
Resources

Article:
Google Brain, Attention is All You Need

Web:

- Transformer: A Novel Neural Network Architecture for Language Understanding
- Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)
- 3Blue1Brown:
  - Deep Learning Chapter 05: *What is a GPT?*
  - Deep Learning Chapter 06: *Attention in Transformers*
- The AI Hacker, *Illustrated Guide to Transformers*
- StatQuest, *Transformer Neural Networks*
- Campus X, *100 Days of Deep Learning Video 68 - 77*

Introduction

Recurrent Neural Networks

- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)

Working and Disadvantages

- Generate sequence of hidden states, $h_{t-1} \to h_t \to \ldots$, for input position t
- Cannot parallelize training, RNNs need to train *in order*. This becomes critical at longer sequence lengths, **memory constraints** limit *batching* across examples.

Improvements

- Conditional computation
- Factorization tricks

Conclusion

Although [Improvements] have been made to the efficiency of RNNs, the fundamental problems discussed, remain.

Attention Mechanisms

- Become an essential part of **sequence modelling**
- They allow the model to consider how **different parts of the input or output relate to each other**, *regardless of how far apart* they are. This is a big improvement over RNNs, which process things step-by-step.
- Used to be implemented in conjunction with a recurrent network

Transformers

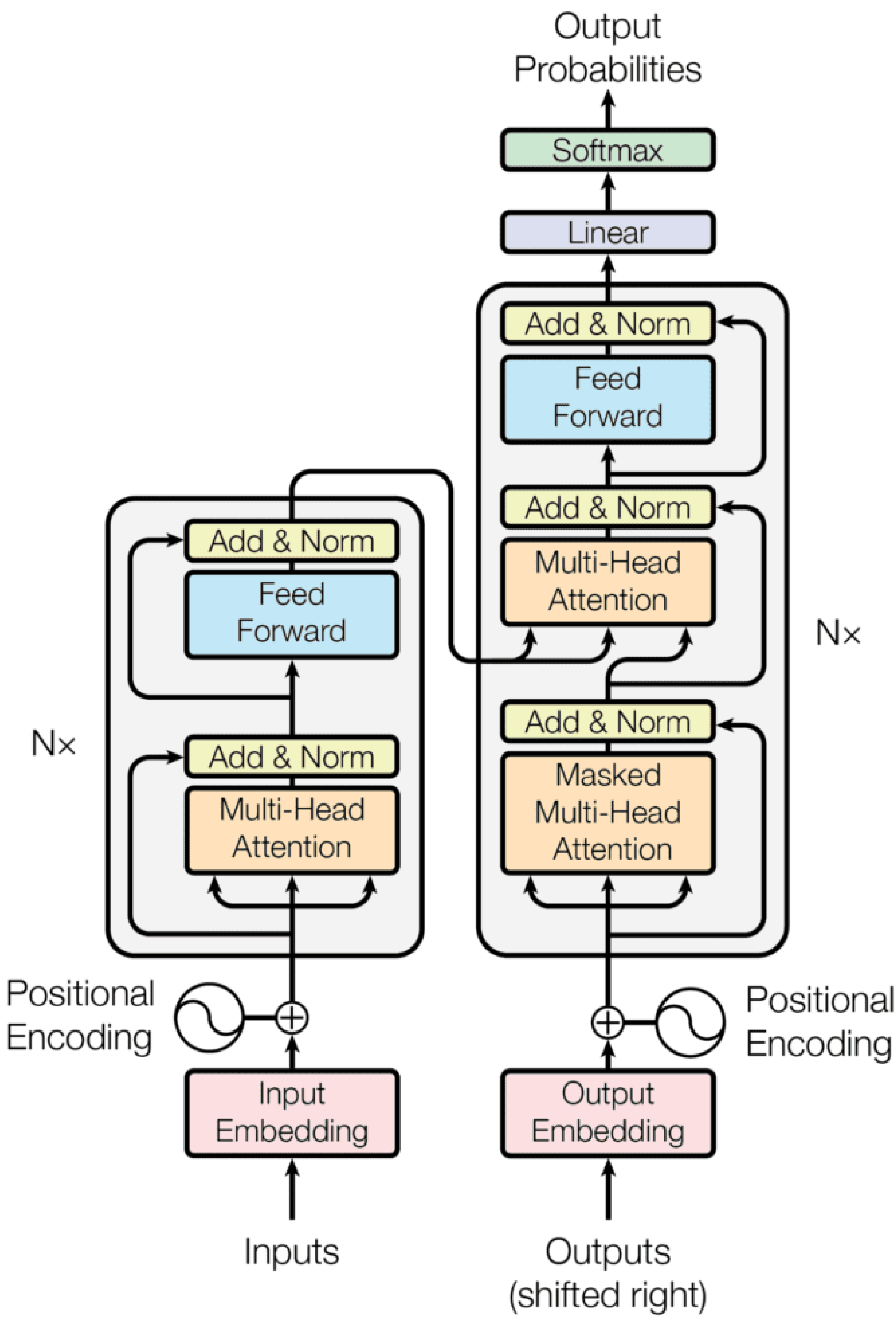
Attention is All You Need introduces the **Transformer** architecture, which

- Rejects recurrence
- Relies entirely on the **Attention** mechanism to figure out *how much the output depends on the input*
- Allows for parallelization

> The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs.

Attention-Is-All-You-Need_Google-Brain, page 2

Transformer Model Architecture


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Add & Norm
Masked
Multi-Head
Attention
Add & Norm
Feed
Forward
Nx
Add & Norm
Multi-Head
Attention
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

Encoder-Decoder Stacks

Transformers follow the *Encoder-Decoder* structure as described in the figure.

- The **encoder** maps the input sequence of symbol representations $\vec{x} = (x_1, x_2, \ldots, x_n)$ to a sequence of continuous representations $\vec{z} = (z_1, z_2, \ldots, z_n)$
- Given $\vec{z}$, the **decoder** generates an output sequence $\vec{y} = (y_1, y_2, \ldots, y_m)$ of symbols, *one element at a time*.
- At each time step the model is **auto-regressive -** consuming the **previously generated symbols** as **additional input** when *generating the next*.