For today's session, kindly clone the **GitHub** repository

https://github.com/aim-msds/msds-2025ft-ml3-transformers

Then follow the setup procedures.

ASIAN INSTITUTE OF MANAGEMENT

Machine Learning 3

Special Topics: Transformers

Session 17

MSDS FT 2025

30 May 2025

ASIAN INSTITUTE OF MANAGEMENT

Session 17 and 18

Gameplan

9:00 AM to 10:30 AM	**Special Topics Lecture Part 1**
10:30 AM to 11:00 AM	**Break**
11:00 AM to 11:45 AM	**Special Topics Lecture Part 2**
11:45 AM to 12:30 PM	**Special Final Project Consultation**

Transformers: Key Concepts

Transformers: Key Concepts

1 Attention

- Query, Key, Values – QKV
- Self-attention

2 Architectural Patterns

- Dense Projections
- Multi-head Attention
- Layer Normalization
- Residual Connections

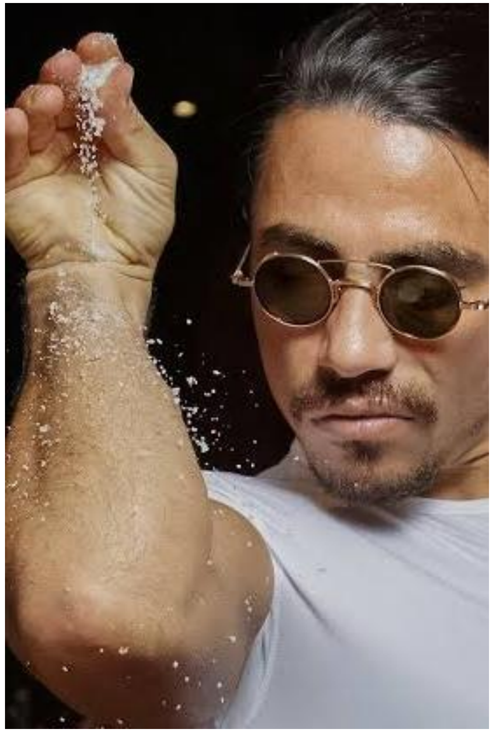
3 Additional Essential Tricks

- Positional Encoding
- Causal Padding

Where were you in 2017?

Ed Sheeran
"Shape of You"

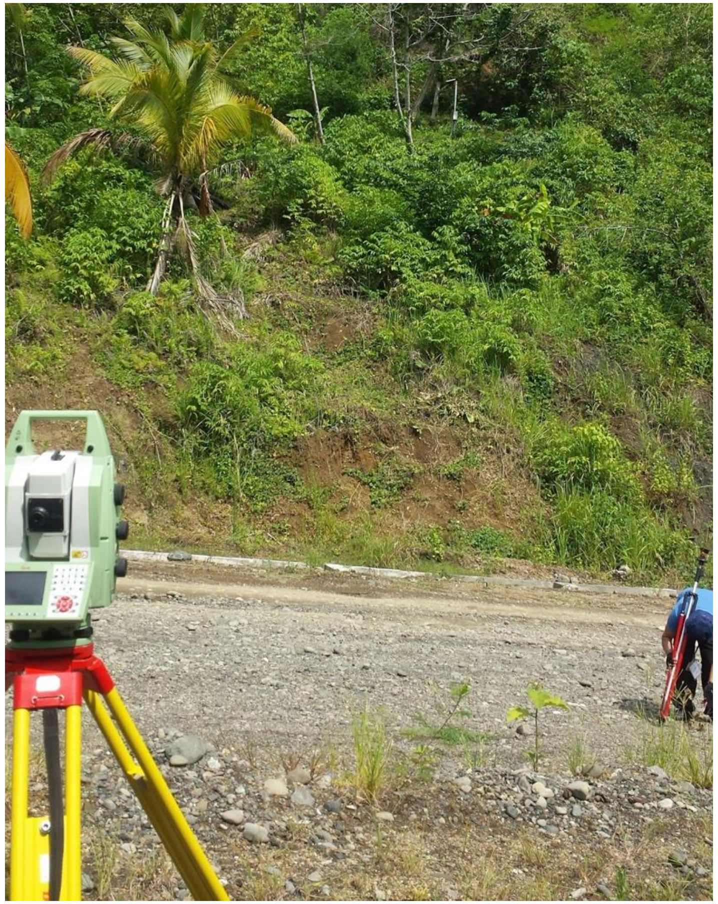
Fidget Spinners

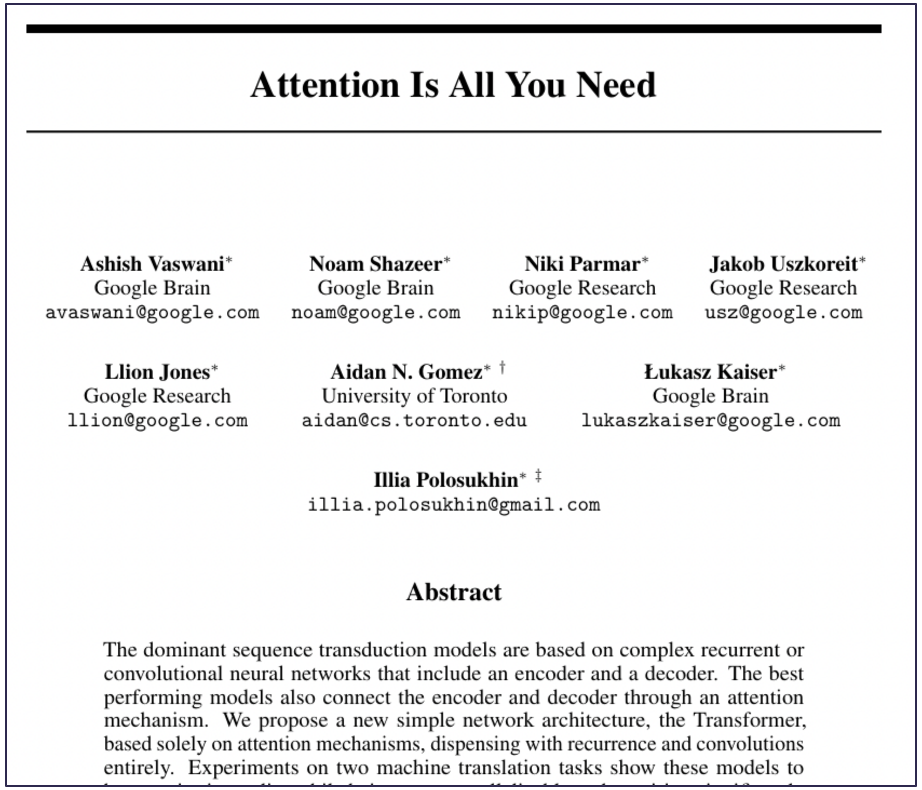


Salt Bae was THE **meme**

AIM

Where were you in 2017?





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

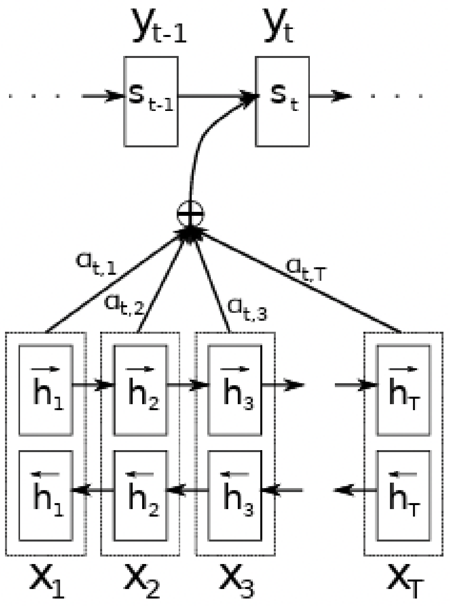
Abstract

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

“Attention Is All You Need”

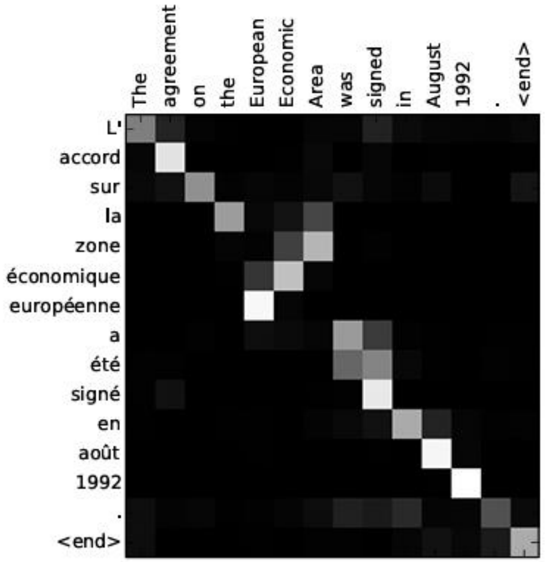
NIPS 2017, Long Beach, CA, USA

Attention Is All You Need

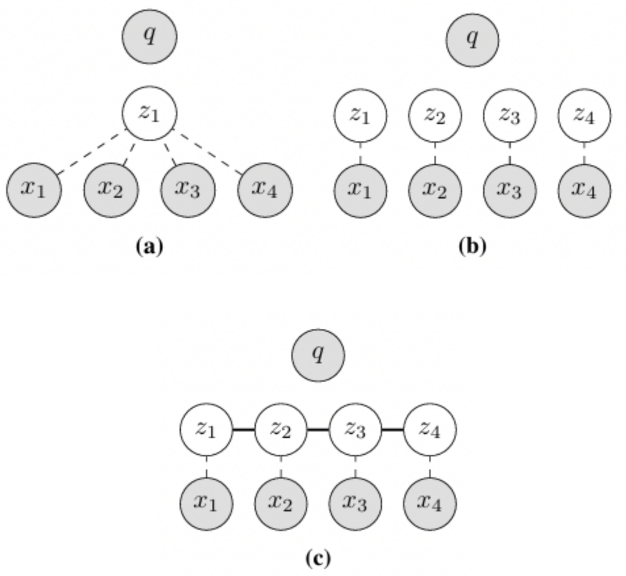


Neural Machine Translation by Jointly Learning to Align and Translate

Bahdanau et al., ICLR 2015



Bi-directional RNNs; Bi-directional LSTMs

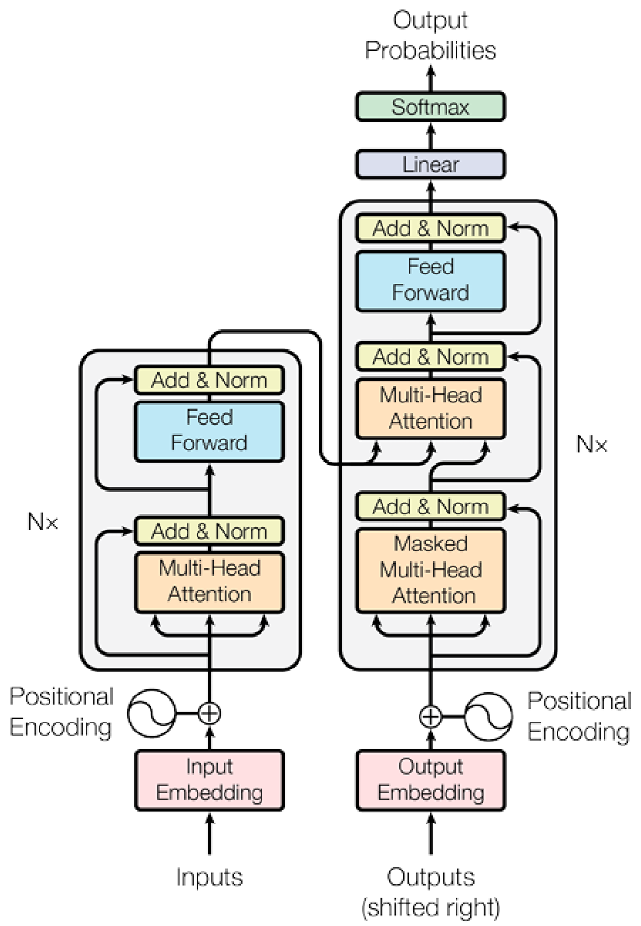


Structured Attention Networks

Kim et al., ICLR 2017

Attention Is All You Need

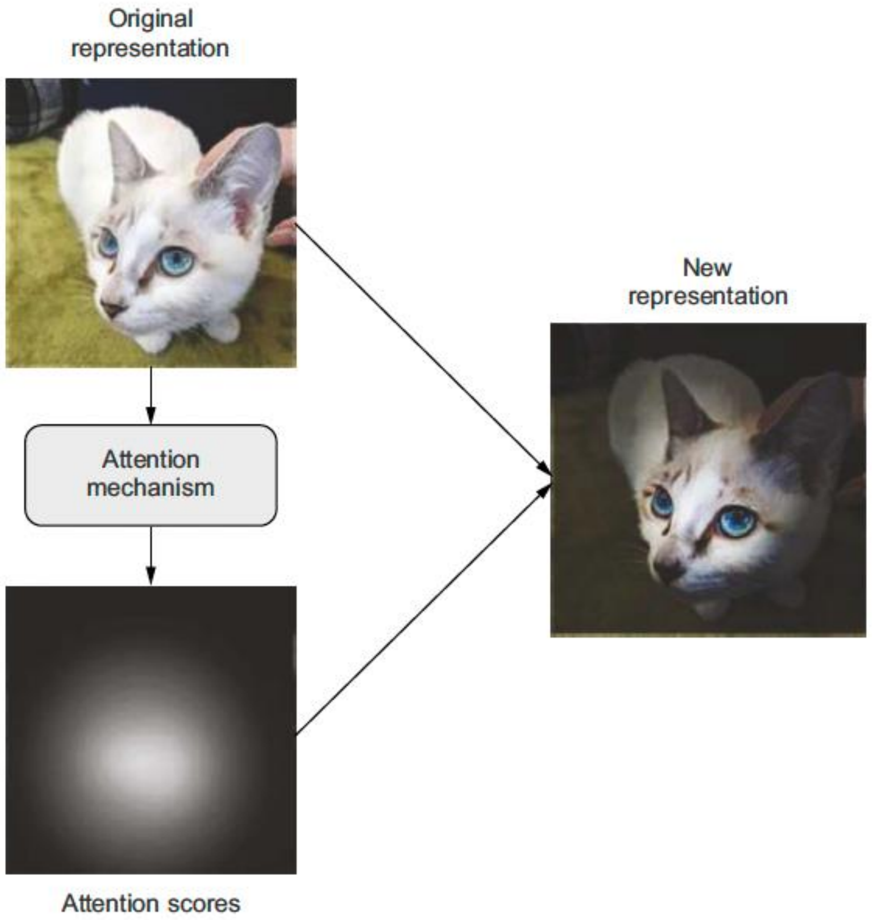
“In this work we propose the ***Transformer***, a model architecture ***eschewing*** recurrence and instead relying entirely on **attention** mechanism to draw global dependencies between input and output.”



The Transformer – model architecture

Understanding Attention

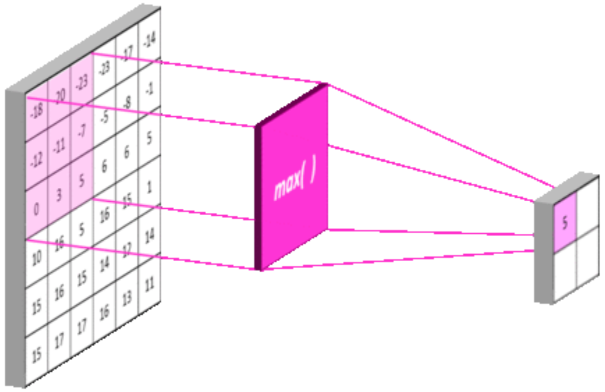
Key idea: “not all input information seen by a model is *equally important* to the task at hand, so models should ***‘pay more attention’*** to some features and *less* to other features”



Understanding Attention

Attention-like Architectural Patterns

Max Pooling



TF-IDF

Term Frequency × Inverse Document Frequency

	the	movie	is	very	fun	and	exciting
TF-IDF	0.00	0.00	0.00	0.68	0.25	0.00	0.68

Understanding Attention

Transformer-style Attention: QKV

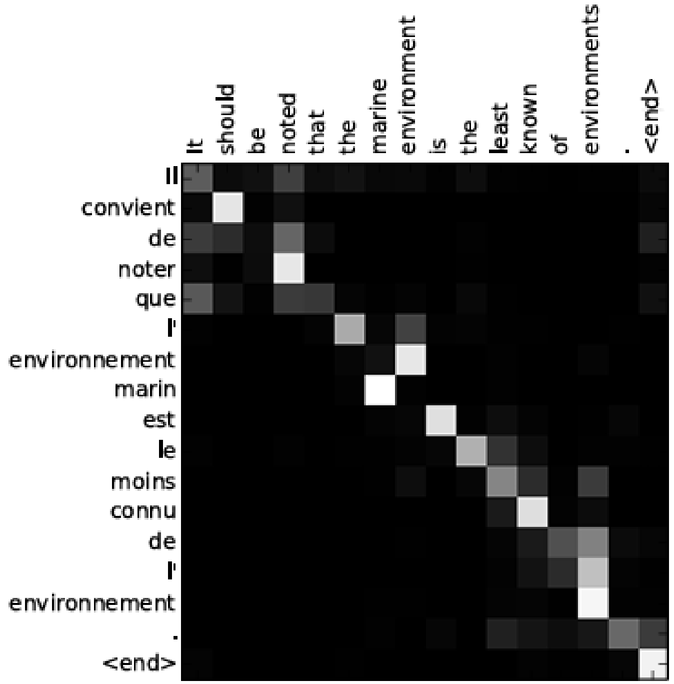
Query	“dogs on the beach”		
Keys	**Beach Tree Boat**	**Beach Tree Dog**	**Dog Grass**
Values			
Output	**0.5**	**1.0**	**0.5**

Understanding Attention

Transformer-style Attention: QKV

Query 🔎 “Il convient de noter que l’ environnement marin est le moins connu de l’ environnement.”

Keys & Values “It should be noted that the marine environment is the least known of environments.”



Understanding Attention

Transformer-style Attention: QKV

Scaled Dot-Product Attention

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

outputs = sum(**values** × pairwise_scores(**query**, **keys**))

Understanding Attention

Transformer-style Attention: QKV

Self-attention

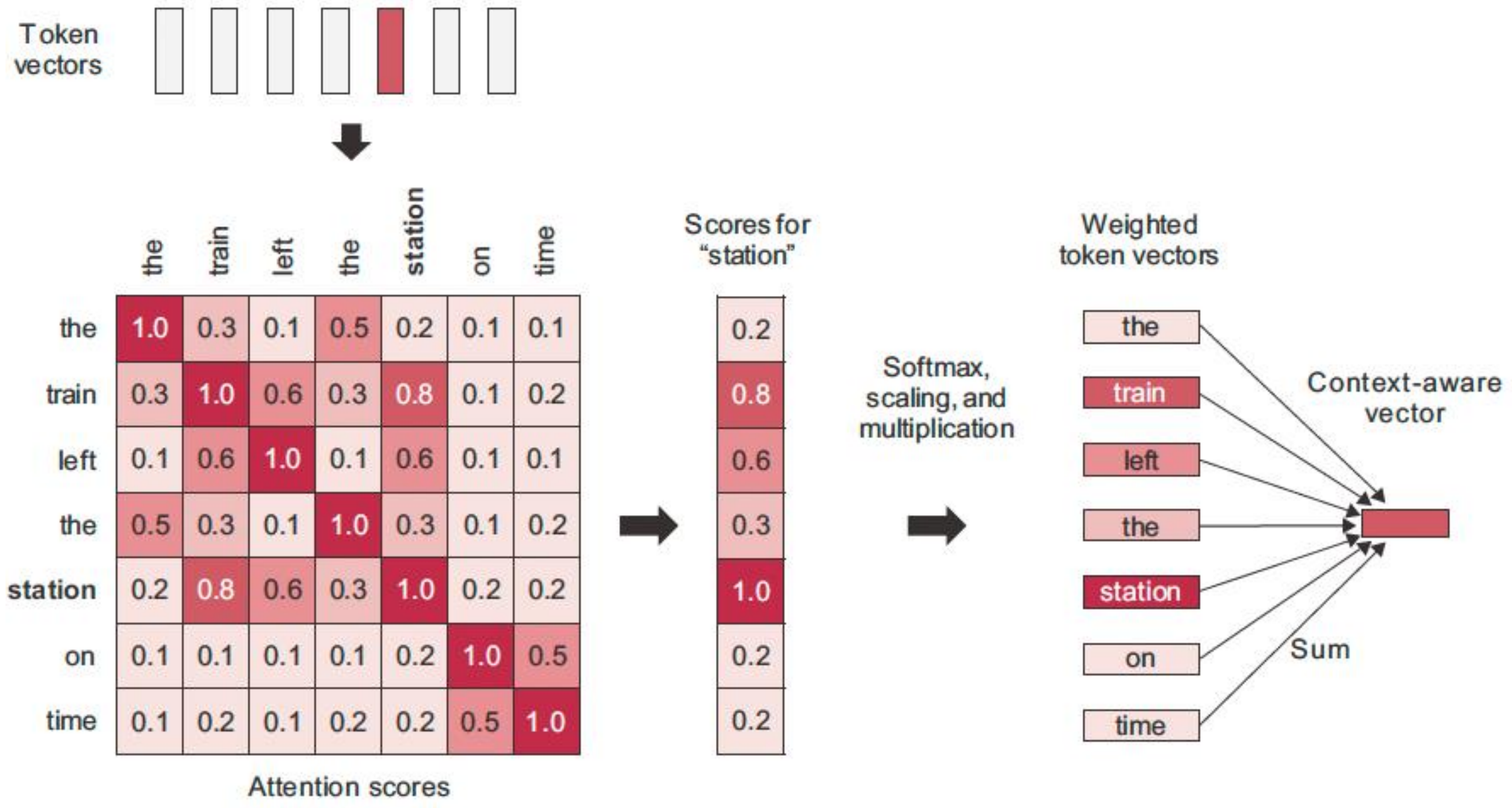
Self-attention produces ***context-aware*** token representations by modulating the representation of a *token* using the representations of *related tokens* in the same sequence.

$$\text{outputs} = \text{sum}(\textbf{input} \times \text{pairwise\_scores}(\textbf{input}, \textbf{input}))$$

Understanding Attention

Transformer-style Attention: Self-attention

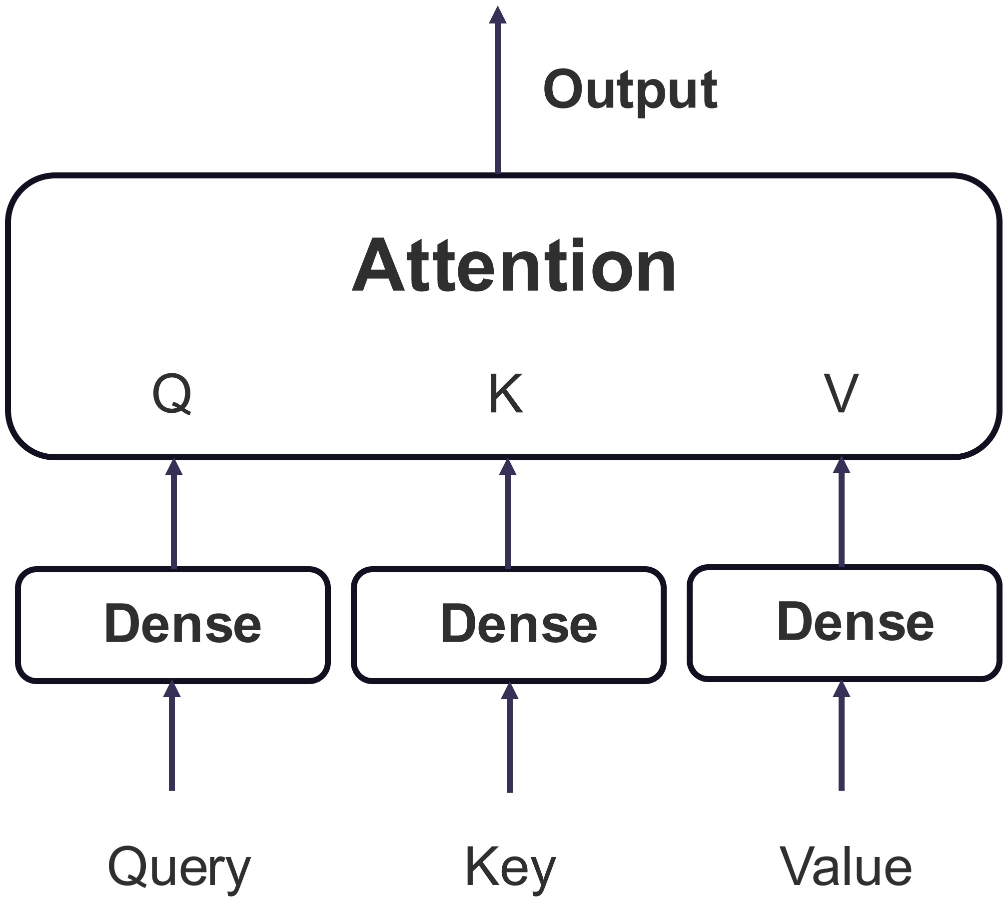
Input Sequence: The train left the station on time



Transformer Architecture

Dense Projections

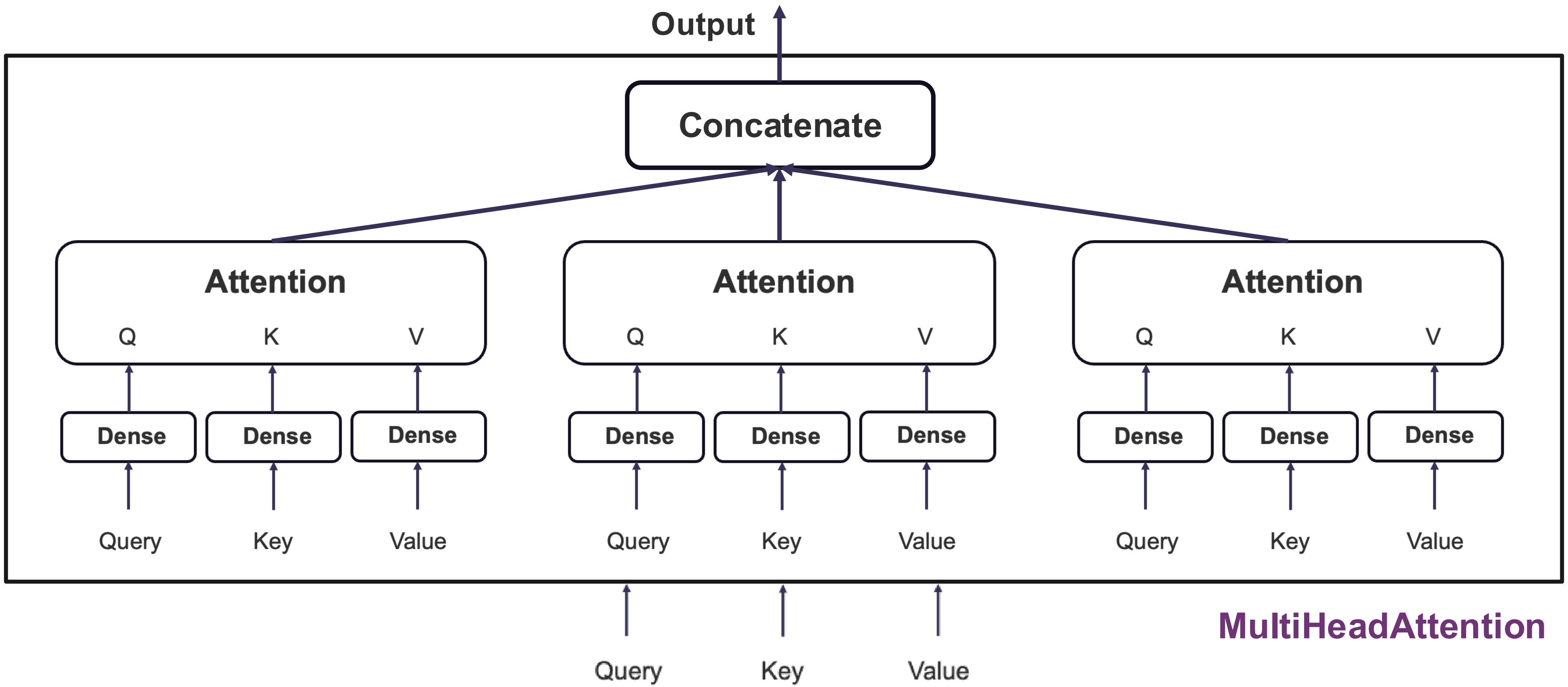
Before being processed via neural attention, the initial query, key, and value are sent into independent sets of ***dense projections***



Transformer Architecture

Multi-head Attention

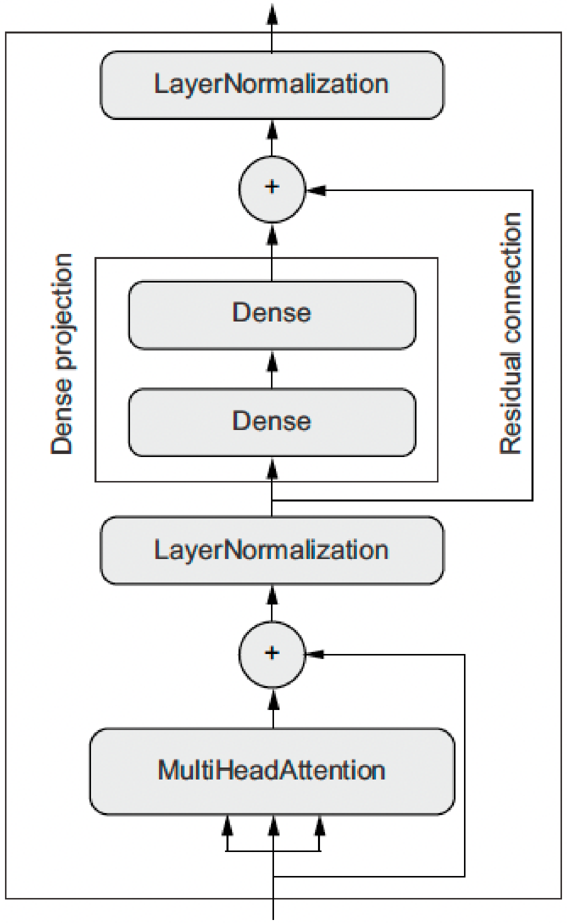
The output space of the **attention layer** gets factored into a set of different *subspaces* called "*head*".



MultiHeadAttention

Transformer Architecture

The Transformer Encoder



Residual Connections

Shortcut connections that allow for us to create sufficiently deep architectures.

Layer Normalization

Help gradients flow better during backpropagation

Transformer Architecture

The Transformer Encoder

Positional Encoding

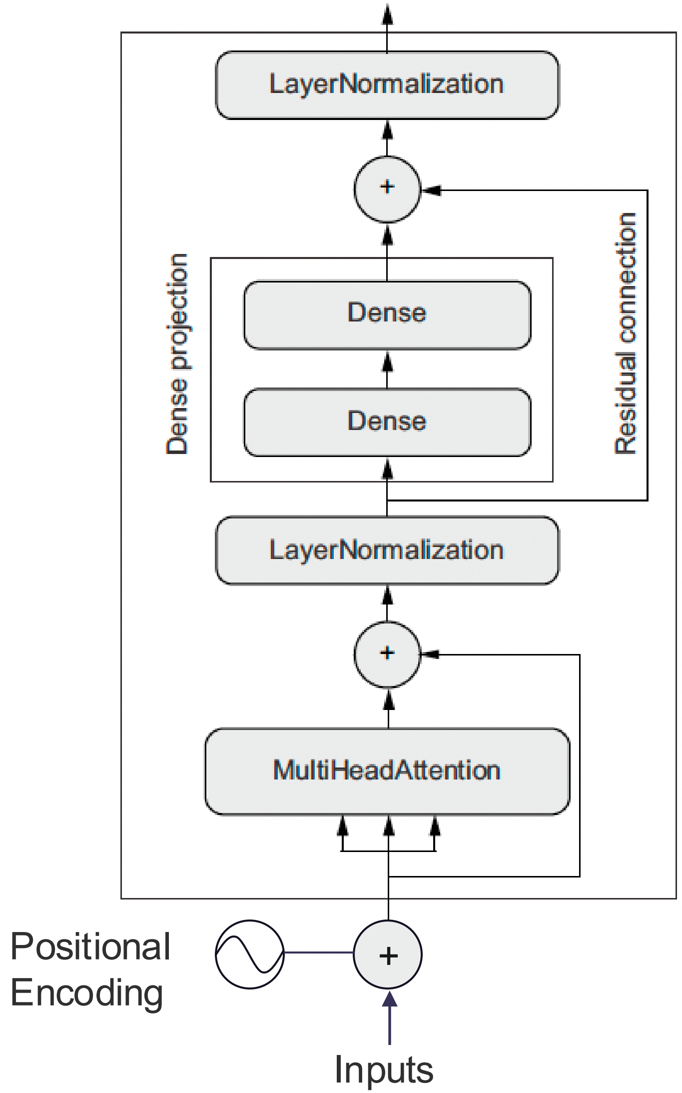
Gives the model access to word-order information

Sequence:	To	Infinity	and	Beyond	!
	0	1	2	3	4

Sinusoidal Positional Encoding

$$PE_{(pos,2i)} = \sin(pos/1000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = \cos(pos/1000^{2i/d_{\text{model}}})$$

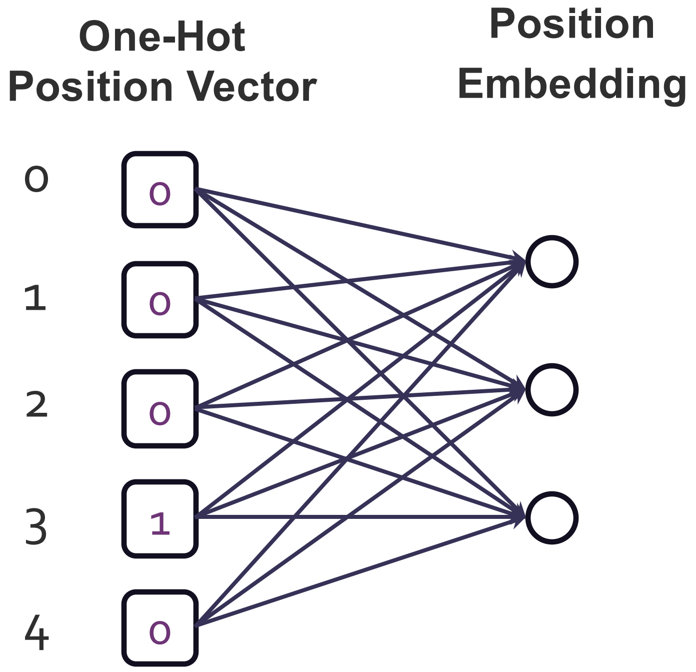


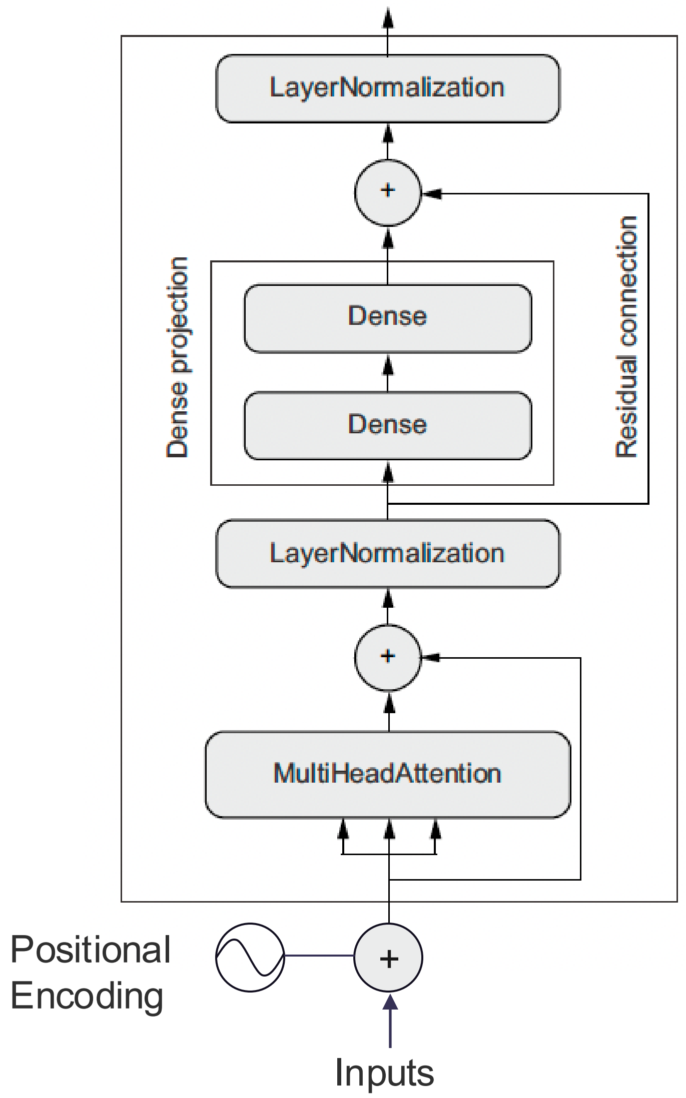
Transformer Architecture

The Transformer Encoder

Positional Embedding

An embedding that uses position indices as input





Sequence to Sequence Learning

Sequence to Sequence Models

A sequence-to-sequence model takes a sequence as input and translates it into a different sequence.

- *Machine translation* – convert a paragraph in a source language to a target language
- *Text summarization* – convert a long document to a shorter version
- *Question answering* – convert an input question into its answer
- *Chatbots* – convert a dialogue prompt into a reply to this prompt
- *Text generation* – convert a text prompt into a paragraph

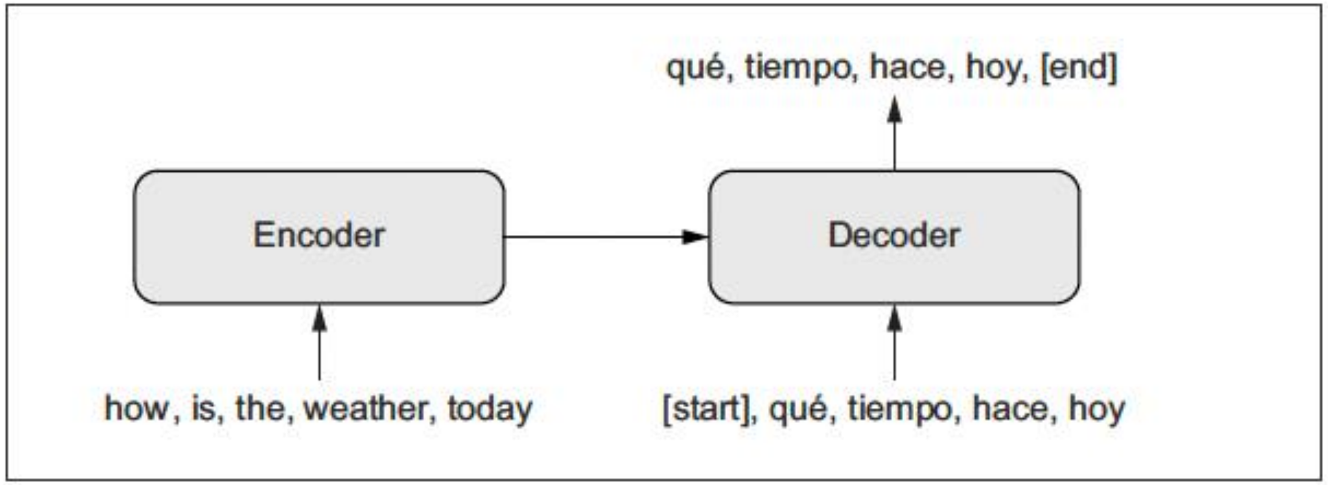
Sequence to Sequence Learning

Sequence to Sequence Models

Sequence to sequence models generally has two parts:

- An **encoder** model that turns the source sequence into an intermediate representation.
- A **decoder** which is trained to predict the next token by looking at both previous tokens and the encoded source sequence.

Training phase

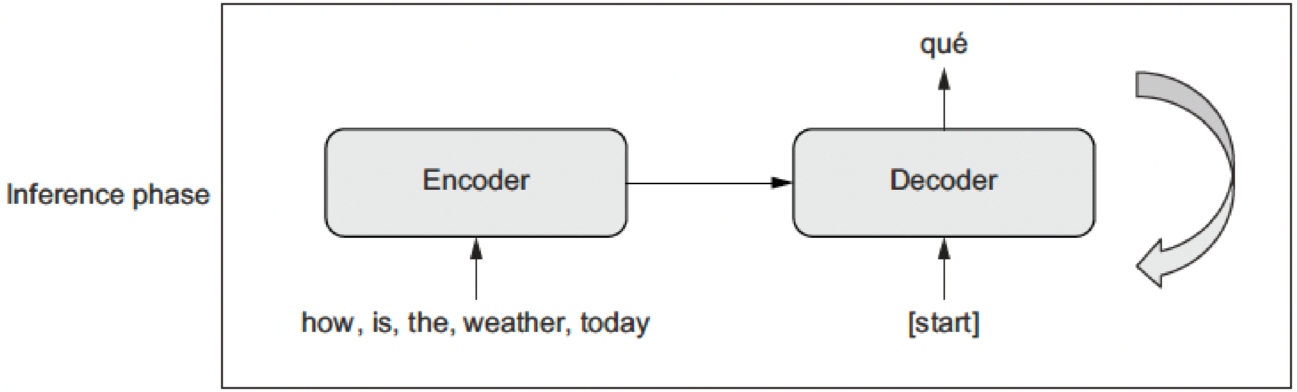


Sequence to Sequence Learning

Sequence to Sequence Models

At *inference* phase, we try to predict the target sequence from scratch.

- An encoded source sequence is obtained from the encoder
- The decoder starts by looking at the encoded source and with an initial **“seed”** token.
- The predicted sequence is fed back into the decoder, which generates the next token.
- Inference stops when a **stop** token is generated.

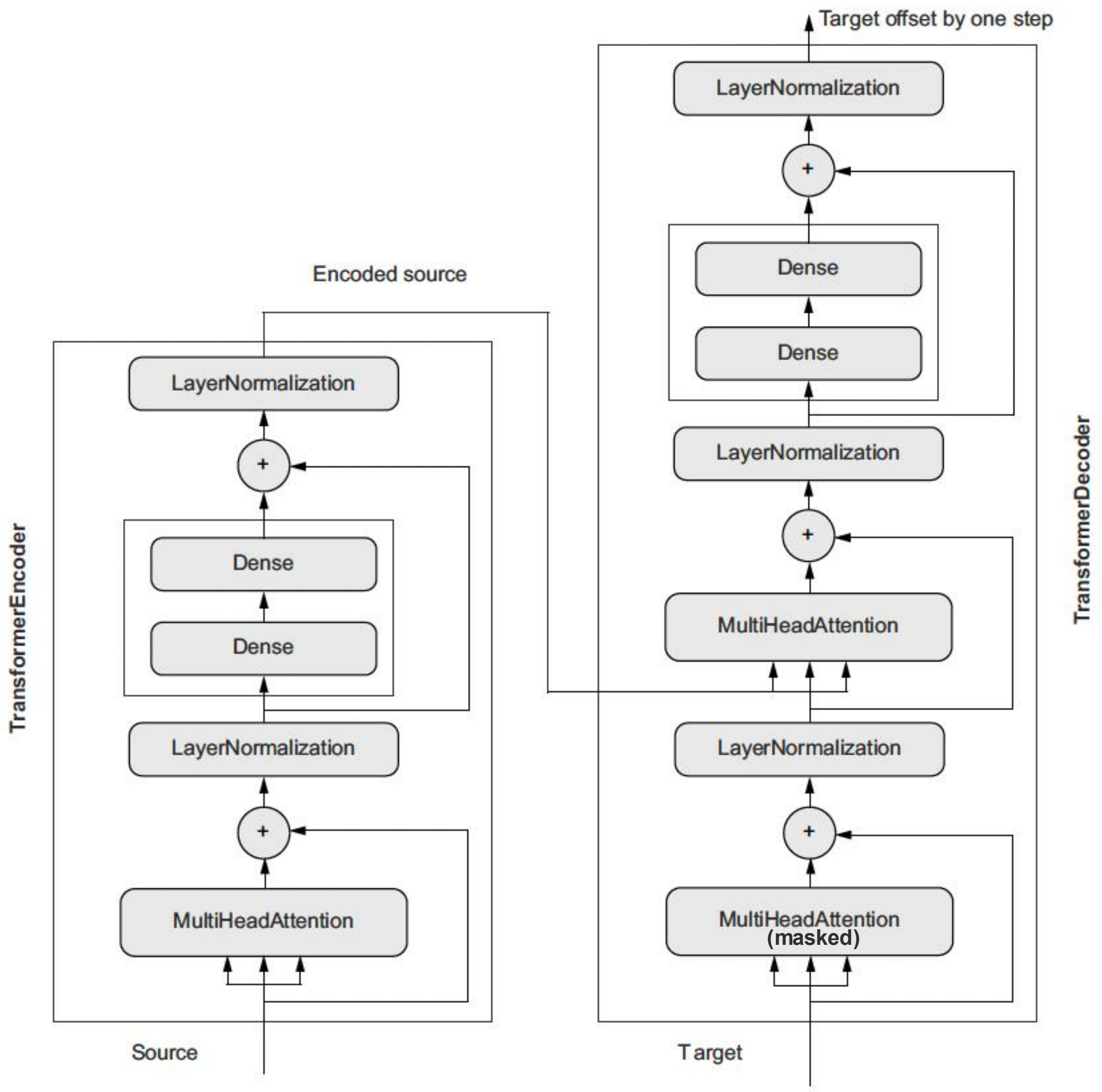


Transformer Architecture

The Transformer Encoder-Decoder

Causal Mask

A mask added to the decoder to prevent the model from paying attention to information from the future.



Transformers:
Key Concepts Review

Transformers: Key Concepts

1 Attention

- Query, Key, Values – QKV
- Self-attention

2 Architectural Patterns

- Dense Projections
- Multi-head Attention
- Layer Normalization
- Residual Connections

3 Additional Essential Tricks

- Positional Encoding
- Causal Padding

Thank you for your Attention!