Pontificia Universidad JAVERIANA Colombia

Procesamiento de Lenguaje Natural

Tópicos Avanzados en Analítica

Maestría en Analítica para la Inteligencia de Negocios

Sergio Alberto Mora Pardo - H2 2023

Encoder-Decoder Attention

Pontificia Universidad JAVERIANA Colombia

Encoder-Decoder

SeqToSeq

Sutskever et al., 2014

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever Google ilyasu@google.com
Oriol Vinyals Google vinyals@google.com
Quoc V. Le Google qvl@google.com

Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT-14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous state of the art. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier.

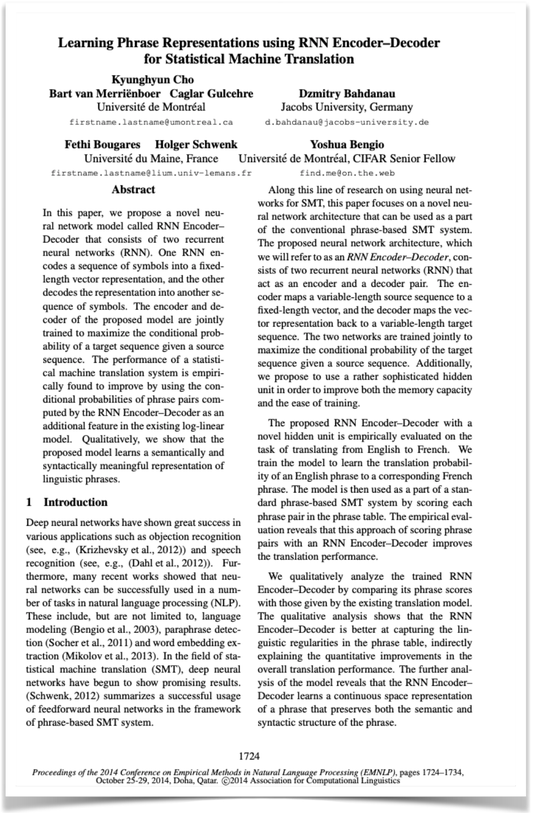
1 Introduction

Deep Neural Networks (DNNs) are extremely powerful machine learning models that achieve excellent performance on difficult problems such as speech recognition [13, 7] and visual object recognition [19, 6, 21, 20]. DNNs are powerful because they can perform arbitrary parallel computation for a modest number of steps. A surprising example of the power of DNNs is their ability to sort N N-bit numbers using only 2 hidden layers of quadratic size [27]. So, while neural networks are related to conventional statistical models, they learn an intricate computation. Furthermore, large DNNs can be trained with supervised backpropagation whenever the labeled training set has enough information to specify the network's parameters. Thus, if there exists a parameter setting of a large DNN that achieves good results (for example, because humans can solve the task very rapidly), supervised backpropagation will find these parameters and solve the problem.

Despite their flexibility and power, DNNs can only be applied to problems whose inputs and targets can be sensibly encoded with vectors of fixed dimensionality. It is a significant limitation, since many important problems are best expressed with sequences whose lengths are not known a-priori. For example, speech recognition and machine translation are sequential problems. Likewise, question answering can also be seen as mapping a sequence of words representing the question to a

1

Cho et al., 2014



Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

Kyunghyun Cho
Bart van Merriënboer **Caglar Gulcehre** Université de Montréal firstname.lastname@umontreal.ca
Dzmitry Bahdanau Jacobs University, Germany d.bahdanau@jacobs-university.de
Fethi Bougares **Holger Schwenk** Université du Maine, France firstname.lastname@lium.univ-lemans.fr
Yoshua Bengio Université de Montréal, CIFAR Senior Fellow find.me@on.the.web

Abstract

In this paper, we propose a novel neural network model called RNN Encoder–Decoder that consists of two recurrent neural networks (RNN). One RNN encodes a sequence of symbols into a fixed-length vector representation, and the other decodes the representation into another sequence of symbols. The encoder and decoder of the proposed model are jointly trained to maximize the conditional probability of a target sequence given a source sequence. The performance of a statistical machine translation system is empirically found to improve by using the conditional probabilities of phrase pairs computed by the RNN Encoder–Decoder as an additional feature in the existing log-linear model. Qualitatively, we show that the proposed model learns a semantically and syntactically meaningful representation of linguistic phrases.

1 Introduction

Deep neural networks have shown great success in various applications such as objection recognition (see, e.g., (Krizhevsky et al., 2012)) and speech recognition (see, e.g., (Dahl et al., 2012)). Furthermore, many recent works showed that neural networks can be successfully used in a number of tasks in natural language processing (NLP). These include, but are not limited to, language modeling (Bengio et al., 2003), paraphrase detection (Socher et al., 2011) and word embedding extraction (Mikolov et al., 2013). In the field of statistical machine translation (SMT), deep neural networks have begun to show promising results. (Schwenk, 2012) summarizes a successful usage of feedforward neural networks in the framework of phrase-based SMT system.

Along this line of research on using neural networks for SMT, this paper focuses on a novel neural network architecture that can be used as a part of the conventional phrase-based SMT system. The proposed neural network architecture, which we will refer to as an *RNN Encoder–Decoder*, consists of two recurrent neural networks (RNN) that act as an encoder and a decoder pair. The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence. The two networks are trained jointly to maximize the conditional probability of the target sequence given a source sequence. Additionally, we propose to use a rather sophisticated hidden unit in order to improve both the memory capacity and the ease of training.

The proposed RNN Encoder–Decoder with a novel hidden unit is empirically evaluated on the task of translating from English to French. We train the model to learn the translation probability of an English phrase to a corresponding French phrase. The model is then used as a part of a standard phrase-based SMT system by scoring each phrase pair in the phrase table. The empirical evaluation reveals that this approach of scoring phrase pairs with an RNN Encoder–Decoder improves the translation performance.

We qualitatively analyze the trained RNN Encoder–Decoder by comparing its phrase scores with those given by the existing translation model. The qualitative analysis shows that the RNN Encoder–Decoder is better at capturing the linguistic regularities in the phrase table, indirectly explaining the quantitative improvements in the overall translation performance. The further analysis of the model reveals that the RNN Encoder–Decoder learns a continuous space representation of a phrase that preserves both the semantic and syntactic structure of the phrase.

1724

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, October 25-29, 2014, Doha, Qatar.

Encoder-Decoder

Neural Machine Translation

Found in translation: More accurate, fluent sentences in Google Translate

Cho et al., 2014

Sutskever et al., 2014

Encoder-Decoder

SeqToSeq

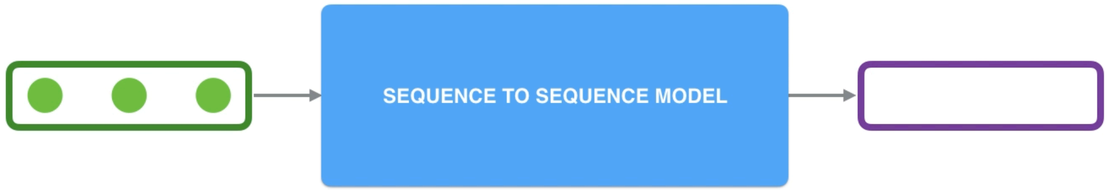
SeqtoSeq (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



Encoder-Decoder

SeqToSeq

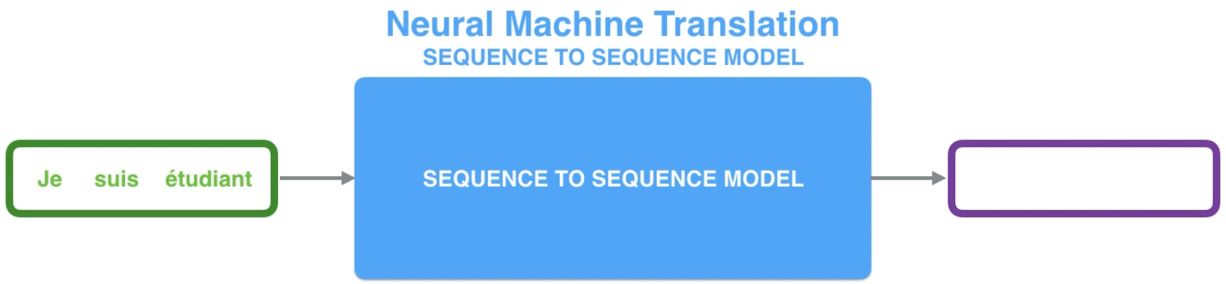
SeqtoSeq (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



Encoder-Decoder

SeqToSeq

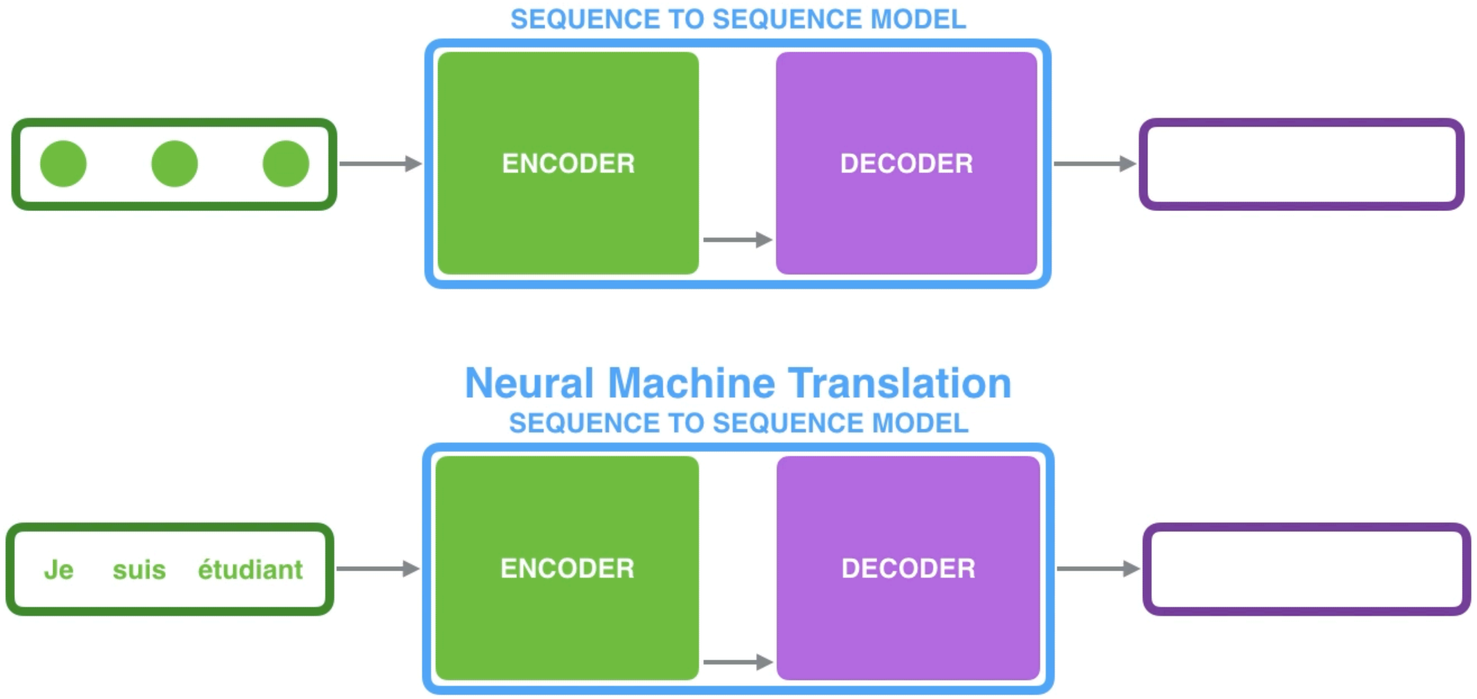
SeqtoSeq (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



Encoder-Decoder

SeqToSeq

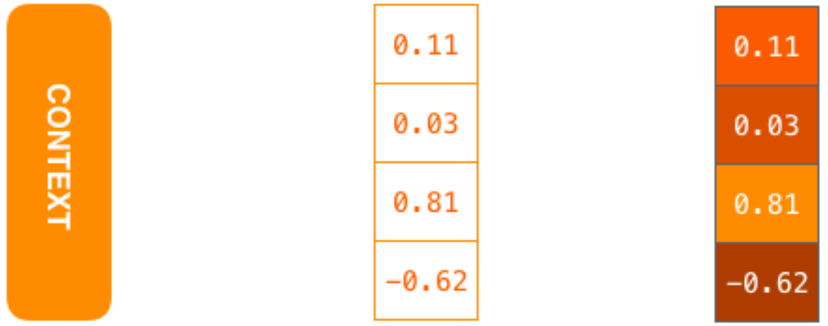
SeqtoSeq (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



Contexto es un vector → *Dimensión Latente?*

Tamaño regular: 256, 512 o 1024

Encoder-Decoder

SeqToSeq

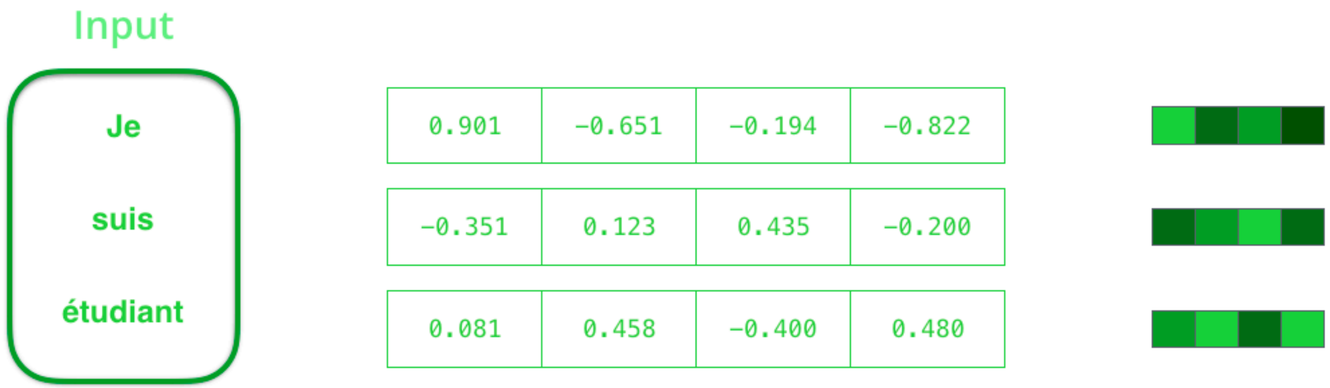
SeqtoSeq (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



Embedding → *Tamaño regular: 200 o 300 dimensiones*

Encoder-Decoder

SeqToSeq

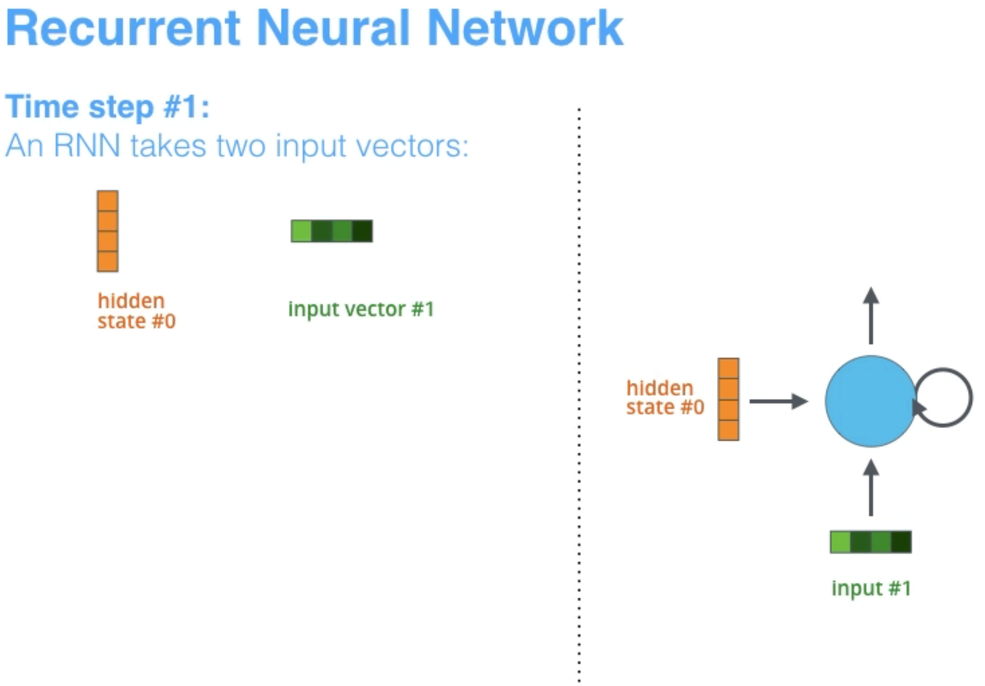
SeqtoSeq (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



Encoder-Decoder

SeqToSeq

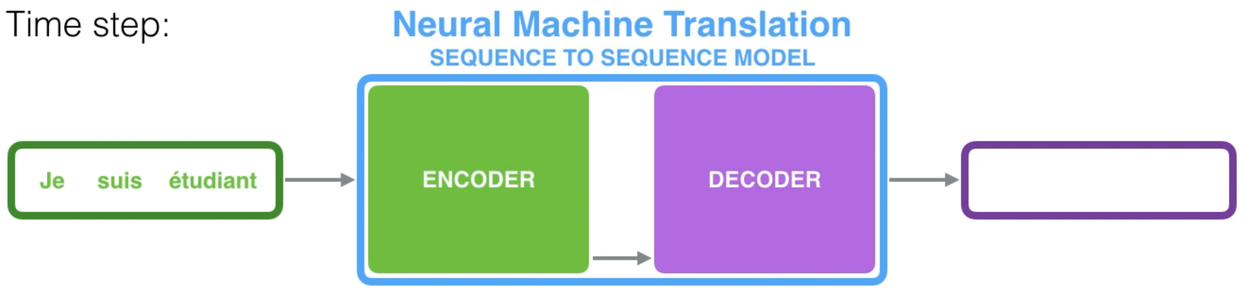
SeqtoSeq (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



Encoder-Decoder

SeqToSeq

SeqtoSeq (Neural Machine Translation)

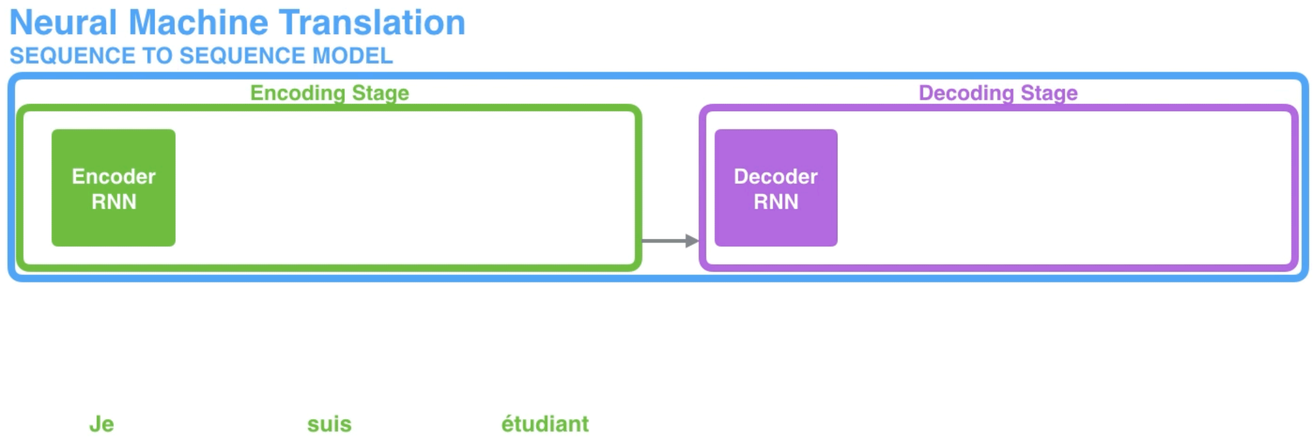
Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen

Desenrollando la RNN



Contexto se vuelve un “cuello de botella”

Encoder-Decoder

SeqToSeq with Attention

SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

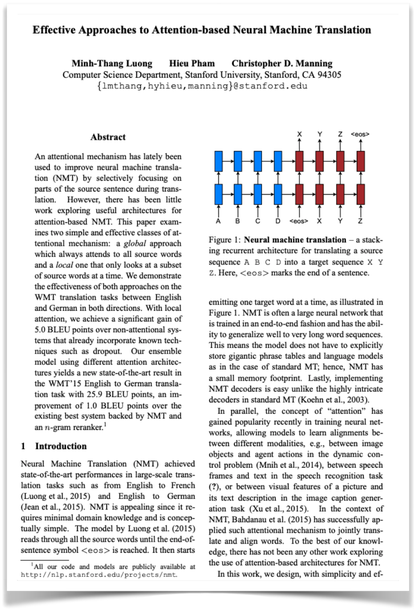
- *palabras*

- *letras*

- *Características de una imagen*

Stanford University

Bahdanau et al., 2014



Effective Approaches to Attention-based Neural Machine Translation

Minh-Thang Luong Hieu Pham Christopher D. Manning
Computer Science Department, Stanford University, Stanford, CA 94305
{lmthang,hyhieu,manning}@stanford.edu

Abstract

An attentional mechanism has lately been used to improve neural machine translation (NMT) by selectively focusing on parts of the source sentence during translation. However, there has been little work exploring useful architectures for attention-based NMT. This paper examines two simple and effective classes of attentional mechanism: a *global* approach which always attends to all source words and a *local* one that only looks at a subset of source words at a time. We demonstrate the effectiveness of both approaches on the WMT translation tasks between English and German in both directions. With local attention, we achieve a significant gain of 5.0 BLEU points over non-attentional systems that already incorporate known techniques such as dropout. Our ensemble model using different attention architectures yields a new state-of-the-art result in the WMT'15 English to German translation task with 25.9 BLEU points, an improvement of 1.0 BLEU points over the existing best system backed by NMT and an n-gram reranker.[1]

1 Introduction

Neural Machine Translation (NMT) achieved state-of-the-art performances in large-scale translation tasks such as from English to French (Luong et al., 2015) and English to German (Jean et al., 2015). NMT is appealing since it requires minimal domain knowledge and is conceptually simple. The model by Luong et al. (2015) reads through all the source words until the end-of-sentence symbol <eos> is reached. It then starts

[1]All our code and models are publicly available at http://nlp.stanford.edu/projects/nmt.

Figure 1: **Neural machine translation** – a stacking recurrent architecture for translating a source sequence A B C D into a target sequence X Y Z. Here, <eos> marks the end of a sentence.

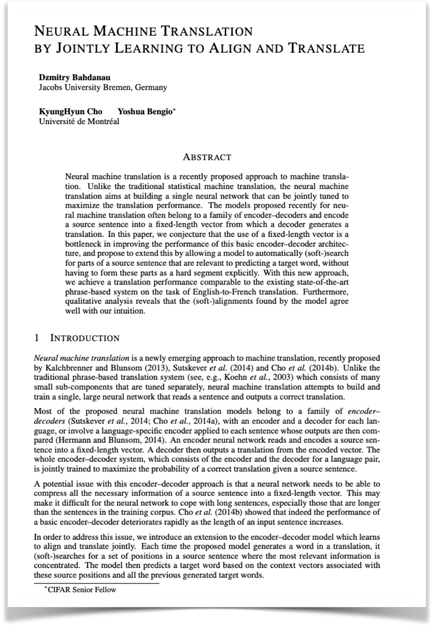
emitting one target word at a time, as illustrated in Figure 1. NMT is often a large neural network that is trained in an end-to-end fashion and has the ability to generalize well to very long word sequences. This means the model does not have to explicitly store gigantic phrase tables and language models as in the case of standard MT; hence, NMT has a small memory footprint. Lastly, implementing NMT decoders is easy unlike the highly intricate decoders in standard MT (Koehn et al., 2003).

In parallel, the concept of "attention" has gained popularity recently in training neural networks, allowing models to learn alignments between different modalities, e.g., between image objects and agent actions in the dynamic control problem (Mnih et al., 2014), between speech frames and text in the speech recognition task (?), or between visual features of a picture and its text description in the image caption generation task (Xu et al., 2015). In the context of NMT, Bahdanau et al. (2015) has successfully applied such attentional mechanism to jointly translate and align words. To the best of our knowledge, there has not been any other work exploring the use of attention-based architectures for NMT.

In this work, we design, with simplicity and ef-

CIFAR

Luong et al., 2015
*Bengio**



NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho Yoshua Bengio*
Université de Montréal

ABSTRACT

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

1 INTRODUCTION

Neural machine translation is a newly emerging approach to machine translation, recently proposed by Kalchbrenner and Blunsom (2013), Sutskever *et al.* (2014) and Cho *et al.* (2014b). Unlike the traditional phrase-based translation system (see, e.g., Koehn *et al.*, 2003) which consists of many small sub-components that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation.

Most of the proposed neural machine translation models belong to a family of *encoder–decoders* (Sutskever *et al.*, 2014; Cho *et al.*, 2014a), with an encoder and a decoder for each language, or involve a language-specific encoder applied to each sentence whose outputs are then compared (Hermann and Blunsom, 2014). An encoder neural network reads and encodes a source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole encoder–decoder system, which consists of the encoder and the decoder for a language pair, is jointly trained to maximize the probability of a correct translation given a source sentence.

A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus. Cho *et al.* (2014b) showed that indeed the performance of a basic encoder–decoder deteriorates rapidly as the length of an input sentence increases.

In order to address this issue, we introduce an extension to the encoder–decoder model which learns to align and translate jointly. Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words.

*CIFAR Senior Fellow

Encoder-Decoder

SeqToSeq with Attention

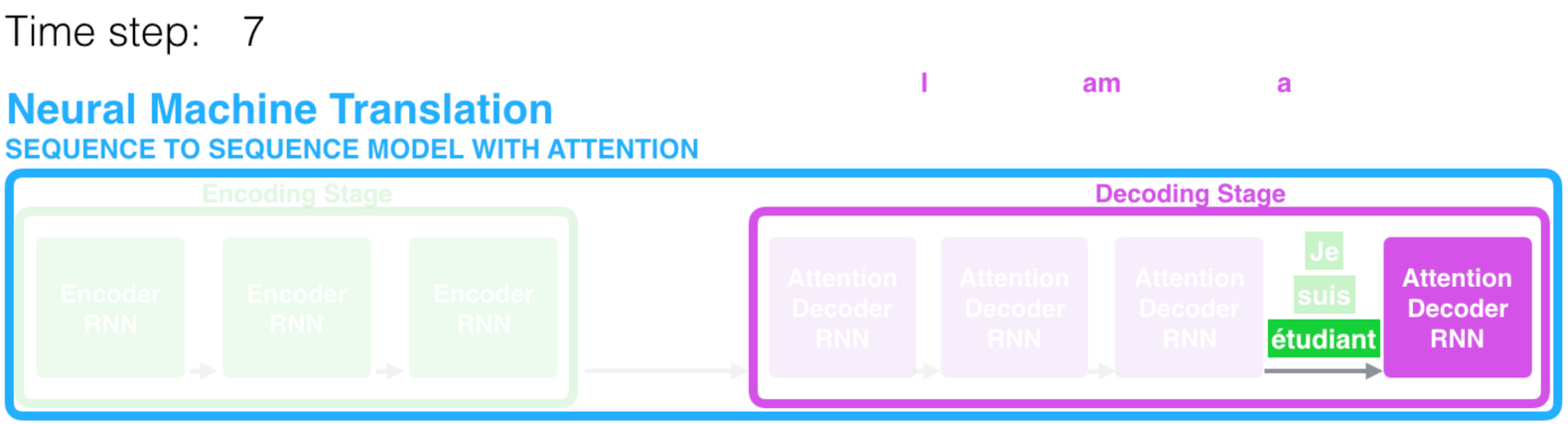
SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

- palabras

- letras

- Características de una imagen



*'Atención' permite al decodificador centrarse en '**étudiant**'*

Amplificar la señal de la parte relevante de la secuencia de entrada hace que los modelos con atención produzcan mejores resultados que los modelos sin atención.

Encoder-Decoder

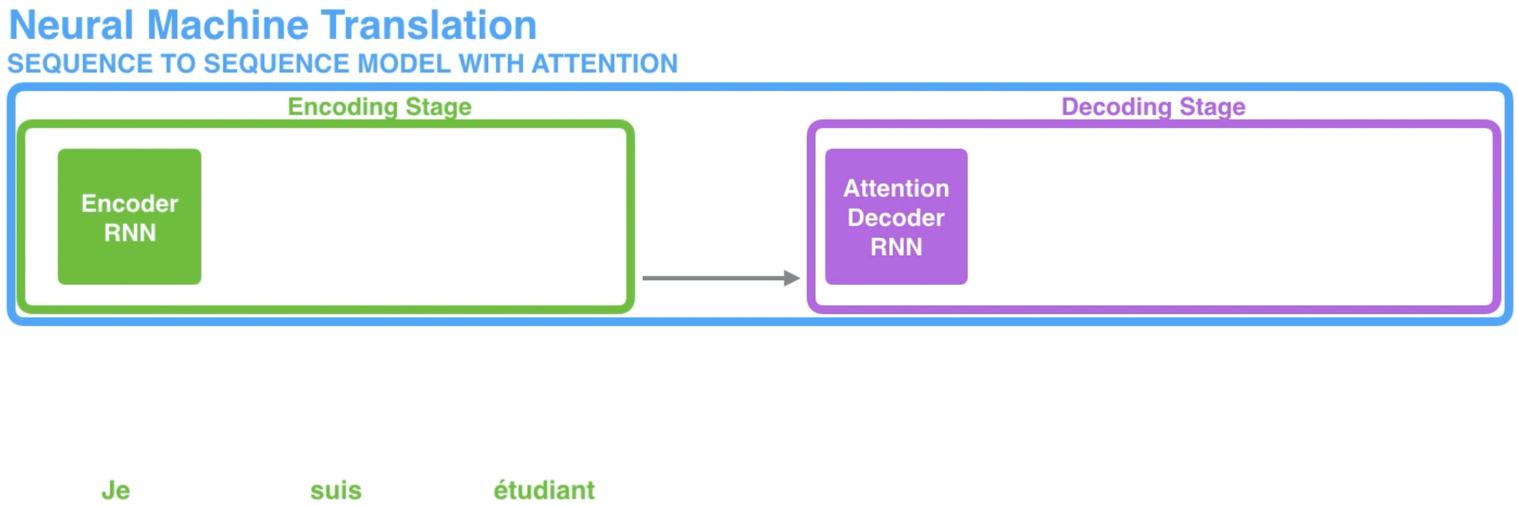
SeqToSeq with Attention

SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

SeqToSeq *Vs.* *SeqToSeq + Attention*



Encoder-Decoder

SeqToSeq with Attention

SeqtoSeq Attention (Neural Machine Translation)

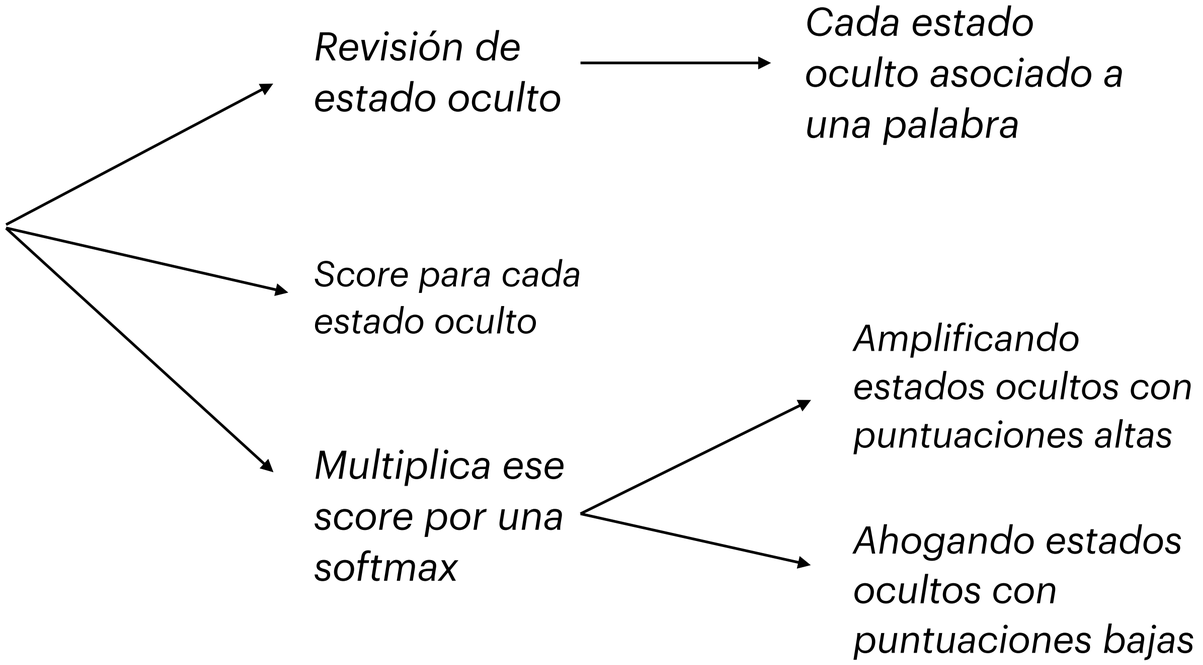
Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

SeqToSeq *Vs.* *SeqToSeq + Attention*

1. Pasamos todos los estados ocultos al decodificador

2. Centrarse en las partes de la entrada que son relevantes.



Encoder-Decoder

SeqToSeq with Attention

SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

SeqToSeq *Vs.* *SeqToSeq + Attention*

Encoder-Decoder

SeqToSeq with Attention

SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

SeqToSeq *Vs.* *SeqToSeq + Attention*

Proceso completo de atención

1. El decodificador de atención RNN incorpora el token <END> y un estado oculto del decodificador inicial .
2. El RNN procesa sus entradas, produciendo una salida y un nuevo vector de estado oculto (h 4). La salida se descarta.
3. Paso de atención: utilizamos los estados ocultos del codificador y el vector h 4 para calcular un vector de contexto (C 4) para este paso de tiempo.
4. Concatenamos h 4 y C 4 en un vector.
5. Pasamos este vector a través de una red neuronal feedforward (una entrenada conjuntamente con el modelo).
6. La salida de las redes neuronales de avance indica la palabra de salida de este paso de tiempo.
7. Repita los pasos la próxima vez.

Encoder-Decoder

SeqToSeq with Attention

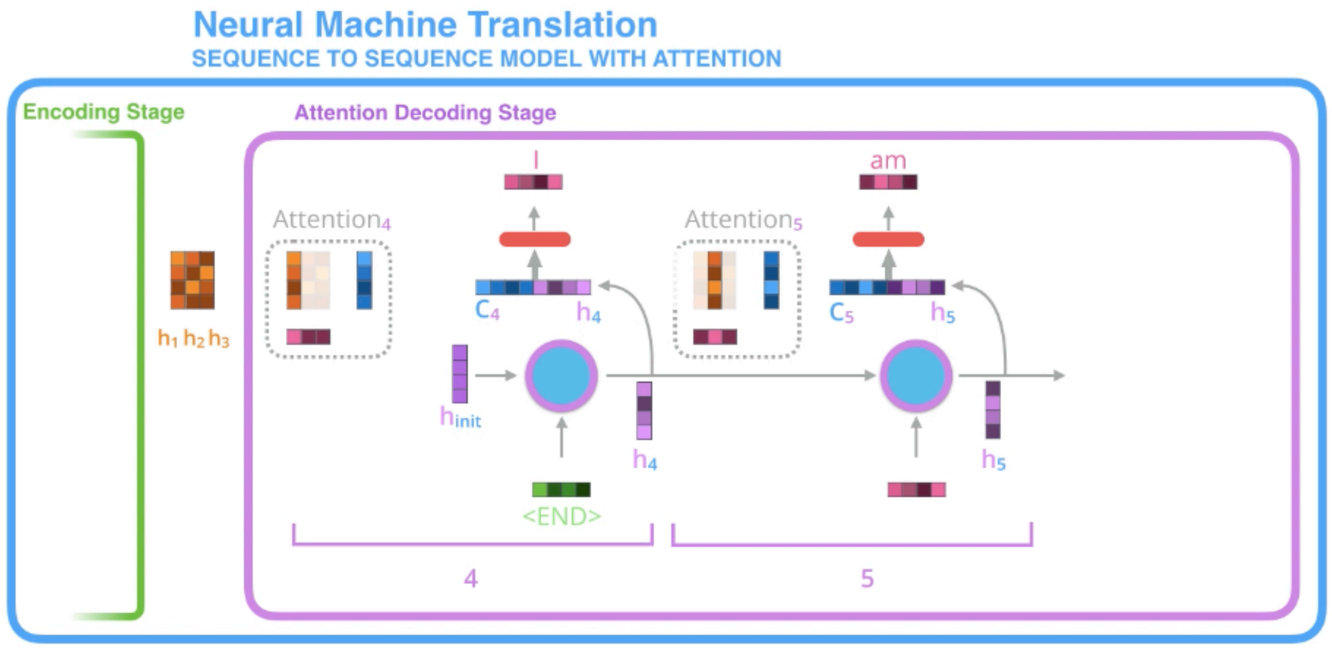
SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

SeqToSeq *Vs.* *SeqToSeq + Attention*

Proceso completo de atención



Encoder-Decoder

SeqToSeq with Attention

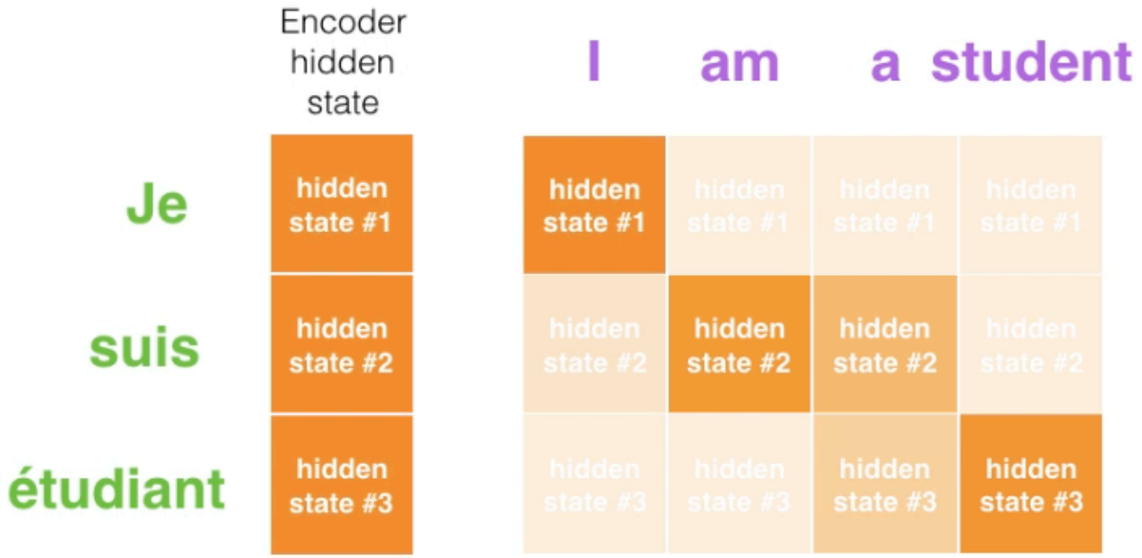
SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

SeqToSeq *Vs.* *SeqToSeq + Attention*

Proceso completo de atención



Encoder-Decoder

SeqToSeq with Attention

SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

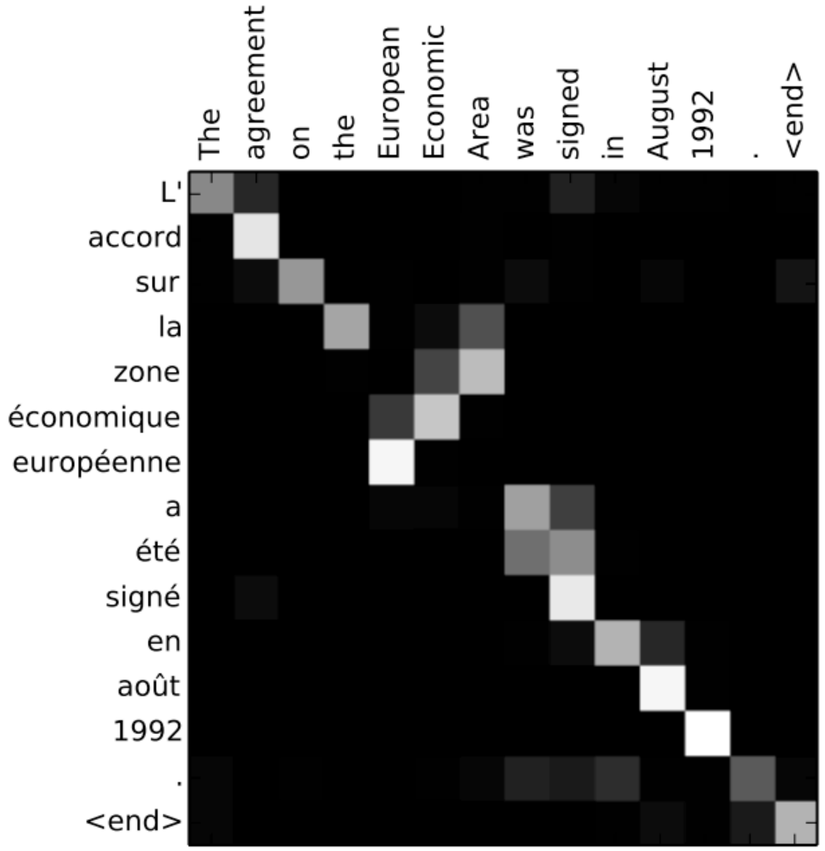
SeqToSeq *Vs.* *SeqToSeq + Attention*

Proceso completo de atención

Traduce indistinto del orden de las palabras.

↓

'Espacio Económico Europeo'



Encoder-Decoder

SeqToSeq with Attention

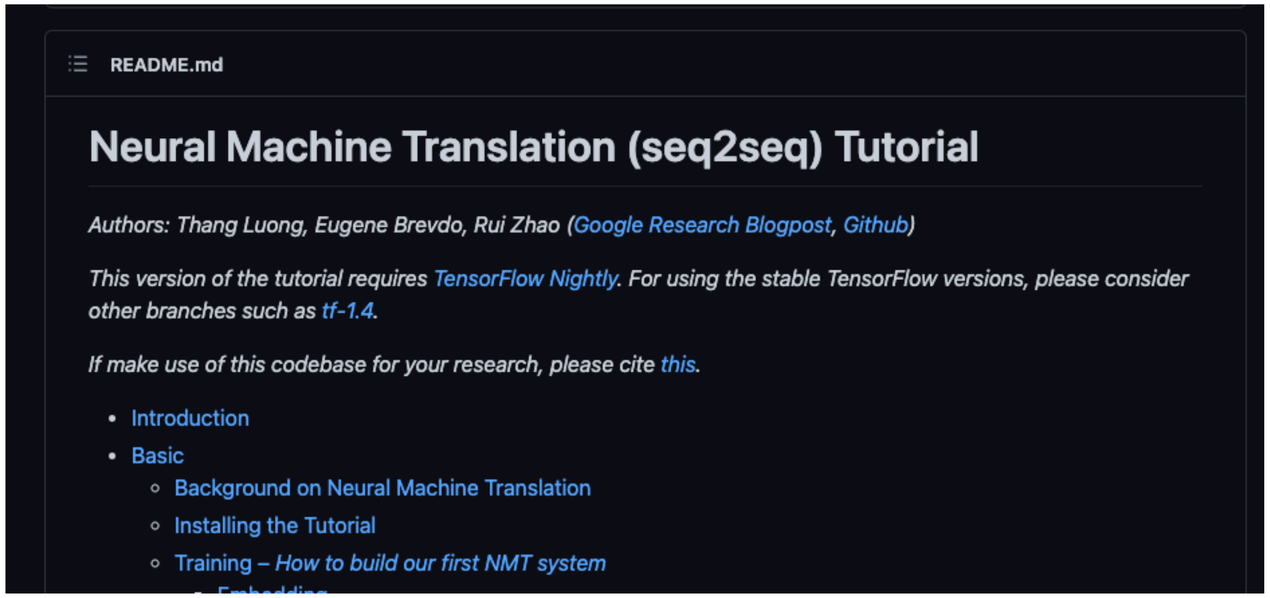
SeqtoSeq Attention (Neural Machine Translation)

Toma una secuencia de elementos y genera otra secuencia de elementos

Diferencias:

SeqToSeq *Vs.* *SeqToSeq + Attention*

TUTORIAL



Transformers and Attention Mechanisms

Pontificia Universidad
JAVERIANA
Colombia

Transformers

Attention Is All You Need

Transformers

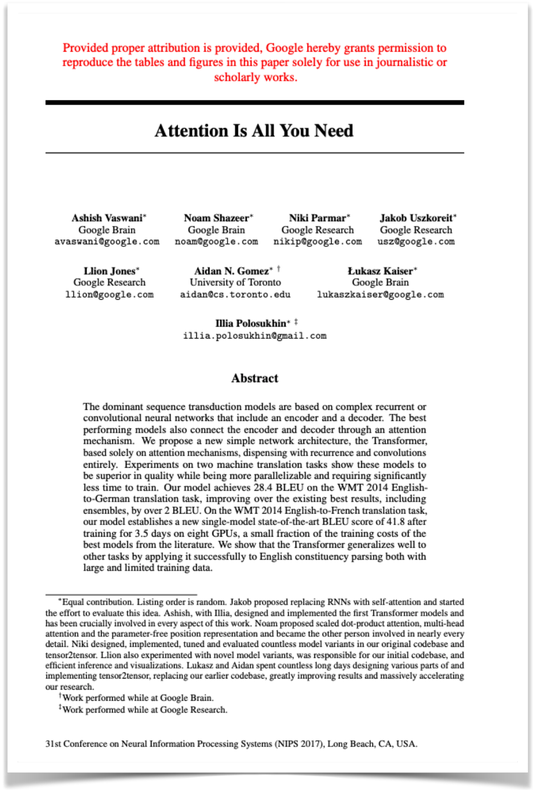
Codificador (encoder)

Decodificador (decodificador)

+

Attention Mechanism

Varias capas ocultas



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Łukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Encoder-Decoder

Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Encoder-Decoder

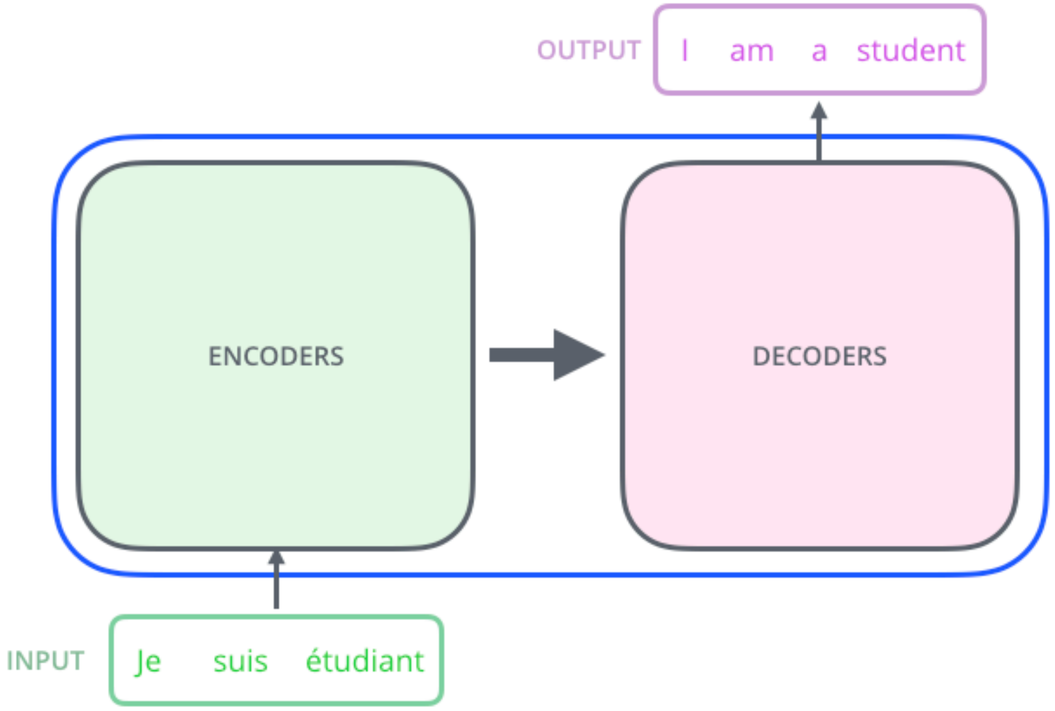
Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica



Encoder-Decoder

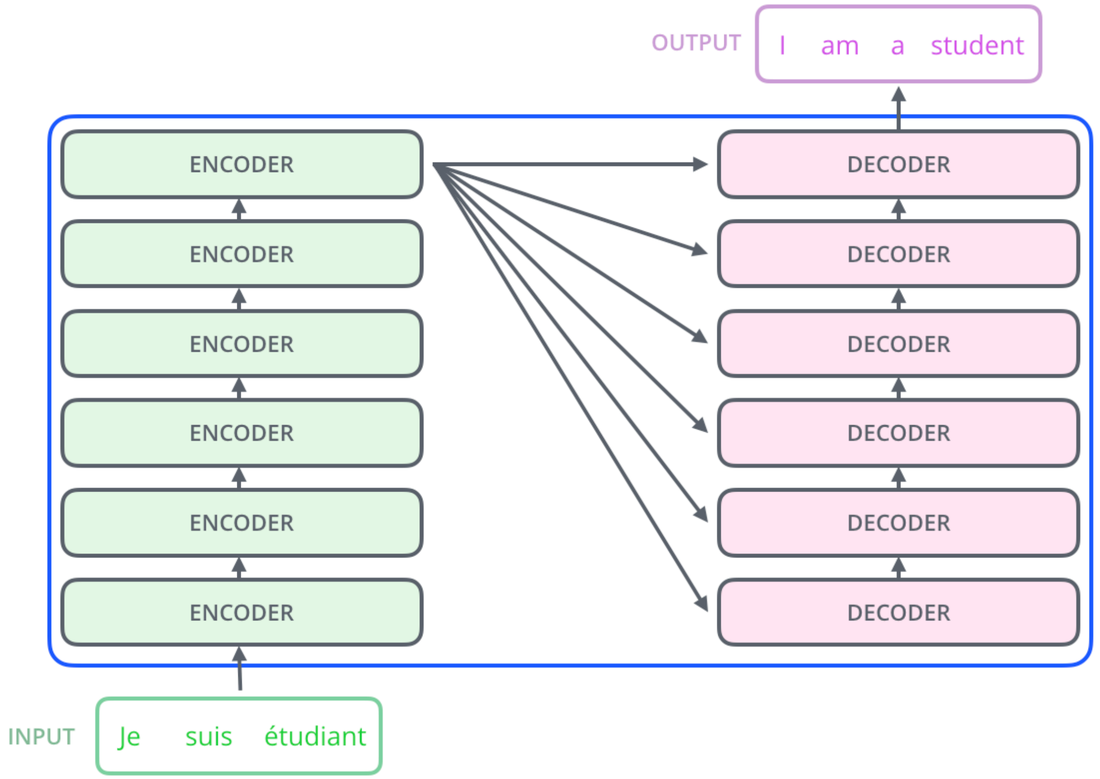
Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica



Encoder-Decoder

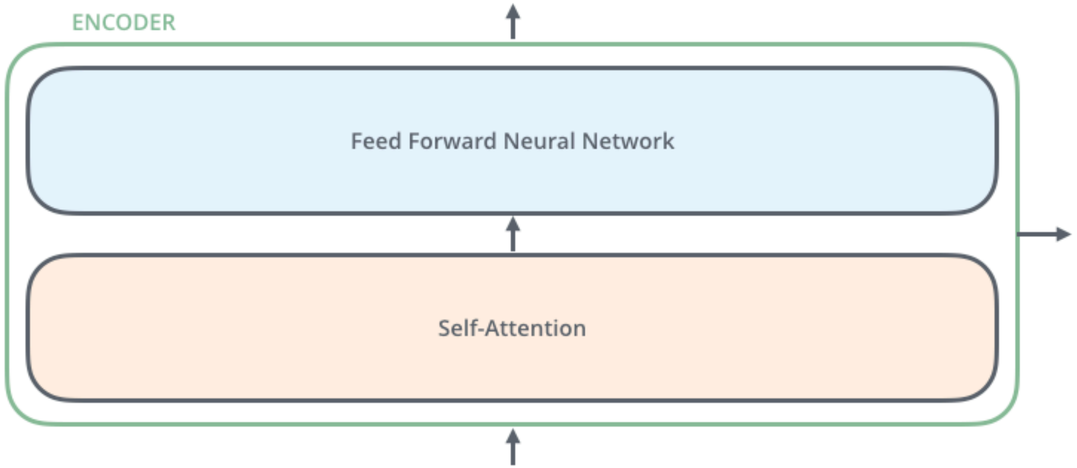
Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica



Cada encoder es igual → *Pero no comparten pesos*

Encoder-Decoder

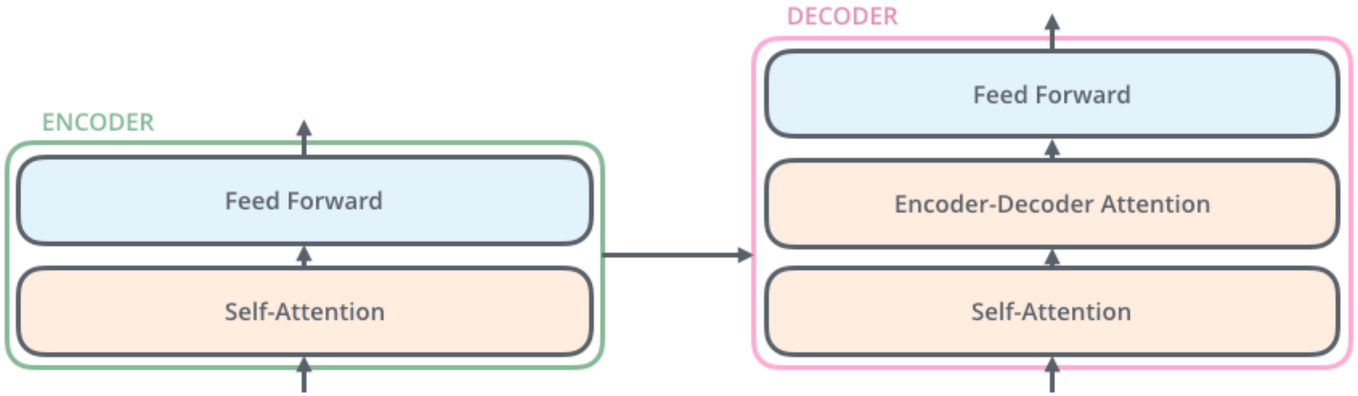
Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica



self-Attention

Encoder-Decoder Attention (seqtoSeq Model)

Encoder-Decoder

Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Word Embedding

- Embedding de 512 dimensiones

- Longitud de la oración más larga en nuestro conjunto de datos

Encoder-Decoder

Self-Attention

Transformers

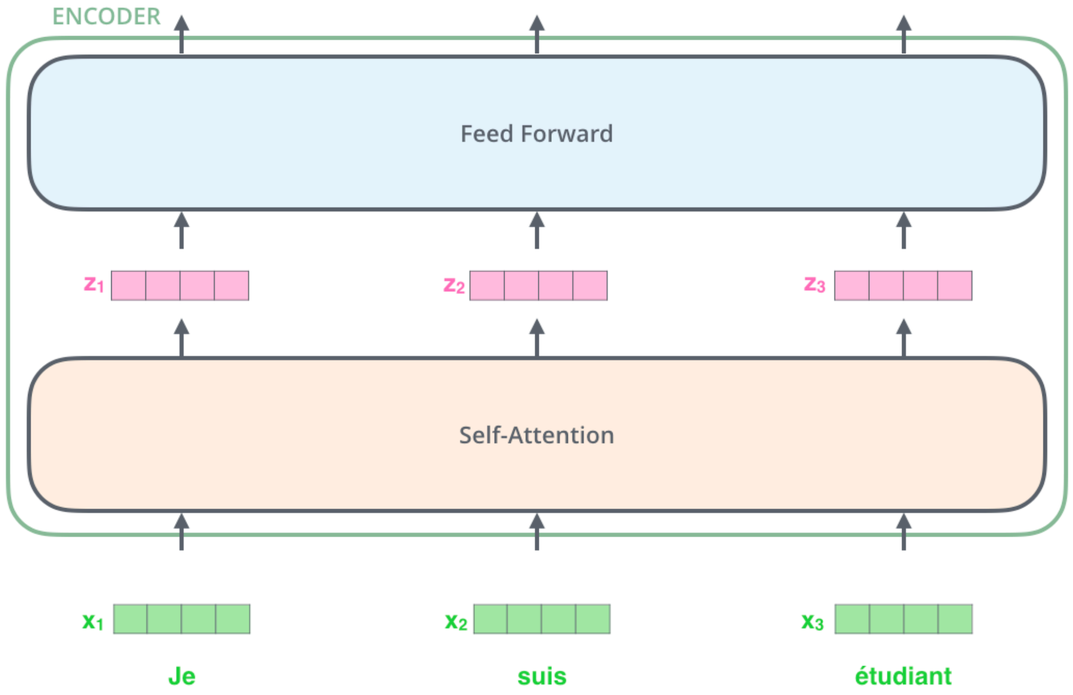
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Word Embedding

Fluyen en paralelo (clave)



Encoder-Decoder

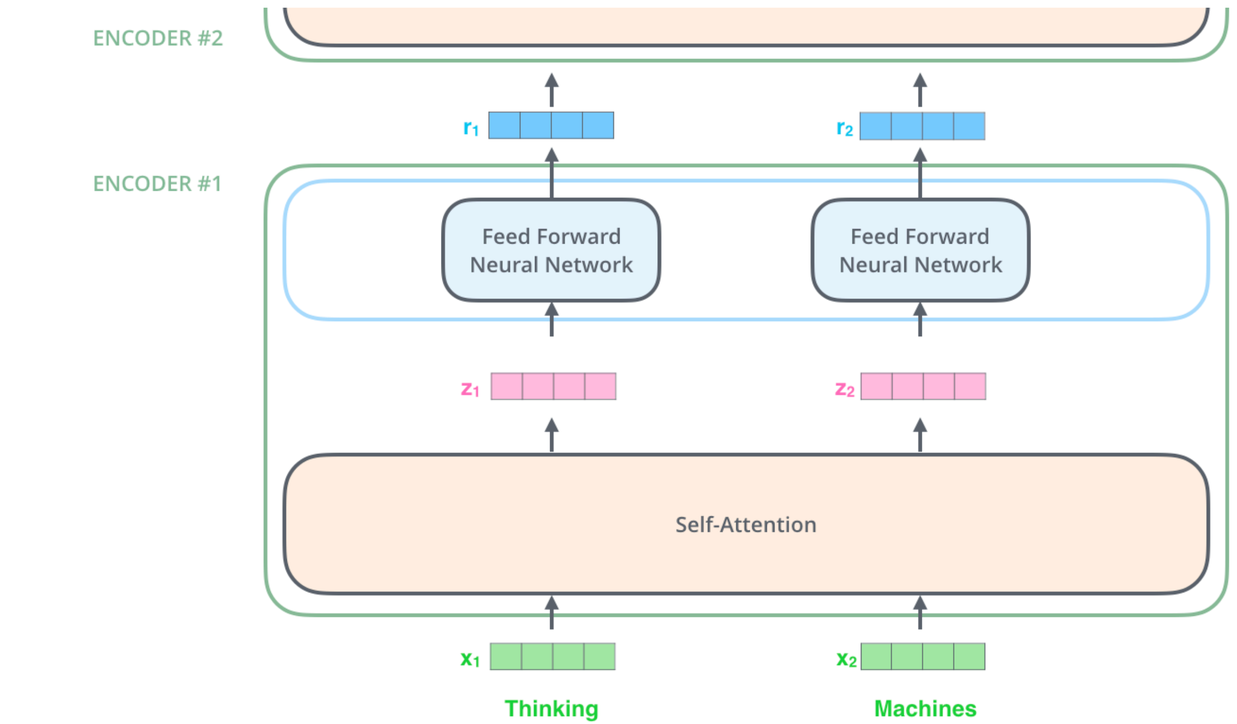
Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica



Encoder-Decoder

Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

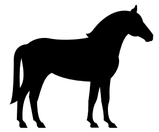
↓

Ej.: Traducción automatica

Self-Attention at a High Level

`The animal didn't cross the street because it was too tired`

¿A qué nos referimos con 'it'?



Encoder-Decoder

Self-Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

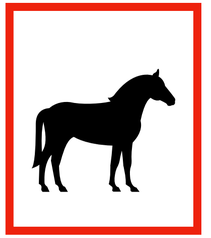
↓

Ej.: Traducción automatica

Self-Attention at a High Level

`The animal didn't cross the street because it was too tired`

¿A qué nos referimos con 'it'?



Encoder-Decoder

Self-Attention

Transformers

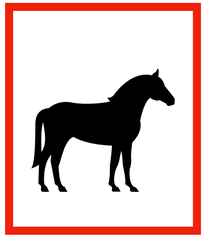
Atención para aumentar la velocidad en entrenamiento de modelos.

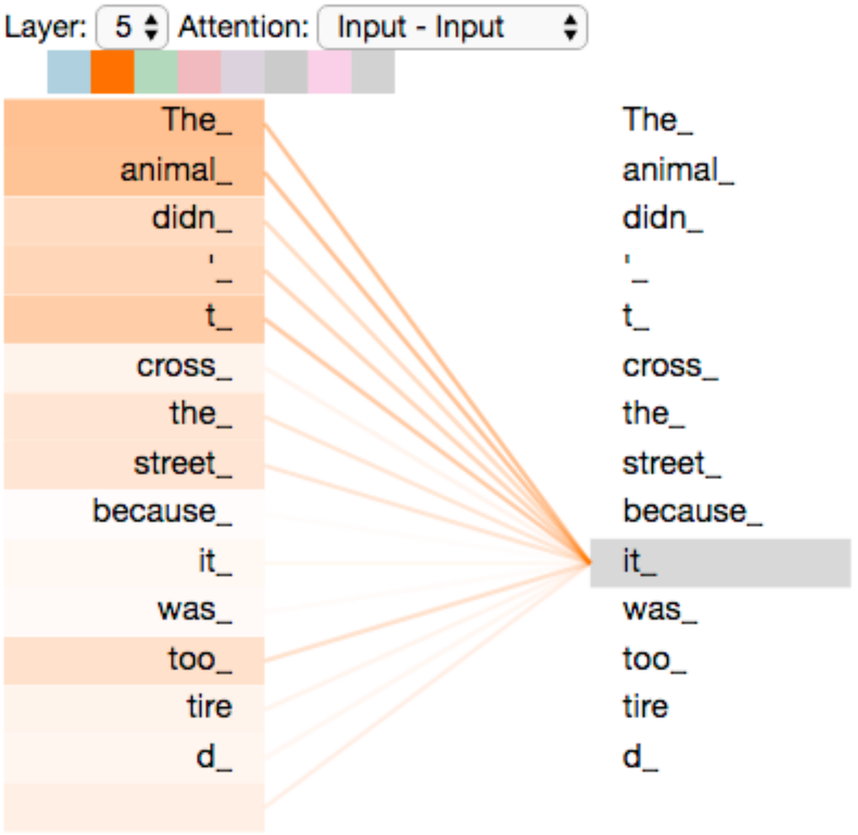
↓

Ej.: Traducción automatica

Self-Attention at a High Level

¿A qué nos referimos con 'it'?





Encoder-Decoder

Self-Attention

Calculo vectorial

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

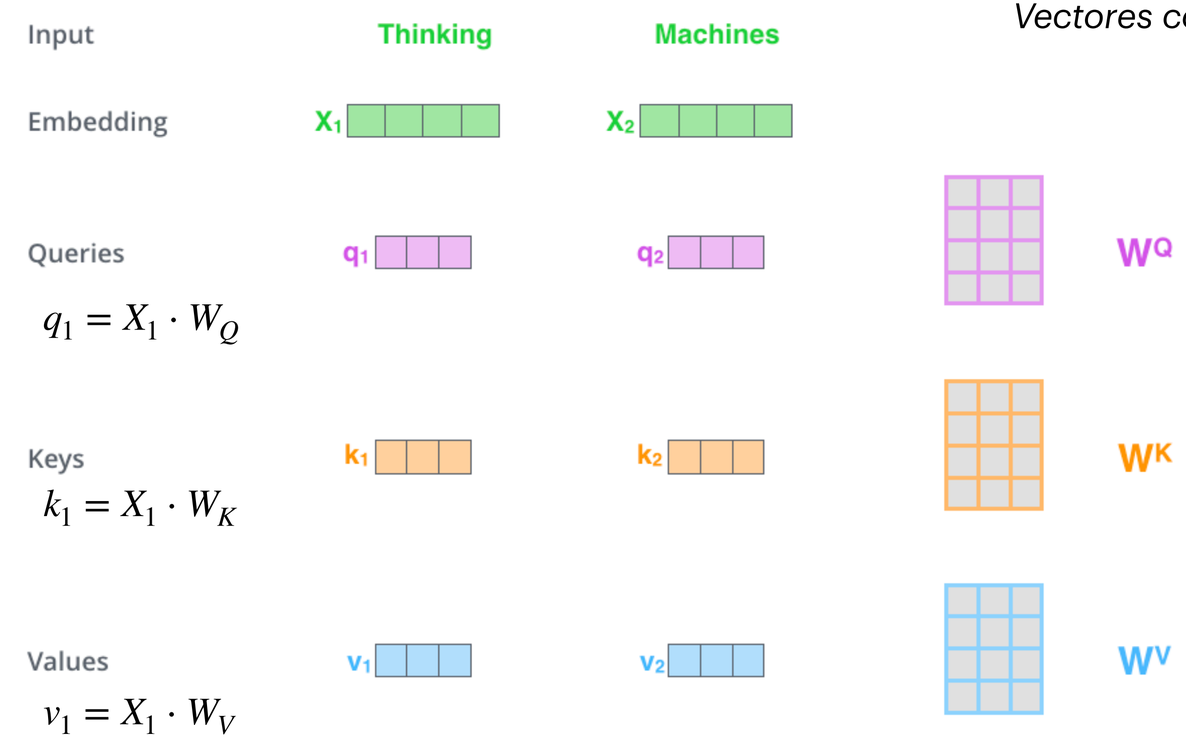
Ej.: Traducción automatica

$$q_1, k_1, v_1 \in palabras$$

Self-Attention in Detail *Implementación de vectores*

Vectores consulta, clave y valor

Vectores con dimension de 64



Encoder-Decoder

Self-Attention

Calculo vectorial

Transformers

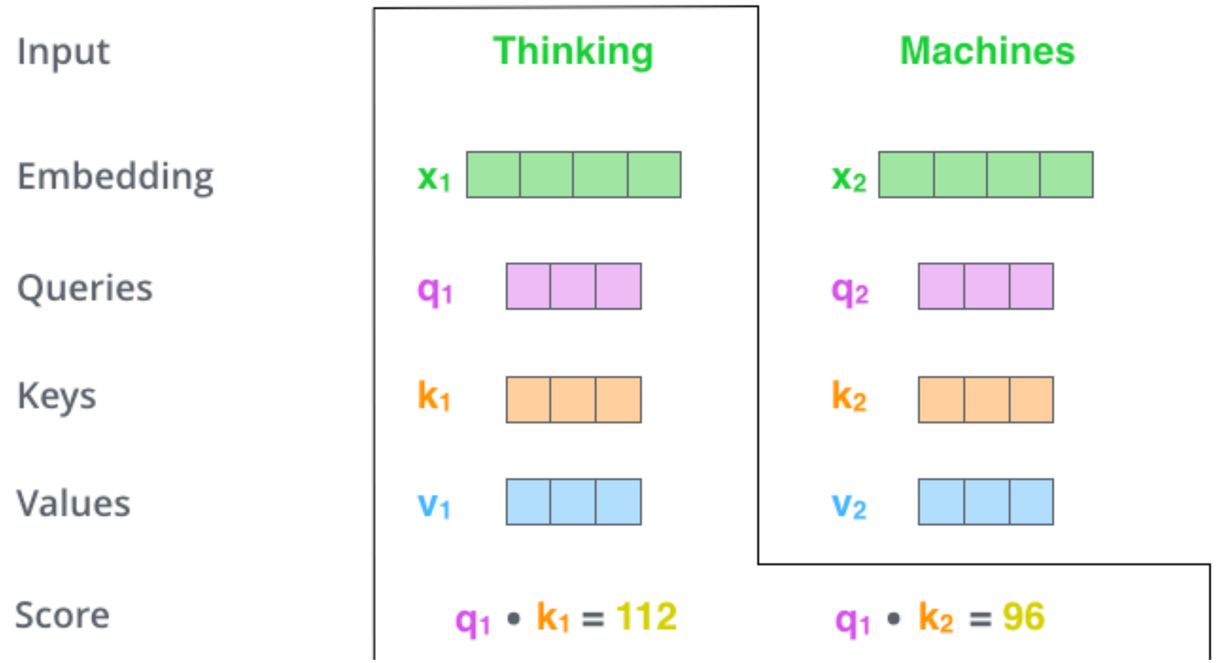
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

$$q_1, k_1, v_1 \in palabras$$

Self-Attention in Detail *Calcular puntuación* *(vectores consulta, clave y valor)*



Encoder-Decoder

Self-Attention

Calculo vectorial

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

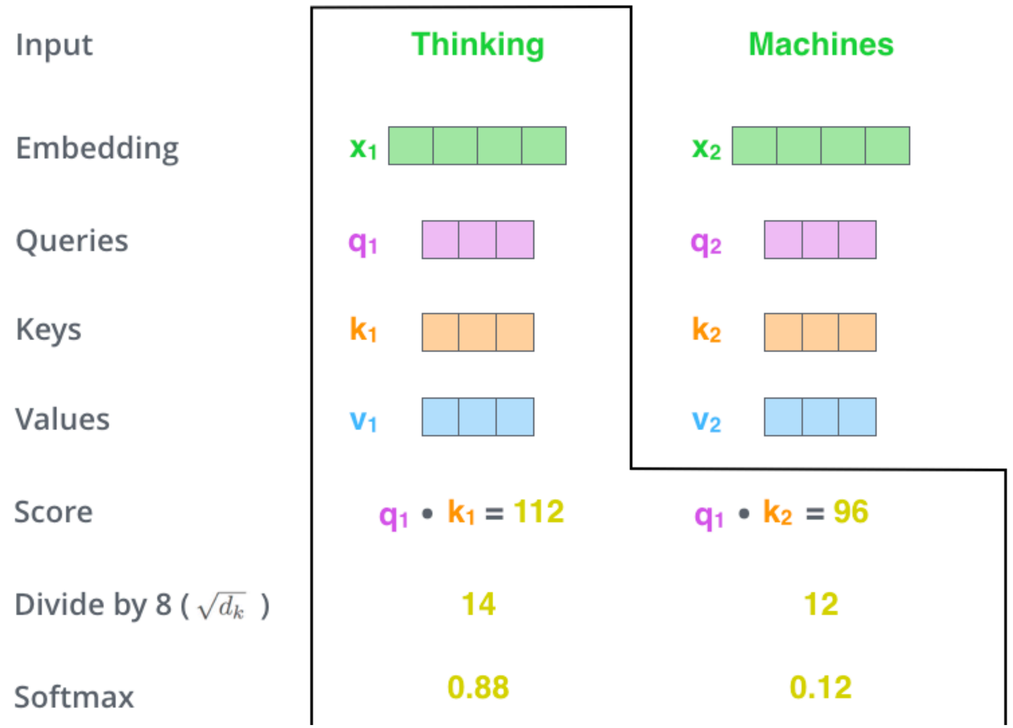
↓

Ej.: Traducción automatica

$$q_1, k_1, v_1 \in palabras$$

Self-Attention in Detail *Normalizar puntuación*

Gradientes estables (sqrt -> 8)



Encoder-Decoder

Self-Attention

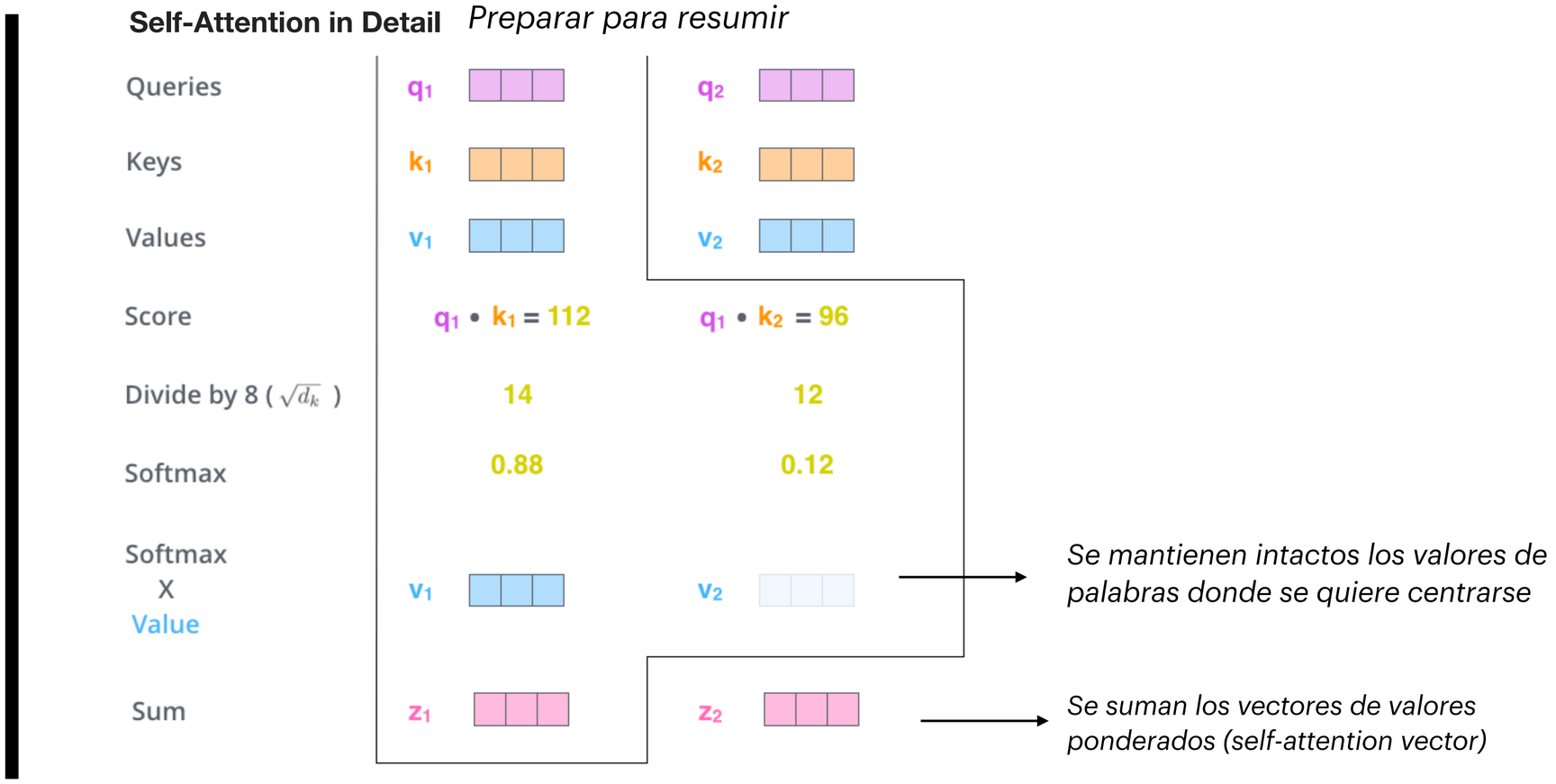
Calculo vectorial

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

Ej.: Traducción automatica

$$q_1, k_1, v_1 \in palabras$$



Encoder-Decoder

Self-Attention

Transformers

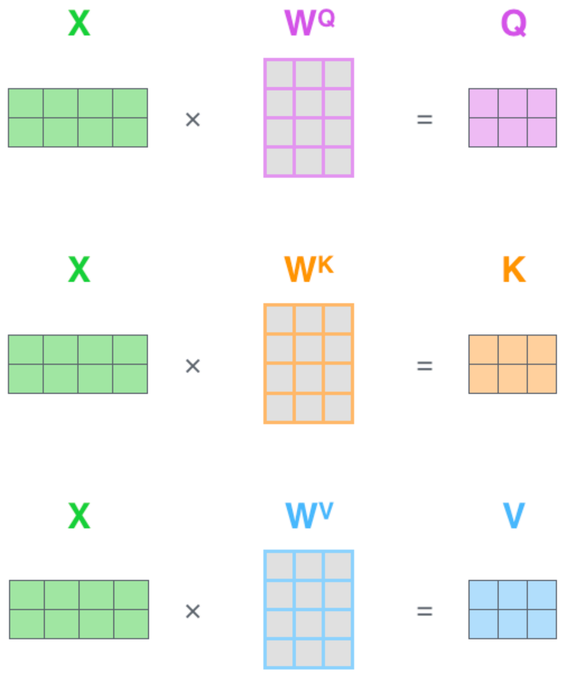
Atención para aumentar la velocidad en entrenamiento de modelos.

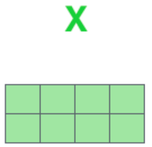
↓

Ej.: Traducción automatica

Matrix Calculation of Self-Attention

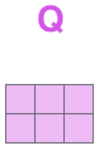
Calculo de matrices de consulta, Clave y Valor.





Dimensión 512

Fila -> palabra



Dimensión 64

Encoder-Decoder

Self-Attention

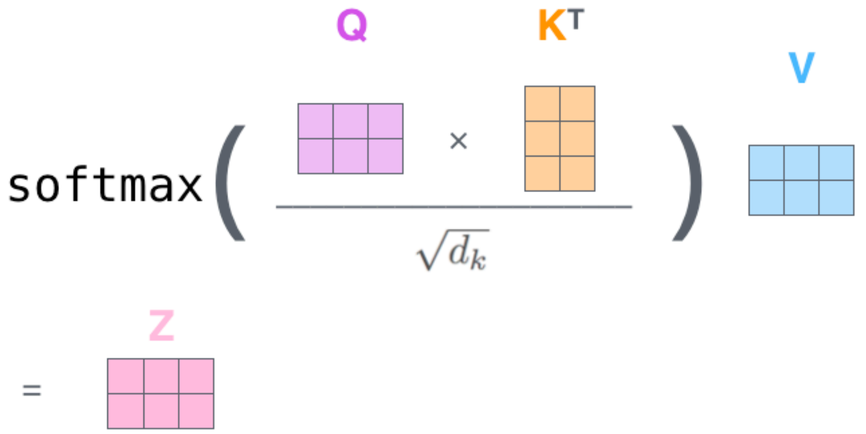
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Matrix Calculation of Self-Attention



Cálculo de la autoatención en forma matricial

Encoder-Decoder

Multi-Head Attention

Transformers

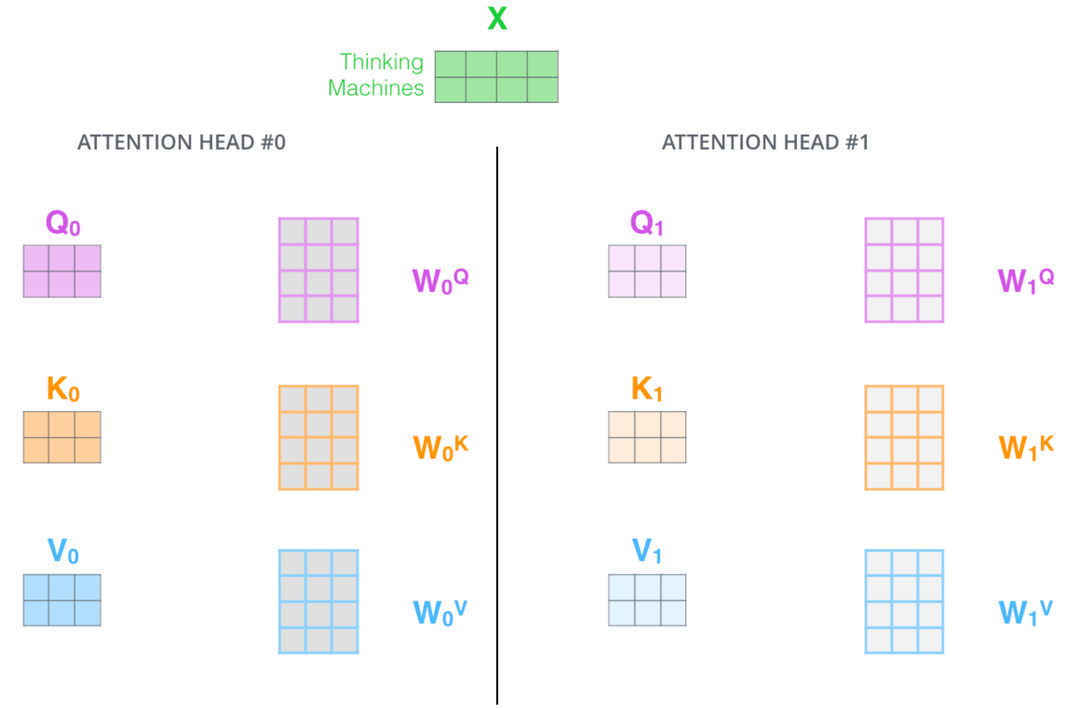
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Atención “multidireccional”

The Beast With Many Heads *Se usan 8 cabezas de atención por cada codificador/decodificador*



Encoder-Decoder

Multi-Head Attention

Transformers

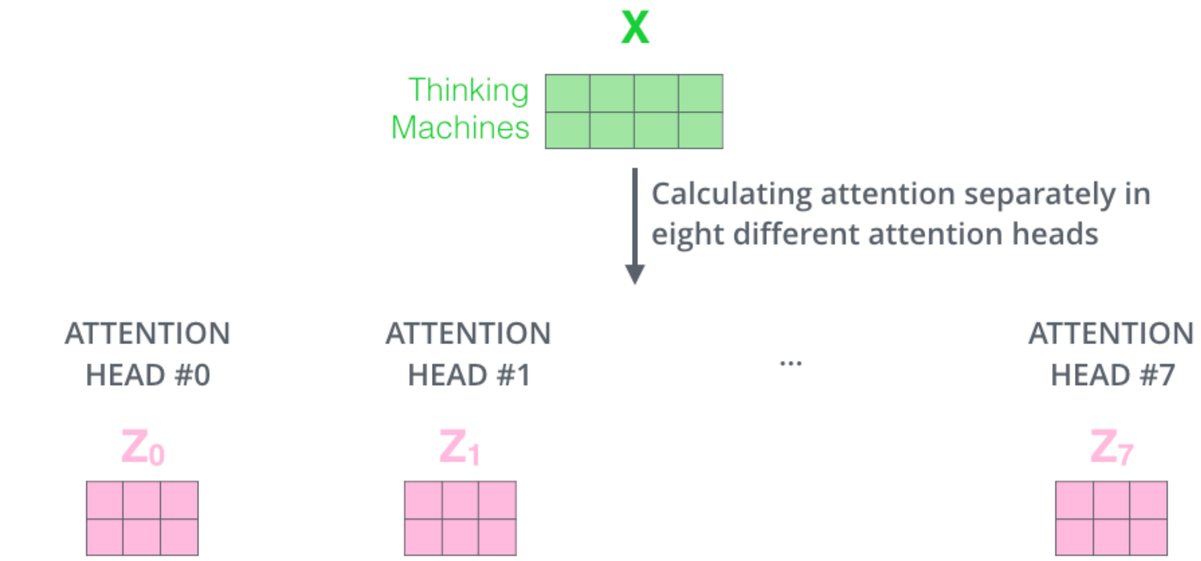
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Atención “multidireccional”

The Beast With Many Heads *Se usan 8 cabezas de atención por cada codificador/decodificador*



Encoder-Decoder

Multi-Head Attention

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

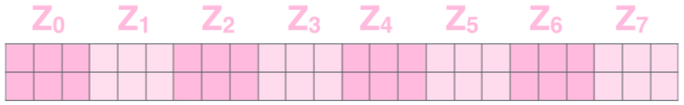
↓

Ej.: Traducción automatica

Atención “multidireccional”

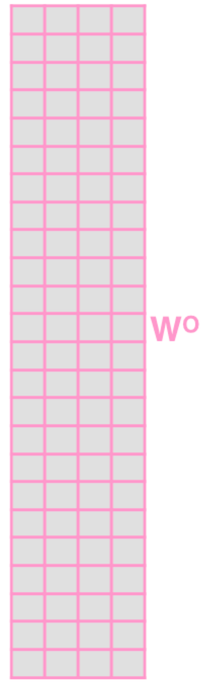
The Beast With Many Heads *Se usan 8 cabezas de atención por cada codificador/decodificador*

1) Concatenate all the attention heads

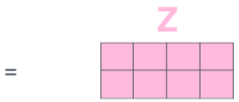


2) Multiply with a weight matrix W^O that was trained jointly with the model

X



3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN



Encoder-Decoder

Multi-Head Attention

Transformers

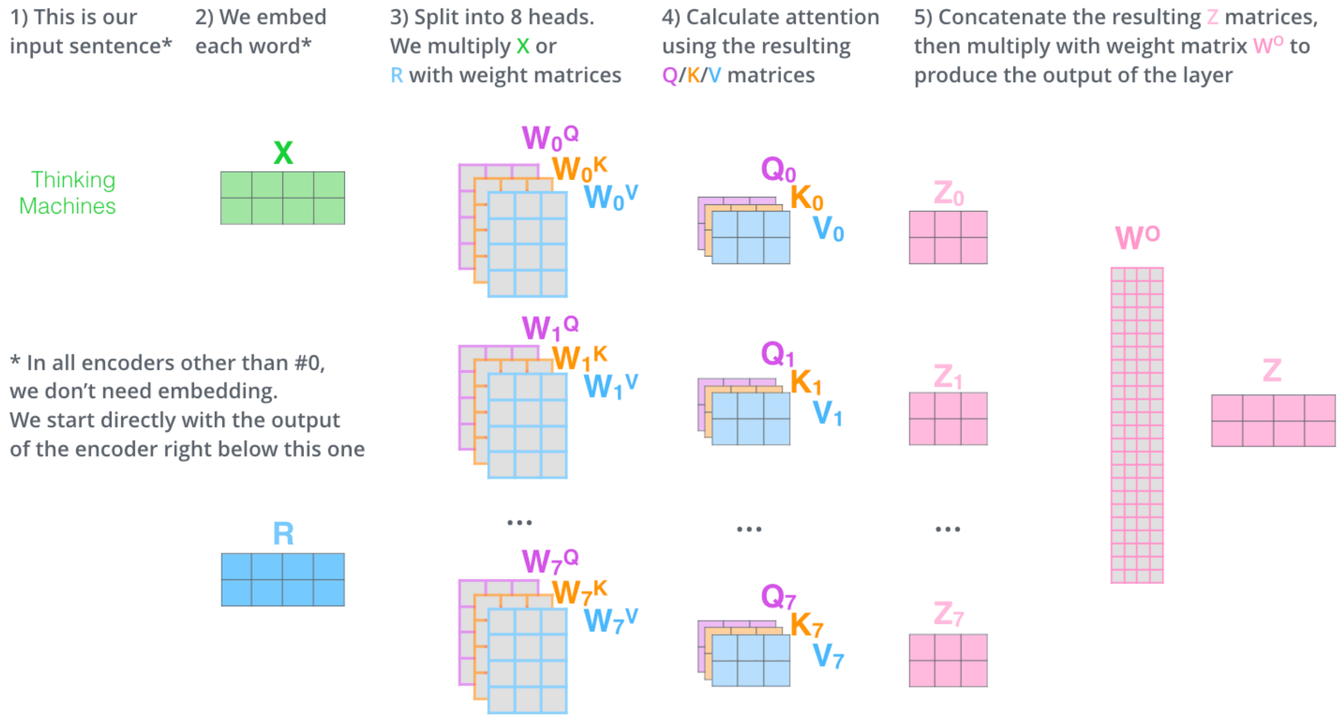
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Atención “multidireccional”

The Beast With Many Heads *Se usan 8 cabezas de atención por cada codificador/decodificador*



Encoder-Decoder

Multi-Head Attention

Transformers

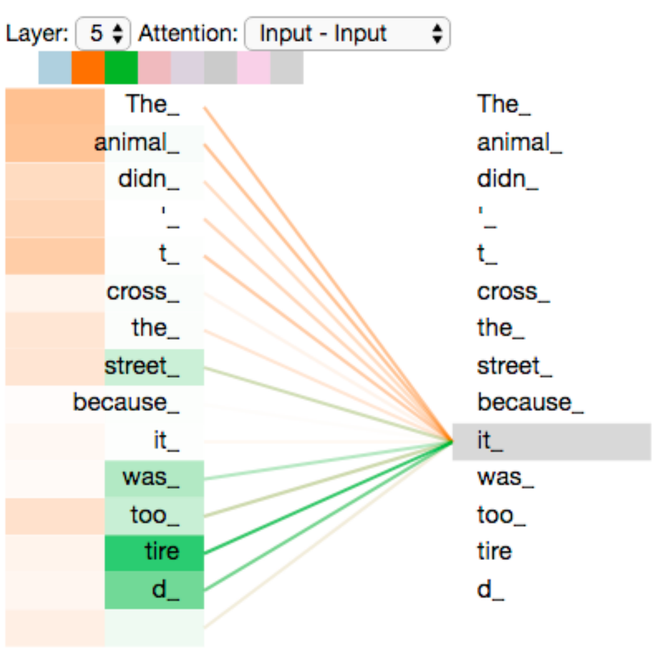
Atención para aumentar la velocidad en entrenamiento de modelos.

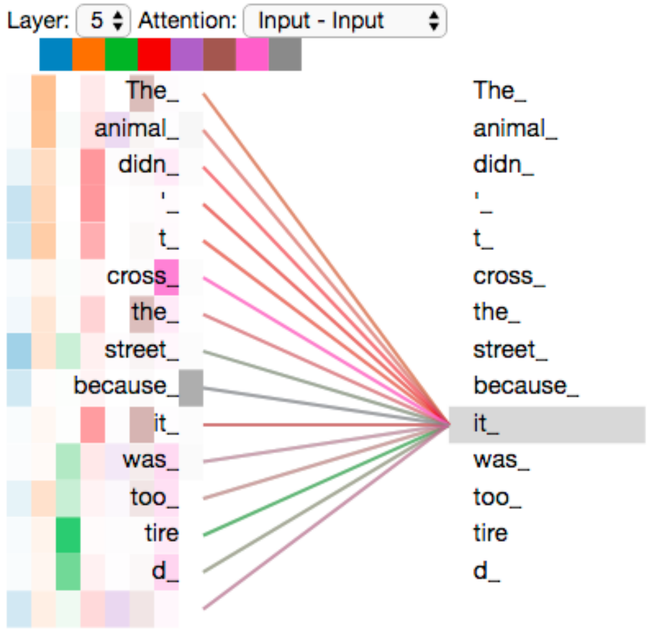
↓

Ej.: Traducción automatica

Atención “multidireccional”

The Beast With Many Heads *Se usan 8 cabezas de atención por cada codificador/decodificador*





Encoder-Decoder

Positional Encoding

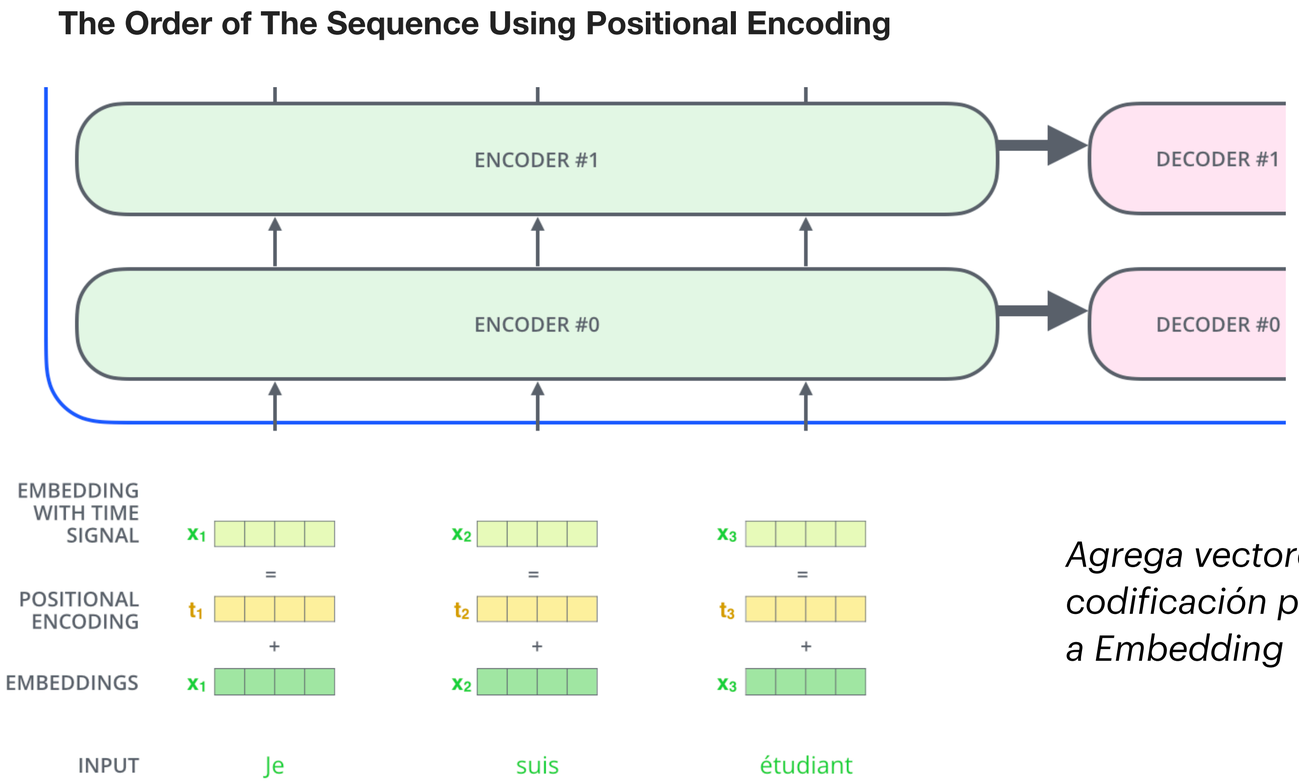
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

'Idea de orden en las palabras'



Agrega vectores de codificación posicional a Embedding

Encoder-Decoder

Positional Encoding

Transformers

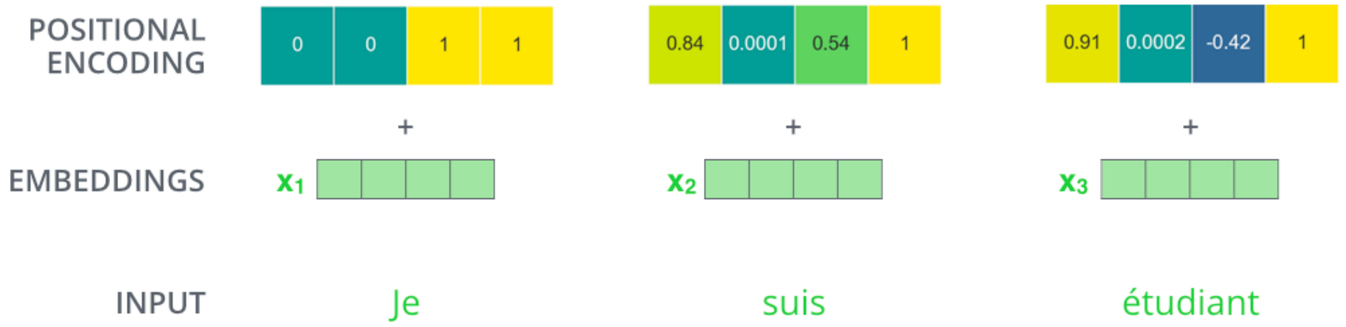
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

'Idea de orden en las palabras'

The Order of The Sequence Using Positional Encoding



Encoder-Decoder

Positional Encoding

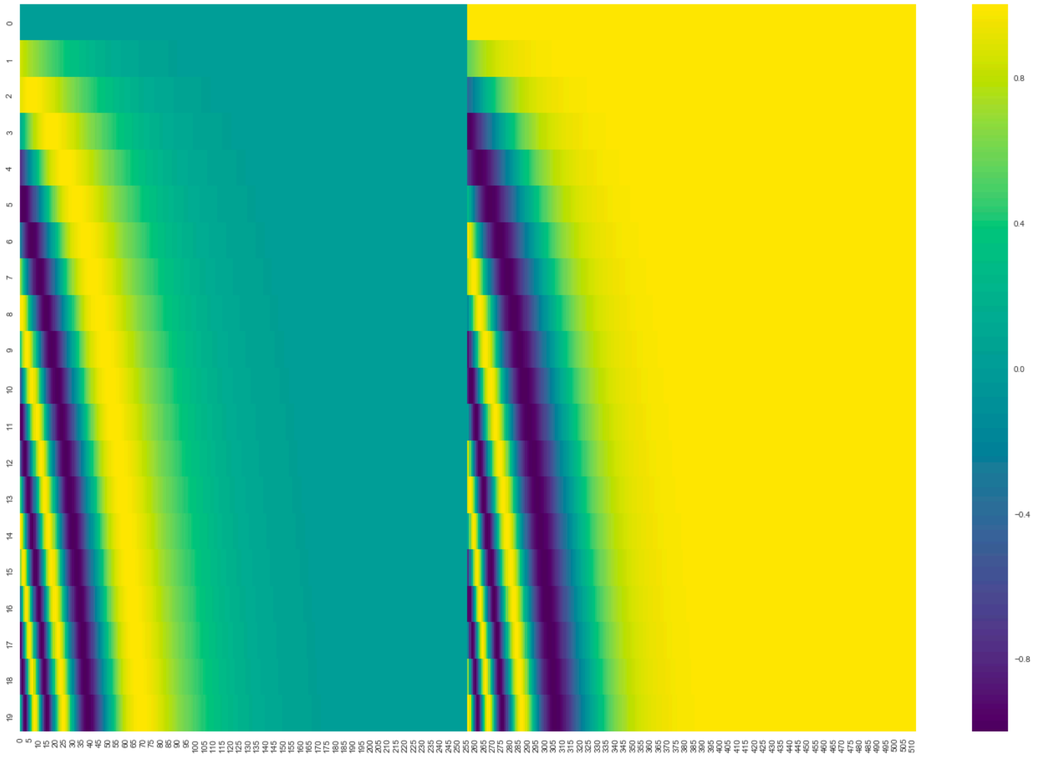
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

Ej.: Traducción automatica

‘Idea de orden en las palabras’

The Order of The Sequence Using Positional Encoding



20 palabras (filas)

512 dimensiones embedding

Valores entre -1 y 1

Función seno

Función coseno

Encoder-Decoder

Positional Encoding

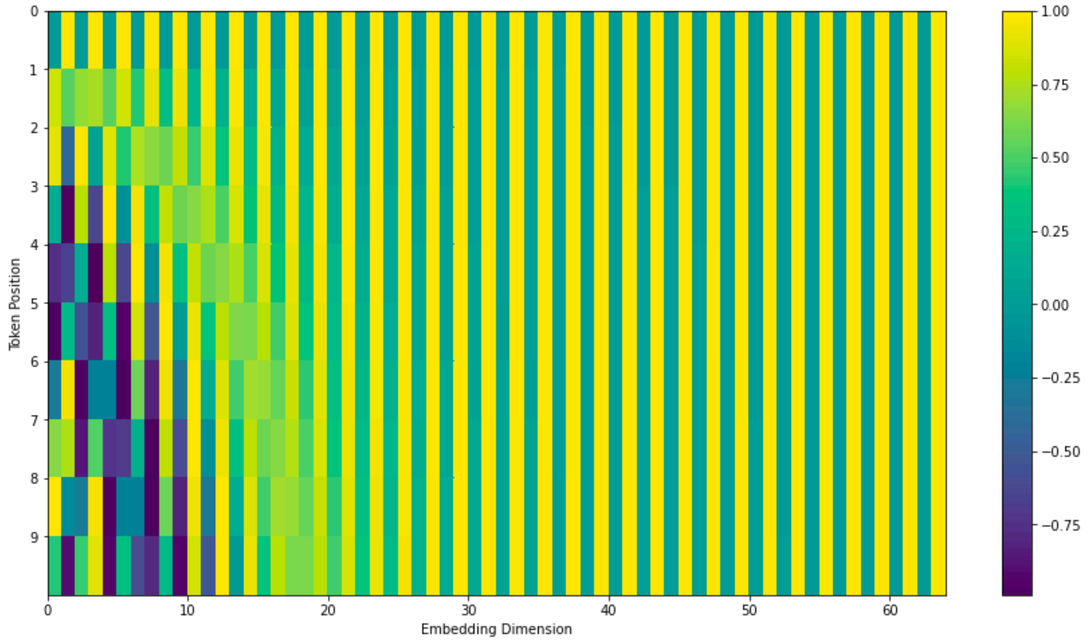
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

'Idea de orden en las palabras'



20 palabras (filas)

512 dimensiones embedding

Valores entre -1 y 1

Entrelazando las dos señales

Función seno

Función coseno

Encoder-Decoder

The Residuals

Transformers

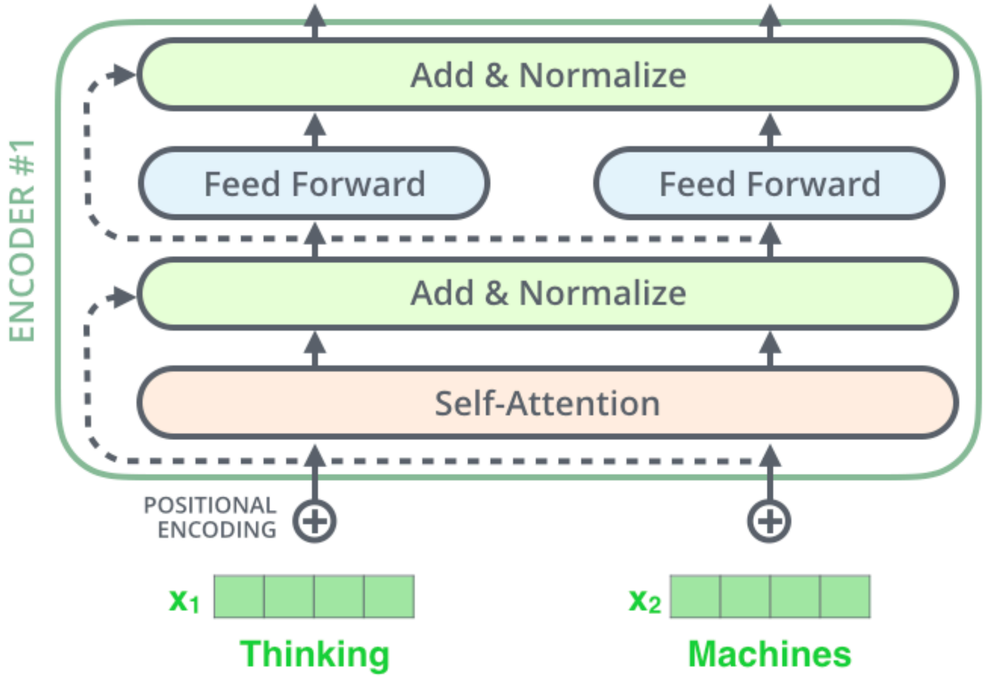
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Normalización entre atención, ffnn

The Order of The Sequence Using Positional Encoding



Conexión residual a una capa de normalización

Encoder-Decoder

The Residuals

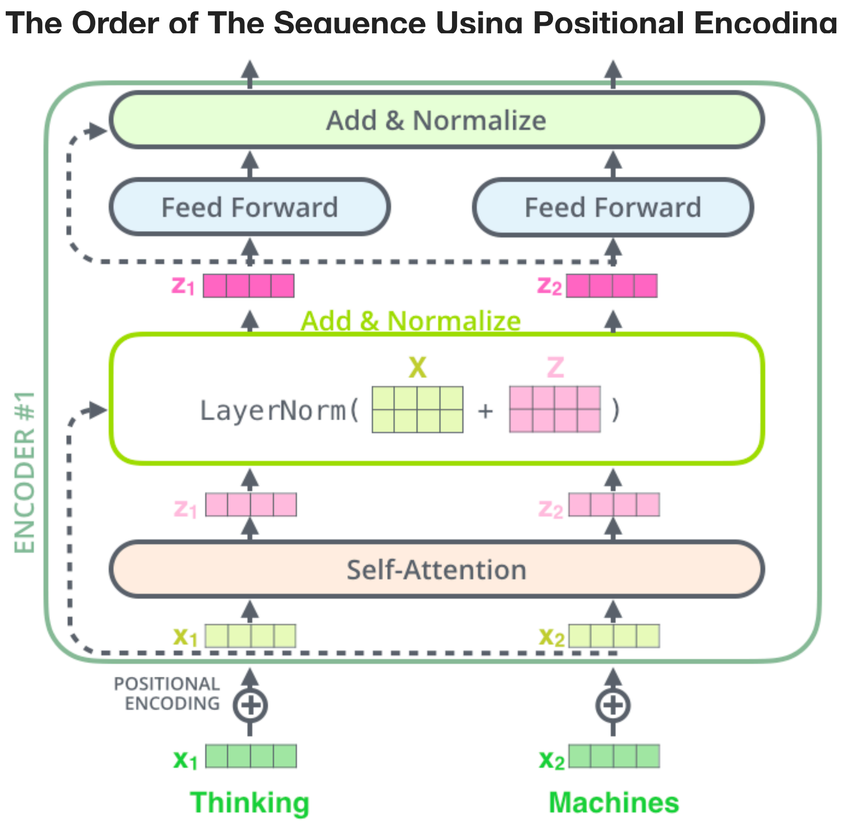
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Normalización entre atención, ffnn



Visualización de parámetros en capa de norma

↓

Esto se extiende a las subcapas del decodificador

Encoder-Decoder

The Residuals

Transformers

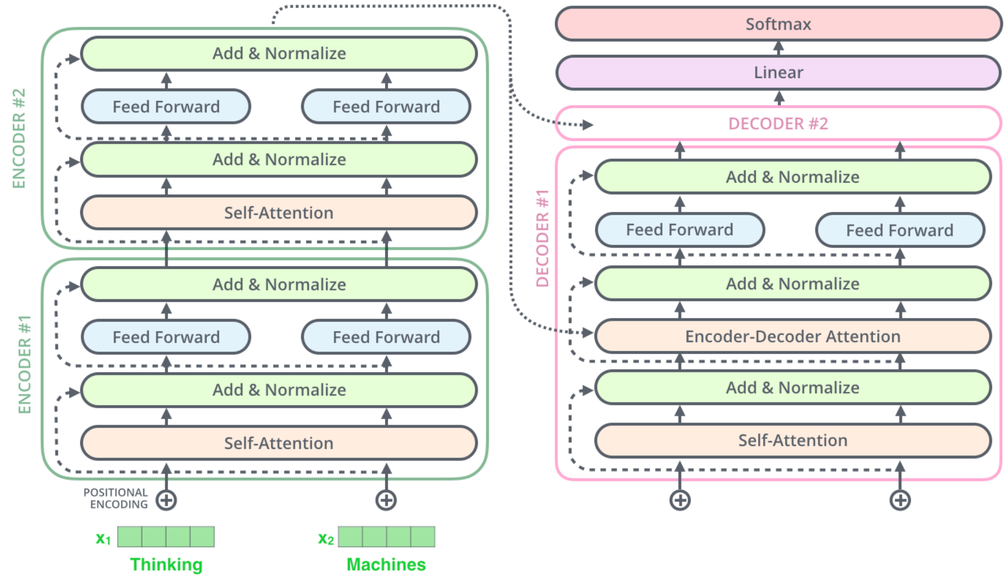
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Normalización entre atención, ffnn

The Order of The Sequence Using Positional Encoding



Visualización de parámetros en capa de norma

↓

Esto se extiende a las subcapas del decodificador

Encoder-Decoder

The Decoder Side

Transformers

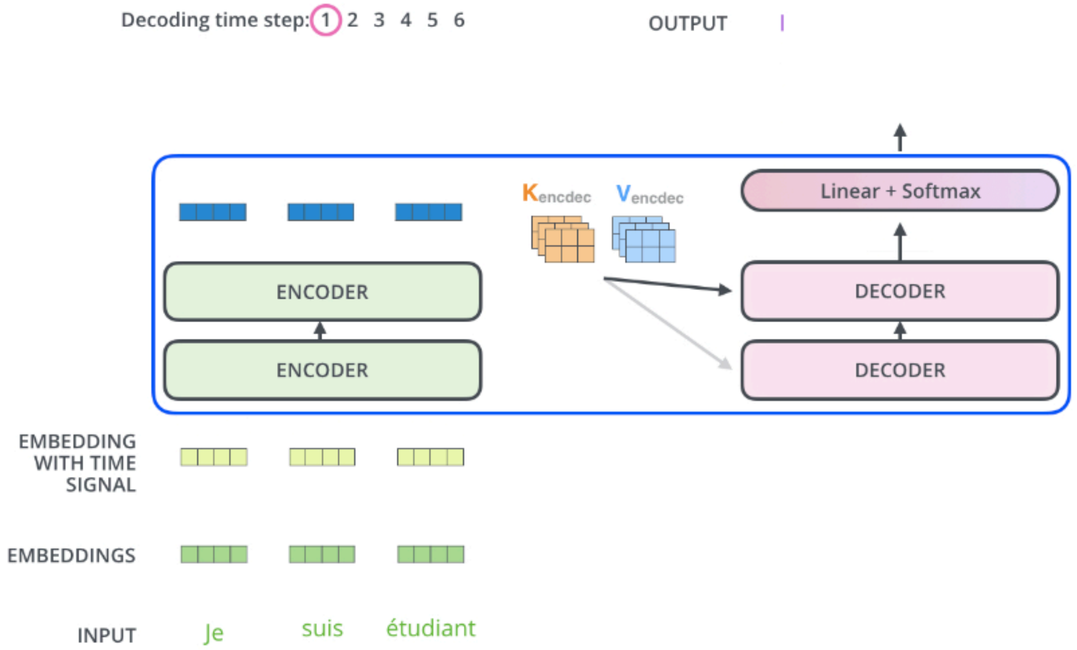
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Normalización entre atención, ffnn

The Decoder



Encoder-Decoder

The Decoder Side

Transformers

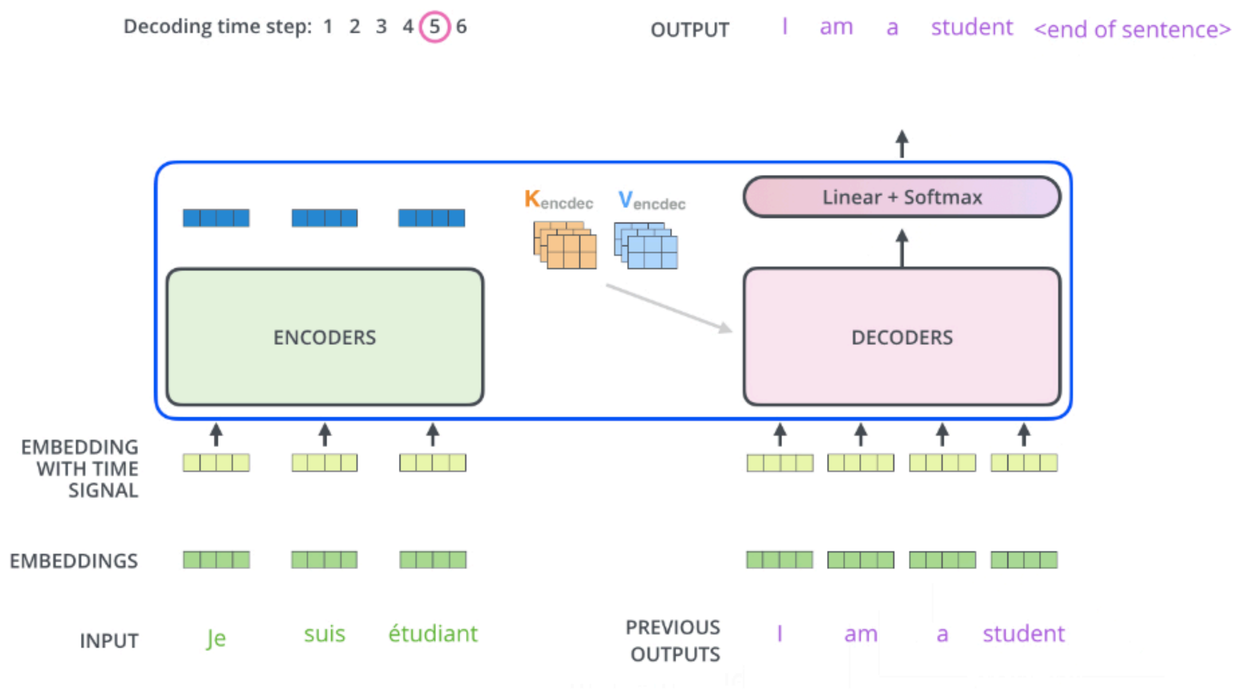
Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Normalización entre atención, ffnn

The Decoder



Encoder-Decoder

The Final

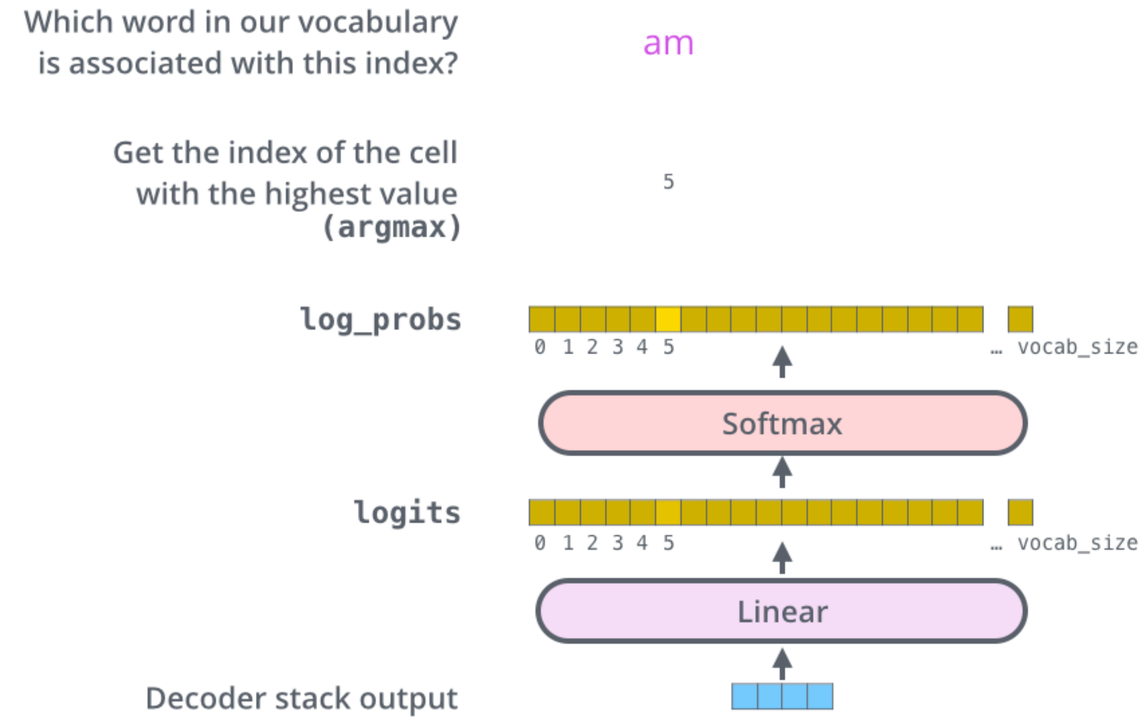
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

The Final Linear and Softmax Layer



Encoder-Decoder

Recap of Training

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

Ej.: Traducción automatica

Output vocabulary *El vocabulario de salida del modelo, se hace en la fase de preprocesamiento*

Output Vocabulary

WORD	a	am	I	thanks	student	<eos>
INDEX	0	1	2	3	4	5

Encoder-Decoder

Recap of Training

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Output vocabulary *El vocabulario de salida del modelo, se hace en la fase de preprocesamiento*

Output Vocabulary

WORD	a	am	I	thanks	student	<eos>
INDEX	0	1	2	3	4	5

One-hot encoding of the word “am”

0.0	1.0	0.0	0.0	0.0	0.0

Encoder-Decoder

Recap of Training

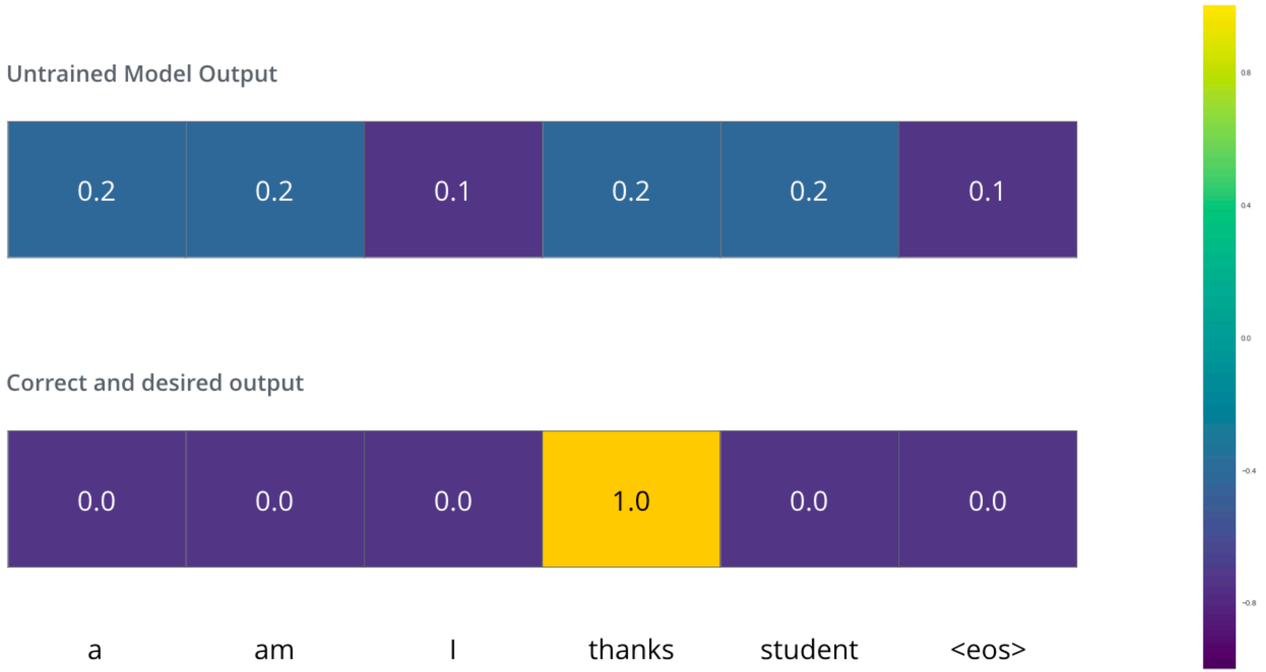
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Loss function



Encoder-Decoder

Recap of Training

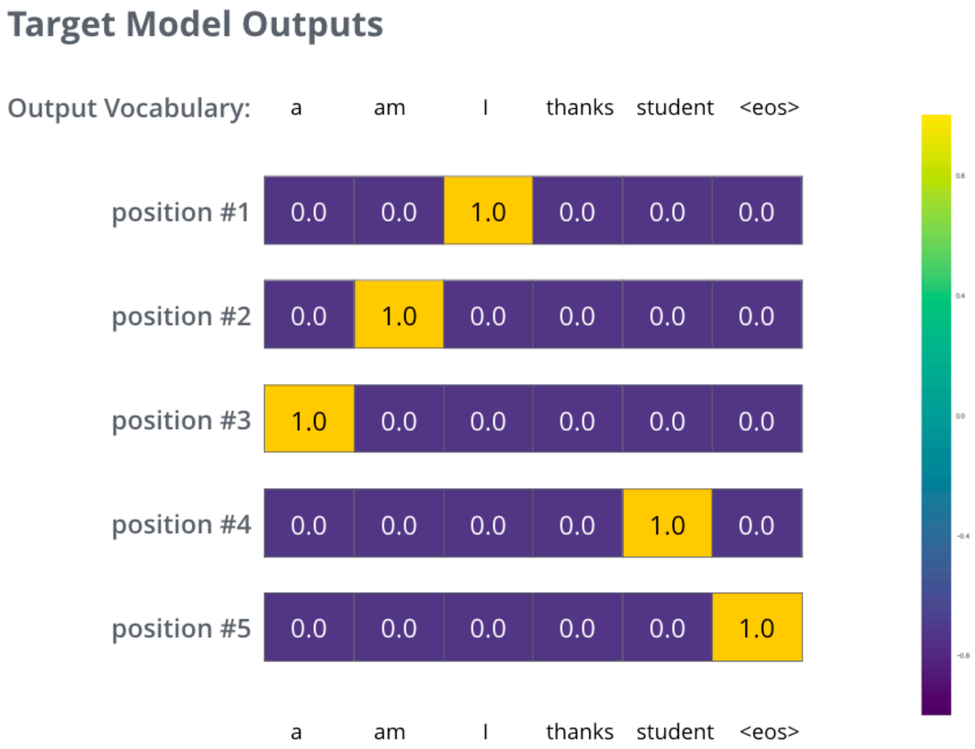
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Loss function



Encoder-Decoder

Recap of Training

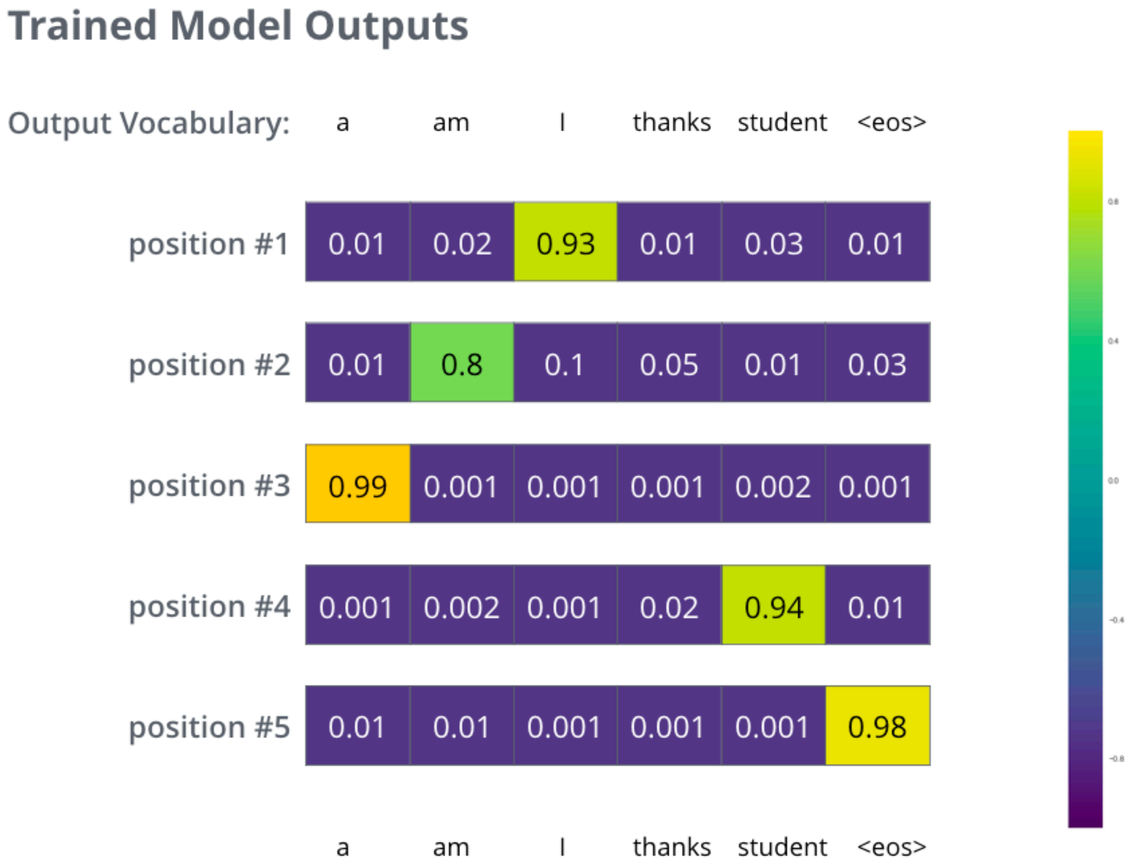
Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Loss function



Encoder-Decoder

Recap of Training

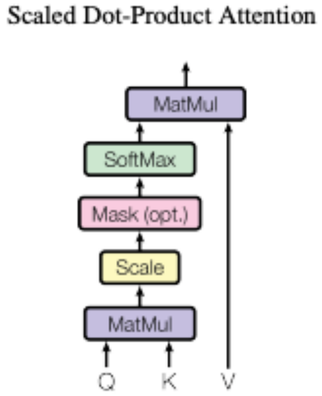
Transformers

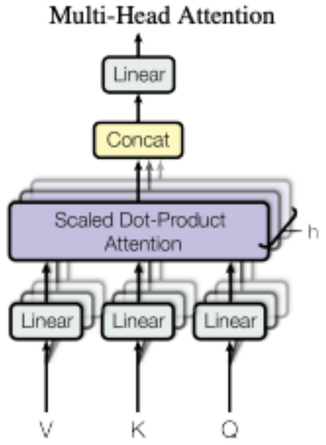
Atención para aumentar la velocidad en entrenamiento de modelos.

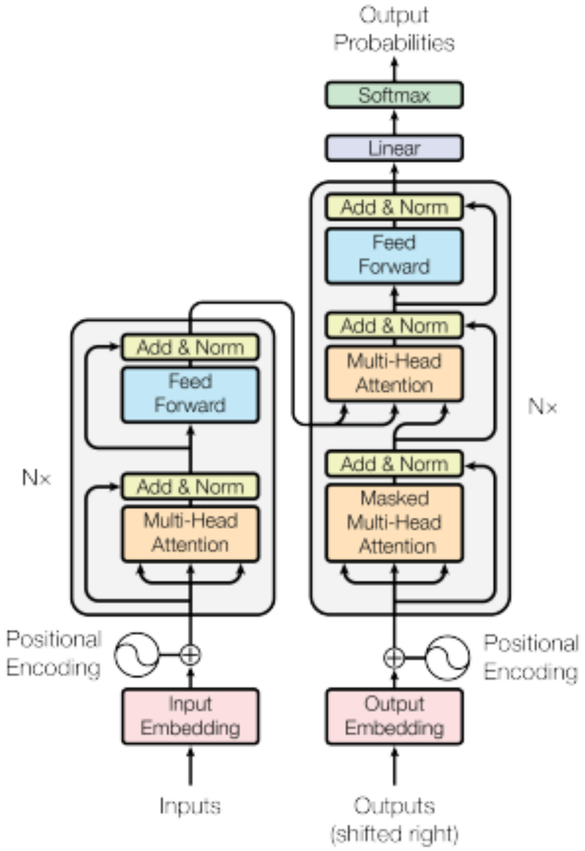
↓

Ej.: Traducción automatica

The Transformer - model architecture.







Encoder-Decoder

Recap of Training

Transformers

Atención para aumentar la velocidad en entrenamiento de modelos.

↓

Ej.: Traducción automatica

Loss function

Kullback-Leibler Divergence Explained

Entropia Cruzada