Self-Attention For Generative Models

Ashish Vaswani and Anna Huang

Joint work with: Noam Shazeer, Niki Parmar, Lukasz Kaiser, Illia Polosukhin, Llion Jones, Justin Gilmer, David Bieber, Jonathan Frankle, Jakob Uszkoreit, and others.

Learning Representations of Variable Length Data

Basic building block of sequence-to-sequence learning

Neural machine translation, summarization, QA, …

Recurrent Neural Networks

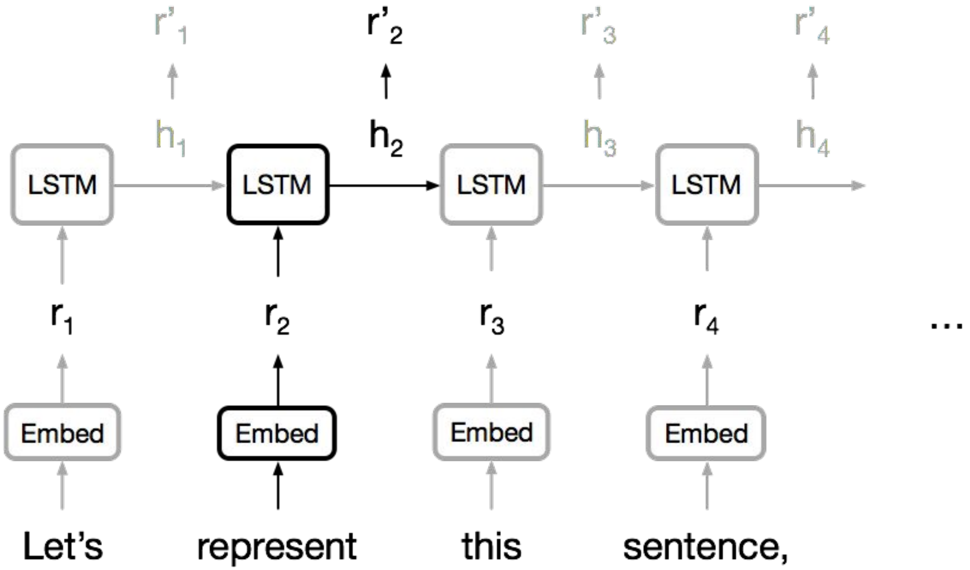
Model of choice for learning variable-length representations.

Natural fit for sentences and sequences of pixels.

LSTMs, GRUs and variants dominate recurrent models.

Recurrent Neural Networks

RNN은 어떻게 학습하는가?
RNN은 어떻게 representation을 생성하는가?



But…

recurrent modeling의 단점.

Sequential computation inhibits parallelization.

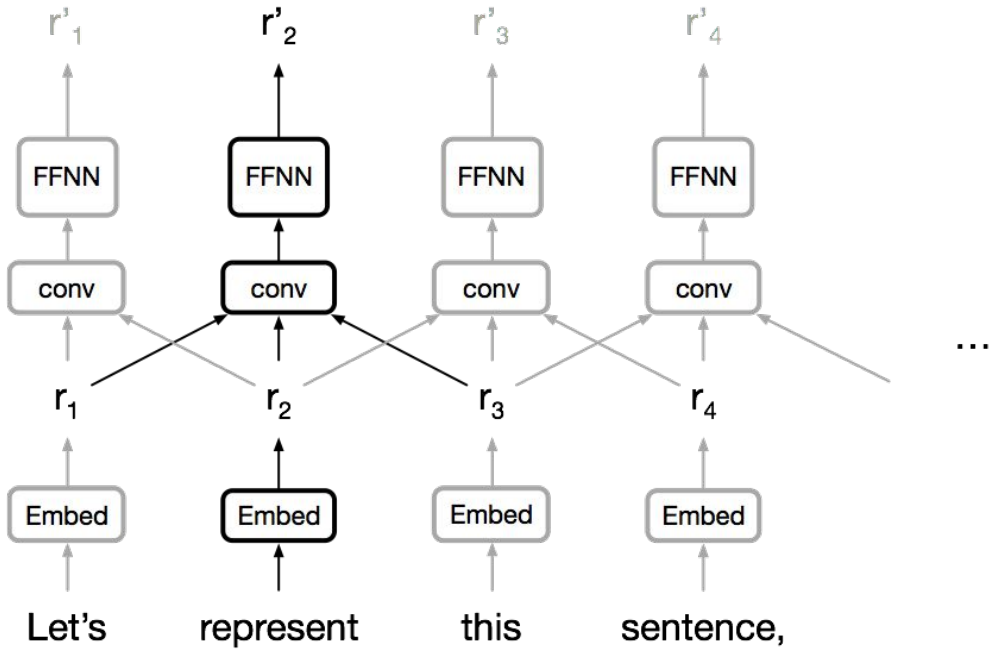
No explicit modeling of long and short range dependencies.

We want to model hierarchy.

hierarchy를 modeling하는 것은
특히나 '언어'에서 중요하다.

RNNs (w/ sequence-aligned states) seem wasteful!

Convolutional Neural Networks?



Convolutional Neural Networks?

Trivial to parallelize (per layer).

Exploits local dependencies

'Interaction distance' between positions linear or logarithmic.

Convolutional modeling의 단점.

Long-distance dependencies require many layers.

Attention

Attention between encoder and decoder is crucial in NMT.

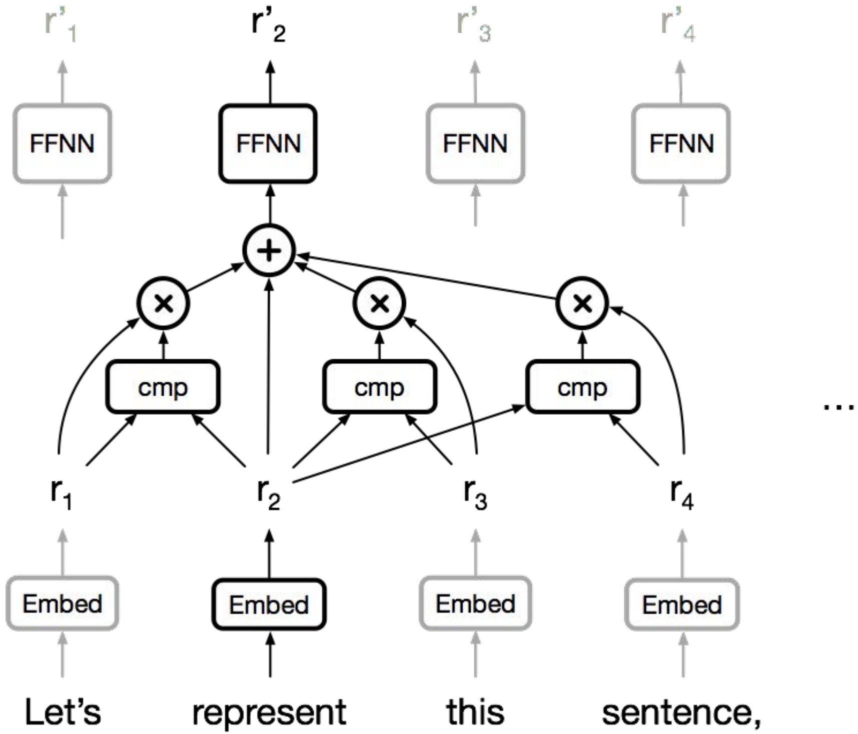
Why not use attention for representations?

Self-Attention

comparison (cmp)
아마.. 코사인 유사도

produce weighted combination of your entire neighborhood.

based on the weighted combination, you summarize all taht information



just like, re - expressing yourself in a weighted combination of your entire neighborhood.

Text generation

self - attention이 text generation을 잘하게 하는지 살펴보자.

what inductive biases are actually useful.? (=어떤 representation modeling이 좋을까?)

and we empirically showed that indeed they move the needle in text generation.

Self-Attention

Constant ‘path length’ between any two positions.

Gating/multiplicative interactions.

Trivial to parallelize (per layer).

Can replace sequential computation entirely?

Previous work

Classification & regression with self-attention:

Parikh et al. (2016), Lin et al. (2016)

Self-attention with RNNs:

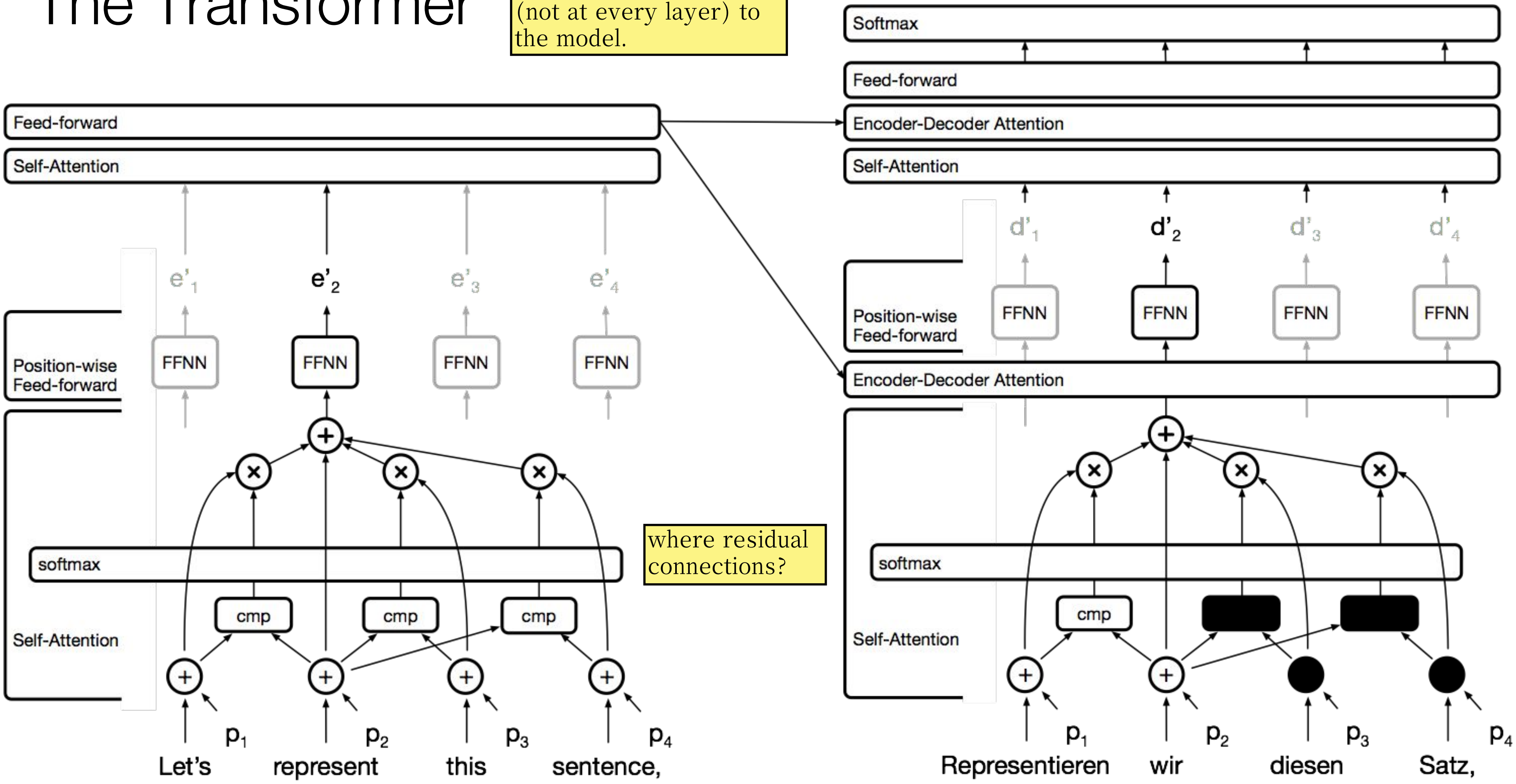
Long et al. (2016), Shao, Gows et al. (2017)

Recurrent attention:

Sukhbaatar et al. (2015)

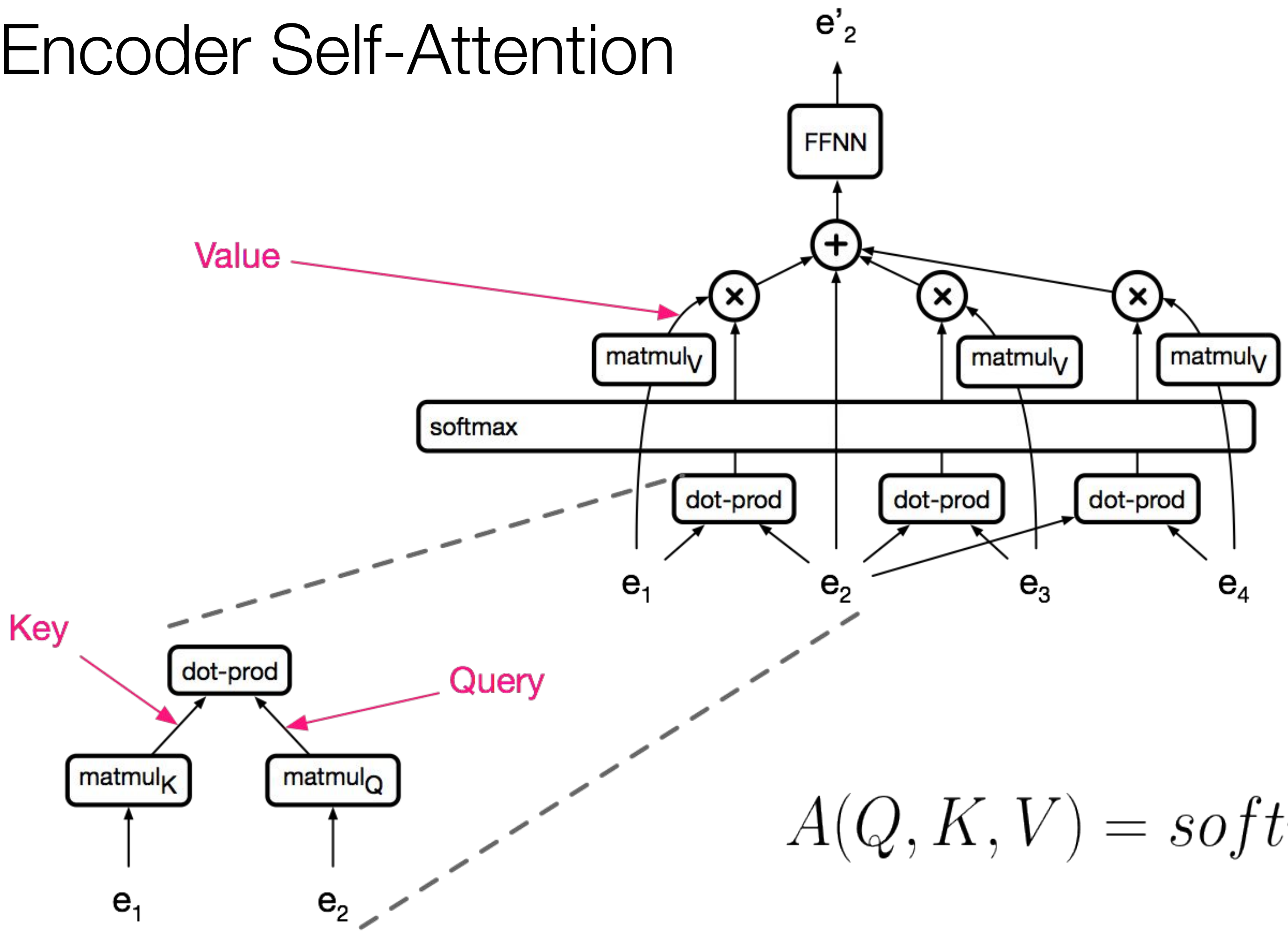
The Transformer

add the position informations at the input (not at every layer) to the model.



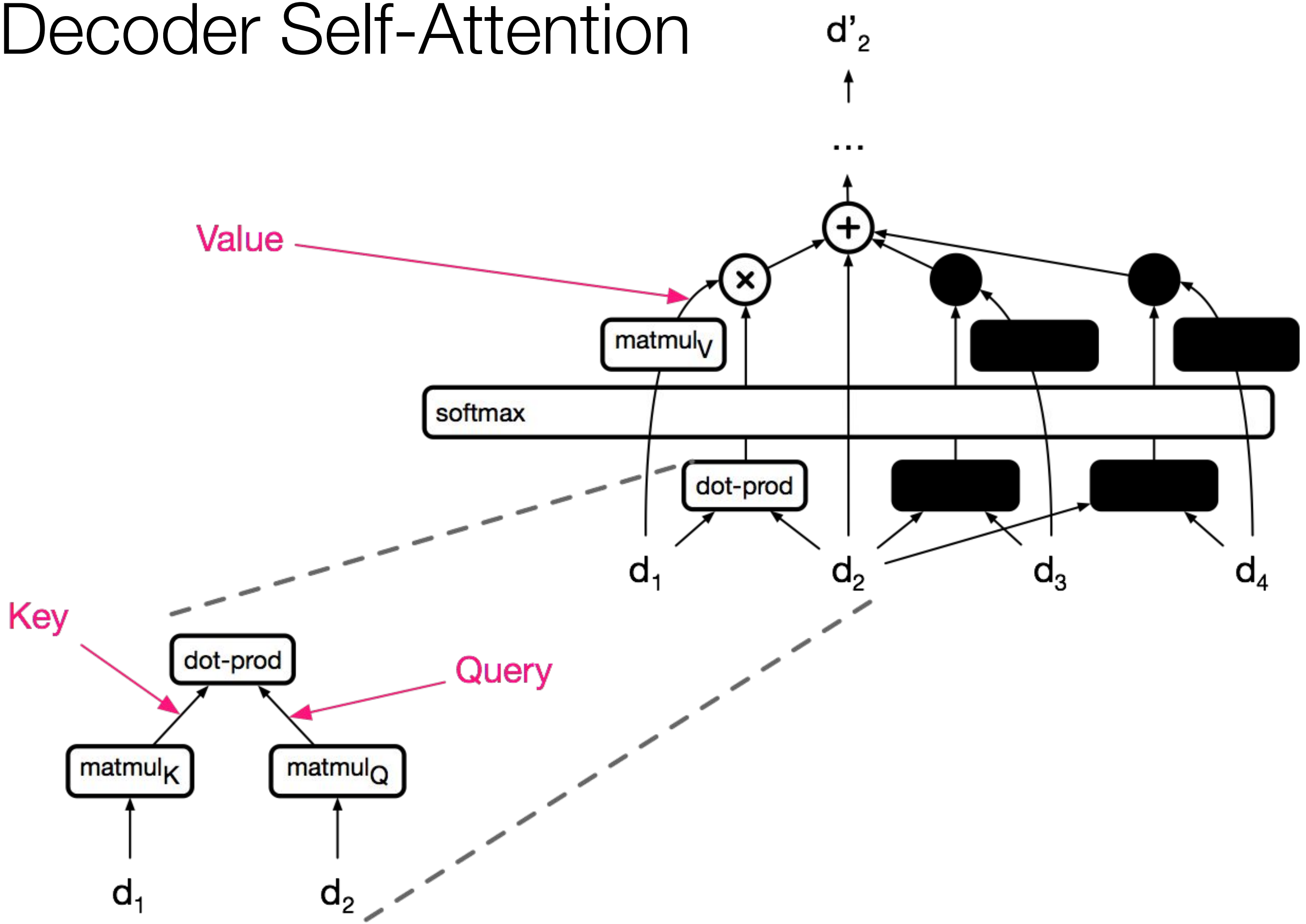
where residual connections?

Encoder Self-Attention



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

Decoder Self-Attention



Attention is Cheap!

	FLOPs
Self-Attention	$O(\text{length}^2 \cdot \text{dim})$
RNN (LSTM)	$O(\text{length} \cdot \text{dim}^2)$
Convolution	$O(\text{length} \cdot \text{dim}^2 \cdot \text{kernel\_width})$

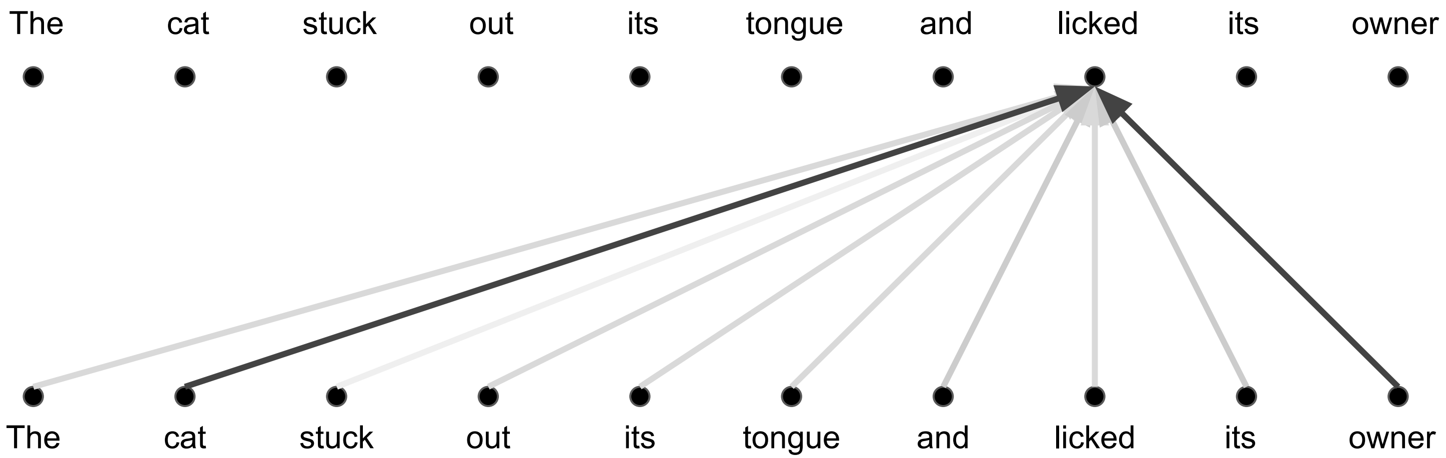
Attention is Cheap!

	FLOPs	
Self-Attention	$O(\text{length}^2 \cdot \text{dim})$	$= 4 \cdot 10^9$
RNN (LSTM)	$O(\text{length} \cdot \text{dim}^2)$	$= 16 \cdot 10^9$
Convolution	$O(\text{length} \cdot \text{dim}^2 \cdot \text{kernel\_width})$	$= 6 \cdot 10^9$

length=1000   dim=1000   kernel_width=3

dim이 length보다 커질수록
more attractive 할 수 있다.
즉, short sequence 에서는
self - attention 기법이 더 좋은 효과.

Attention: a weighted average

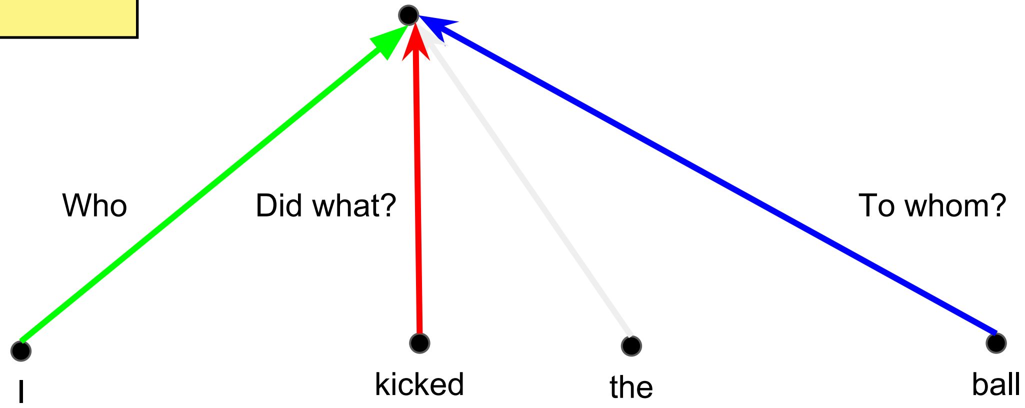


(single)
self - attention에 한 가지 문제가 있다.

일반적인 언어 문제로 Who / Did what? / To whom? 이라는 것을 알고 싶어한다.

convolution filter는
relative distance에 대해 다른 선형 변환을 할 수 있다. (색깔마다 선형 변환을 다르게 실시)

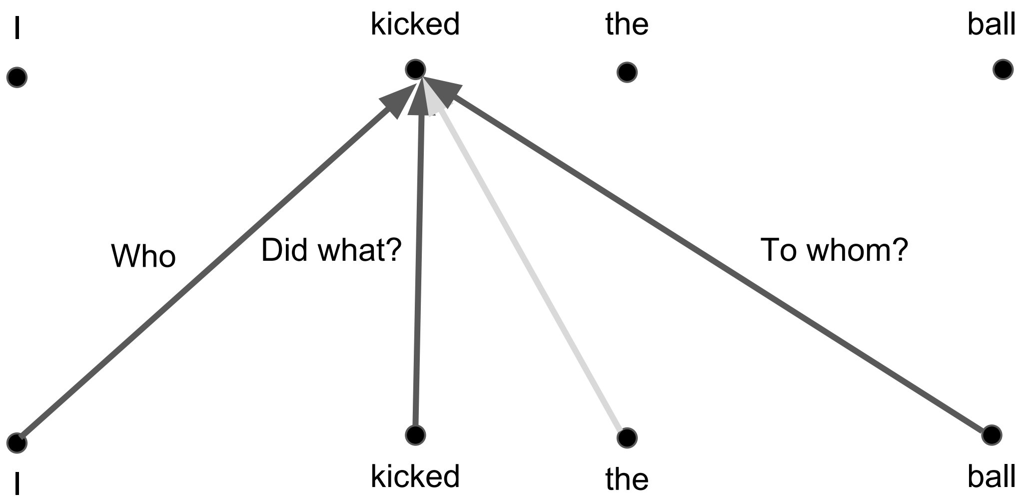
Convolutions



single attention layer에서는...

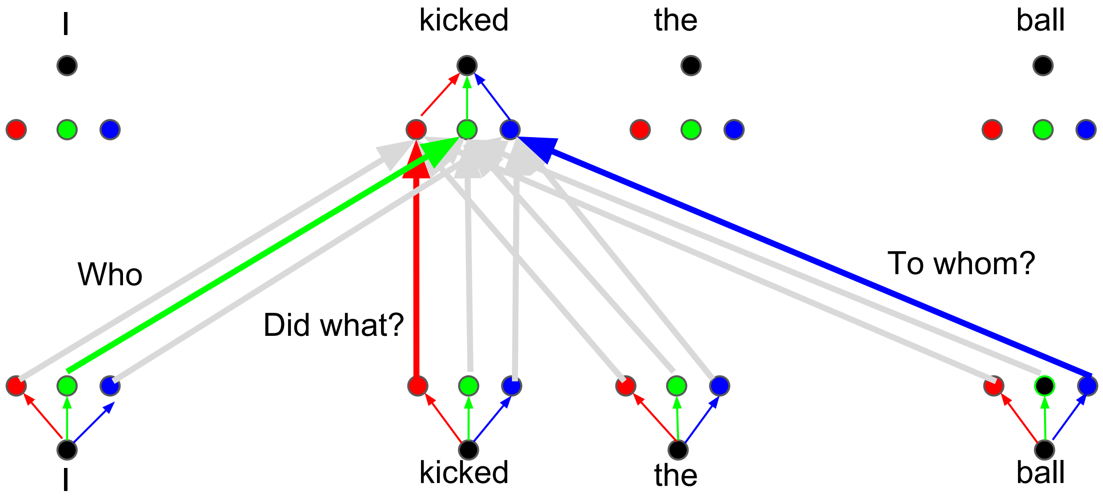
relative distance에 대해
다른 선형 변환을 하기 어렵다.
모두에 똑같은 선형 변환을 실시하기 때문.
you can't pick out different pieces of information from different places.

Self-Attention



Who,
Did what?
To whom?
각각에 대한 모델링을 할 수 있도록 attention layer를 여러 개 두면 어떨까?

Parallel attention heads

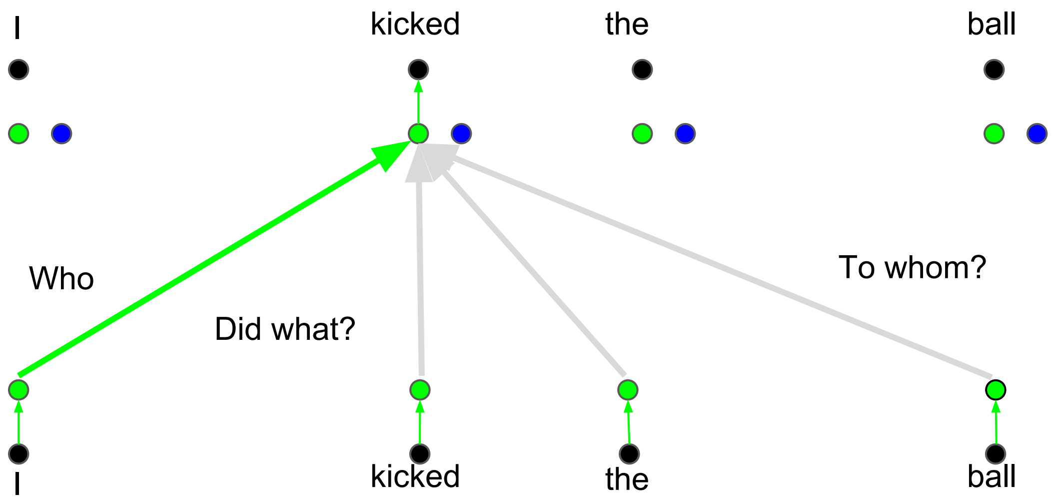


이러한 parallel attention heads는 비슷하게 convolution을 모델링한다.

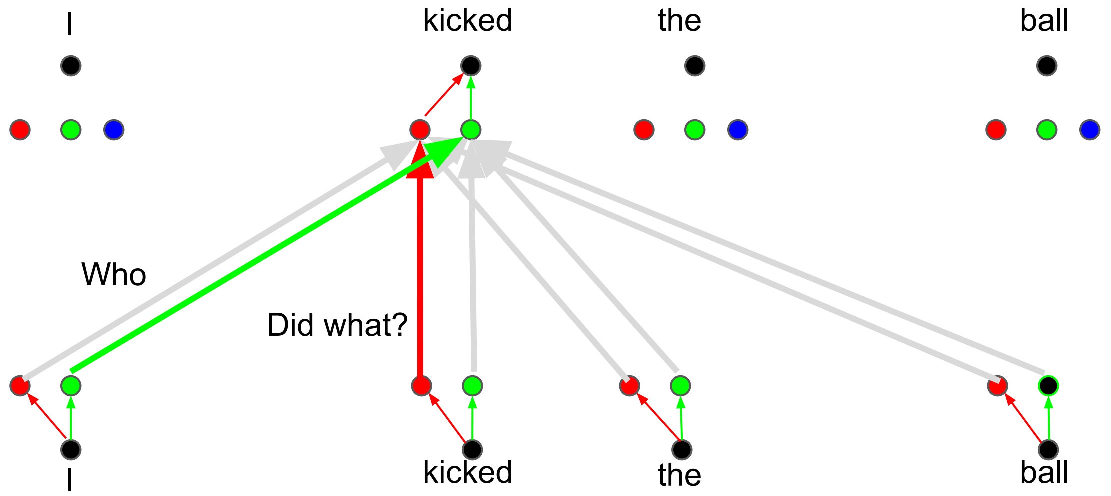
if you had more heads,
(heads are a function of positions) you could probably just simulate a convolution although with more parameters.

softmax를 더 추가했지만, FLOPs는 그렇게 바뀌지 않는다. 매우 큰 dim을 사용하지 않는 이상.

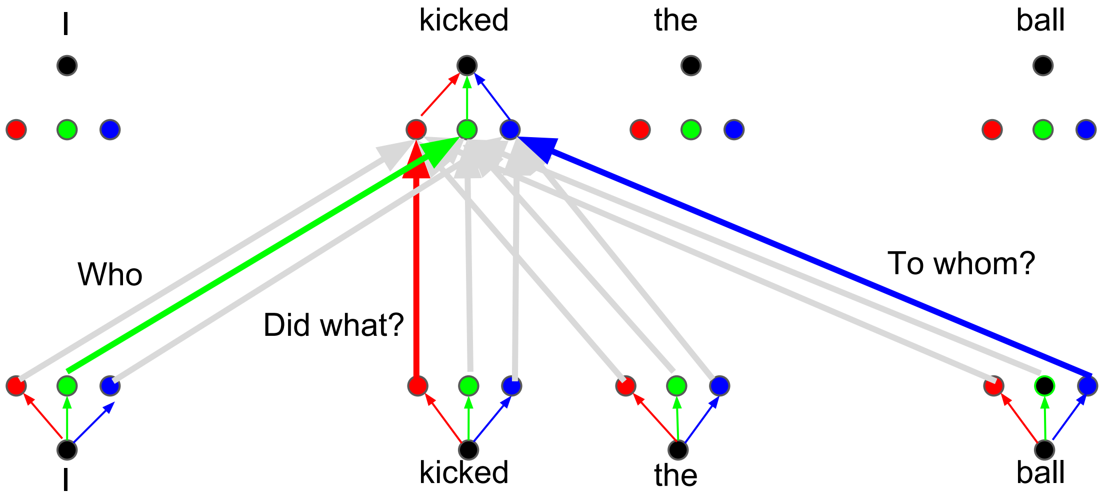
Attention head: Who



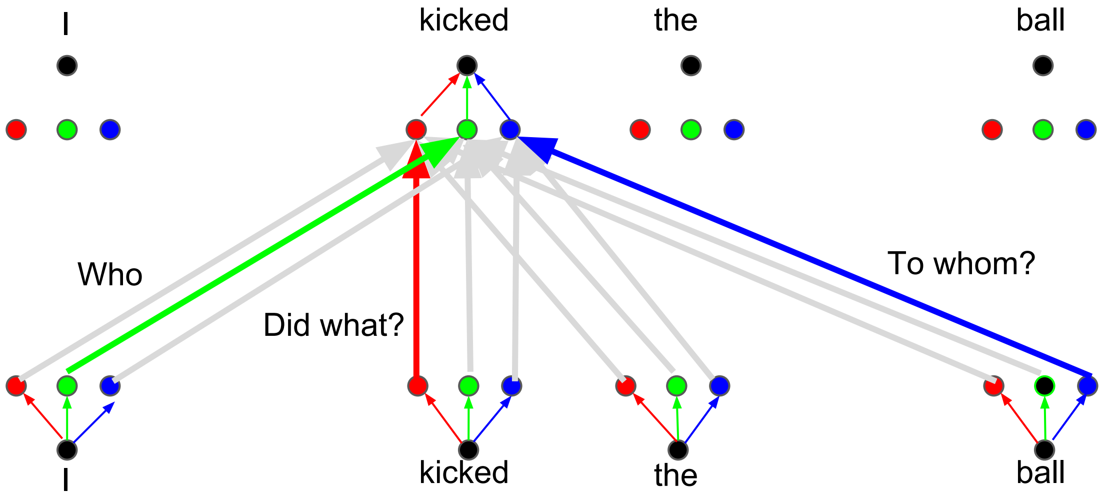
Parallel attention heads



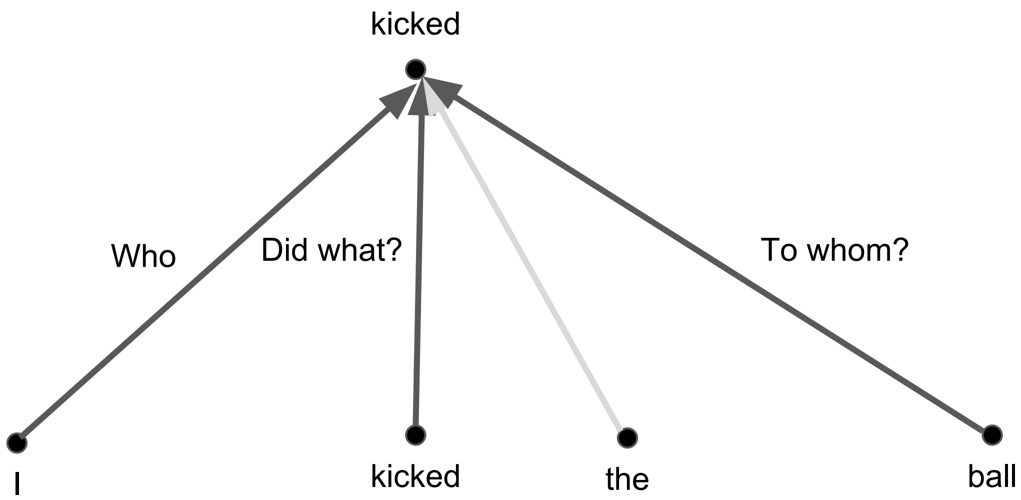
Parallel attention heads



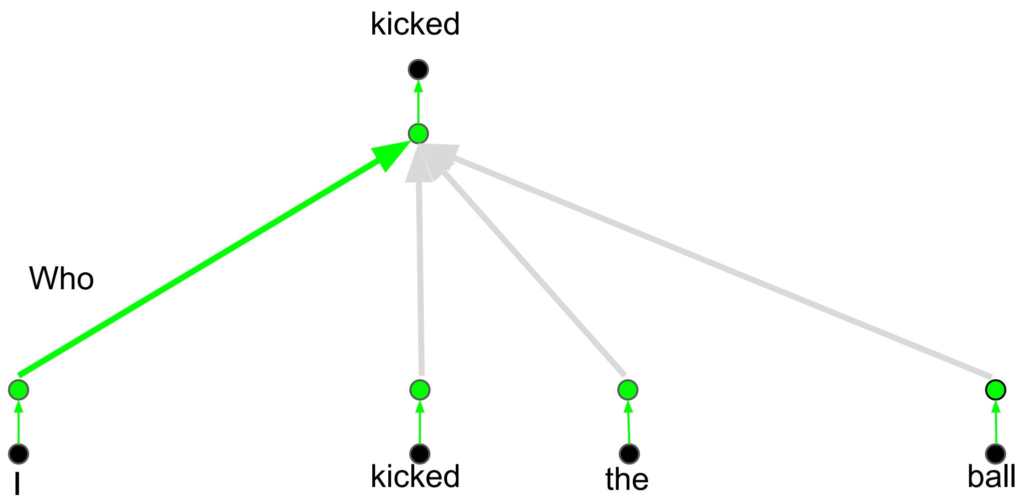
Parallel attention heads



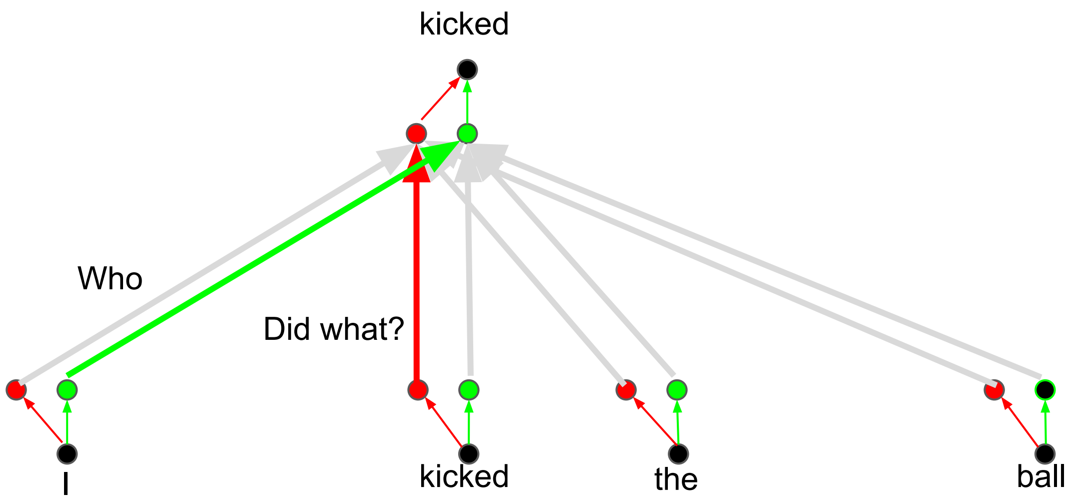
Self-Attention: Averaging



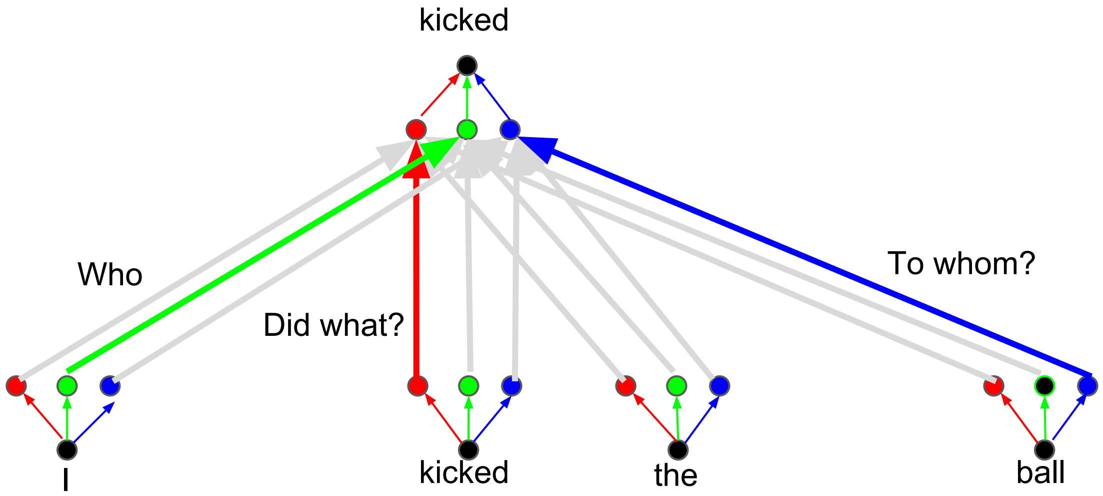
Attention head: Who



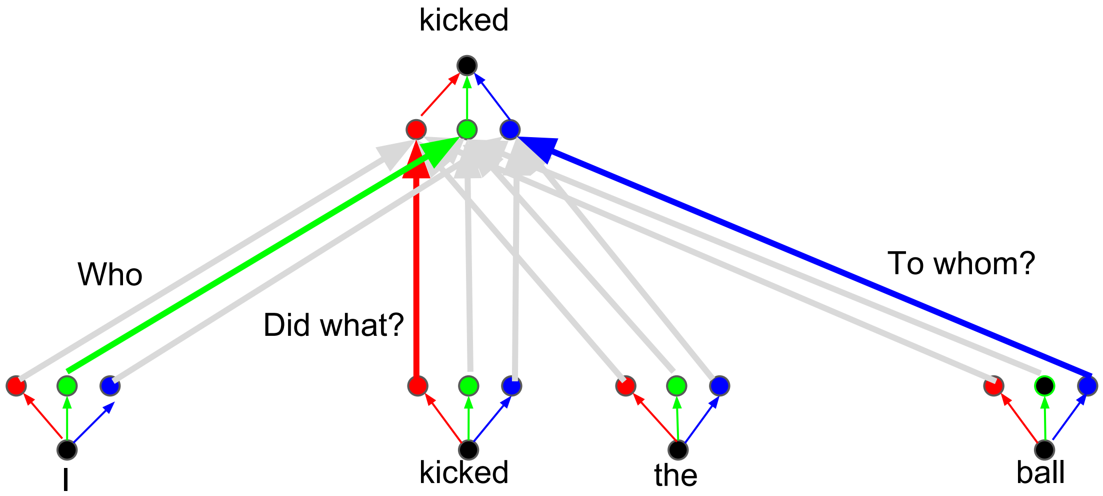
Attention head: Did What?



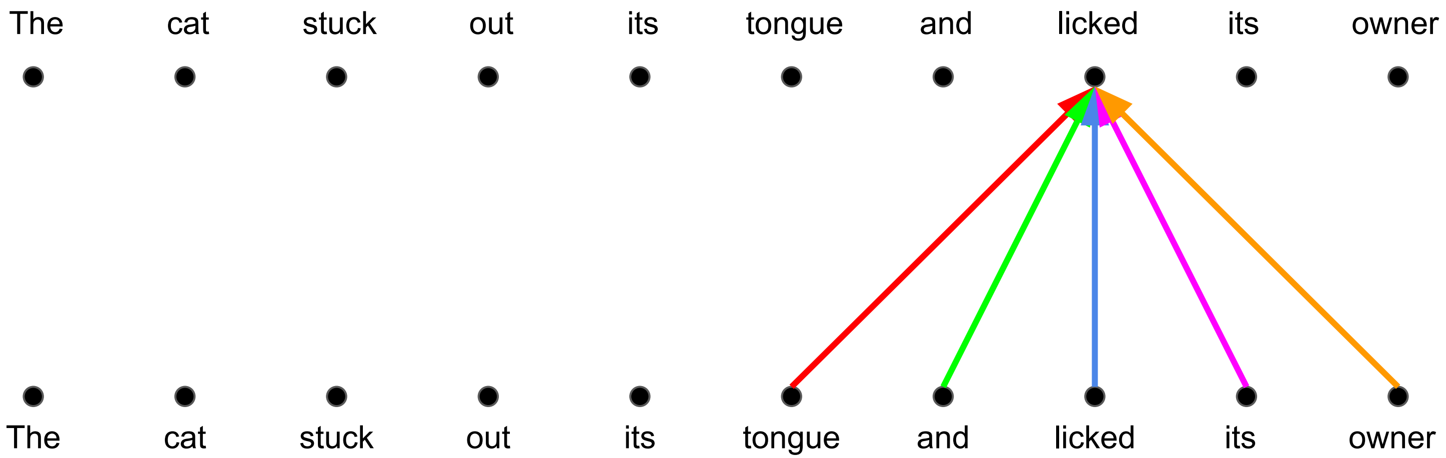
Attention head: To Whom?



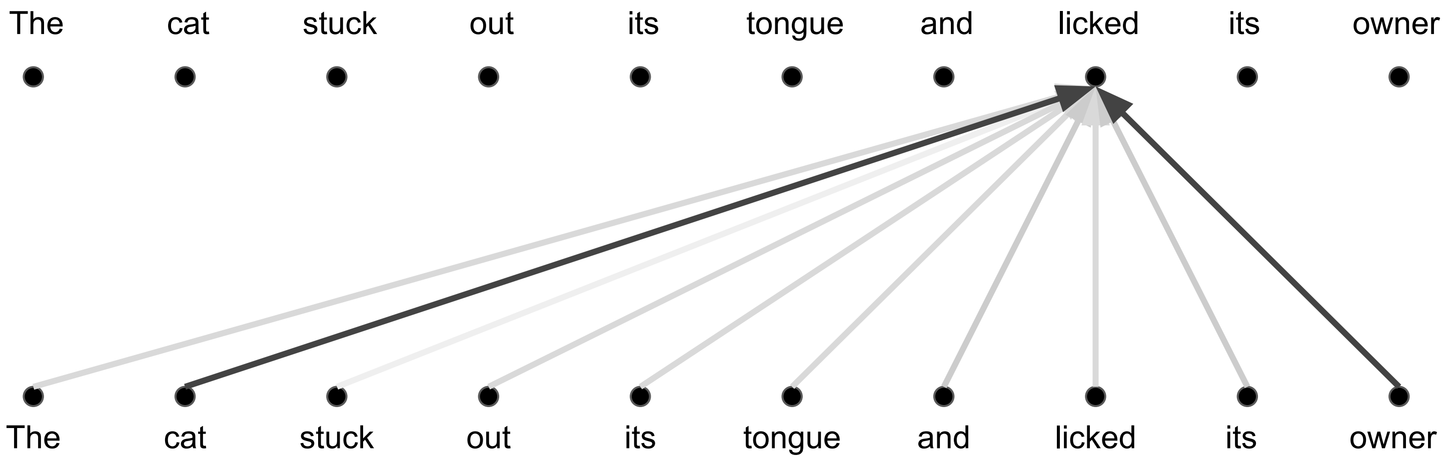
Multihead Attention



Convolution:
Different linear transformations by relative position.

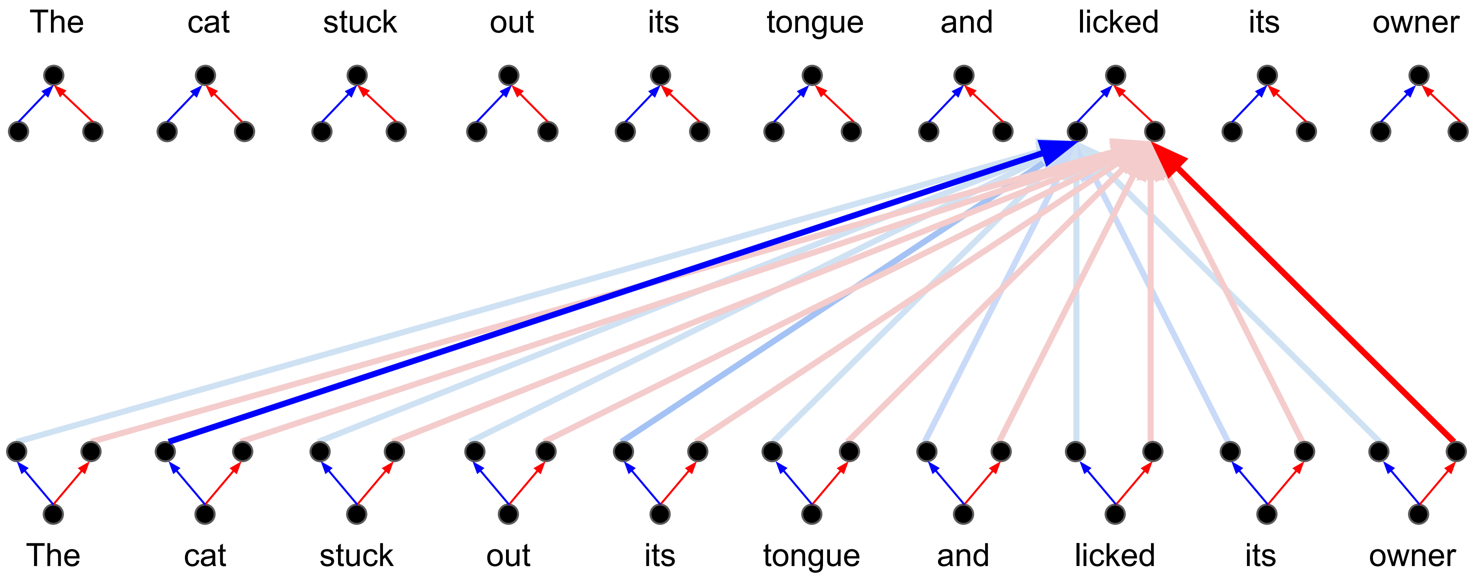


Attention: a weighted average



Multi-head Attention

Parallel attention layers with different linear transformations on input and output.



이론적으로 어떠한 함수도 모델링할 수 있는
LSTM 보다 더 나은 아키텍쳐를 만들었다고는 볼 수 없다.

단지, 우리는 SGD에 더 좋은 아키텍쳐를 만들었을 뿐이다.
because the gradient dynamics and attention are very simple.
and attentions are just a linear combination.

위는 호위적으로 말할 뿐이다.
사실 우리는 이러한 장점이 있다.

우리는 explicitly model all pairwise connection 을 했기에
두 단어 사이의 very clear relationship 모델링을 할 수 있었다.

Results

Machine Translation: WMT-2014 BLEU

	EN-DE	EN-FR
GNMT (orig)	24.6	39.9
ConvSeq2Seq	25.2	40.5
Transformer*	**28.4**	**41.8**

*Transformer models trained >3x faster than the others.

Attention is All You Need (NeurIPS 2017) Vaswani*, Shazeer*, Parmar*, Uszkoreit*, Jones*, Kaiser*, Gomez*, Polosukhin*

Frameworks:

[tensor2tensor]

[Sockeye]

Importance of residuals

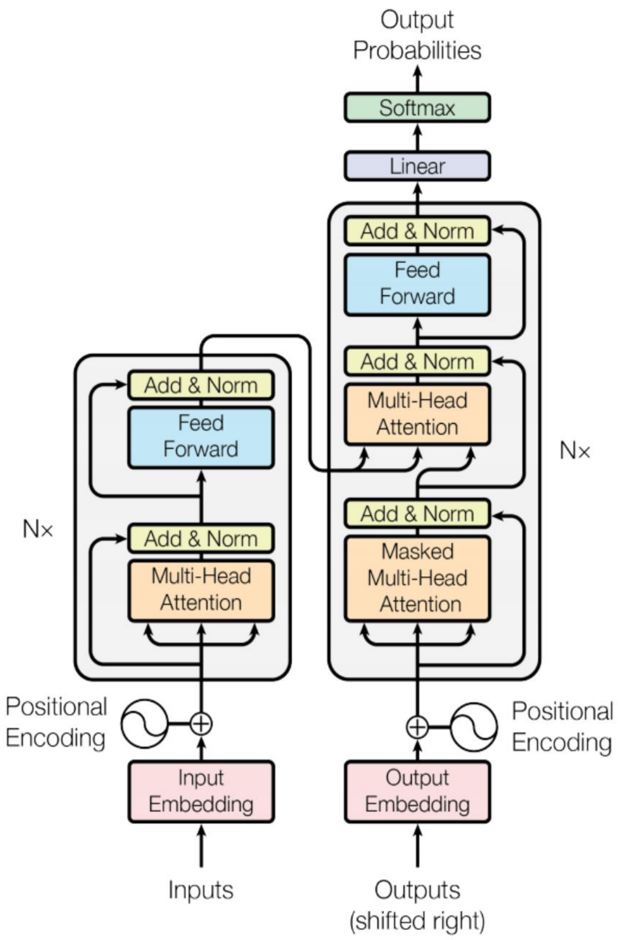
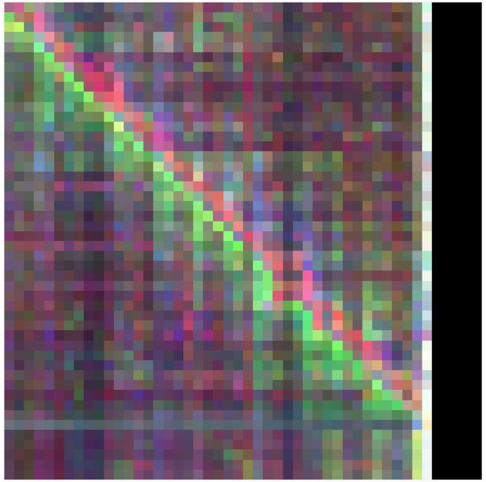


Figure 1: The Transformer - model architecture.

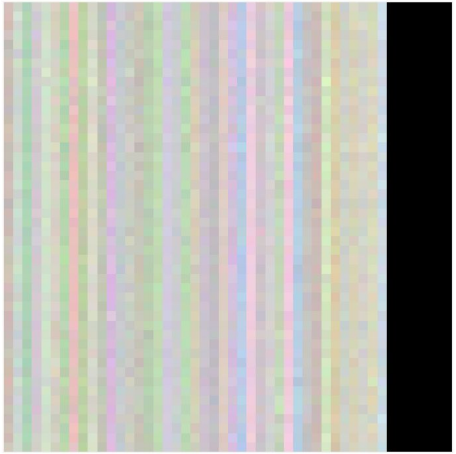
Importance of Residuals

residule을 통해 입력에 들어간 positional 정보를 모든 layer에 옮겨다 준다.

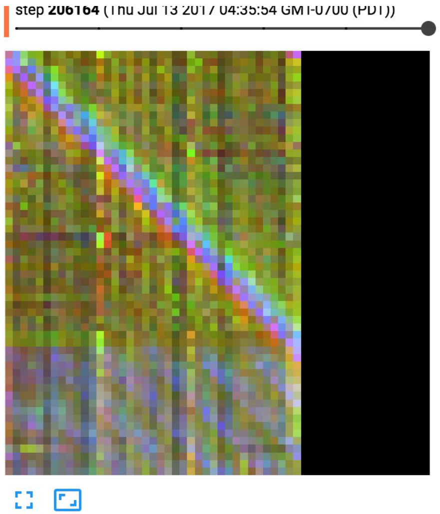
Residuals carry positional information to higher layers, among other information.



With residuals



Without residuals



Without residuals,
with timing signals

Training Details

ADAM optimizer with a learning rate warmup (warmup + exponential decay)

Dropout during training at every layer just before adding residual

Layer-norm

Attention dropout (for some experiments)

Checkpoint-averaging

Label smoothing

Auto-regressive decoding with beam search and length biasing
…

What Matters?

Resul

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

Generating Wikipedia by Summarizing Long Sequences

msaleh@ et al. submission to ICLR'18

	ROUGE
seq2seq-attention	12.7
Transformer-ED (L=500)	34.2
Transformer-DMCA (L=11000)	**36.2**

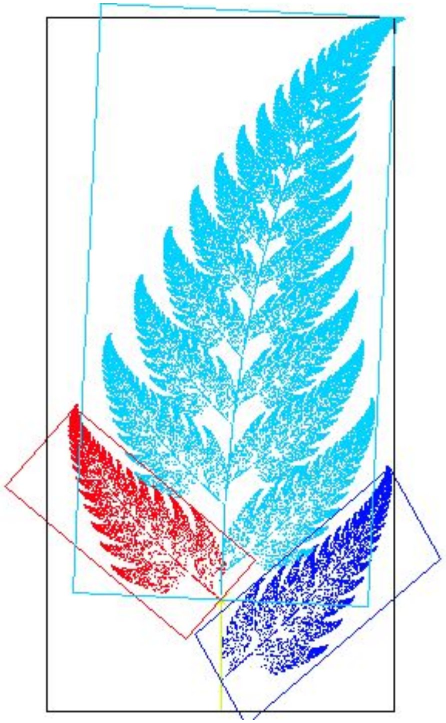
Self-Similarity, Image and Music Generation

Self-similarity in images

inductive biases는
representations 라고 생각해도된다.

이미지에서는 inductive biases가 어떻게 일어날까?

이미지나 음악의 특징은
비슷한 것끼리의 반복되는 구조를 가진다. (스케일은 다를 수 있음)



https://en.wikipedia.org/wiki/Self-similarity

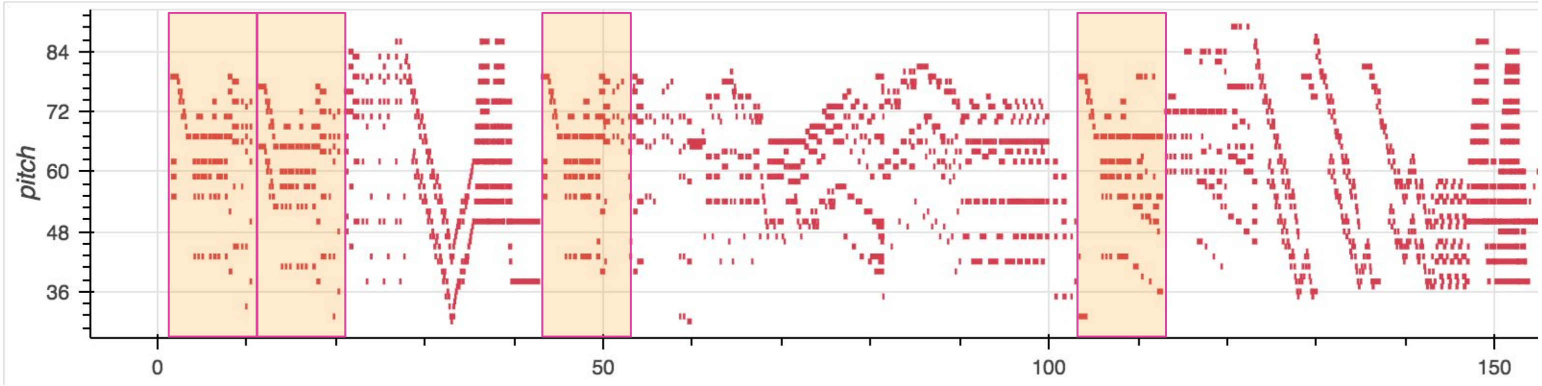
Self-Similarity in Images

These images are different pieces of the image. They are very similar to each other. But, they might have different scales.

Starry Night (Van Gogh, June 1889)

Self-similarity in music

Motifs repeat, immediately and also at a distance



Probabilistic Image Generation

Model the joint distribution of pixels

Turning it into a sequence modeling problem

Assigning probabilities allows measuring generalization

Can self - attention help us in modeling other objects like images?

Probabilistic Image Generation

RNNs and CNNs are state-of-the-art (PixelRNN, PixelCNN)

CNNs incorporating gating now match RNNs in quality

CNNs are much faster due to parallelization

A Oord et al. (2016), Salimans et al. (2017), Kalchbrenner et al. (2016)

마치, self - attention은 good computational mechanism을 가졌다고 볼 수 있다.

Probabilistic Image Generation

Long-range dependencies matter for images (e.g. symmetry)

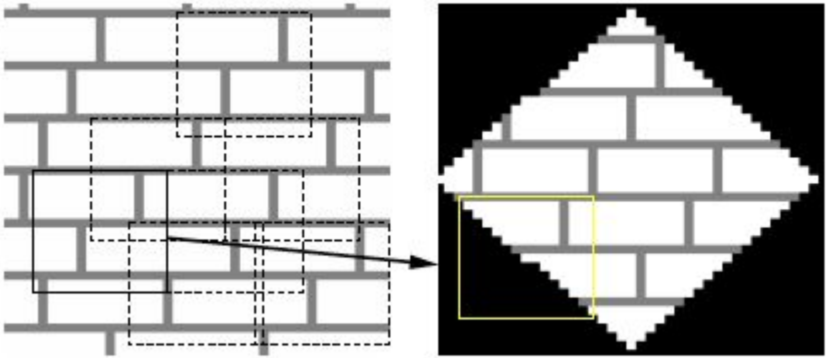
Likely increasingly important with increasing image size

Modeling long-range dependencies with CNNs requires either

long - range dependency를 모델링하는 데 있어서 CNN의 단점.

- **Many layers** likely making training harder
- **Large kernels** at large parameter/computational cost

Texture Synthesis with Self-Similarity



Texture Synthesis by Non-parametric Sampling (Efros and Leung, 1999)

Non-local Means

image denoising

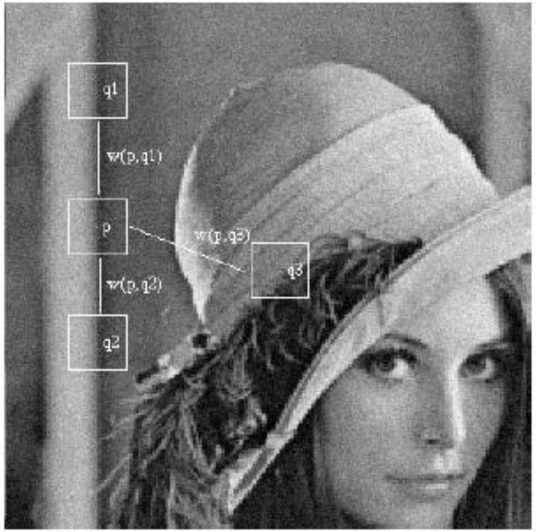


Figure 1. Scheme of NL-means strategy. Similar pixel neighborhoods give a large weight, w(p,q1) and w(p,q2), while much different neighborhoods give a small weight w(p,q3).

BCM 2005

Non-local Means

A Non-local Algorithm for Image Denoising (Buades, Coll, and Morel. CVPR 2005)

Non-local Neural Networks (Wang et al., 2018)

Previous work

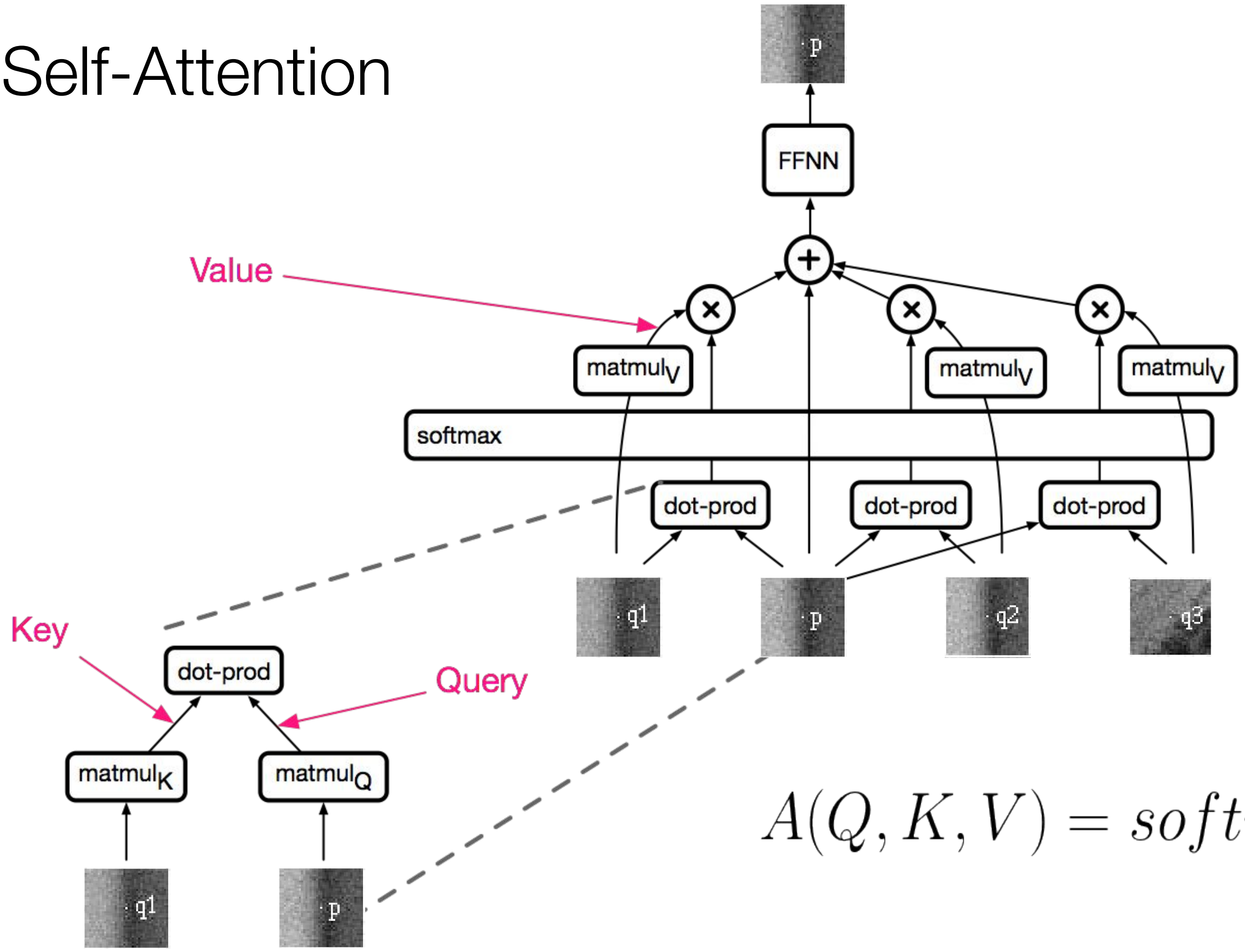
Self-attention:

Parikh et al. (2016), Lin et al. (2016), Vaswani et al. (2017)

Autoregressive Image Generation:

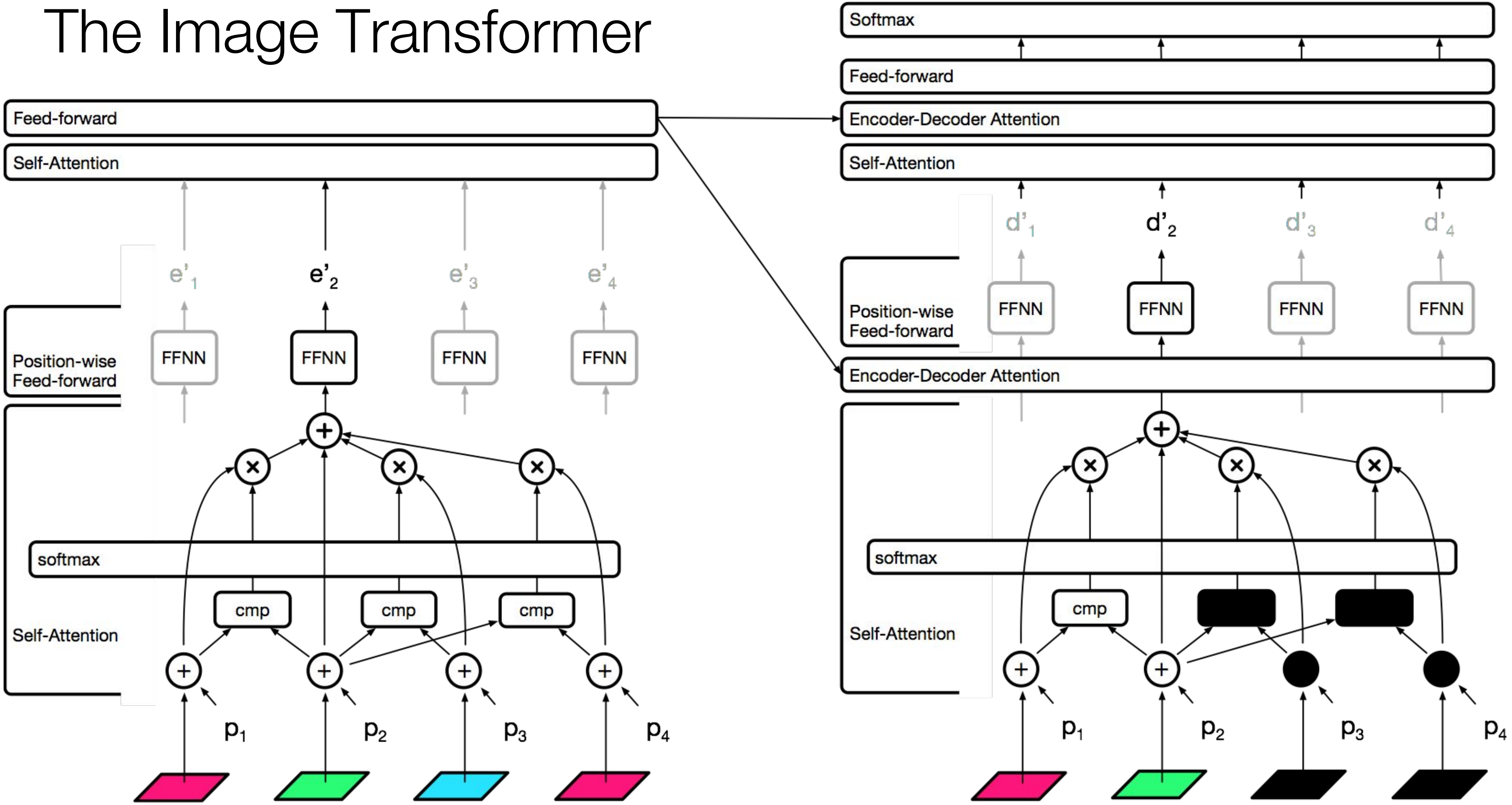
A Oord et al. (2016), Salimans et al. (2017)

Self-Attention

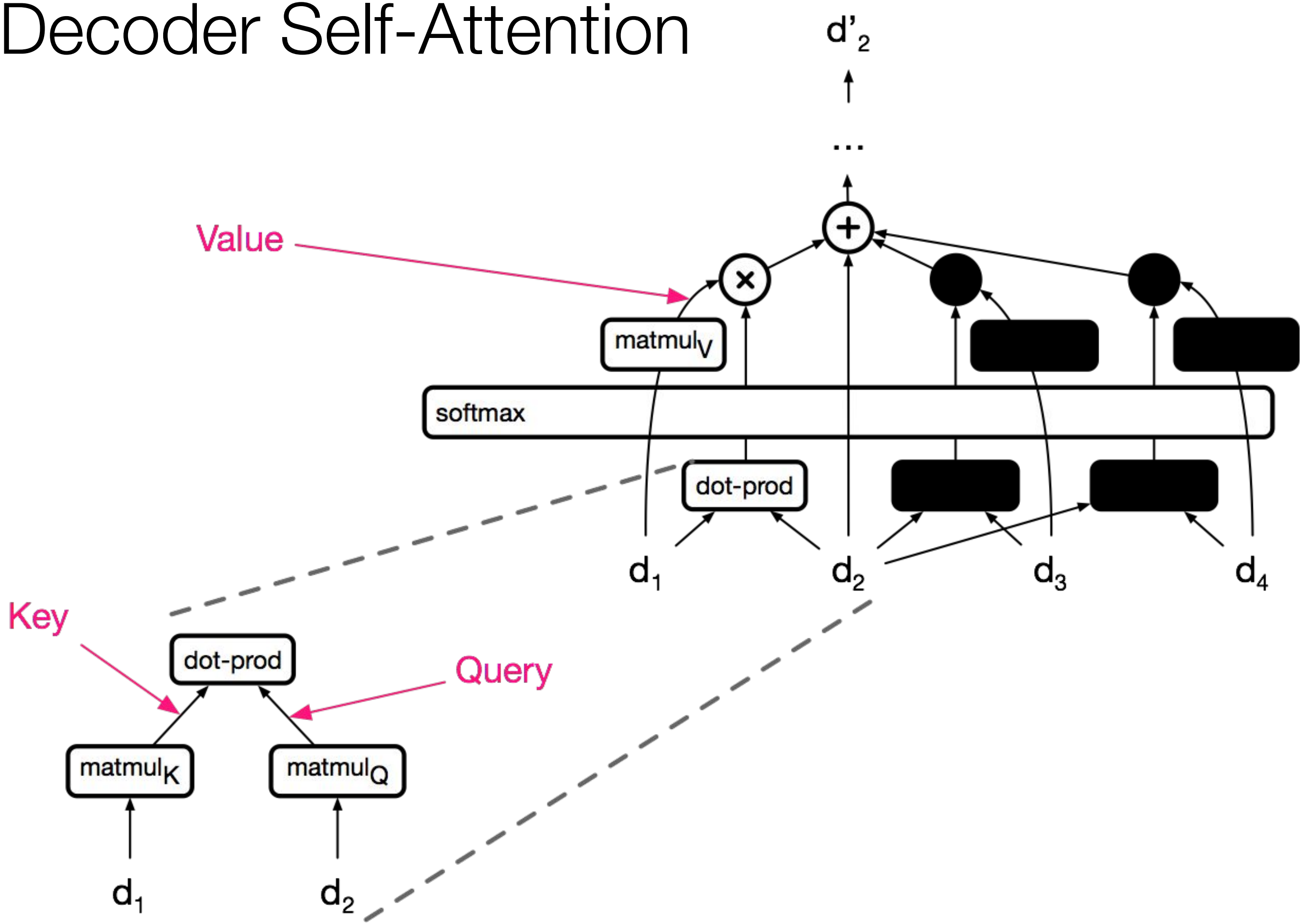


$$A(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

The Image Transformer



Decoder Self-Attention



Attention is Cheap!

	FLOPs
Self-Attention	$O(\text{length}^2 \cdot \text{dim})$
RNN (LSTM)	$O(\text{length} \cdot \text{dim}^2)$
Convolution	$O(\text{length} \cdot \text{dim}^2 \cdot \text{kernel\_width})$

Attention is Cheap if length << dim!

이렇기에 사실 이미지는 self - attention에 적합하지 않다?

FLOPs

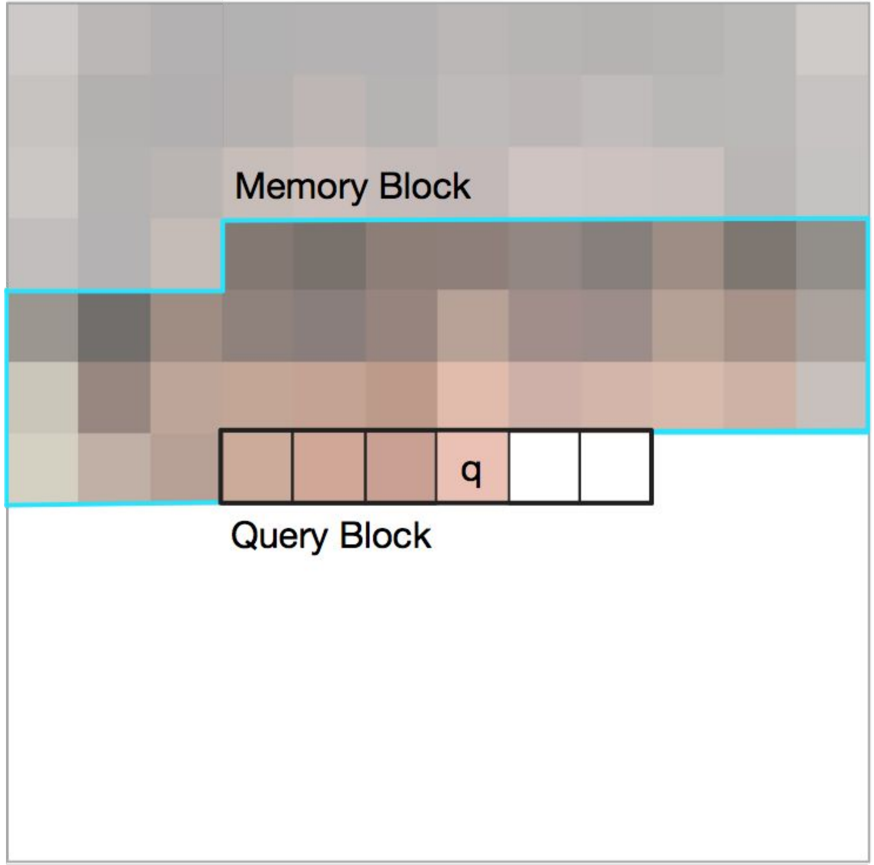
Self-Attention	$O(\text{length}^2 \cdot \text{dim})$ (length=3072 for images)
RNN (LSTM)	$O(\text{length} \cdot \text{dim}^2)$
Convolution	$O(\text{length} \cdot \text{dim}^2 \cdot \text{kernel\_width})$

Combining Locality with Self-Attention

Restrict the attention windows to be local neighborhoods

Good assumption for images because of spatial locality

Local 1D Attention



Local 2D Attention

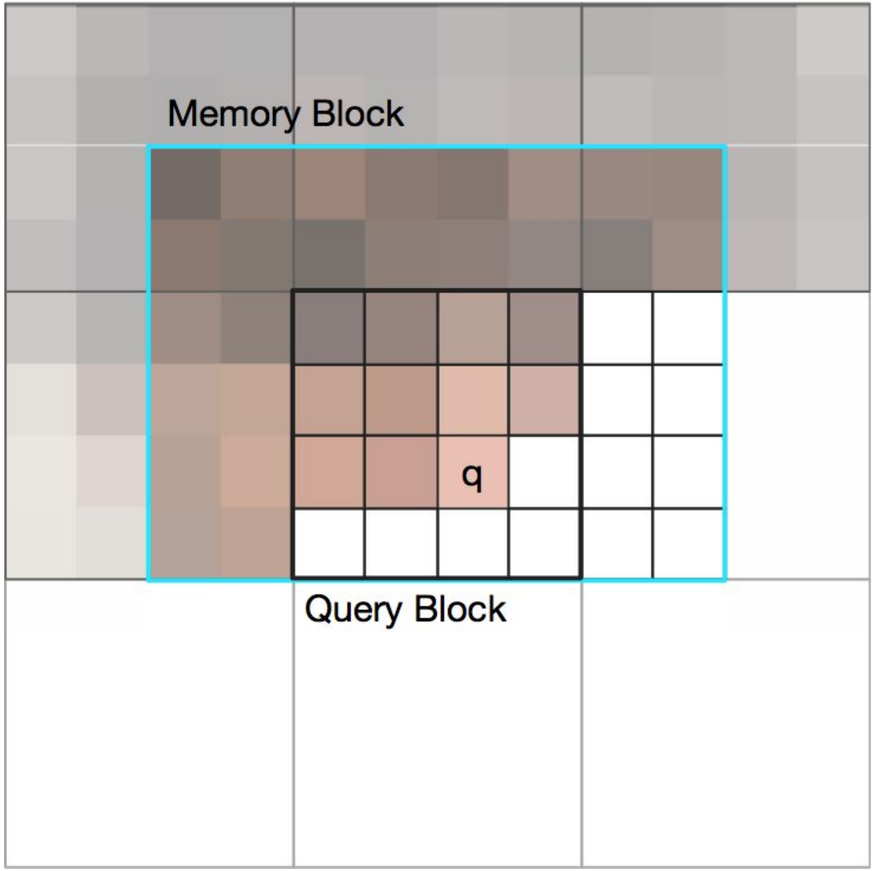
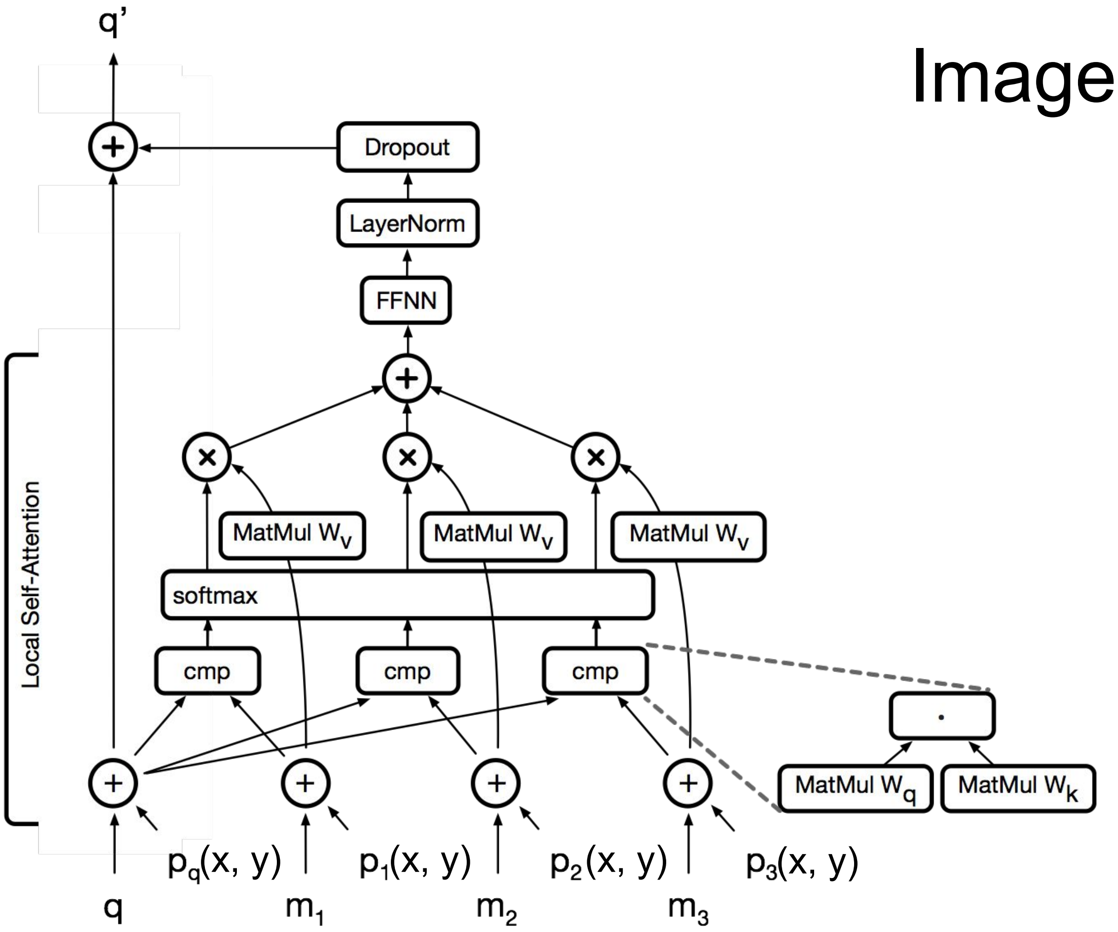


Image Transformer Layer



Tasks

Super-resolution

Unconditional and Conditional Image generation

Results

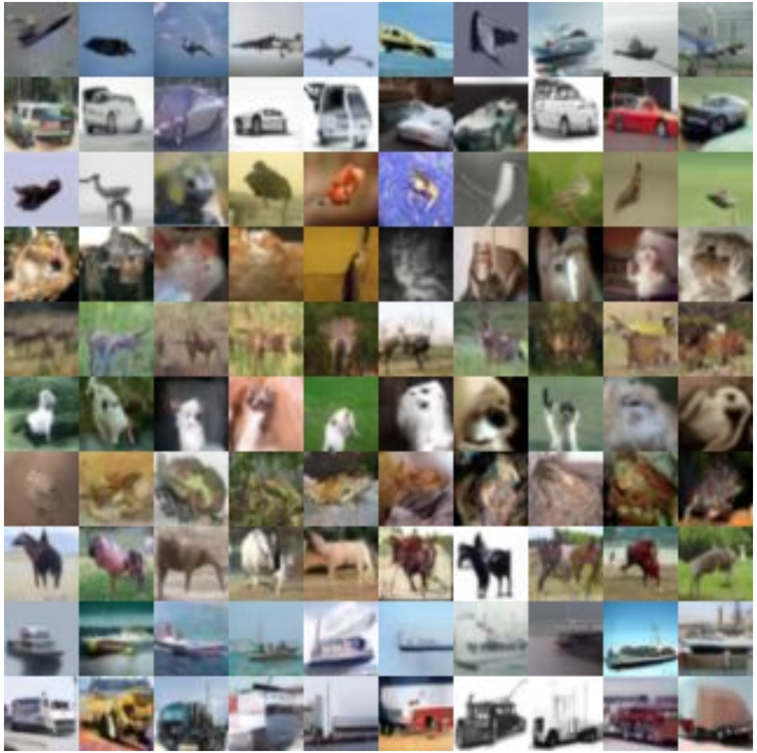
Image Transformer
Parmar*, Vaswani*, Uszkoreit, Kaiser, Shazeer, Ku, and Tran. ICML 2018

Unconditional Image Generation

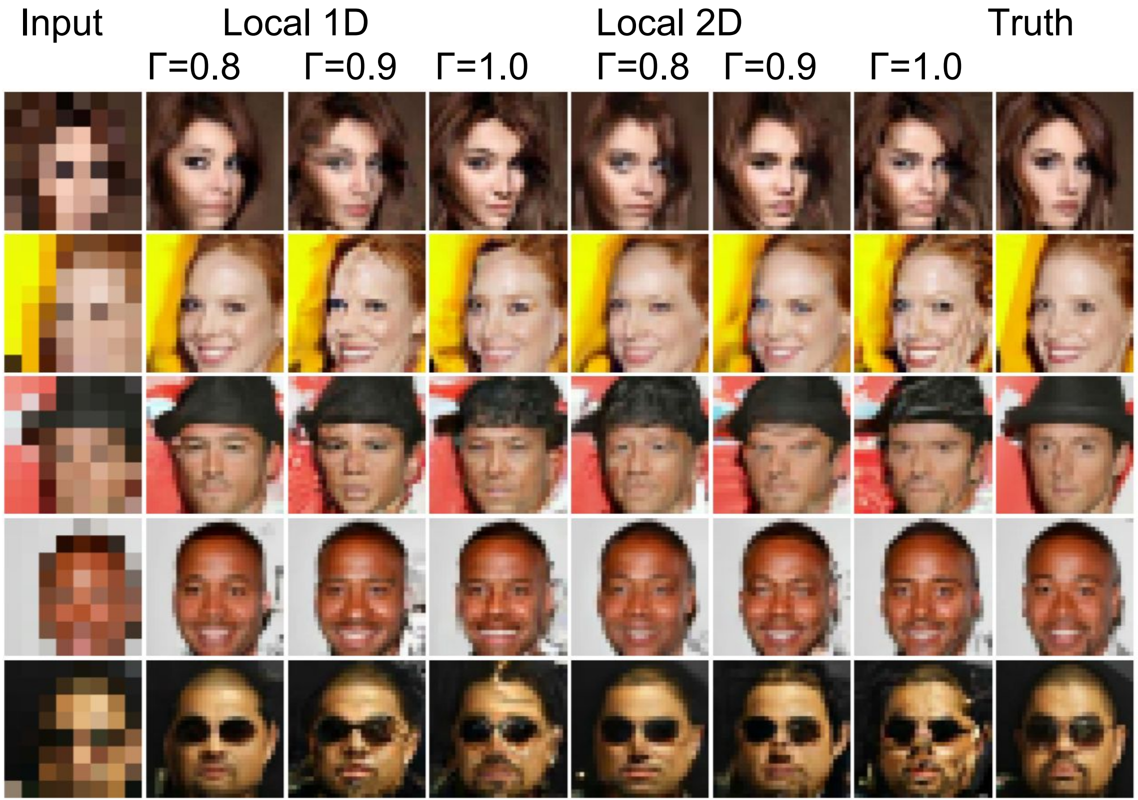
	Cifar-10 (Test)	Imagenet (Validation)
PixelRNN	3.00	3.86
Gated PixelCNN	3.03	3.83
PixelCNN++	2.92 (dmol)	-
PixelSNAIL	**2.85**	3.8
Image Transformer, 1D local	2.9 (xent)	**3.77**
Image Transformer, 1D local	2.9 (dmol)	3.78

Cross entropy of various models on CIFAR-10 and Imagenet datasets.

Cifar10 Samples



CelebA Super Resolution

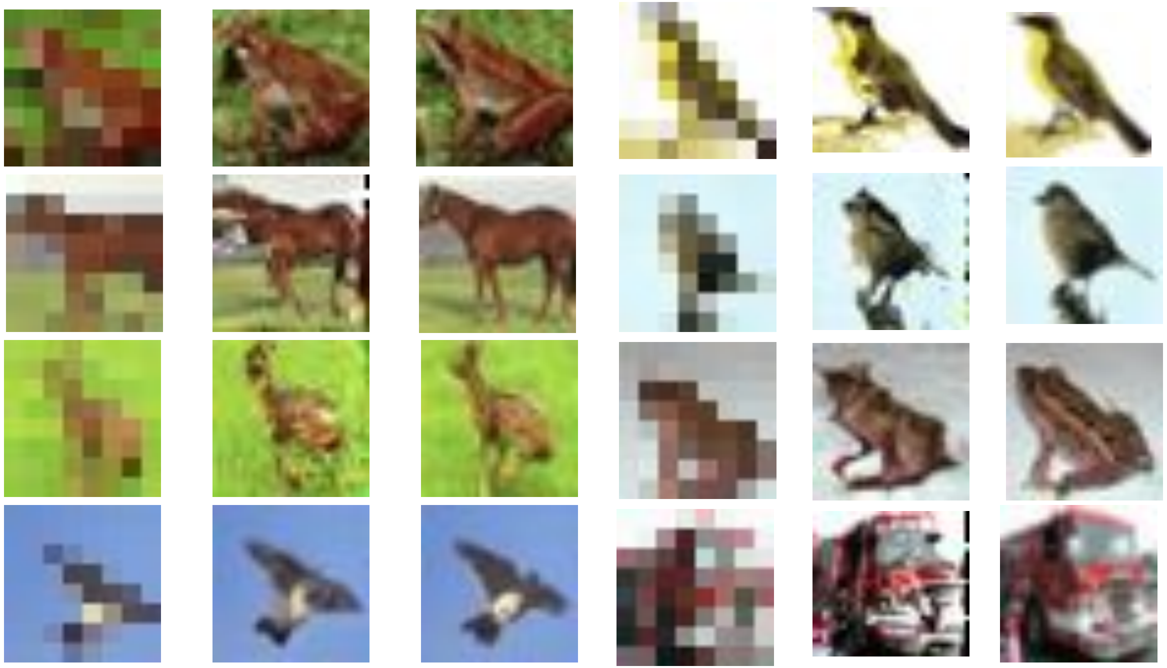


CelebA Super Resolution

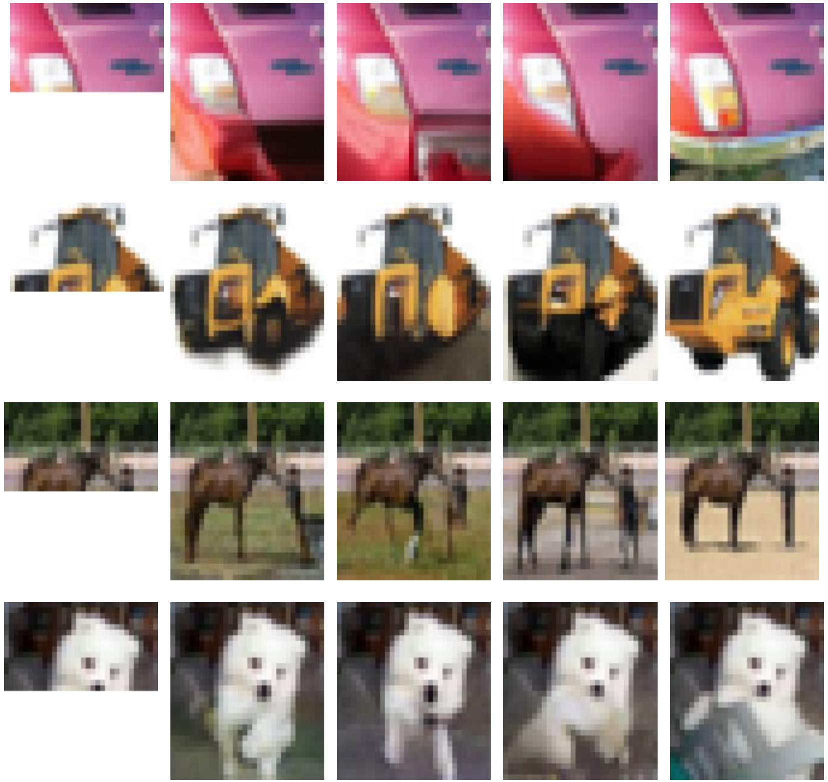
	% Fooled			
	Γ = n/a	Γ = 1.0	Γ = 0.9	Γ = 0.8
ResNet	4.0	-	-	-
srez GAN (Garcia, 2016)	8.5	-	-	-
Pixel Recursive (Dahl et al., 2017)	-	11.0	10.4	10.25
Image Transformer, 1D local		**35.94** ± 3.0	33.5 ± 3.5	29.6 ± 4.0
Image Transformer, 2D local		**36.11** ±2.5	34 ± 3.5	30.64 ± 4.0

Human Eval performance for the Image Transformer on CelebA. The fraction of humans fooled is significantly better than the previous state of art.

Cifar10 SuperResolution



Conditional Image Completion



Music generation using relative self-attention

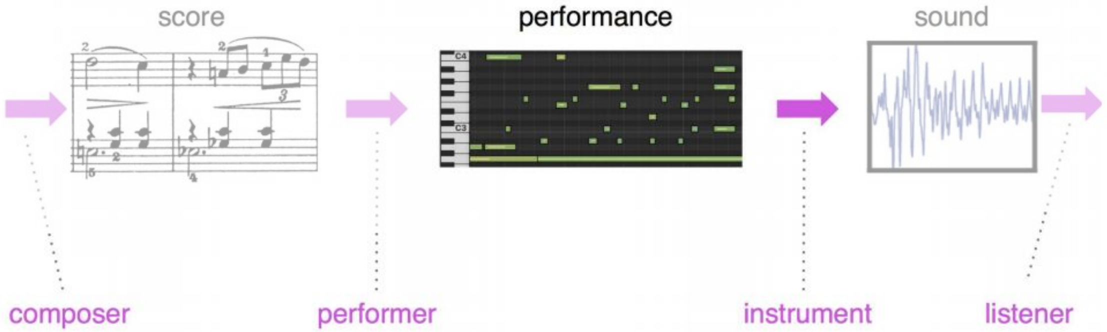
Music Transformer (ICLR 2019) by Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu and Douglas Eck.

Blog post: https://magenta.tensorflow.org/music-transformer

Raw representations in music and language

Language text → speech

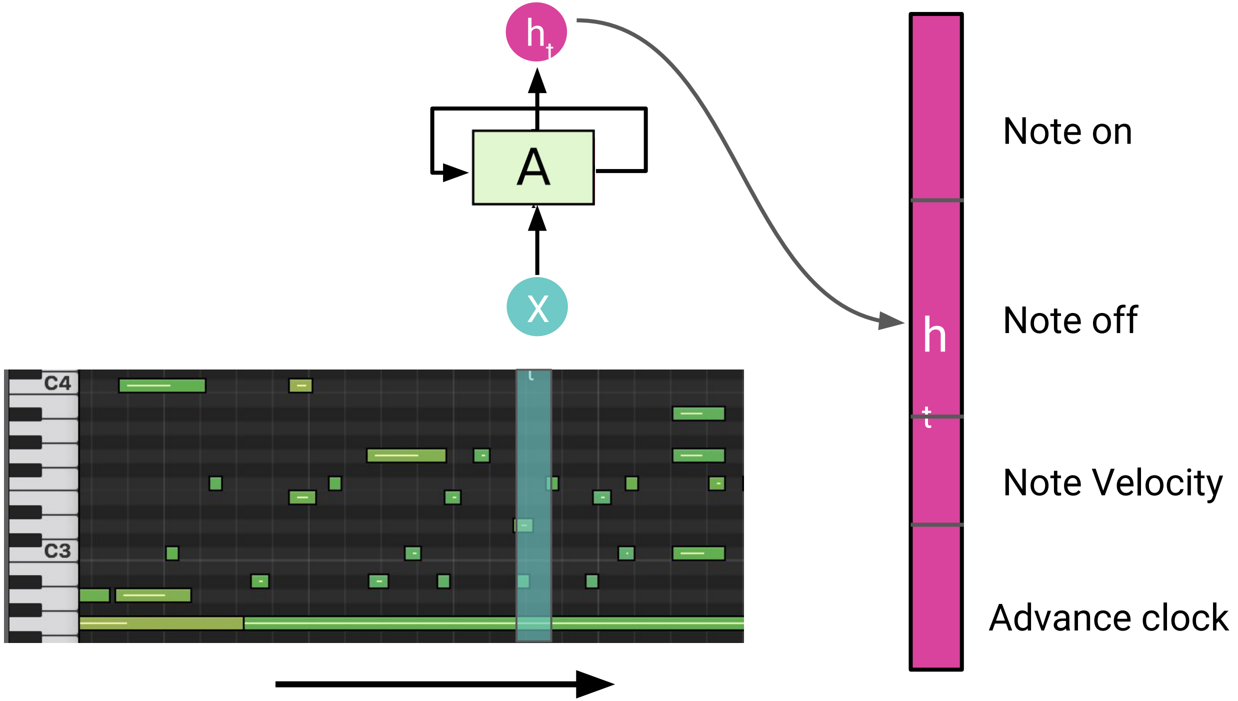
Music



(Image from Simon & Oore, 2016)

Music Language model:

Prior work Performance RNN (Simon & Oore, 2016)



Continuations to given initial motif

Given motif

RNN-LSTM

중간을 넘어가면
모델링을 잘 못한다.

Transformer

Music Transformer

중간을 넘어가도
repetition을 잘한다.

Continuations to given initial motif

Given
motif

Continuations to given initial motif

Given
motif

Continuations to given initial motif

Given motif

RNN-LSTM

Continuations to given initial motif

Given motif

RNN-LSTM

Continuations to given initial motif

Given motif

RNN-LSTM

Transformer

Continuations to given initial motif

Given motif

RNN-LSTM

Transformer

Continuations to given initial motif

Given motif

RNN-LSTM

Transformer

Music Transformer

Continuations to given initial motif

Given motif

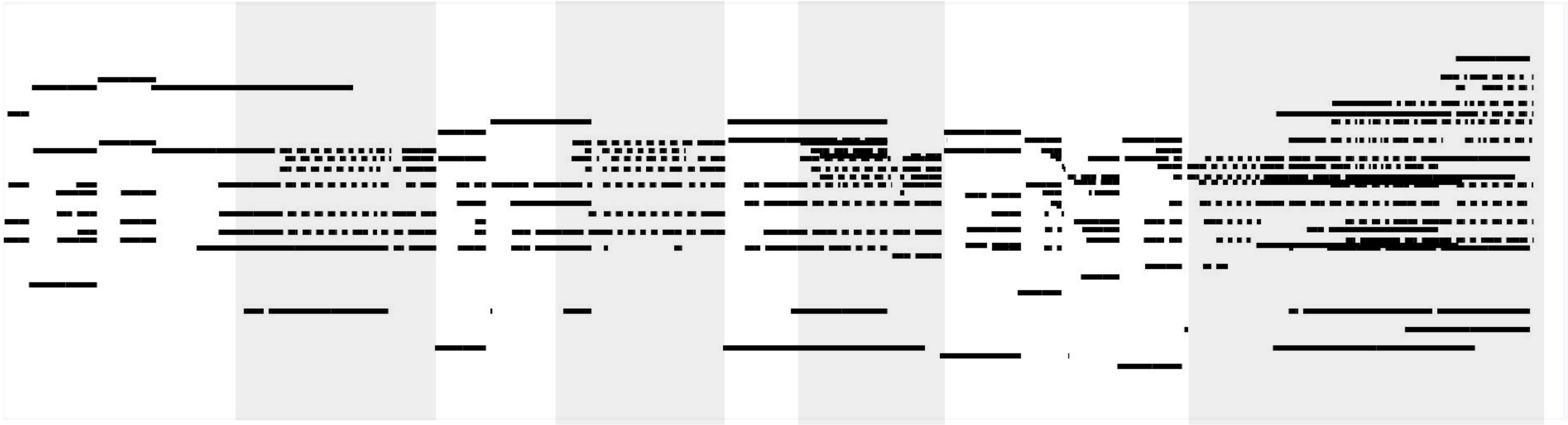
RNN-LSTM

Transformer

Music Transformer

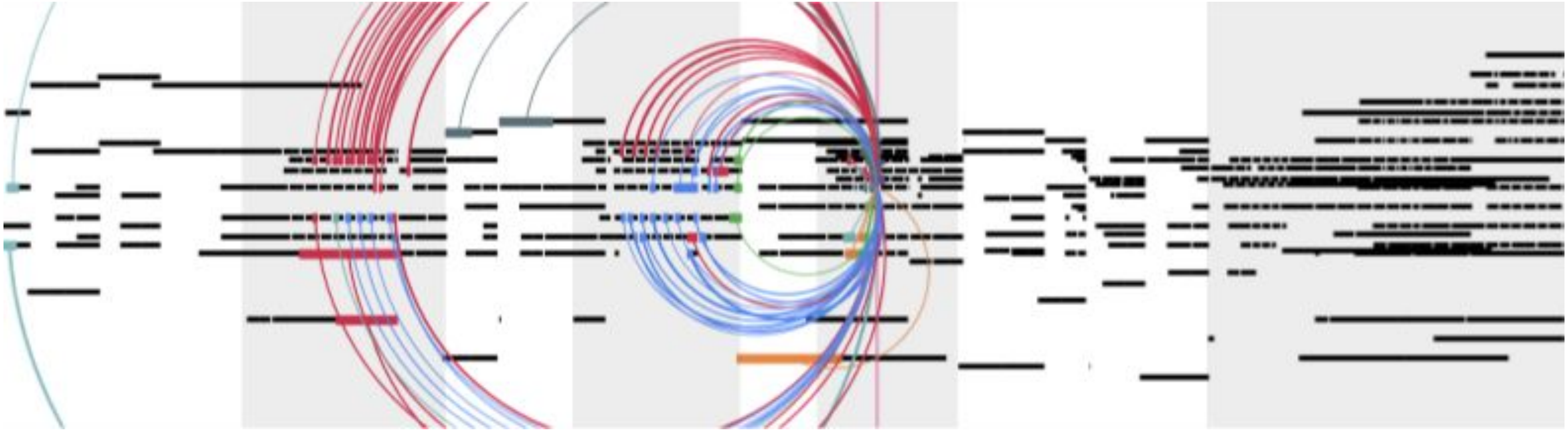
Self-Similarity in Music

there are a log of repetition with gaps in between

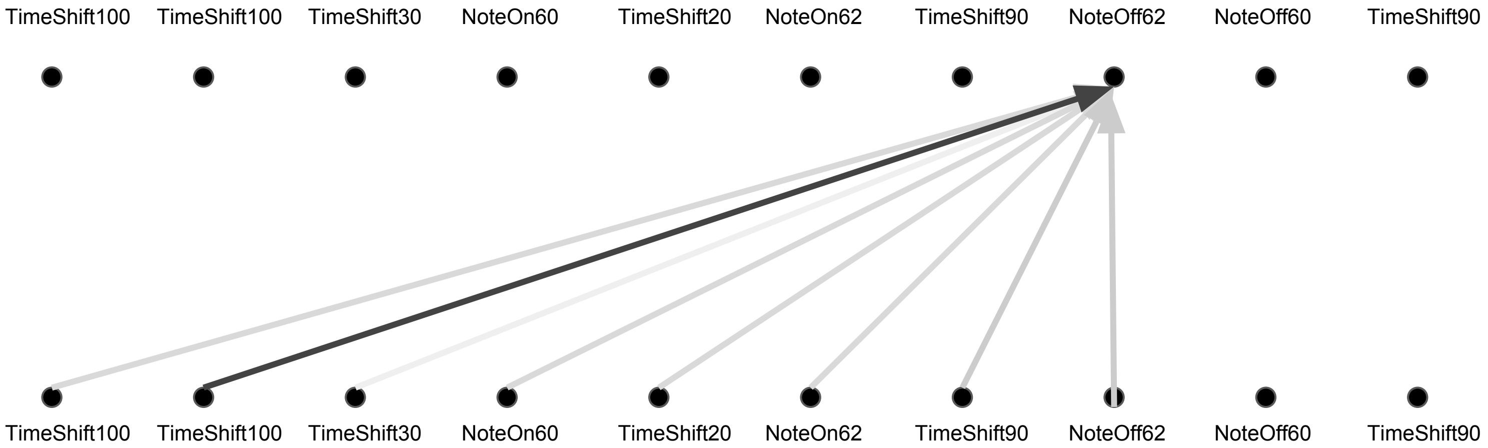


Sample from Music Transformer

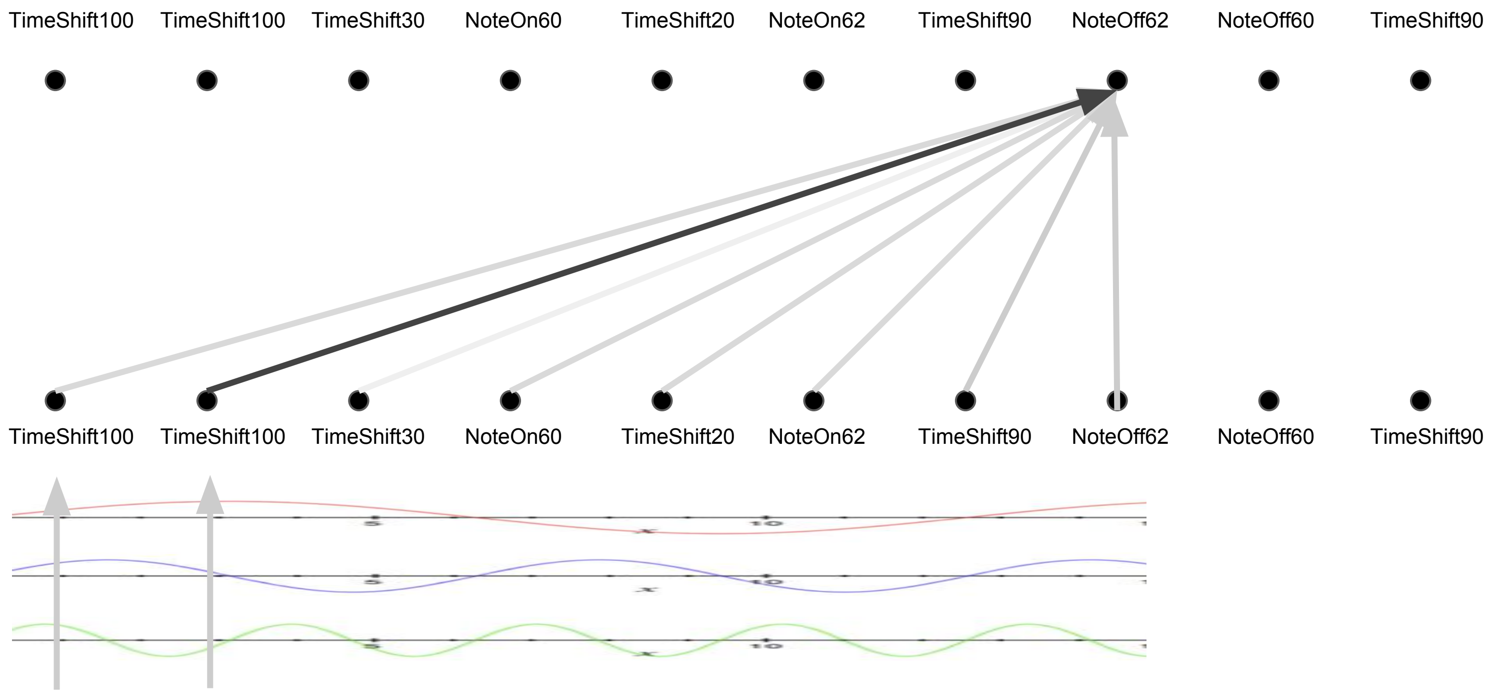
self - attention is from note to note level or event to event level



Attention: a weighted average

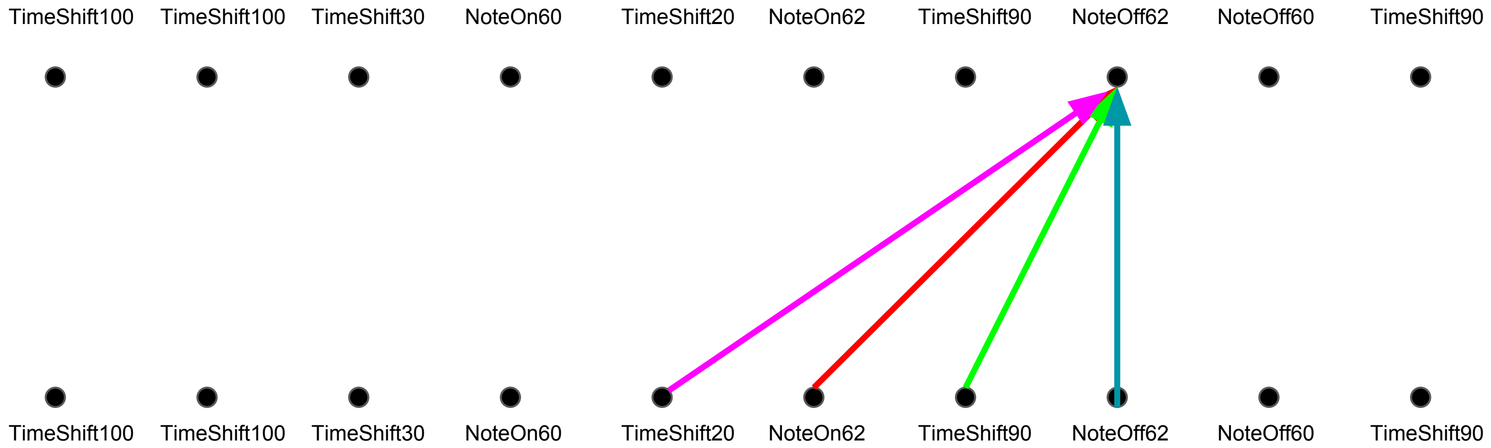


Attention: a weighted average



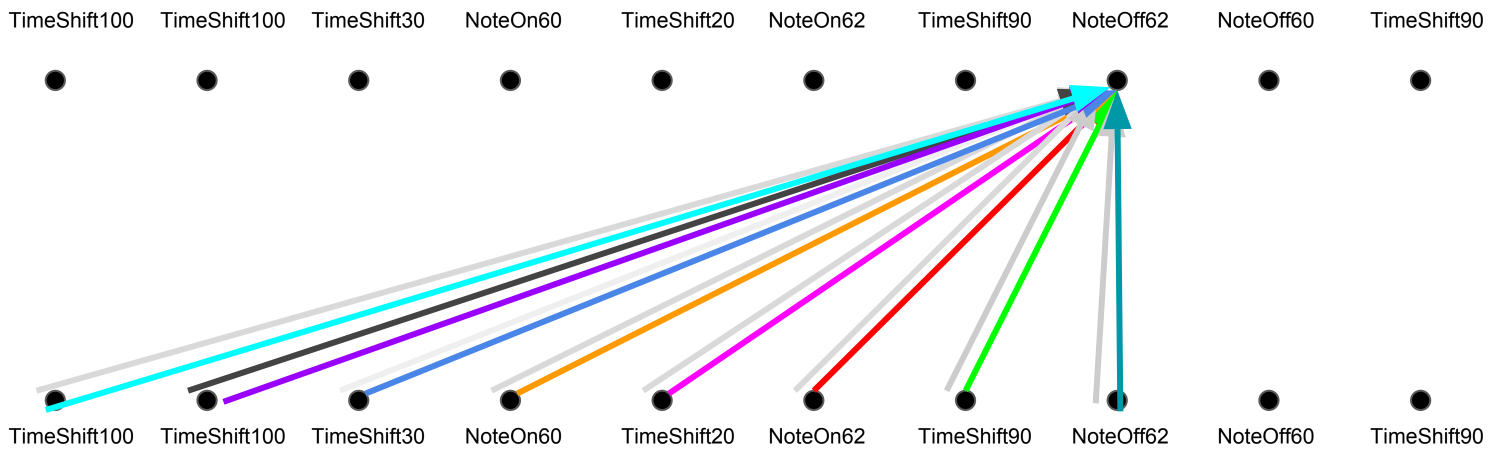
Convolution:

Different linear transformations by relative position.



Relative attention (Shaw et al, 2018)

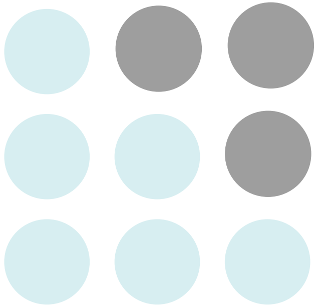
Multihead attention + convolution?



Closer look at attention

Regular Transfomer

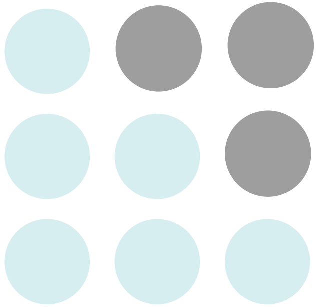
$$softmax(QK^\top)$$



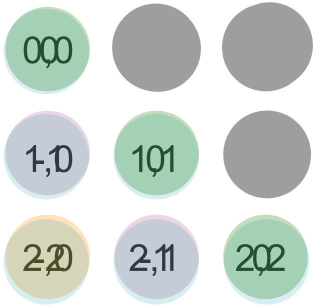
Closer look at relative attention

Regular Transfomer + Relative Attention

$$softmax(QK^\top + Qf(E_{rel}))$$



Modulated by
relative positions



Machine Translation (Shaw et al, 2018)

Model	Position Representation	BLEU En-De	BLEU En-Fr
Transformer Big	Absolute	27.9	41.3
Transformer Big	Relative	**29.2**	**41.5**

영어 - 독일어 번역용 Transformer에서는 입력의 sequence는 비교적 짧다. (50~100 words)

그러나 음악에서는 2000 time - steps에 달하는 range가 있다. 따라서 2000 tokens이 메모리에 저장되어야 한다.

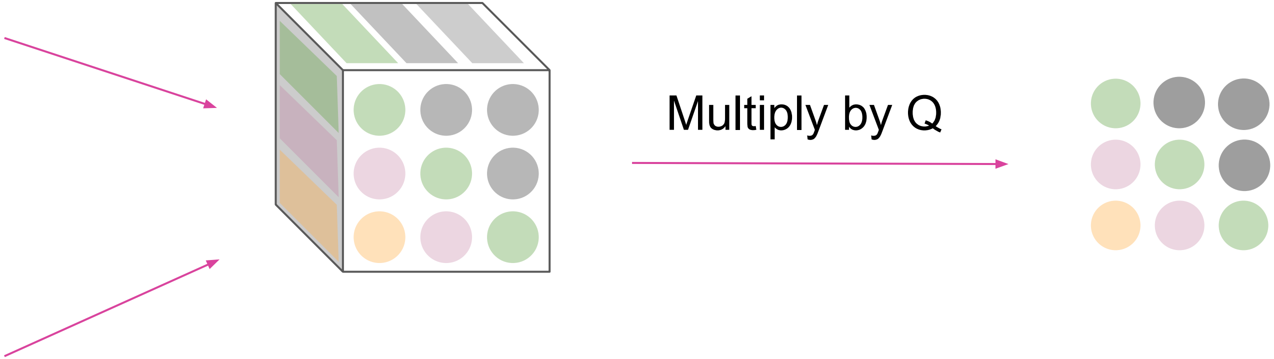
Previous work O(L^2D): **8.5 GB** per layer (Shaw et al, 2018)

Per layer, L=2048, D=512

$$softmax(QK^\top + Qf(E_{rel}))$$

Relative embeddings E_{rel}

$f(E_{rel})$



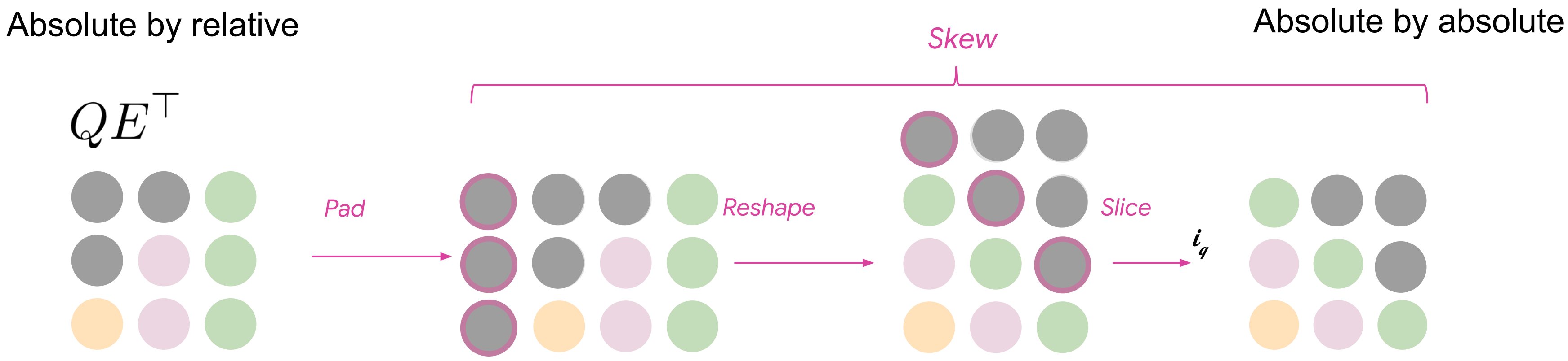
Relative distances

0		
-1	0	
-2	-1	0

Our formulation O(LD): **4.2 MB** per layer

$$softmax(QK^\top + skew(QE_{rel}^\top))$$

Per layer, L=2048, D=512

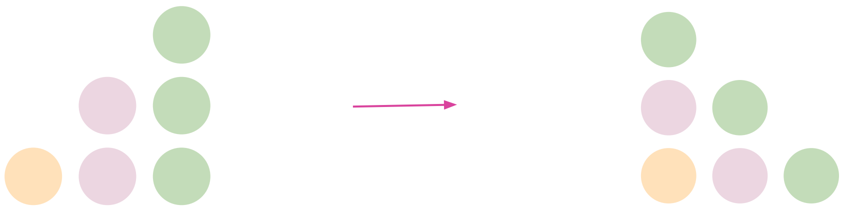


Goal of skewing procedure

Indexed by

absolute by relative

absolute by absolute



Skewing to reduce relative memory from $O(L^2D)$ to $O(LD)$

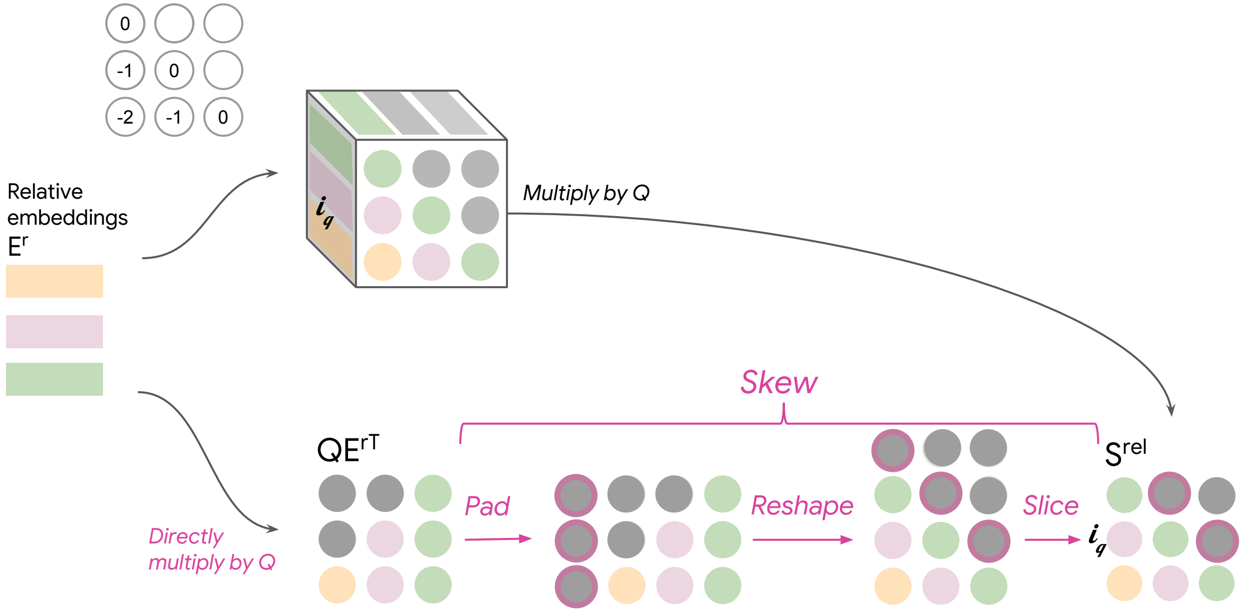
Per layer, L=2048, D=512

Previous work

$O(L^2D)$: **8.5 GB**

Our work

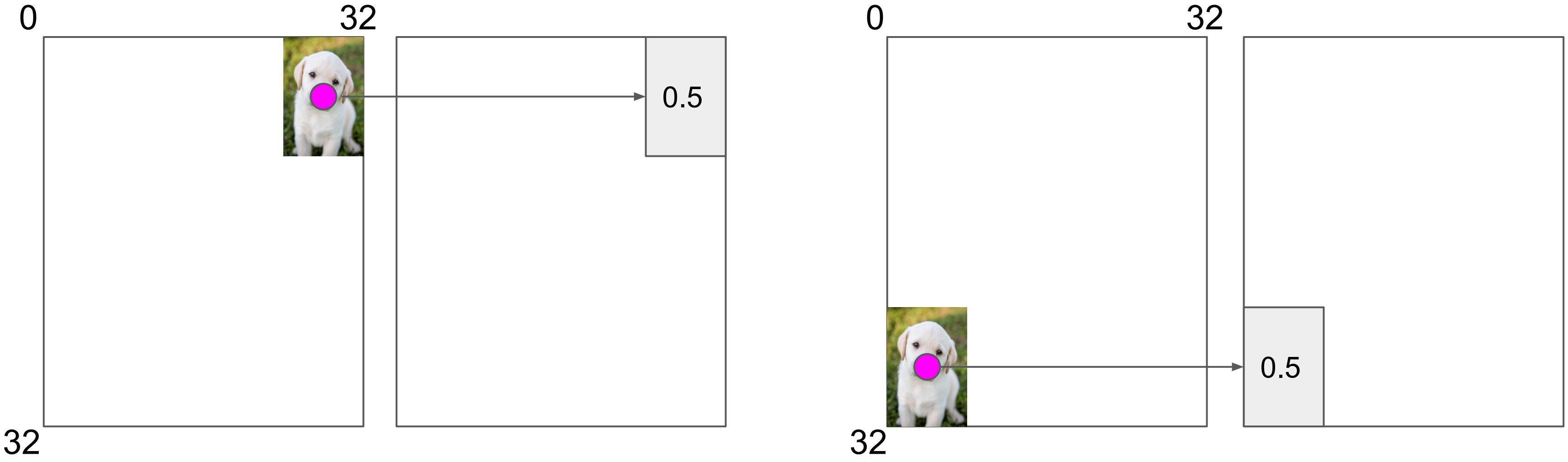
$O(LD)$: **4.2 MB**



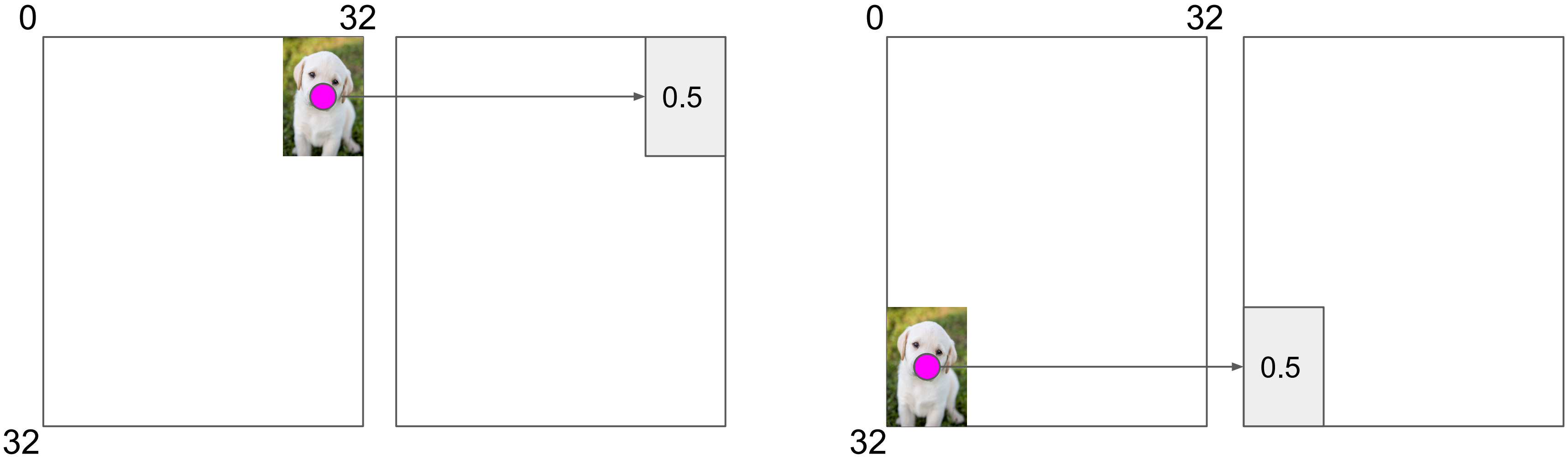
A Jazz sample from Music Transformer

A Jazz sample from Music Transformer

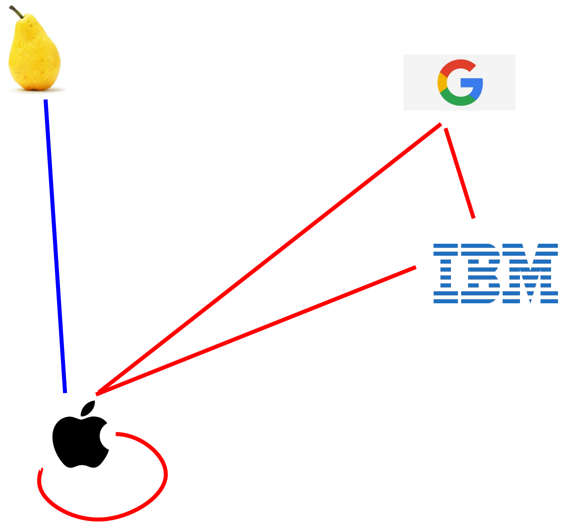
Convolutions and Translational Equivariance



Relative positions Translational Equivariance

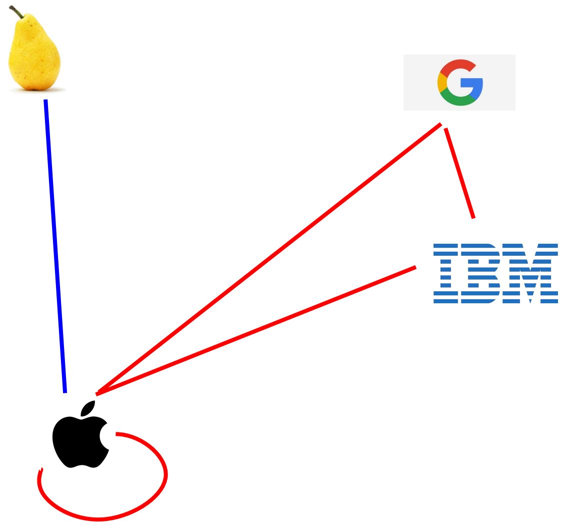


Relative Attention And Graphs

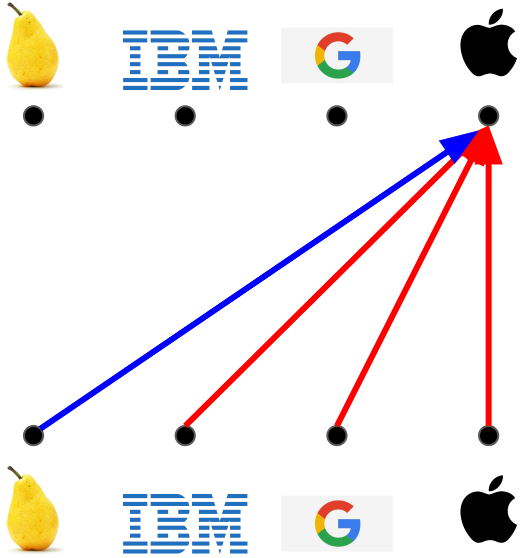


Relative Attention And Graphs

relative attention으로
graph 구조를 잘 모델링
할 수 있다.

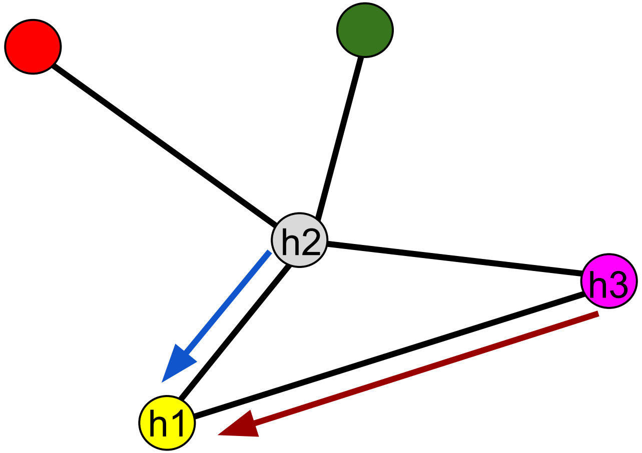


Relational inductive biases, deep learning, and graph networks. (Battaglia et al., 2018)



Self-Attention With Relative Position Representations (Shaw et al., 2018)

Message Passing Neural Networks



$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

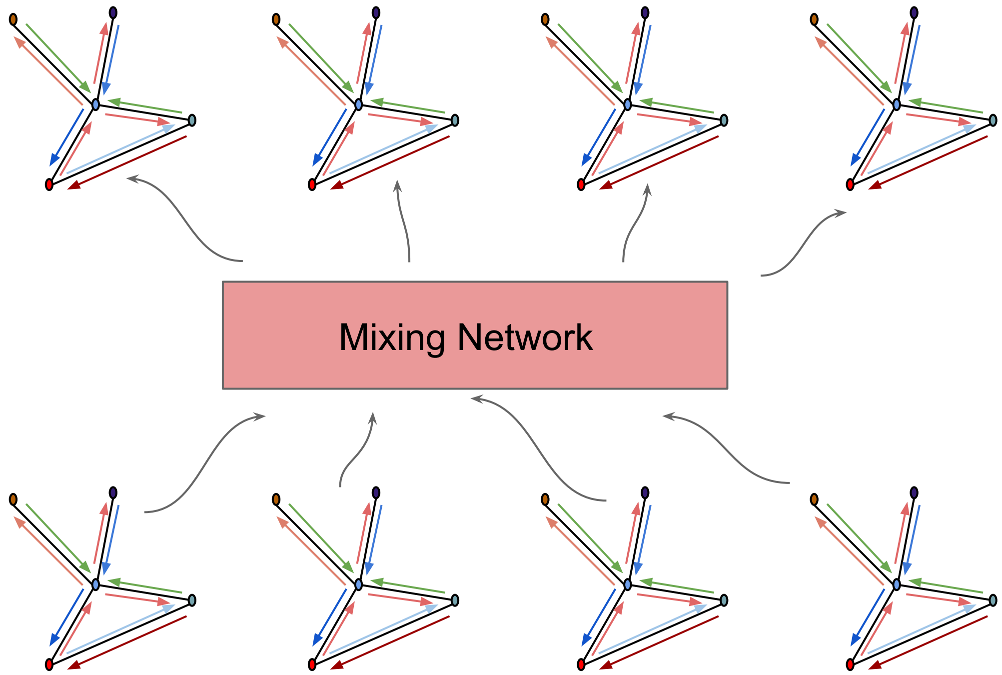
$$\hat{y} = R(\{h_v^T | v \in G\})$$

Neural Message Passing For Quantum Chemistry. Gilmer et al. ICML 2017

Google

Slide credit: Justin Gilmer

Multiple Towers



- Run k smaller copies of the MPNN in parallel.
- Mix node states after each message pass.
- Offers a factor of k speedup for the same node dimension d (> 2x speedup when d=200).
- Also helped improve performance when used with matrix multiply message function.

Slide credit: Justin Gilmer

Graph Library

Code

With Justin Gilmer, Jonathan Frankle, and David Bieber

Self-Attention

Constant ‘path length’ between any two positions.

Unbounded memory.

Trivial to parallelize (per layer).

Models Self-Similarity.

Relative attention provides expressive timing, equivariance, and extends naturally to graphs.

Active Research Area

Non autoregressive transformer (Gu and Bradbury et al., 2018)

Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement (Lee, Manismov, and Cho, 2018)

Fast Decoding in Sequence Models Using Discrete Latent Variables (ICML 2018)
Kaiser, Roy, Vaswani, Pamar, Bengio, Uszkoreit, Shazeer

Towards a Better Understanding of Vector Quantized Autoencoders
Roy, Vaswani, Parmar, Neelakantan, 2018

Blockwise Parallel Decoding For Deep Autogressive Models (NeurIPS 2019)
Stern, Shazeer, Uszkoreit,

Transfer learning

Improving Language Understanding by Generative Pre-Training (Radford, Narsimhan, Salimans, and Sutskever)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin, Chang, Lee, and Toutanova)

Optimization and Large Models

Adafactor: Adaptive Learning Rates with Sublinear Memory Cost (ICML 2018). Shazeer, Stern.

Memory-Efficient Adaptive Optimization for Large-Scale Learning (2019). Anil, Gupta, Koren, Singer.

Mesh-TensorFlow: Deep Learning for Supercomputers (NeurIPS 2019). Shazeer, Cheng, Parmar, Tran, Vaswani, Koanantakool, Hawkins, Lee, Hong, Young, Sepassi, Hechtman) Code (5 billion parameters)

Self-attention in Other Work.

Generating Wikipedia by Summarizing Long sequences. (ICLR 2018). Liu, Saleh, Pot, Goodrich, Sepassi, Shazeer, Kaiser.

Universal Transformers (ICLR 2019). Deghiani*, Gouws*, Vinyals, Uszkoreit, Kaiser.

Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (2019). Dai, Yang, Yang, Carbonell, Le, Salakhutdinov.

A Time-Restricted Self-Attention Layer for ASR (ICASSP 2018). Povey, Hadian, Gharemani, Li, Khudanpur.

Character-Level Language Modeling with Deeper Self-Attention (2018). Roufou*, Choe*, Guo*, Constant*, Jones*

Ongoing and Future Work

Ongoing

Self-supervision and classification for images and video

Understanding Transfer

Future

Multitask learning

Long-range attention