[10주차] Recurrent Neural Networks

1기 강다연
1기 장예서

EURON

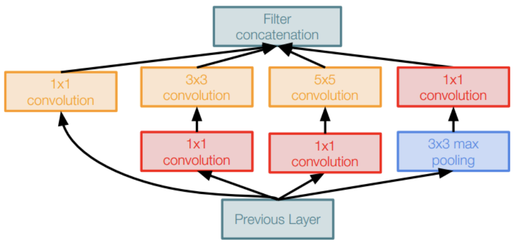
목차

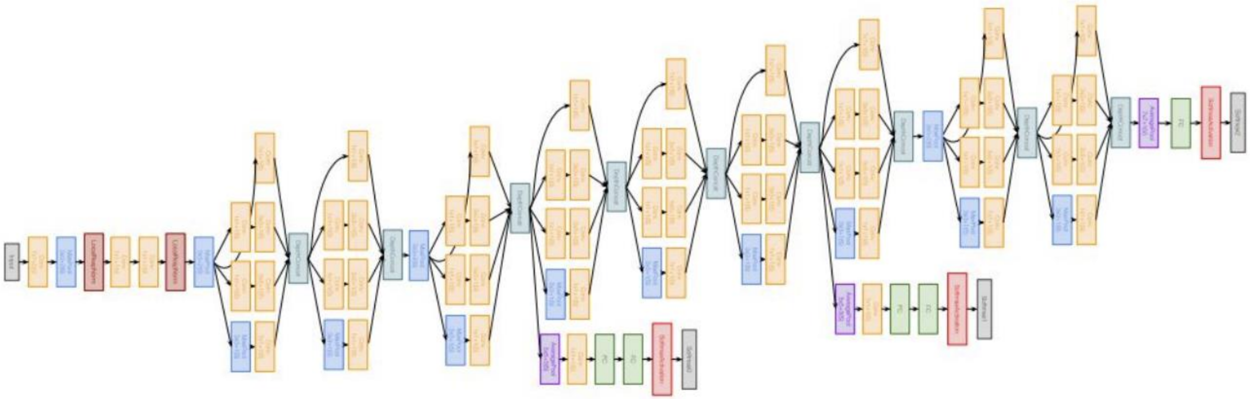
9주차 Review

VGGNet

VGG16 VGG19

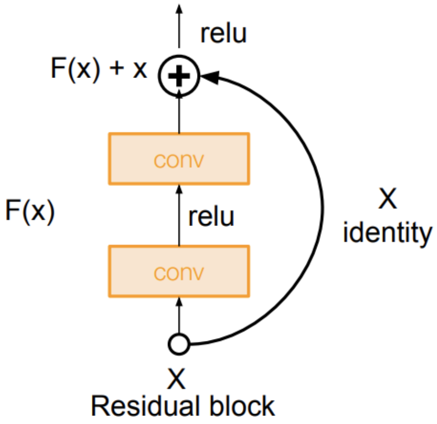
GoogLeNet

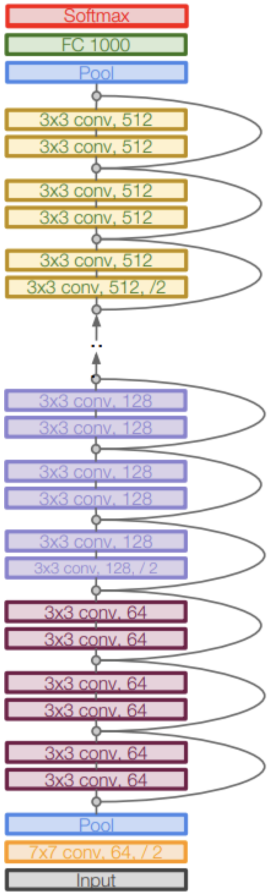




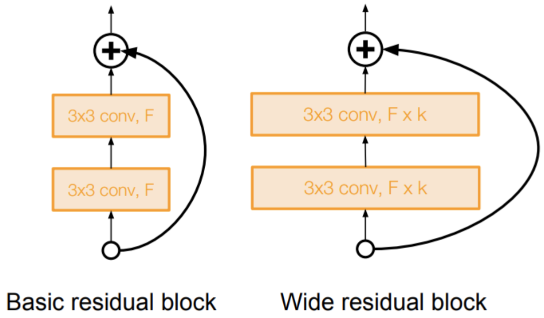
9주차 Review

ResNet

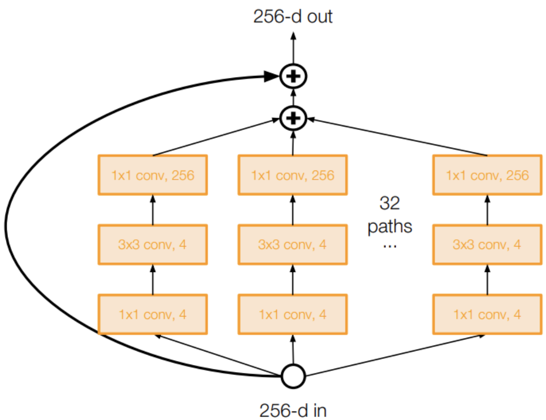




Wide Residual Network

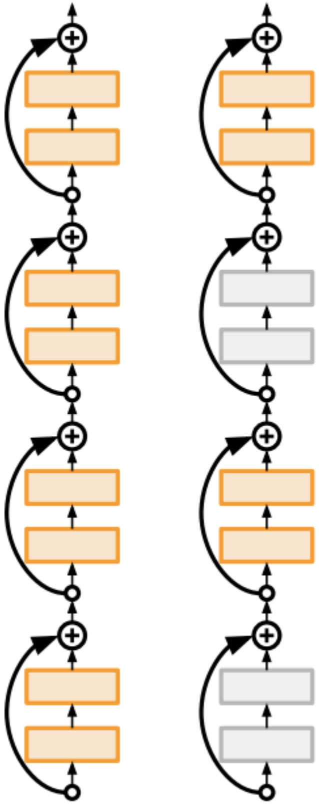


ResNeXt

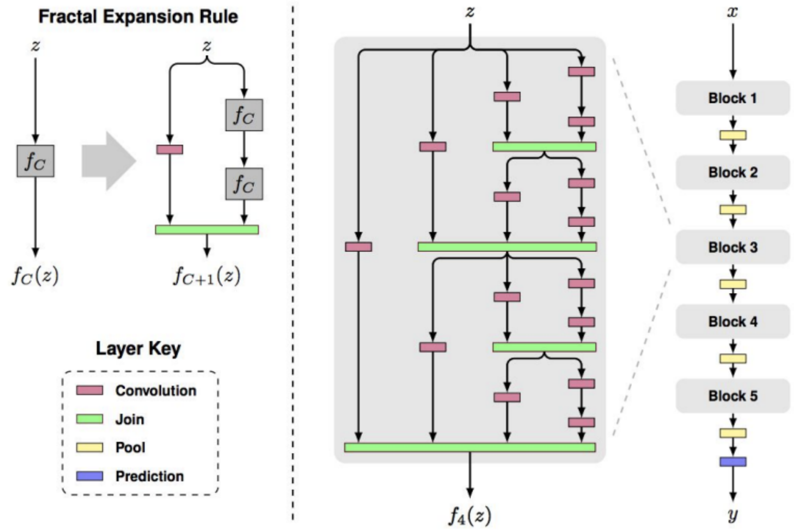


9주차 Review

Deep Networks with Stochastic Depth

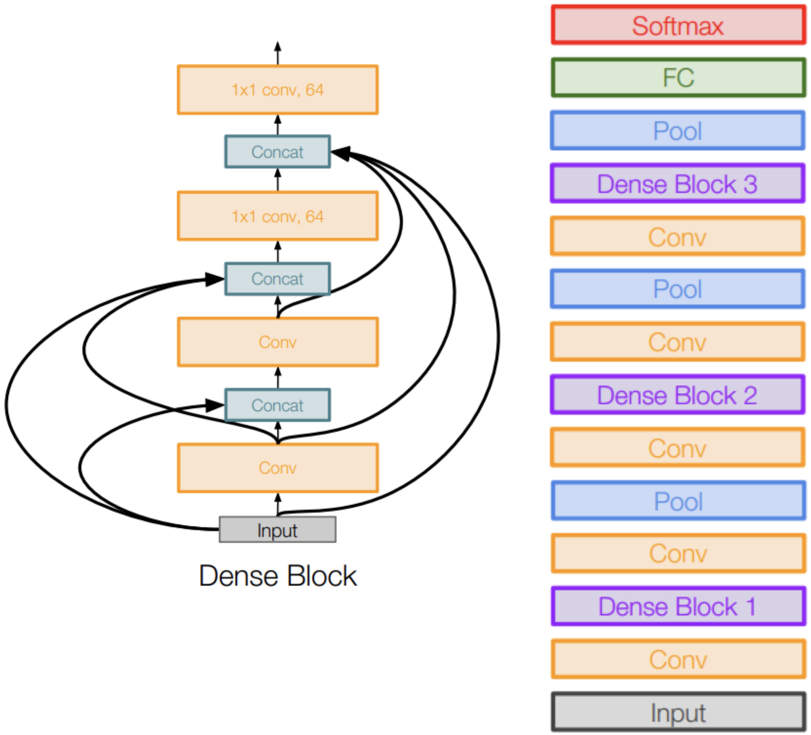


FractalNet



9주차 Review

Densely Connected Convolution Network



SqueezeNet

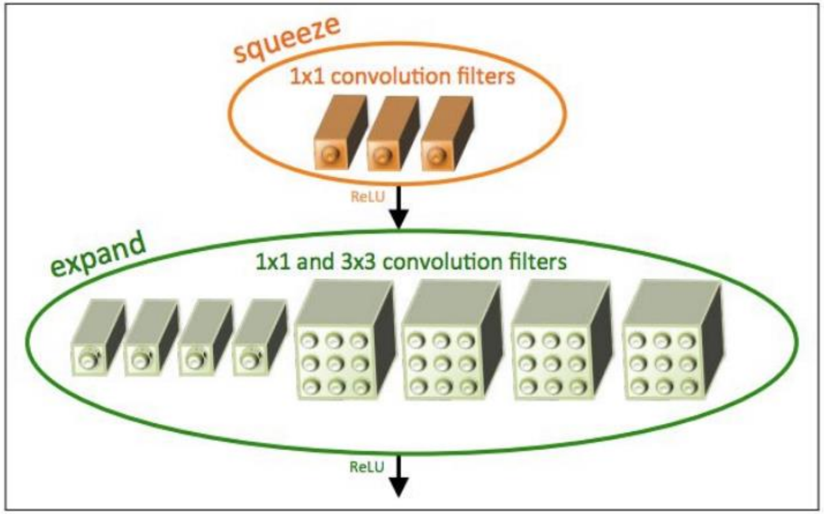
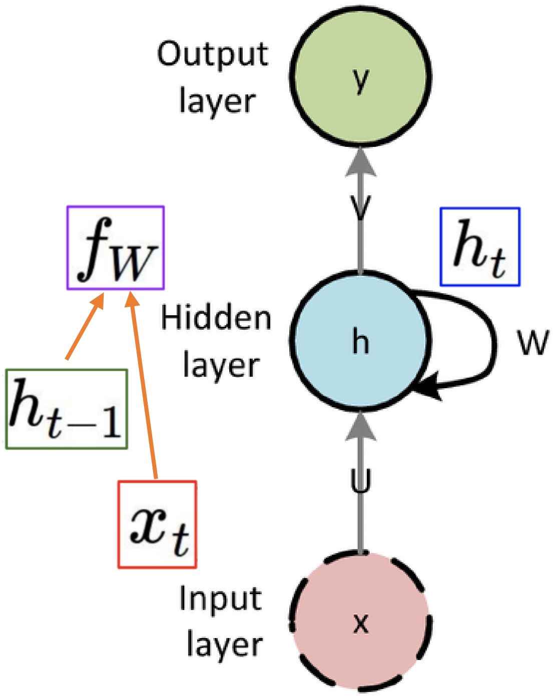


Figure copyright Iandola, Han, Moskewicz, Ashraf, Dally, Keutzer, 2017. Reproduced with permission.

RNN Basics



$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

some function with parameters W

old state

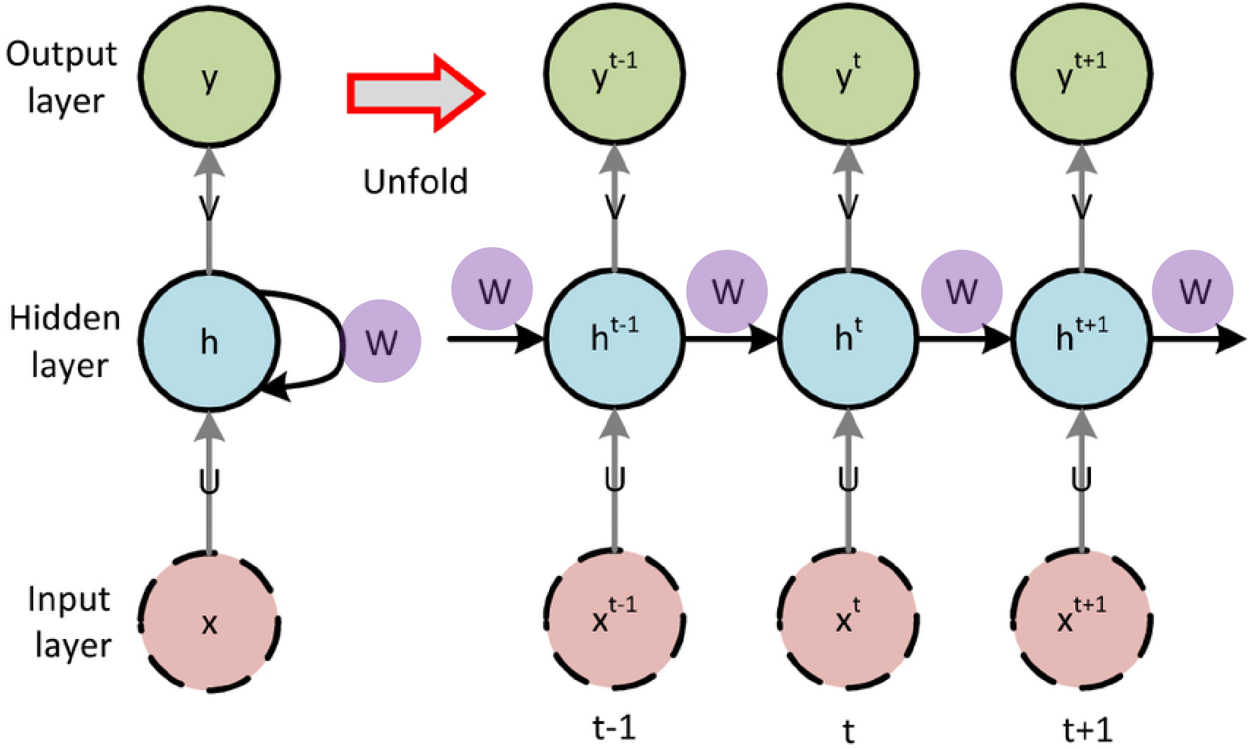
input vector at some time step

Vanila RNN

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

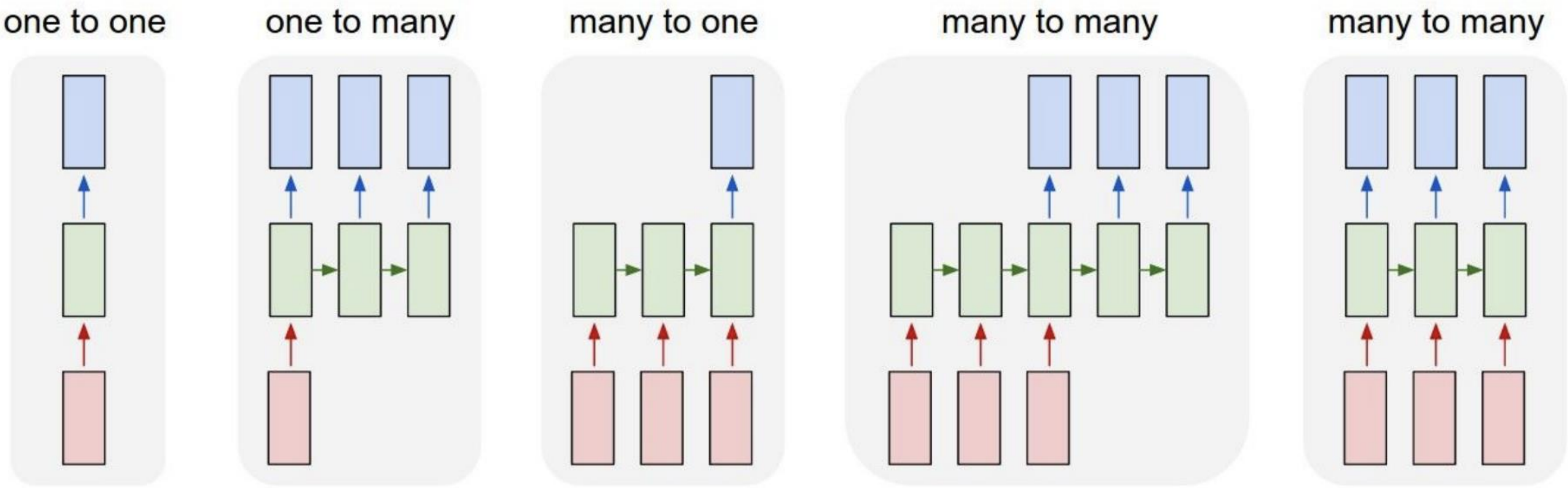
$$y_t = W_{hy}h_t$$

RNN Basics



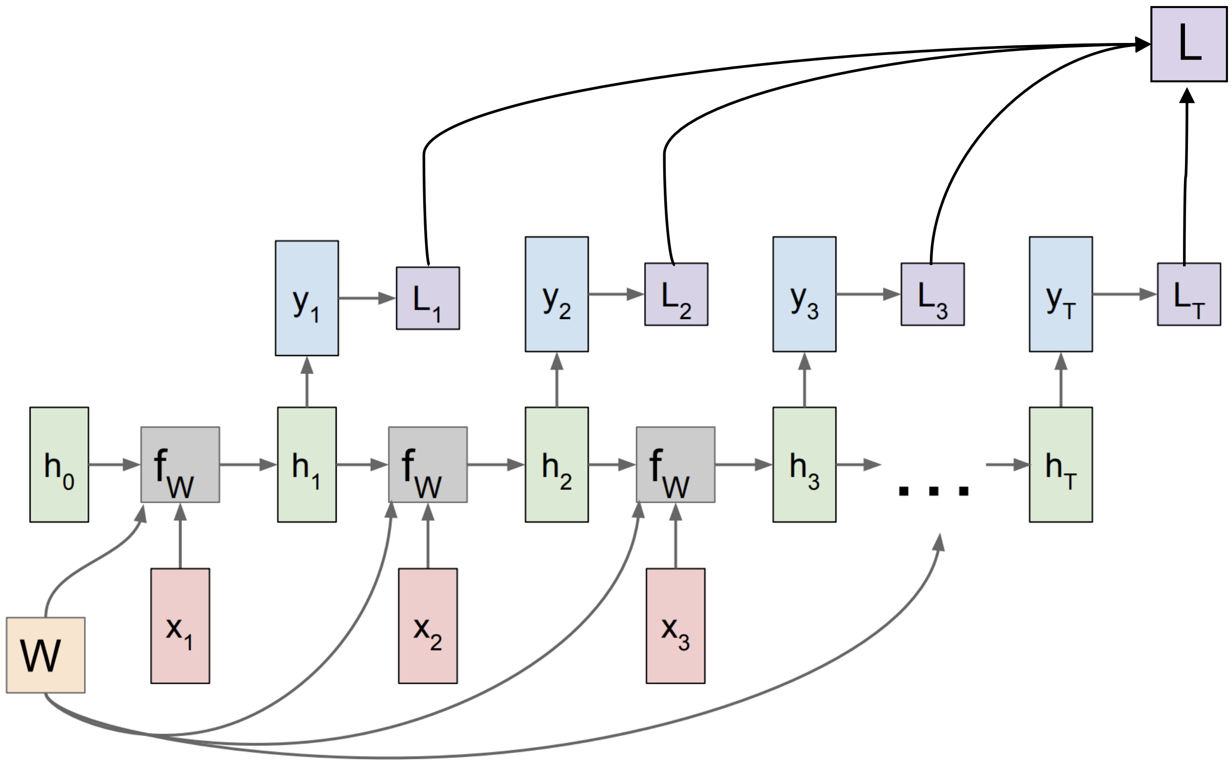
Output layer
y
V
Unfold
y^{t-1}
y^{t}
y^{t+1}
V
V
V
Hidden layer
h
W
W
h^{t-1}
W
h^{t}
W
h^{t+1}
W
U
U
U
U
Input layer
x
x^{t-1}
x^{t}
x^{t+1}
t-1
t
t+1

RNN Basics



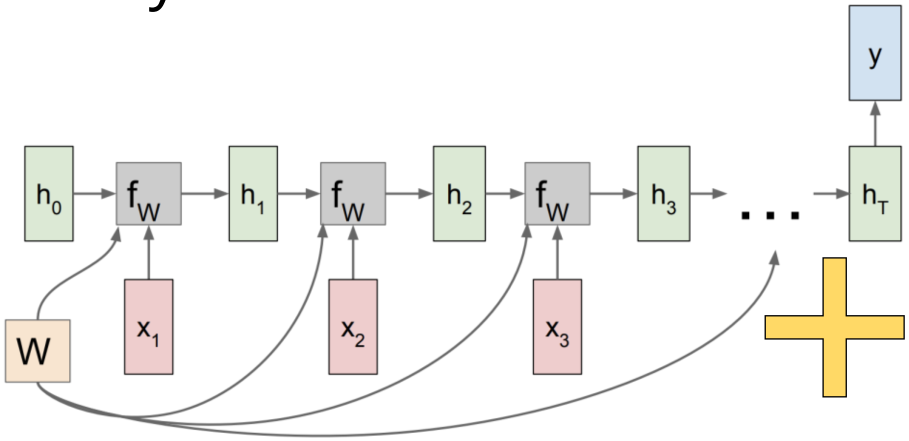
RNN Basics

Many-to-many

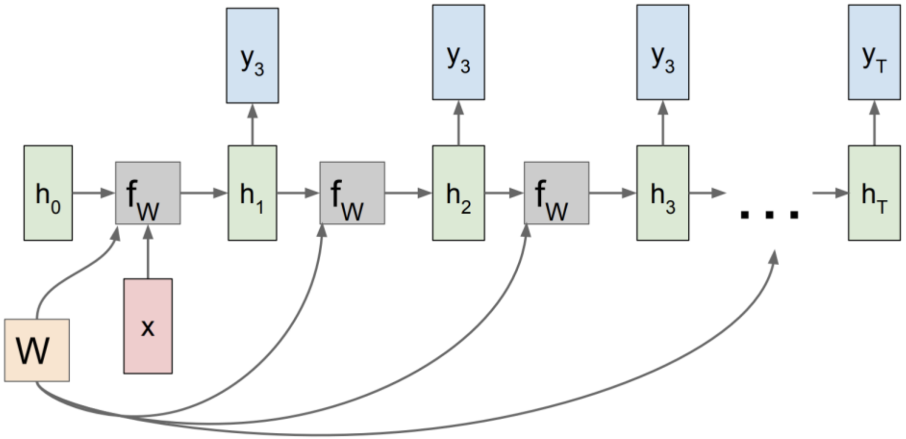


RNN Basics

Many-to-one

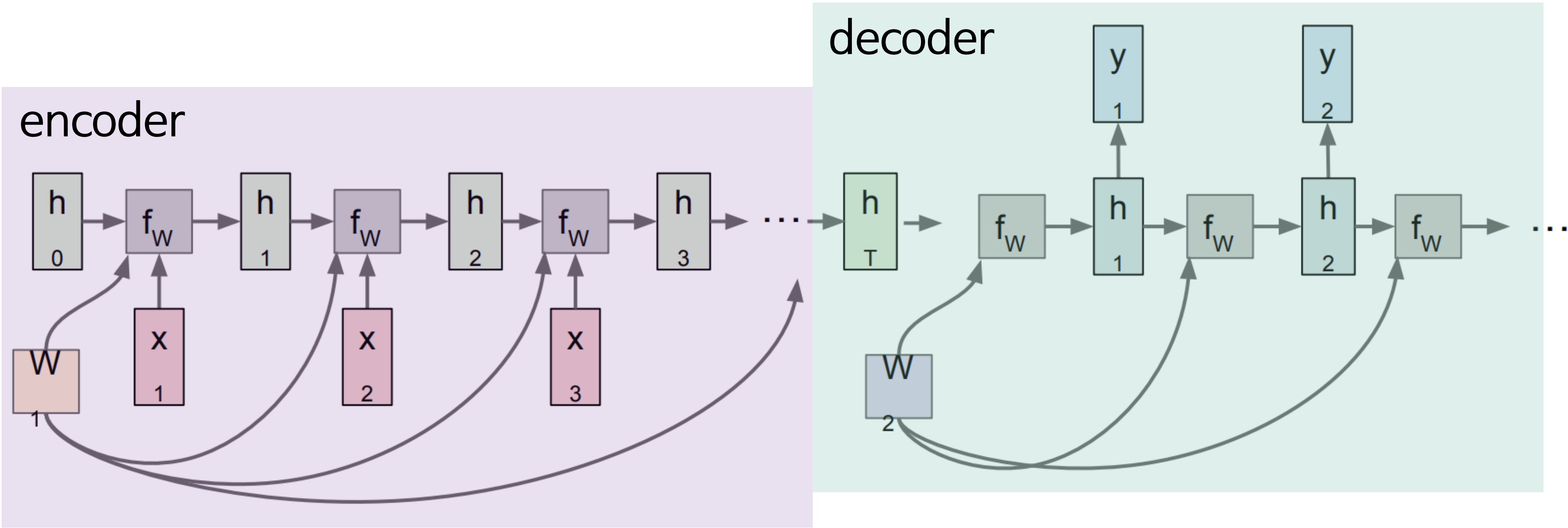


One-to-many



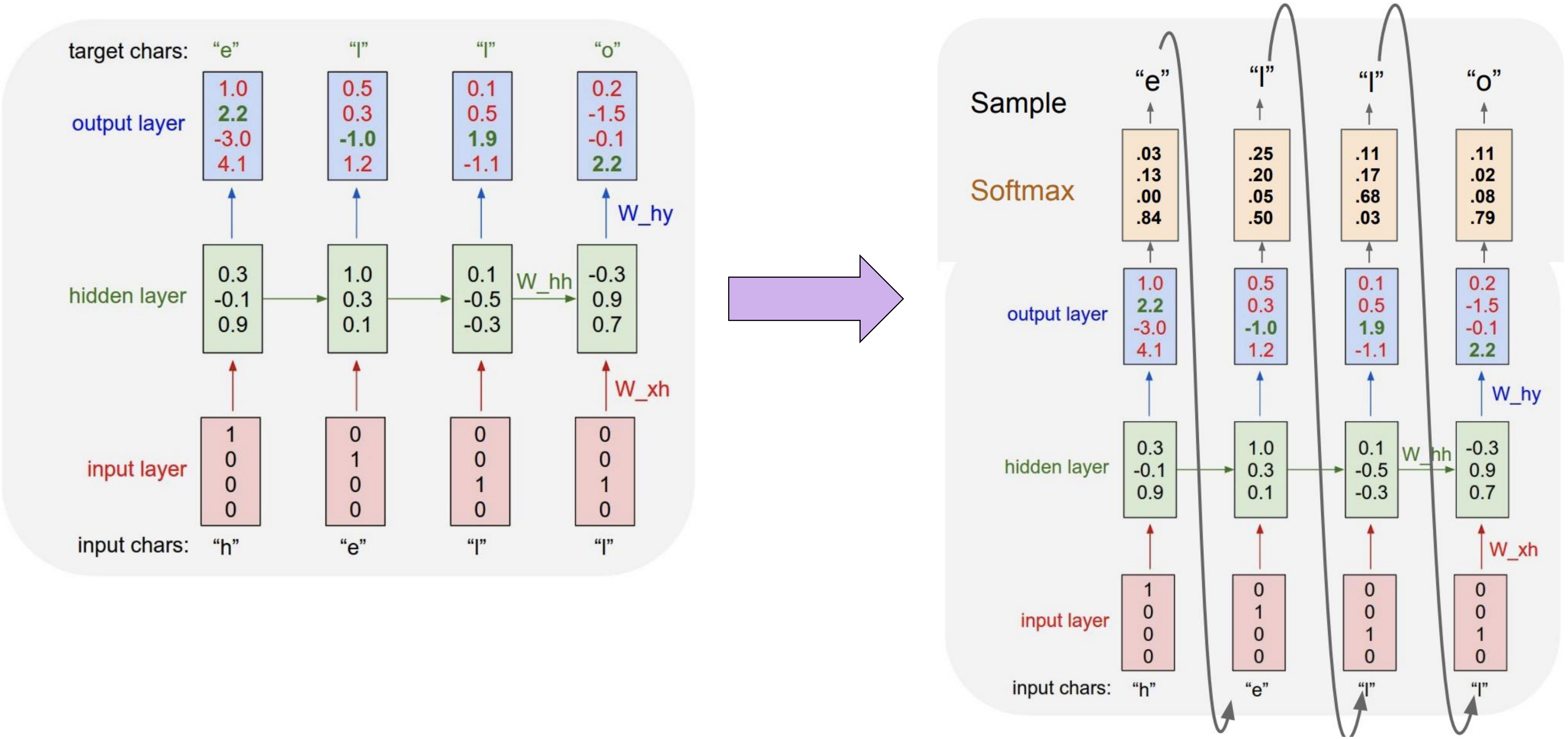
RNN Basics

Sequence-to-sequence



Character-level language model

Training | Test time



Character-level language model

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

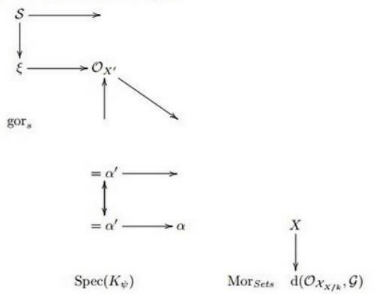
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \; -1(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Another examples of RNN

Examples of sequence data to apply RNN

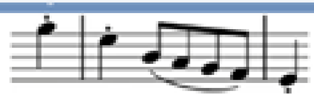
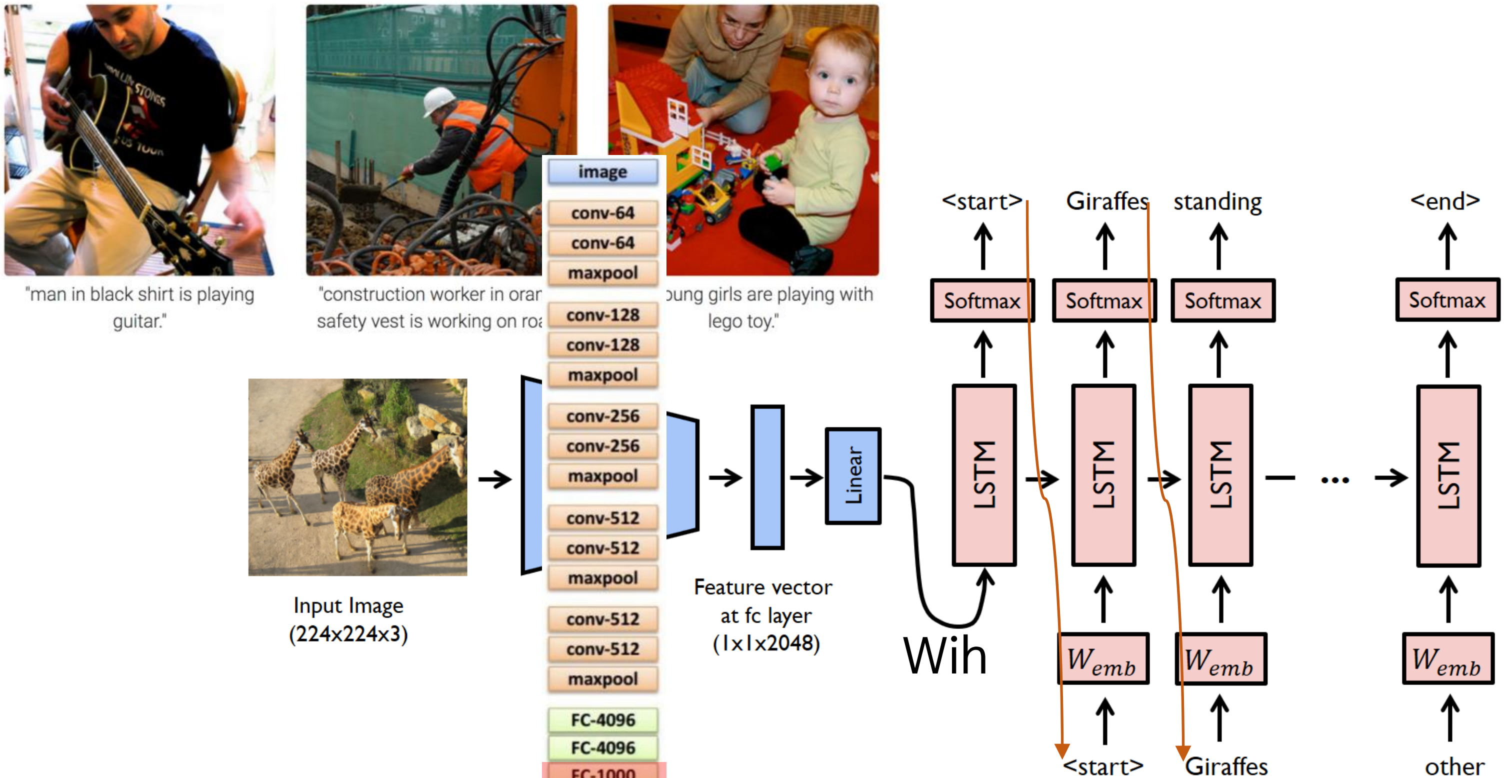
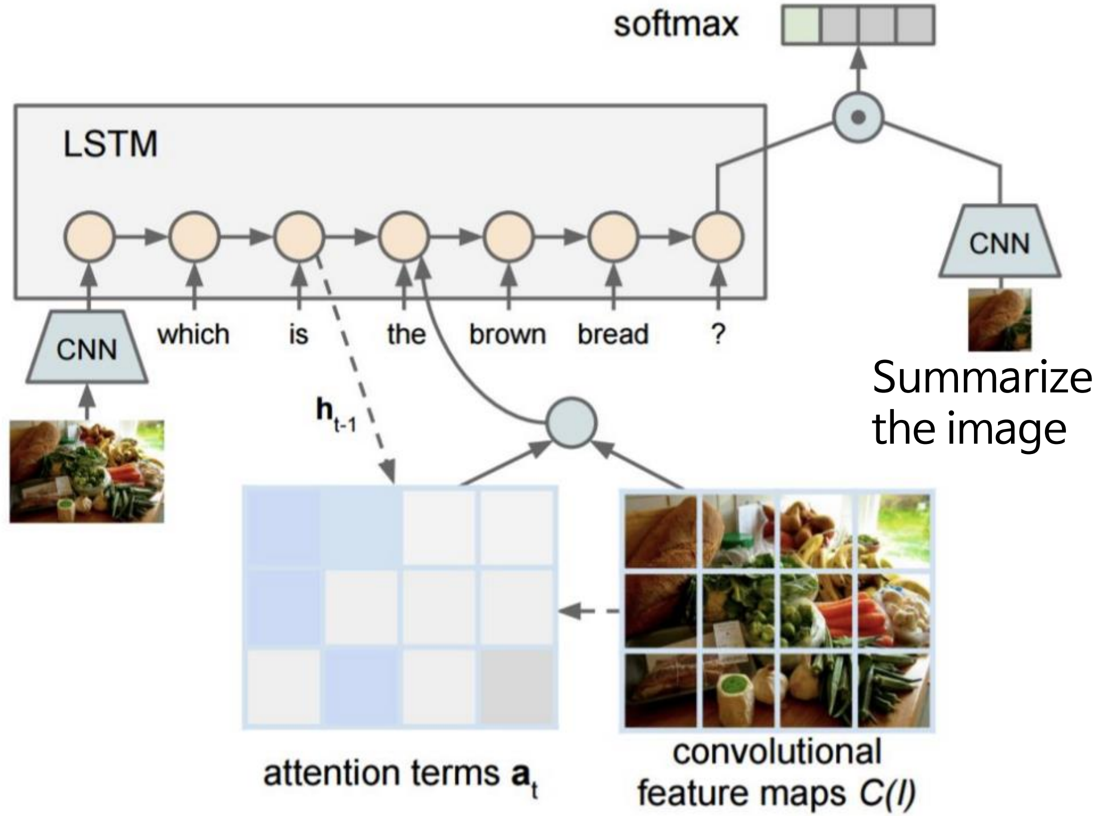
Speech recognition		"The quick brown fox jumped over the lazy dog."
Music generation	∅ (Nothing)	
Sentiment classification	"There is nothing to like in this movie."	★☆☆☆☆
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	Do you want to sing with me?
Video activity recognition		Running

Image Captioning



$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

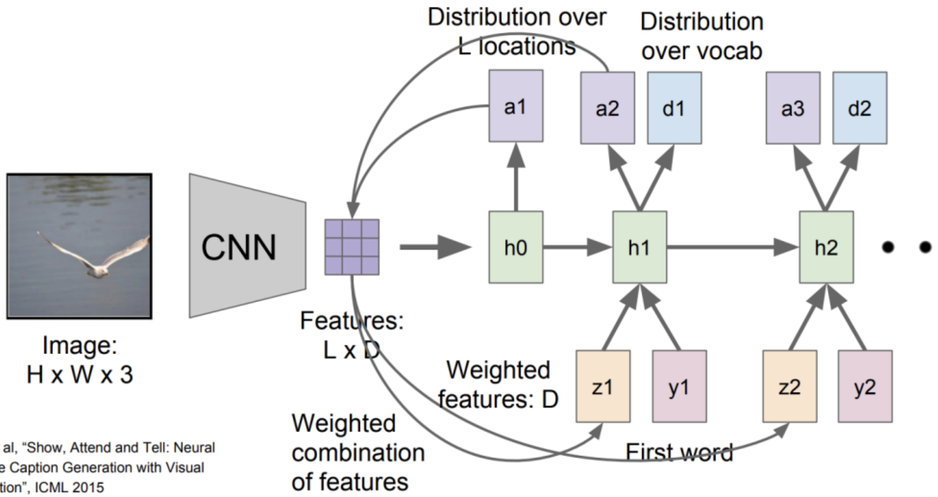
Visual Question Answering



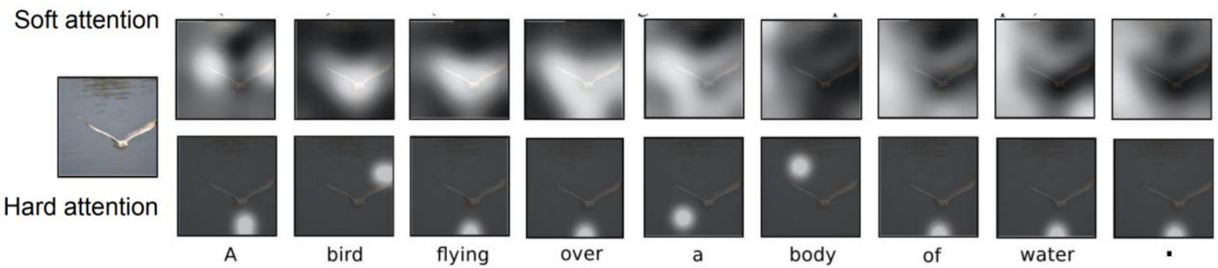
Summarize the image

RNN with Attention

Image Captioning with Attention

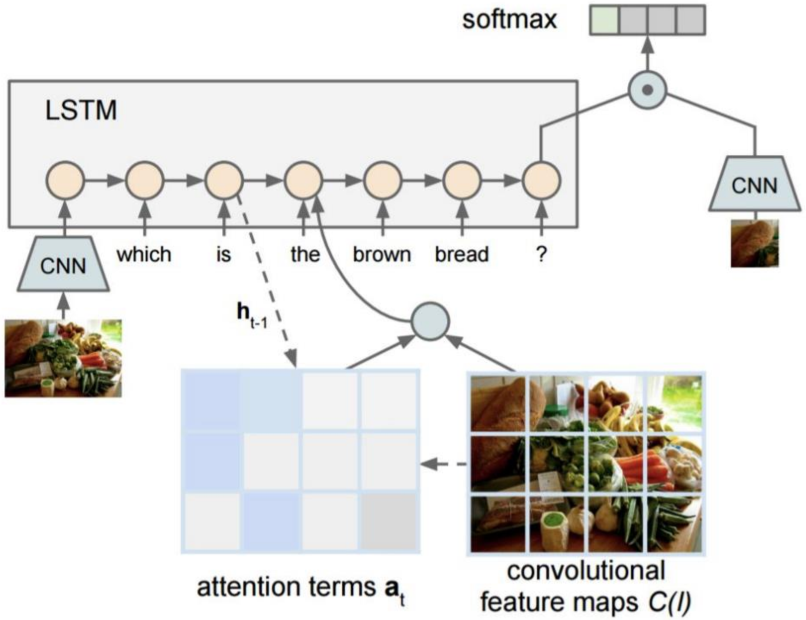


Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015



RNN with Attention

Visual Question Answering with Attention



Zhu et al, "Visual 7W: Grounded Question Answering in Images", CVPR 2016

What kind of animal is in the photo?

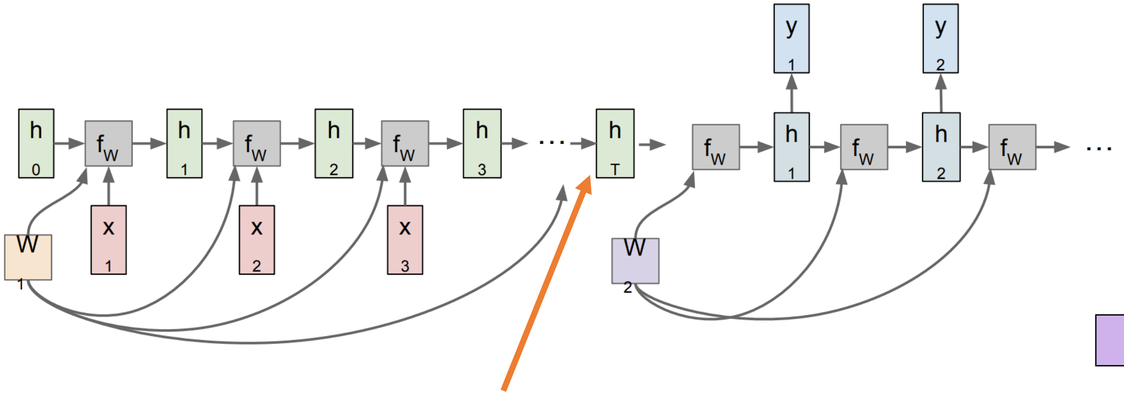
A **cat**.

Why is the person holding a knife?

To cut the **cake** with.

Attention

RNN 기반 sequence-to-sequence model



하나의 고정된 크기의 벡터에 모든 정보 압축

Attention
Decoder에서 단어를 예측하는 매시점마다 예측할 단어와 연관이 있는 encoder의 입력 부분을 더 집중해서 다시 참고

Problems:

1. 정보 손실 발생
2. RNN의 고질적인 문제 - Vanishing Gradient

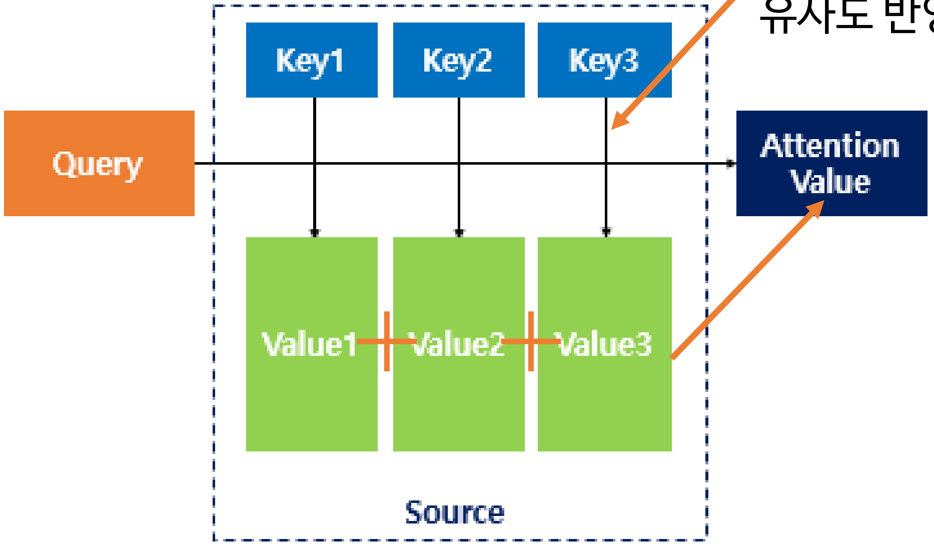
Attention

Attention 함수

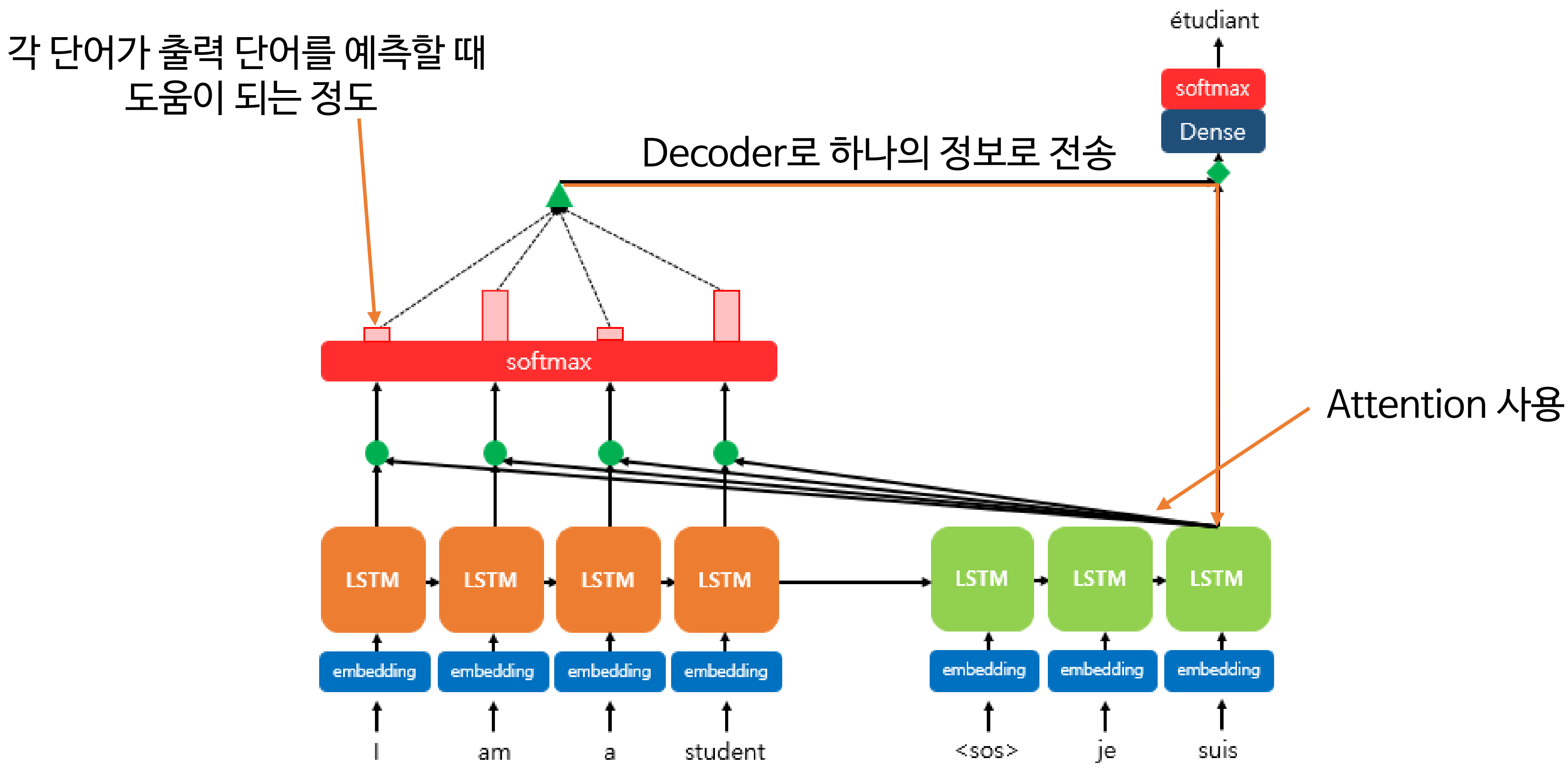
Attention(Q, K, V) = Attention Value

Q = Query: t 시점의 디코더 셀에서의 은닉 상태
K = Keys: 모든 시점의 인코더 셀의 은닉 상태들
V = Values: 모든 시점의 인코더 셀의 은닉 상태들

주어진 쿼리에 대한 모든 키와의 유사도 반영



Dot-Product Attention

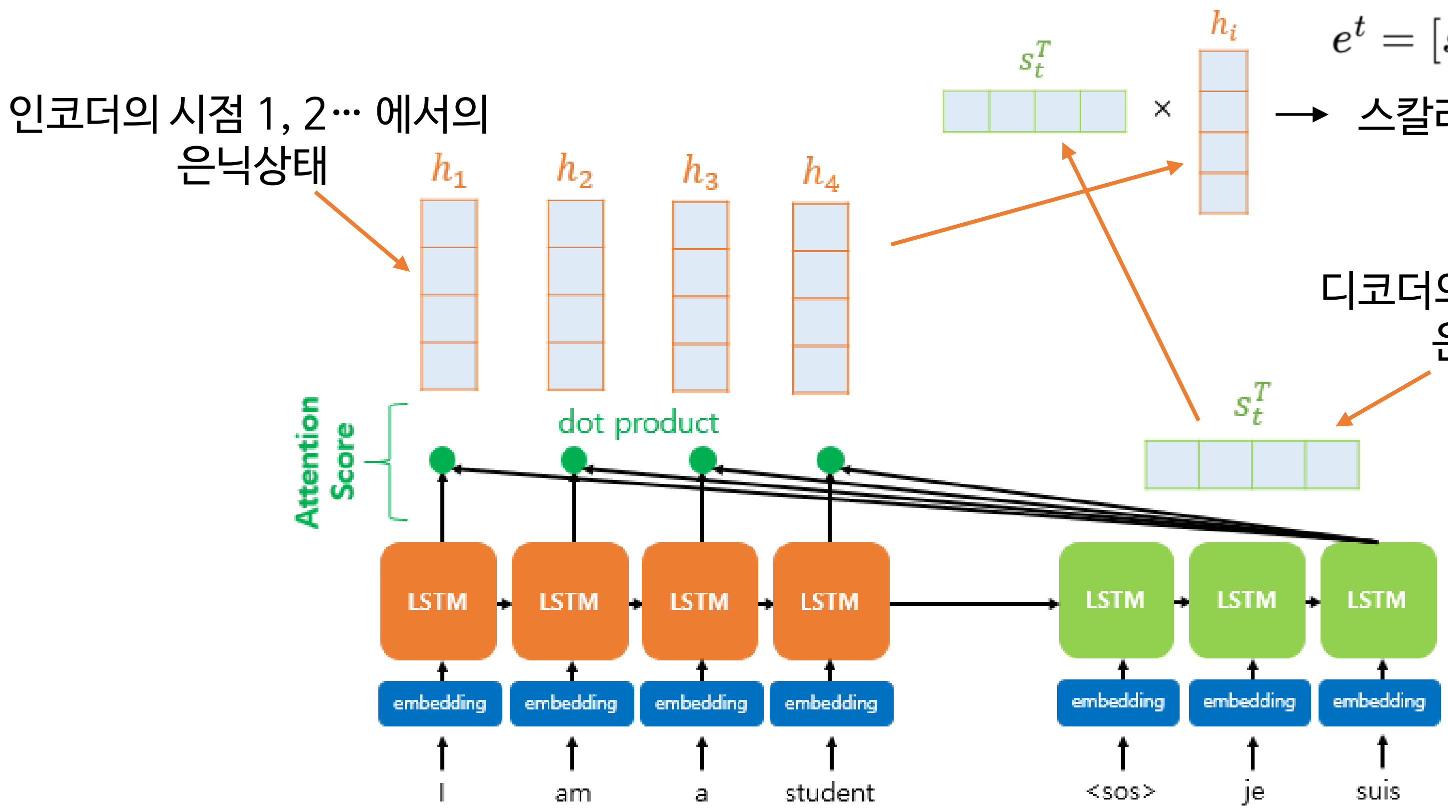


Dot-Product Attention

1) attention score를 구한다

$$score(s_t,\ h_i) = s_t^T h_i$$

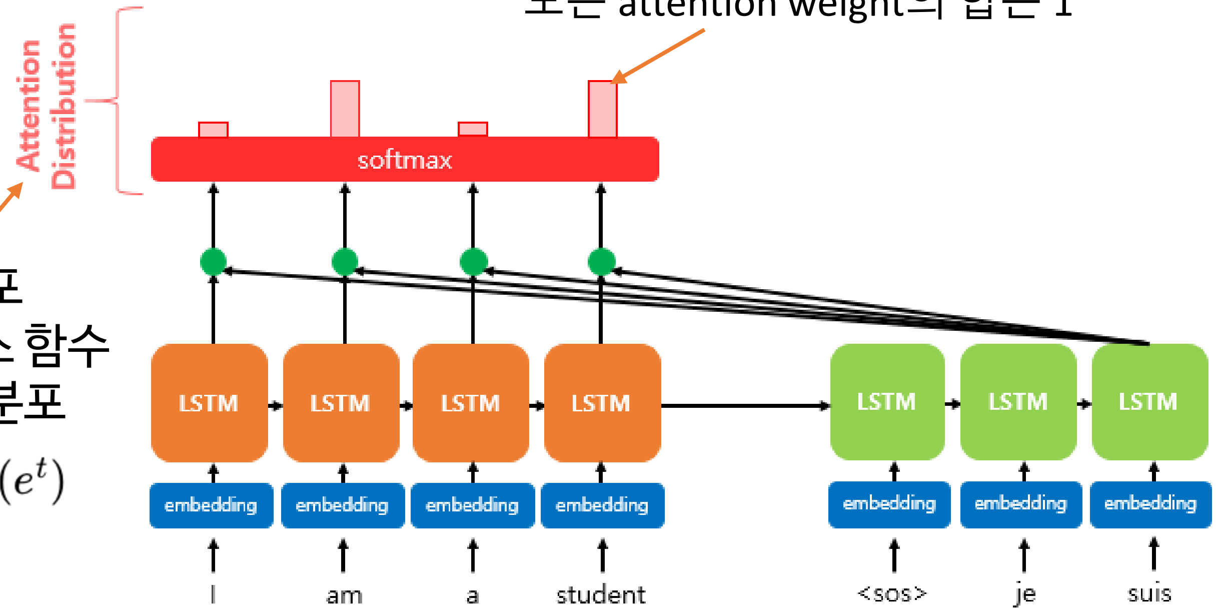
$$e^t = [s_t^T h_1, \ldots, s_t^T h_N]$$



Dot-Product Attention

2) 소프트맥스 함수를 통해 어텐션 분포를 구한다

Attention Weight
모든 attention weight의 합은 1

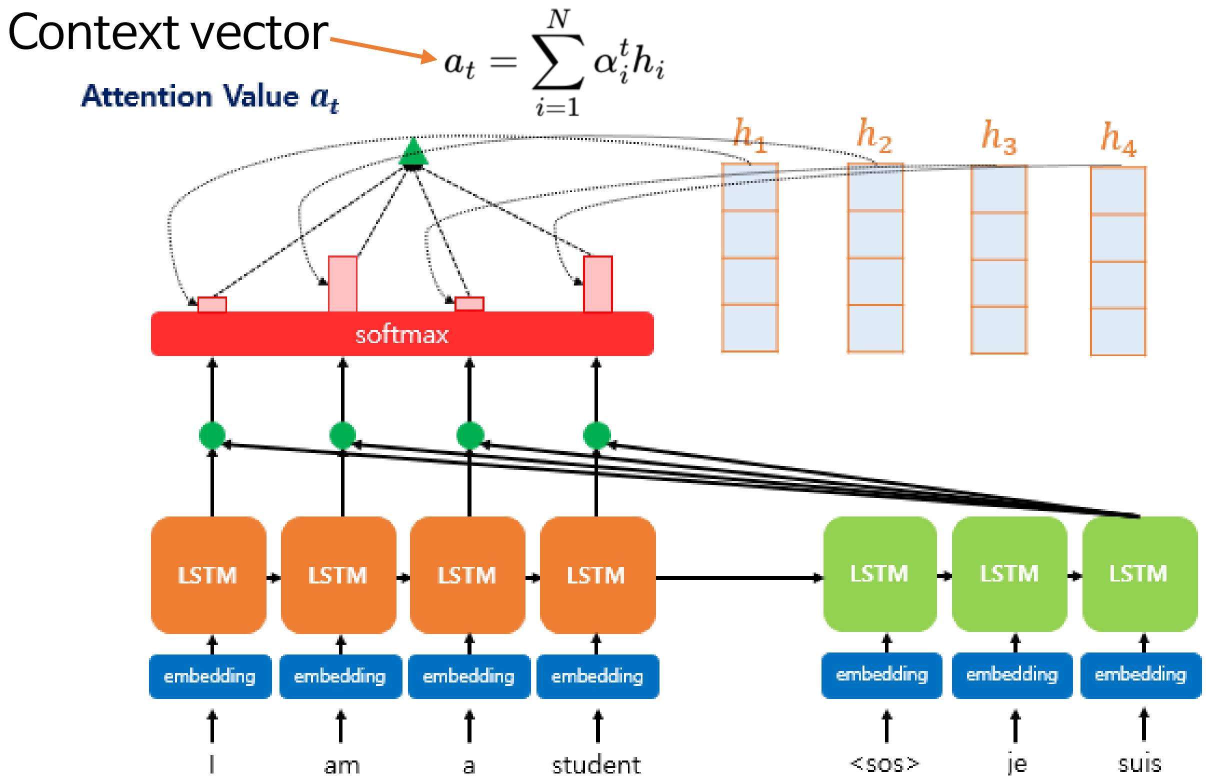


어텐션 분포
e^t에 소프트맥스 함수 적용한 확률분포

$$\alpha^t = softmax(e^t)$$

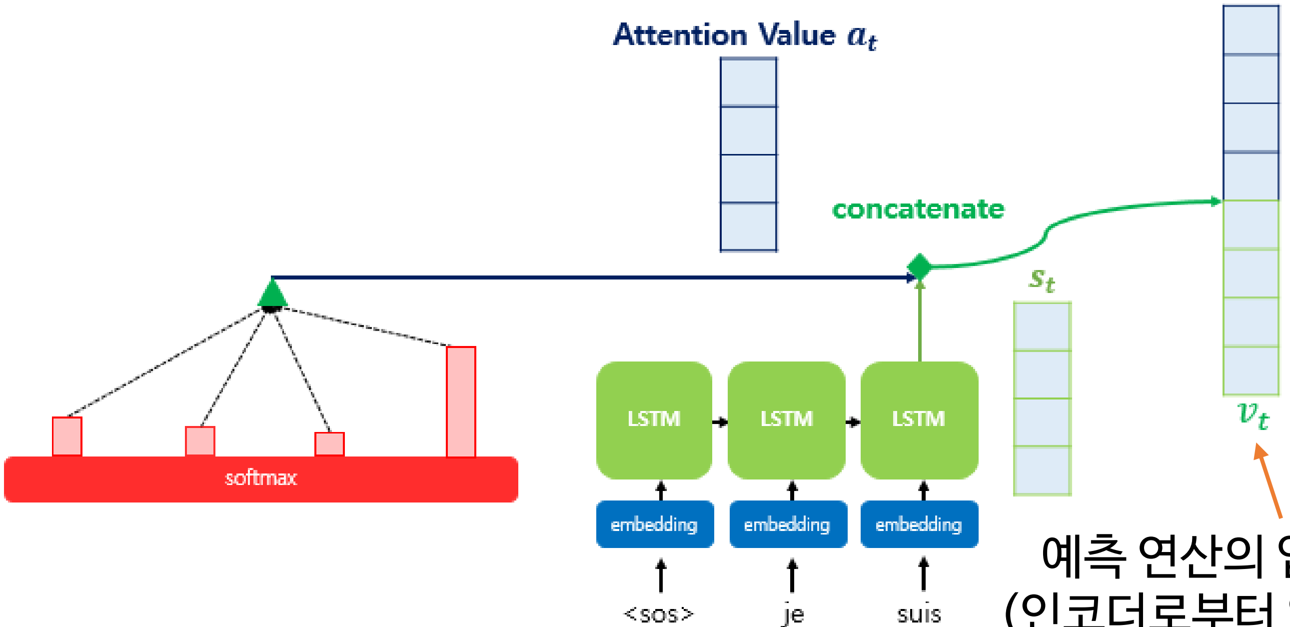
Dot-Product Attention

3) 각 인코더의 어텐션 weight와 은닉 상태를 가중합하여 어텐션 value를 구한다



Dot-Product Attention

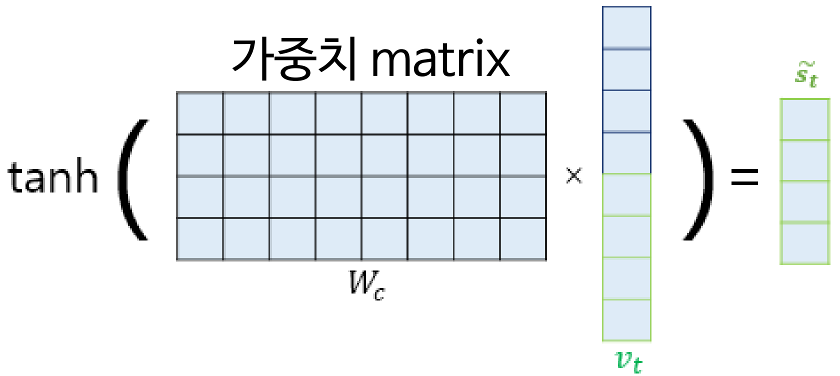
4) Attention value와 디코더의 t 시점의 은닉 상태를 연결한다 (concatenate)



예측 연산의 입력으로 사용
(인코더로부터 얻은 정보 활용)

Dot-Product Attention

5) 출력층 연산의 입력이 되는 s_t를 계산한다



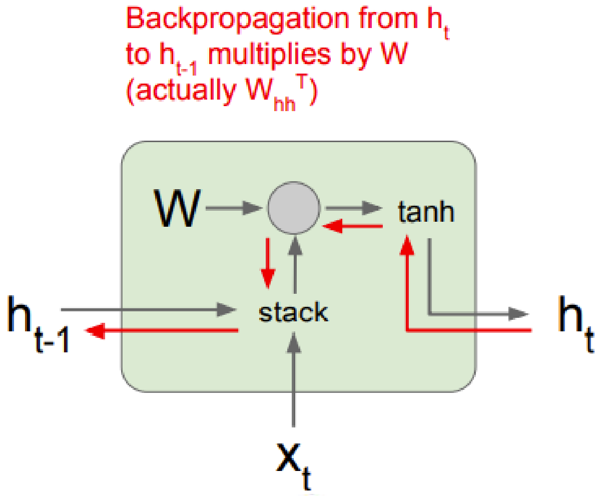
$$\tilde{s}_t = \tanh\left(\mathbf{W_c}[a_t; s_t] + b_c\right)$$

6) $\tilde{s}_t$를 출력층의 입력으로 사용한다

$$\hat{y}_t = \mathrm{Softmax}\left(W_y \tilde{s}_t + b_y\right)$$

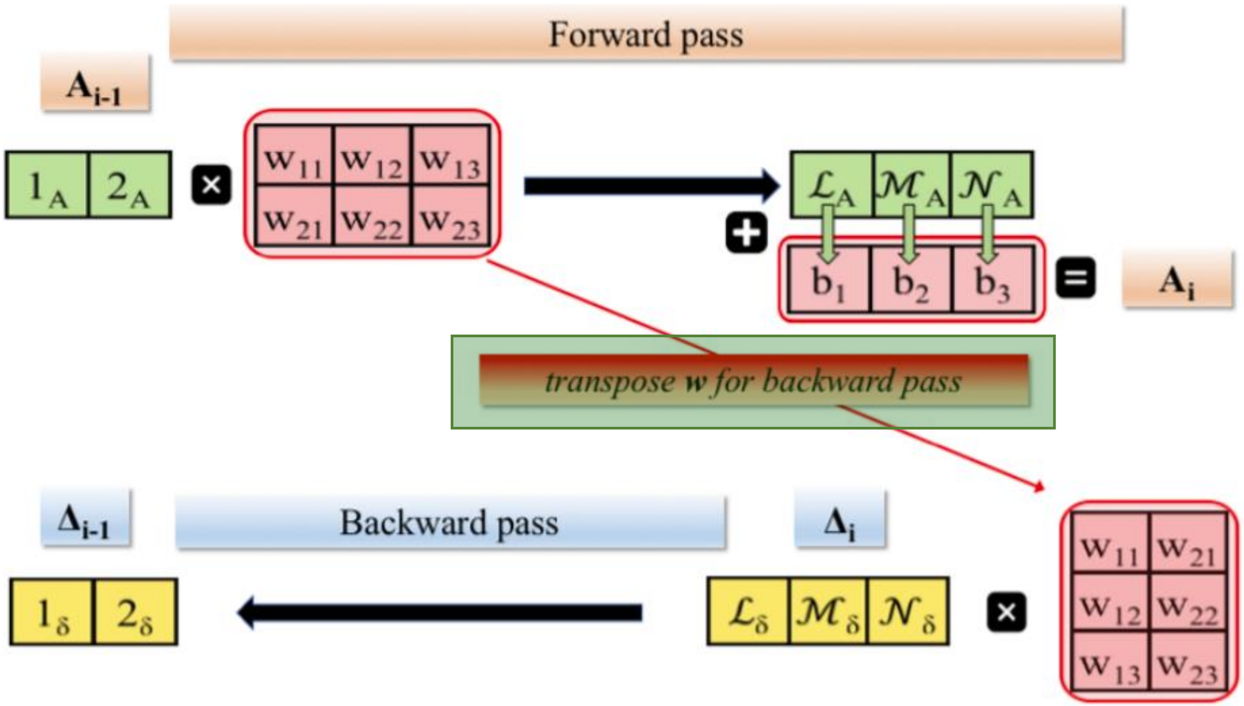
예측 벡터

Backpropagation of RNN



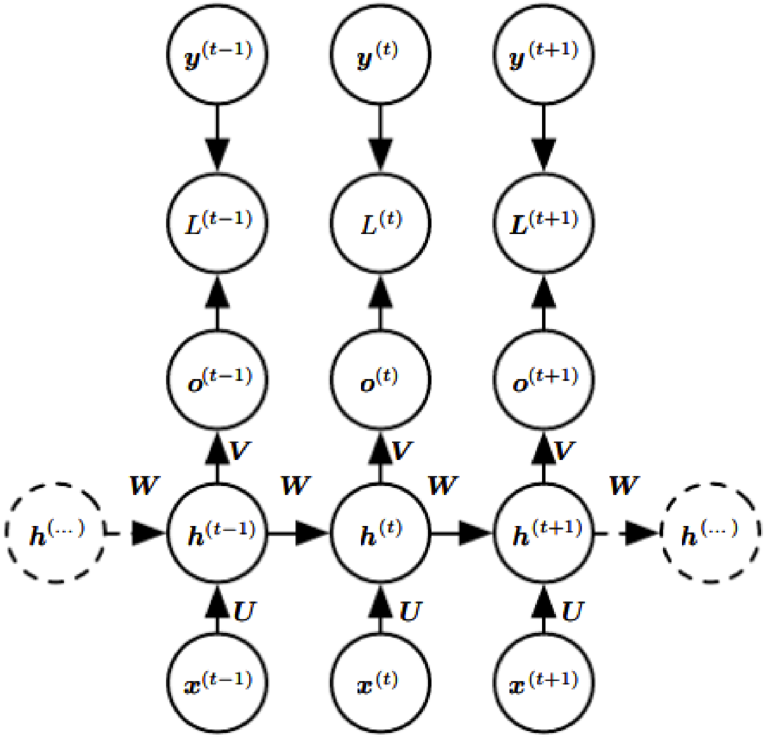
$$
\begin{aligned}
h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\
&= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\
&= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)
\end{aligned}
$$

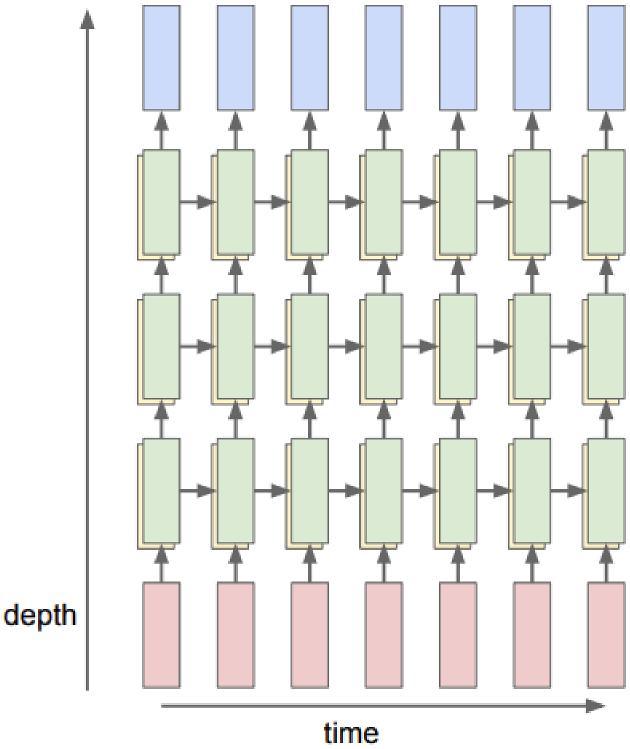
Backpropagation of RNN



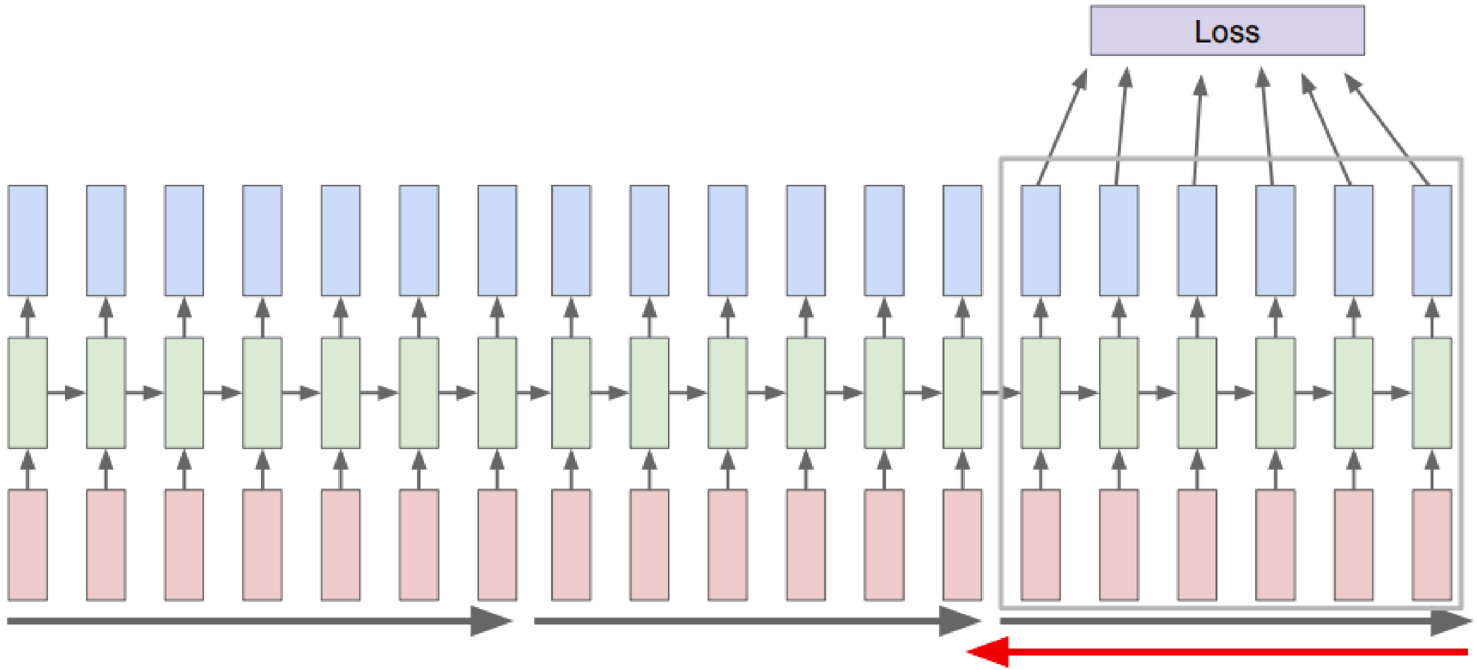
Deep learning for pedestrians: backpropagation in CNNs Laurent Bou ′ e

Backpropagation of RNN: Multi-Layer RNN

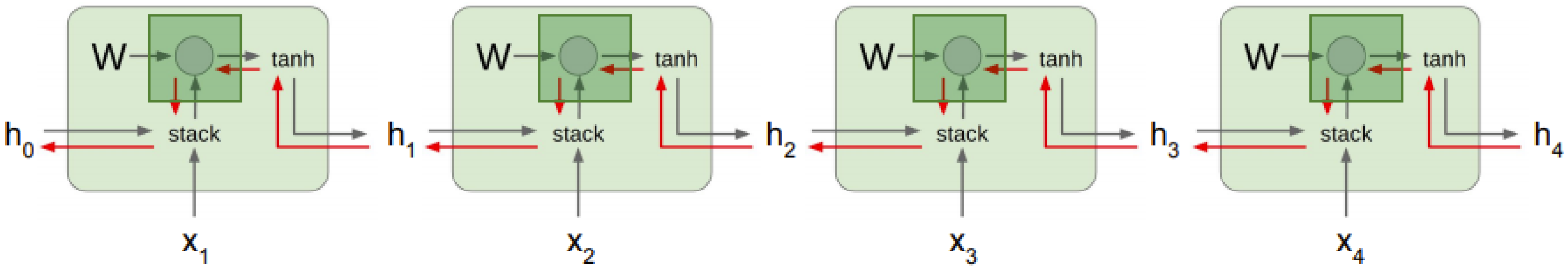




Backpropagation of RNN: Truncated Backpropagation



Backpropagation of RNN: Multi-Layer RNN

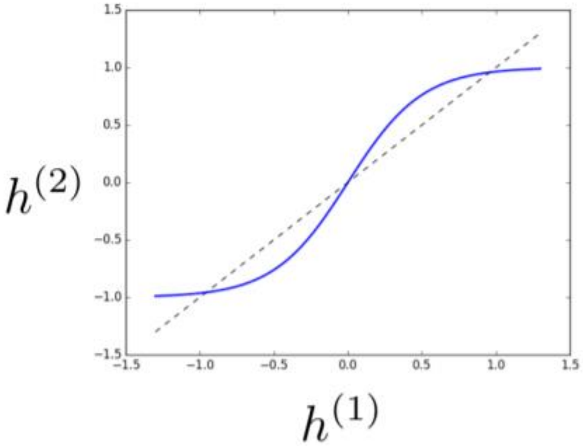


Cell 하나를 **통과할 때마다 mul gate 존재**

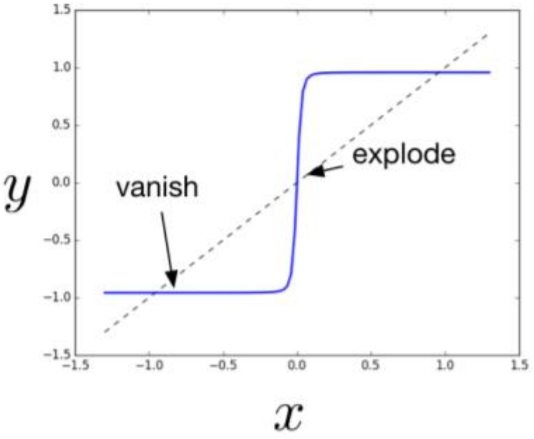
각 Cell의 W transpose factor 관여

매우 많은 W가 관여하게 됨

Backpropagation of RNN



Largest Singular Value > 1:
Exploding Gradients

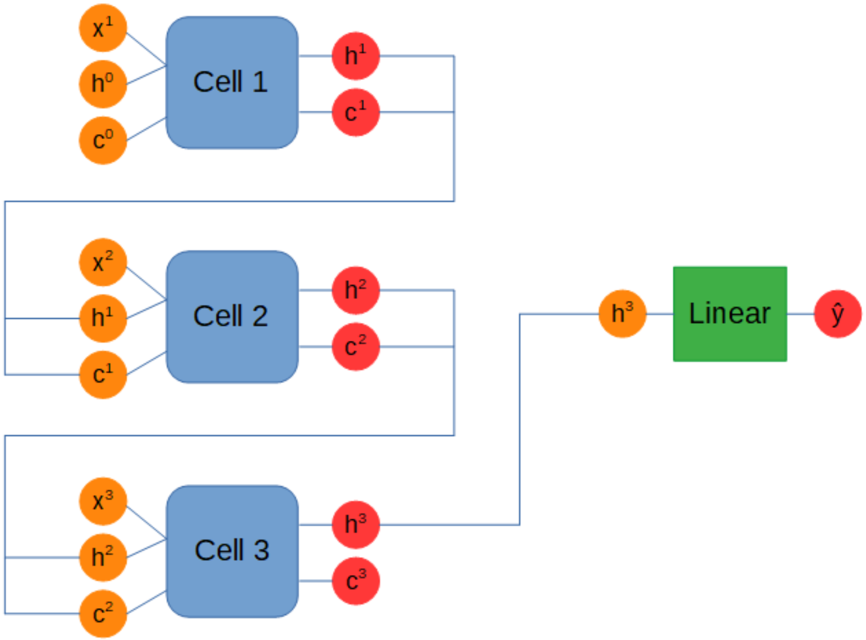


Largest Singular Value < 1:
Vanishing Gradients

▶ **Change Architecture!**

Lecture 15: Exploding and Vanishing Gradients, Roger Grosse, cs.toronto.edu

LSTM-Long Short Term Memory



LSTM

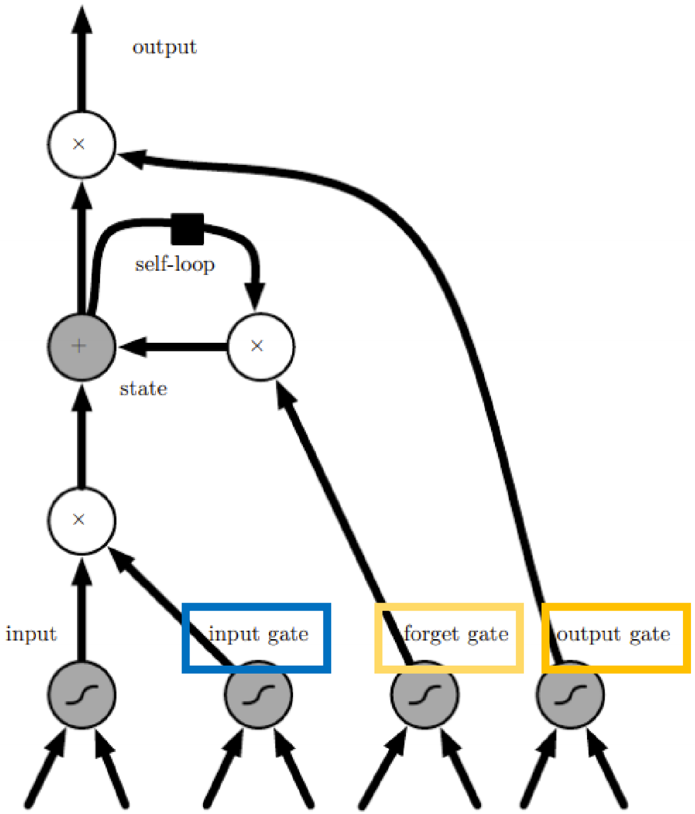
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

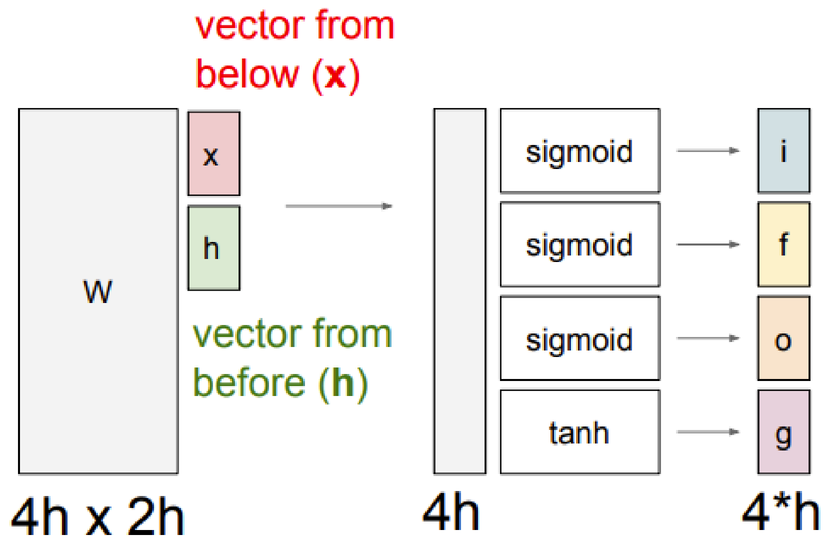
$$c_t = f \odot c_{t-1} + i \odot g$$
$$h_t = o \odot \tanh(c_t)$$

cell당 하나의 hidden states ▶ cell당 두 개의 hidden states

How the LSTM improves the RNN, Tiago Miguel, towardsdatascience.com

LSTM-Long Short Term Memory



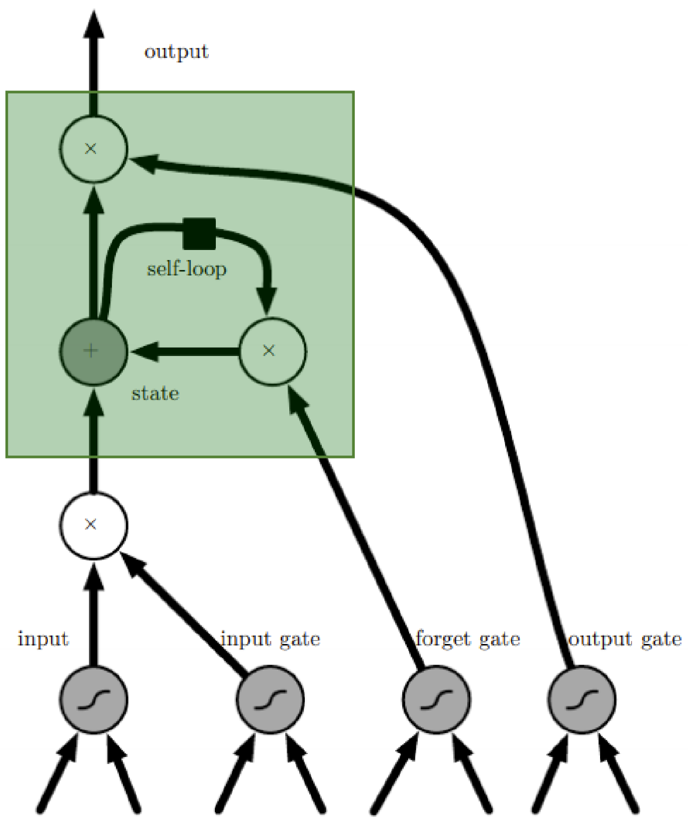


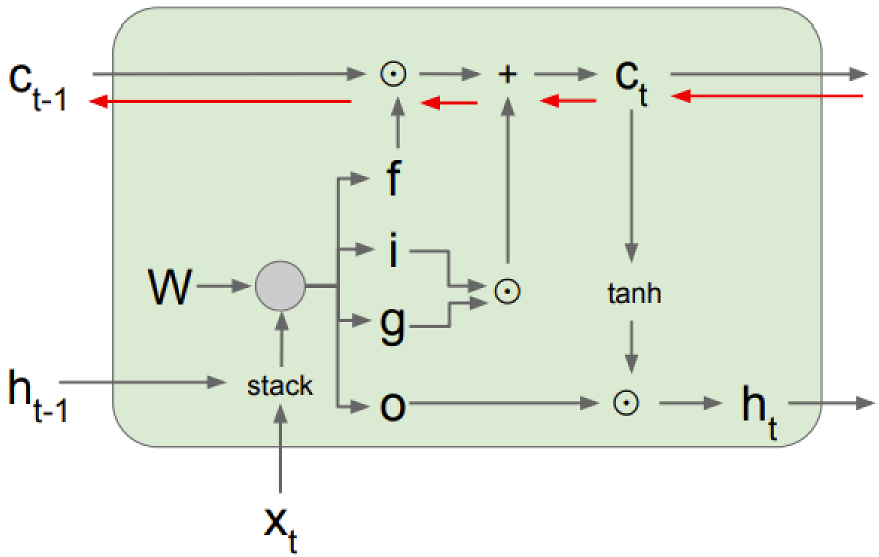
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

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

$$h_t = \boxed{o} \odot \tanh(c_t)$$

LSTM-Long Short Term Memory



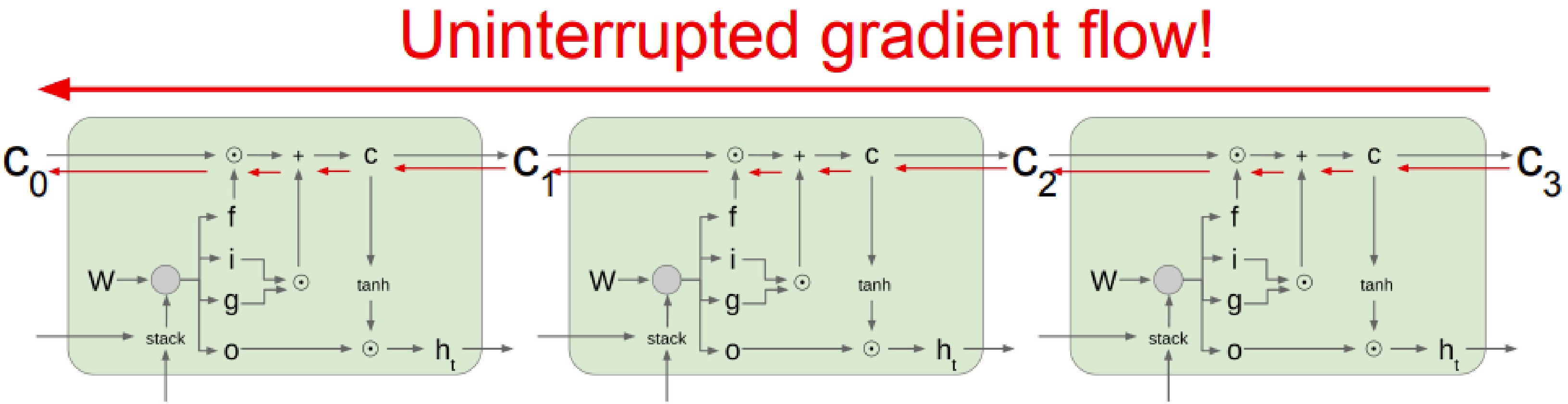


Addition Operation

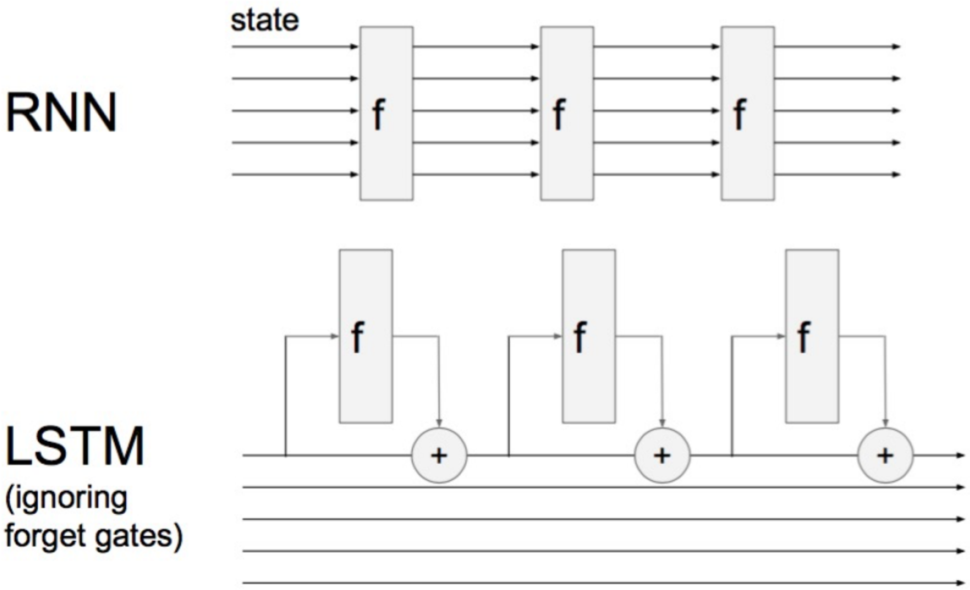
▶ C의 backpropagation 과정에서

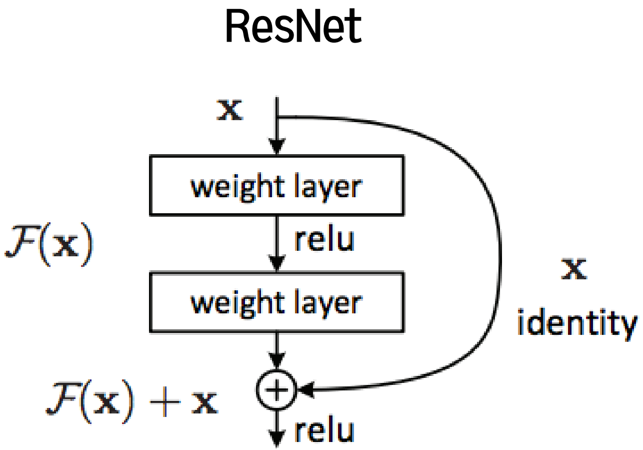
W에 대한 multiplication 존재 X

LSTM-Long Short Term Memory



LSTM: and ResNet





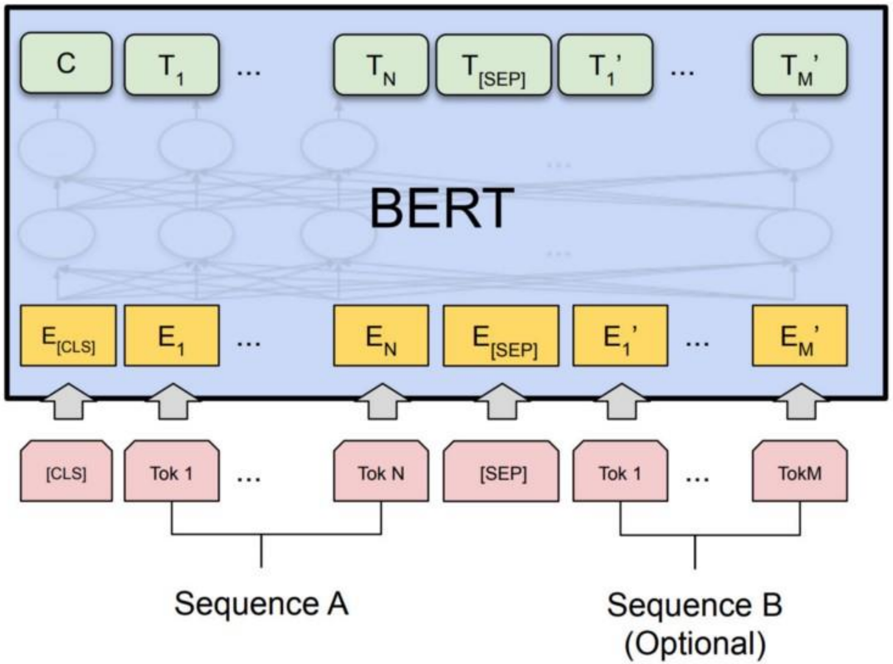
Element Wise Multiplication
▶ **매 step마다 다른** forget gate와 곱해질 수 있음

Activation을 직접적으로 transformation하는 weight 학습 X ▶ input과 ouput의 차이인 residual을 학습

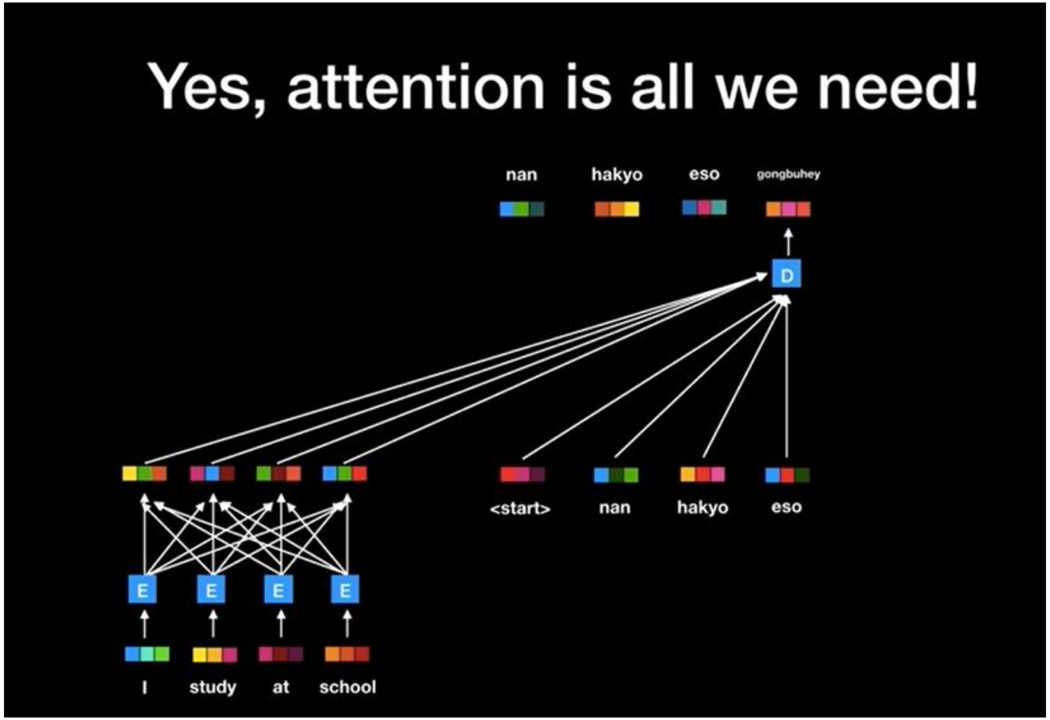
BERT: Bidirectional Encoder Representations from Transformers

Pre-trained Model

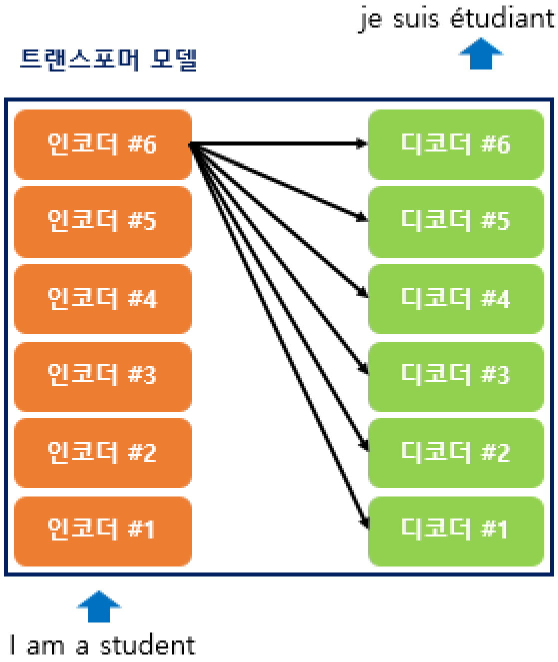
"Self-Attention"



BERT-Transformer: All you need is Attention

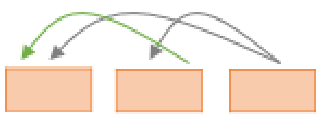


“RNN 구조를 사용하지 않고,
Attention만을 사용해도
원하는 결과를 얻어낼 수 있지 않을까?”

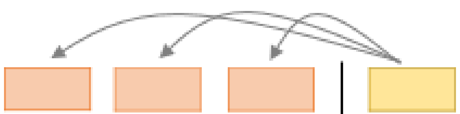


BERT-Transformer

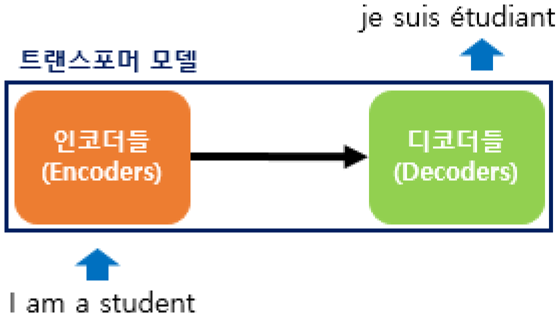
Encoder Self-Attention

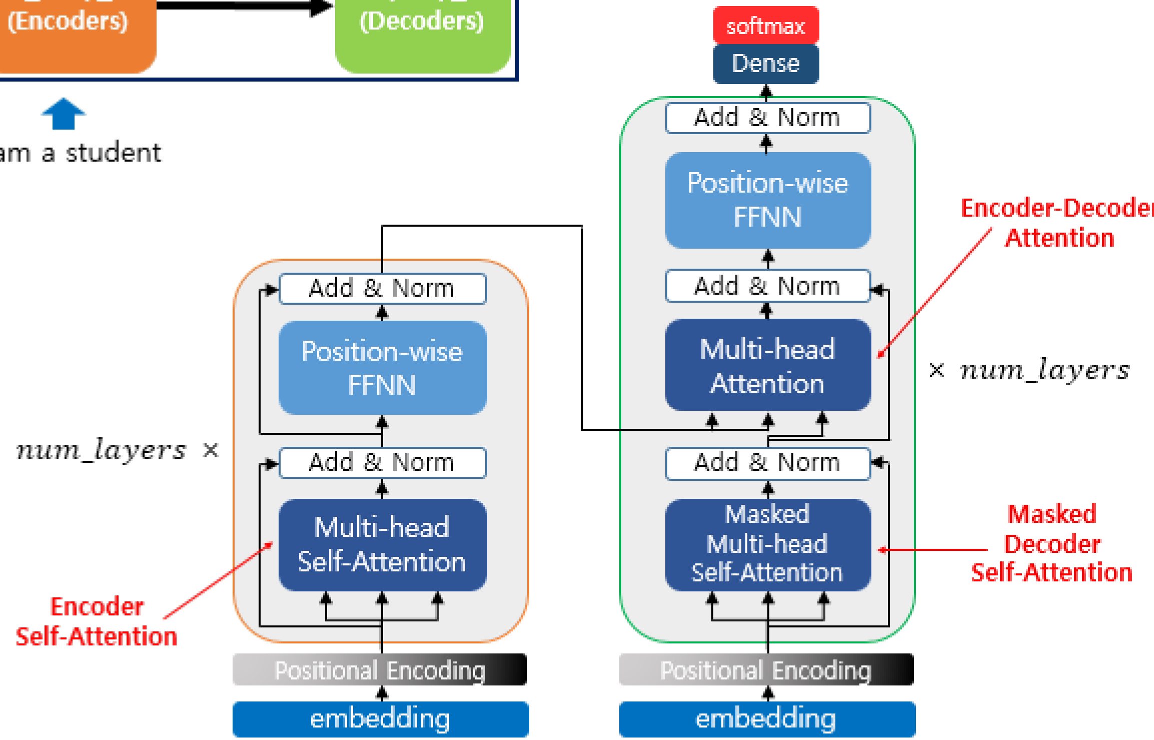


Masked Decoder Self-Attention

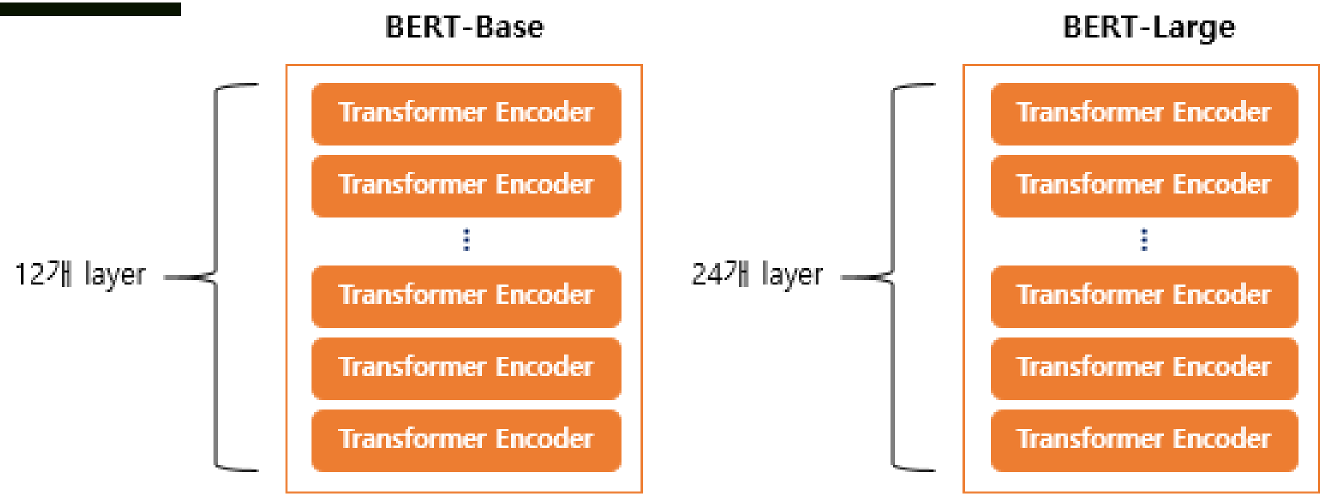


Encoder-Decoder Attention

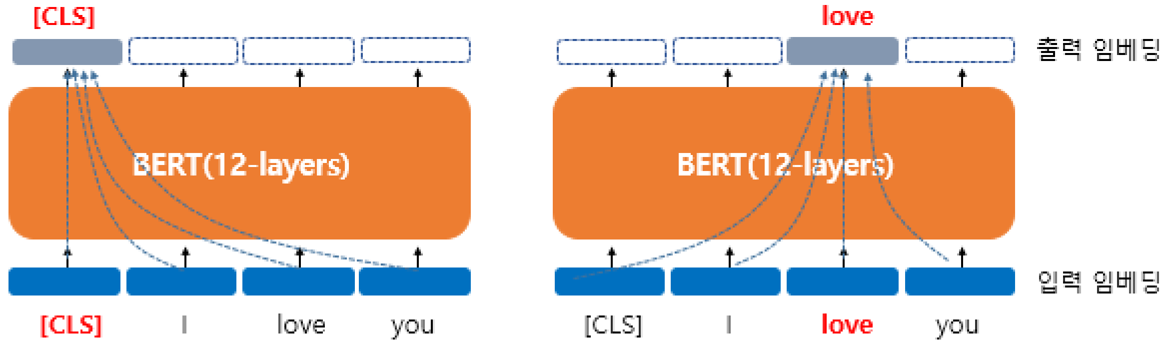




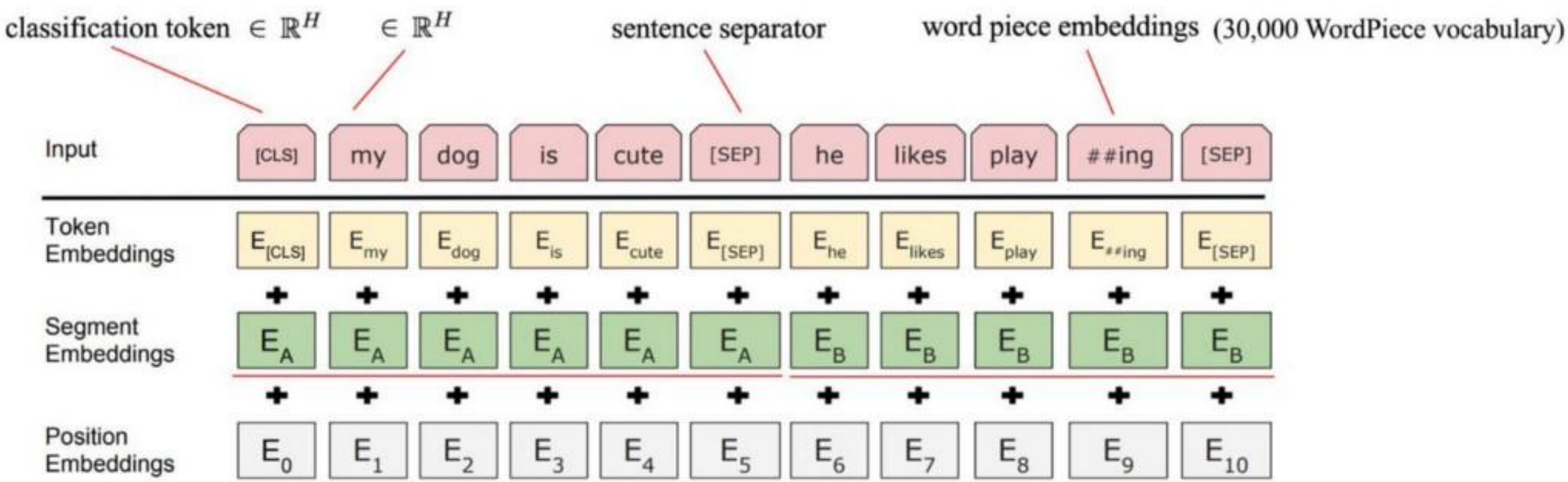
BERT



transformer 중에서도
encoder 부분만을 사용

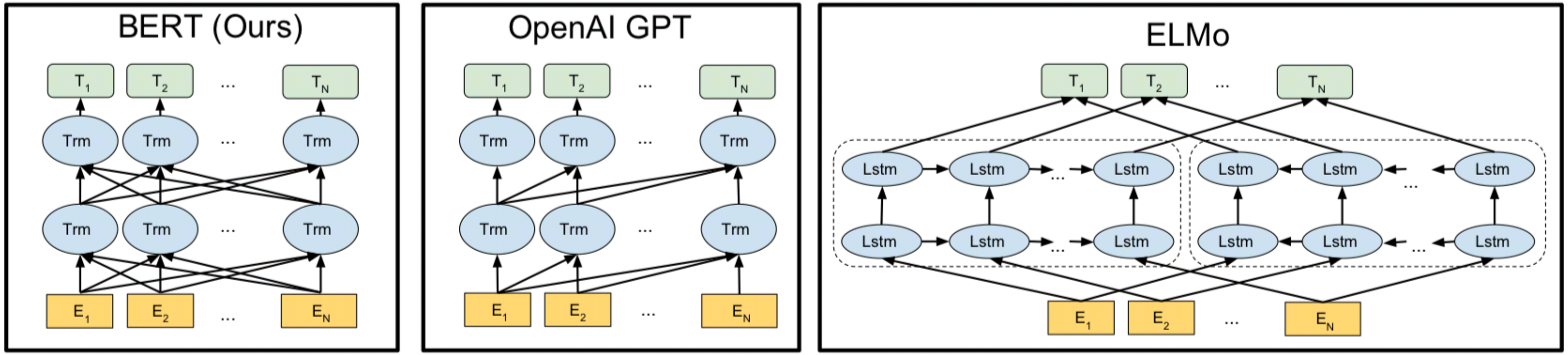


BERT


classification token $\in \mathbb{R}^H$
$\in \mathbb{R}^H$
sentence separator
word piece embeddings (30,000 WordPiece vocabulary)
Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$
E_{my}
E_{dog}
E_{is}
E_{cute}
$E_{[SEP]}$
E_{he}
E_{likes}
E_{play}
$E_{\#\#ing}$
$E_{[SEP]}$
Segment Embeddings
E_A
E_A
E_A
E_A
E_A
E_A
E_B
E_B
E_B
E_B
E_B
Position Embeddings
E_0
E_1
E_2
E_3
E_4
E_5
E_6
E_7
E_8
E_9
E_{10}

BERT

새로운 unsupervised prediction task로 pre-training을 수행



감사합니다

Q&A