Fast Campus
Online

Life Changing
Education

Chapter 06. 무엇이든 진짜처럼 생성하는 생성 모델(Generative Networks)

Self-Attention For Generative Models

Image Transformer

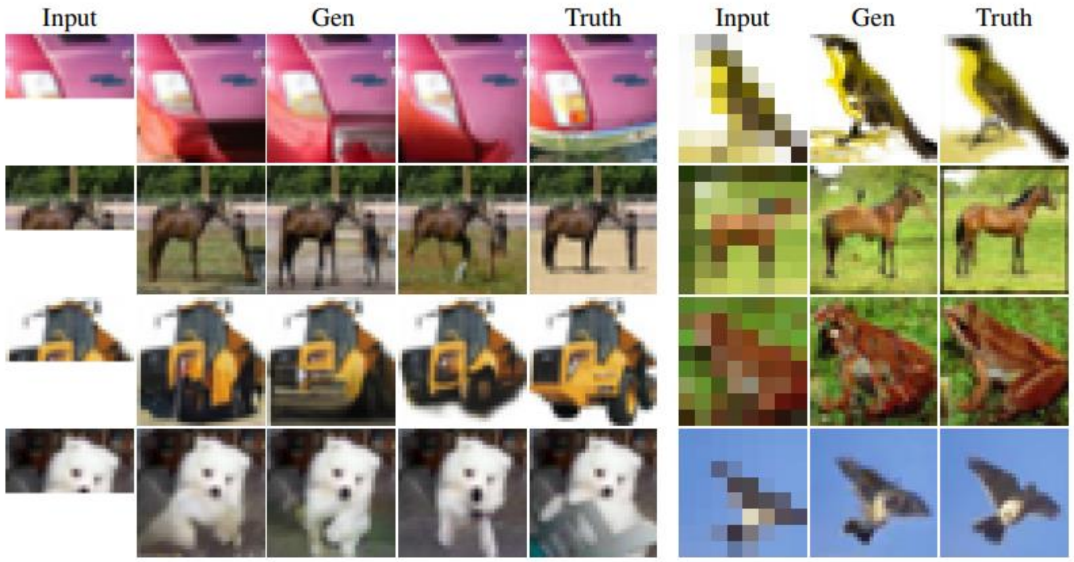
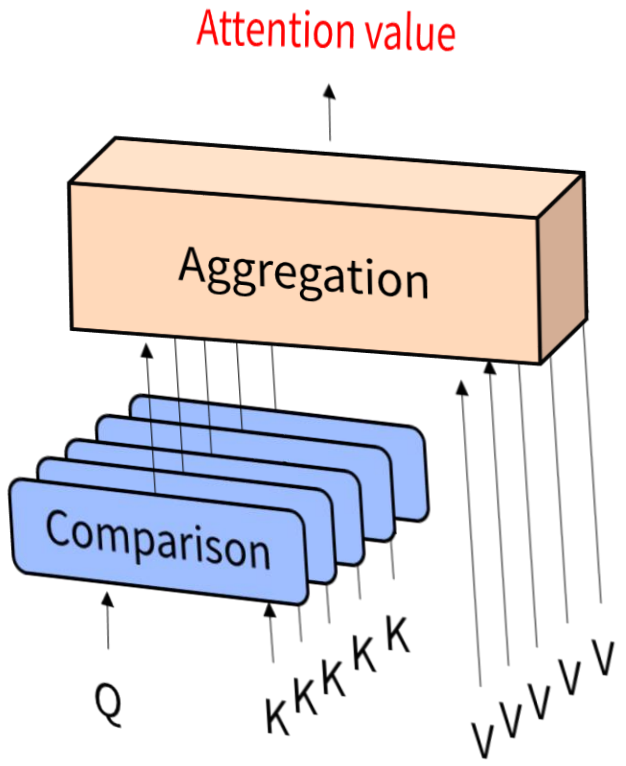


Table 2. On the left are image completions from our best conditional generation model, where we sample the second half. On the right are samples from our four-fold super-resolution model trained on CIFAR-10. Our images look realistic and plausible, show good diversity among the completion samples and observe the outputs carry surprising details for coarse inputs in super-resolution.

Google Brain에서 발표한 강력한 Generative Model인 Image Transformer.

Self-Attention을 적극적으로 활용하여 Convolution을 대체하였다.

Attention Mechanism



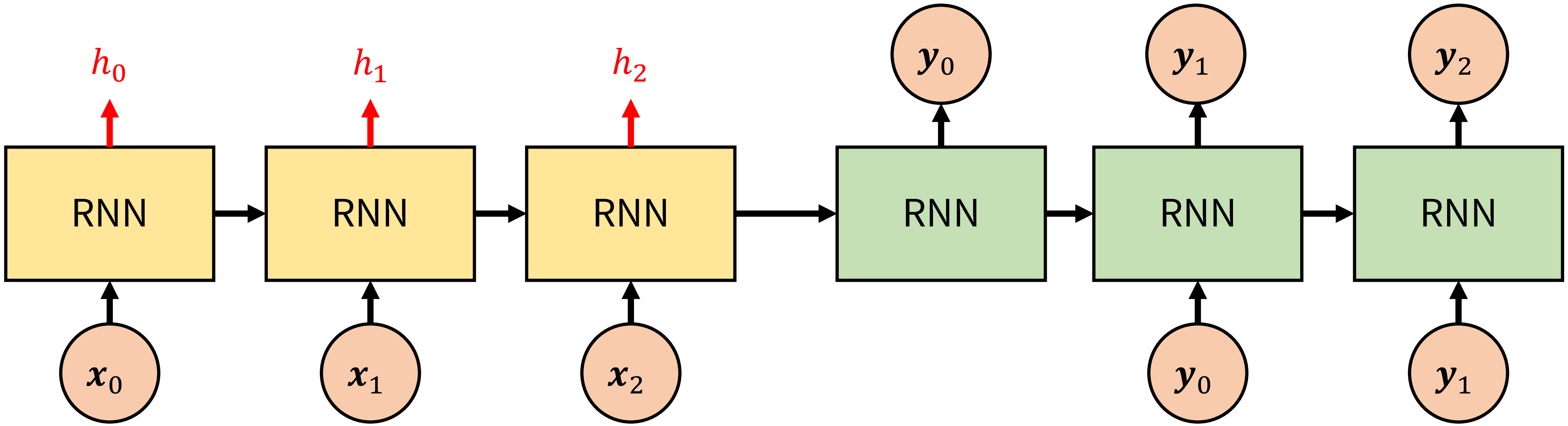
$$\boldsymbol{q} \in \mathbb{R}^n, \boldsymbol{k}_j \in \mathbb{R}^n$$

$$\text{Compare}(\boldsymbol{q}, \boldsymbol{k}_j) = \boldsymbol{q} \cdot \boldsymbol{k}_j = \boldsymbol{q}^T \boldsymbol{k}_j$$

$$\text{Aggregate}(\boldsymbol{c}, V) = \sum_j c_j \boldsymbol{v}_j$$

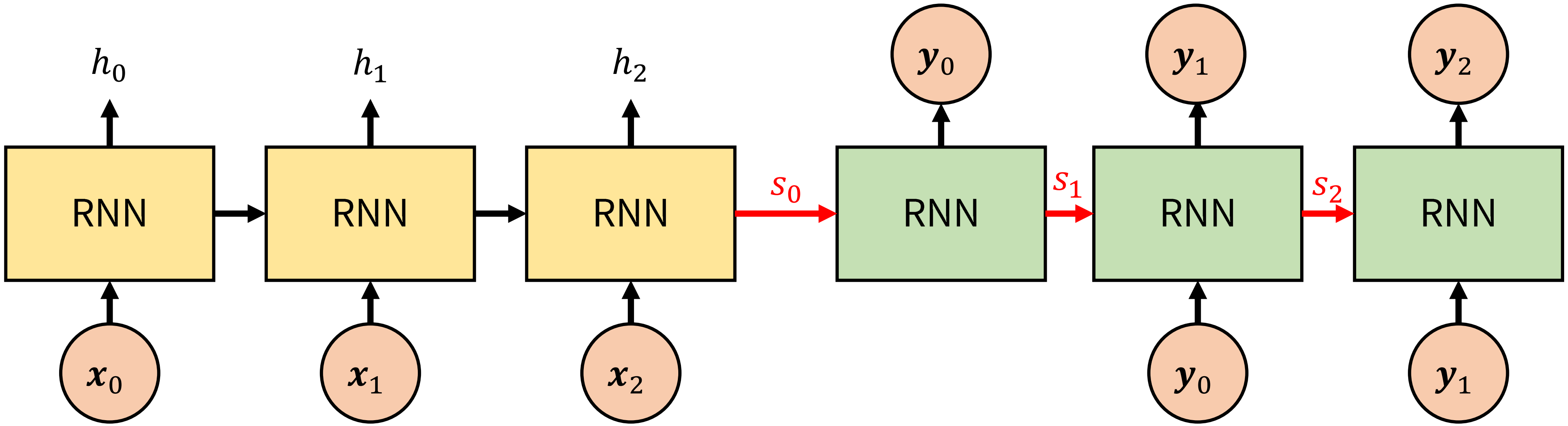
Attention Mechanism을 기억하십니까? 간단히 복습하고 넘어갑시다.

Seq2seq – Key-Value



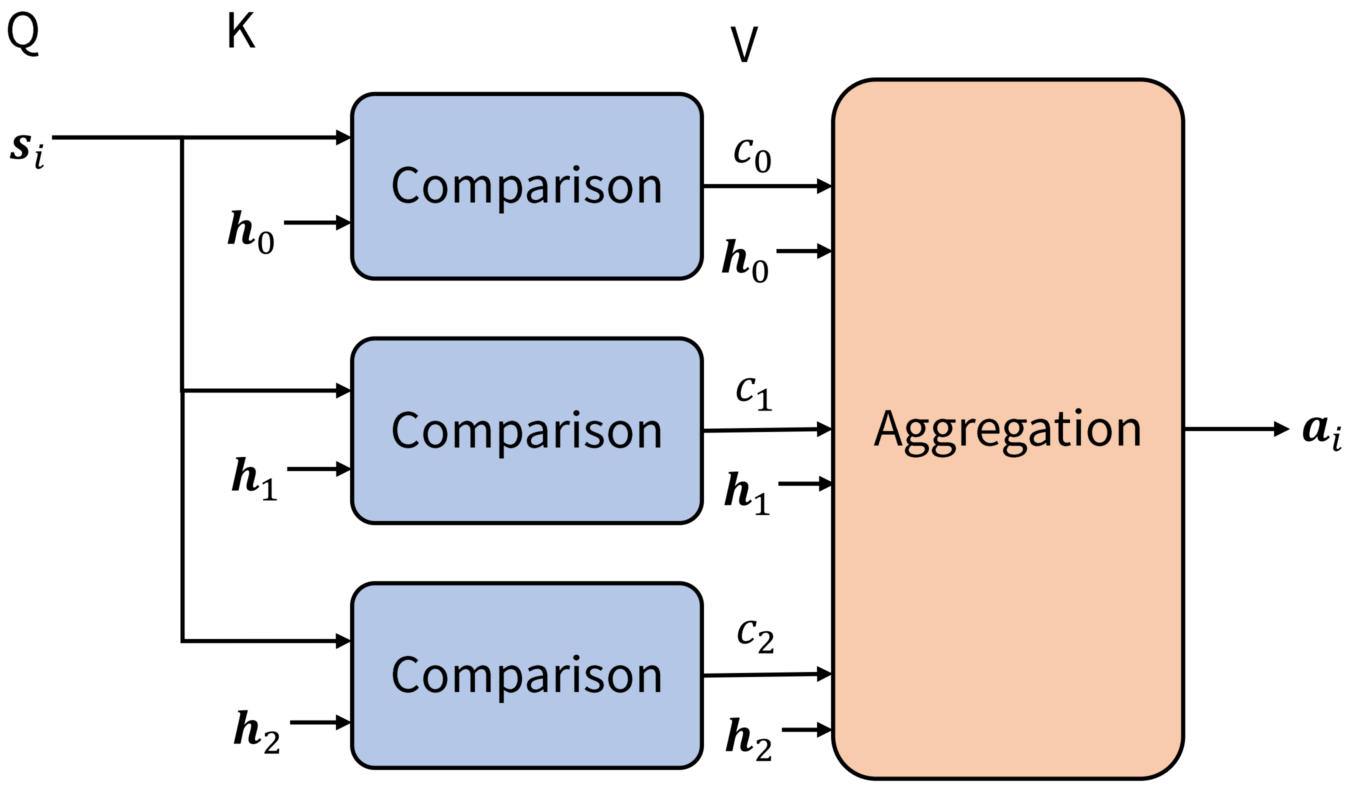
Seq2seq에서는 Encoder의 hidden layer들을 key와 value로 사용한다.

Seq2seq – Query



Seq2seq에서는 Decoder의 hidden layer들을 Query로 사용한다.

Seq2seq – Attention Module (2/2)

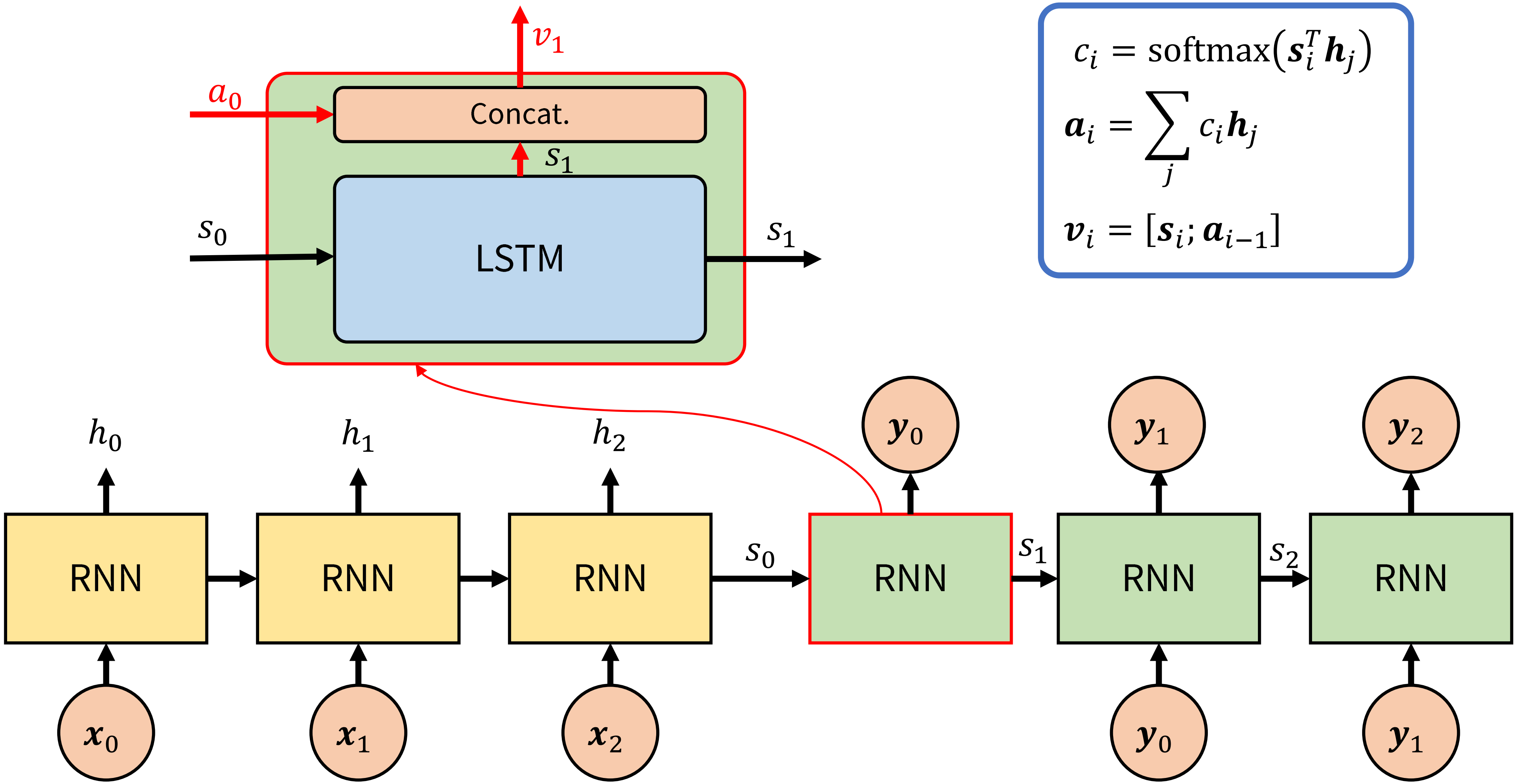


$$c_i = \text{softmax}\left(s_i^T h_j\right)$$

$$a_i = \sum_j c_i h_j$$

블록도에 비해 수식이 오히려 간단하다. 비교 함수와 결합 함수의 의미를 잘 이해하자.

Seq2seq – Attention Module (2/2)



$$c_i = \mathrm{softmax}(\boldsymbol{s}_i^T \boldsymbol{h}_j)$$

$$\boldsymbol{a}_i = \sum_j c_i \boldsymbol{h}_j$$

$$\boldsymbol{v}_i = [\boldsymbol{s}_i; \boldsymbol{a}_{i-1}]$$

Hidden state에 attention value를 concatenate까지 하면 모든 수식적 표현이 끝난다.

The Transformer

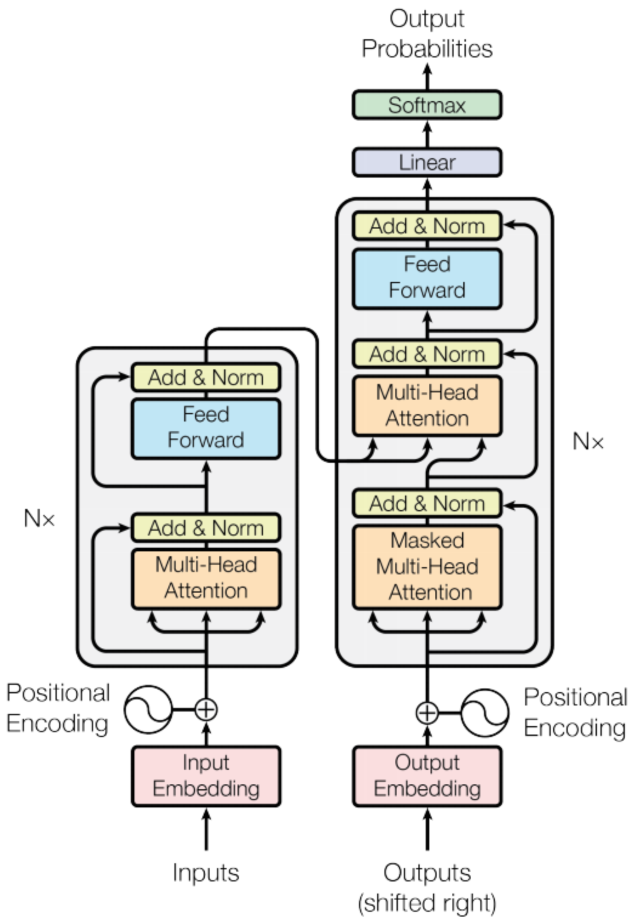


Figure 1: The Transformer - model architecture.

Transformer 구조를 기억하십니까? 간략하게만 짚고 넘어갑시다.

Inputs & Output

$Y = [y_0, y_1, \dots y_m]$

출력 Sequence 길이 m

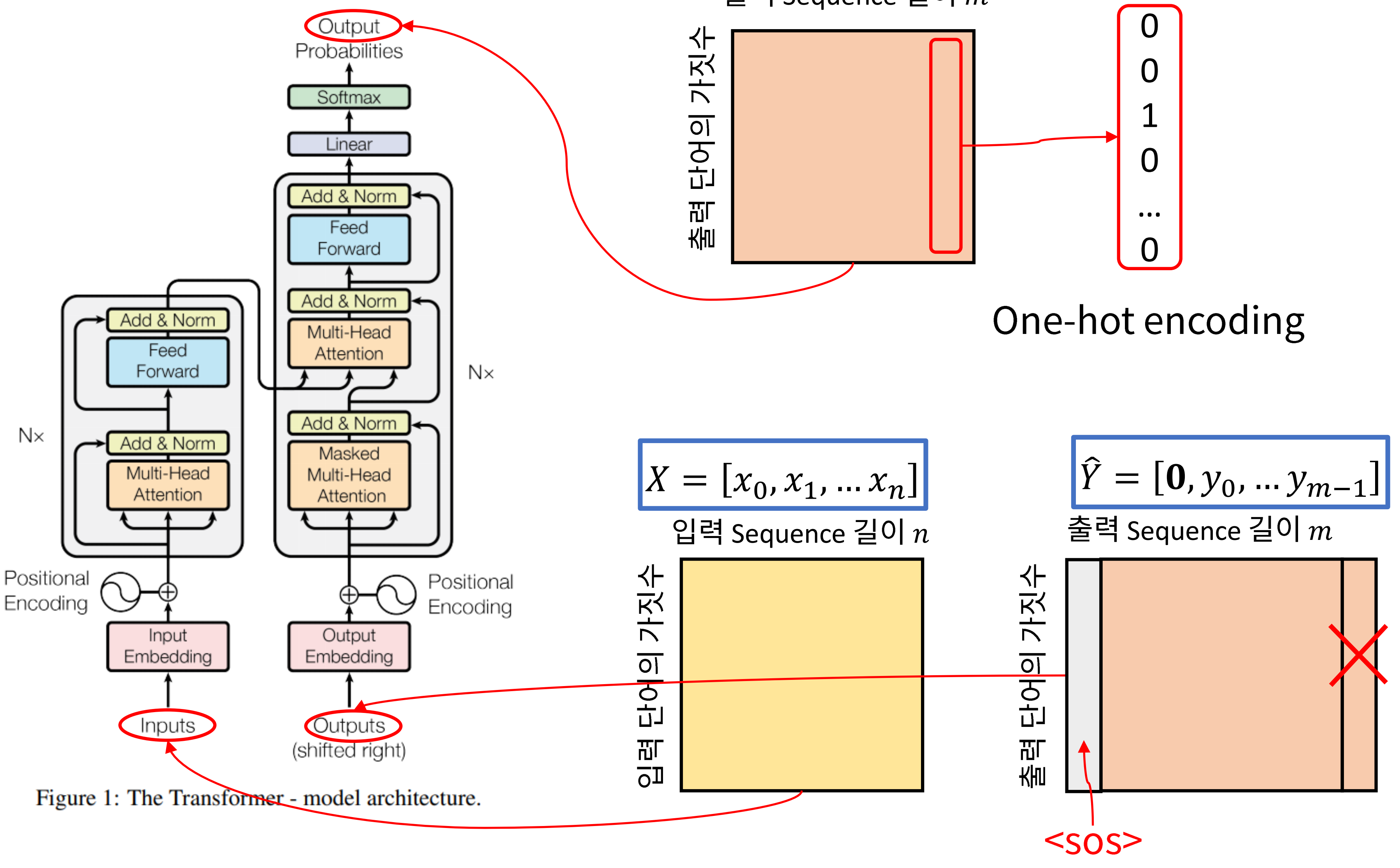
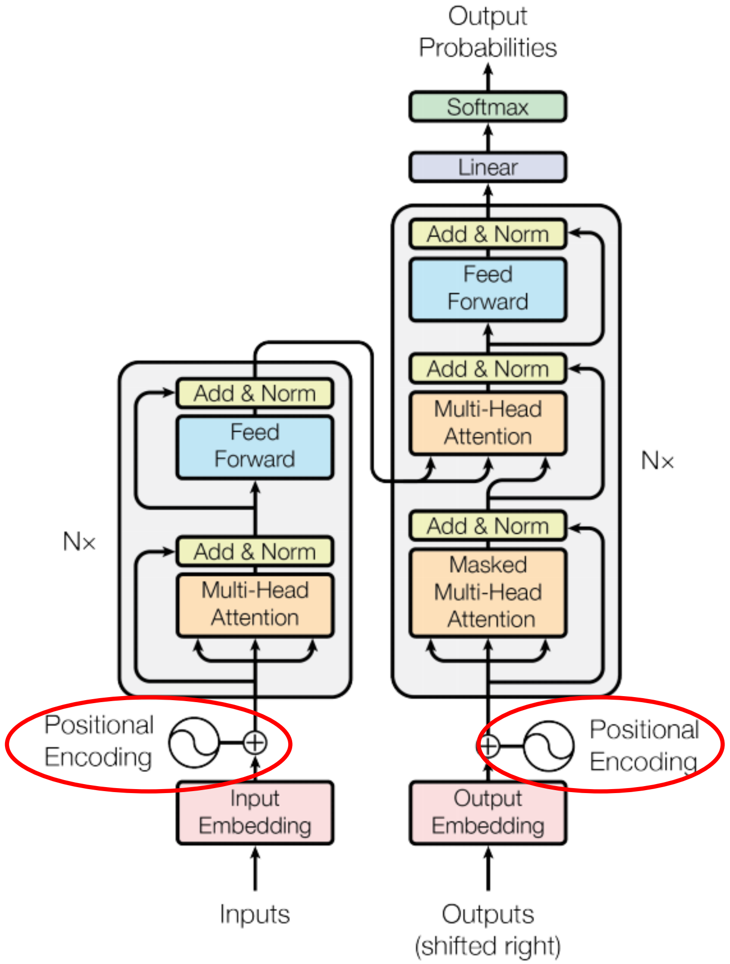


Figure 1: The Transformer - model architecture.

Positional Encoding



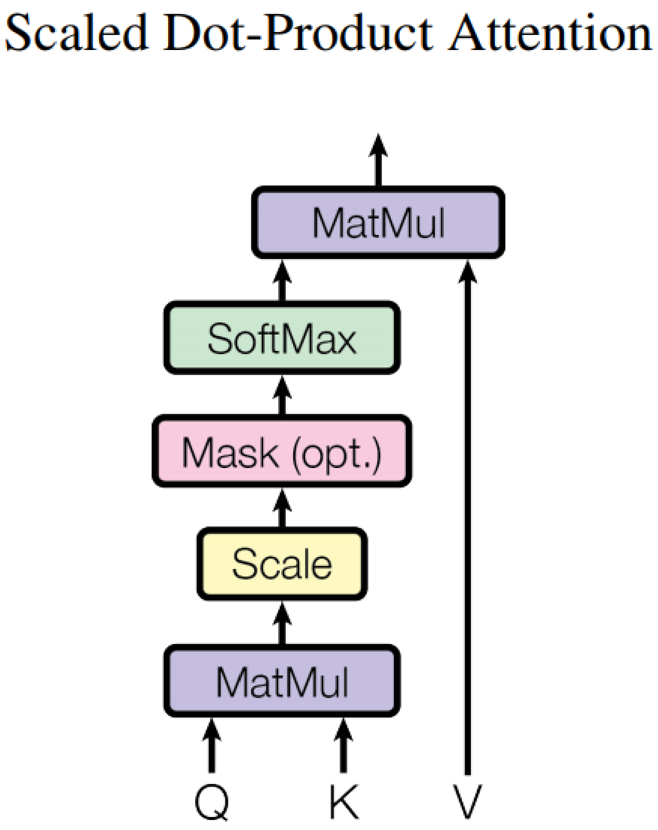
- 시간적 위치별로 고유의 Code를 생성하여 더하는 방식
- 전체 Sequence의 길이 중 상대적 위치에 따라 고유의 벡터를 생성하여 Embedding된 벡터에 더해줌

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

Position별로 구분되는 Encoding을 얻게 됨

pos: 상대적 위치, i: 벡터의 element 인덱스

Scaled Dot-Product Attention



- Query, Key-Value의 구조를 띄고 있음
- Q와 K의 비교 함수는 Dot-Product와 Scale로 이루어짐
- Mask를 이용해 Illegal connection의 attention을 금지
- Softmax로 유사도를 0 ~ 1의 값으로 Normalize
- 유사도와 V를 결합해 Attention value 계산

$$Q = [\boldsymbol{q}_0, \boldsymbol{q}_1, \dots, \boldsymbol{q}_n]$$

$$K = [\boldsymbol{k}_0, \boldsymbol{k}_1, \dots, \boldsymbol{k}_n]$$

$$V = [\boldsymbol{v}_0, \boldsymbol{v}_1, \dots, \boldsymbol{v}_n]$$

$$C = \text{softmax}\left(\frac{K^T Q}{\sqrt{d_k}}\right)$$

$$\boxed{\boldsymbol{a} = C^T V = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V}$$

Image Transformer

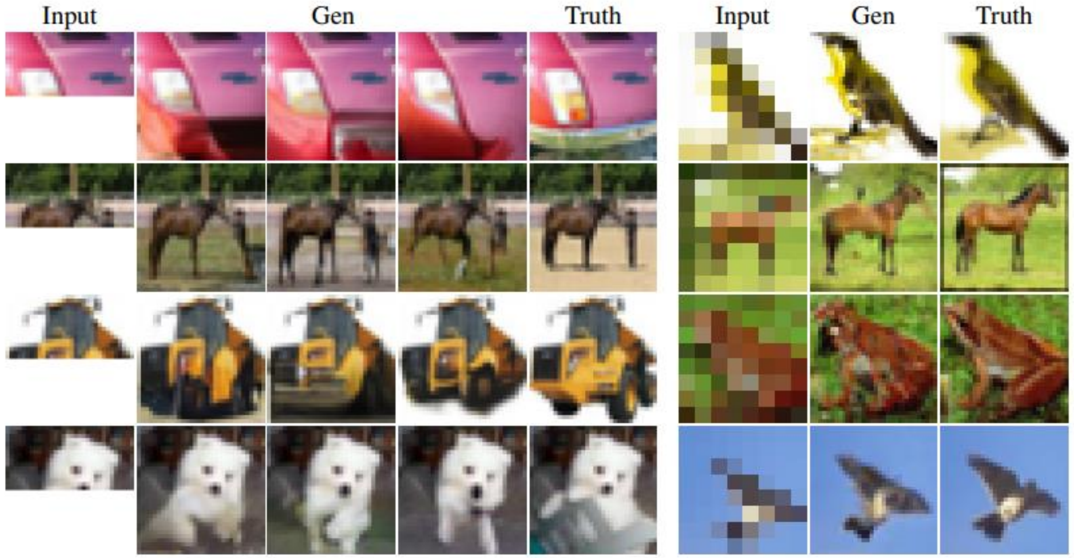
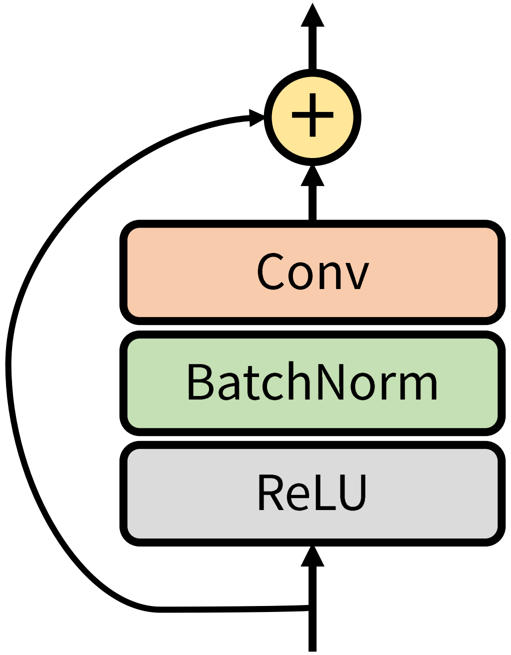


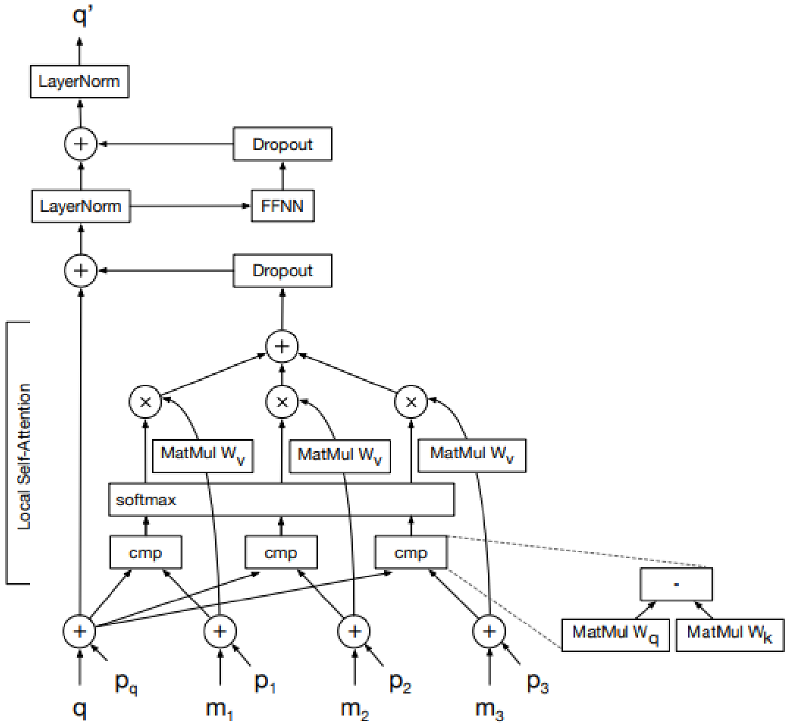
Table 2. On the left are image completions from our best conditional generation model, where we sample the second half. On the right are samples from our four-fold super-resolution model trained on CIFAR-10. Our images look realistic and plausible, show good diversity among the completion samples and observe the outputs carry surprising details for coarse inputs in super-resolution.

다시 원래 주제로 돌아와서, Image Transformer의 구조를 이해해 보자.

Local Self-Attention



Conventional ResNet

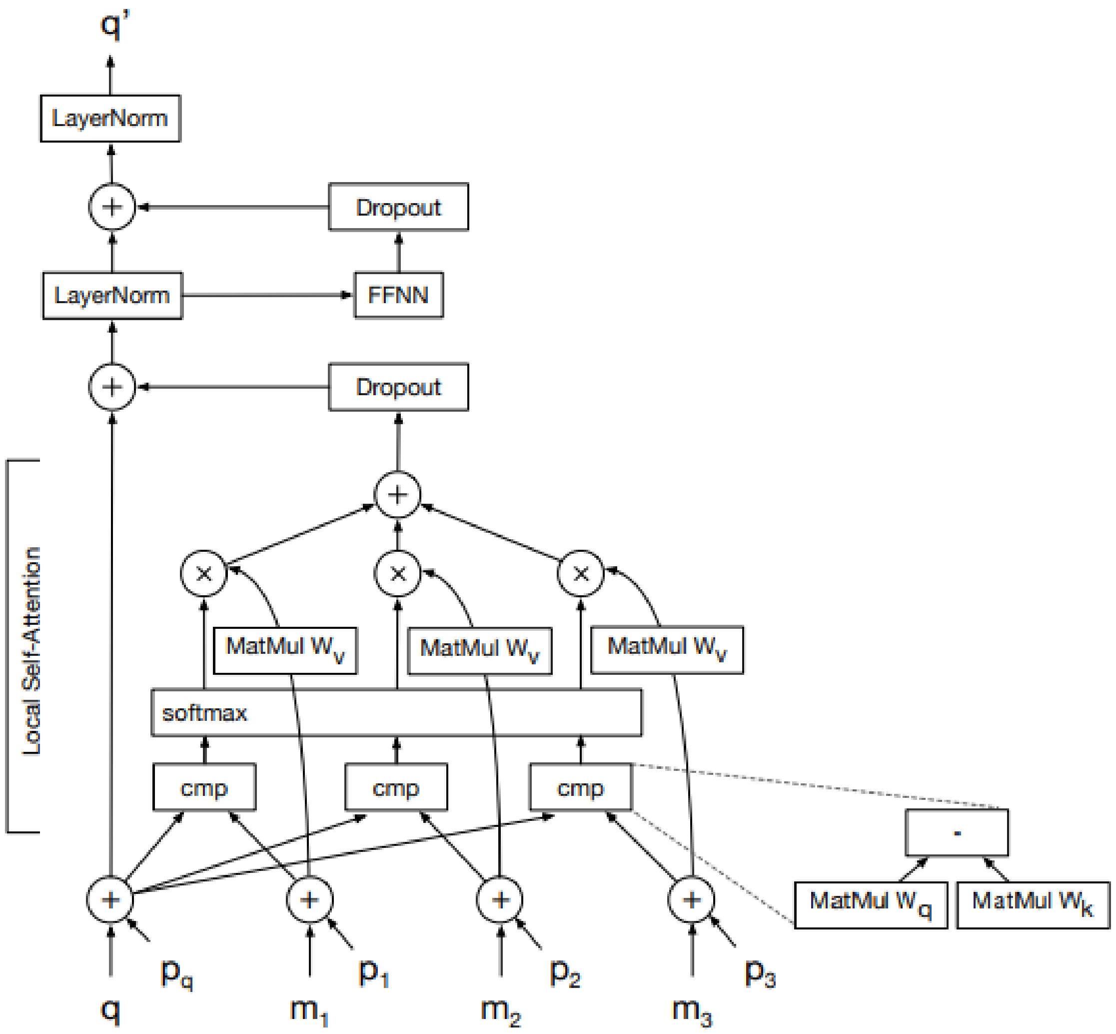


Local Self-Attention Layer

기존의 Convolutional Layer를 Self-Attention Layer로 교체하였다.

RNN을 Transformer로 대체한 것 처럼, CNN을 Image Transformer로 교체한 것.

Local Self-Attention – Equation

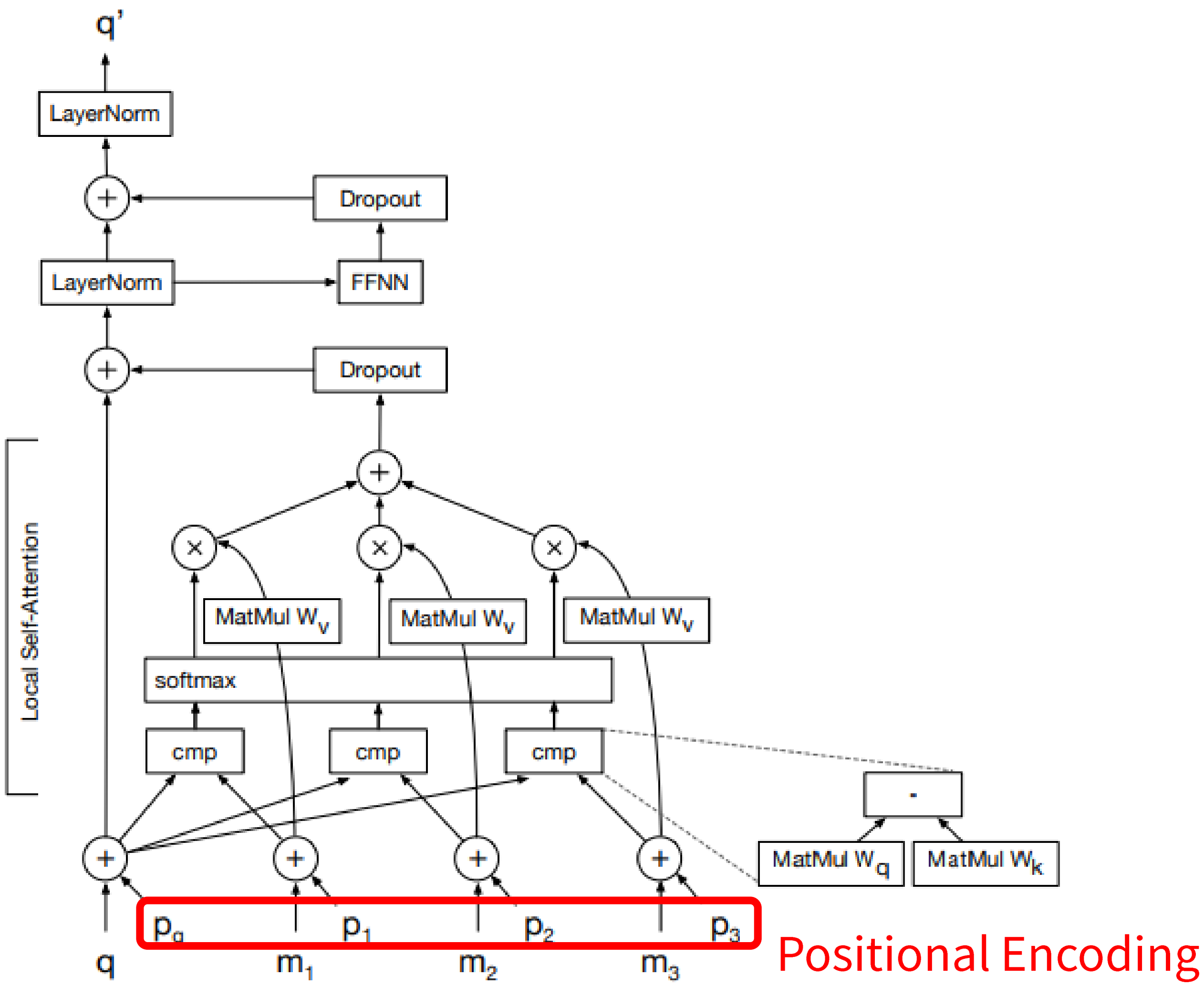


$$q_a = \text{layernorm}(q + \text{dropout}(\text{softmax}\left(\frac{W_q q (MW_k)^T}{\sqrt{d}}\right) MW_v)) \quad (1)$$

$$q' = \text{layernorm}(q_a + \text{dropout}(W_1 \text{ReLu}(W_2 q_a))) \quad (2)$$

Learned Embedding 등, Transformer에서 차용한 부분이 상당히 많다.

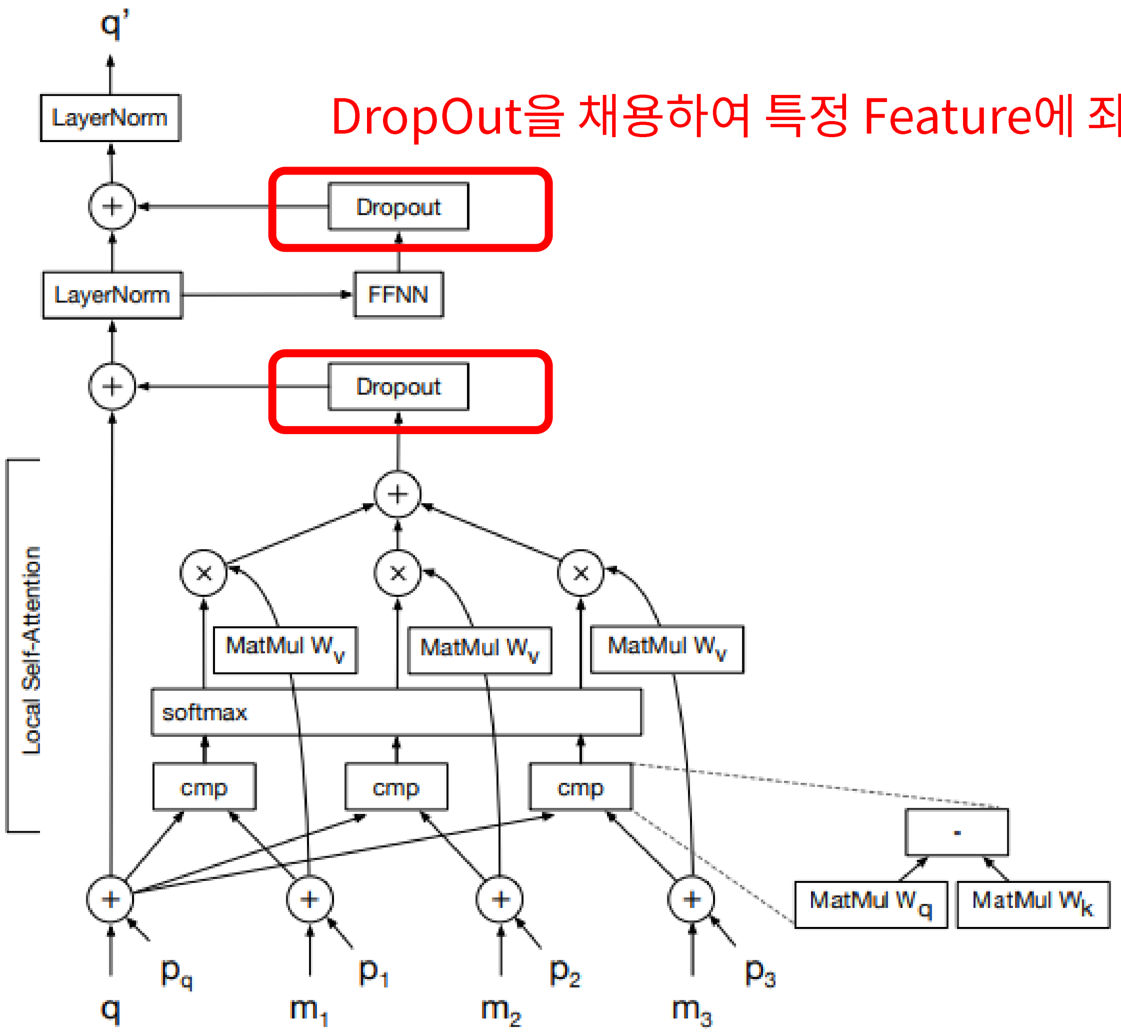
Positional Encoding



Transformer와 유사한 방식의 PE를 사용하였다.

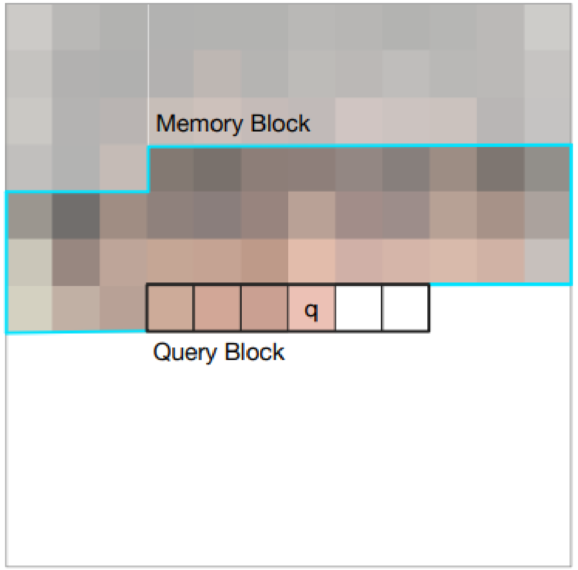
DropOut

DropOut을 채용하여 특정 Feature에 좌우되지 않도록 하였다.

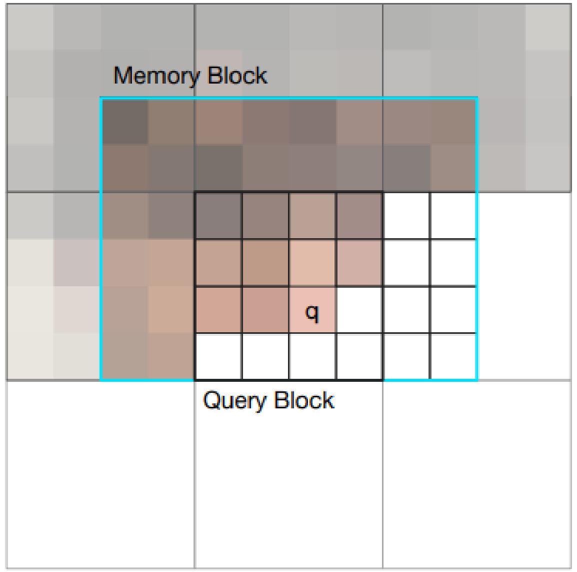


Local 1D and 2D Attention

Local 1D Attention



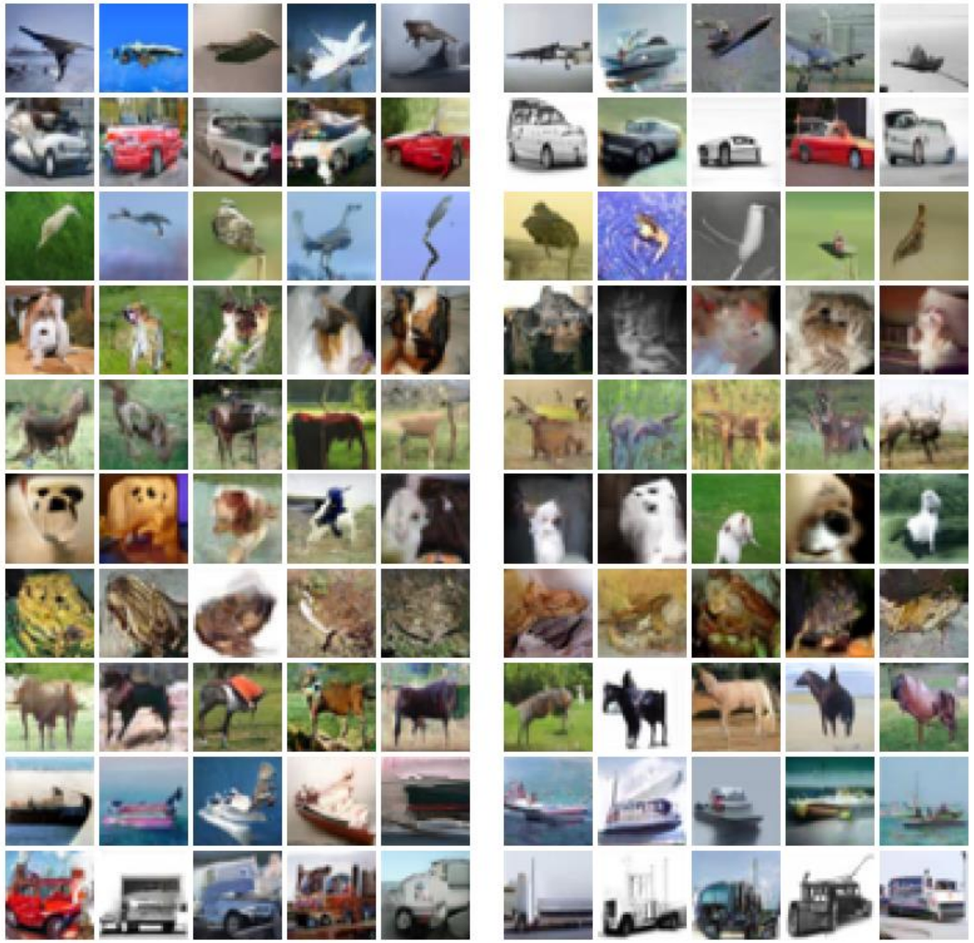
Local 2D Attention



Memory-Efficient한 구현을 위해, Memory Block과 Query Block을 구분하였다.

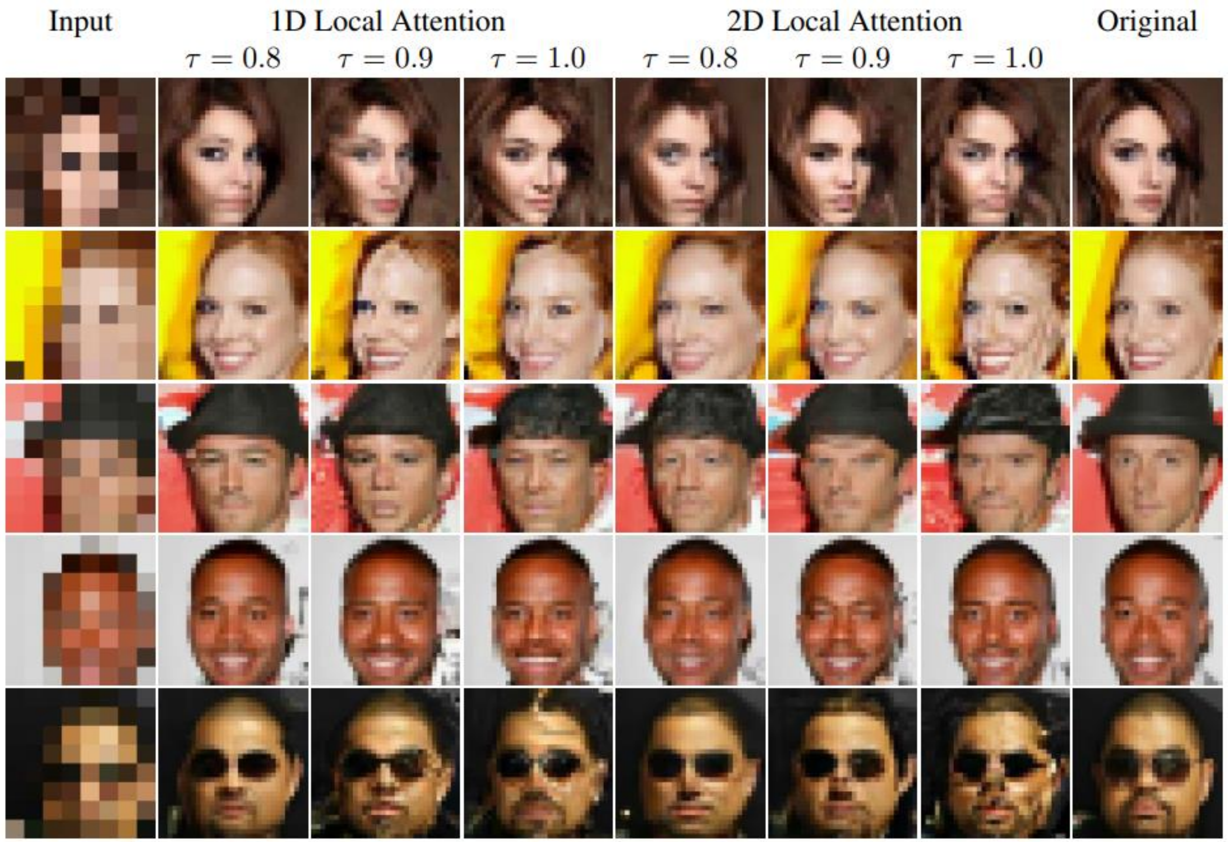
아직 생성되지 않은 부분은 연산하지 않도록 Masking하여 속도를 높였다.

Generation Results



기존 Autoregressive Model에 비해 훨씬 그럴듯한 결과를 보여준다.

Super-Resolution Results



CelebA 데이터셋에서 실험한 Super-Resolution 결과. 놀라운 수준의 Detail을 보여준다.