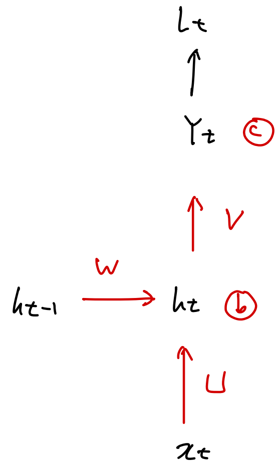
Vanilla RNN

$$\begin{cases} h_t = \sigma(W h_{t-1} + U x_t + b) \\ Y_t = V h_t + c \end{cases}$$

$$\begin{cases} L_t = \frac{1}{2}(Y_t - Z_t)^2 \\ L = \sum L_t \end{cases}$$



$$\frac{\partial L}{\partial V} = \sum_t \frac{\partial L_t}{\partial Y_t} \cdot \frac{\partial Y_t}{\partial V}$$

$$\frac{\partial L_t}{\partial Y_t} = Y_t - Z_t \qquad \frac{\partial Y_t}{\partial V} = h_t$$

$$\frac{\partial L}{\partial c} = \sum_t \frac{\partial L_t}{\partial Y_t} \cdot \frac{\partial Y_t}{\partial c}$$

$$\frac{\partial L_t}{\partial Y_t} = Y_t - Z_t \qquad \frac{\partial Y_t}{\partial c} = 1$$

Backprop Through Time

$$\frac{\partial L}{\partial W} = \left(\frac{\partial L_1}{\partial h_1} \cdot \frac{\partial h_1}{\partial W}\right) + \boxed{\left(\frac{\partial L_2}{\partial h_2} \cdot \frac{\partial h_2}{\partial W}\right) + \left(\frac{\partial L_2}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W}\right)}$$

$$\frac{\partial h_L}{\partial W} = \sum_{\ell=1}^{L} \left(\frac{\partial h_{\ell+1}}{\partial h_\ell} \cdots \frac{\partial h_L}{\partial h_{L-1}}\right) \cdot \frac{\partial h_\ell}{\partial W}$$

$$\frac{\partial h_L}{\partial h_{L-1}} = (1-h_L^2) \cdot W \qquad \frac{\partial h_\ell}{\partial W} = (1-h_\ell^2)\, h_{\ell-1}$$

Note

$$(\tanh(x))' = 1 - \tanh^2(x)$$

$$\left|\frac{\partial h_s}{\partial h_{s-1}}\right| > 1 \text{ then } (\#)^{100} \to \infty \Rightarrow$$

clipping : $(\text{threshold}) \cdot \frac{g}{\|g\|}$

$$\left|\frac{\partial h_s}{\partial h_{s-1}}\right| < 1 \text{ then } (\#)^{100} \to 0$$

$$\frac{\partial L}{\partial h_\ell} = \frac{\partial L}{\partial h_{\ell+1}} \cdot \frac{\partial h_{\ell+1}}{\partial h_\ell}$$

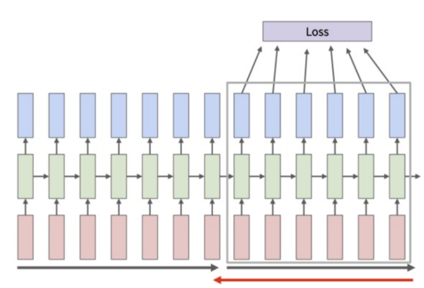
$\approx I$ then grad can flow

LSTM

GRU

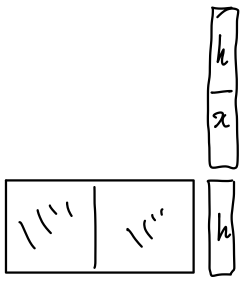
Note.

Truncated Backprop.



Vanilla RNN

$$h_t = \tanh\left(W\begin{pmatrix}h_{t-1}\\ x\end{pmatrix}\right)$$

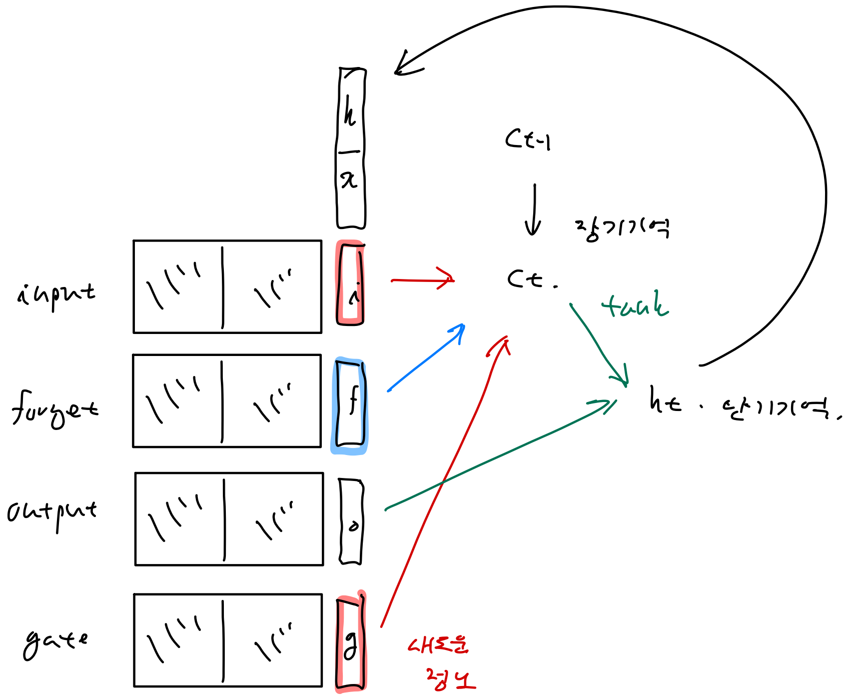


LSTM

$$\begin{pmatrix}i\\ f\\ o\\ g\end{pmatrix} = \begin{pmatrix}\sigma\\ \sigma\\ \sigma\\ \tanh\end{pmatrix} W \begin{pmatrix}h_{t-1}\\ x_t\end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$



- $$\frac{\partial L}{\partial c_t} = \frac{\partial L}{\partial c_{t+1}} \frac{\partial c_{t+1}}{\partial c_t}$$

- $$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot g_{t+1}$$

  - $$\frac{\partial c_{t+1}}{\partial c_t} = f_{t+1} + (\cdots)$$

  - 중요한 정보에 대해 $f_{t+1} \approx 1$

    uninterrupted grad flow

$h_t = LSTM(z_t, h_{t-1}, c_{t-1} ; \theta)$

Note. One-Hot Encoding

"a" $[0 \ 0 \ 1]$

"man" $[0 \ 1 \ 0]$

⋱

- $z_t = W_{emb} \, x_t$

one-hot

- $\hat{x}_{t+1} = W_d h_t + b_d$

"다음 단어로 등장할 확률"

---

training $\sum_{t=1}^{T} \mathcal{L}(\hat{x}_t, x_t)$

cross-entropy loss $-\sum p(x) \log q(x)$

inference $\begin{bmatrix} \text{Greedy} \\ \arg\max S(x_t) \end{bmatrix}$ or $\begin{bmatrix} \text{stochastic} \\ x_t \sim S(x_t) \end{bmatrix}$

... <EOS> $\Rightarrow$ 생성 중단.

---

img to text

CNN

$f_\theta(y \mid f_\phi(x))$

RNN

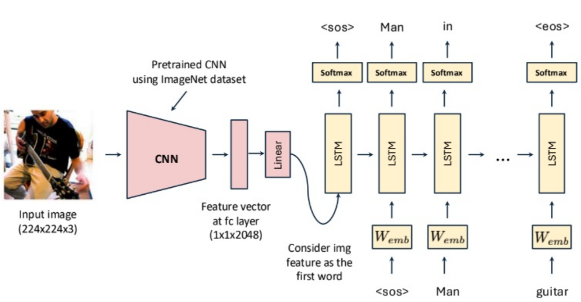
"use feature vector as first word"

"h_0 정보를 잃어버린다면?"

개선 ① 매 순간 단어벡터나 함께 입력

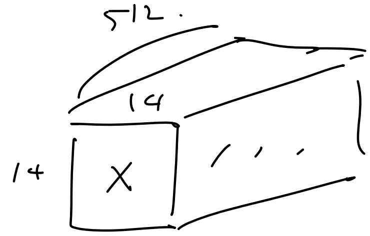
개선 ② Attention

: 단어를 생성할 때 관련된 이미지의 특정 부분을 확인!



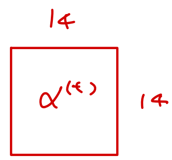
Spatial Attention

- X Conv5 (14×14×512)



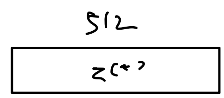
Score $e^{(t)} = f_{att}(X, h_{t-1})$

각 조각은 "a bird"와 얼마나 관련 있는가?



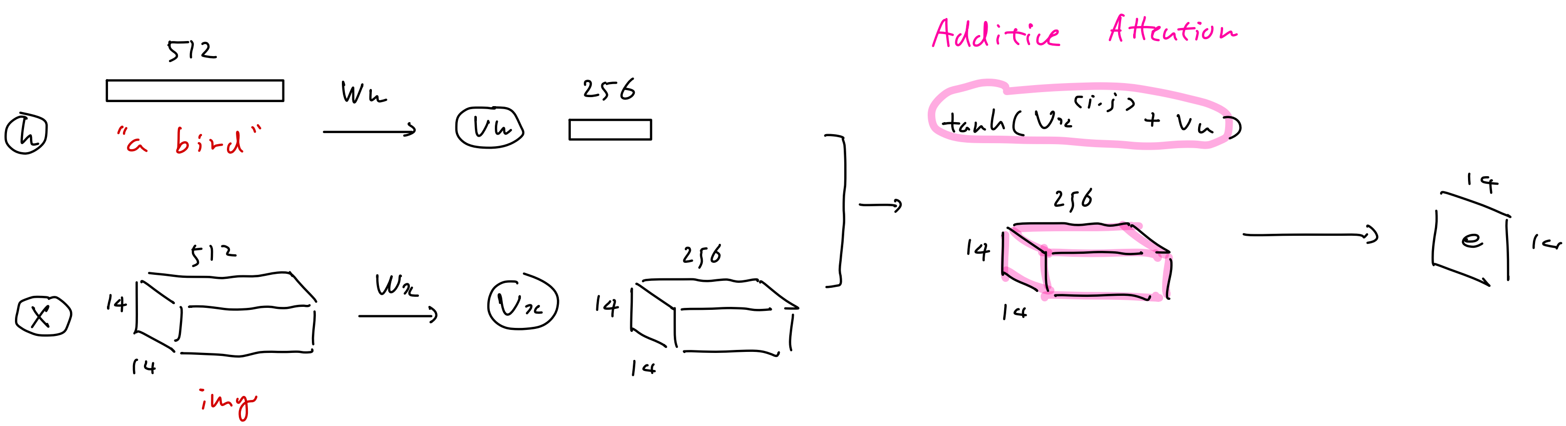
Normalize $\alpha^{(t)} = \mathrm{softmax}(e^{(t)})$

Context $z^{(t)} = \sum_{i,j} \alpha^{(t)}_{i,j} x_{i,j}$

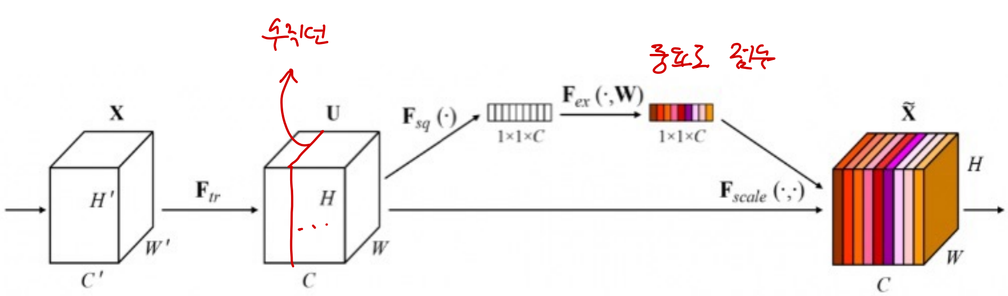


- $h^{(t)} = \mathrm{LSTM}(h_{t-1}, W_{emb} y_t, z^{(t)})$

Simple Spacial Attention



Squeeze-and-Excitation Net : Channel Attention.

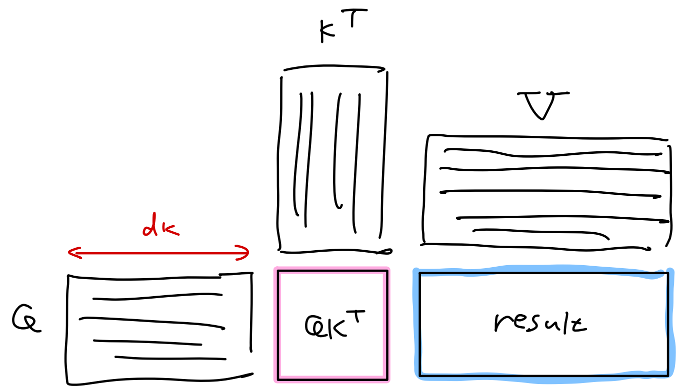


$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\cdot V$$

softmax $\frac{e^{z_j}}{\sum e^{z_j}}$

train

- Query $= XW_q$: $n \times d_k$
- Key $= XW_k$: $n \times d_k$
- Value $= XW_v$: $n \times d_v$



Self-Attention vs Cross-Attention

$Q, K, V \Rightarrow X$

$Q \Rightarrow Y$ (decoder)

$K, V \Rightarrow X$ (encoder)

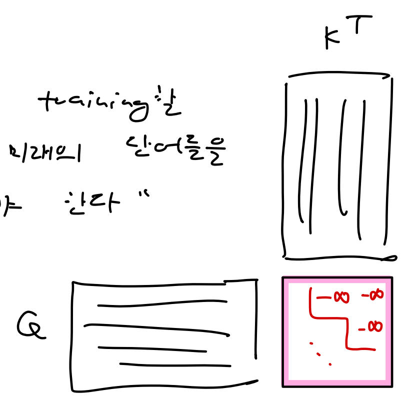
(예시)

기계번역
- Q : 번역본
- K, V : 원문

이미지 캡션
- Q : h_{t-1}
- K, V : 이미지 특징

Masked Self-Attention

"decoder를 training할 때에는 미래의 단어들을 가려야 한다"



i^{th} row $<$ j^{th} column

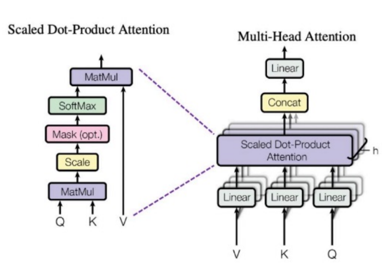
then $-\infty$

positional encoding

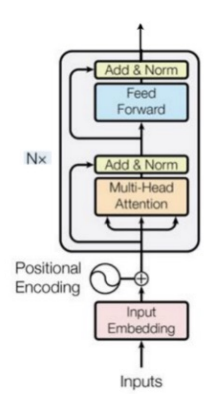
: $x_j + p_j$ or concat(x_j, p_j)

$$2d\begin{bmatrix} \sin(w_1 \times pos) \\ \cos(w_1 \times pos) \\ \vdots \\ \sin(w_d \times pos) \\ \cos(w_d \times pos) \end{bmatrix} \quad \text{where} \quad w_k = \frac{1}{10000^{k/d}}$$

Multi-Head Attention : h개의 다양한 W_q, W_k, W_v를 이용



Transformer Encoder.



→ LayerNorm $(h + FF(h))$

→ LayerNorm $(h + Attn(h,h,h))$

→ $FF(x_i)$

"Attention is All You Need"

Attention & Transformer 의 이해

색깔 질문 Attention

V

$$\begin{bmatrix} \text{리니에이지} & \cdots \\ \text{파란} & \cdots \\ \text{에메랄드} & \cdots \end{bmatrix}$$

K

$$\begin{bmatrix} \text{리니에이지} & \text{파란} & \text{에메랄드} \end{bmatrix}$$

Q $\begin{bmatrix} \text{에메랄드} \end{bmatrix}$

수식어 질문 Attention

V

$$\begin{bmatrix} \text{리니에이지} & \cdots \\ \text{파란} & \cdots \\ \text{에메랄드} & \cdots \end{bmatrix}$$

K

$$\begin{bmatrix} \text{리니에이지} & \text{파란} & \text{에메랄드} \end{bmatrix}$$

Q $\begin{bmatrix} \text{에메랄드} \end{bmatrix}$

$\Longrightarrow$

Q_1

Q_2

Q_3

$\vdots$

다음 레이어로

이동한다..

Linearlization $\quad softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$ is too big!

$sim(q, K)$

$$= \exp\left(\frac{q^T K}{\sqrt{D}}\right) \longrightarrow$$

$ker(q, k)$

$$= \phi(q)^T \phi(k)$$

then

$$\tilde{V}_i = \frac{\sum_{j=1}^{\tilde{N}} ker(q_i, k_j) \cdot v_j}{\sum_{j=1}^{\tilde{N}} ker(q_i, k_j)} = \frac{\phi(q_i)^T \sum_{j=1}^{N} \phi(k_j) \cdot v_j}{\phi(q_i)^T \sum_{j=1}^{\tilde{N}} \phi(k_j)}$$

미리 계산 — 저장 공간 $(r \times d_v)$

미리 계산 — 저장공간 (d_v)

Normalization.

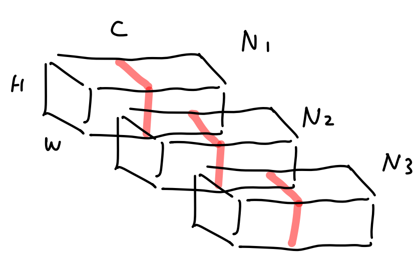
배치 채널 높이/너비

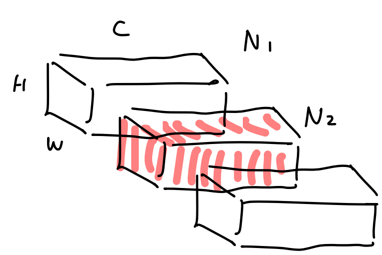
$$X \in \mathbb{R}^{N \times C \times \underbrace{(H \times W)}_{\text{or } L}}$$

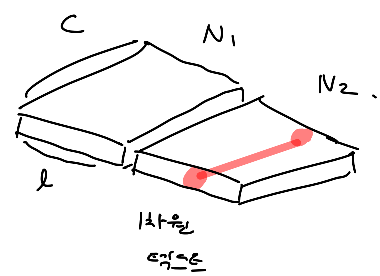
$$LN(x_i) = \beta \cdot \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} \cdot \gamma$$

"얘들은 가변적일 수 있다!"

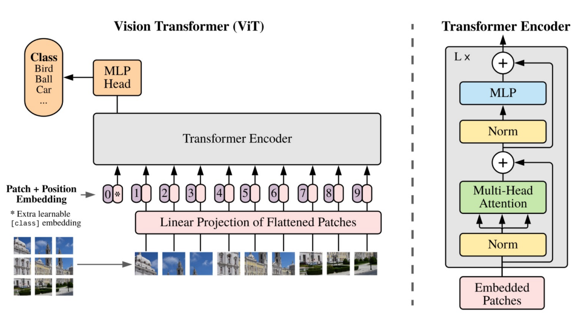
① Batch Norm (CNN) ② Layer Norm (CNN) ③ Layer Norm (Transformer)







$$\mu_c = \frac{1}{NHW} \sum_{n,h,w} x_{n,c,h,w} \qquad \mu_n = \frac{1}{CHW} \cdot \sum_{c,h,w} x_{n,c,h,w} \qquad \mu_{n,l} = \frac{1}{C} \cdot \sum_{c} x_{n,l,c}$$



- 이미지를 N개로 쪼갠다 (patching)
- $z_0 = [\underbrace{x_{class}}_{D차원}; \underbrace{x_p^1 E}_{D차원}; \cdots; \underbrace{x_p^N E}_{D차원}] + E_{pos}$
- $\begin{cases} z'_l = MSA(LN(z_{l-1})) + z_{l-1} \\ z_l = MLP(LN(z'_l)) + z'_l \end{cases}$
- 맨 앞에 있는 D차원 정보만 가져옴.

$GELU = x \cdot \underbrace{\Phi(x)}$

가우시안 분포의 누적확률분포함수

CNN

: 작은 filter가
sliding window로
feature map을 만든다.

✓ 계층적 구조

: 엣지, 선 → 눈, 코, 바퀴

✓ 지역적 처리

: 필터 크기에 의존

ViT

: 이미지를 patch로 나눈 뒤
{ flattening
position encoding 을 거쳐
transformer에 넣는다.

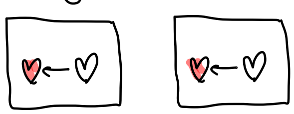
✓ 전역적 처리

: 패치 간의 관계를
한번에 계산

강한 사전가정

- locality
- Translation Equivariance

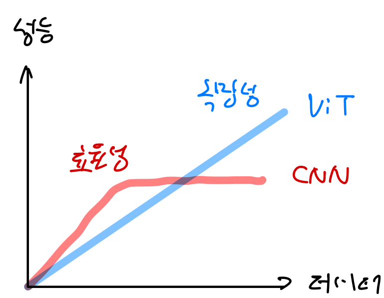
img feature



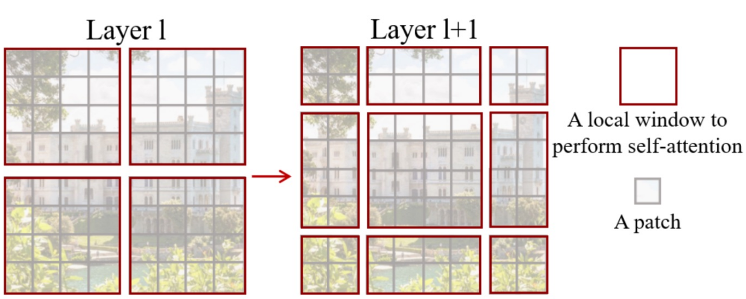
$F(T(x)) = T(F(x))$

약한 사전 가정

- 無 locality
- 無 Translation Equivariance ⟸ position encoding



Note.
translation invariance
$F(T(x)) = F(x)$



① ViA vs W-MSA.

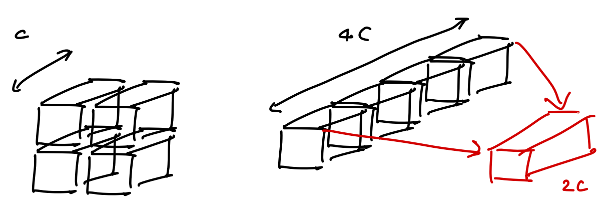
Window 안에서만 Attention.

$$\begin{cases} \Omega(MSA) = 4hwC^2 + 2(hw)^2 \cdot C \\ \Omega(W\text{-}MSA) = \cdots + 2M^2hwC \end{cases}$$

② Shifted window

: 윈도우의 경계를
M/2 만큼 굴림.

③ Patch Merging



like pooling
in CNN.