텐서플로와 머신러닝으로 시작하는 자연어처리

Lecture 7
2019. 07. 13

Transformer

Transformer

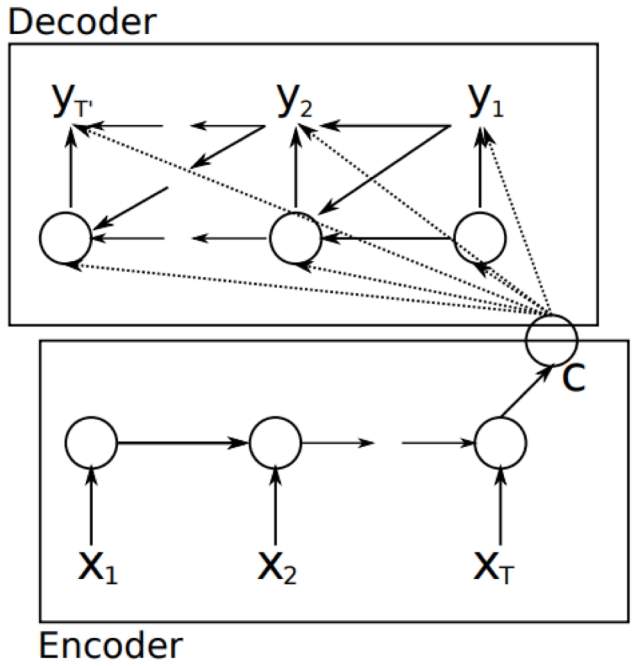
- History of NMT
- Attention is all you need

History of NMT

History of NMT

Phase 1: Enhancing the performance

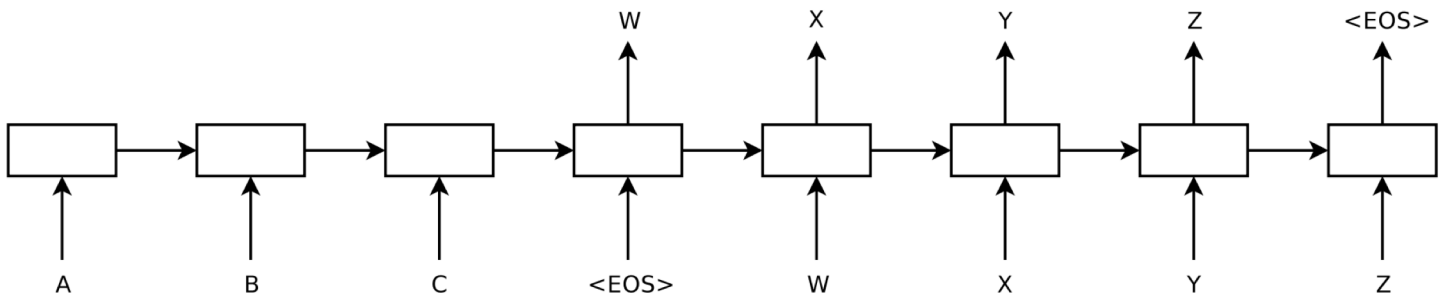
- **Neural machine translation by jointly learning to align and translate.**



History of NMT

Phase 1: Enhancing the performance

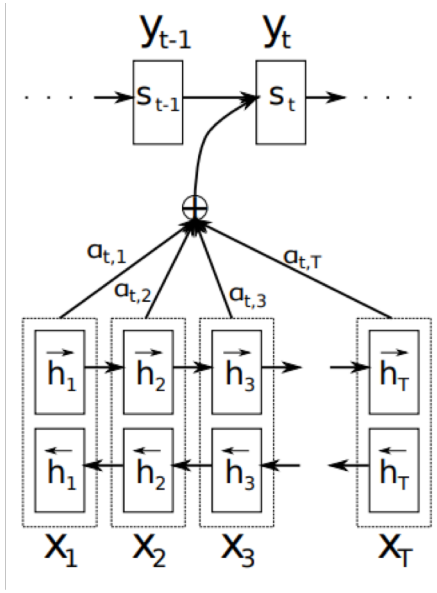
- Neural machine translation by jointly learning to align and translate.
- **Sequence to sequence learning with neural networks**



History of NMT

Phase 1: Enhancing the performance

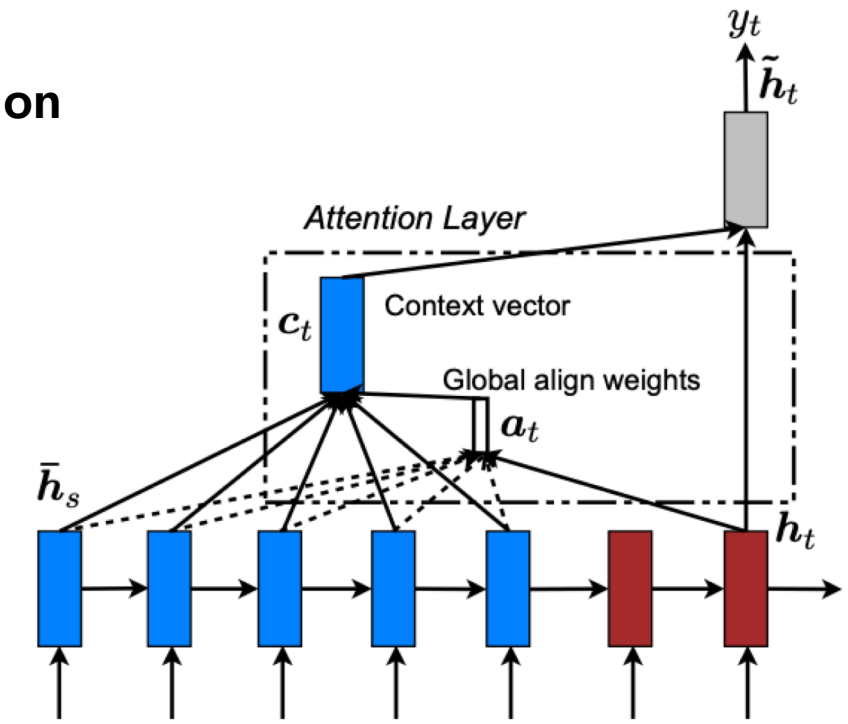
- Neural machine translation by jointly learning to align and translate.
- Sequence to sequence learning with neural networks
- **Neural machine translation by jointly learning to align and translate**



History of NMT

Phase 1: Enhancing the performance

- Neural machine translation by jointly learning to align and translate.
- Sequence to sequence learning with neural networks
- Neural machine translation by jointly learning to align and translate
- **Effective approaches to attention-based neural machine translation**



History of NMT

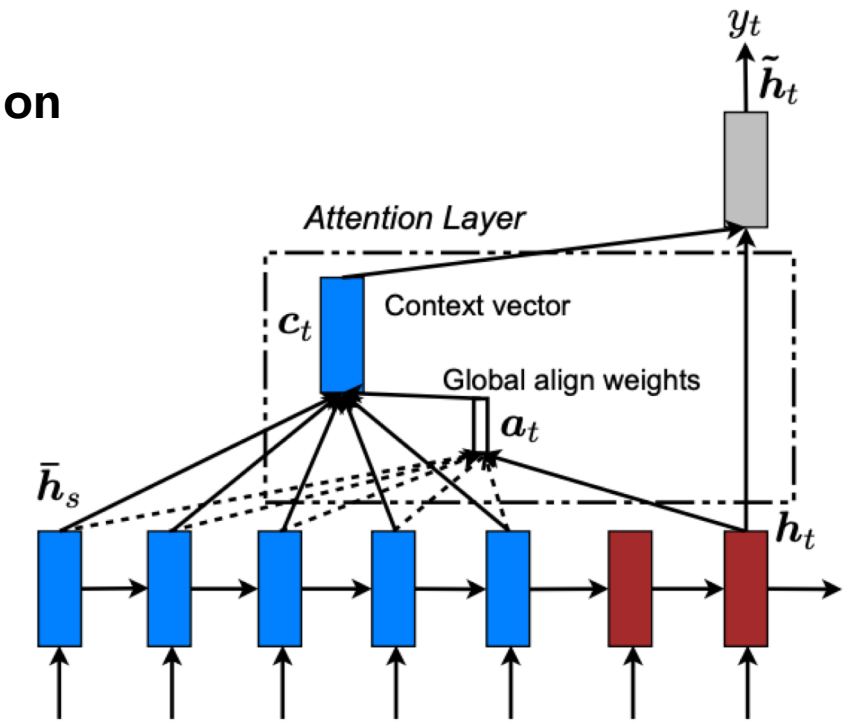
Phase 1: Enhancing the performance

- Neural machine translation by jointly learning to align and translate.
- Sequence to sequence learning with neural networks
- Neural machine translation by jointly learning to align and translate
- **Effective approaches to attention-based neural machine translation**

Global attention

$$\boldsymbol{a}_t(s) = \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_{\boldsymbol{a}} \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{W}_{\boldsymbol{a}}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s] & \textit{concat} \end{cases}$$



History of NMT

Phase 1: Enhancing the performance

- Neural machine translation by jointly learning to align and translate.
- Sequence to sequence learning with neural networks
- Neural machine translation by jointly learning to align and translate
- **Effective approaches to attention-based neural machine translation**

Global attention

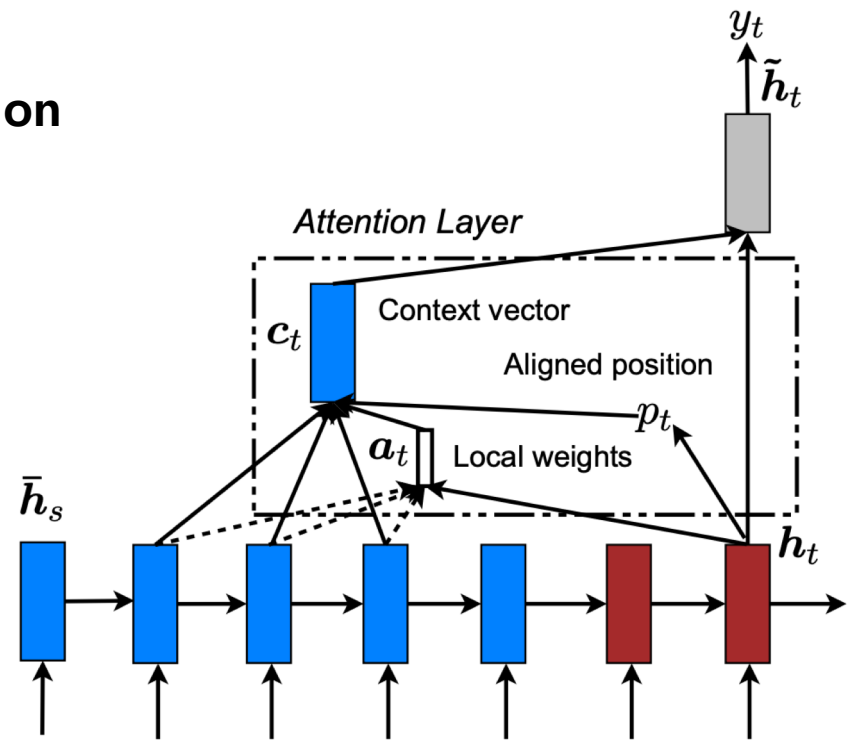
$$\boldsymbol{a}_t(s) = \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_{\boldsymbol{a}} \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{W}_{\boldsymbol{a}}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s] & \textit{concat} \end{cases}$$

Local attention

$$p_t = S \cdot \text{sigmoid}(\boldsymbol{v}_p^\top \tanh(\boldsymbol{W}_{\boldsymbol{p}} \boldsymbol{h}_t)),$$

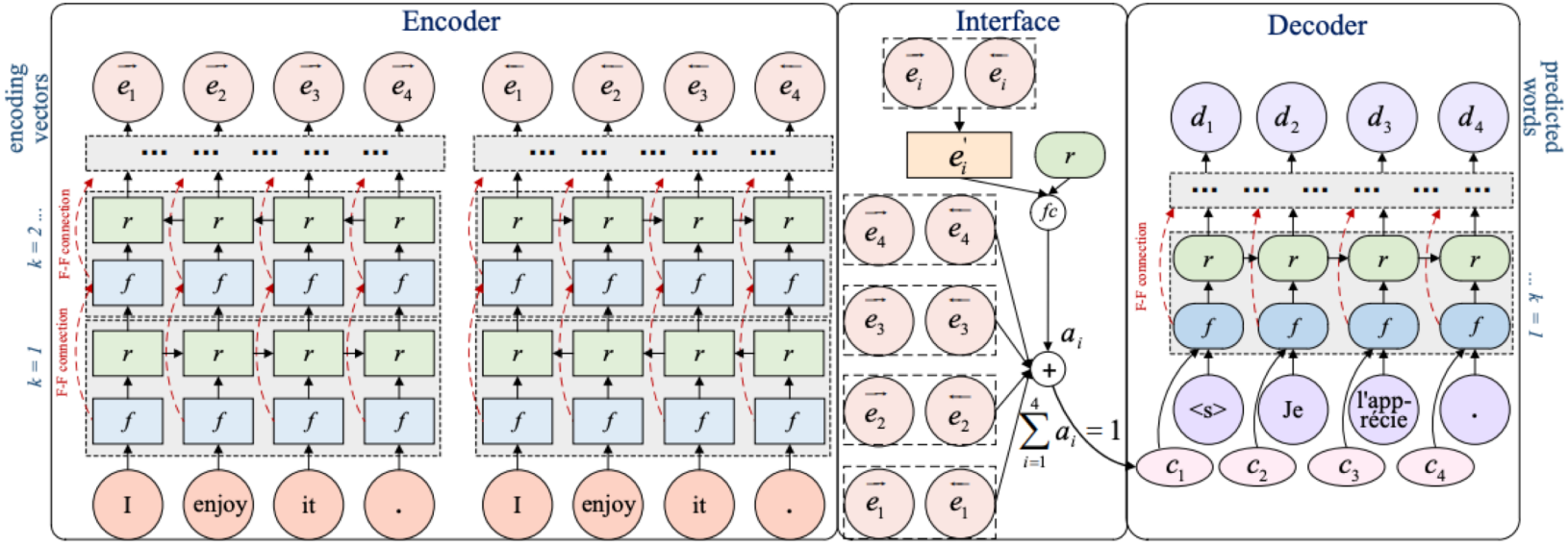
$$\boldsymbol{a}_t(s) = \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right)$$



History of NMT

Phase 1: Enhancing the performance

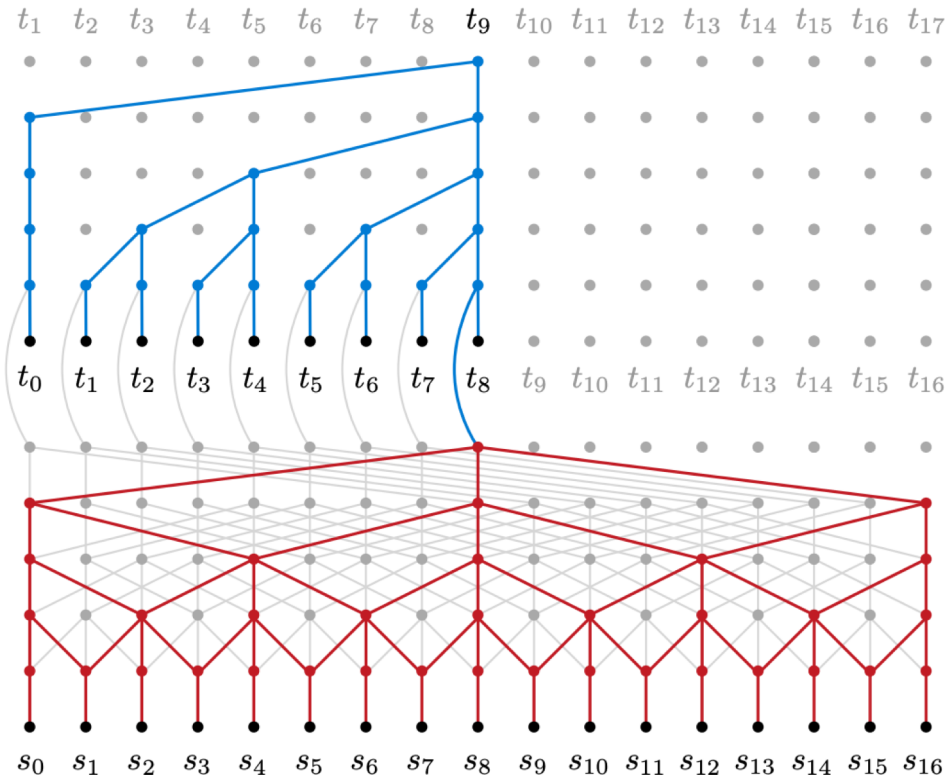
- Neural machine translation by jointly learning to align and translate.
- Sequence to sequence learning with neural networks
- Neural machine translation by jointly learning to align and translate
- Effective approaches to attention-based neural machine translation
- **Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation**



History of NMT

Phase 2: Parallel computation

- **Neural machine translation in linear time (ByteNet)**



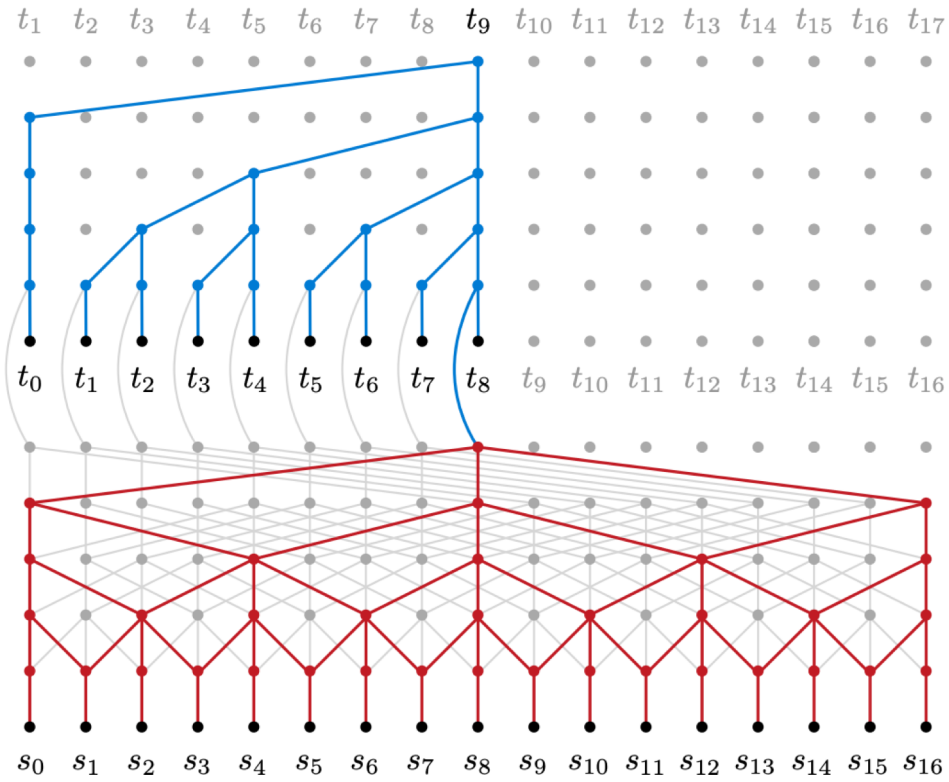
History of NMT

Phase 2: Parallel computation

- **Neural machine translation in linear time (ByteNet)**

Architecture

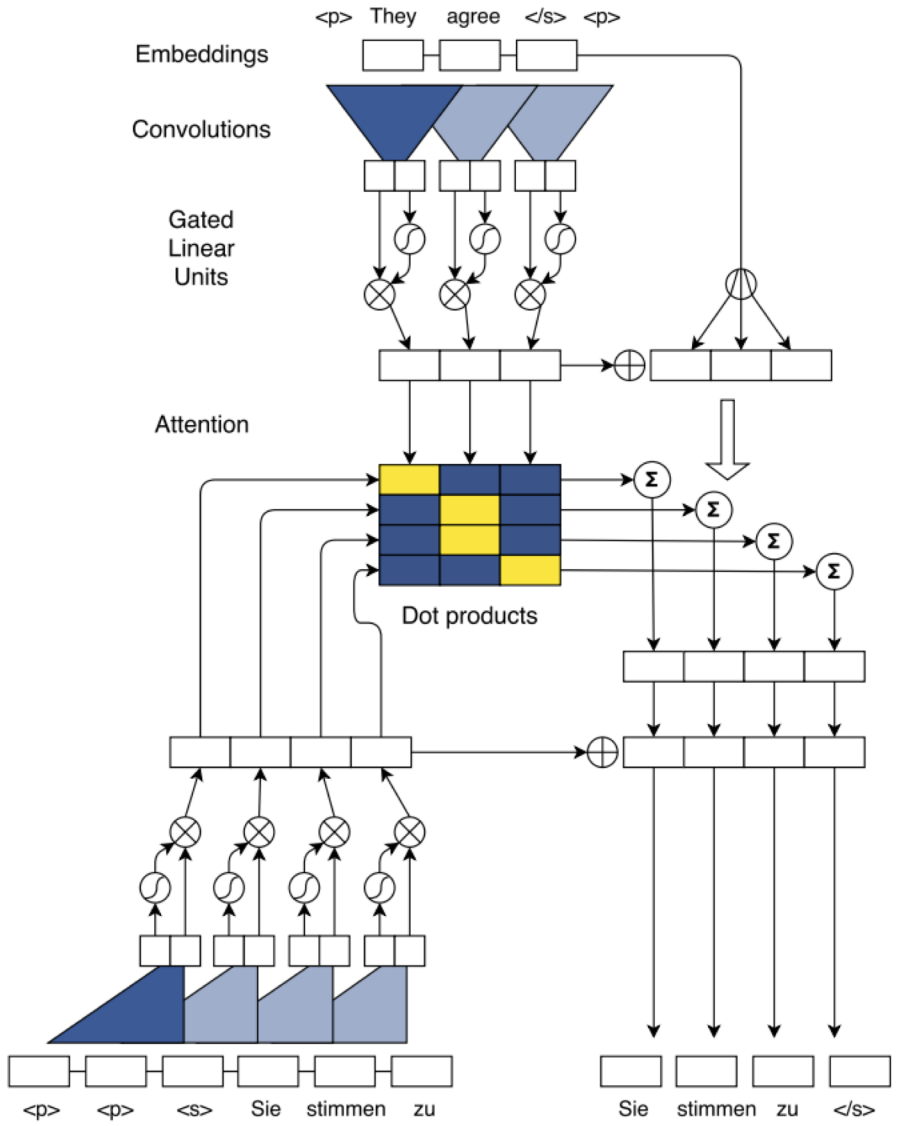
- 1D CNN (RNN도 가능)
- Encoder output => Decoder input
- Dynamic unfolding
- Masked one-dimensional convolutions
- Dilation
- Residual blocks



History of NMT

Phase 2: Parallel computation

- Neural machine translation in linear time (ByteNet)
- **Convolutional Sequence to Sequence Learning (Conv S2S)**



Architecture

- Positional embeddings
- Convolutional block structure
- Multi-step attention

History of NMT

- **Phase 1 (Performance)**
  - Neural machine translation by jointly learning to align and translate.
  - Sequence to sequence learning with neural networks
  - Neural machine translation by jointly learning to align and translate
  - Effective approaches to attention-based neural machine translation
  - Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation
- **Phase 2 (Parallel computation)**
  - Neural machine translation in linear time (ByteNet)
  - Convolutional Sequence to Sequence Learning (Conv S2S)

Attention is all you need

Attention Is All You Need

- Google Brain, 2017
- 기존의 RNN, CNN으로 feature를 뽑던 방식에서 attention만을 이용해 모델링함

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.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

*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. Lukasz 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.

Attention Is All You Need

Abstract

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.

Attention Is All You Need

Abstract

- 대부분의 sequence transduction 모델은 RNN, CNN 기반이다.

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.

Attention Is All You Need

Abstract

- 대부분의 sequence transduction 모델은 RNN, CNN 기반이다.
- Attention만으로 구성된 모델을 만들었다.

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.

Attention Is All You Need

Abstract

- 대부분의 sequence transduction 모델은 RNN, CNN 기반이다.
- Attention만으로 구성된 모델을 만들었다.
- WMT 2014 En-Ge 28.4, En-Fr 41.8의 성능으로 기존의 최고의 점수보다 더 높은 점수를 보였다.

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.

Attention Is All You Need

Abstract

- 대부분의 sequence transduction 모델은 RNN, CNN 기반이다.
- Attention만으로 구성된 모델을 만들었다.
- WMT 2014 En-Ge 28.4, En-Fr 41.8의 성능으로 기존의 최고의 점수보다 더 높은 점수를 보였다.
- 학습은 8개의 GPU로 3.5일동안 학습했다.

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.

Attention Is All You Need

Introduction

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

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

Introduction

- 최근 LSTM, GRU를 활용한 모델이 많이 사용되고 추가적인 성능 향상을 위한 다양한 방법들이 나왔다.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

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

Introduction

- 최근 LSTM, GRU를 활용한 모델이 많이 사용되고 추가적인 성능 향상을 위한 다양한 방법들이 나왔다.
- Recurrent기반의 모델은 각 step마다 계산이 되기 때문에 병렬화가 불가능하다.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

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

Introduction

- 최근 LSTM, GRU를 활용한 모델이 많이 사용되고 추가적인 성능 향상을 위한 다양한 방법들이 나왔다.
- Recurrent기반의 모델은 각 step마다 계산이 되기 때문에 병렬화가 불가능하다.
- 약간의 계산 효율화를 달성했지만, 본질적인 제약은 해결하지 못했다.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

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

Introduction

- 최근 LSTM, GRU를 활용한 모델이 많이 사용되고 추가적인 성능 향상을 위한 다양한 방법들이 나왔다.
- Recurrent기반의 모델은 각 step마다 계산이 되기 때문에 병렬화가 불가능하다.
- 약간의 계산 효율화를 달성했지만, 본질적인 제약은 해결하지 못했다.
- Attention mechanism이 long-term dependency를 해결했지만, 결국 대부분 recurrent 모델과 같이 사용되었다.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

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

Introduction

- 최근 LSTM, GRU를 활용한 모델이 많이 사용되고 추가적인 성능 향상을 위한 다양한 방법들이 나왔다.
- Recurrent기반의 모델은 각 step마다 계산이 되기 때문에 병렬화가 불가능하다.
- 약간의 계산 효율화를 달성했지만, 본질적인 제약은 해결하지 못했다.
- Attention mechanism이 long-term dependency를 해결했지만, 결국 대부분 recurrent 모델과 같이 사용되었다.
- 병렬화가 가능하고 attention mechanism을 적용한 Transformer 모델을 제안한다.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

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

Background

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

Attention Is All You Need

Background

- ByteNet, ConvS2S과 같이 sequential한 computation을 제거하는 것의 목적은 병렬 연산을 위해서다.

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

Attention Is All You Need

Background

- ByteNet, ConvS2S과 같이 sequential한 computation을 제거하는 것의 목적은 병렬 연산을 위해서다.
- 이러한 모델들은 연산량을 ConvS2S: linear, ByteNet: log 로 줄였지만, 전제 distant에 대한 dependency를 학습하기 어려워졌다.

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

Attention Is All You Need

Background

- ByteNet, ConvS2S과 같이 sequential한 computation을 제거하는 것의 목적은 병렬 연산을 위해서다.
- 이러한 모델들은 연산량을 ConvS2S: linear, ByteNet: log 로 줄였지만, 전제 distant에 대한 dependency를 학습하기 어려워졌다.
- Transformer에서는 상수시간으로 연산량을 줄이면서 효과적인 Multi-Head Attention을 사용한다.

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

Attention Is All You Need

Background

- ByteNet, ConvS2S과 같이 sequential한 computation을 제거하는 것의 목적은 병렬 연산을 위해서다.
- 이러한 모델들은 연산량을 ConvS2S: linear, ByteNet: log 로 줄였지만, 전제 distant에 대한 dependency를 학습하기 어려워졌다.
- Transformer에서는 상수시간으로 연산량을 줄이면서 효과적인 Multi-Head Attention을 사용한다.
- 단일 문장에 적용하는 self-attention은 다양한 분야에서 사용되었다.

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

Attention Is All You Need

Background

- ByteNet, ConvS2S과 같이 sequential한 computation을 제거하는 것의 목적은 병렬 연산을 위해서다.
- 이러한 모델들은 연산량을 ConvS2S: linear, ByteNet: log 로 줄였지만, 전제 distant에 대한 dependency를 학습하기 어려워졌다.
- Transformer에서는 상수시간으로 연산량을 줄이면서 효과적인 Multi-Head Attention을 사용한다.
- 단일 문장에 적용하는 self-attention은 다양한 분야에서 사용되었다.
- Transformer는 최초의 RNN, CNN등을 사용하지 않고 self-attention만을 사용해서 input, output을 계산한 모델이다.

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

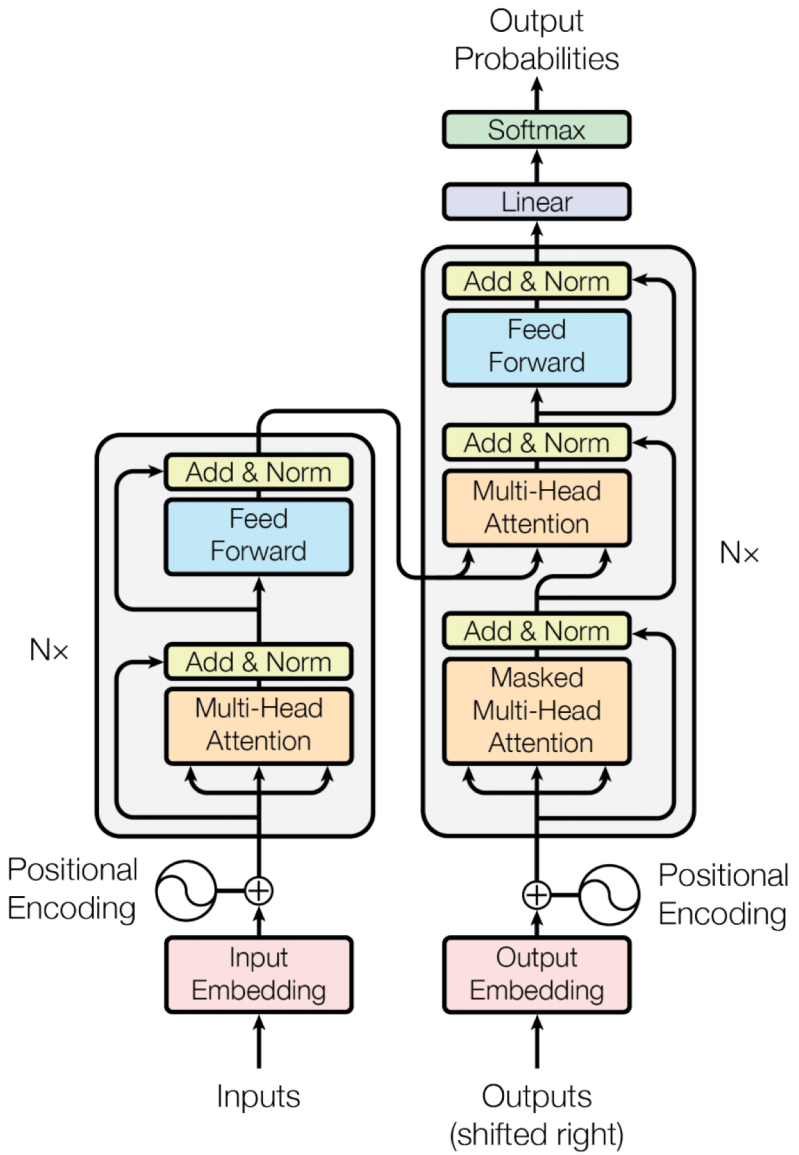
Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

Attention Is All You Need

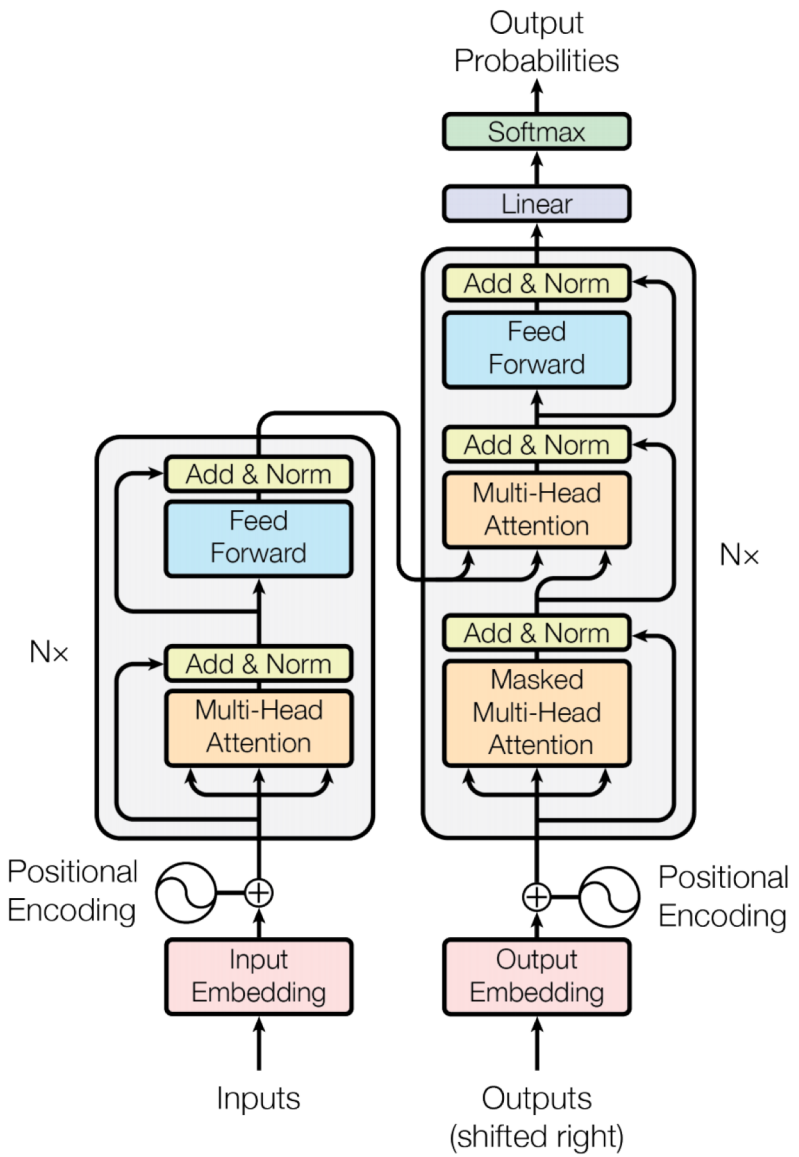
Model Architecture



Attention Is All You Need

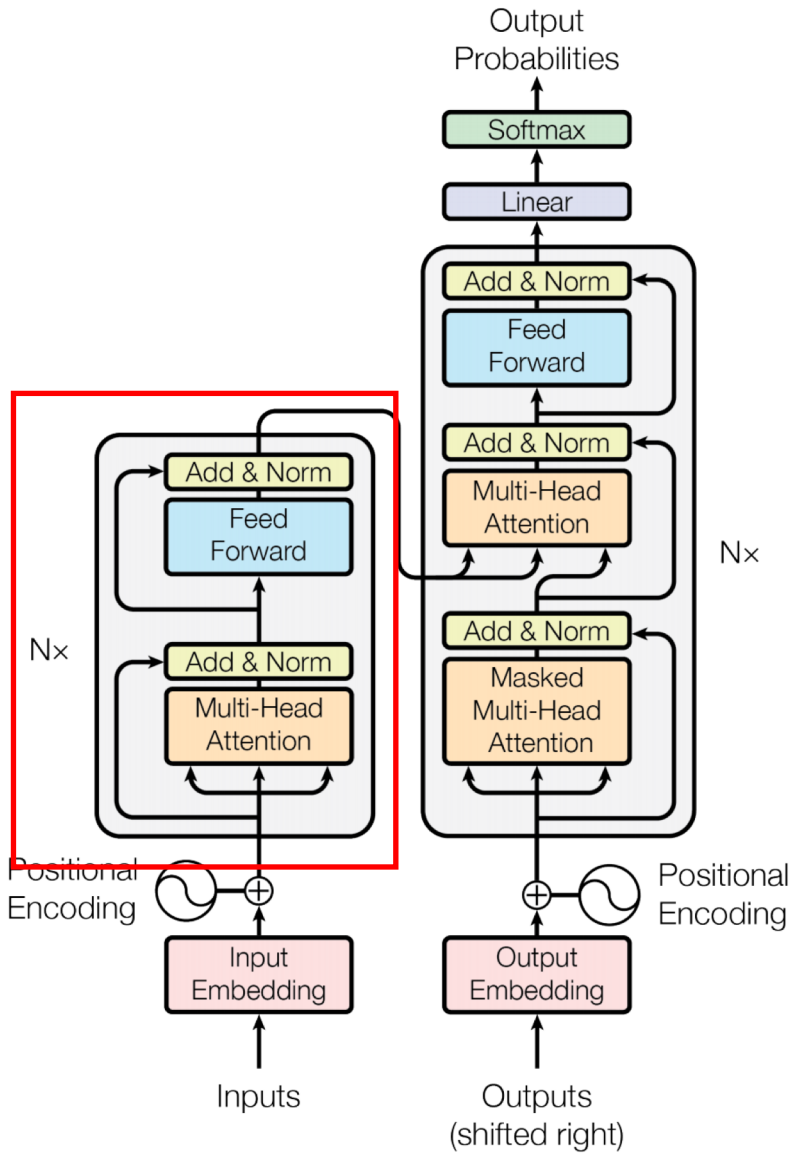
Model Architecture

- Encoder and Decoder Stacks
- Attention
  - Scaled-dot Attention
  - Multi-Head Attention
  - Applications of Attention in our Model
- Position-wise Feed-Forward Networks
- Embeddings and Softmax
- Position Encoding



Attention Is All You Need

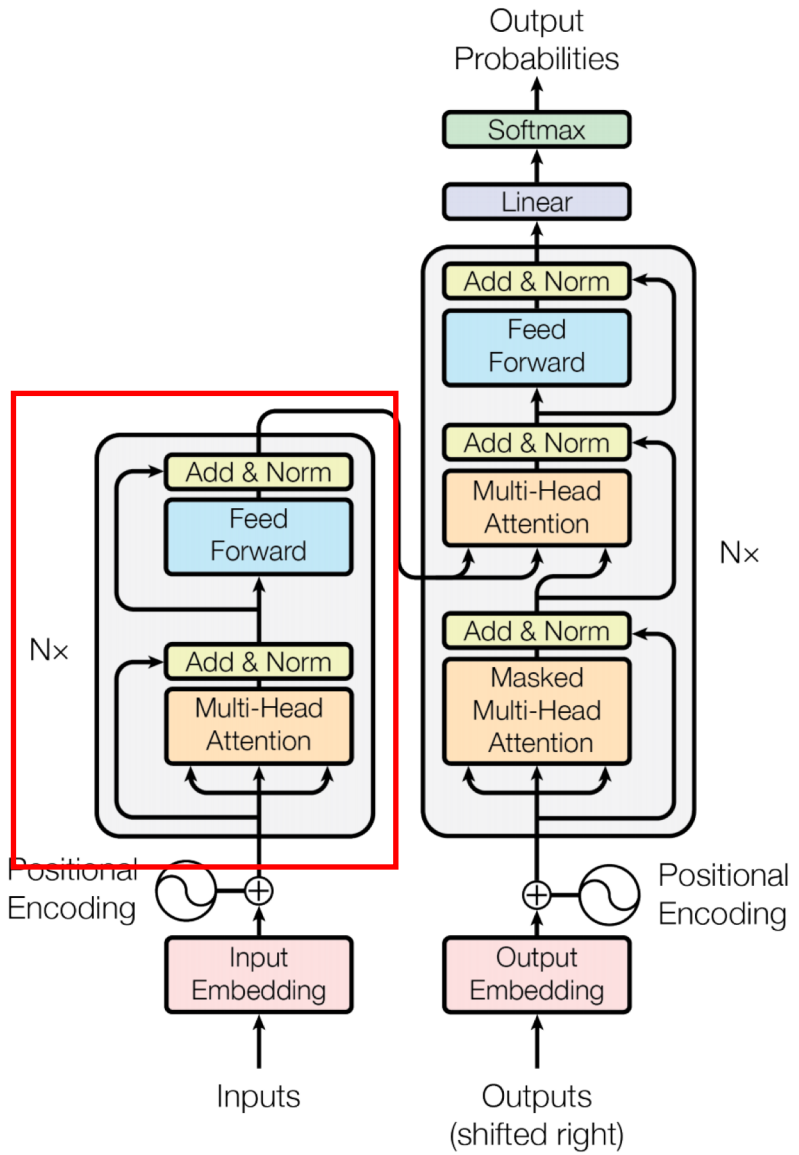
Model Architecture – Encoder Decoder Stacks



Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

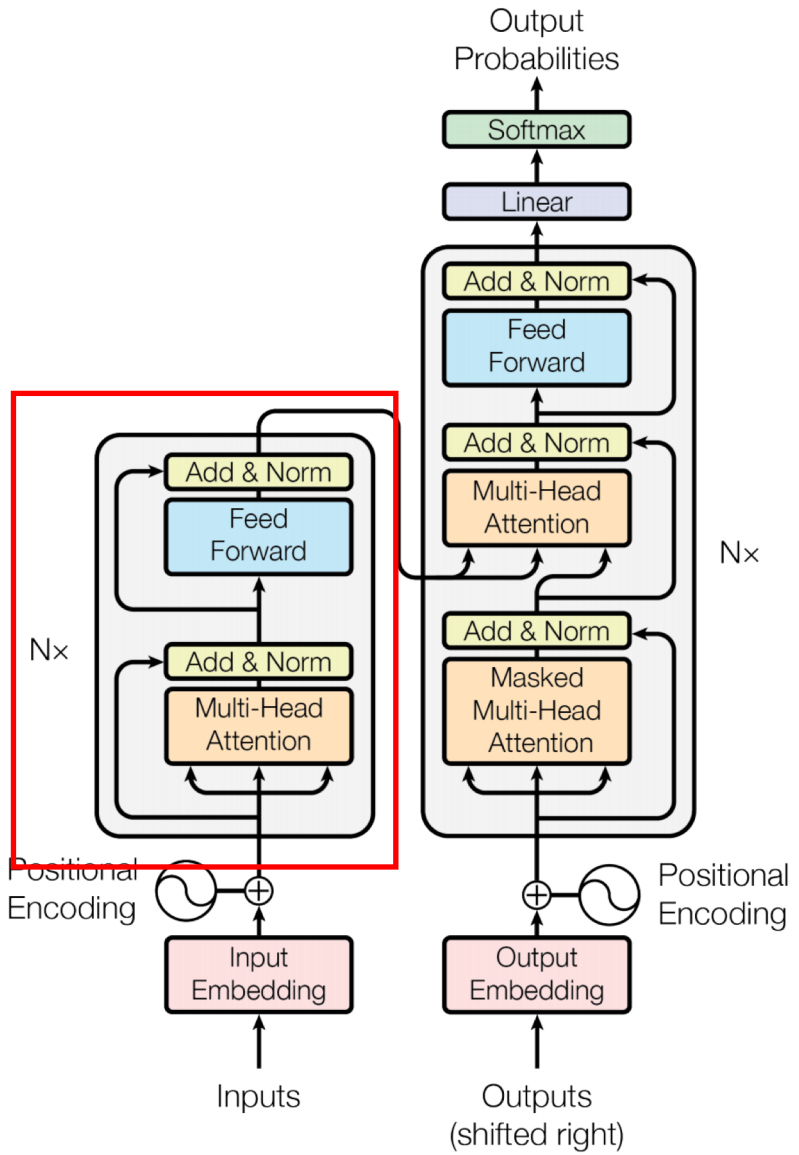


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.

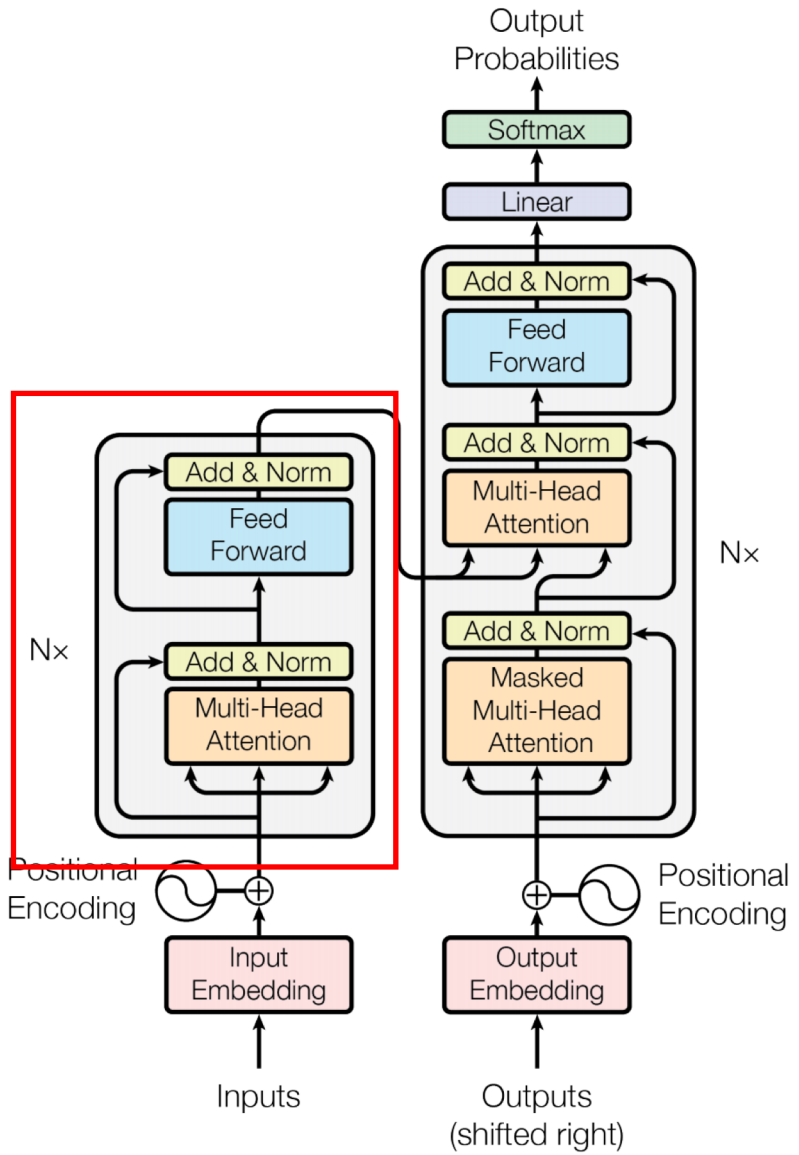


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- 각 layer는 두개의 sub-layer로 구성되어 있다.

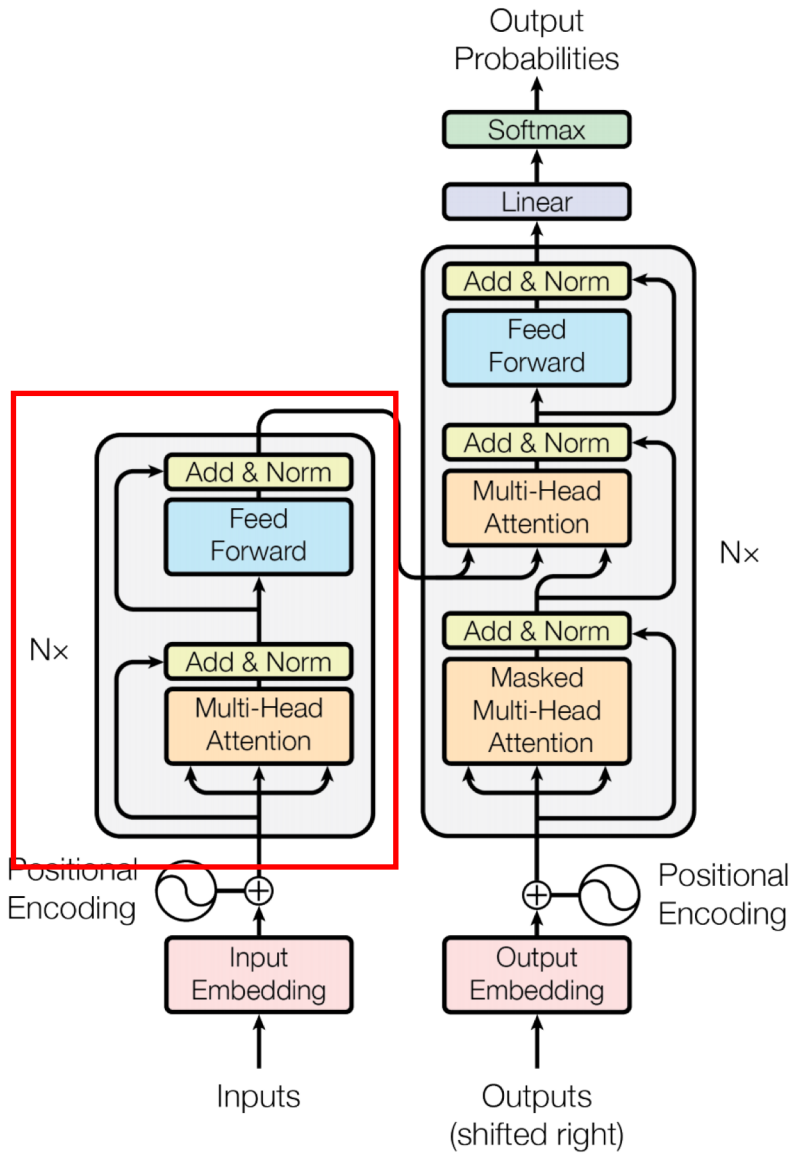


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- 각 layer는 두개의 sub-layer로 구성되어 있다.
  - Multi-head self-attention layer

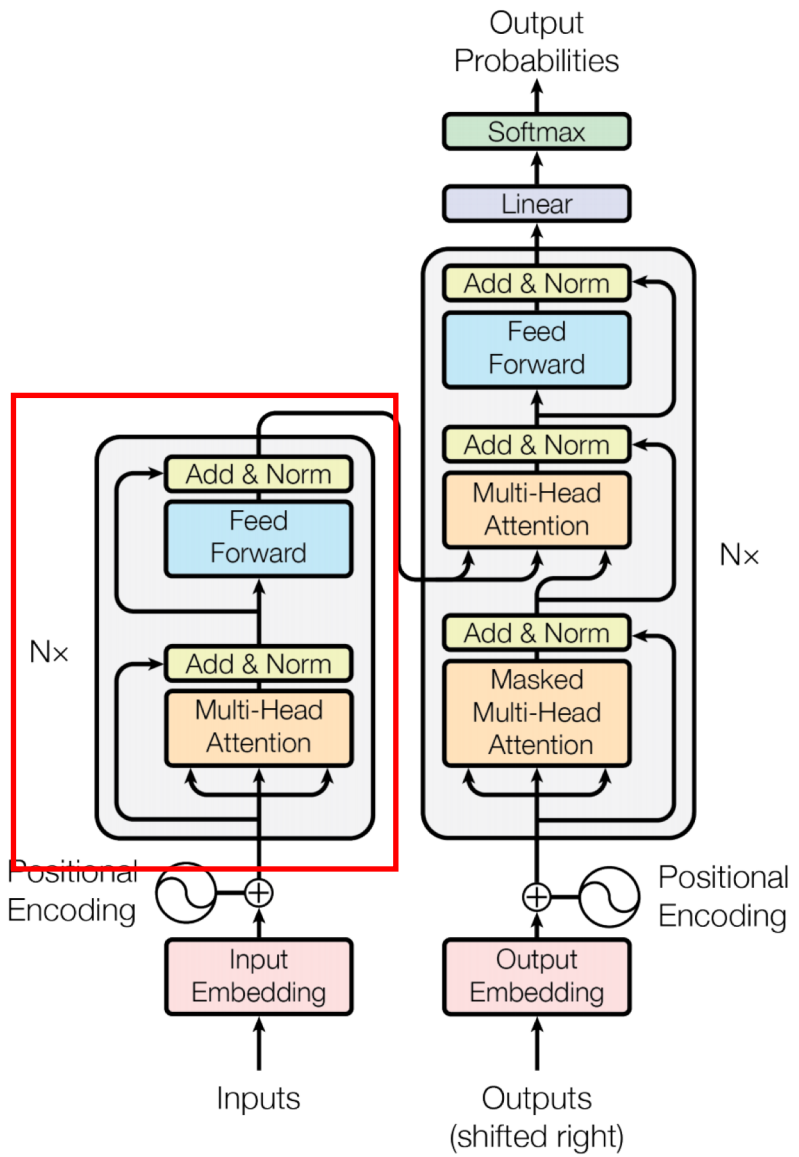


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- 각 layer는 두개의 sub-layer로 구성되어 있다.
  - Multi-head self-attention layer
  - Feed-forward network layer

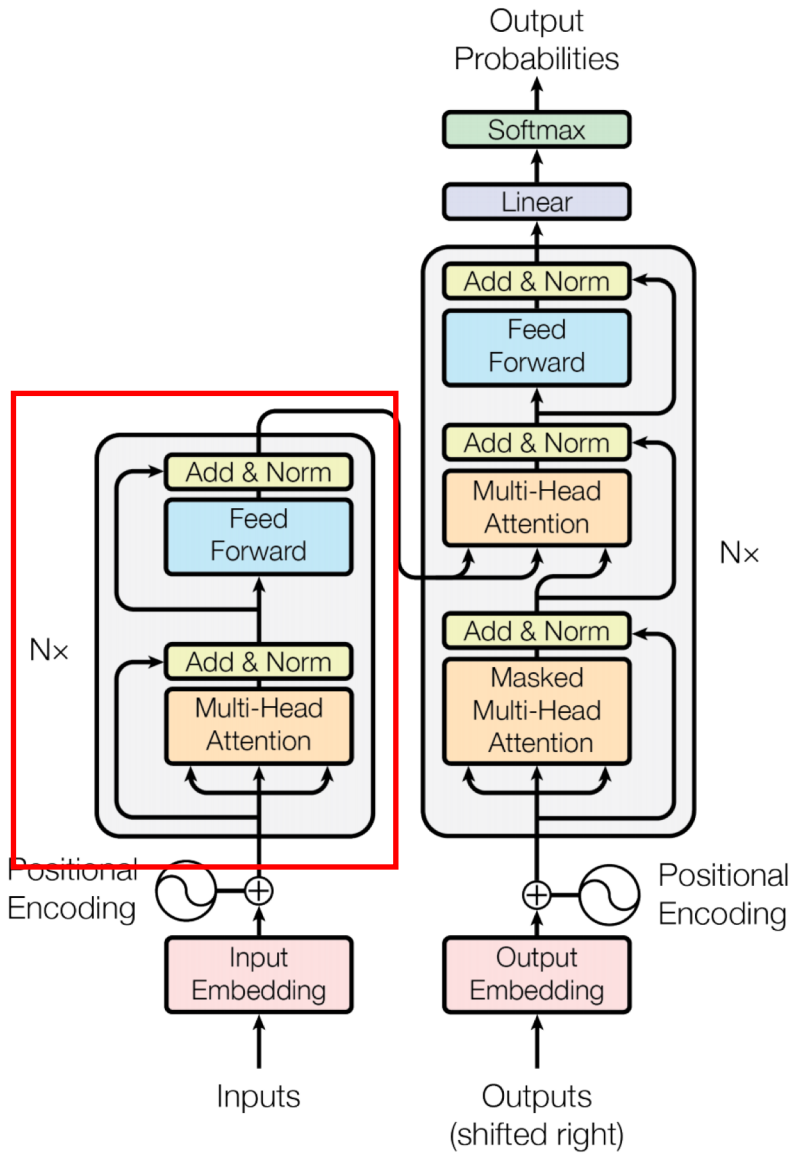


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- 각 layer는 두개의 sub-layer로 구성되어 있다.
  - Multi-head self-attention layer
  - Feed-forward network layer
- 두 개의 sub-layer에 각각 residual connection 과 layer normalization 을 적용했다.

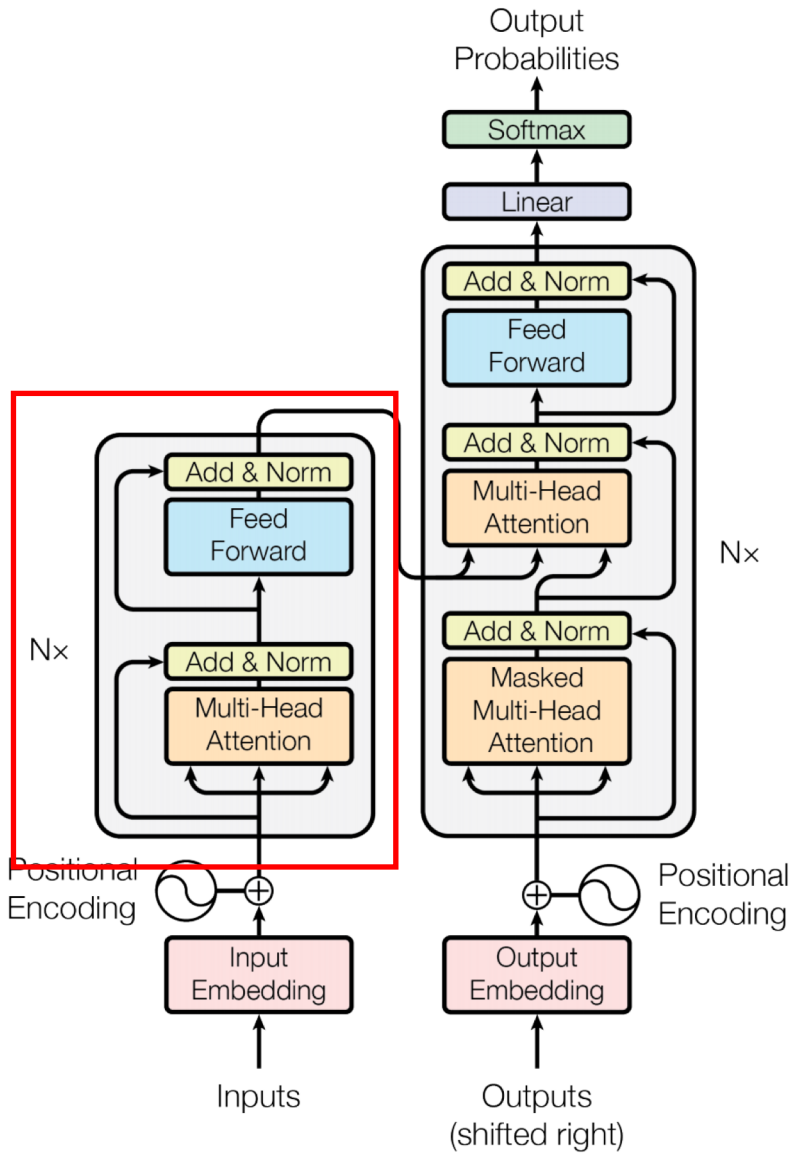


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- 각 layer는 두개의 sub-layer로 구성되어 있다.
  - Multi-head self-attention layer
  - Feed-forward network layer
- 두 개의 sub-layer에 각각 residual connection 과 layer normalization 을 적용했다.
  - $LayerNorm(x + Sublayer(x))$

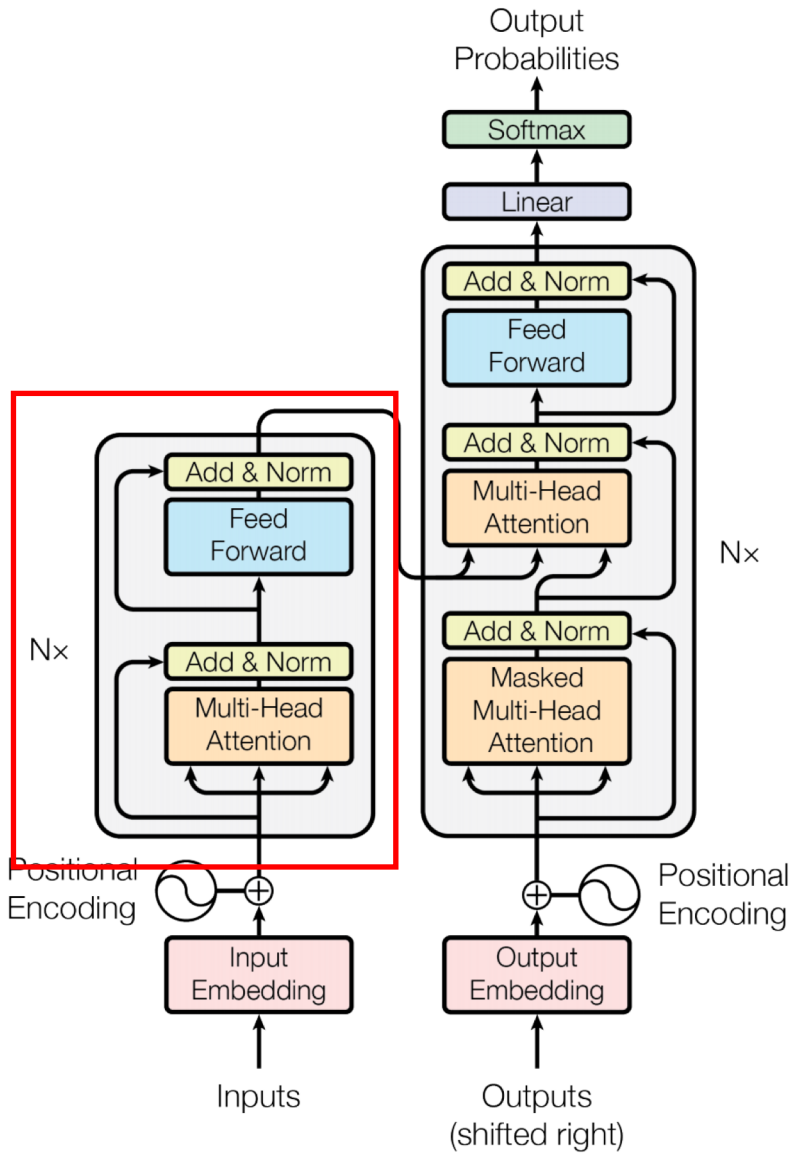


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Encoder

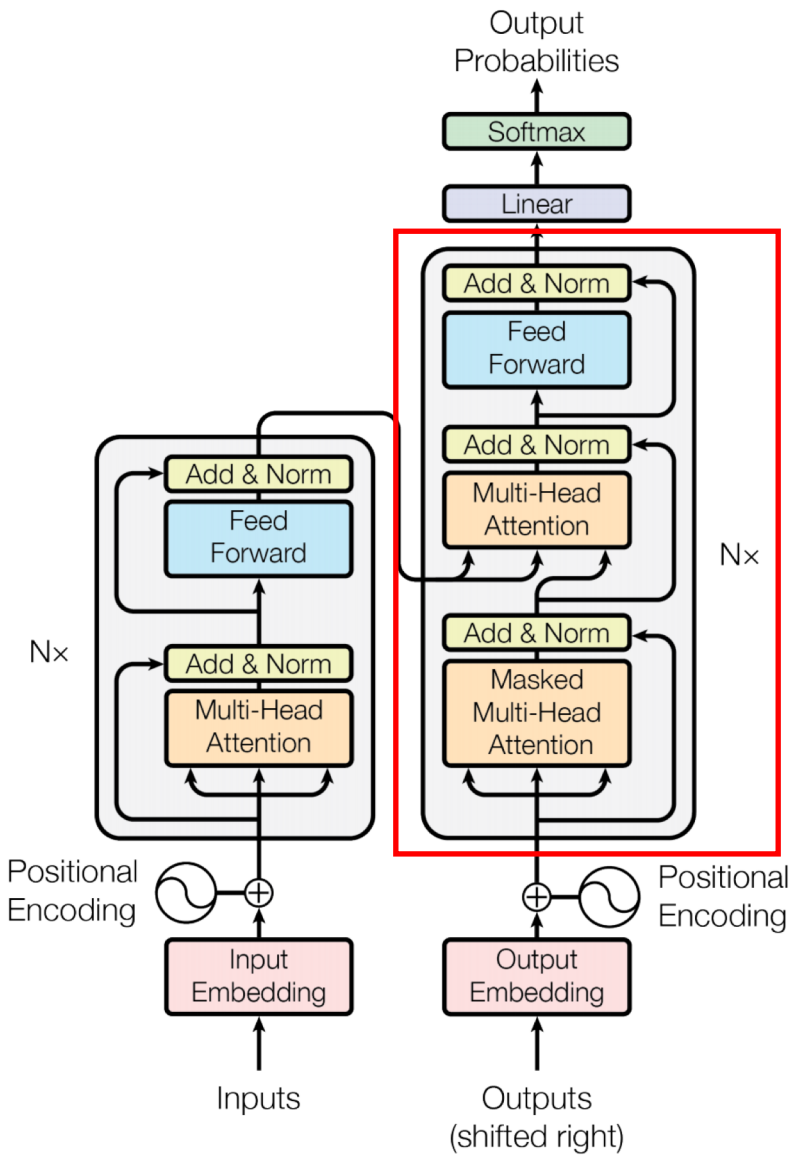
- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- 각 layer는 두개의 sub-layer로 구성되어 있다.
  - Multi-head self-attention layer
  - Feed-forward network layer
- 두 개의 sub-layer에 각각 residual connection 과 layer normalization 을 적용했다.
  - $LayerNorm(x + Sublayer(x))$
  - Residual connection을 적용하기 위해 embedding layer를 포함한 모든 layer의 output의 dimension을 동일하게 했다. $(d_{model} = 512)$



Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Decoder

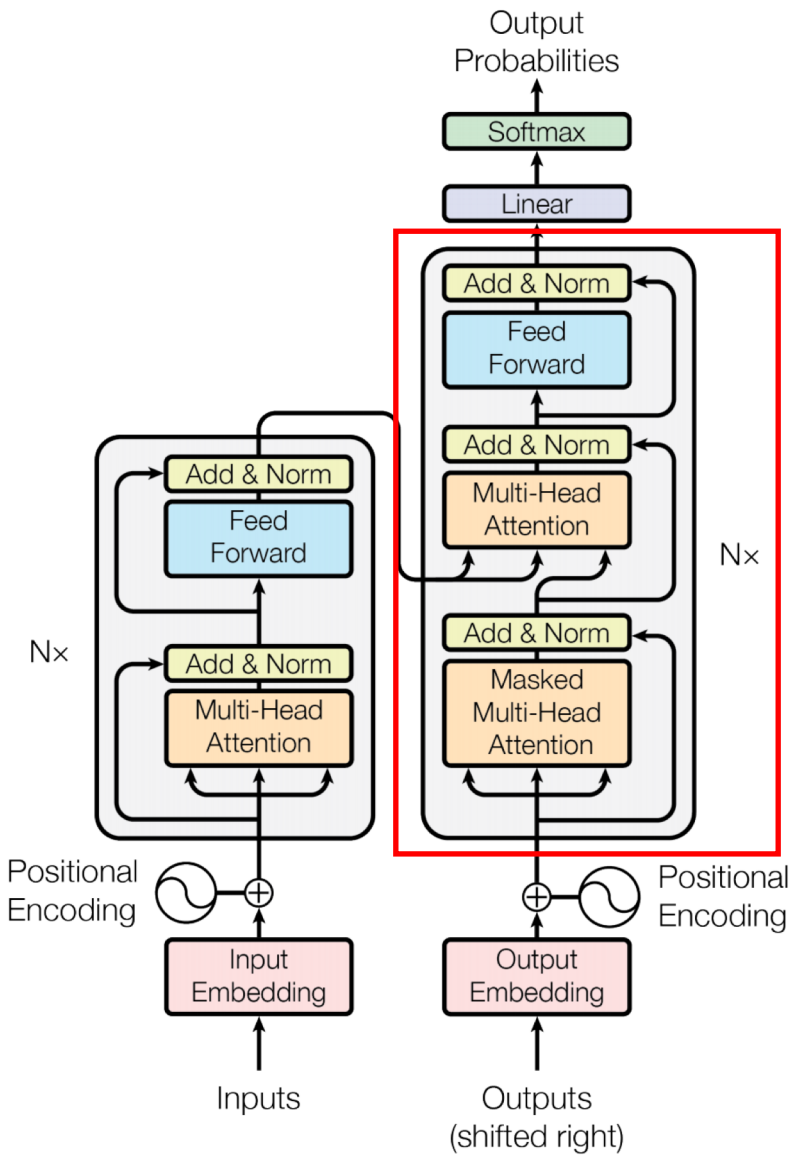


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Decoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.

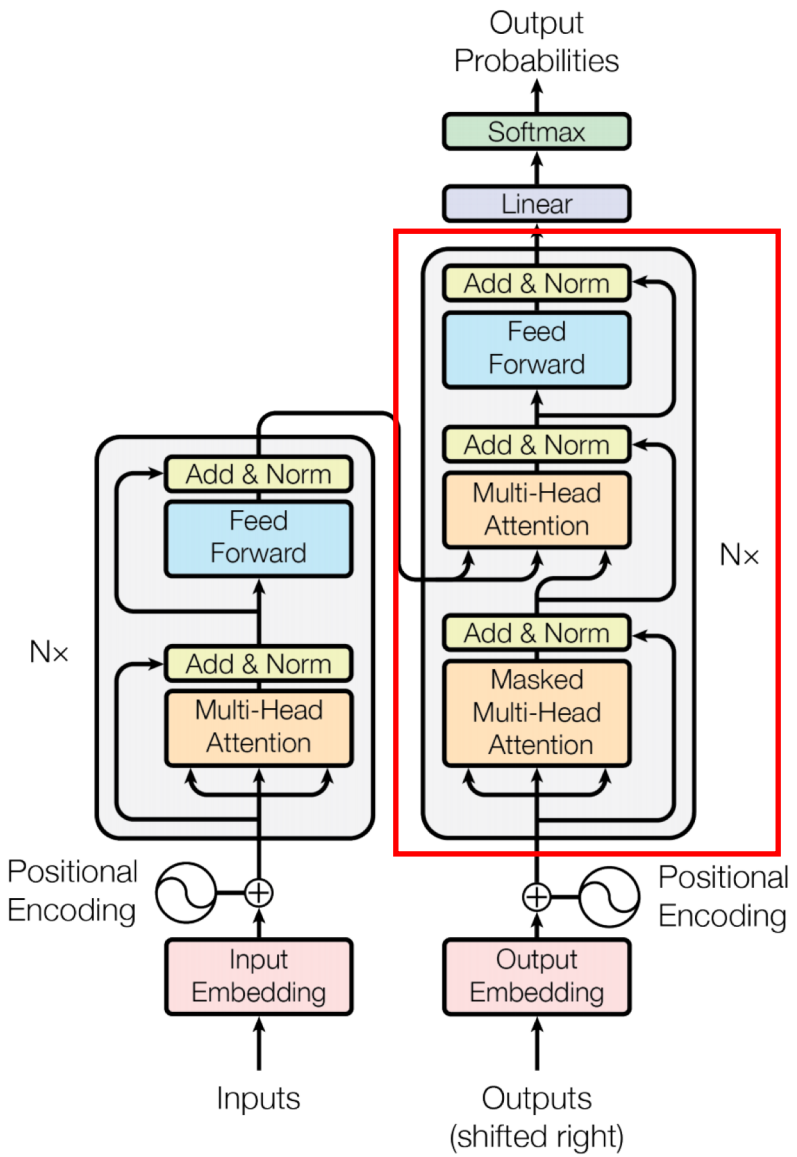


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Decoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- Encoder 와 동일한 2개의 sub-layer에 추가로 encoder output과 attention을 구하는 layer가 추가되었다.

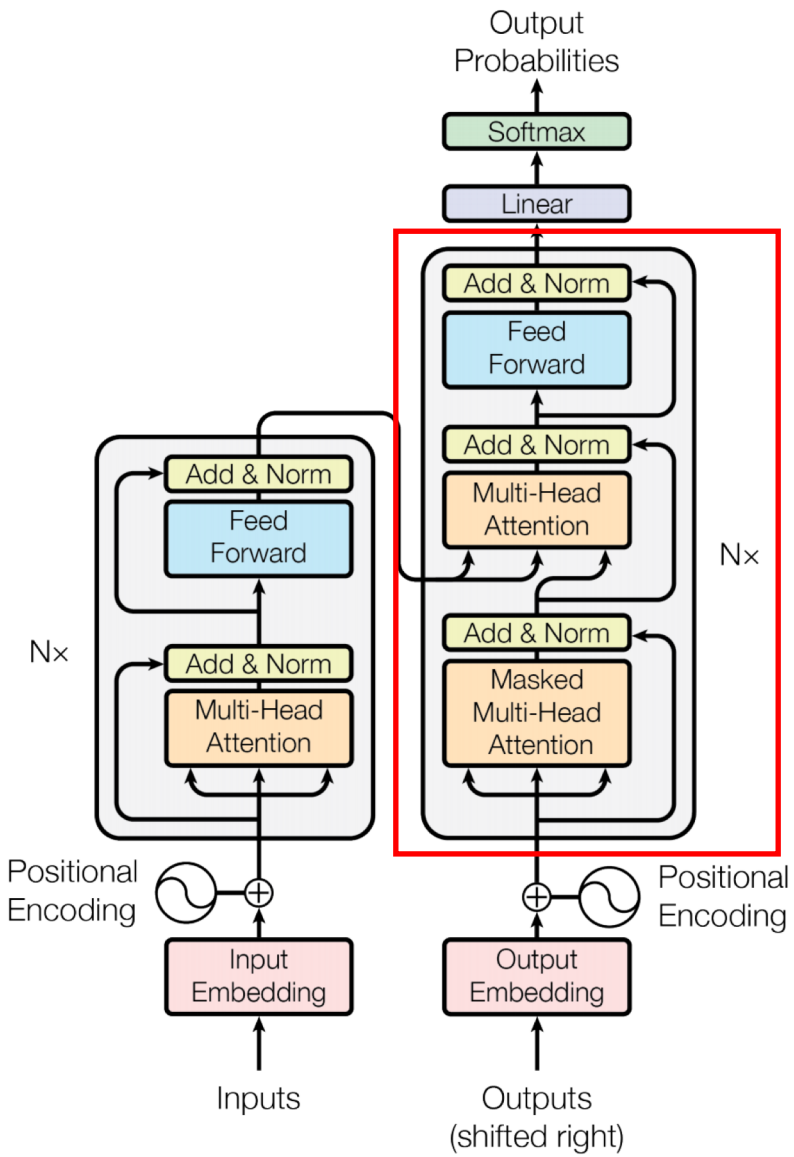


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Decoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- Encoder 와 동일한 2개의 sub-layer에 추가로 encoder output과 attention을 구하는 layer가 추가되었다.
- Encoder와 마찬가지로 residual connection과 layer normalization을 사용했다.

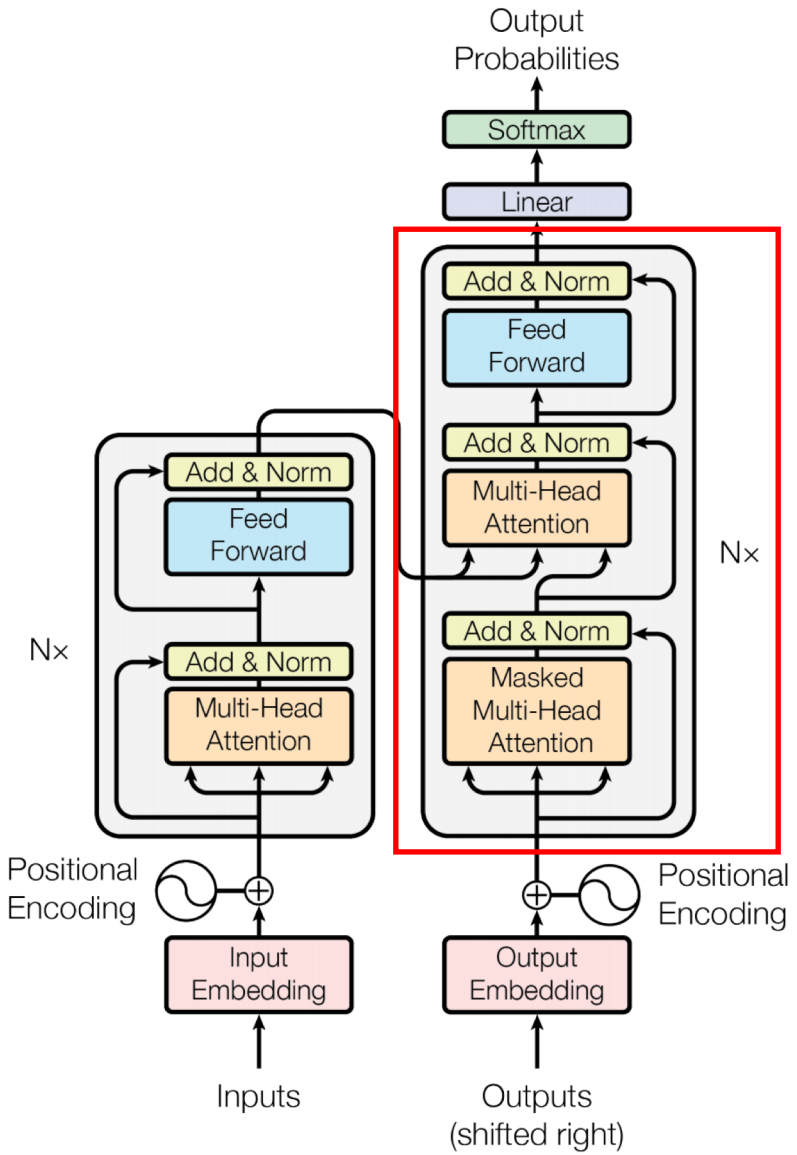


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

Decoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- Encoder 와 동일한 2개의 sub-layer에 추가로 encoder output과 attention을 구하는 layer가 추가되었다.
- Encoder와 마찬가지로 residual connection과 layer normalization을 사용했다.
- Self-Attention부분에 masking을 추가했다(subsequent masking)

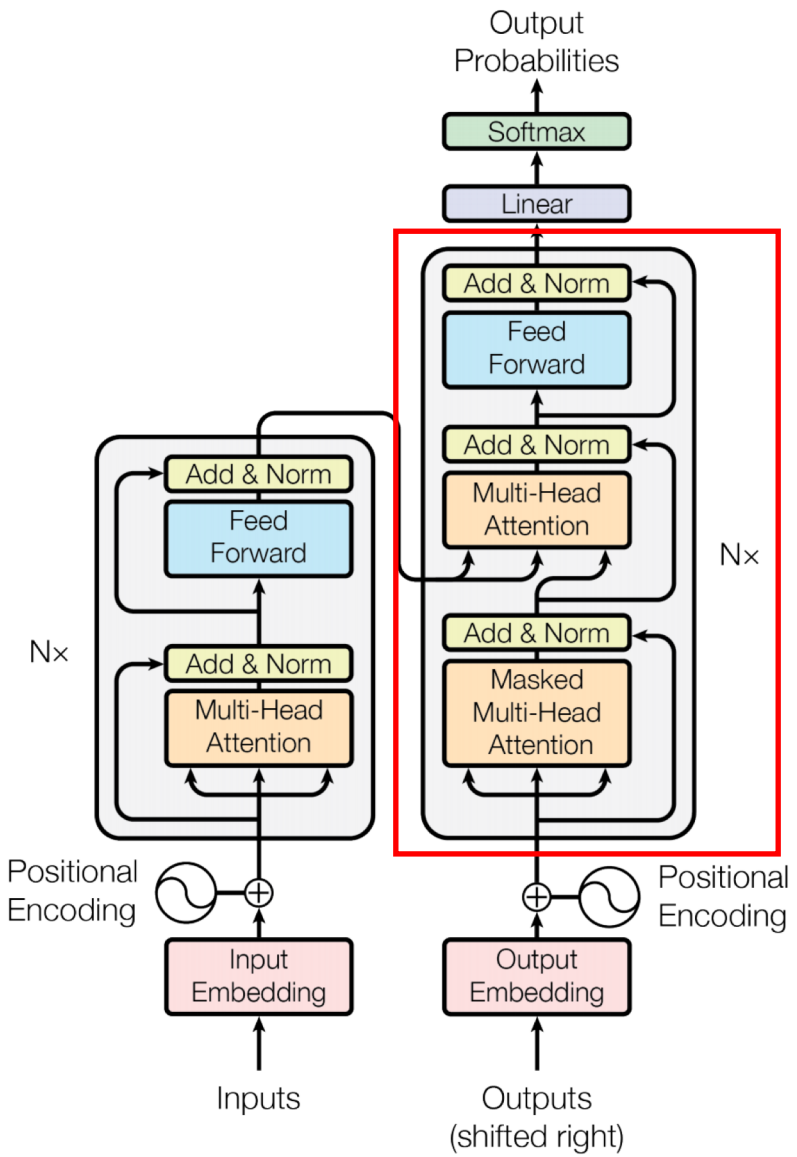


Attention Is All You Need

Model Architecture – Encoder Decoder Stacks

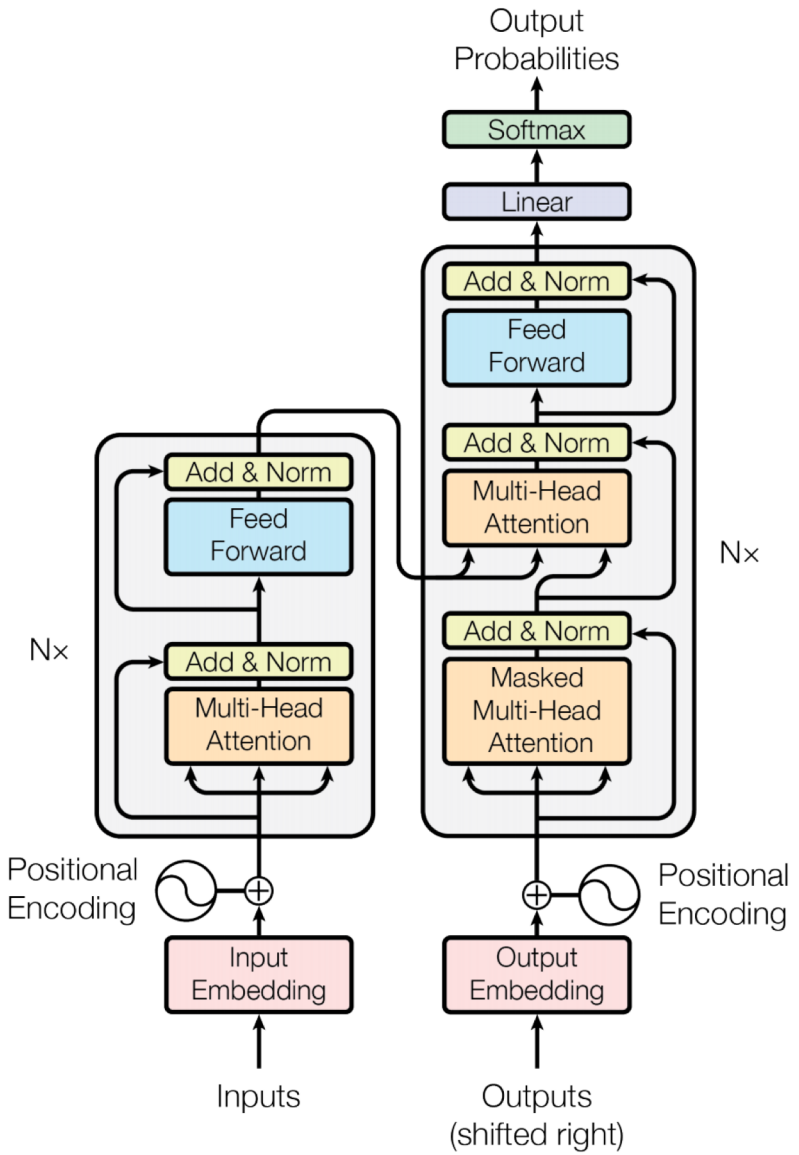
Decoder

- 총 N=6개의 동일한 layer(블럭)으로 구성되어 있다.
- Encoder 와 동일한 2개의 sub-layer에 추가로 encoder output과 attention을 구하는 layer가 추가되었다.
- Encoder와 마찬가지로 residual connection과 layer normalization을 사용했다.
- Self-Attention부분에 masking을 추가했다(subsequent masking)
  - t step 예측 시 t step 이후의 값을 이용하지 못하도록



Attention Is All You Need

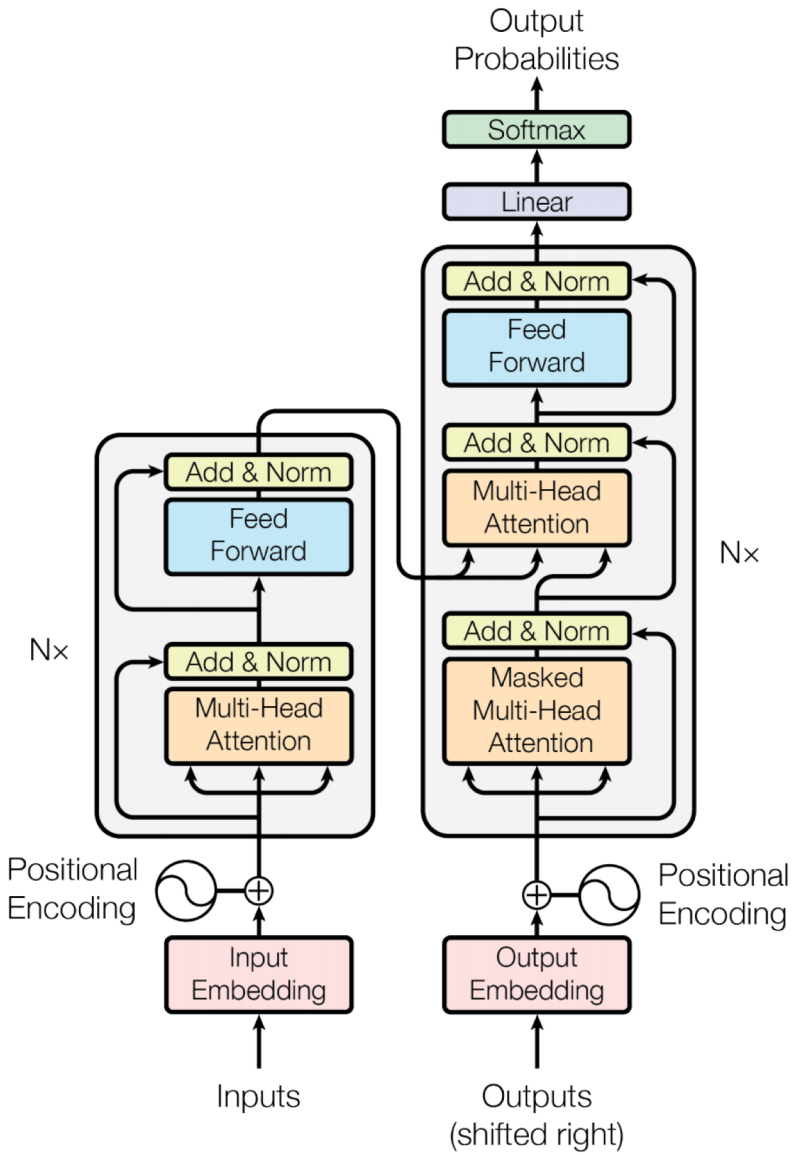
Model Architecture – Attention



Attention Is All You Need

Model Architecture – Attention

Scaled-dot Attention

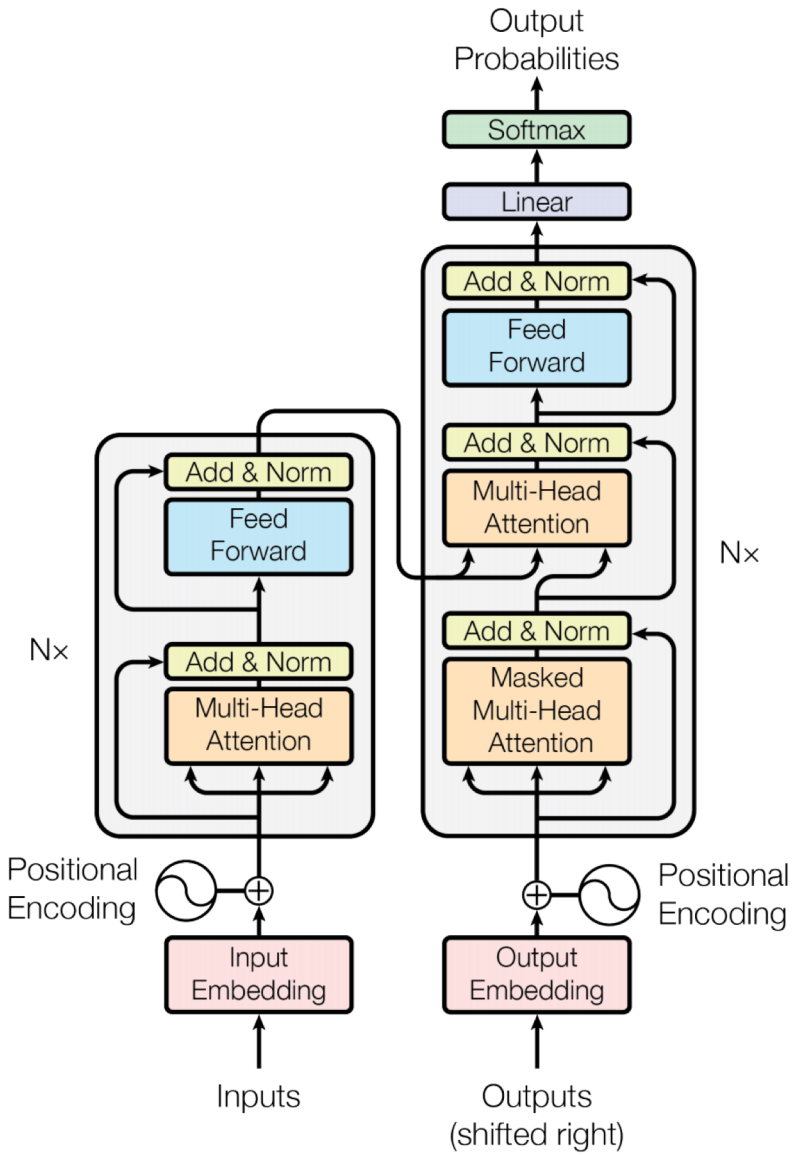


Attention Is All You Need

Model Architecture – Attention

Scaled-dot Attention

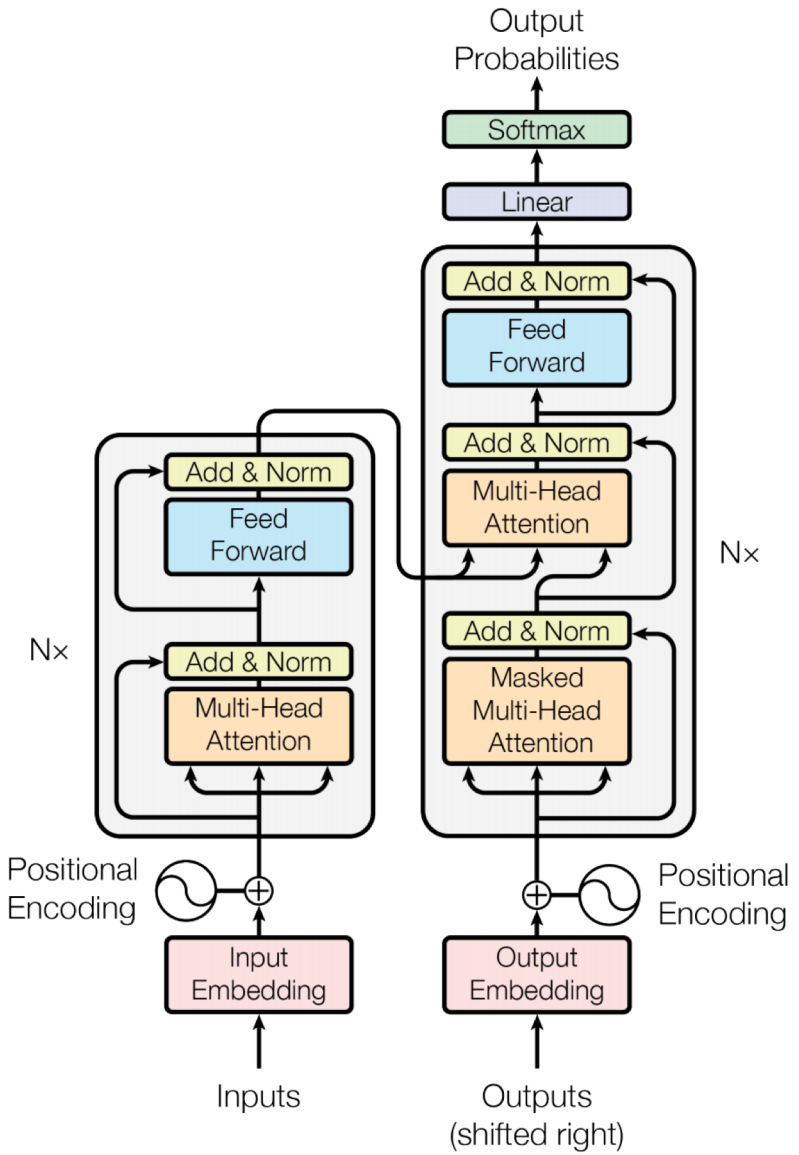
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Attention Is All You Need

Model Architecture – Attention

Multi-Head Attention

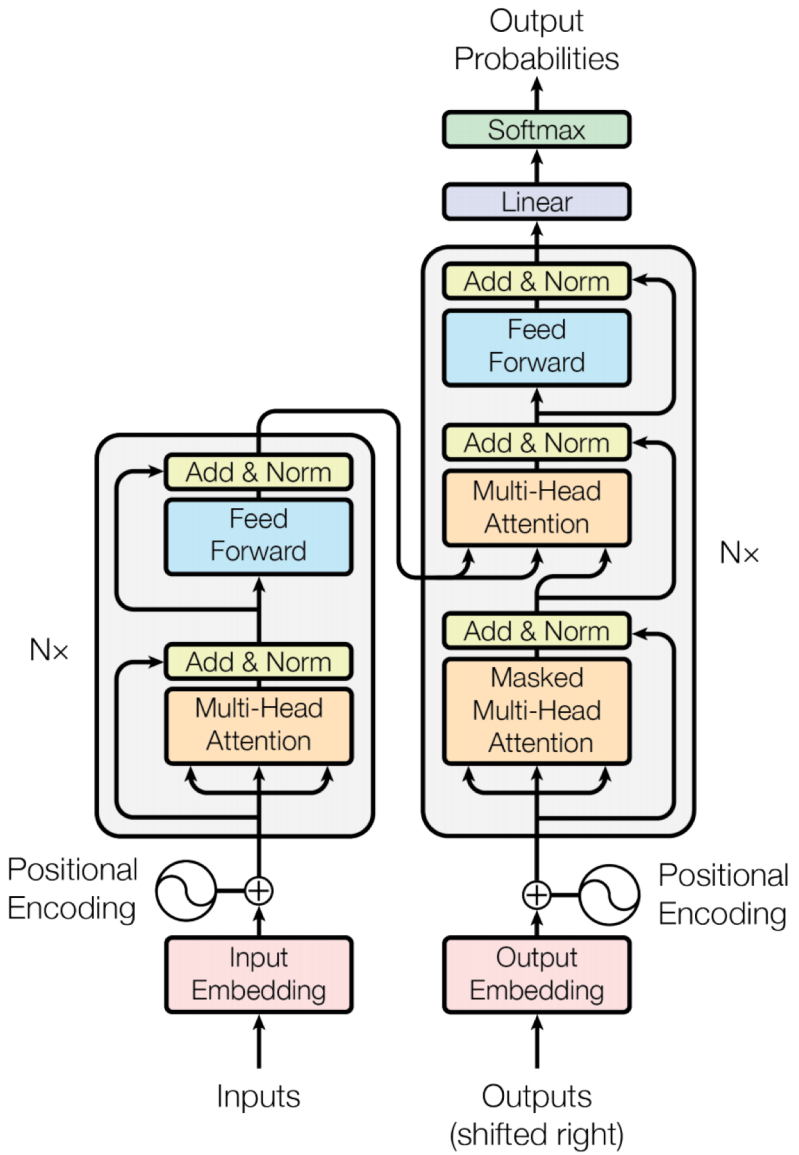


Attention Is All You Need

Model Architecture – Attention

Multi-Head Attention

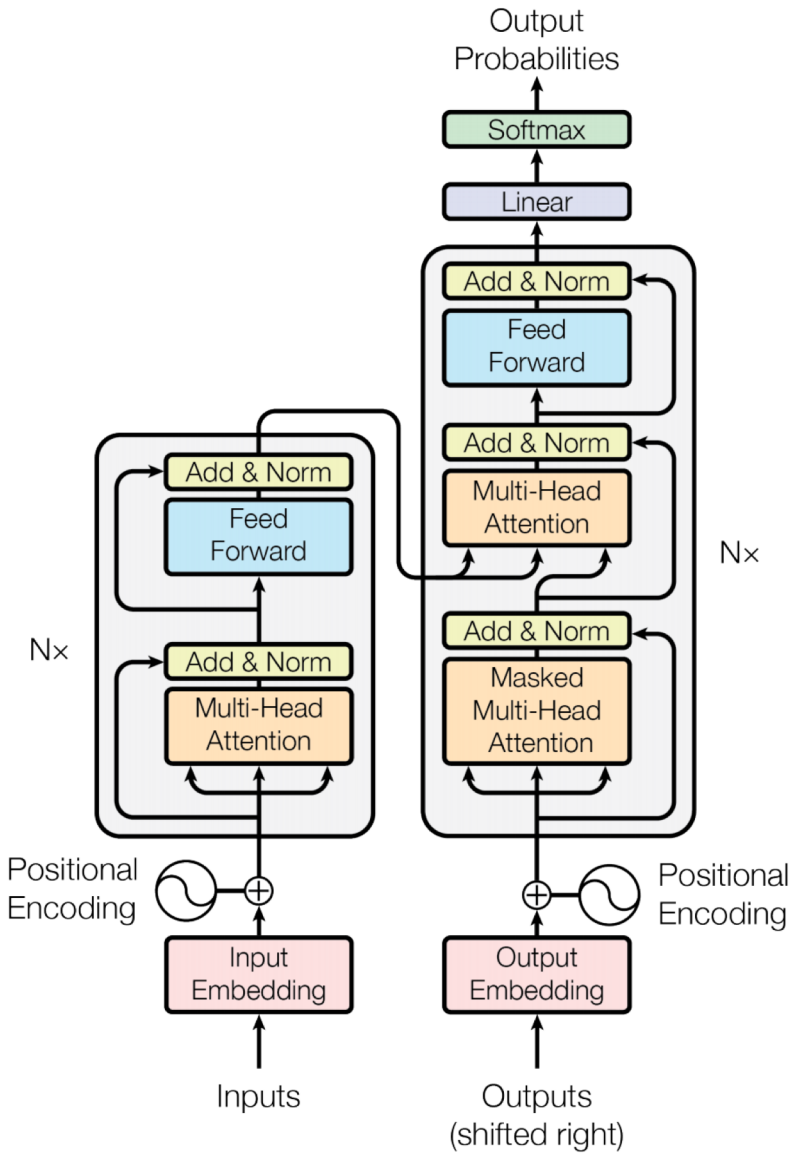
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Attention Is All You Need

Model Architecture – Attention

Applications of Attention in our Model

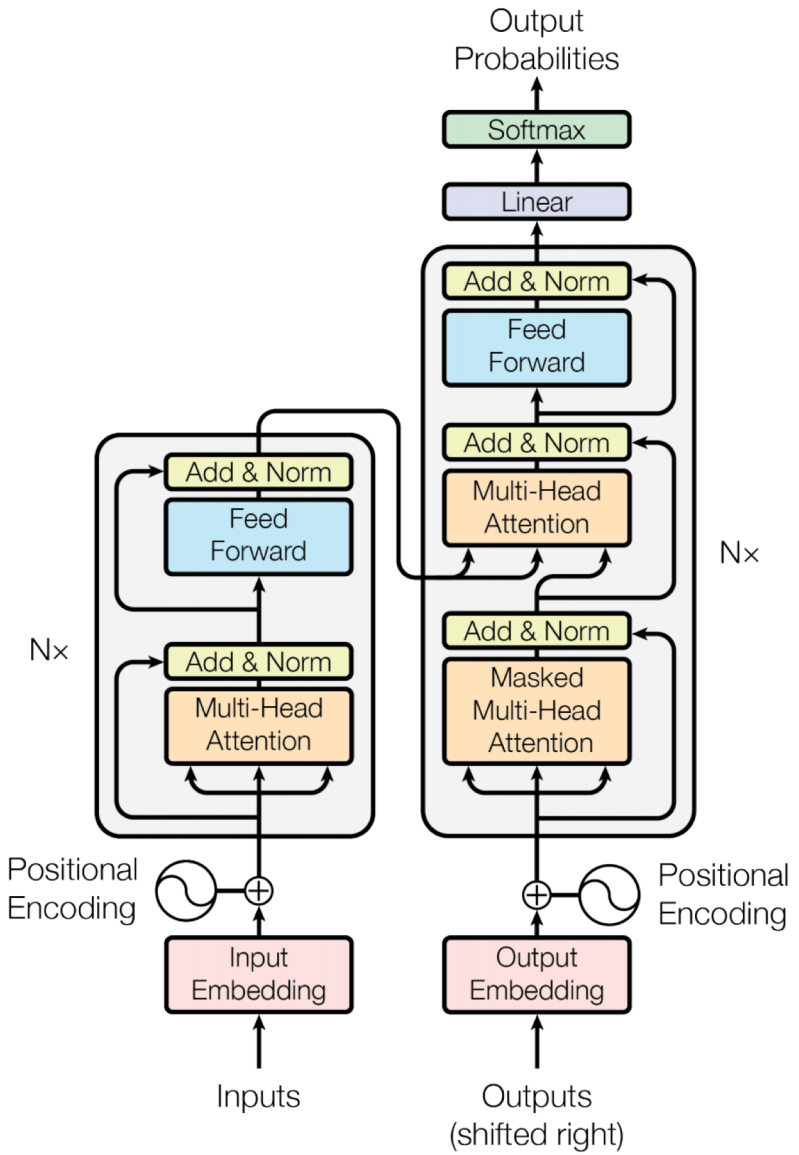


Attention Is All You Need

Model Architecture – Attention

Applications of Attention in our Model

- In "Encoder-Decoder Attention Layer" query는 decoder의 이전 layer에서 오고 key, value는 encoder의 output이다.

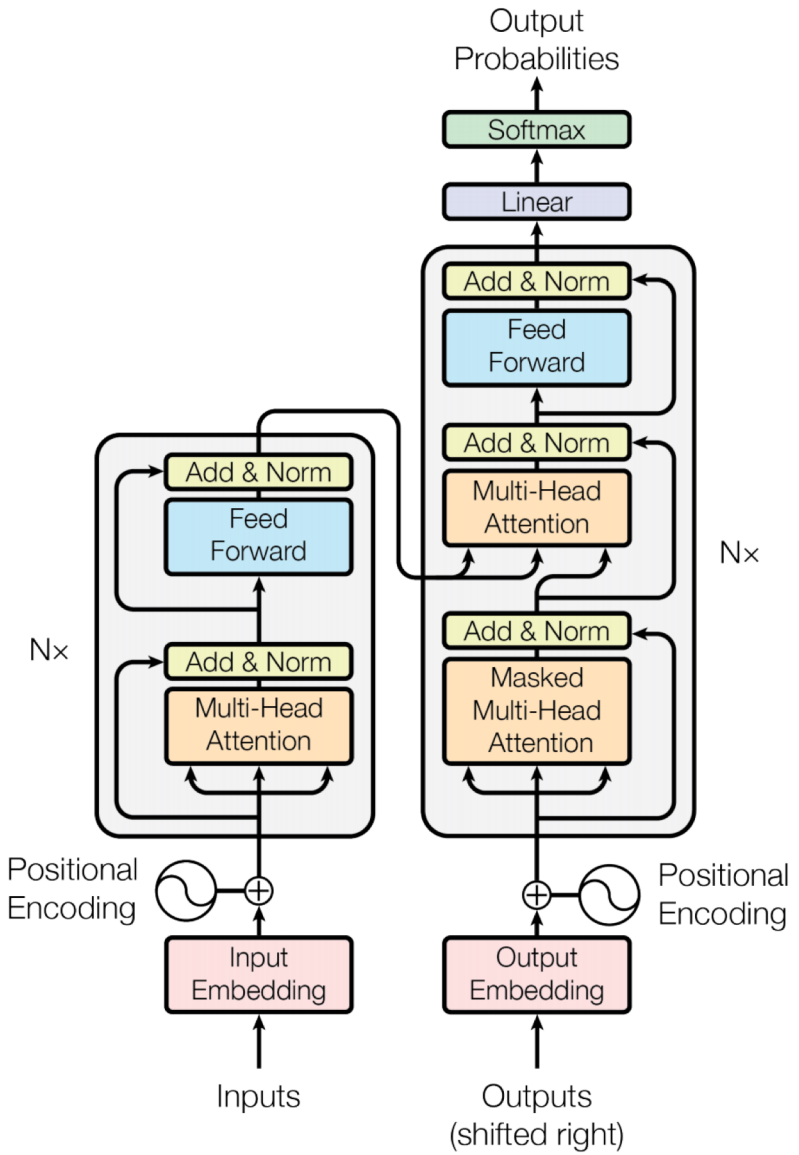


Attention Is All You Need

Model Architecture – Attention

Applications of Attention in our Model

- In "Encoder-Decoder Attention Layer" query는 decoder의 이전 layer에서 오고 key, value는 encoder의 output이다.
- Encoder에서의 Self-Attention은 query, key, value가 모두 동일한 값에서 생성된다.

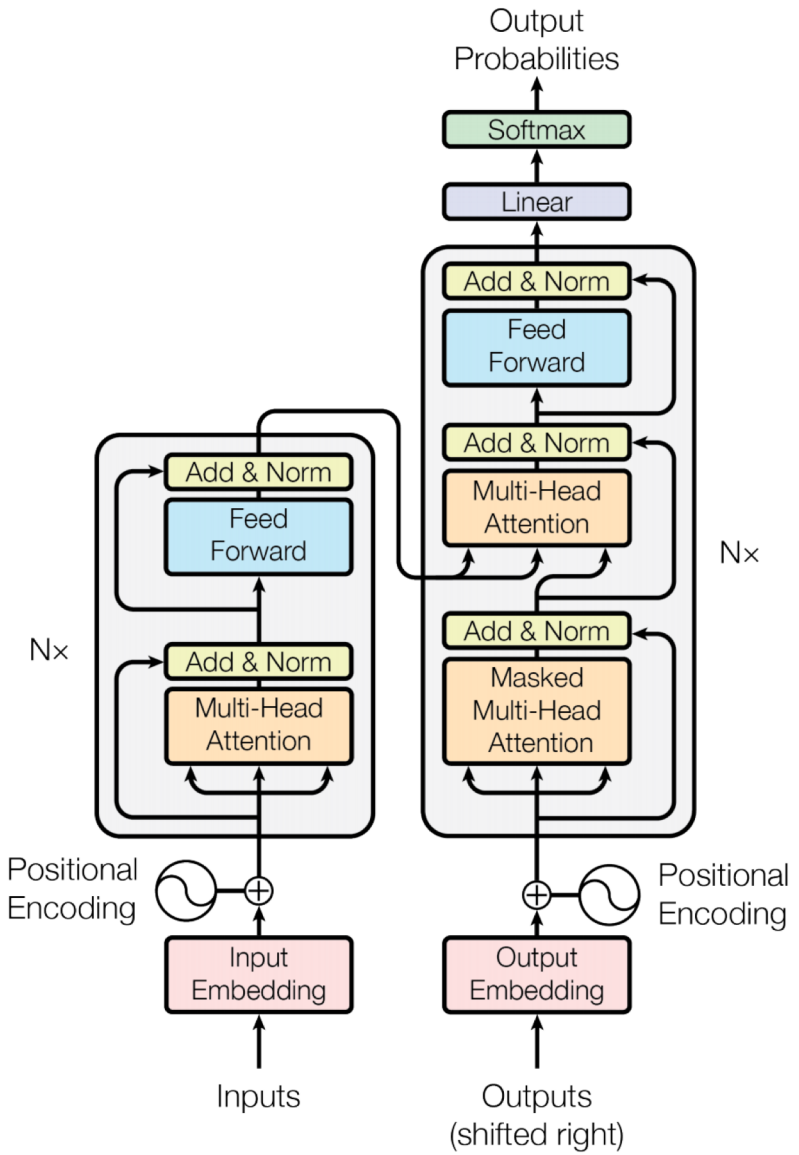


Attention Is All You Need

Model Architecture – Attention

Applications of Attention in our Model

- In "Encoder-Decoder Attention Layer" query는 decoder의 이전 layer에서 오고 key, value는 encoder의 output이다.
- Encoder에서의 Self-Attention은 query, key, value가 모두 동일한 값에서 생성된다.
  - Encoder에서는 모든 position을 attend 할 수 있다.

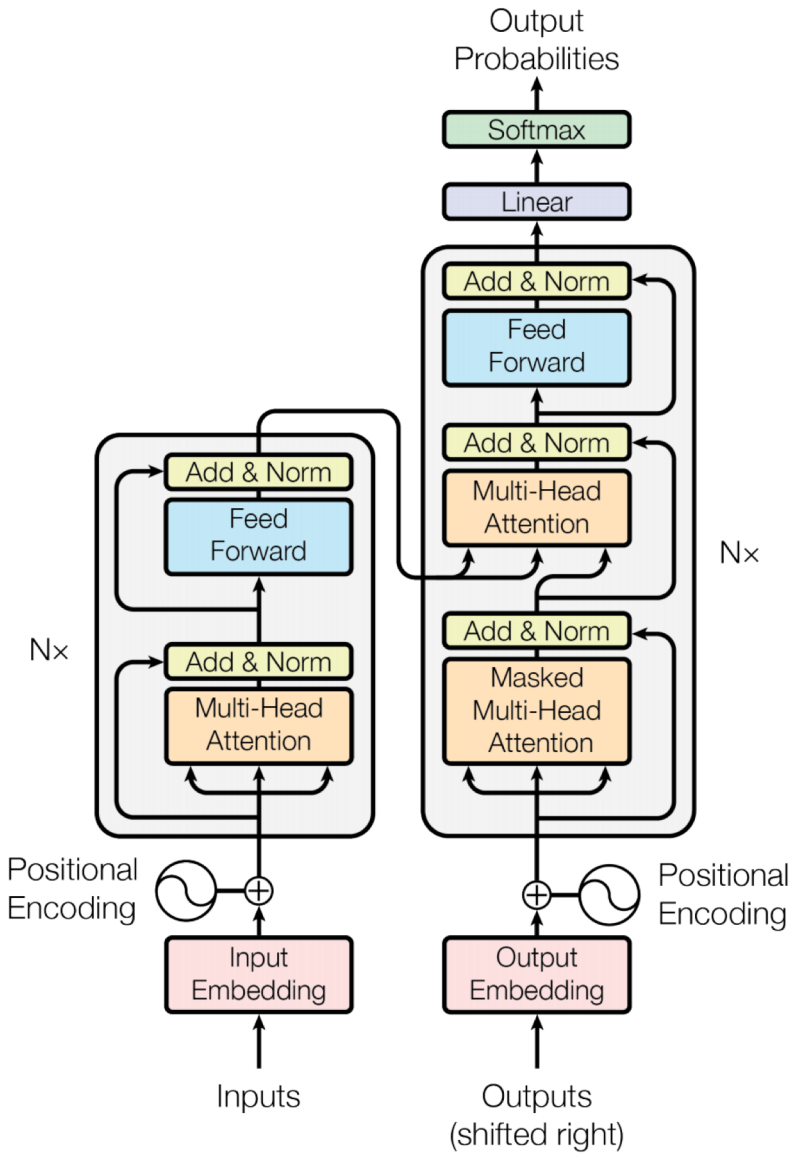


Attention Is All You Need

Model Architecture – Attention

Applications of Attention in our Model

- In "Encoder-Decoder Attention Layer" query는 decoder의 이전 layer에서 오고 key, value는 encoder의 output이다.
- Encoder에서의 Self-Attention은 query, key, value가 모두 동일한 값에서 생성된다.
  - Encoder에서는 모든 position을 attend 할 수 있다.
- Decoder에서의 Self-Attention은 해당 position이전 까지만 attend할 수 있도록 masking처리한다.

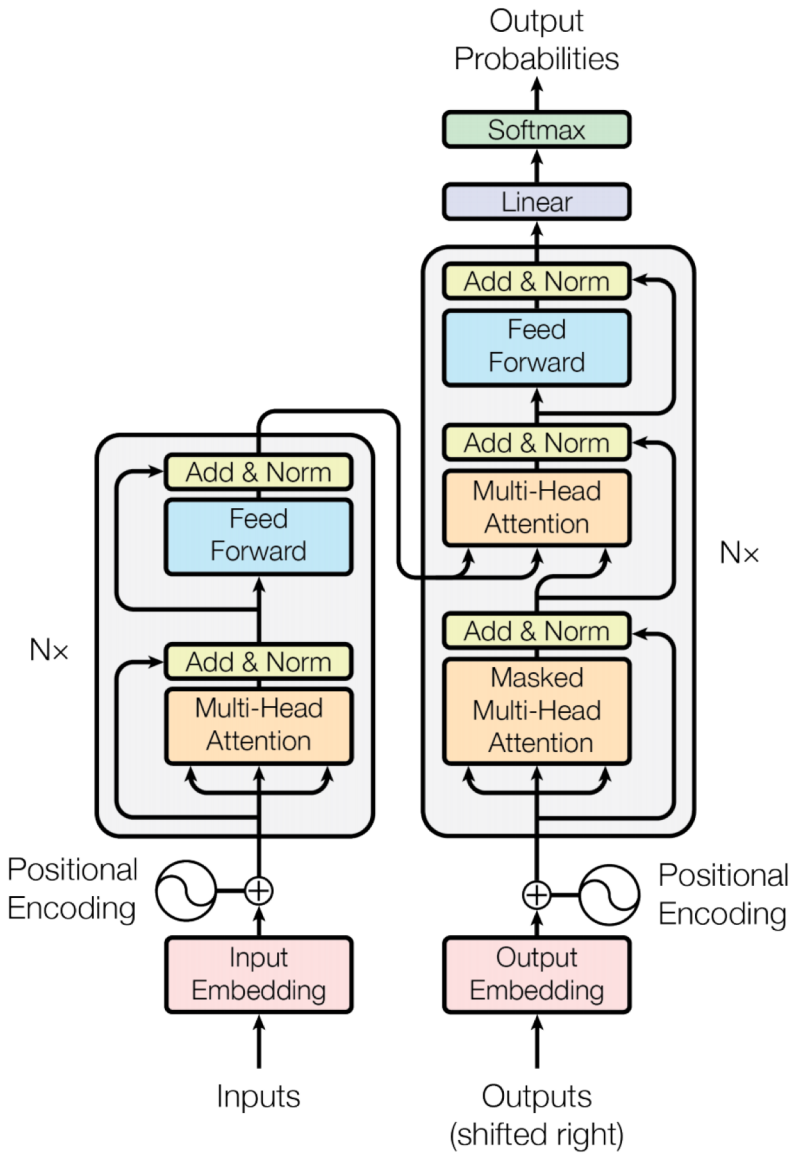


Attention Is All You Need

Model Architecture – Attention

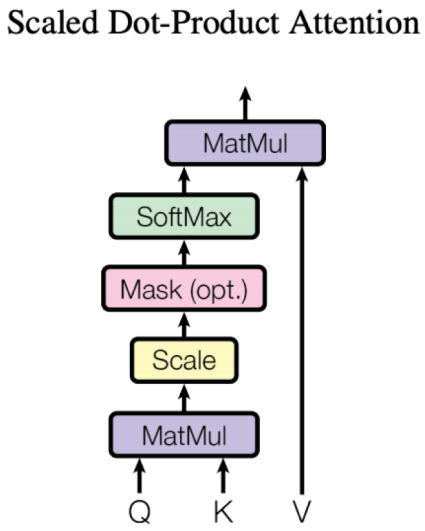
Applications of Attention in our Model

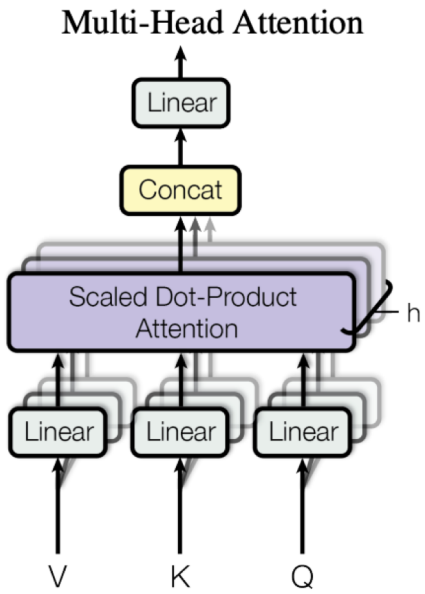
- In "Encoder-Decoder Attention Layer" query는 decoder의 이전 layer에서 오고 key, value는 encoder의 output이다.
- Encoder에서의 Self-Attention은 query, key, value가 모두 동일한 값에서 생성된다.
  - Encoder에서는 모든 position을 attend 할 수 있다.
- Decoder에서의 Self-Attention은 해당 position이전 까지만 attend할 수 있도록 masking처리한다.
  - Masking값은 $-\infty$로 처리한다.

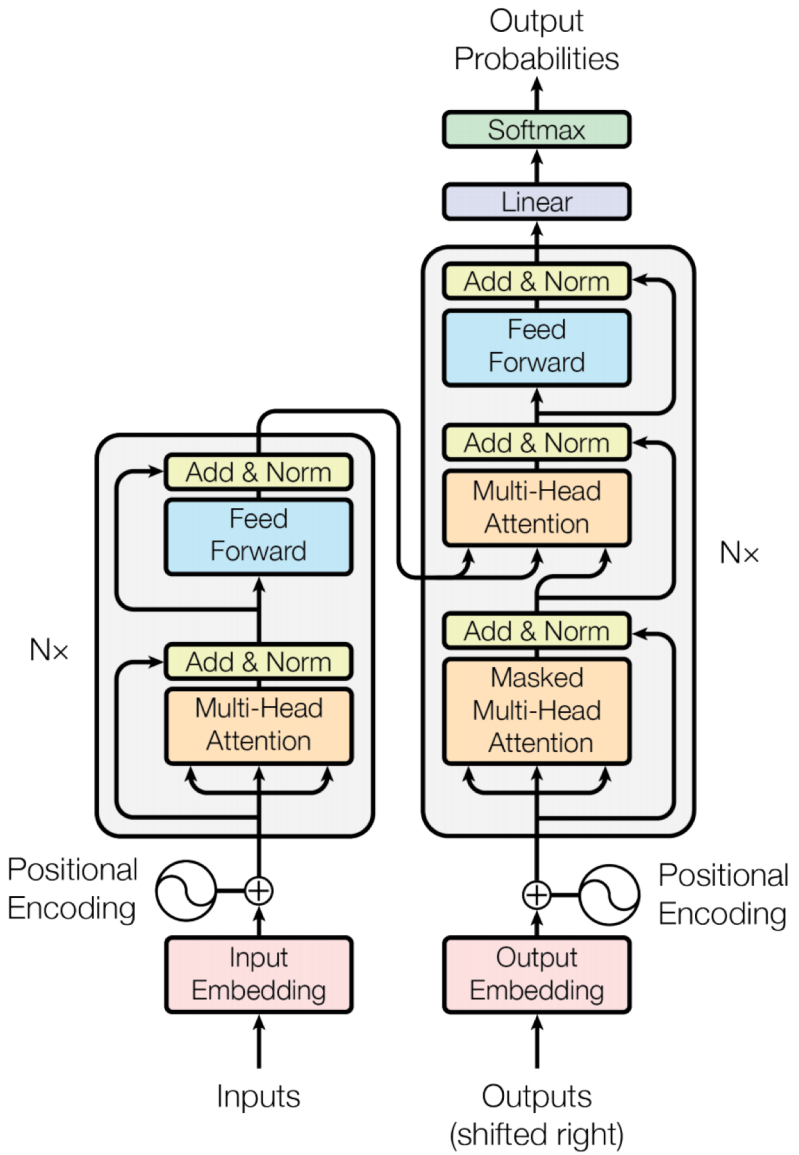


Attention Is All You Need

Model Architecture – Attention



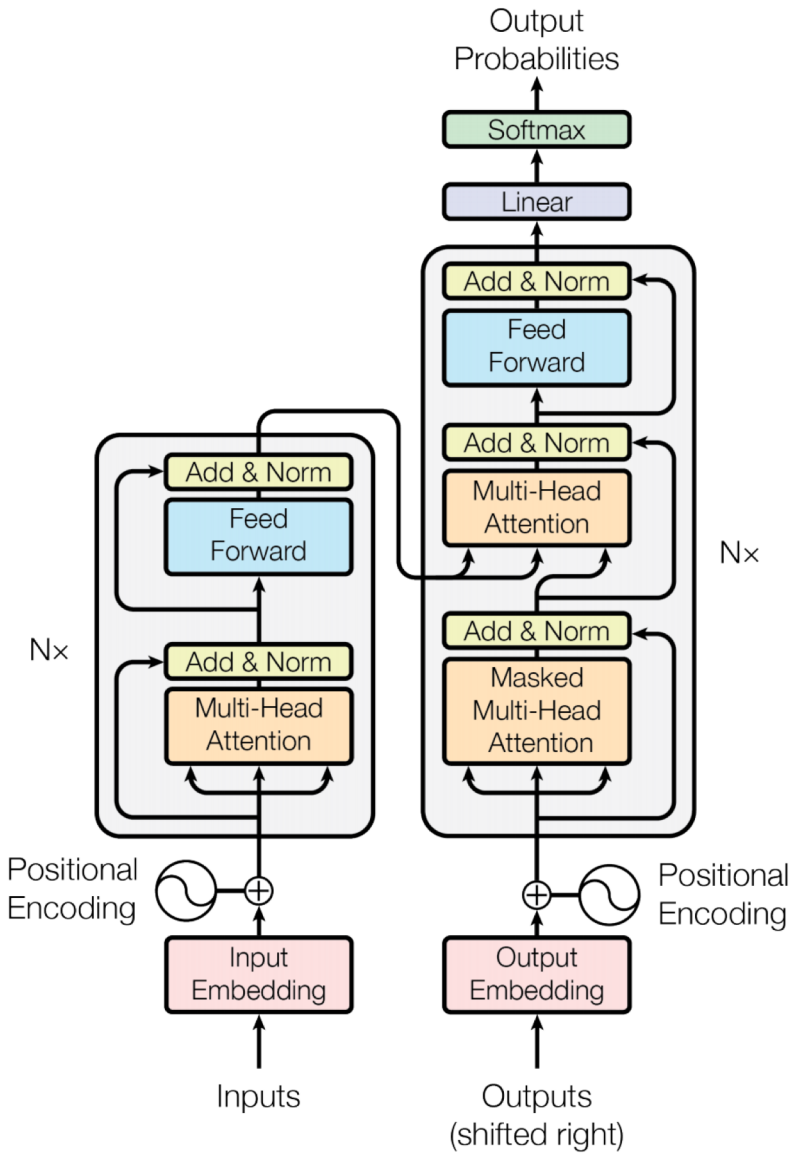




Attention Is All You Need

Model Architecture – Attention

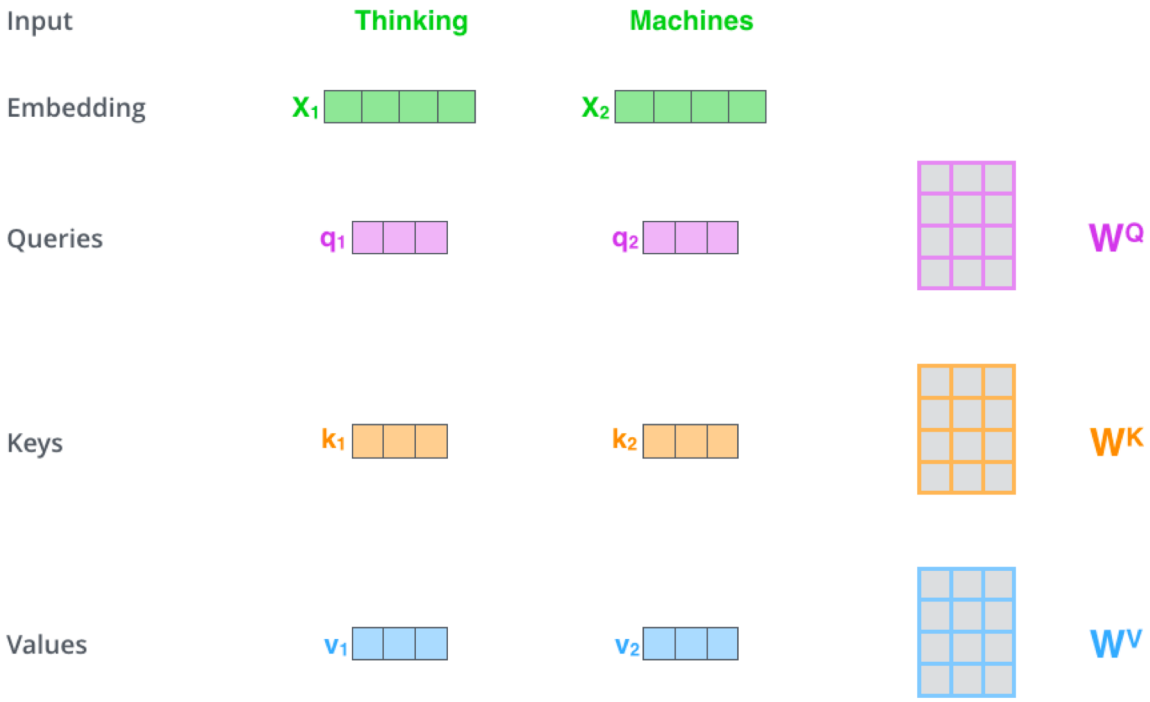
Illustrated-transformer(http://jalammar.github.io/illustrated-transformer/)

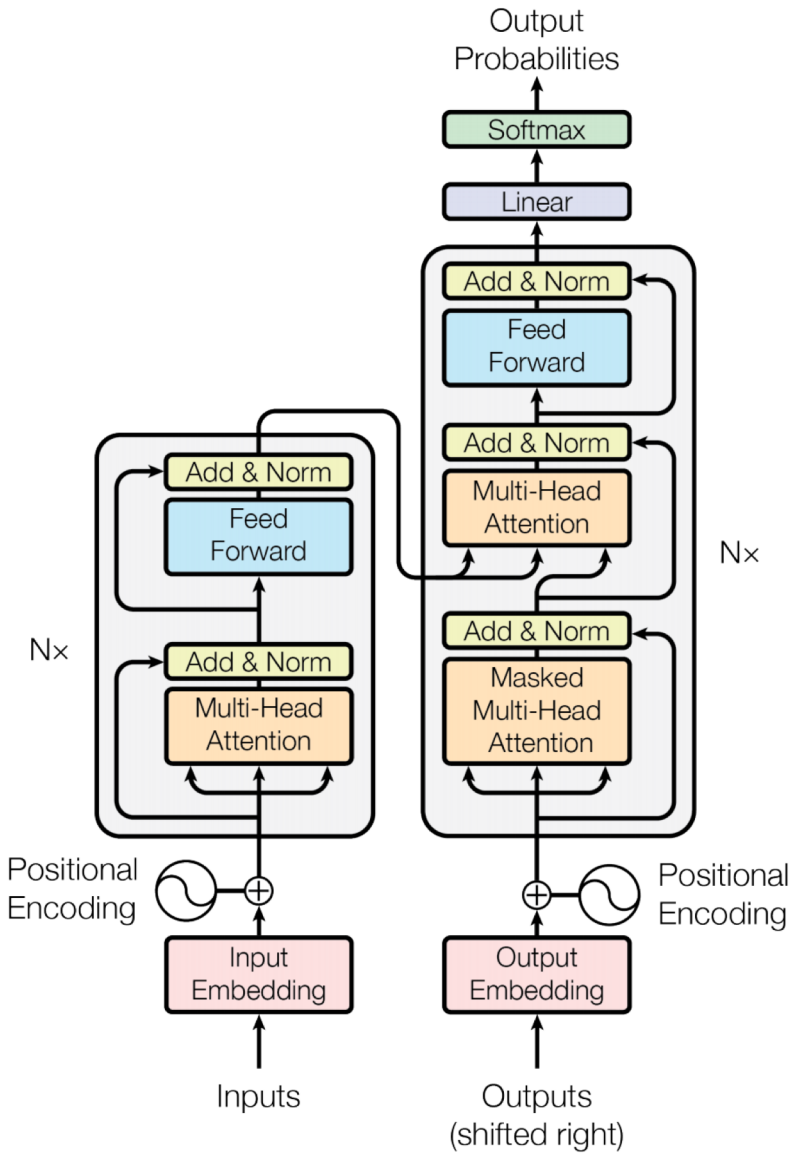


Attention Is All You Need

Model Architecture – Attention

Illustrated-transformer(http://jalammar.github.io/illustrated-transformer/)

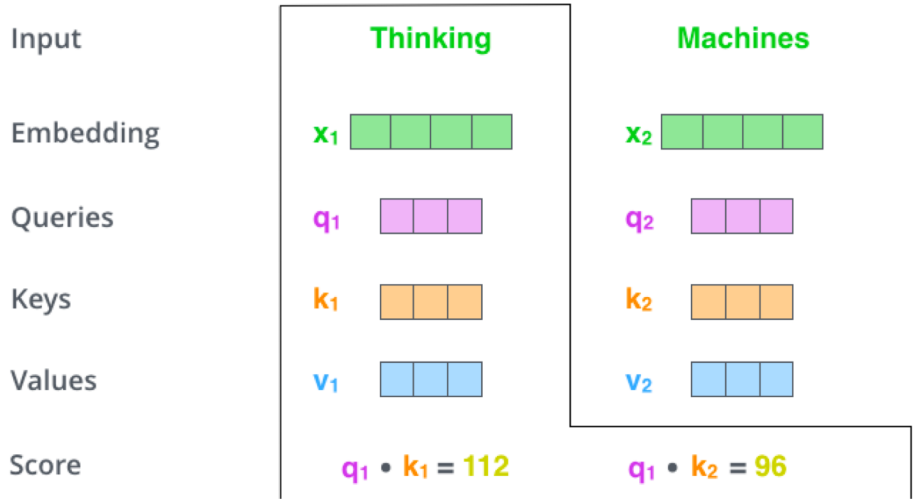


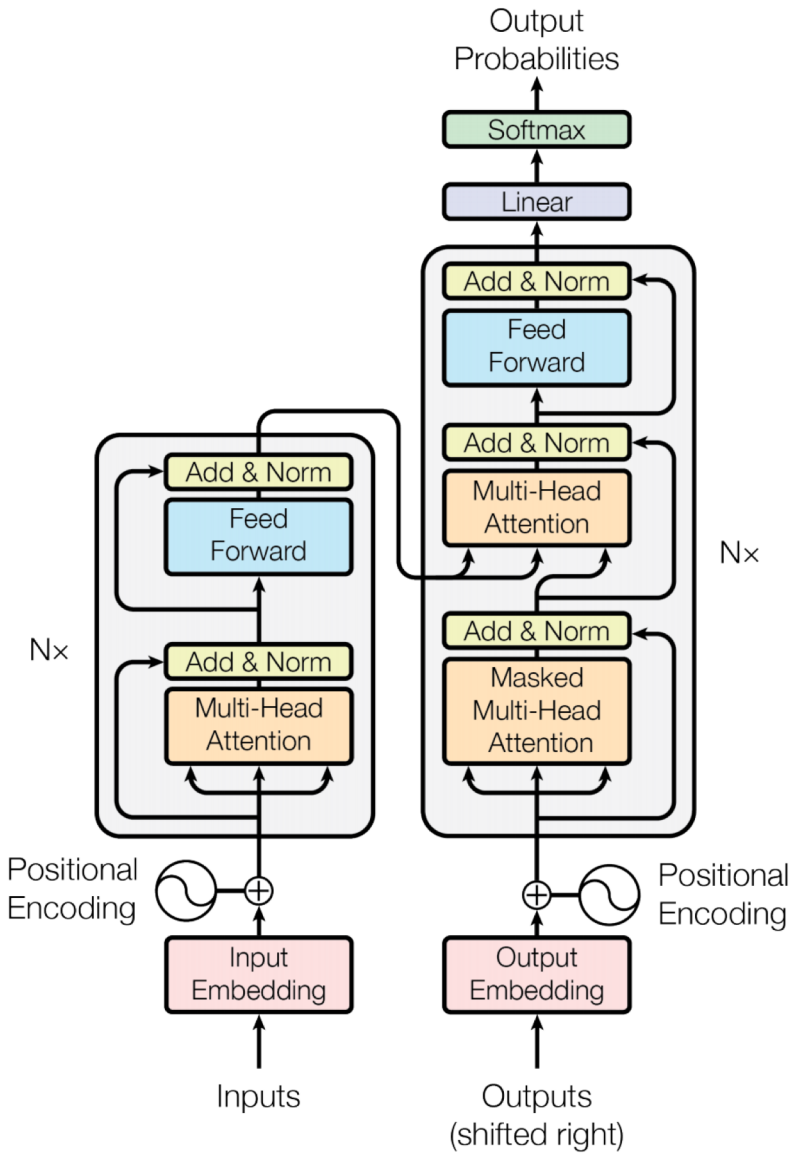


Attention Is All You Need

Model Architecture – Attention

Illustrated-transformer(http://jalammar.github.io/illustrated-transformer/)

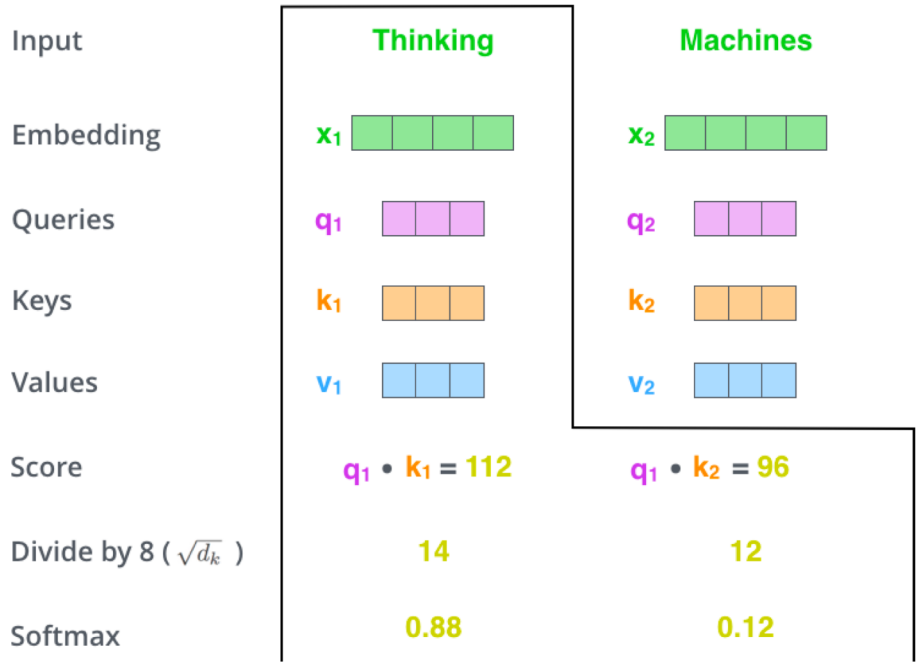


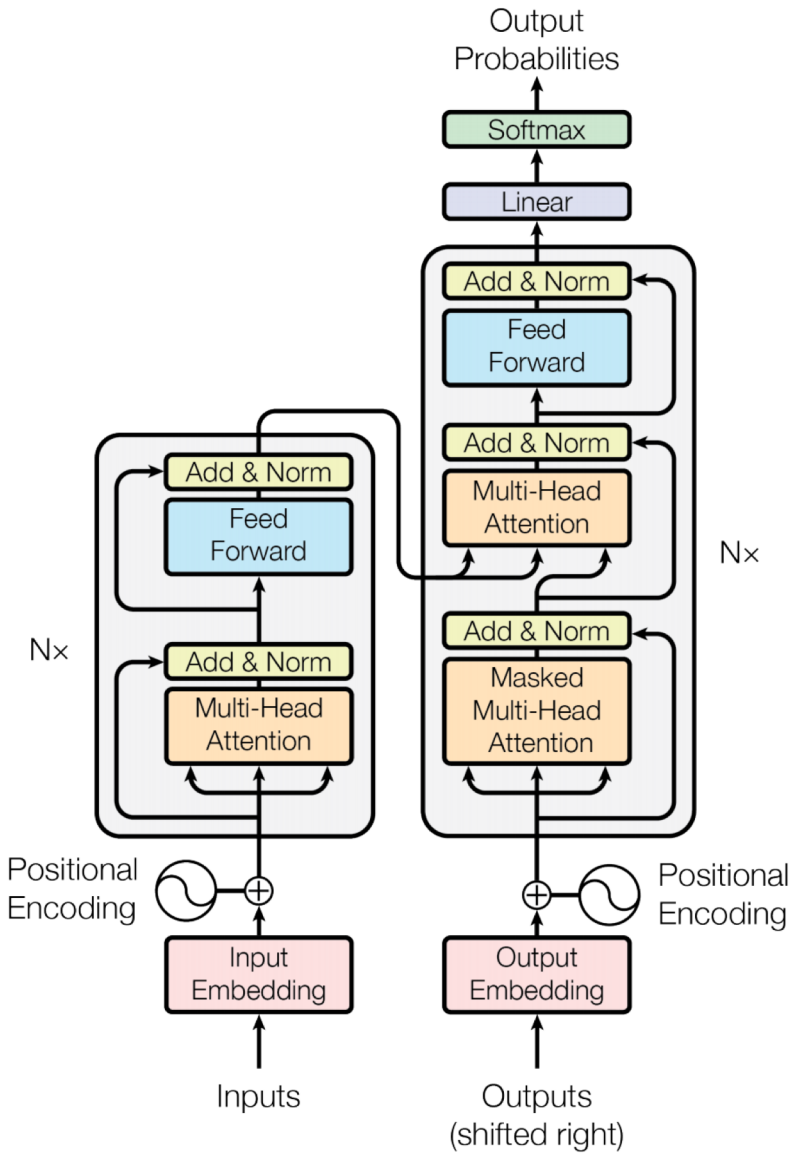


Attention Is All You Need

Model Architecture – Attention

Illustrated-transformer(http://jalammar.github.io/illustrated-transformer/)

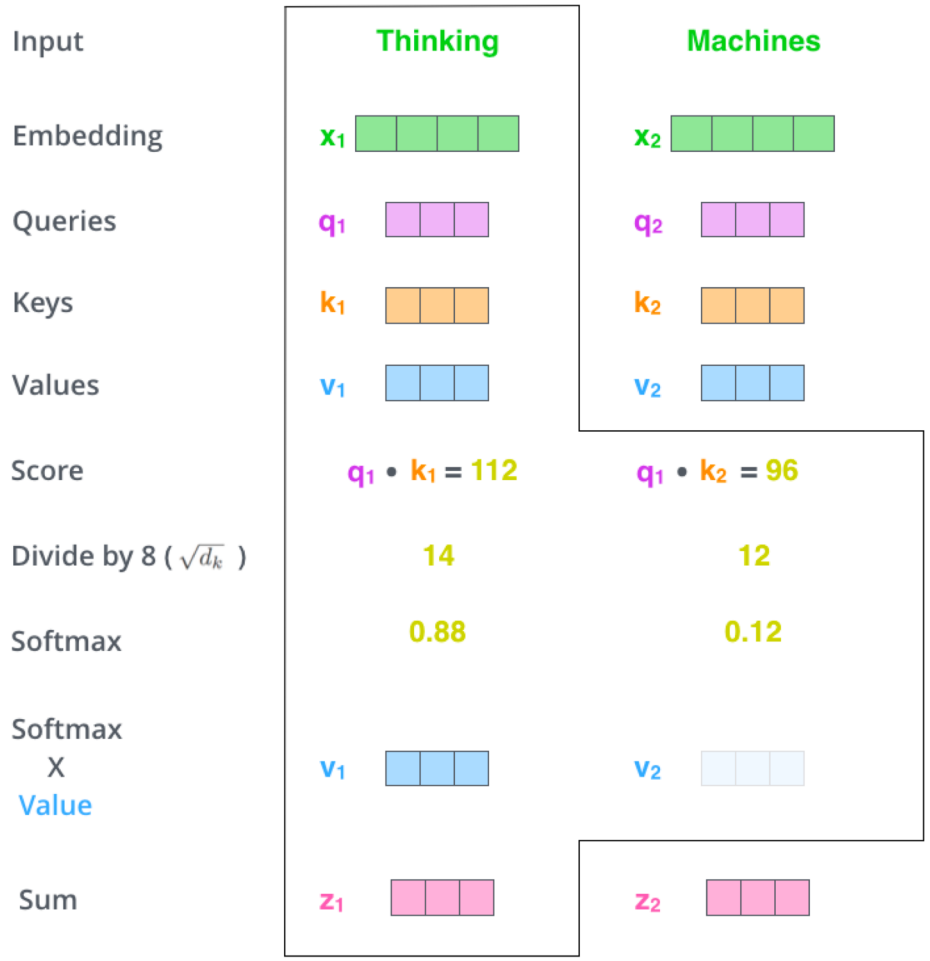


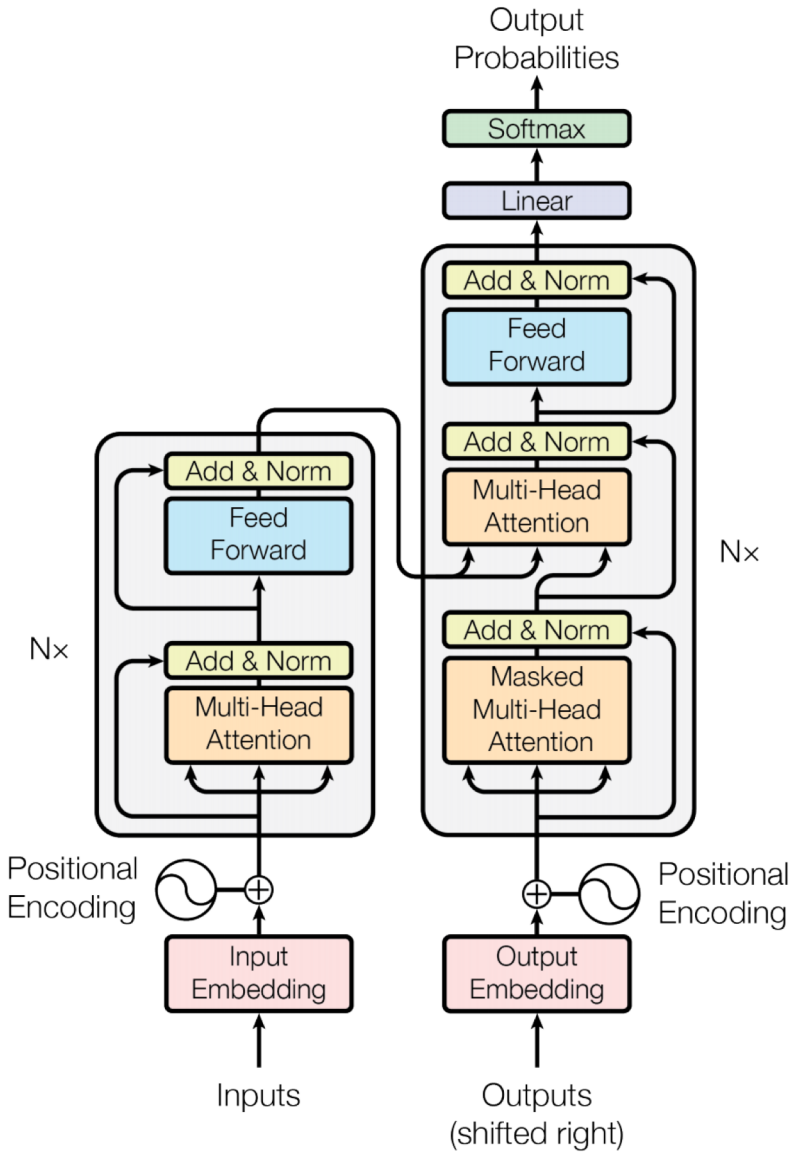


Attention Is All You Need

Model Architecture – Attention

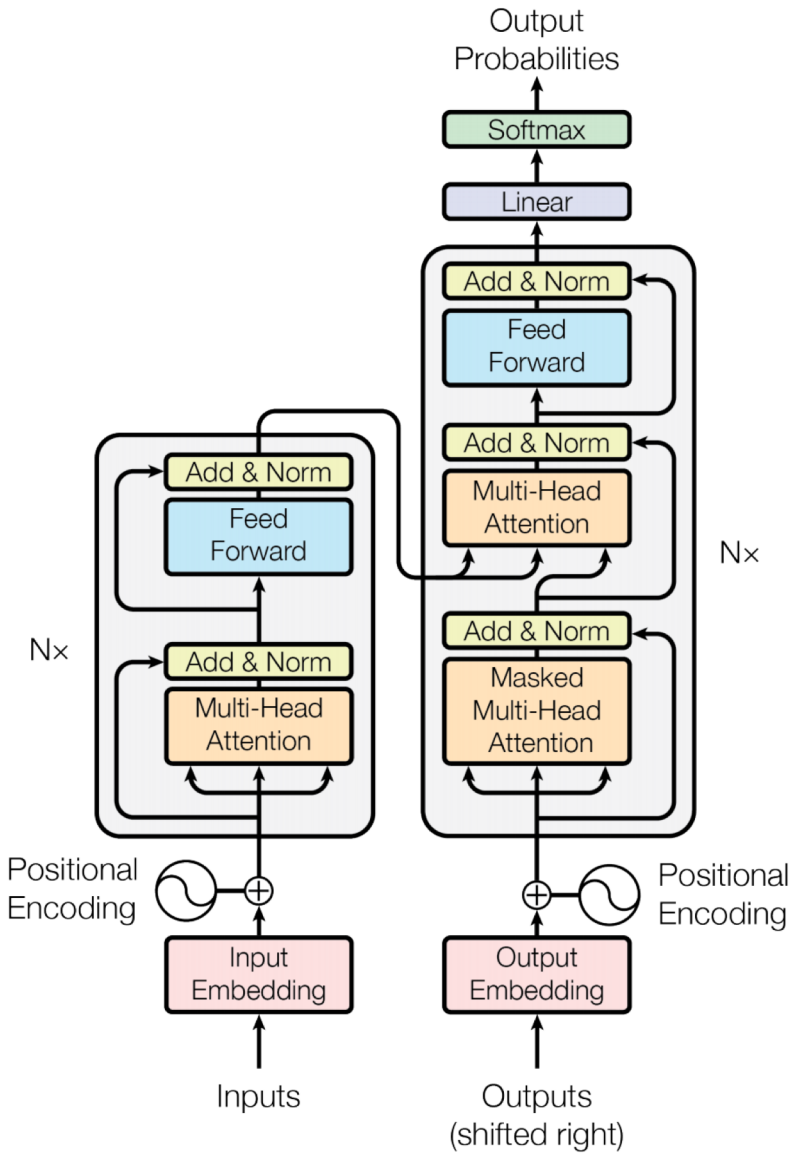
Illustrated-transformer(http://jalammar.github.io/illustrated-transformer/)





Attention Is All You Need

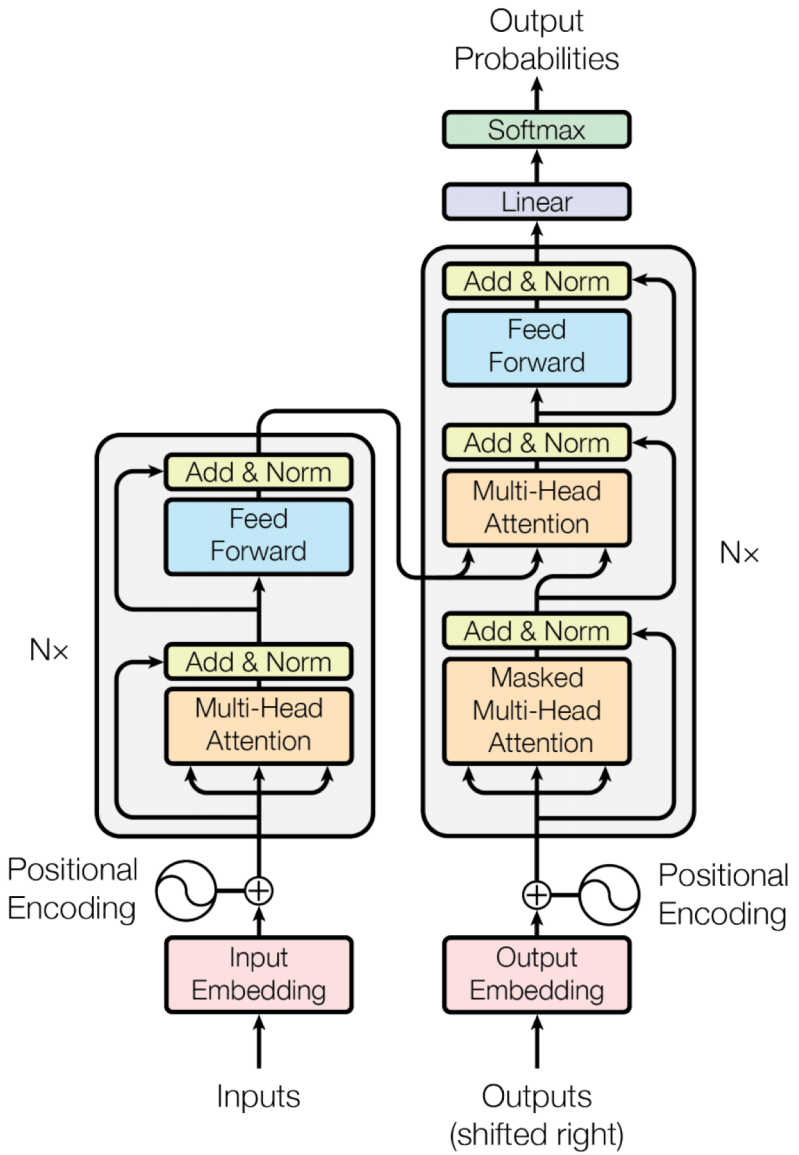
Model Architecture – Position-wise Feed-Forward Networks



Attention Is All You Need

Model Architecture – Position-wise Feed-Forward Networks

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

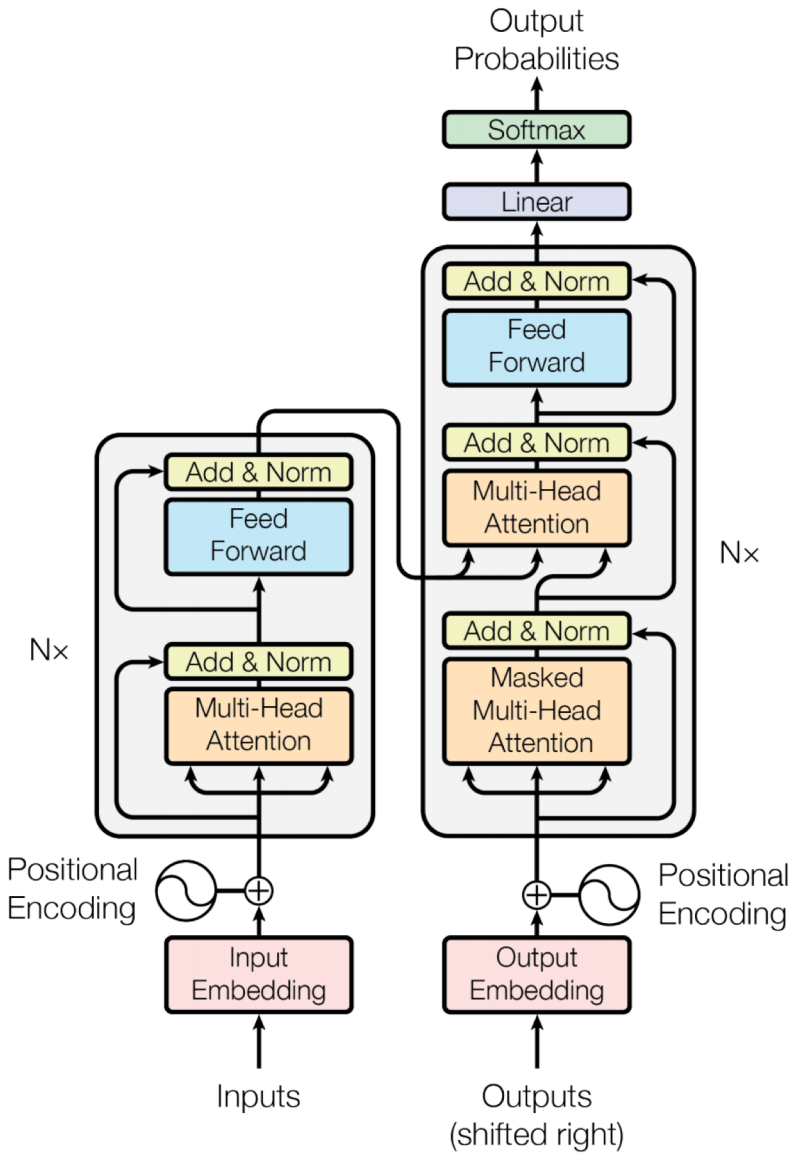


Attention Is All You Need

Model Architecture – Position-wise Feed-Forward Networks

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

- $d_f = 512,\ d_{ff} = 2048$

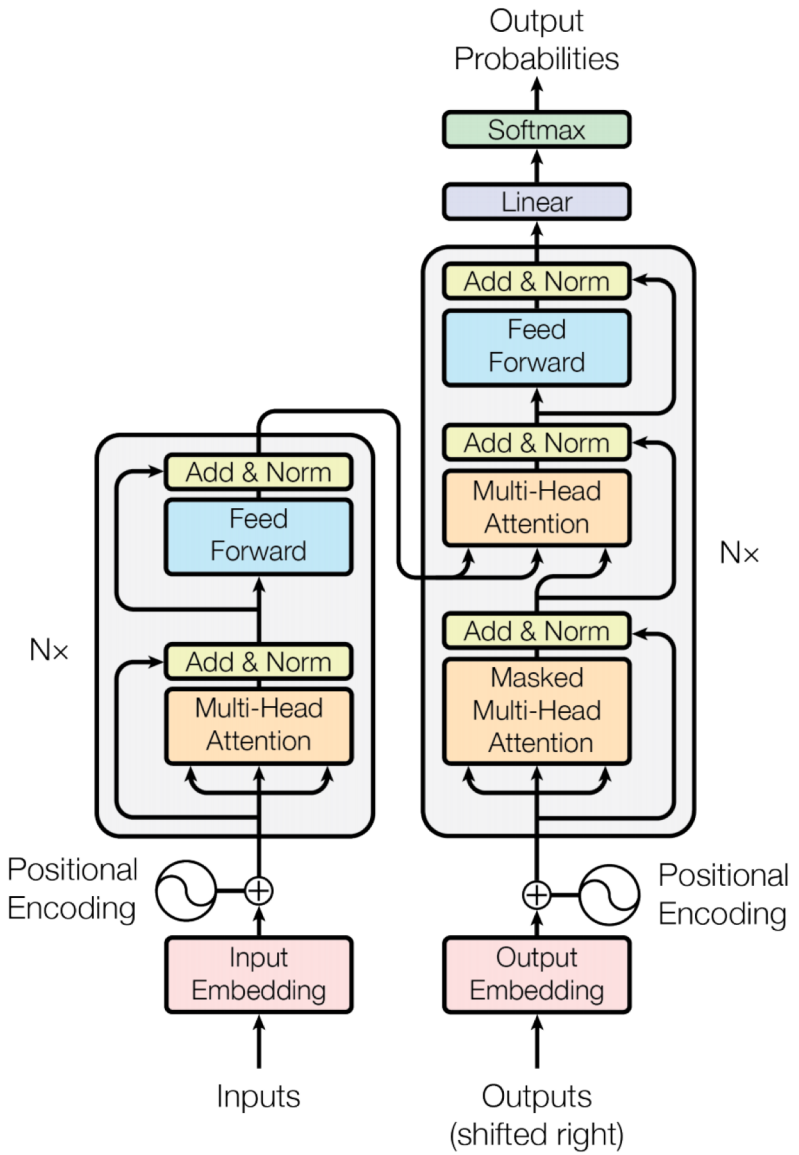


Attention Is All You Need

Model Architecture – Position-wise Feed-Forward Networks

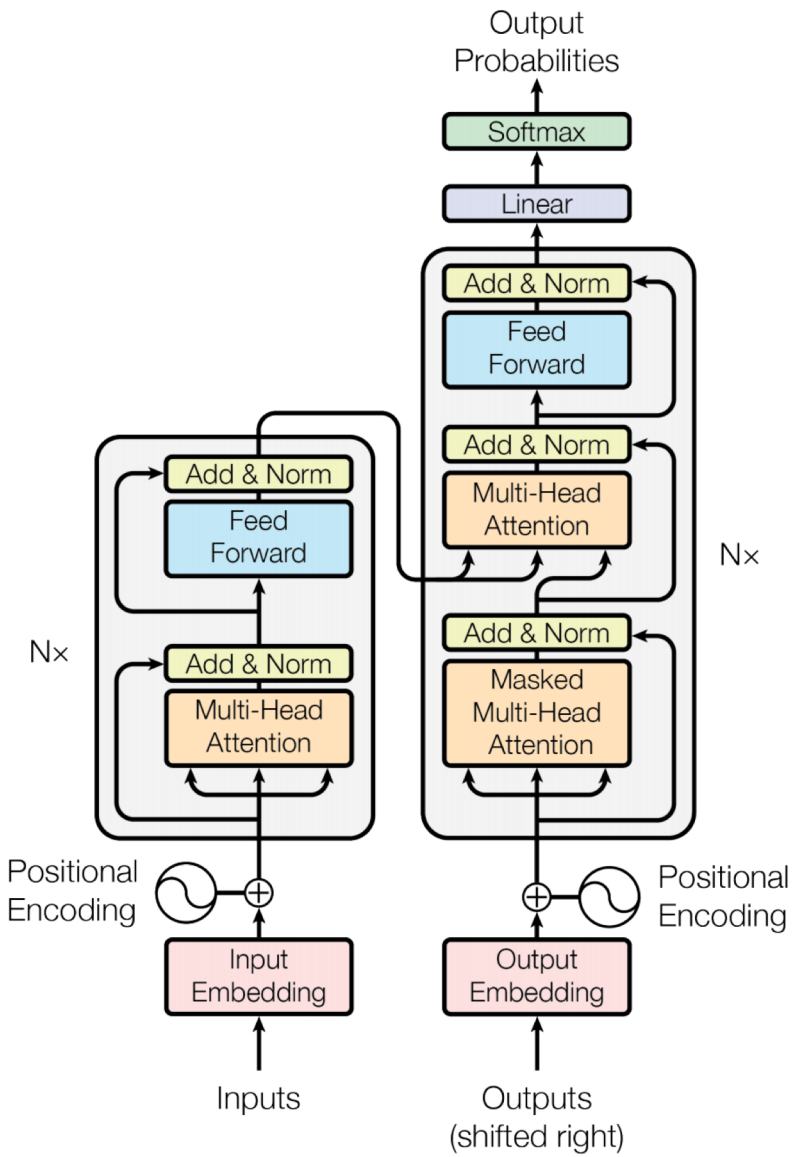
$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

- $d_f = 512,\ d_{ff} = 2048$
- Convolutional networks with kernel size 1 사용해도 됨



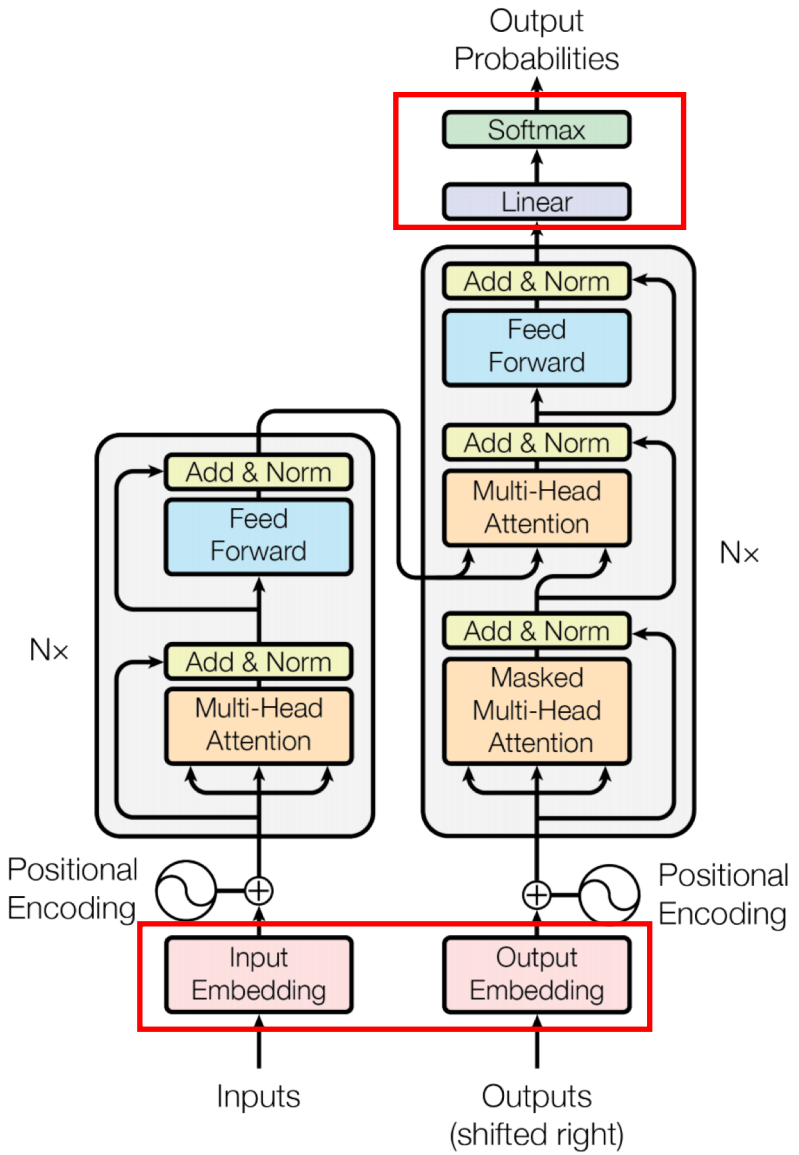
Attention Is All You Need

Model Architecture – Embedding Softmax



Attention Is All You Need

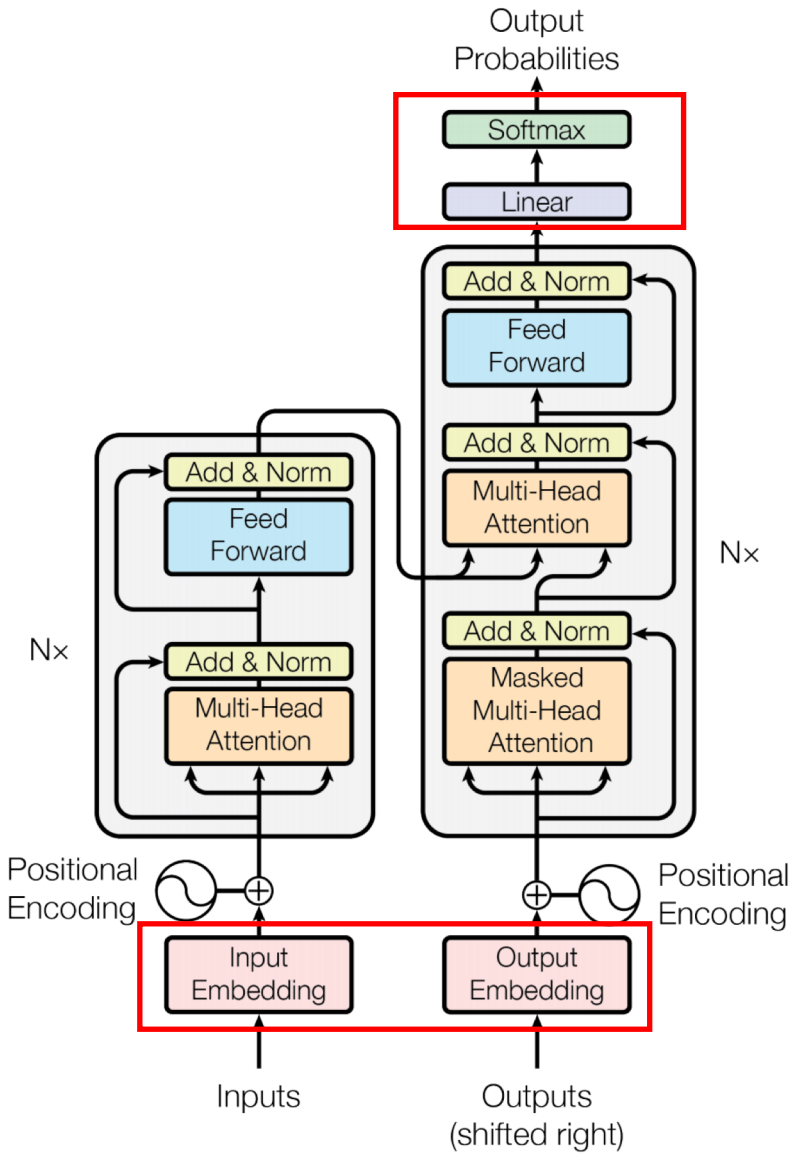
Model Architecture – Embedding Softmax



Attention Is All You Need

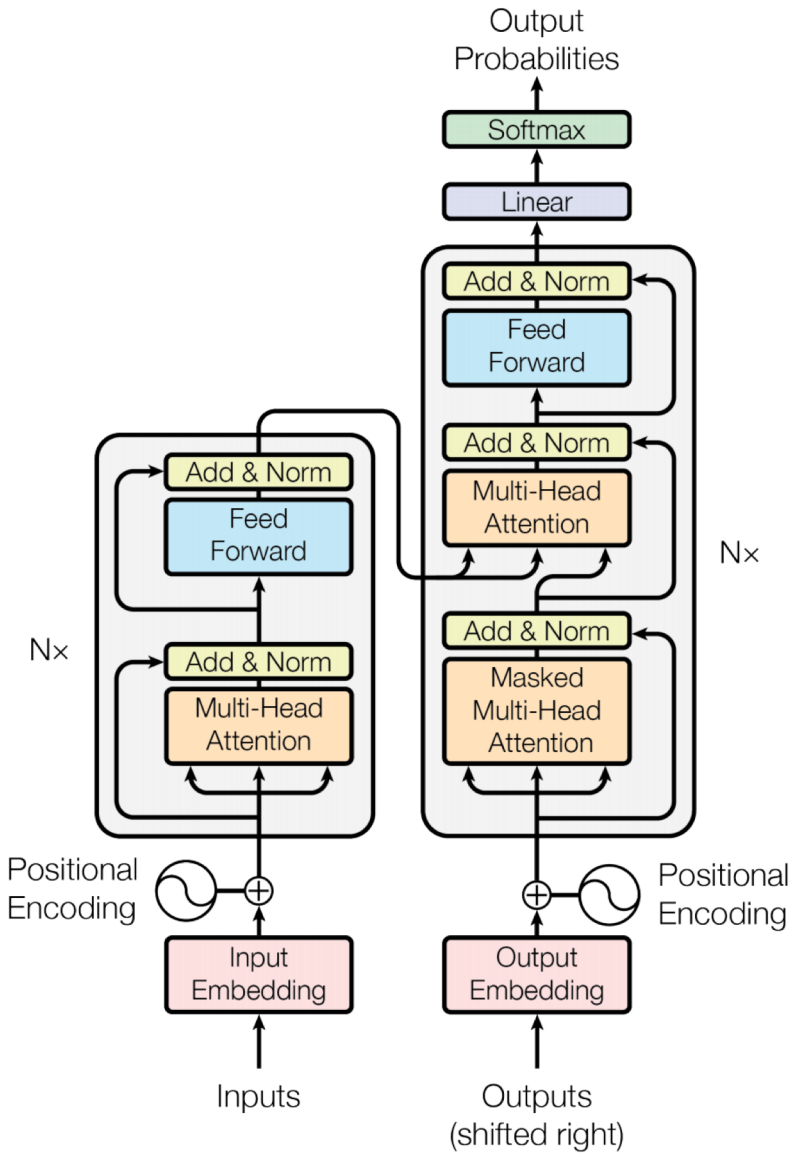
Model Architecture – Embedding Softmax

- Embedding과 softmax전 layer weight sharing



Attention Is All You Need

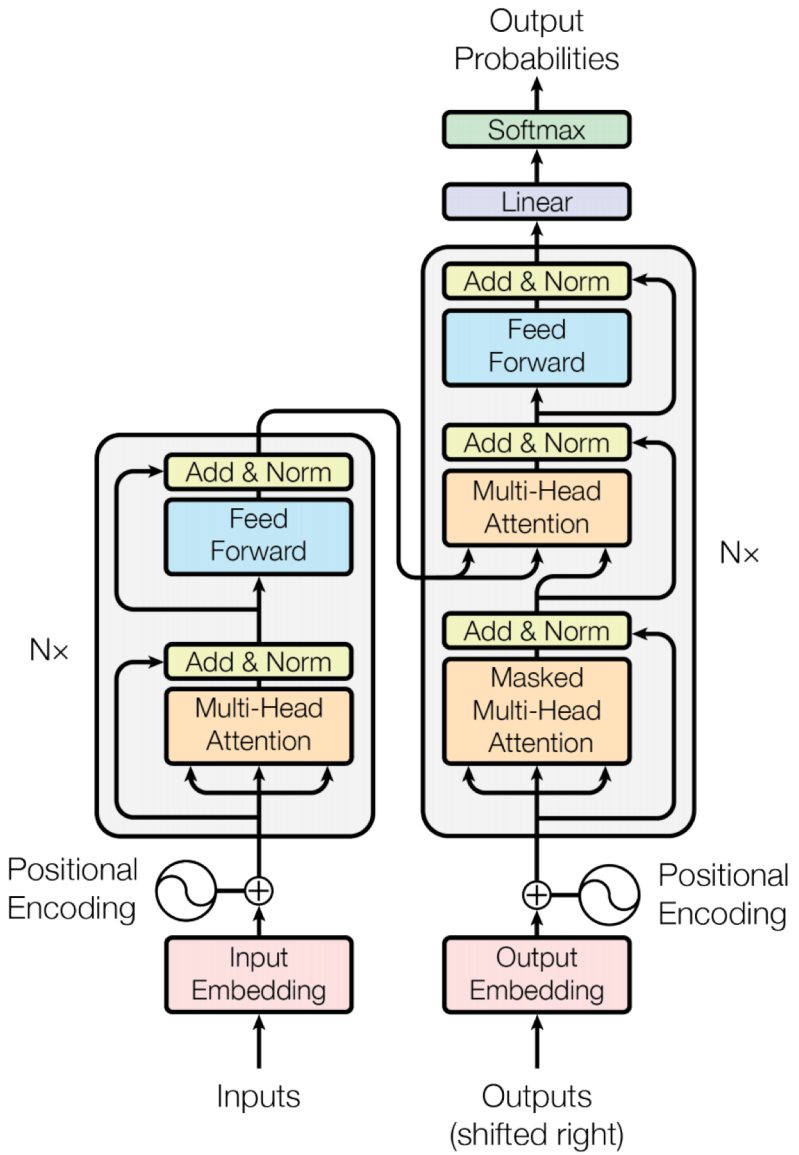
Model Architecture – Positional Encoding



Attention Is All You Need

Model Architecture – Positional Encoding

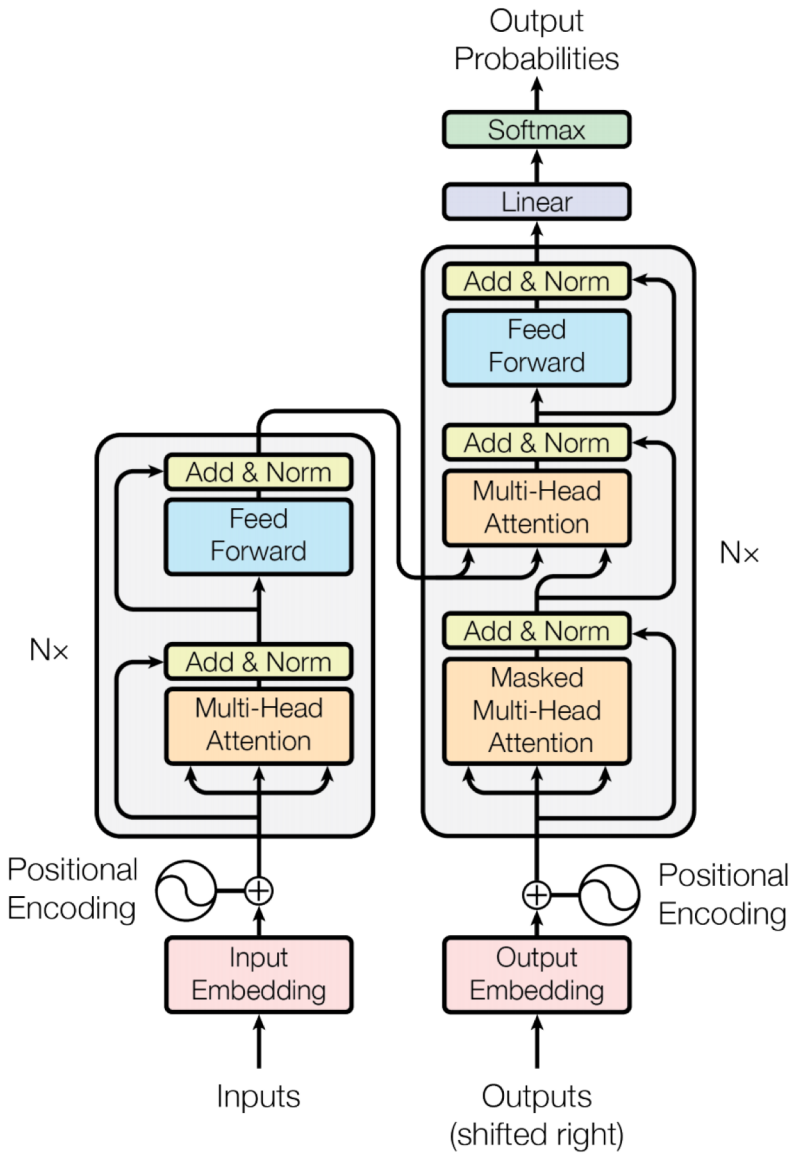
- 다양한 기법 적용가능 (learned or fixed)



Attention Is All You Need

Model Architecture – Positional Encoding

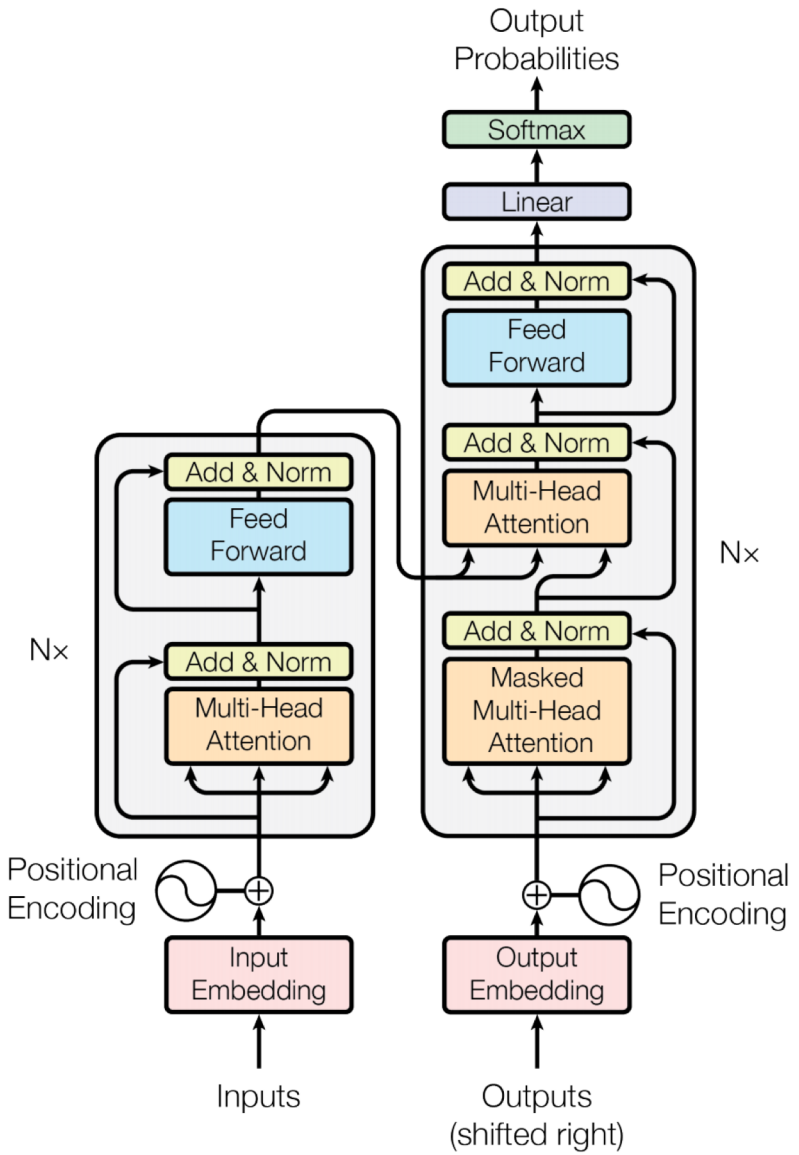
- 다양한 기법 적용가능 (learned or fixed)
- learned의 경우도 실험 해봤지만 성능차이가 별로없고,
  길이에 대응력이 떨어져 fixed sinusoid positional encoding 사용



Attention Is All You Need

Model Architecture – Positional Encoding

- 다양한 기법 적용가능 (learned or fixed)
- learned의 경우도 실험 해봤지만 성능차이가 별로없고,
  길이에 대응력이 떨어져 fixed sinusoid positional encoding 사용

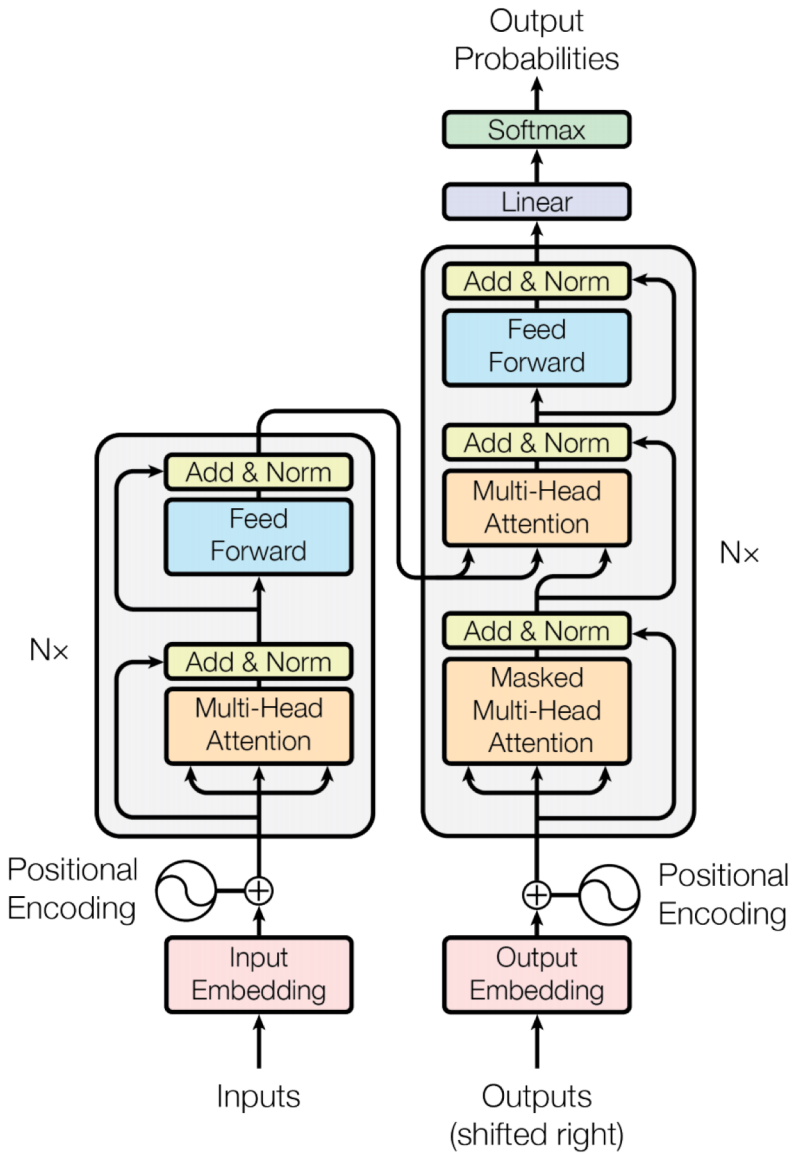


Attention Is All You Need

Model Architecture – Positional Encoding

- 다양한 기법 적용가능 (learned or fixed)
- learned의 경우도 실험 해봤지만 성능차이가 별로없고,
  길이에 대응력이 떨어져 fixed sinusoid positional encoding 사용

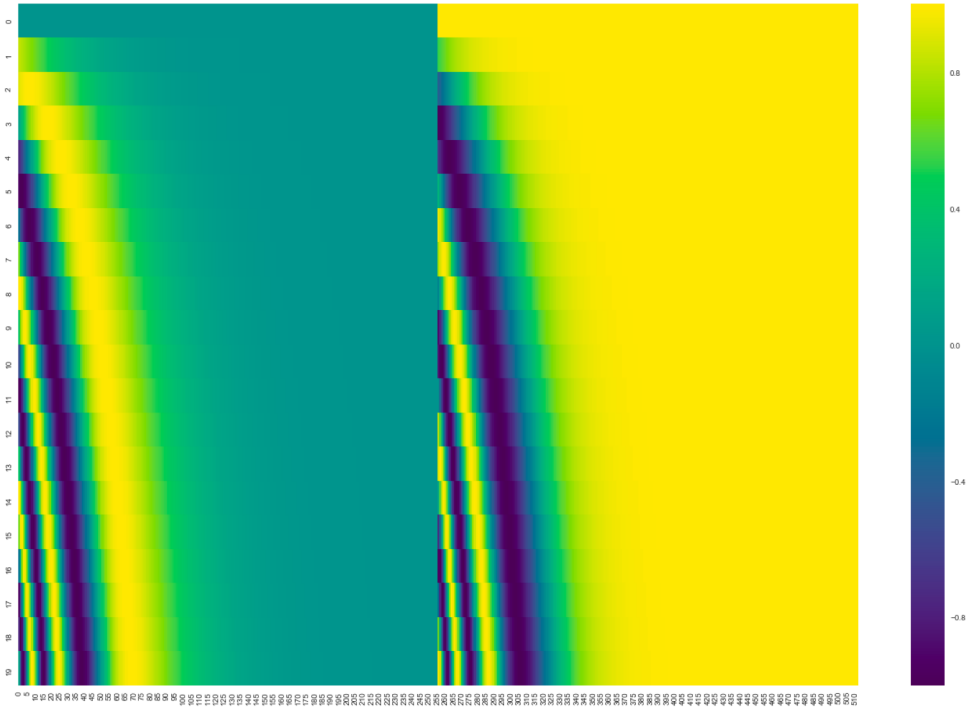
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

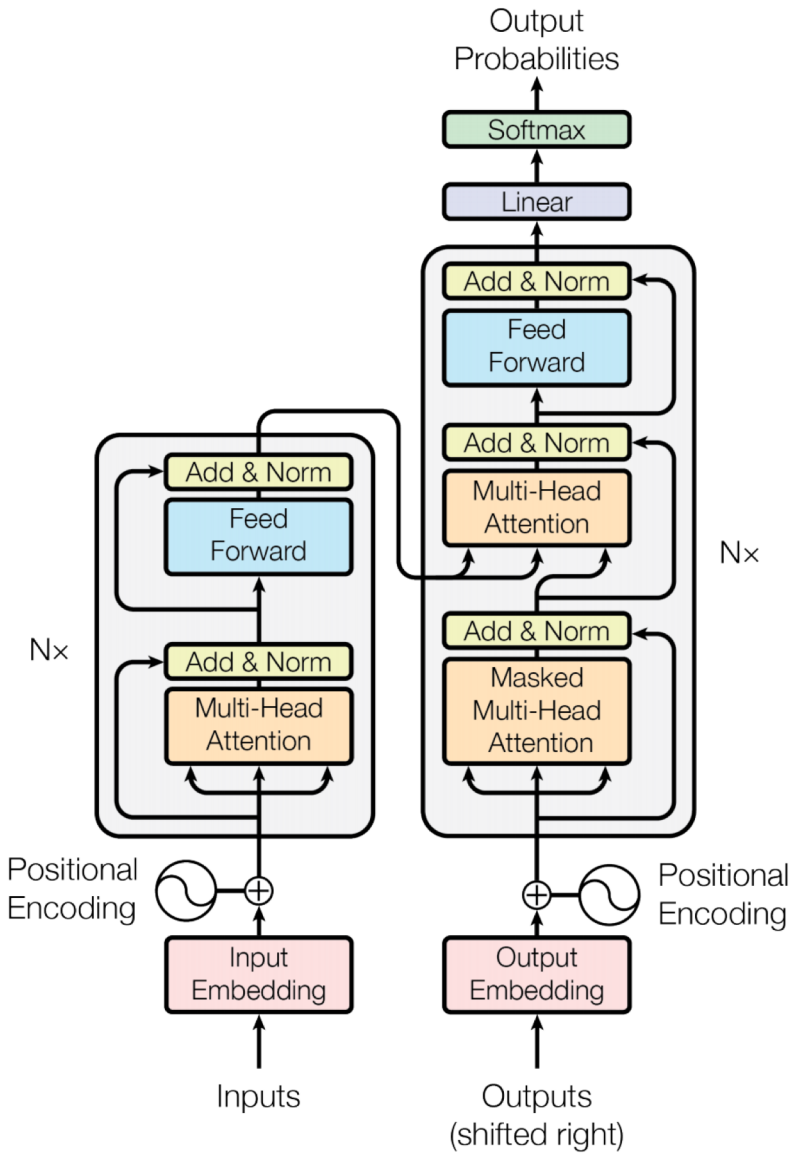


Attention Is All You Need

Model Architecture – Positional Encoding

- 다양한 기법 적용가능 (learned or fixed)
- learned의 경우도 실험 해봤지만 성능차이가 별로없고,
  길이에 대응력이 떨어져 fixed sinusoid positional encoding 사용





Attention Is All You Need

Why Self-Attention

Attention Is All You Need

Why Self-Attention

- Total Computational Complexity per Layer

Attention Is All You Need

Why Self-Attention

- Total Computational Complexity per Layer
- Amount of Computation that can be parallelized

Attention Is All You Need

Why Self-Attention

- Total Computational Complexity per Layer
- Amount of Computation that can be parallelized
- Path length between long-range dependencies in the network

Attention Is All You Need

Why Self-Attention

- Total Computational Complexity per Layer
- Amount of Computation that can be parallelized
- Path length between long-range dependencies in the network

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Attention Is All You Need

Training

Attention Is All You Need

Training

- Training Data and Batching
- Hardware and Schedule
- Optimizer
- Regularization

Attention Is All You Need

Training - Training Data and Batching

Attention Is All You Need

Training - Training Data and Batching

- WMT 2014 English-German dataset
- WMT 2014 English-French dataset

Attention Is All You Need

Training - Hardware and Schedule

Attention Is All You Need

Training - Hardware and Schedule

- Train with 8 NVIDIA P100 GPUs
- Base model: 100,000 steps, 12hours (0.4 second/step)
- Big model: 300,000 steps, 3.5days (1.0 second/step)

Attention Is All You Need

Training - Optimizer

Attention Is All You Need

Training - Optimizer

- Use Adam optimizer with $\beta_1 = 0.9,\ \beta_2 = 0.98,\ \epsilon = 10^{-9}$

Attention Is All You Need

Training - Optimizer

- Use Adam optimizer with $\beta_1 = 0.9,\ \beta_2 = 0.98,\ \epsilon = 10^{-9}$
- Varied learning rate over course of training

Attention Is All You Need

Training - Optimizer

- Use Adam optimizer with $\beta_1 = 0.9,\ \beta_2 = 0.98,\ \epsilon = 10^{-9}$
- Varied learning rate over course of training

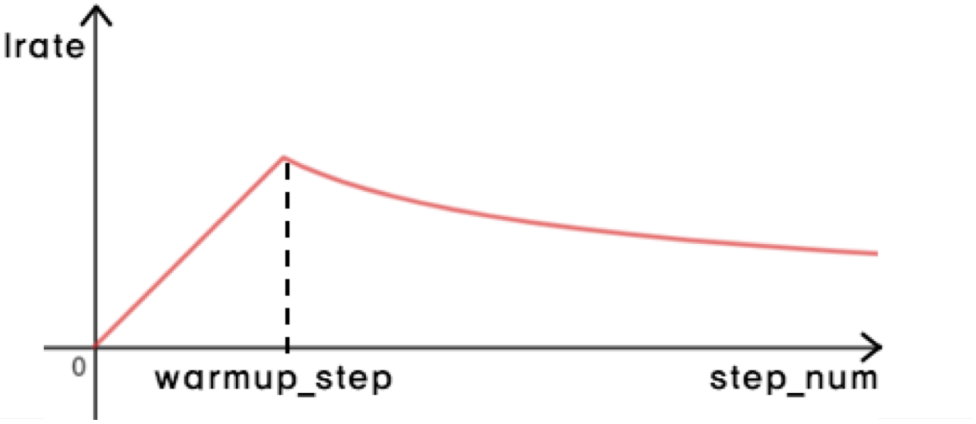
$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

Attention Is All You Need

Training - Optimizer

- Use Adam optimizer with $\beta_1 = 0.9,\ \beta_2 = 0.98,\ \epsilon = 10^{-9}$
- Varied learning rate over course of training

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$



Attention Is All You Need

Training - Regularization

Attention Is All You Need

Training - Regularization

- Residual Dropout
- Label Smoothing