Models
(Bert, RoBERTa, ALBERT...)

Slide from Thomas Wolf (HuggingFace)

BERT



Slide from Thomas Wolf (HuggingFace)

Takeaways

- Seq2Seq: variable number of input and ouput tokens
- Decoder-only vs Encoder-Decoder
- Decoder always trained by **Teacher Forcing**
- Can use either RNN or **Transformers** with *causal* attention
- Autogressive language modeling (GPT)

Next: Lab 7!