구루의 생성형 언어모델 app 만들기

쏘칼 러닝 컨텐츠

구루의 생성형 언어모델 app 만들기

구루 (러닝 크리에이터) 소개

이름: 이태교

업무: 거대 언어모델(LLM)의 응용

Github: https://github.com/Taekyo-Lee

Blog: https://blog.naver.com/jetlee18

구루의 생성형 언어모델 app 만들기

구루 (러닝 크리에이터) 소개

이름: 이태교

업무: 거대 언어모델(LLM)의 응용

Github: https://github.com/Taekyo-Lee

Blog: https://blog.naver.com/jetlee18

컨텐츠 제작 계획

목표: 오픈소스 생성형 언어모델로 나만의 챗봇 만들기

이론 (수식 X)

application 파이프라인 구축

GUI

구루의 생성형 언어모델 app 만들기

구루 (러닝 크리에이터) 소개

이름: 이태교

업무: 거대 언어모델(LLM)의 응용

Github: https://github.com/Taekyo-Lee

Blog: https://blog.naver.com/jetlee18

컨텐츠 제작 계획

목표: 오픈소스 생성형 언어모델로 나만의 챗봇 만들기

이론 (수식 X)

application 파이프라인 구축

GUI

목표로 하는 청중은?

딥러닝의 원리에 대해서 조금은 알고 있으신 분 (backpropagation 등)

약간의 Python 지식이 있으신 분

구루의 생성형 언어모델 app 만들기

컨텐츠 기획

기획	내용
목차	• 왜 Transformer는 성능이 좋을까? (이론) • Query, key, value? Scaled dot-product attention의 motivation 및 원리 (이론) • 생성형 언어모델의 'knowledge cutoff' 문제를 어떻게 해결해야 할까? (이론) • Python으로 구현하는 PAL (program-aided language model) application (실습) • Instruction fine-tuning 이란? (이론) • Parameter-efficient fine-tuning (PEFT)과 LoRA (Low-rank adaptation) (이론) • PyTorch로 구현하는 생성형 언어모델 (Llama2)의 fine-tuning (with single GPU) (실습) • Retrival agmented generation (RAG)의 개념 (이론) • Python으로 구현하는 RAG chatbot (실습)
업로드 주기	2주
포맷	동영상
타겟	자신만의 데이터로 생성형 언어모델 app을 만들고자 하는 사람

구루의 생성형 언어모델 app 만들기

컨텐츠 기획

기획	내용
목차	• 왜 Transformer는 성능이 좋을까? (이론) • Query, key, value? Scaled dot-product attention의 motivation 및 원리 (이론) • 생성형 언어모델의 'knowledge cutoff' 문제를 어떻게 해결해야 할까? (이론) • Python으로 구현하는 PAL (program-aided language model) application (실습) • Instruction fine-tuning 이란? (이론) • Parameter-efficient fine-tuning (PEFT)과 LoRA (Low-rank adaptation) (이론) • PyTorch로 구현하는 생성형 언어모델 (Llama2)의 fine-tuning (with single GPU) (실습) • Retrival agmented generation (RAG)의 개념 (이론) • Python으로 구현하는 RAG chatbot (실습)
업로드 주기	2주
포맷	동영상
타겟	자신만의 데이터로 생성형 언어모델 app을 만들고자 하는 사람

Let's get started!!!!!

1회: 왜 transformer는 성능이 좋을까?

GPT

CTRL

T5

BART

BERT

Transformer

GPT

Generative pre-trained transformer

CTRL

Conditional transformer language model

T5

Text-to-text transfer transformer

BART

Bidirectional autoregressive transformer

BERT

Bidirectional encoder representations from transformer

NLP WORLD
SINCE 2017

Image Credit: https://psychology.iresearchnet.com/wp-content/uploads/2016/01/Social-Dominance-Orientation.jpg

Attention Is All You Need

Ashish Vaswani*	**Noam Shazeer***	**Niki Parmar***	**Jakob Uszkoreit***
Google Brain	Google Brain	Google Research	Google Research
avaswani@google.com	noam@google.com	nikip@google.com	usz@google.com

Llion Jones*	**Aidan N. Gomez* †**	**Łukasz Kaiser***
Google Research	University of Toronto	Google Brain
llion@google.com	aidan@cs.toronto.edu	lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

모든 것은 2017년 이 논문으로 부터 시작되었다…

Ashish Vaswani

Image credit: https://twitter.com/ashvaswani

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

모든 것은 2017년 이 논문으로 부터 시작되었다…

Ashish Vaswani

Image credit: https://twitter.com/ashvaswani

Attention :

중요한 곳에 더 집중하고,

덜 중요한 곳에 덜 집중한다

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

모든 것은 2017년 이 논문으로 부터 시작되었다…

Ashish Vaswani

Image credit: https://twitter.com/ashVaswani

Attention :

중요한 곳에 더 집중하고,

덜 중요한 곳에 덜 집중한다

I love apple so I expect this fall when they release the new iphone.

'apple'의 의미를 알기 위해 어느 단어에 집중해야 할까?

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

모든 것은 2017년 이 논문으로 부터 시작되었다···

Ashish Vaswani

Image credit: https://twitter.com/ashvaswani

Attention :
중요한 곳에 더 집중하고,
덜 중요한 곳에 덜 집중한다

I love apple so I expect this fall when they release the new iphone.

'apple'의 의미를 알기 위해 어느 단어에 집중해야 할까?

context-aware!!


MARVEL STUDIOS
CAPTAIN AMERICA
THE FIRST AVENGER

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.

2014년, 처음으로 NLP에 attention 매커니즘 도입.

(RNN + attention)

MARVEL STUDIOS

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

THE FIRST attention

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.

2014년, 처음으로 NLP에 attention 매커니즘 도입.
(RNN + attention)

Convolutional Sequence to Sequence Learning

Jonas Gehring
Michael Auli
David Grangier
Denis Yarats
Yann N. Dauphin
Facebook AI Research

Abstract

The prevalent approach to sequence to sequence learning maps an input sequence to a variable length output sequence via recurrent neural networks. We introduce an architecture based entirely on convolutional neural networks.[1] Compared to recurrent models, computations over all elements can be fully parallelized during training to better exploit the GPU hardware and optimization is easier since the number of non-linearities is fixed and independent of the input length. Our use of gated linear units eases gradient propagation and we equip each decoder layer with a separate attention module. We outperform the accu-

Convolutional neural networks are less common for sequence modeling, despite several advantages (Waibel et al., 1989; LeCun & Bengio, 1995). Compared to recurrent layers, convolutions create representations for fixed size contexts, however, the effective context size of the network can easily be made larger by stacking several layers on top of each other. This allows to precisely control the maximum length of dependencies to be modeled. Convolutional networks do not depend on the computations of the previous time step and therefore allow parallelization over every element in a sequence. This contrasts with RNNs which maintain a hidden state of the entire past that prevents parallel computation within a sequence.

Multi-layer convolutional neural networks create hierarchi-

이후 여러 연구자들이 attention을 활용한 NLP 알고리즘을 개발
(Convolution + attention)

MARVEL STUDIOS

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

THE FIRST attention

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.

2014년, 처음으로 NLP에 attention 매커니즘 도입.
(RNN + attention)

Convolutional Sequence to Sequence Learning

Jonas Gehring
Michael Auli
David Grangier
Denis Yarats
Yann N. Dauphin
Facebook AI Research

Abstract

The prevalent approach to sequence to sequence learning maps an input sequence to a variable length output sequence via recurrent neural networks. We introduce an architecture based entirely on convolutional neural networks.[1] Compared to recurrent models, computations over all elements can be fully parallelized during training to better exploit the GPU hardware and optimization is easier since the number of non-linearities is fixed and independent of the input length. Our use of gated linear units eases gradient propagation and we equip each decoder layer with a separate attention module. We outperform the accu-

Convolutional neural networks are less common for sequence modeling, despite several advantages (Waibel et al., 1989; LeCun & Bengio, 1995). Compared to recurrent layers, convolutions create representations for fixed size contexts, however, the effective context size of the network can easily be made larger by stacking several layers on top of each other. This allows to precisely control the maximum length of dependencies to be modeled. Convolutional networks do not depend on the computations of the previous time step and therefore allow parallelization over every element in a sequence. This contrasts with RNNs which maintain a hidden state of the entire past that prevents parallel computation within a sequence.

Multi-layer convolutional neural networks create hierarchi-

이후 여러 연구자들이 attention을 활용한 NLP 알고리즘을 개발
(Convolution + attention)

Before transformer: context 파악을 위해 기존의 아키텍쳐(RNN, convolution 등)에 attention 매커니즘을 결합

With transformers, Attention Is All You Need for context-understanding

제목의 의미는 이것!!

(~~RNN~~ + attention)

(~~convolution~~ + attention)

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.

Transformer: context를 파악하기 위해 오직 attention 만을 필요로 함

"Transformers have completely taken NLP world by storm." by Andrew Ng in 2018

Image credit: https://www.andrewng.org/

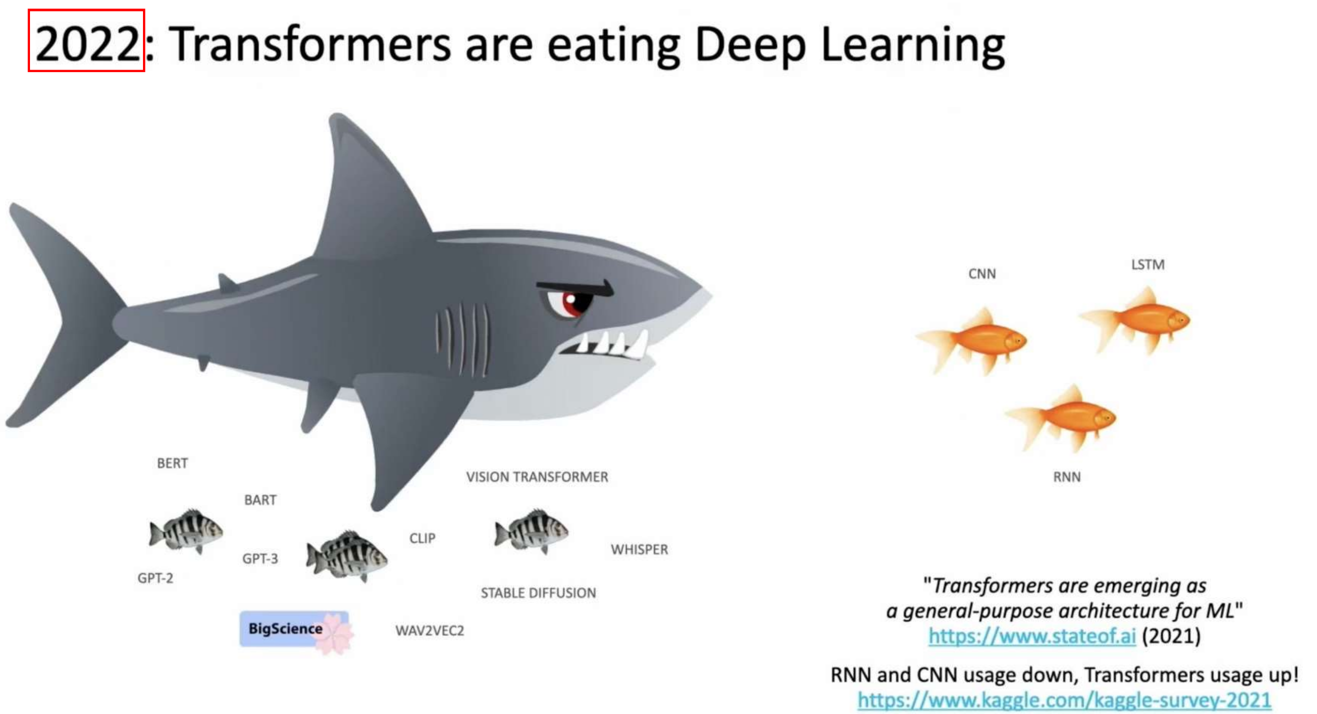


Image Credit: Richard MacManus

이쯤에서 본능적 궁금증. . .

왜 transformer는 성능이 좋은거야???

저자들은 그냥 여러 아키텍처 이것저것 시도 해보다가 우연히 성능 좋은 것 발견해서 논문 쓴 거야???

우리는 그냥 성능 좋다는 결과만 알면 되고 그 이유는 이해할 수 없어???

이쯤에서 본능적 궁금증. . .

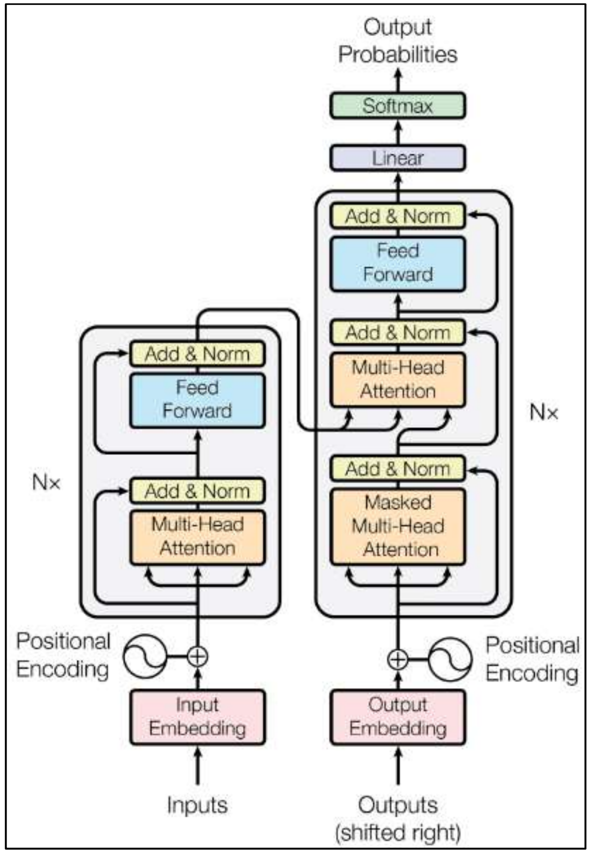
왜 transformer는 성능이 좋은거야???

저자들은 그냥 여러 아키텍처 이것저것 시도 해보다가 우연히 성능 좋은 것 발견해서 논문 쓴 거야???

우리 같은 보통 사람들은 그냥 성능 좋다는 결과만 알면 되고 그 이유는 이해할 수 없어???

저자들이 이런 아키텍처를 생각하고 성능이 좋을 것이라 기대한 이유가 있지 않을까?

Of course!!!



Today's Goal

Transformer와 RNN의 아키텍처를 비교하고 '아~ transformer가 RNN보다 성능이 좋을 것 같다~' 라는 느낌을 받는 것

- NO 아키텍처에 대한 디테일한 설명
- NO scaled dot-product attention에 대한 수학적 설명 (motivation은 다음 컨텐츠에서...)
- RNN과 transformer의 context-retrieval 방식의 차이점에 중점

Q1 : 왜 Transformer가 RNN류 보다 뛰어난 거야?

Q1 : 왜 Transformer가 RNN류 보다 뛰어난 거야?

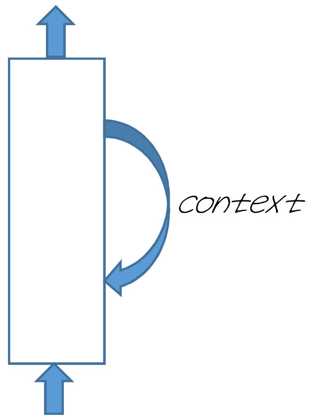
A1 : Context를 retrieval 하는 방식이 다름

Q1 : 왜 Transformer가 RNN류 보다 뛰어난 거야?

A1 : Context를 retrieval 하는 방식이 다름

RNN

아웃풋 sequence

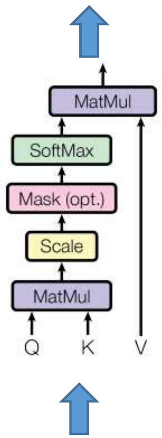


인풋 sequence

Recursive 매커니즘 이용

Transformer

context



인풋 sequence

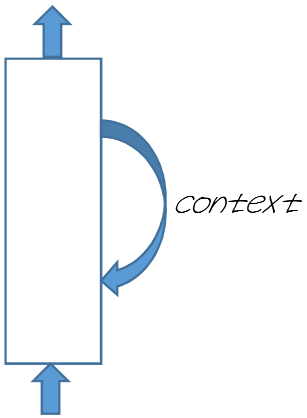
Attention 매커니즘 이용

Q1 : 왜 Transformer가 RNN류 보다 뛰어난 거야?

A1 : Context를 retrieval 하는 방식이 ~~다른~~ 우월함

RNN

아웃풋 sequence

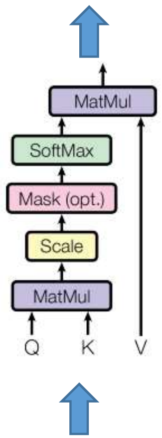


인풋 sequence

Recursive 매커니즘 이용

Transformer

context



인풋 sequence

Attention 매커니즘 이용

모든 RNN 기반 network이 가지고 있는
치명적 약점을 해결!!

temporal vanishing gradients 현상
(인풋의 길이가 길어지면 gradient가 0이 되는 현상)
을 제거함
→ long-range dependencies에 대응 가능

Input의 길이에 상관없이
context를 파악할 수 있음

안정적인 training
(local minima에 빠지지 않음)

Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

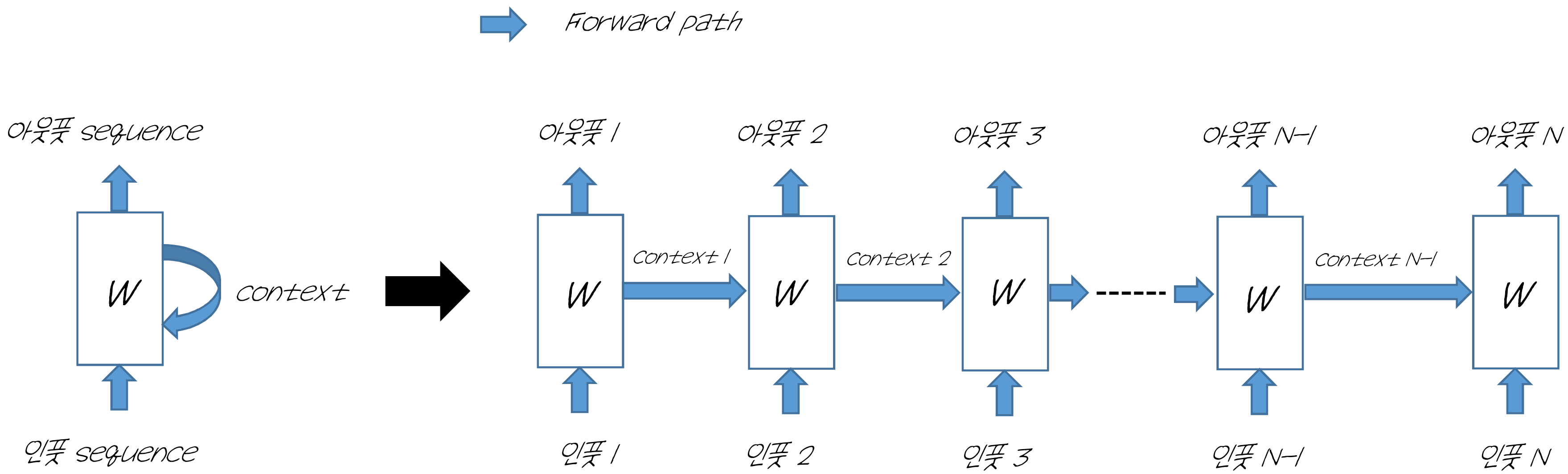
Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

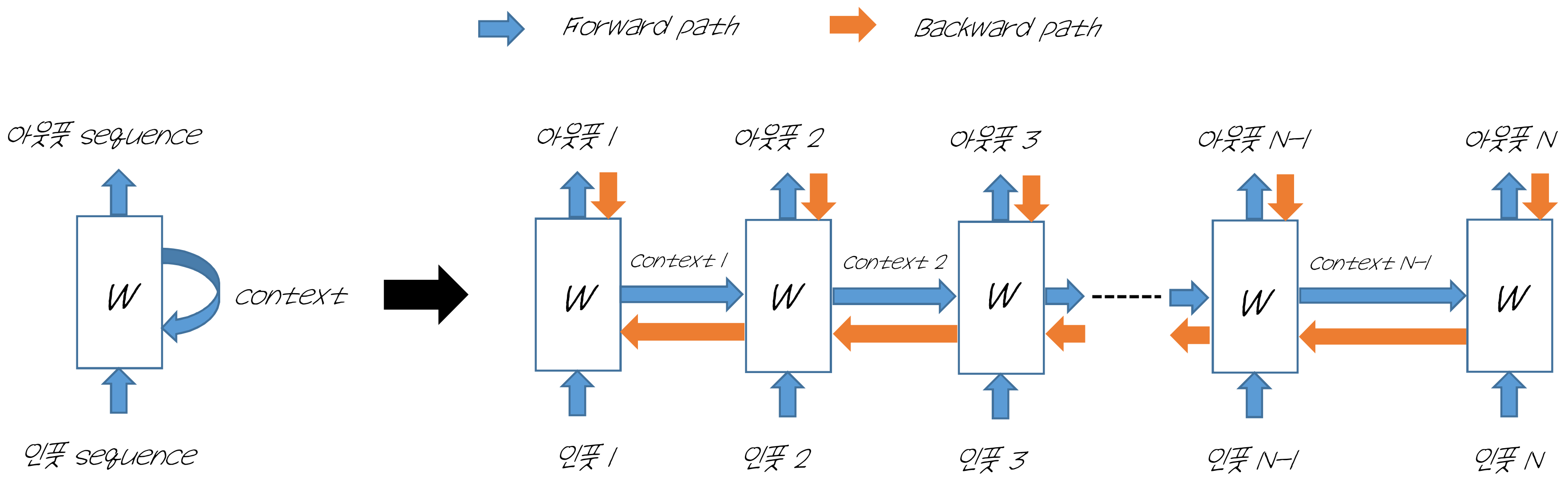
RNN 아키텍처



Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

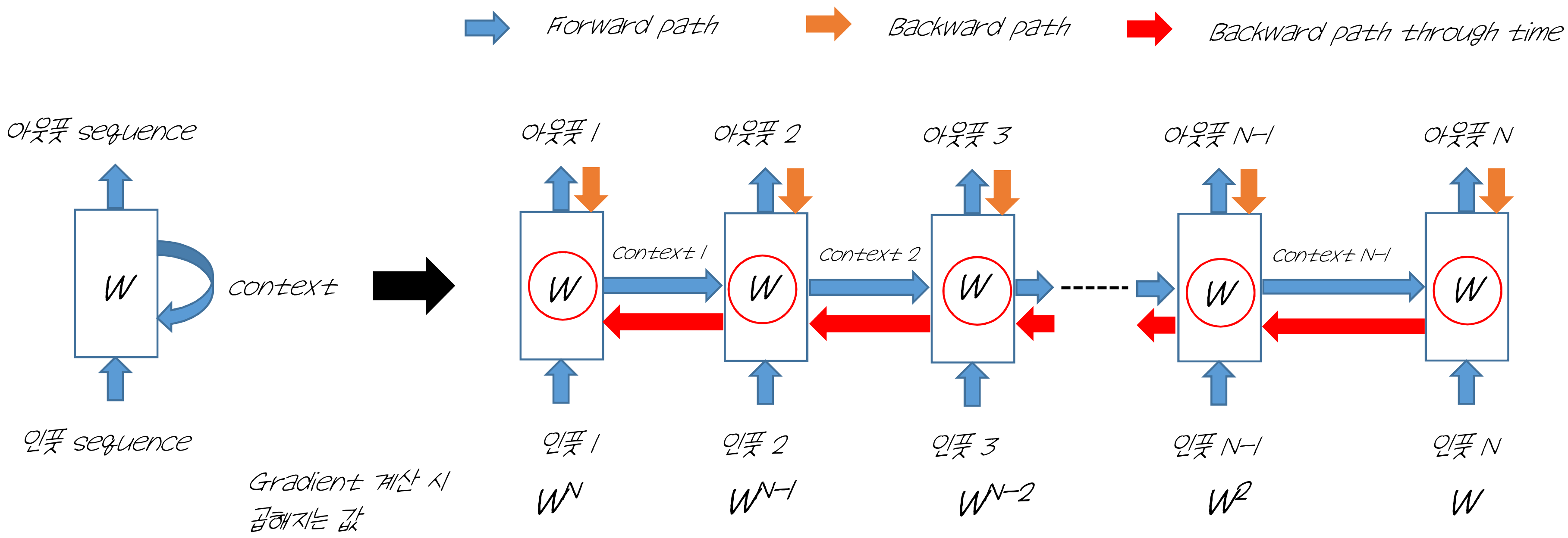
RNN 아키텍처



Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

RNN 아키텍처



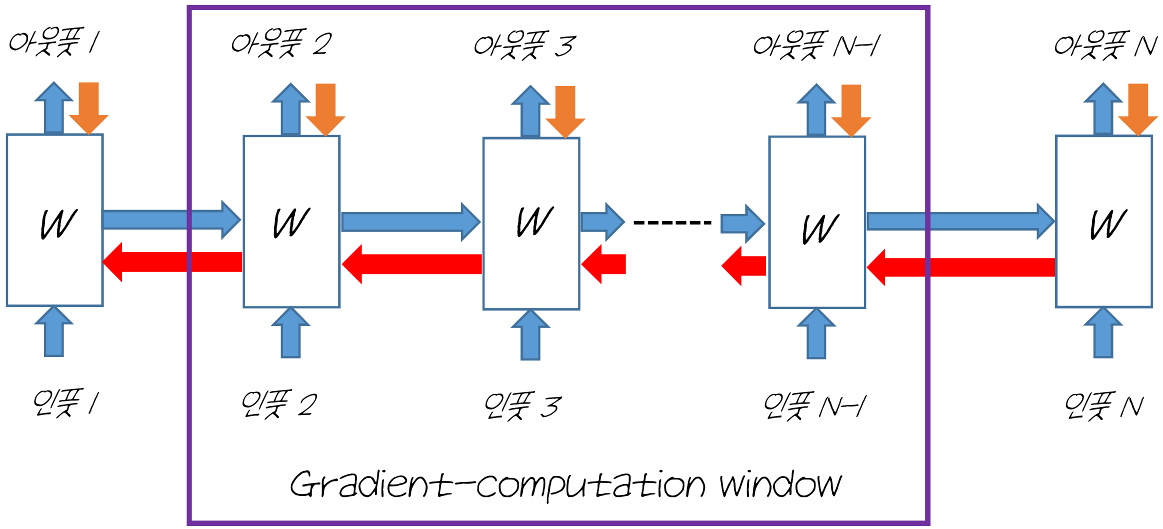
Weight 이 1보다 크면? W^N → 무한대, Weight 이 1보다 작으면? W^N → 0

Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

RNN 아키텍처의 진화 : temporal vanishing gradients를 줄여라!!

Fixed-window backpropagation

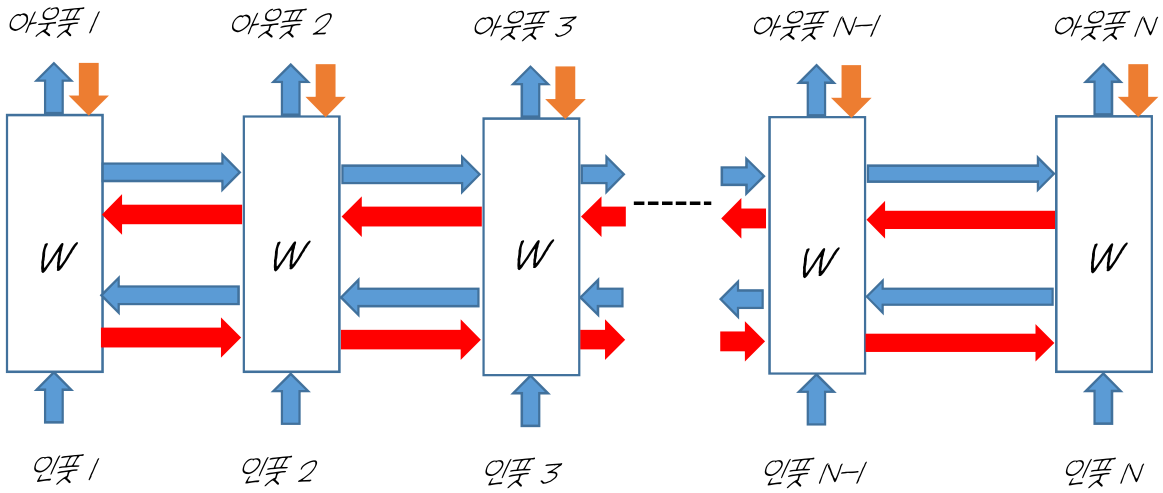


Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

RNN 아키텍처의 진화 : temporal vanishing gradients를 줄여라!!

Bidirectional RNN

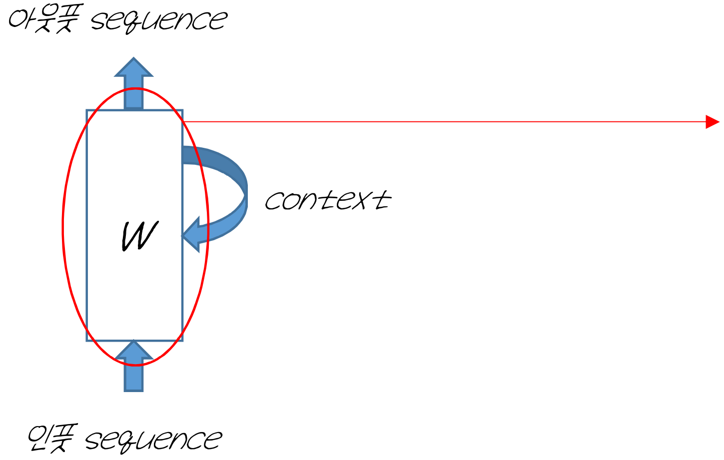


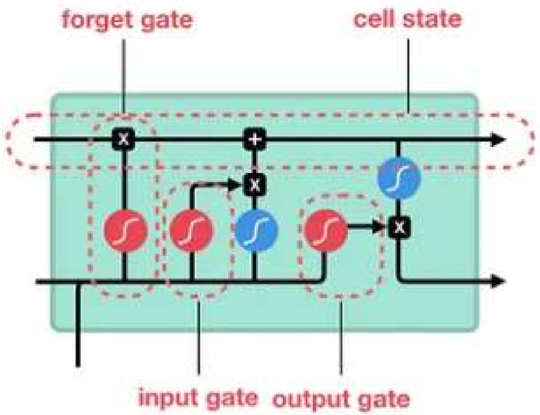
Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

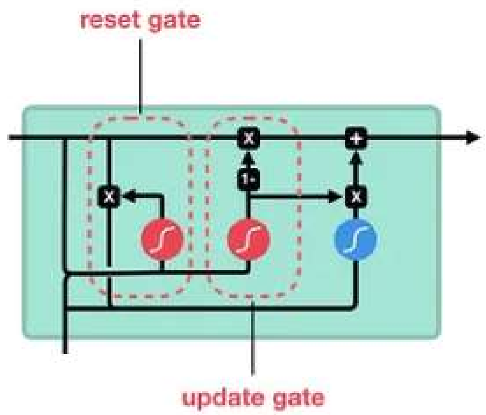
RNN 아키텍처의 진화 : temporal vanishing gradients를 줄여라!!

<u>Gradient-flow control</u>





LSTM



GRU

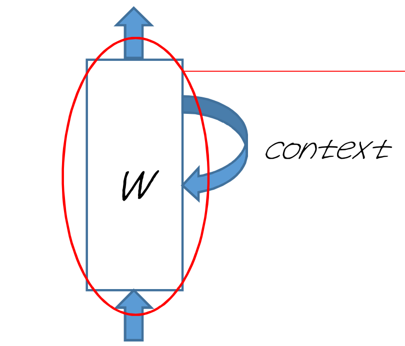
Image credit: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

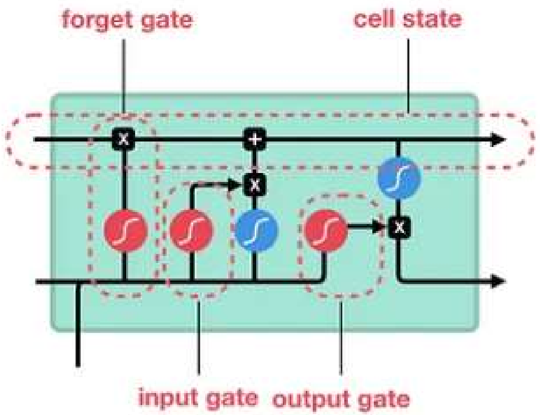
A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

RNN 아키텍처의 진화 : temporal vanishing gradients를 줄여라!!

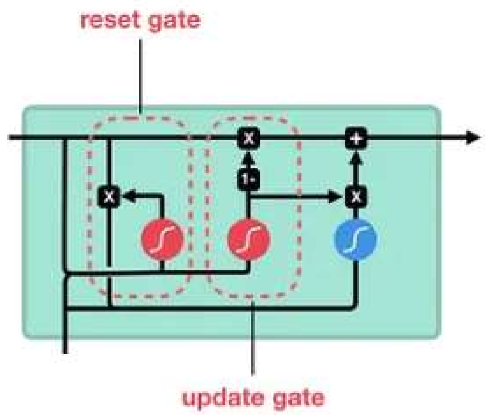
Gradient-flow control



Gate 개념을 도입하여 매 스텝마다 gradient를 적절하게 유지 → vanishing gradients를 획기적으로 개선



LSTM



GRU

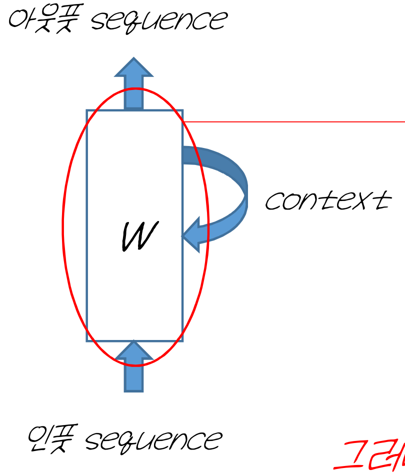
Image credit: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Q 2 : 왜 RNN류는 temporal vanishing gradients 현상이 나타나는 거야?

A 2 : Recursive 구조로 인해 동일한 weight이 누적되어 곱해지기 때문에

RNN 아키텍처의 진화 : temporal vanishing gradients를 줄여라!!

Gradient-flow control



Gate 개념을 도입하여 매 스텝마다 gradient를 적절하게 유지 → vanishing gradients를 획기적으로 개선

그러나 길이 N이 길어지면 답이 없다

d-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

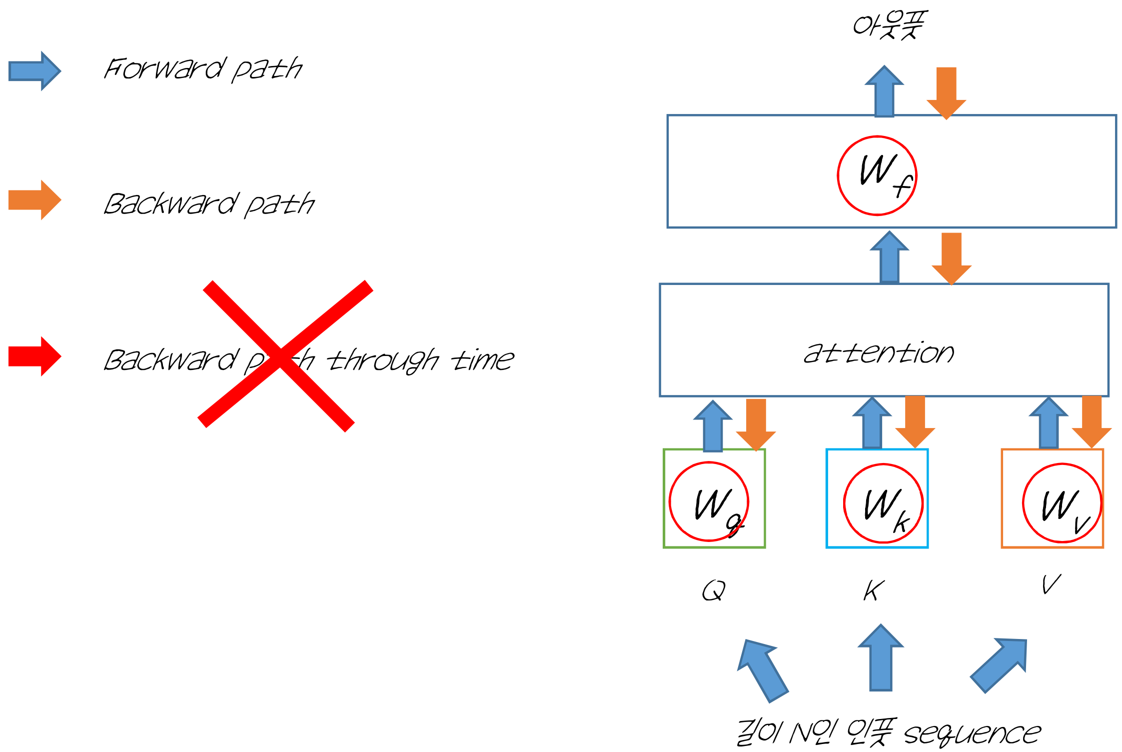
Q 3 : 그럼 왜 transformer는 temporal vanishing gradients 현상이 없는 거야?

Q 3 : 그럼 왜 transformer는 temporal vanishing gradients 현상이 없는 거야?

A 3 : Backpropagation 시 곱해지는 *weight*들이 다 다르기 때문에

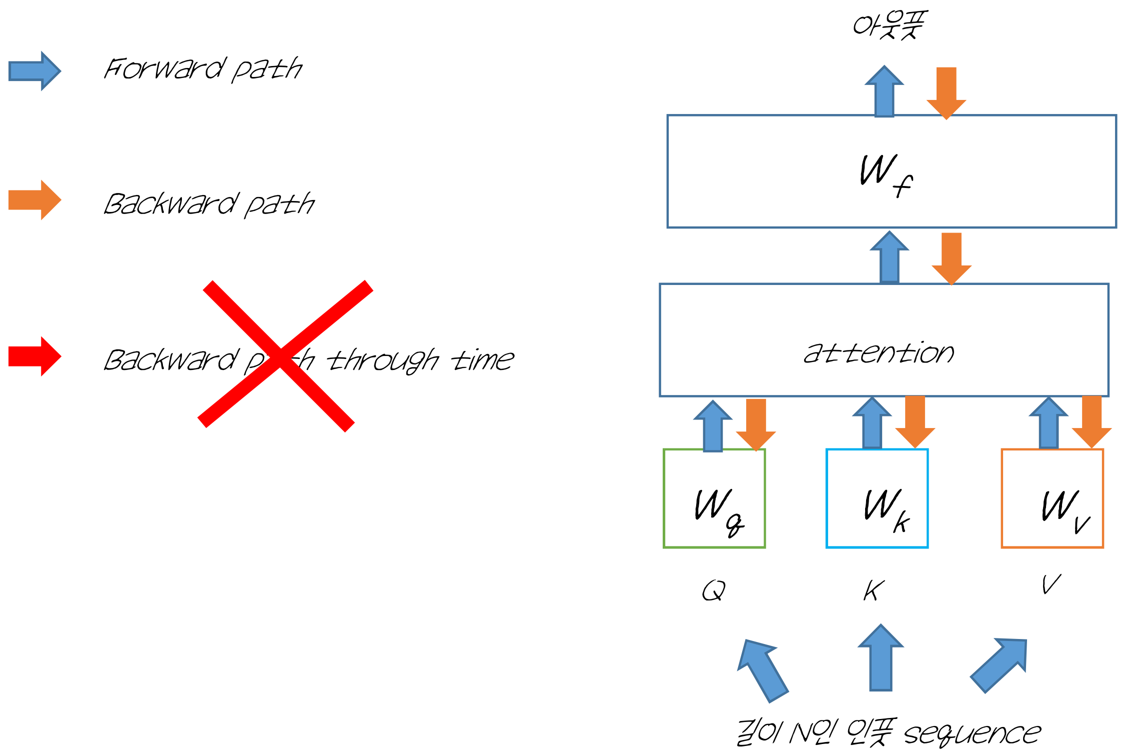
Q 3 : 그럼 왜 transformer는 temporal vanishing gradients 현상이 없는 거야?

A 3 : Backpropagation 시 곱해지는 weight들이 다 다르기 때문에

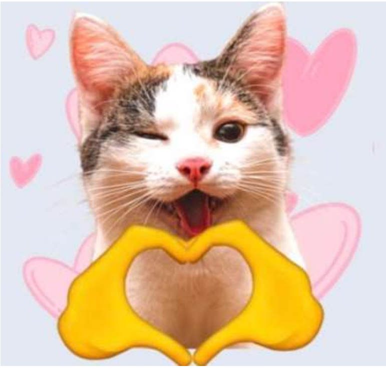


Q 3 : 그럼 왜 transformer는 temporal vanishing gradients 현상이 없는 거야?

A 3 : Backpropagation 시 곱해지는 weight들이 다 다르기 때문에



메모리가 허락하는 한 인풋의 길이 N을 임의로 길게 해도 (temporal) vanishing gradients 현상 없음



Q 4 : Recursive 구조를 포기했다면 어순에 대한 정보도 포기한거 아냐?

Q 4 : Recursive 구조를 포기했다면 어순에 대한 정보도 포기한거 아냐?

A 4 : Positional 정보를 첨가해 주지!!

Q 4 : Recursive 구조를 포기했다면 어순에 대한 정보도 포기한거 아냐?

A 4 : Positional 정보를 첨가해 주지!!

오리지널 sequence: Transformers have completely taken NLP world by storm, which have made LLM possible.

오리지널 sequence:	Transformers	have	completely	taken	NLP	world	by	storm,	which	have	made	LLM	possible.
+													
positional sequence:	1	2	3	4	5	6	7	8	9	10	11	12	13

Q 4 : Recursive 구조를 포기했다면 어순에 대한 정보도 포기한거 아냐?

A 4 : Positional 정보를 첨가해 주지!!

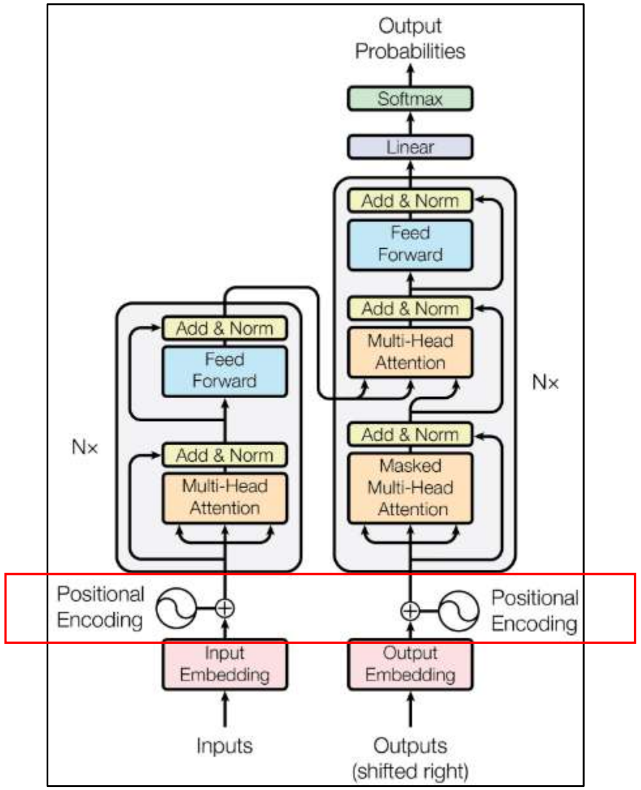
오리지널 sequence: Transformers have completely taken NLP world by storm, which have made LLM possible.

오리지널 sequence:	Transformers	have	completely	taken	NLP	world	by	storm,	which	have	made	LLM	possible.
							+						
positional sequence:	1	2	3	4	5	6	7	8	9	10	11	12	13
							=						
인풋 sequence:	Transformers	have	completely	taken	NLP	world	by	storm,	which	have	made	LLM	possible

Q 4 : Recursive 구조를 포기했다면 어순에 대한 정보도 포기한거 아냐?

A 4 : Positional 정보를 첨가해 주지!!

실제 transformer 아키텍처의 positional 정보 첨가하는 부분



Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

모든 RNN 기반 network이 가지고 있는
치명적 약점을 해결!!

temporal vanishing gradients 현상
(인풋의 길이가 길어지면 gradient가 0이 되는 현상)
을 제거함
→ long-range dependencies에 대응 가능

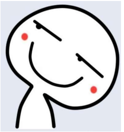
Input의 길이에 상관없이
context를 파악할 수 있음

안정적인 training
(local minima에 빠지지 않음)

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : *Input의 길이에 상관없이 context를 잘 파악할 수 있음*

I love apple so I expect this fall when they release the new iphone.



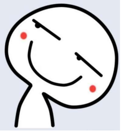
이 문장은 짧은 문장이지만
굉장히 긴 문장이라고 가정합시다^^

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음

이 'apple' 의 의미를 파악하기 위해선
문장의 어떤 단어에 집중해야 할까?

I love apple so I expect this fall when they release the new iphone.



이 문장은 짧은 문장이지만
굉장히 긴 문장이라고 가정합시다^^

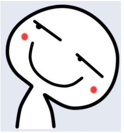
당연히 바로 이 'iphone' 일 것입니다.

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음

이 'apple' 의 의미를 파악하기 위해선
문장의 어떤 단어에 집중해야 할까?

I love apple so I expect this fall when they release the new iphone.



이 문장은 짧은 문장이지만
굉장히 긴 문장이라고 가정합시다^^

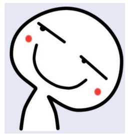
당연히 바로 이 'iphone' 일 것입니다.

그러나 RNN류에서는 'apple'과 'iphone' 의 거리가
너무 멀기 때문에 temporal vanishing
gradients가 발생하여 'apple'과 'iphone'을 연관
시키지 못하는 것입니다.

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : *Input의 길이에 상관없이 context를 잘 파악할 수 있음*

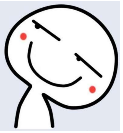
I love apple so I expect this fall when they release the new iphone.



이 문장은 짧은 문장이지만
굉장히 긴 문장이라고 가정합시다^^

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음

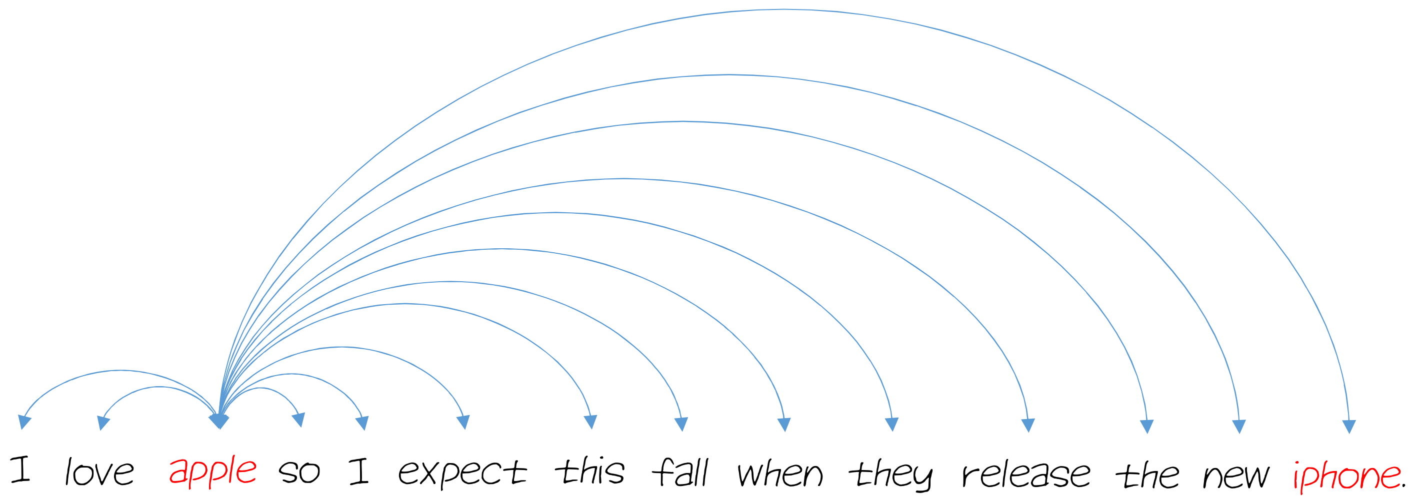


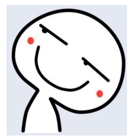
이 문장은 짧은 문장이지만
굉장히 긴 문장이라고 가정합시다^^

I love apple so I expect this fall when they release the new iphone.

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음



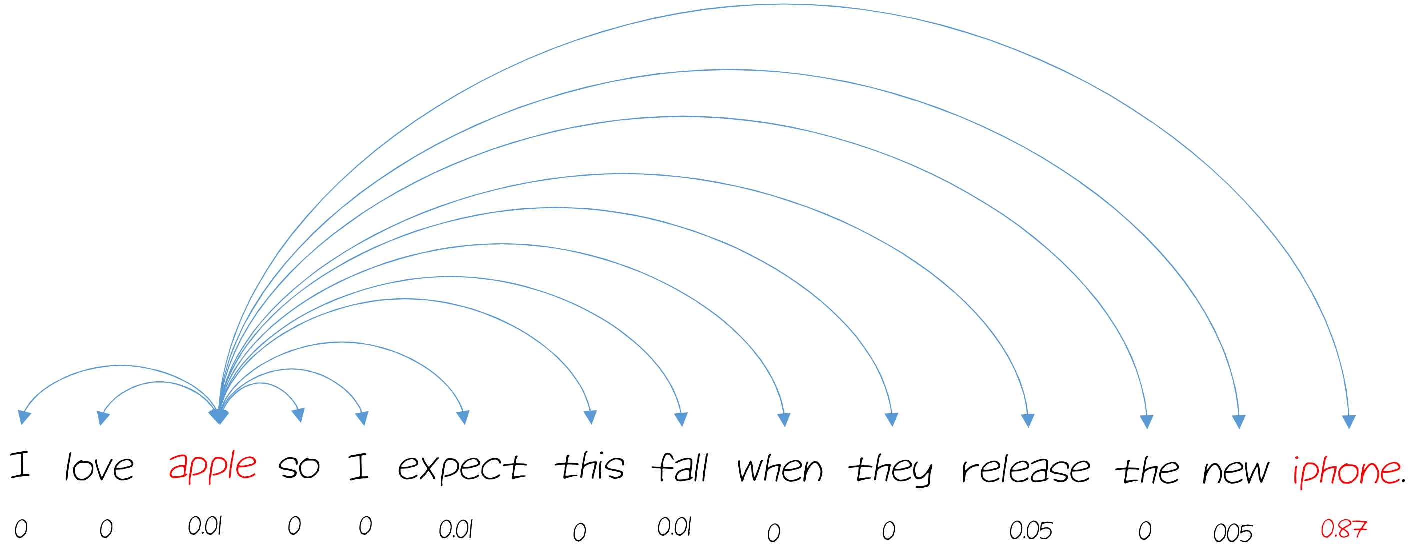


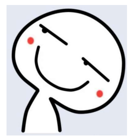
이 문장은 짧은 문장이지만
굉장히 긴 문장이라고 가정합시다^^

Attention 메커니즘 작동 → 거리에 상관없이 모든 단어들에 대하여 집중해야 하는 가중치(attention score)를 동시에 계산

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음



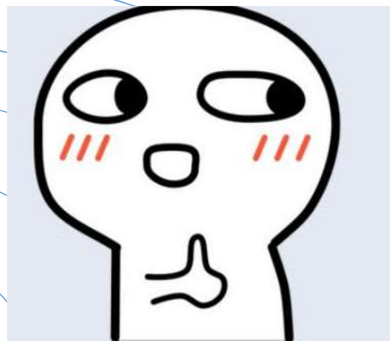


이 문장은 짧은 문장이지만
굉장히 긴 문장이라고 가정합시다^^

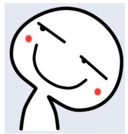
Attention 메커니즘 작동 → 거리에 상관없이 모든 단어들에 대하여 집중해야 하는 가중치(attention score)를 동시에 계산

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음



아하! iphone 이라는 단어에 가장 집중해야 하는구나!!!!



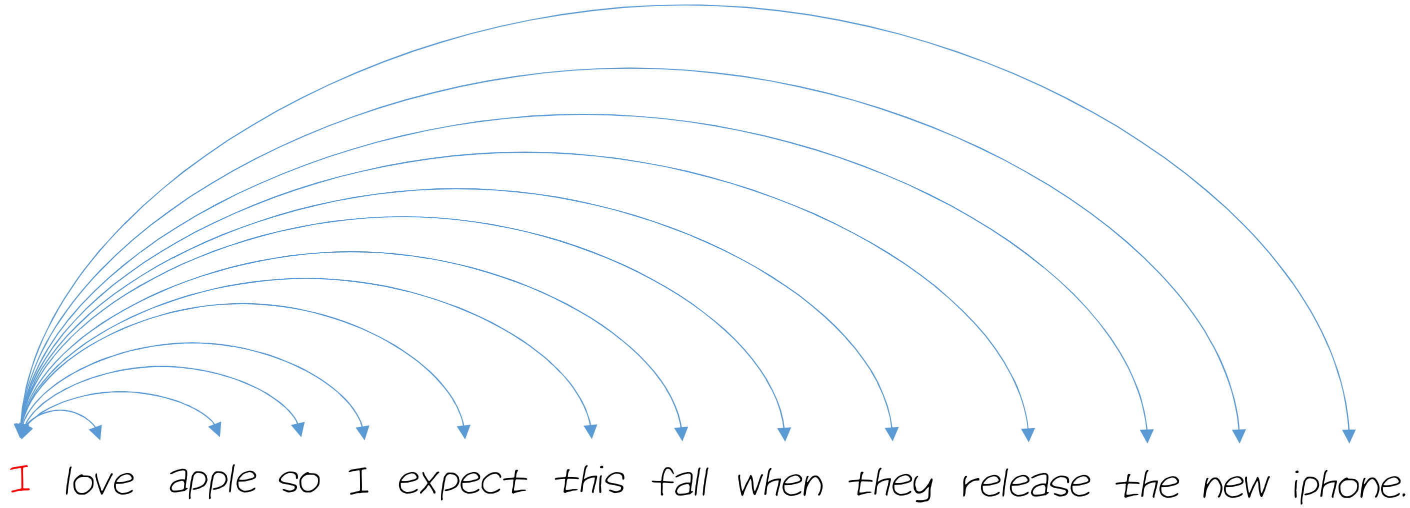
이 문장은 짧은 문장이지만 굉장히 긴 문장이라고 가정합시다^^

I	love	apple	so	I	expect	this	fall	when	they	release	the	new	iphone.
0	0	0.01	0	0	0.01	0	0.01	0	0	0.05	0	0.05	0.87

Attention 메커니즘 작동 → 거리에 상관없이 모든 단어들에 대하여 집중해야 하는 가중치(attention score)를 동시에 계산

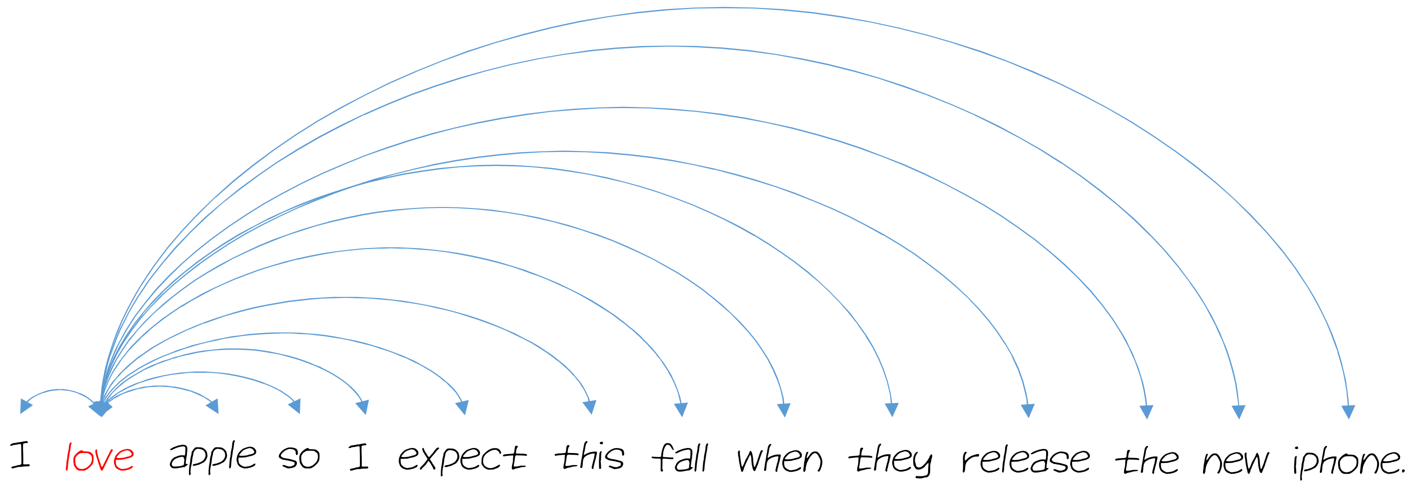
Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음



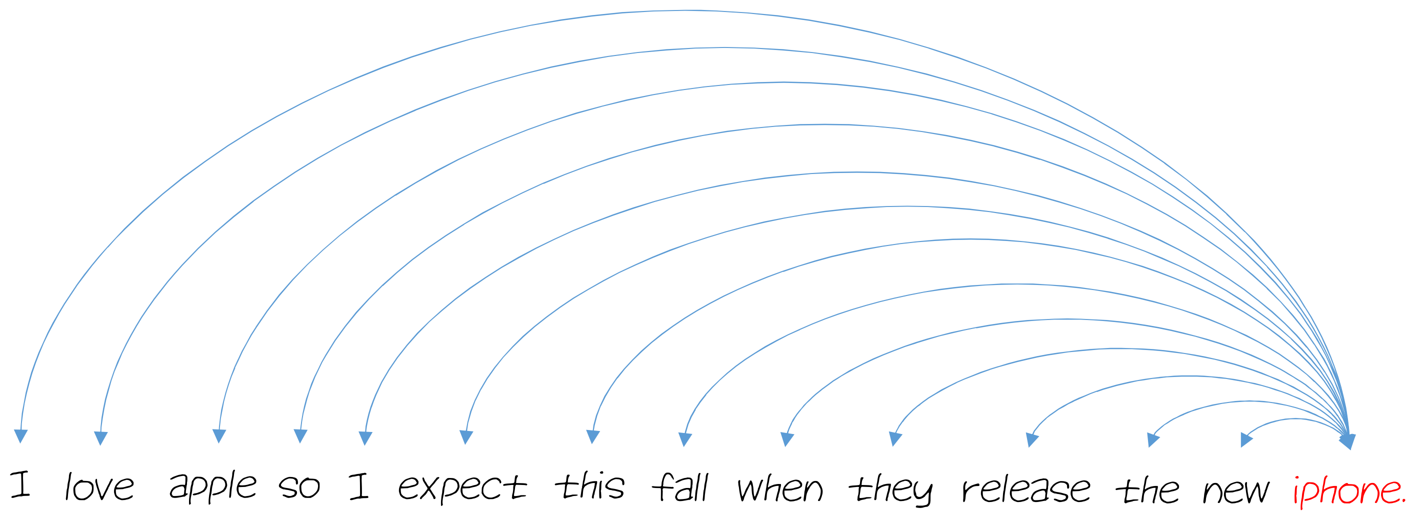
Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음



Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-1 : Input의 길이에 상관없이 context를 잘 파악할 수 있음



Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-2 : *안정적인 training이 가능*

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-2 : 안정적인 training이 가능

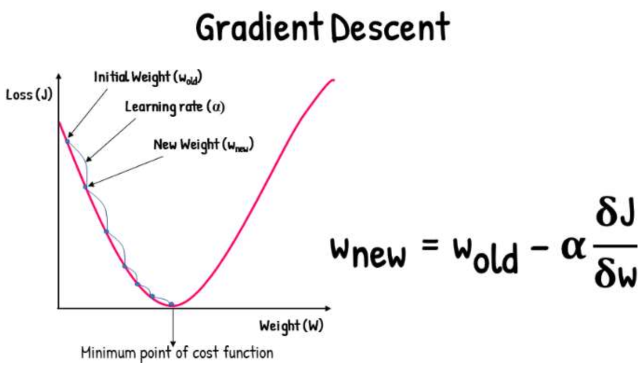


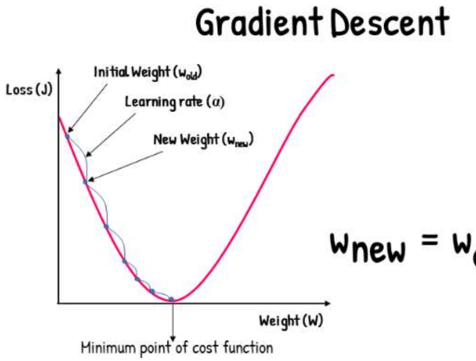
Image Credit: https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/

머신 러닝에서 gradient란 무슨 의미일까?

"피드백"

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-2 : 안정적인 training이 가능



$$w_{new} = w_{old} - \alpha \frac{\delta J}{\delta w}$$

Image Credit: https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/

머신 러닝에서 gradient란 무슨 의미일까?

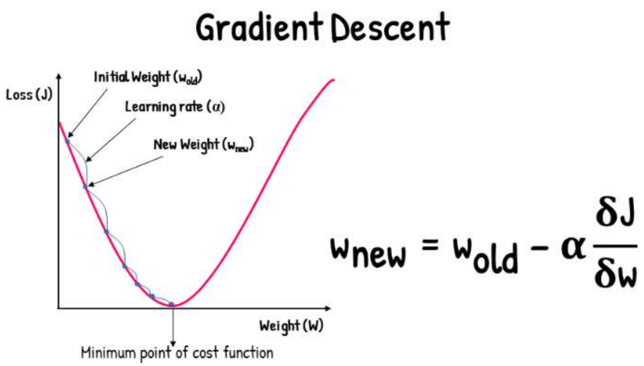
➡ "피드백"

그렇다면 vanishing gradient는 무슨 의미일까?

➡ "피드백이 없다"

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-2 : 안정적인 training이 가능



머신 러닝에서 gradient란 무슨 의미일까?

➡ "피드백"

그렇다면 vanishing gradient는 무슨 의미일까?

➡ "피드백이 없다"

피드백이 없다는 것은 무슨 의미일까?

➡ "학습을 멈춘다" ➡ Local minima

Q 5 : 좋아. Transformer는 temporal vanishing gradients도 없고 어순도 알아. 그럼 뭐가 좋아?

A 5-2 : *안정적인 training이 가능*

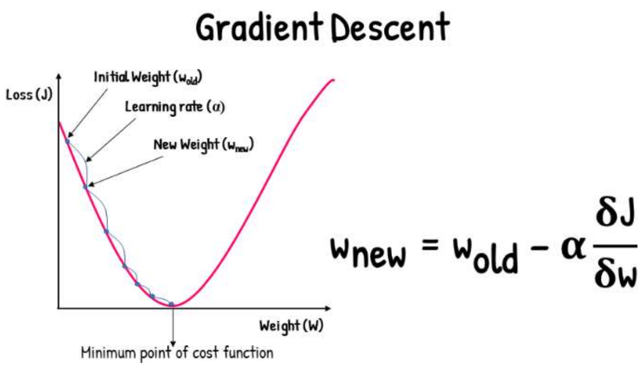


Image Credit: https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/

Transformer는 temporal vanishing gradients가 없기 때문에 local minima에 빠질 위험이 그만큼 줄어들어 보다 안정적인 training이 가능하다.

\+

Recursive(sequential) 구조가 아니므로 쉽게 병렬학습 가능

1회: 왜 transformer는 성능이 좋을까?

1회: 왜 transformer는 성능이 좋을까?

한 줄 요약 : Recursive 방식이 아닌 병렬화가 가능한 attention 매커니즘을 적용하므로 인풋의 길이에 상관 없이 심도깊은 context 파악이 가능하며, local minima에 빠지지 않고 대규모 병렬 학습이 가능하기 때문에

첫 컨텐츠 어떠셨나요?

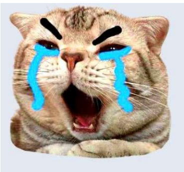
taekyo.lee@lge.com 으로 메일 주세요~

첫 컨텐츠 어떠셨나요?

taekyo.lee@lge.com 으로 메일 주세요~

횡설수설... 무슨 말 하는지 못알아 먹겠다

이 정도면 나쁘지 않네



너무 어려워!!!! 좀 쉽게!!!!

내가 알고 있던 거랑 좀 다른데?? 구라치는거 아냐?

강의자료 좀...

분량 조절 좀 똑바로 해라


감사합니다