안녕하세요!

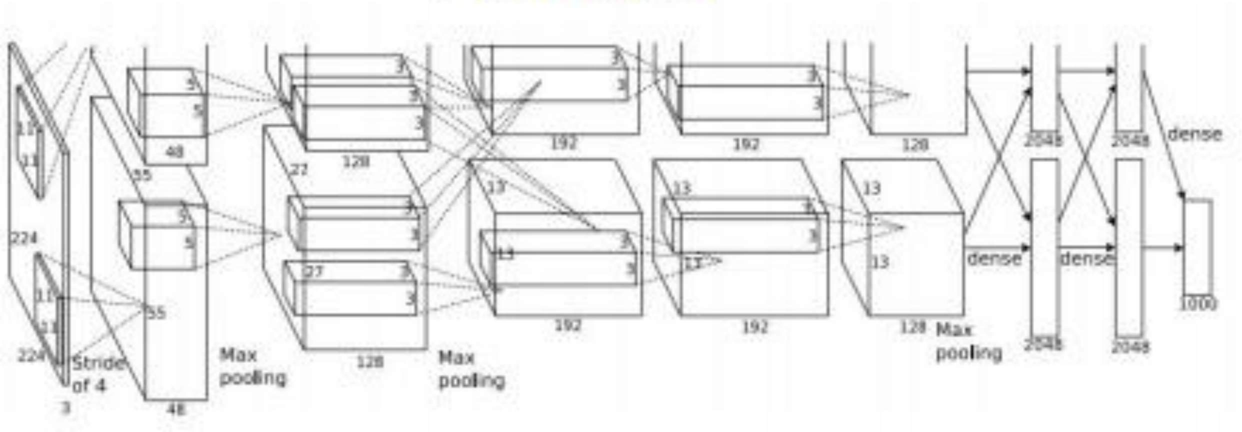
Lecture 10. Recurrent Neural Networks

김영은

Review

Last Time: CNN Architectures

AlexNet



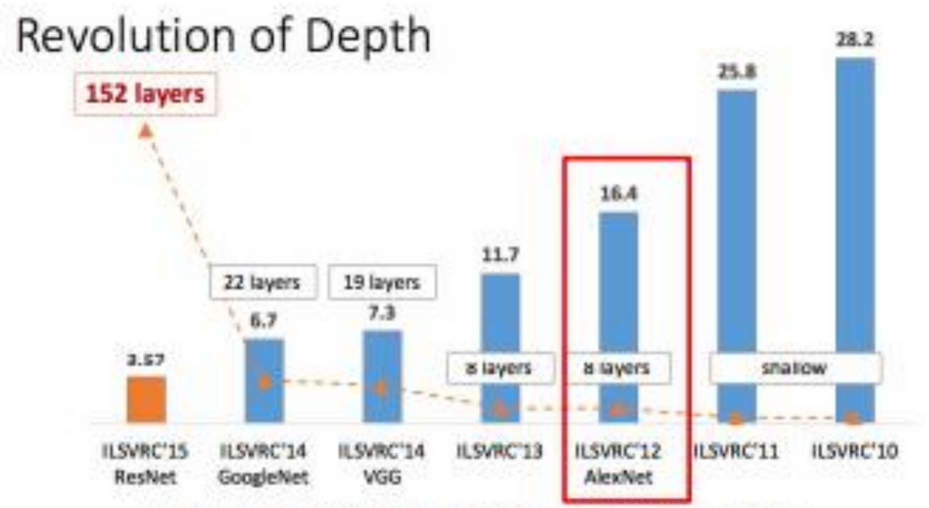
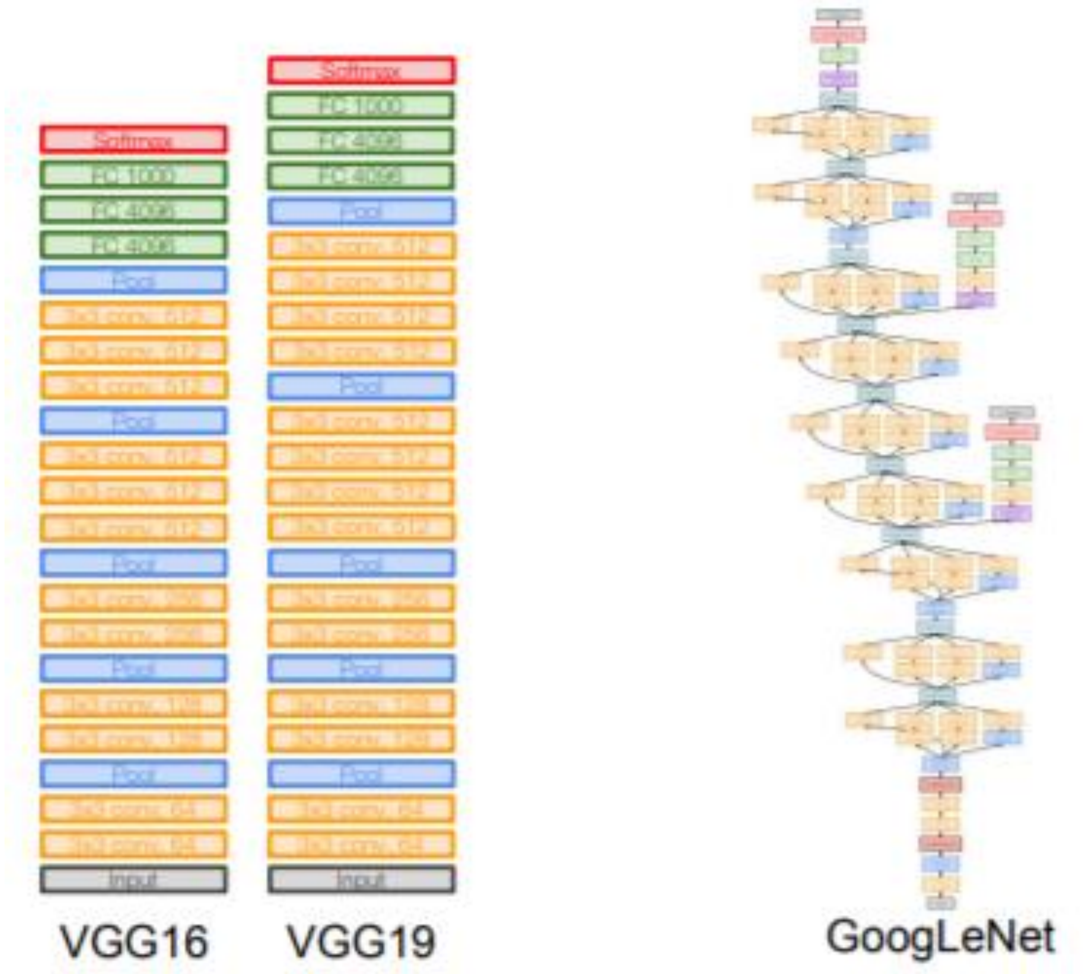


Figure copyright Kaiming He, 2016. Reproduced with permission.

Review

Last Time: CNN Architectures

GoogleNet(2014) – Auxillary classifier 도입
성능을 올리기 위함이 아니라 단지 네트워크의 초기 레이어에 gradient를 직접 흘려보내기 위한 수단.



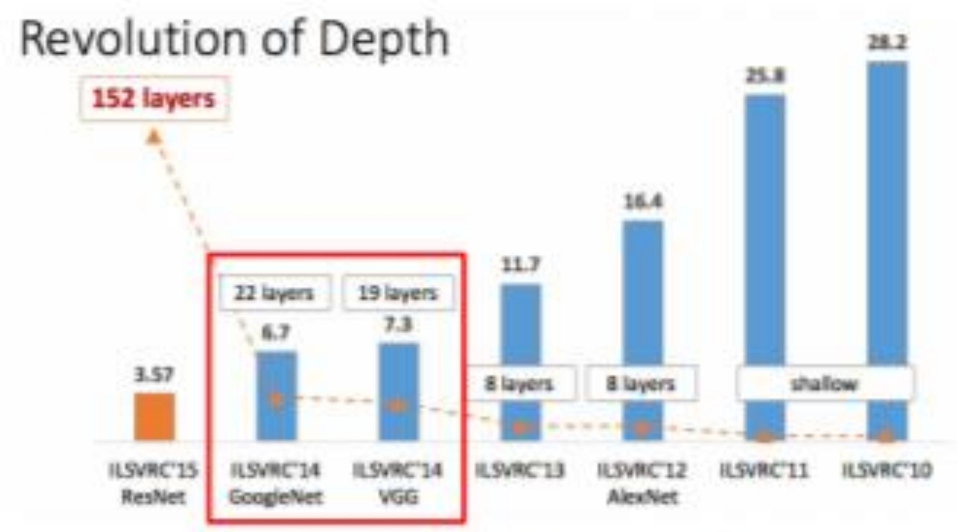
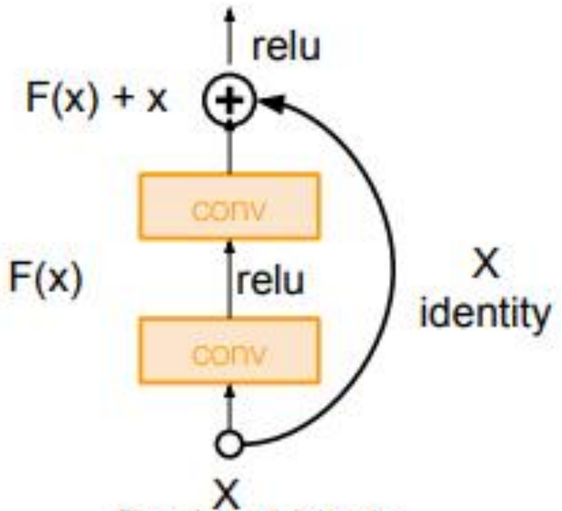


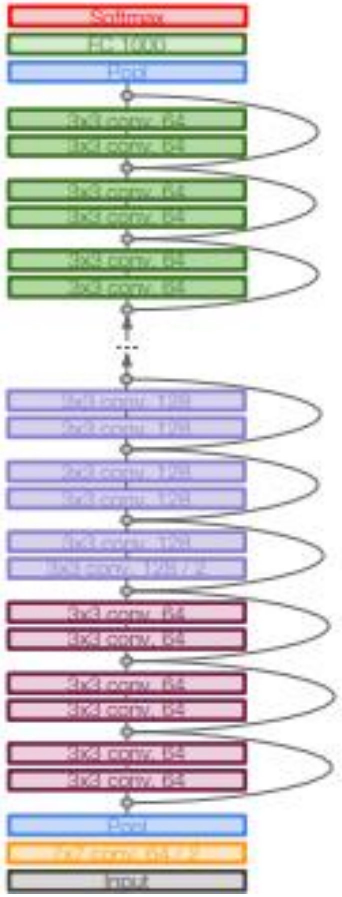
Figure copyright Kaiming He, 2016. Reproduced with permission.

Review

Last Time: CNN Architectures



Residual block



ResNet(2015)
=Short connection & Residual block
-필요없는 레이어를 사용하지 않도록 학습하는데 아주 유용.
-Backward pass에서의 gradient flow
Residual connection은 gradient을 위한 일종의 고속도로 역할!

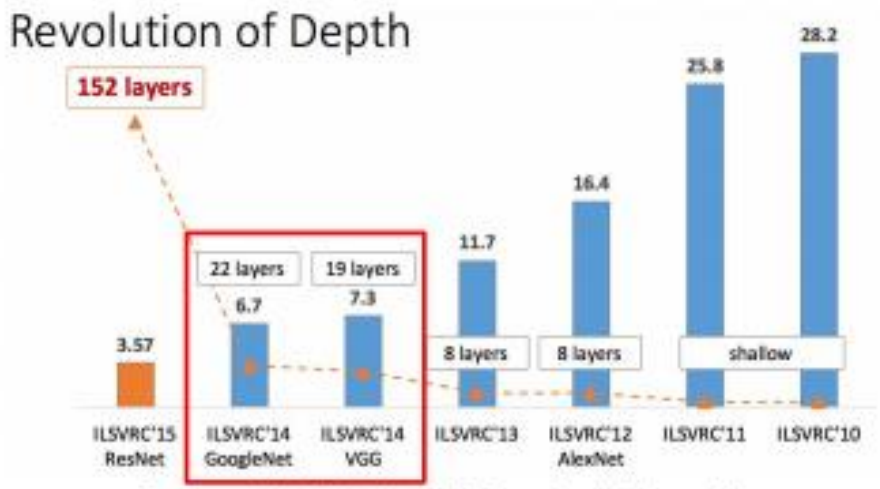
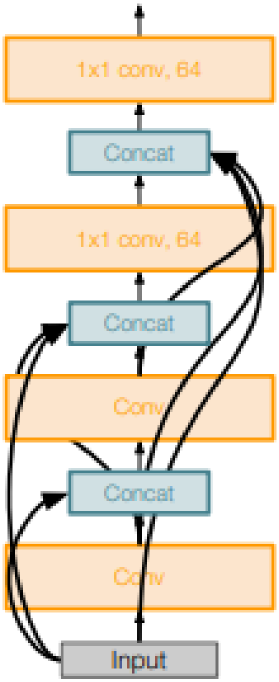


Figure copyright Kaiming He, 2016. Reproduced with permission.

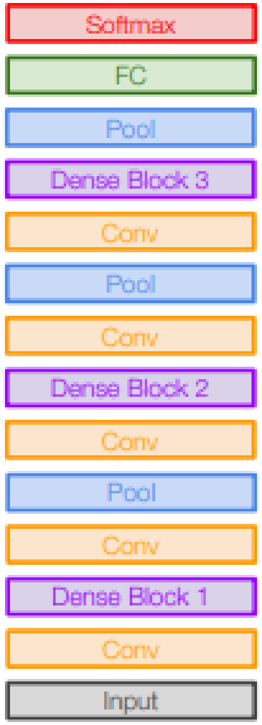
Review

Last Time: CNN Architectures

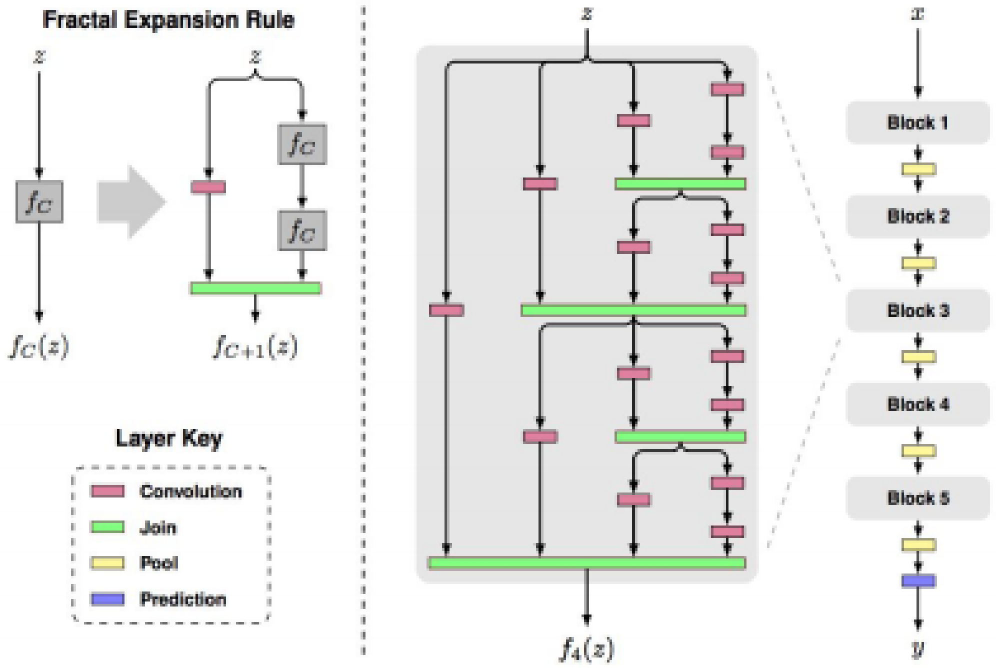
DenseNet



Dense Block



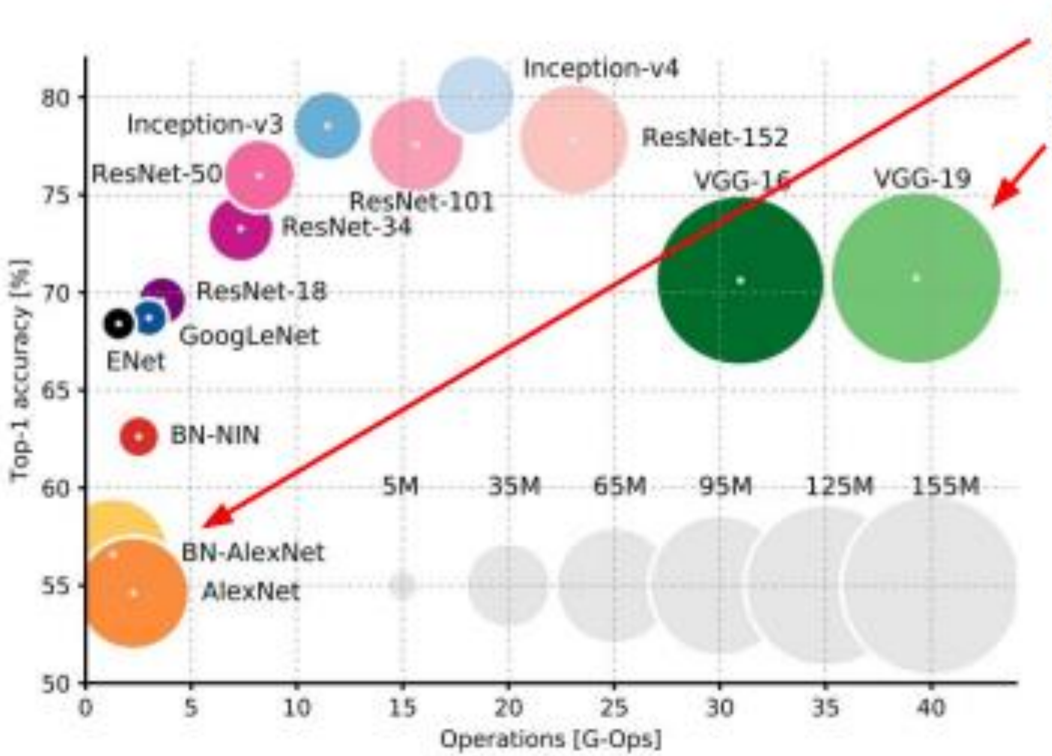
FractalNet



Figures copyright Larsson et al., 2017. Reproduced with permission.

Review

Last Time: CNN Architectures



AlexNet and VGG have tons of parameters in the fully connected layers

AlexNet: ~62M parameters

FC6: 256x6x6 -> 4096: 38M params
FC7: 4096 -> 4096: 17M params
FC8: 4096 -> 1000: 4M params
~59M params in FC layers!

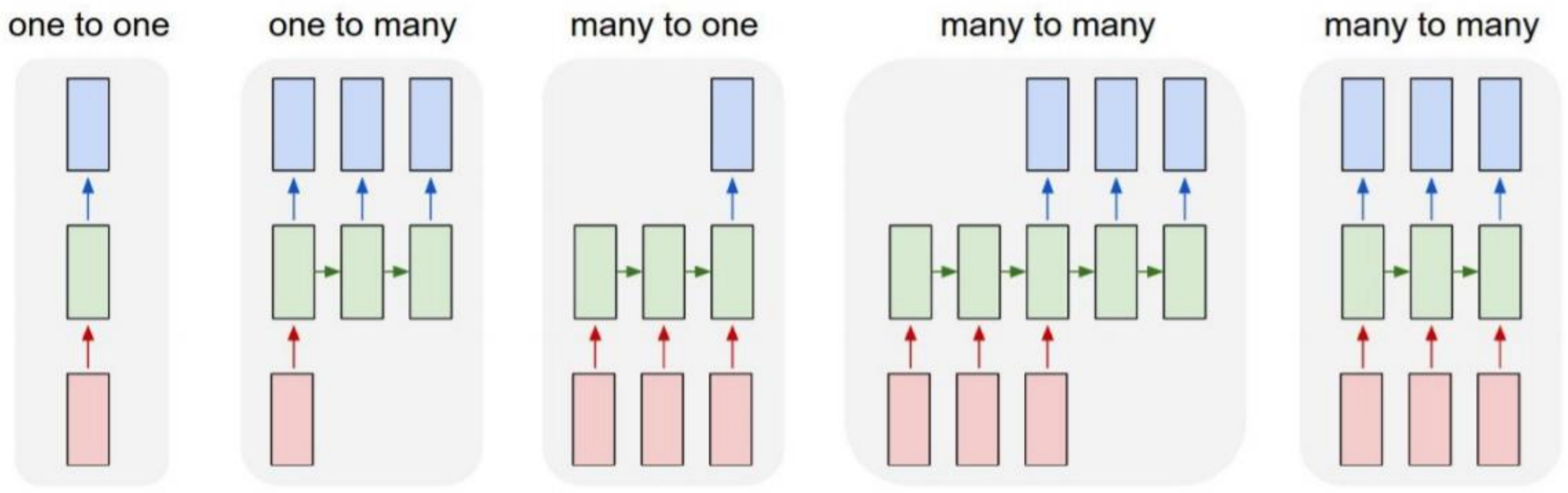
→ GoogleNet, ResNet 아키텍처들은 FC 대신 Golbal Average Pooling(GAP) 사용

INDEX

1. RNN?
2. Truncated Backpropagation through time
3. RNN Language Model
4. Image Captioning
5. LSTM

1 RNN

Recurrent Neural Networks: Process Sequences



Vanilla Neural Networks

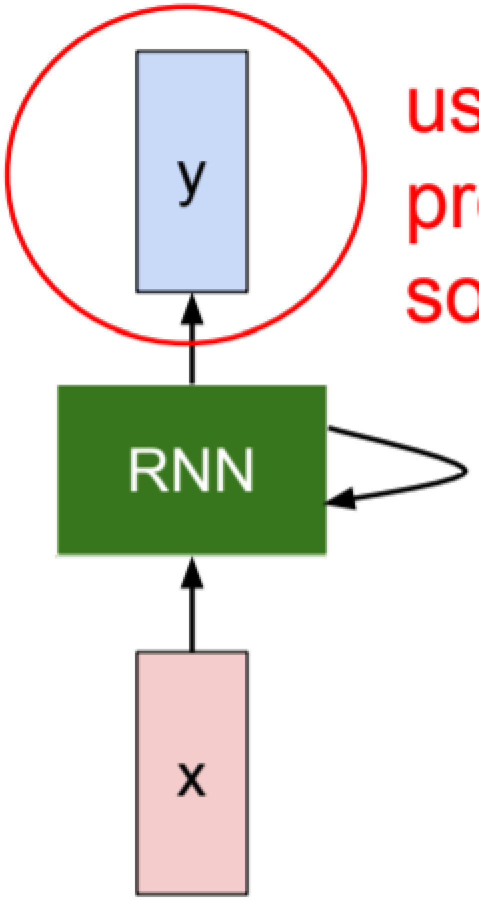
Image Captioning

Sentiment Classification

Machine Translation

Video Classification On frame level

Recurrent Neural Network



usually want to predict a vector at some time steps

내부에 Hidden state 가 지고 있다.
새로운 입력을 받을 때마다 매번 업데이트된다.

<순서>

1. RNN 입력받기
2. Hideen state 업데이트
3. 출력값 내보내기

Recurrent Neural Network

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

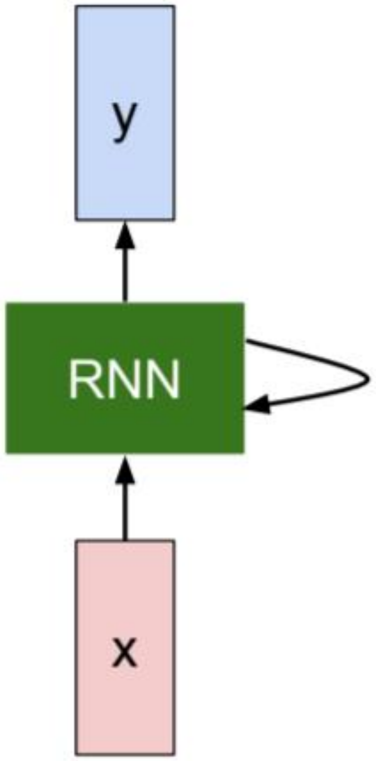
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

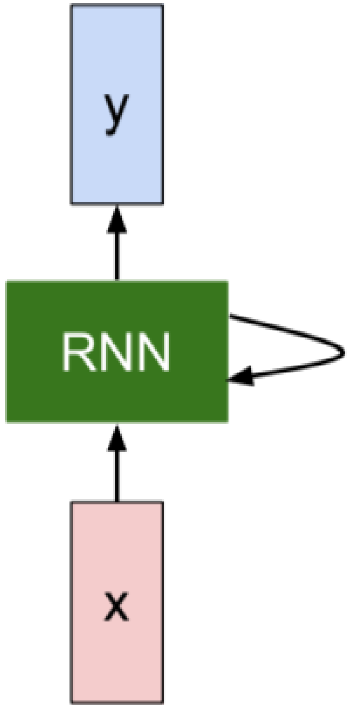
some function with parameters W



***함수 f와 파라미터 w는 모든 스텝마다 동일하다.

(Vanilla) Recurrent Neural Network

The state consists of a single *“hidden”* vector **h**:



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

$$\downarrow$$

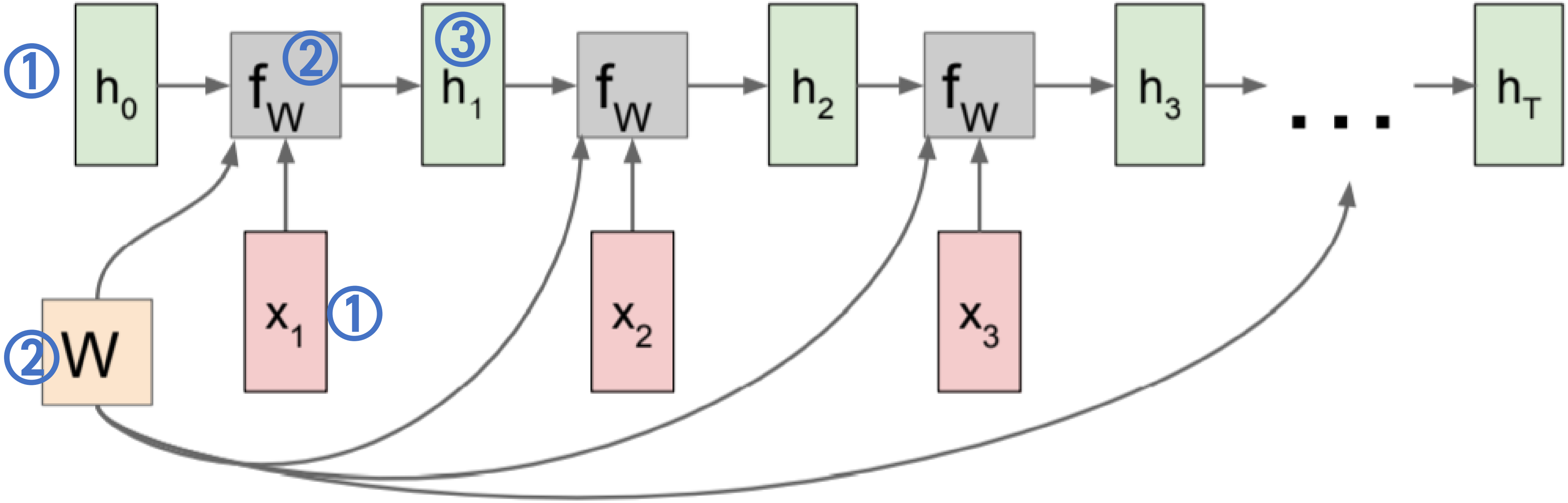
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

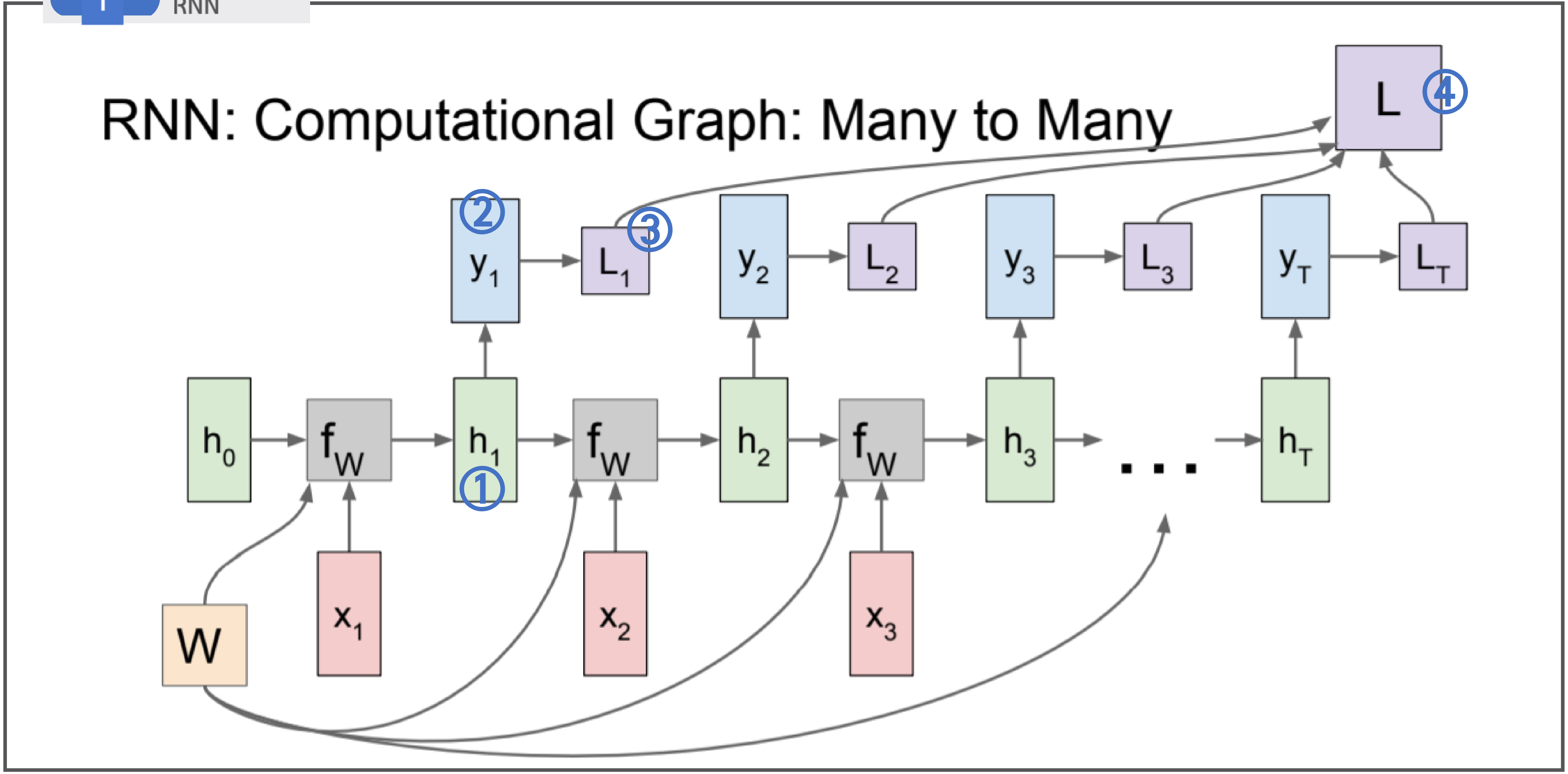
$$y_t = W_{hy}h_t$$

1 RNN

RNN: Computational Graph

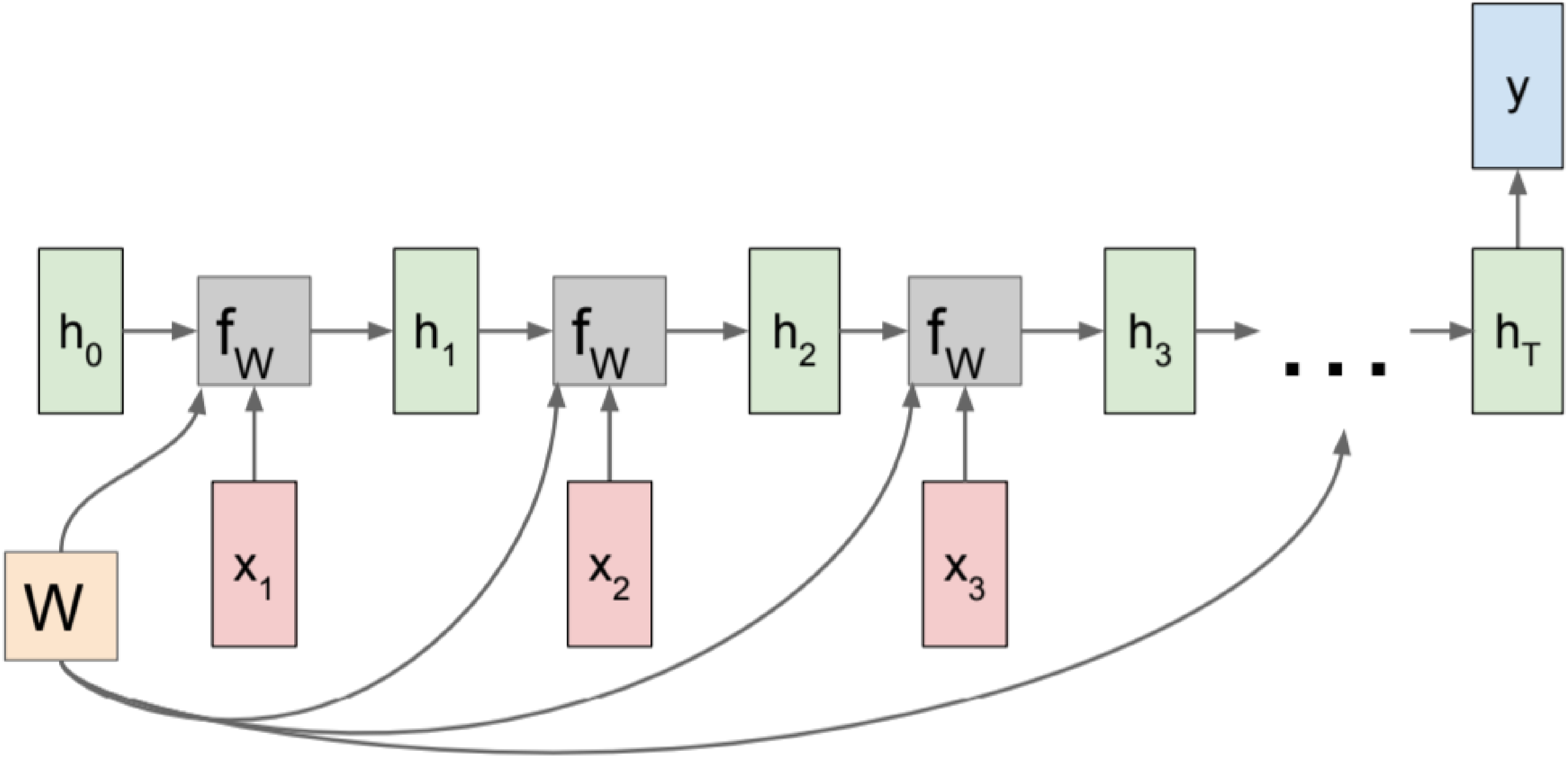
Re-use the same weight matrix at every time-step





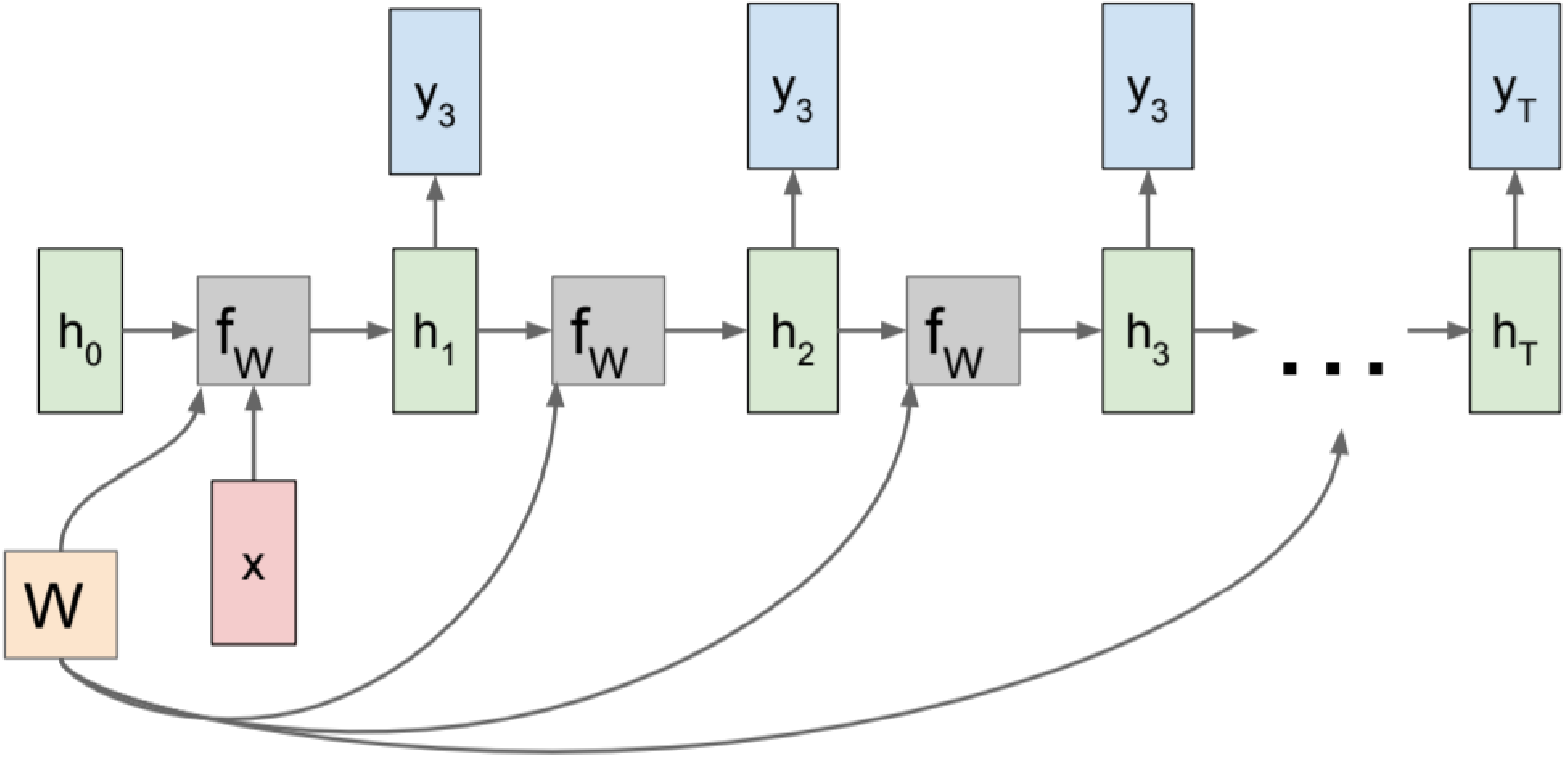
RNN: Computational Graph: Many to Many

RNN: Computational Graph: Many to One 예. 감정분석



1 RNN

RNN: Computational Graph: One to Many



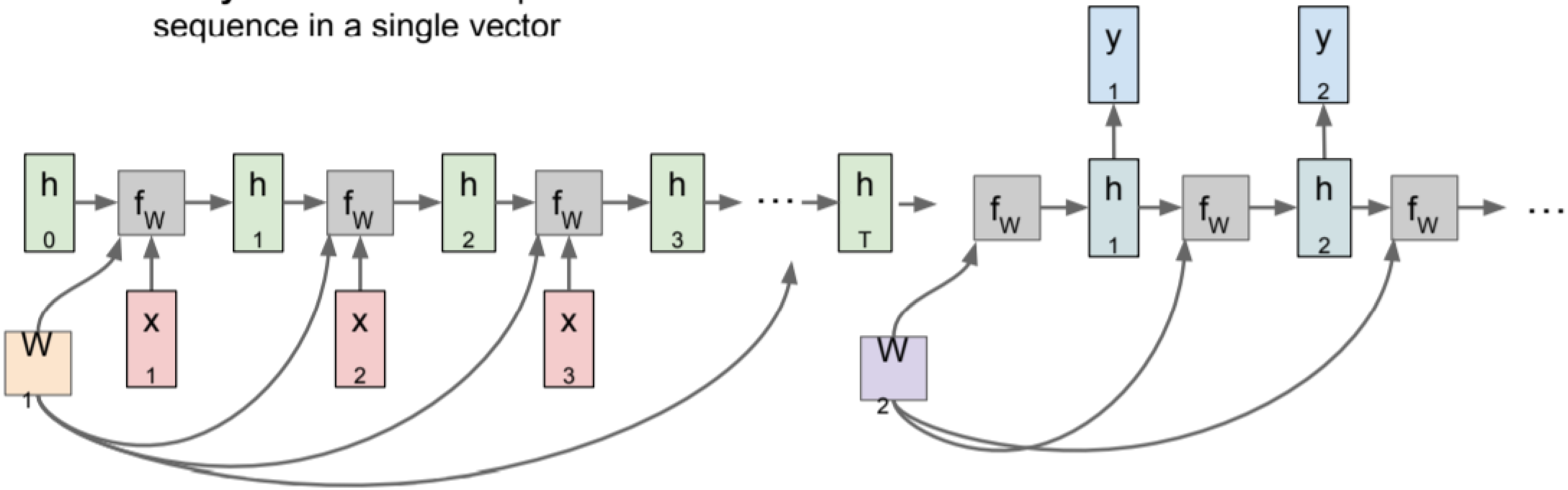
Sequence to Sequence: Many-to-one + one-to-many

다른 언어로 번역된 문장

One to many: Produce output sequence from single input vector

영어 문장을 하나의 벡터로 요약

Many to one: Encode input sequence in a single vector

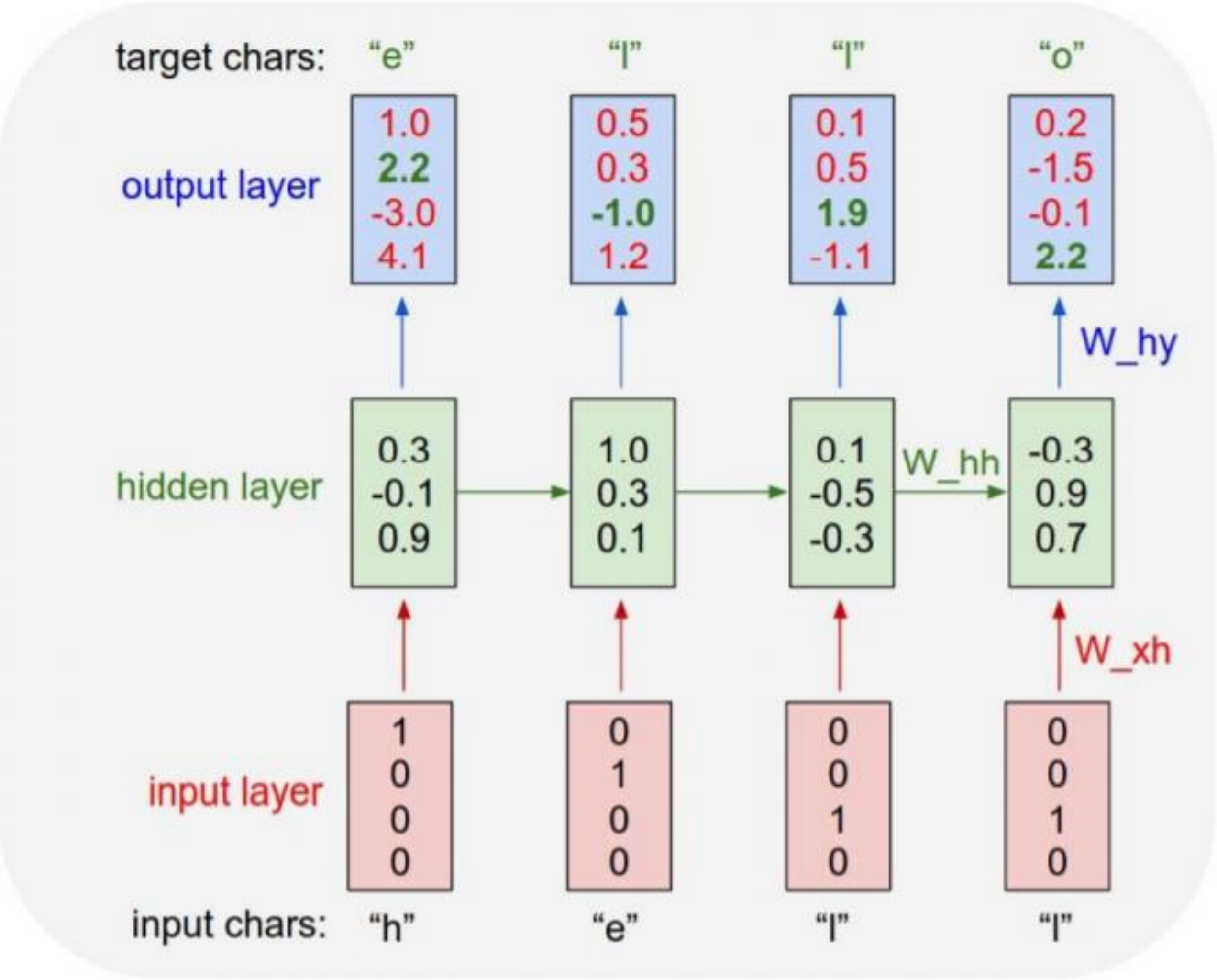


Example: Character-level Language Model

<Train time>
*H,e,l,o 단어로 H,e,l,l,o 학습시키기

*입력할 때 한글자씩! (1,0)
해당 글자 위치만 1로 표시하여 하나의 벡터로 표현

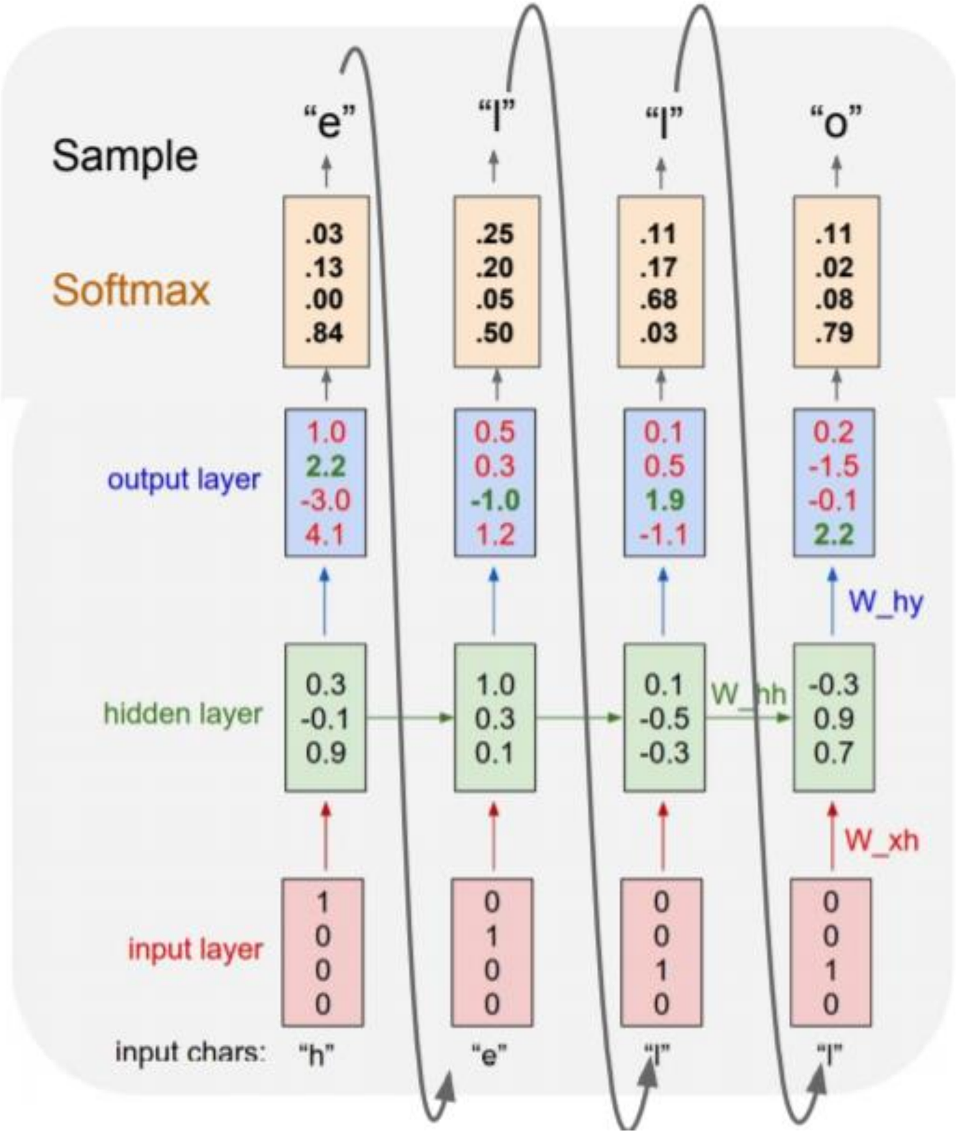
*y_t = 어떤 문자가 h 다음에 올지를 예측하는 것



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

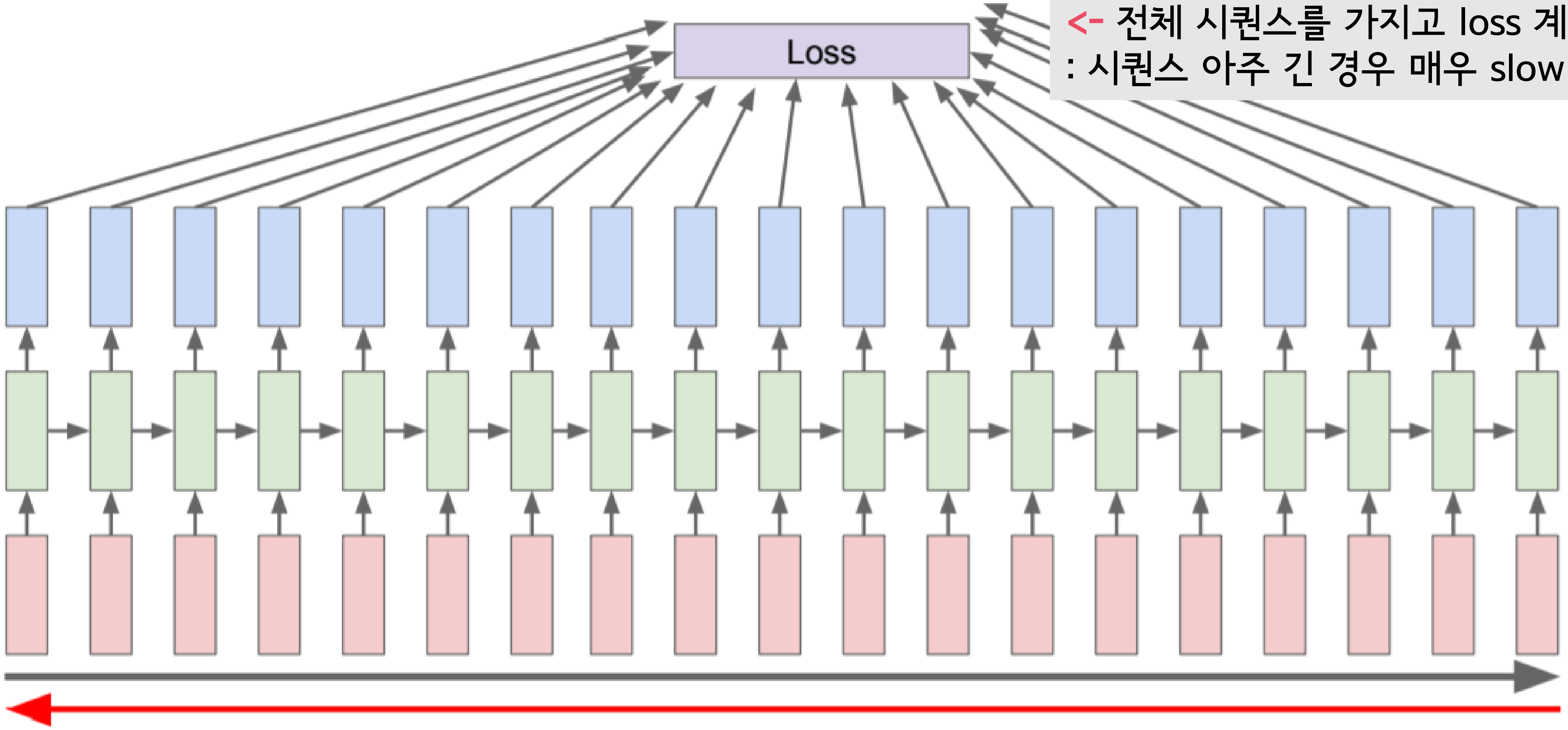
At test-time sample characters one at a time, feed back to model



2 Truncated Backpropagtion

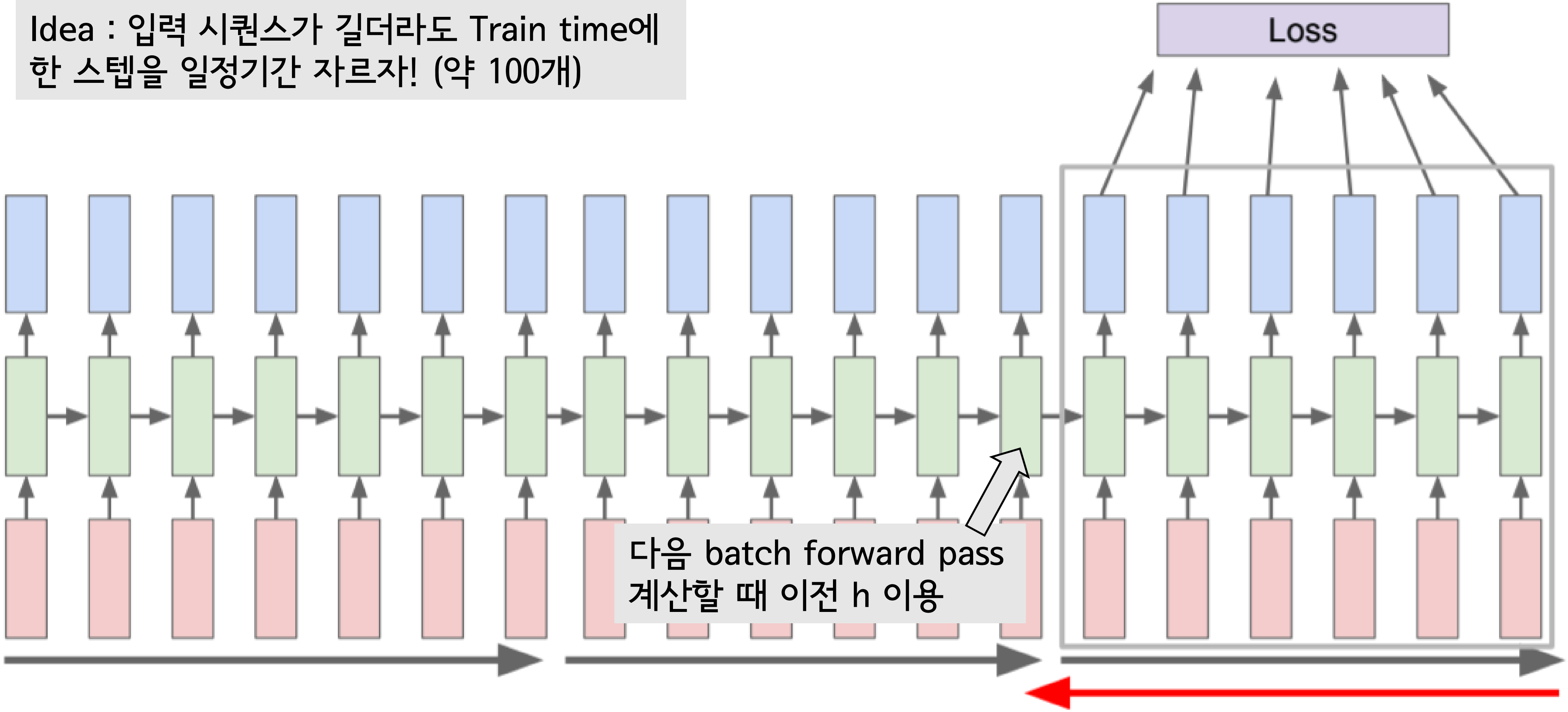
Forward pass
-> 전체 시퀀스가 끝날 때까지 출력값 생성
Backward pass
<- 전체 시퀀스를 가지고 loss 계산
: 시퀀스 아주 긴 경우 매우 slow! Bad!

Backpropagation through time



Truncated Backpropagation through time

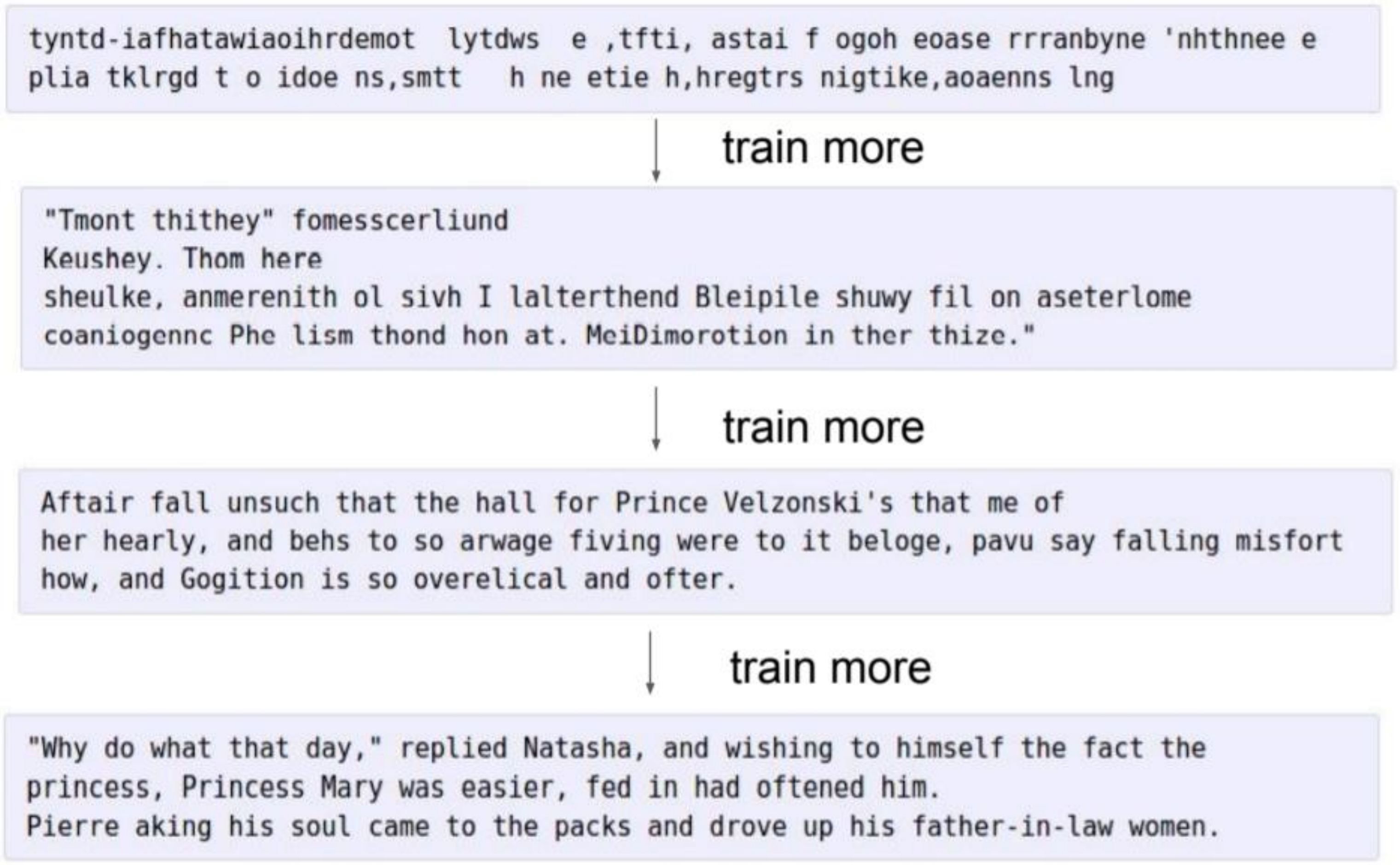
Idea : 입력 시퀀스가 길더라도 Train time에 한 스텝을 일정기간 자르자! (약 100개)



다음 batch forward pass 계산할 때 이전 h 이용

3 RNN Language Model 예시

세익스피어 작품들



```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

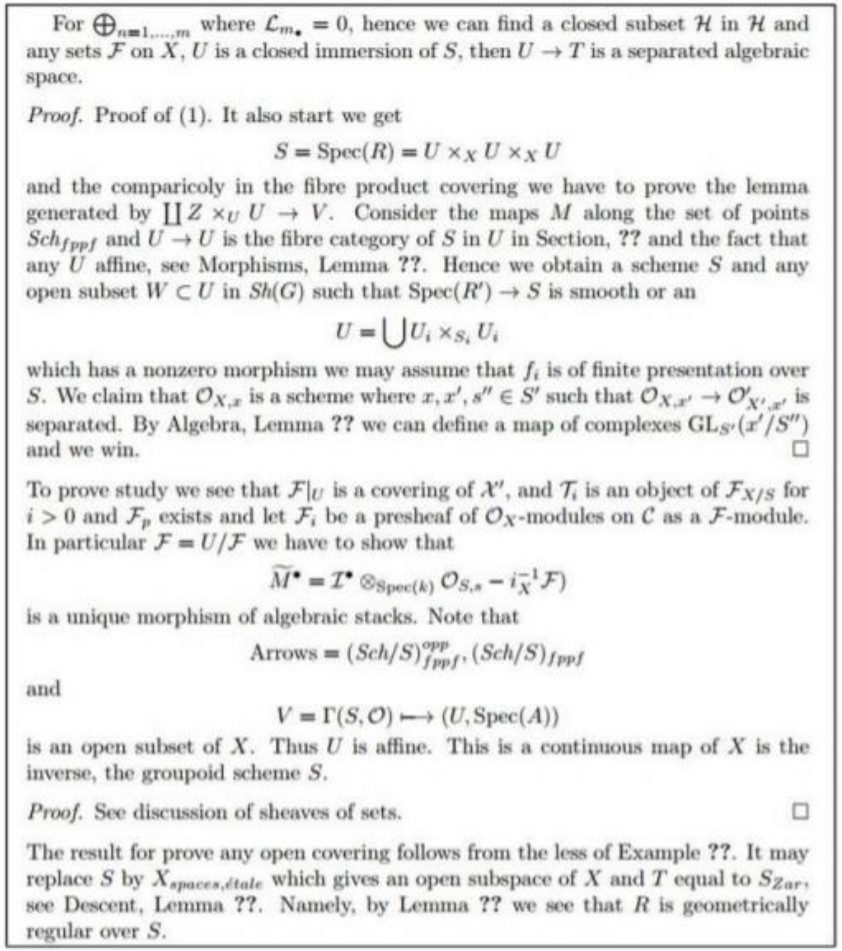
```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had ofteneded him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

-RNN으로 어떤 문장이든 학습 O
-학습을 시킬수록
의미 있는 문장 O
긴 문장 O
다른 문장 시작할 때 한 줄 띄우기 O
....

수학 문장들(algebraic geometry textbook)



For $\bigoplus_{n=1,\dots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

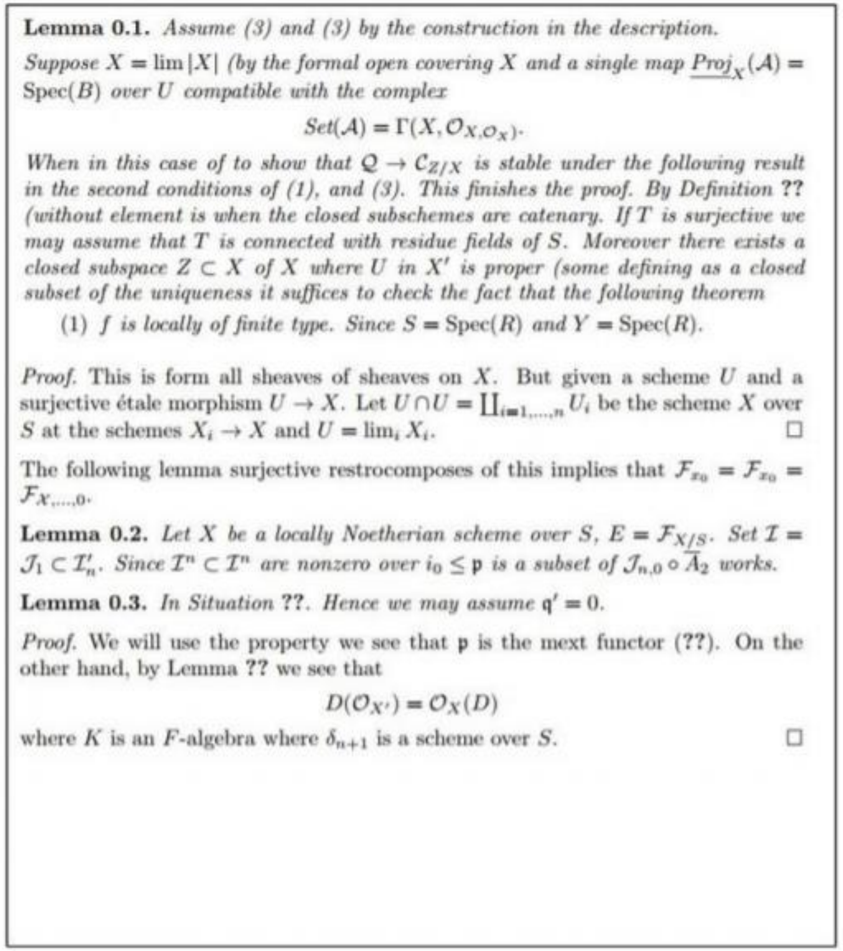
and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,\acute{e}tale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.



Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1) *f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.*

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\dots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X,\dots,0}$.

Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation ??. Hence we may assume $\mathfrak{q}' = 0$.*

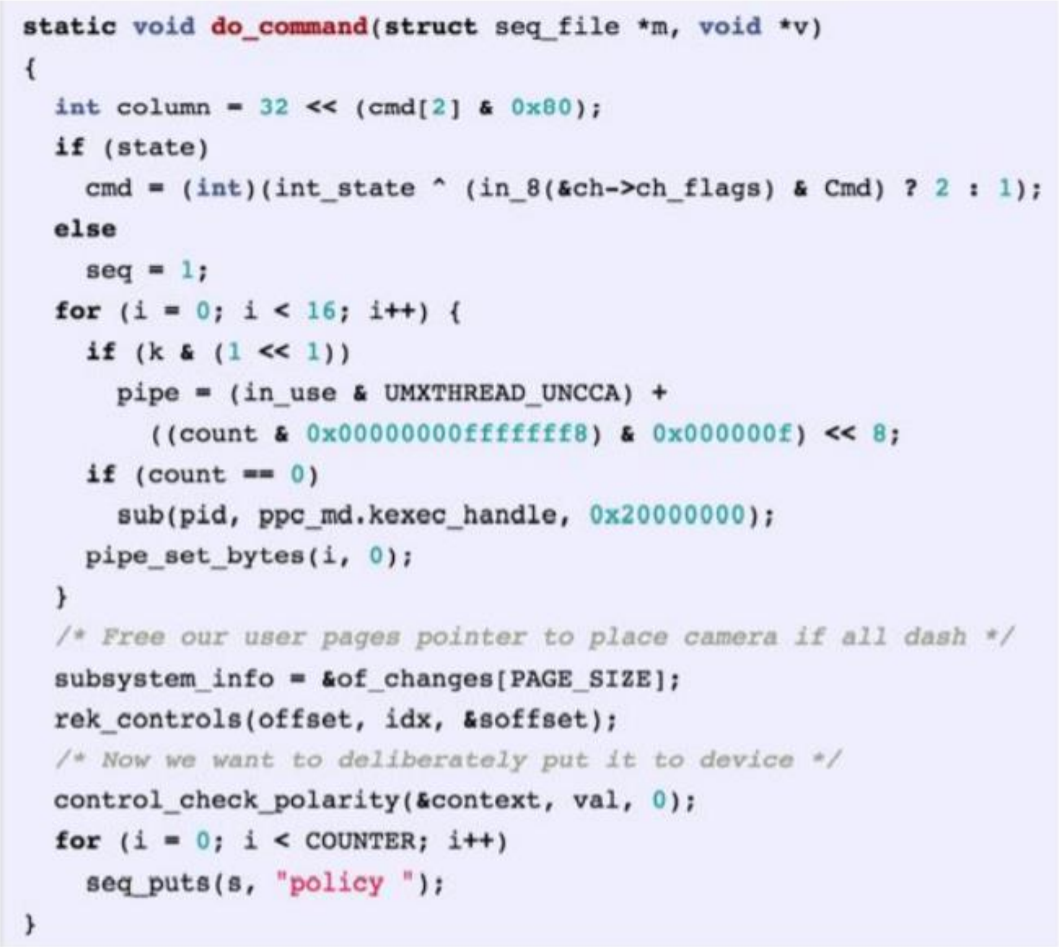
Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

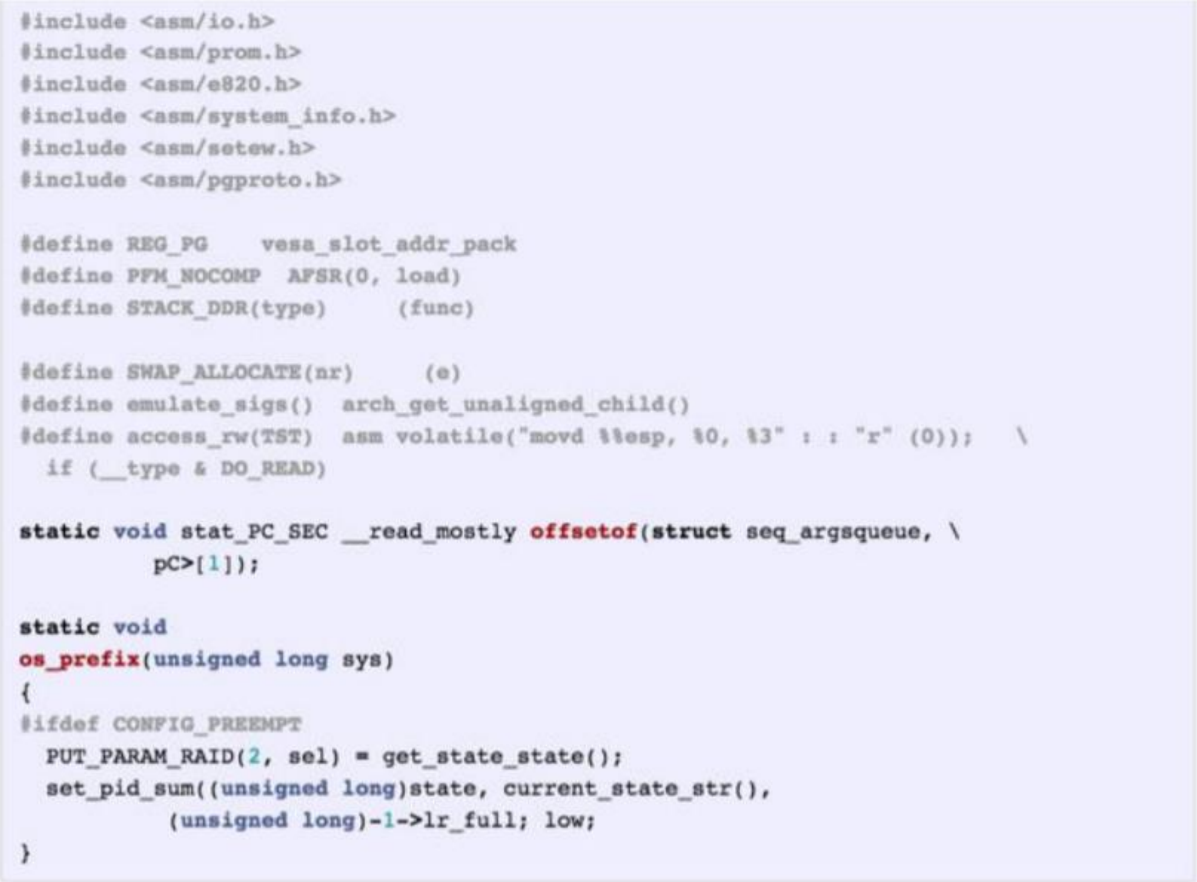
where K is an F-algebra where δ_{n+1} is a scheme over S. □

3 RNN Language Model 예시

C code

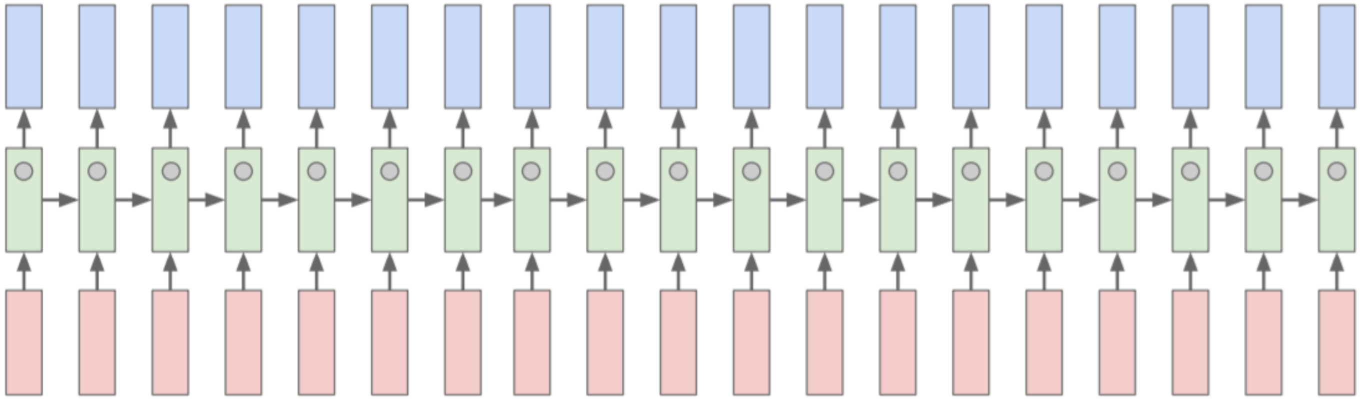


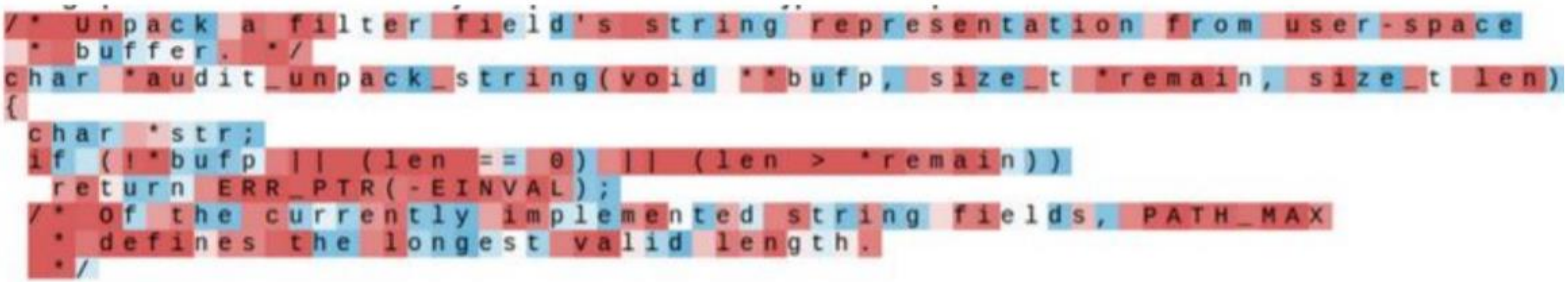
```
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```



```
#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>

#define REG_PG    vesa_slot_addr_pack
#define PFM_NOCOMP  AFSR(0, load)
#define STACK_DDR(type)     (func)

#define SWAP_ALLOCATE(nr)     (e)
#define emulate_sigs()  arch_get_unaligned_child()
#define access_rw(TST)  asm volatile("movd %%esp, %0, %3" : : "r" (0));   \
  if (__type & DO_READ)

static void stat_PC_SEC __read_mostly offsetof(struct seq_argsqueue, \
          pC>[1]);

static void
os_prefix(unsigned long sys)
{
#ifdef CONFIG_PREEMPT
  PUT_PARAM_RAID(2, sel) = get_state_state();
  set_pid_sum((unsigned long)state, current_state_str(),
           (unsigned long)-1->lr_full; low;
}
```

Searching for interpretable cells

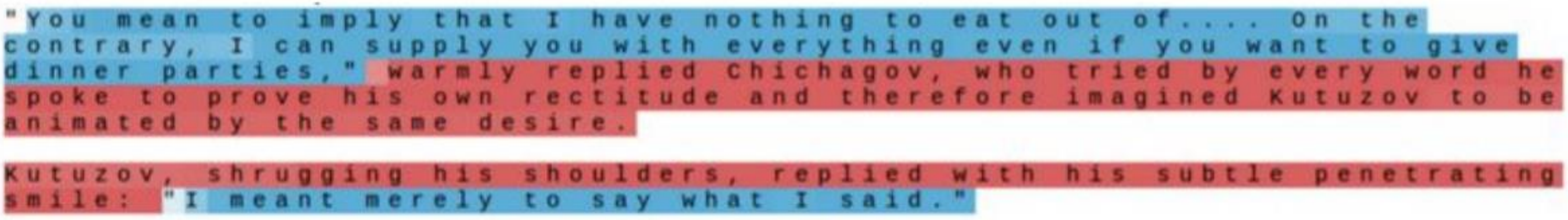




```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
 char *str;
 if (!*bufp || (len == 0) || (len > *remain))
  return ERR_PTR(-EINVAL);
 /* Of the currently implemented string fields, PATH_MAX
  * defines the longest valid length.
  */
```

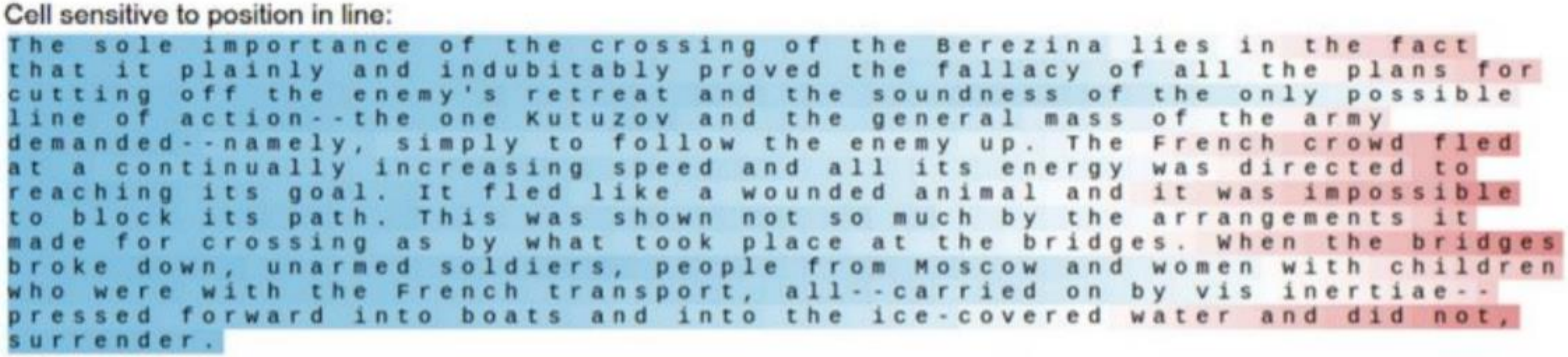
3 RNN Language Model 예시

Quote detection cell(따옴표 찾는 셀)



```
"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."
```

line length tracking cell(각 줄의 단어수 세는 셀)

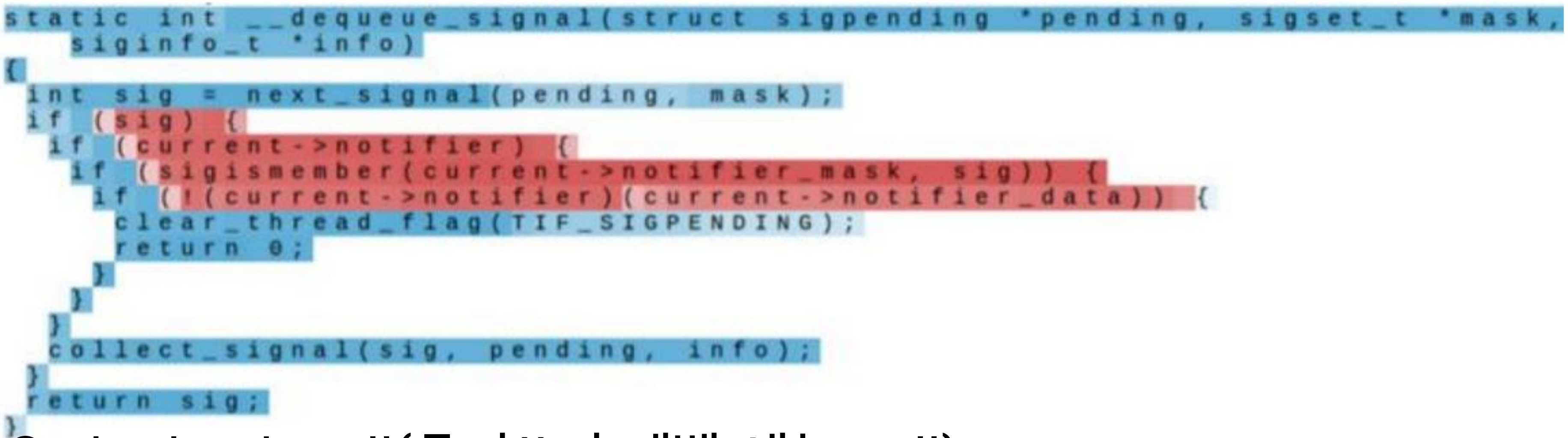


Cell sensitive to position in line:

```
The sole importance of the crossing of the Berezina lies in the fact
that it plainly and indubitably proved the fallacy of all the plans for
cutting off the enemy's retreat and the soundness of the only possible
line of action--the one Kutuzov and the general mass of the army
demanded--namely, simply to follow the enemy up. The French crowd fled
at a continually increasing speed and all its energy was directed to
reaching its goal. It fled like a wounded animal and it was impossible
to block its path. This was shown not so much by the arrangements it
made for crossing as by what took place at the bridges. When the bridges
broke down, unarmed soldiers, people from Moscow and women with children
who were with the French transport, all--carried on by vis inertiae--
pressed forward into boats and into the ice-covered water and did not,
surrender.
```

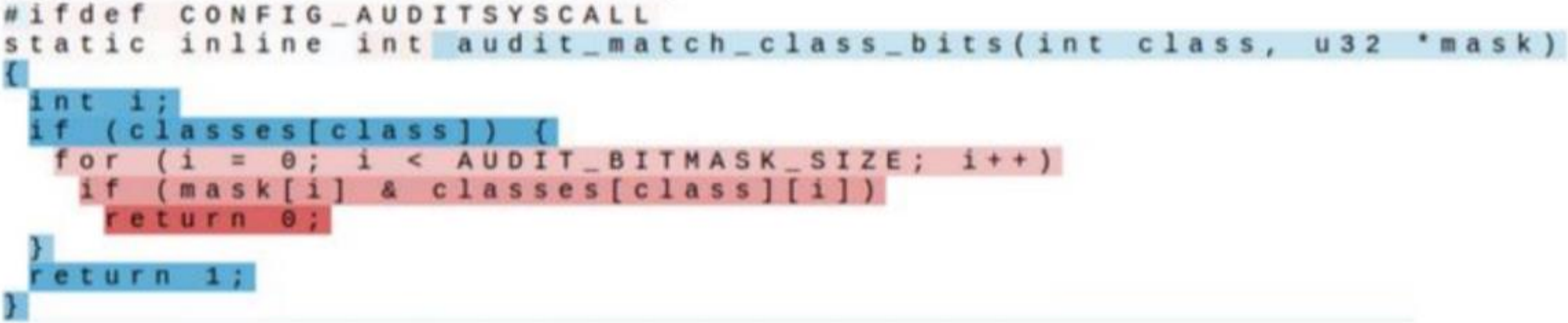
3 RNN Language Model 예시

If statement cell(조건부의 내부와 외부 구별하는 cell)



```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
   siginfo_t *info)
{
 int sig = next_signal(pending, mask);
 if (sig) {
  if (current->notifier) {
   if (sigismember(current->notifier_mask, sig)) {
    if (!(current->notifier)(current->notifier_data)) {
     clear_thread_flag(TIF_SIGPENDING);
     return 0;
    }
   }
  }
  collect_signal(sig, pending, info);
 }
 return sig;
}
```

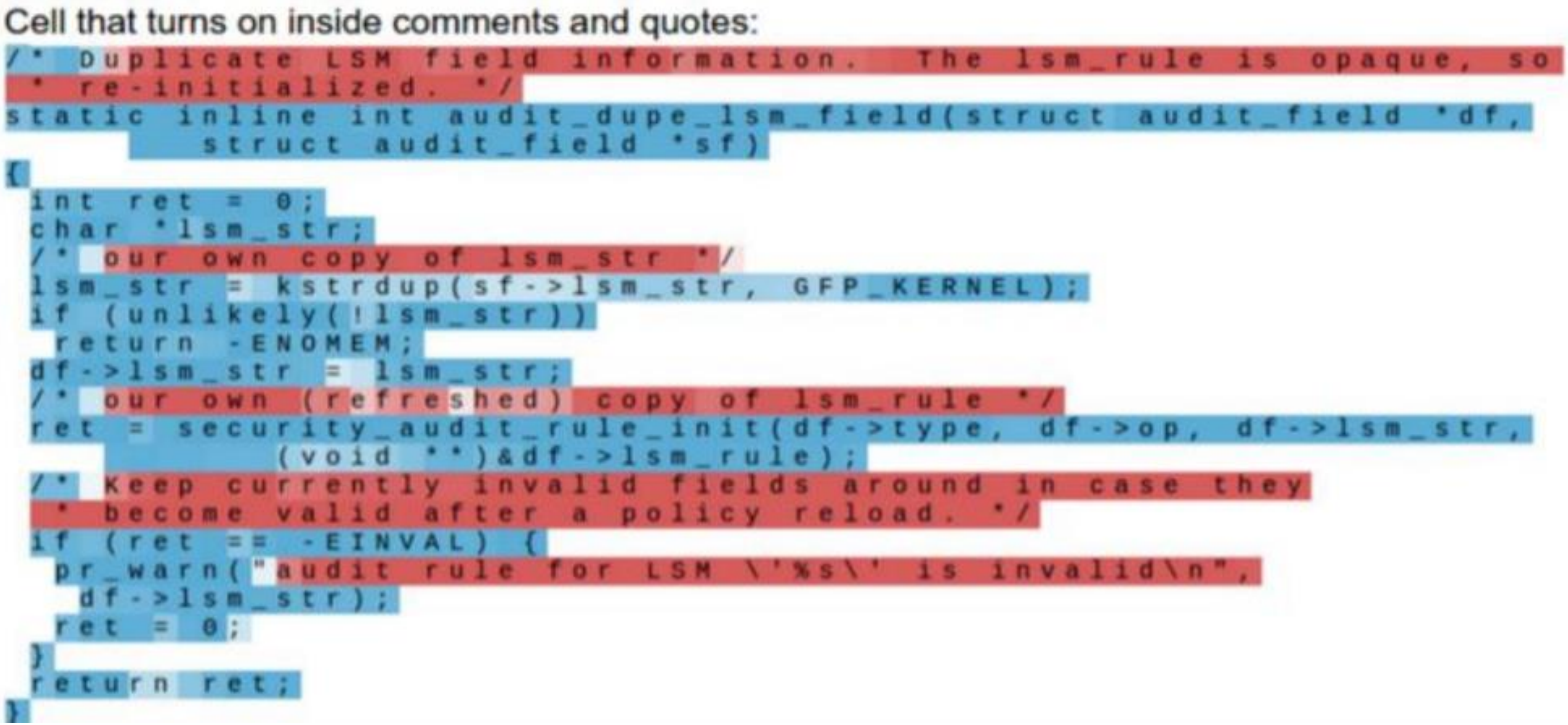
Code depth cell(들여쓰기 레벨 세는 cell)



```
#ifdef CONFIG_AUDITSYSCALL
static inline int audit_match_class_bits(int class, u32 *mask)
{
 int i;
 if (classes[class]) {
  for (i = 0; i < AUDIT_BITMASK_SIZE; i++)
   if (mask[i] & classes[class][i])
    return 0;
 }
 return 1;
}
```

3 RNN Language Model 예시

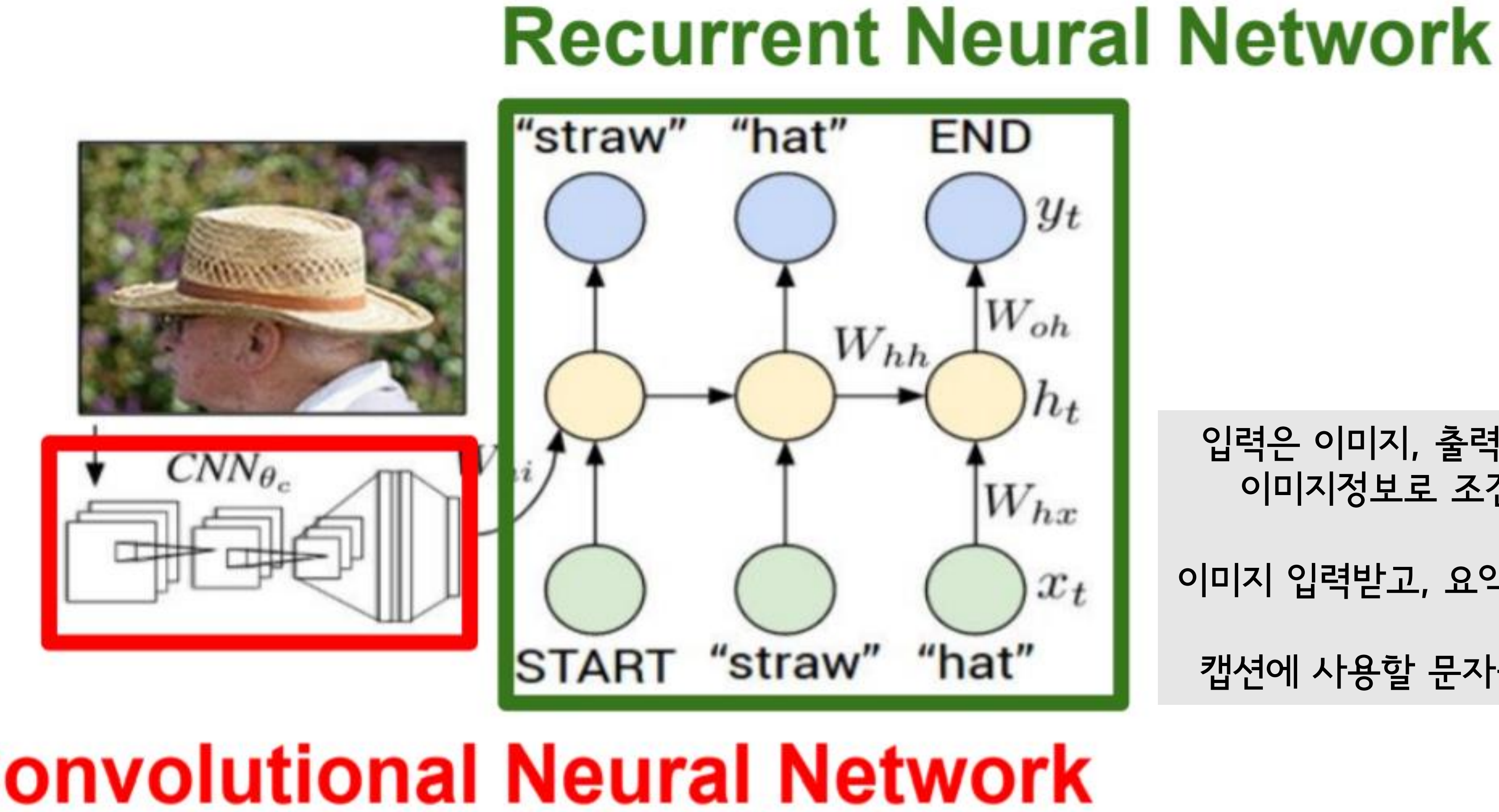
Quote/comment cell(코멘트 찾는 cell)



Cell that turns on inside comments and quotes:

```
/* Duplicate LSM field information.  The lsm_rule is opaque, so
 * re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
     struct audit_field *sf)
{
 int ret = 0;
 char *lsm_str;
 /* our own copy of lsm_str */
 lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
 if (unlikely(!lsm_str))
  return -ENOMEM;
 df->lsm_str = lsm_str;
 /* our own (refreshed) copy of lsm_rule */
 ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
           (void **)&df->lsm_rule);
 /* Keep currently invalid fields around in case they
  * become valid after a policy reload. */
 if (ret == -EINVAL) {
  pr_warn("audit rule for LSM \'%s\' is invalid\n",
   df->lsm_str);
  ret = 0;
 }
 return ret;
}
```

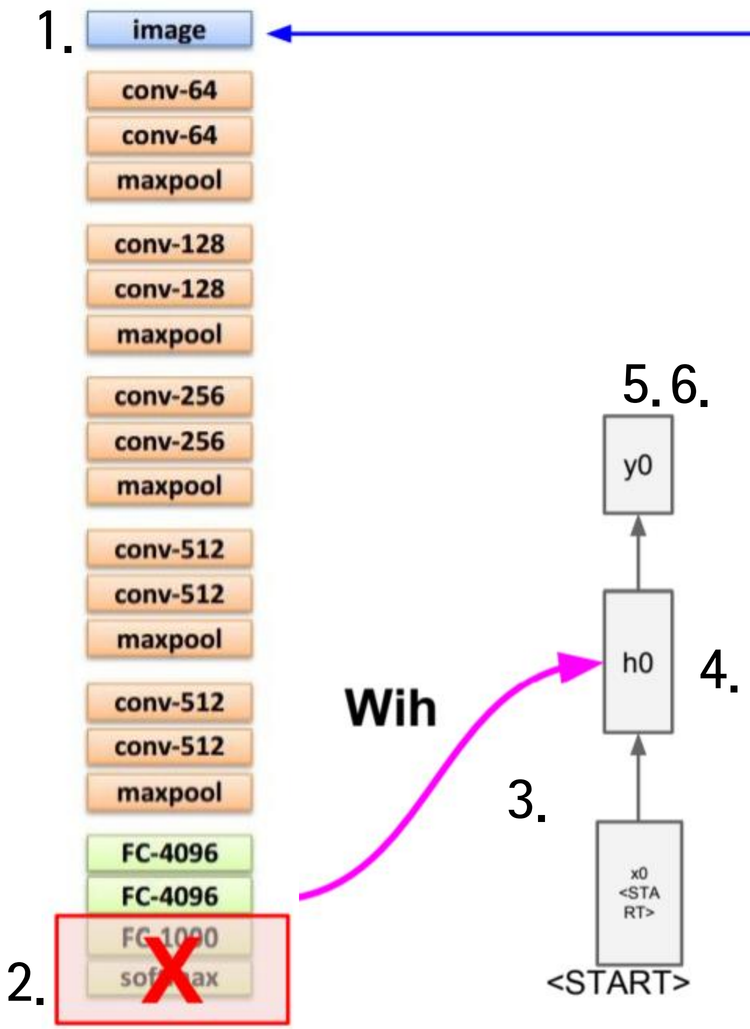
Image Captioning



입력은 이미지, 출력은 자연어로 된 caption!
이미지정보로 조건에 맞는 문장 만들기!

이미지 입력받고, 요약된 vector 출력하는 CNN
+
캡션에 사용할 문자들을 하나씩 만드는 RNN

4 Image captioning



test image

before:

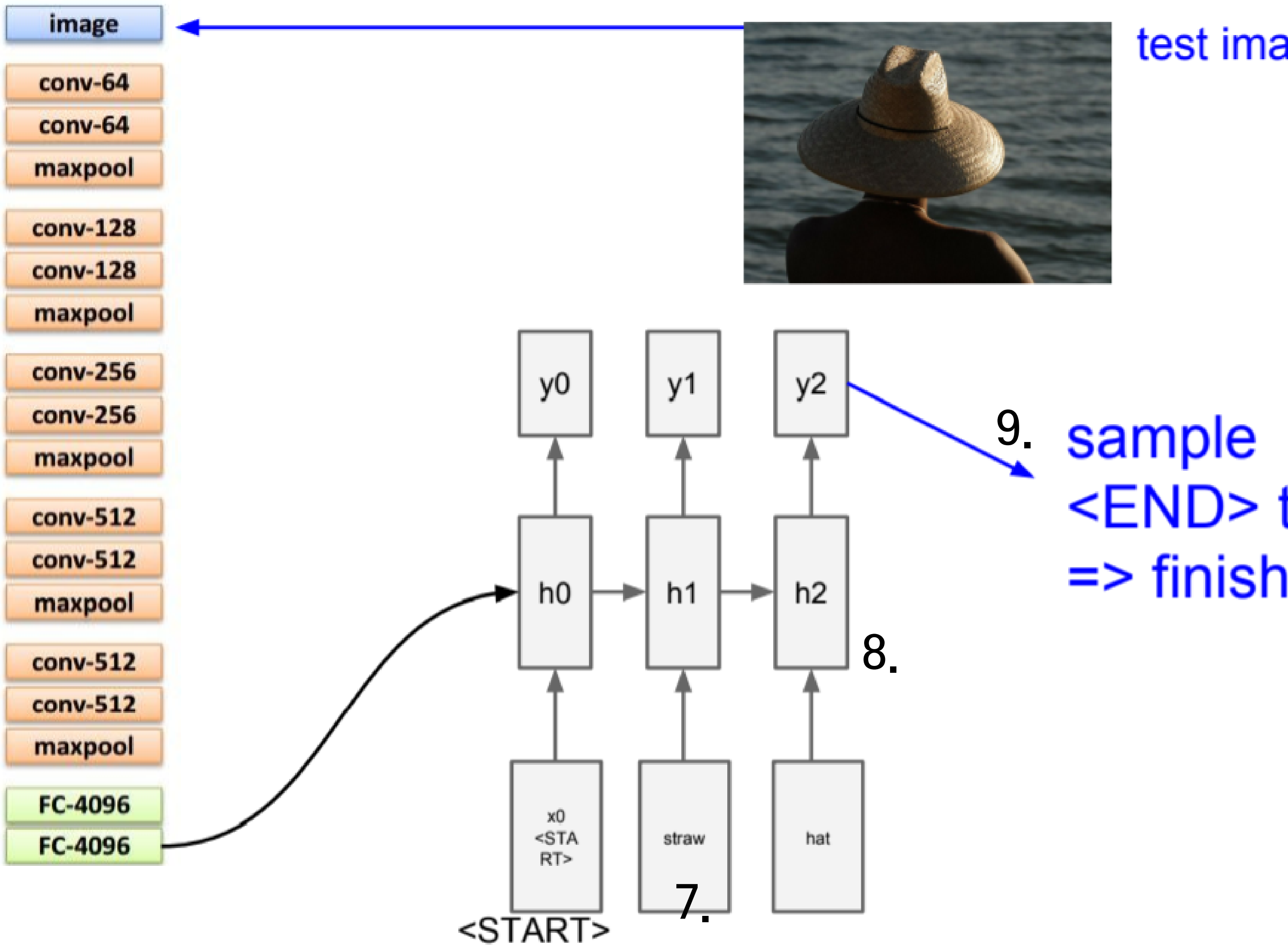
$h = \tanh(Wxh * x + Whh * h)$

now:

$h = \tanh(Wxh * x + Whh * h + \mathbf{Wih} * v)$

<방법>
1. 이미지를 input
2. CNN
마지막 FC는 사용X(=softmax score사용X)
직전의 4096-dim vector 사용
3. 초기값 h_0, x_0 입력
4. RNN 모델이 두개의 가중치 행렬을 입력으로 받았음 +하나 추가!
이미지 정보도 더해서 가중치 행렬 해준다!
5. 모든 score에 대한 분포 계산
6. 분포에서 sampling

4 Image captioning



test image

9. sample <END> token => finish

<방법>

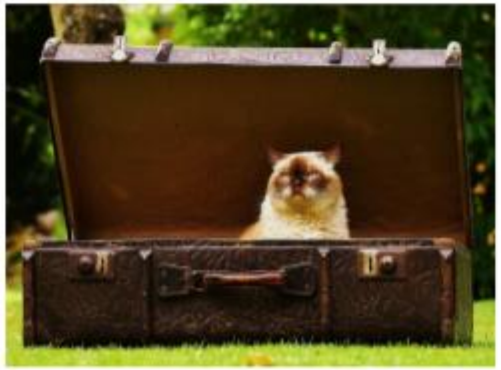
7. 그 단어를 다음 스텝의 입력으로 다시 넣어줌

8. 반복

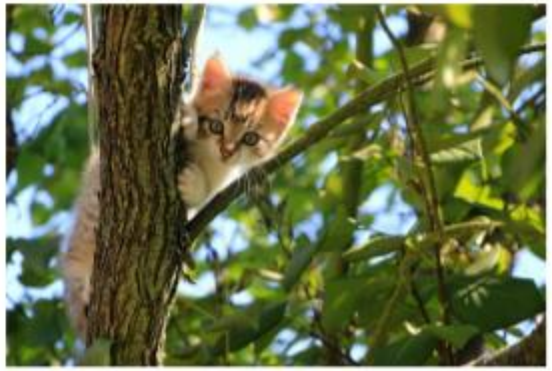
9. Train time에는 모든 captio의 종료지점에 END token 삽입.

->더 이상의 단어 생성 X 이미지에 대한 caption 완성

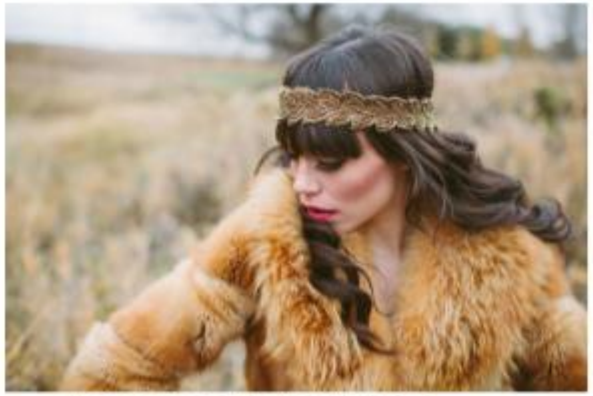
Example Results



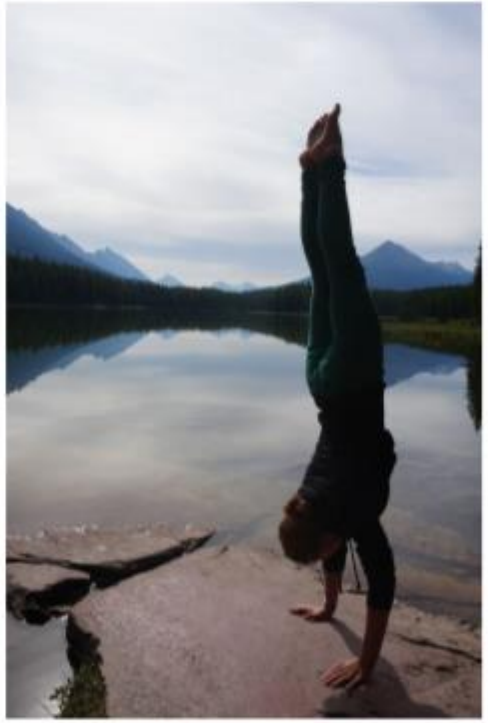
A cat sitting on a suitcase on the floor



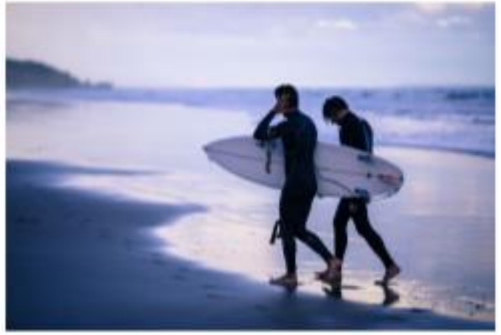
A cat is sitting on a tree branch



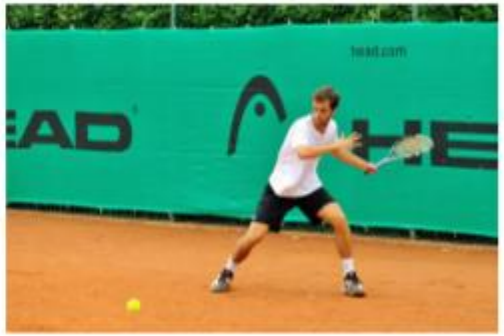
A woman is holding a cat in her hand



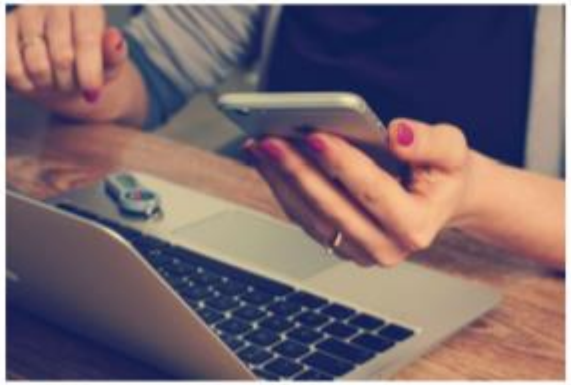
A woman standing on a beach holding a surfboard



Two people walking on the beach with surfboards



A tennis player in action on the court



A person holding a computer mouse on a desk

Image Captioning with Attention

Caption 생성할 때 이미지의 다양한 부분을 집중해서 볼 수 있다.

RNN focuses its attention at a different spatial location when generating each word

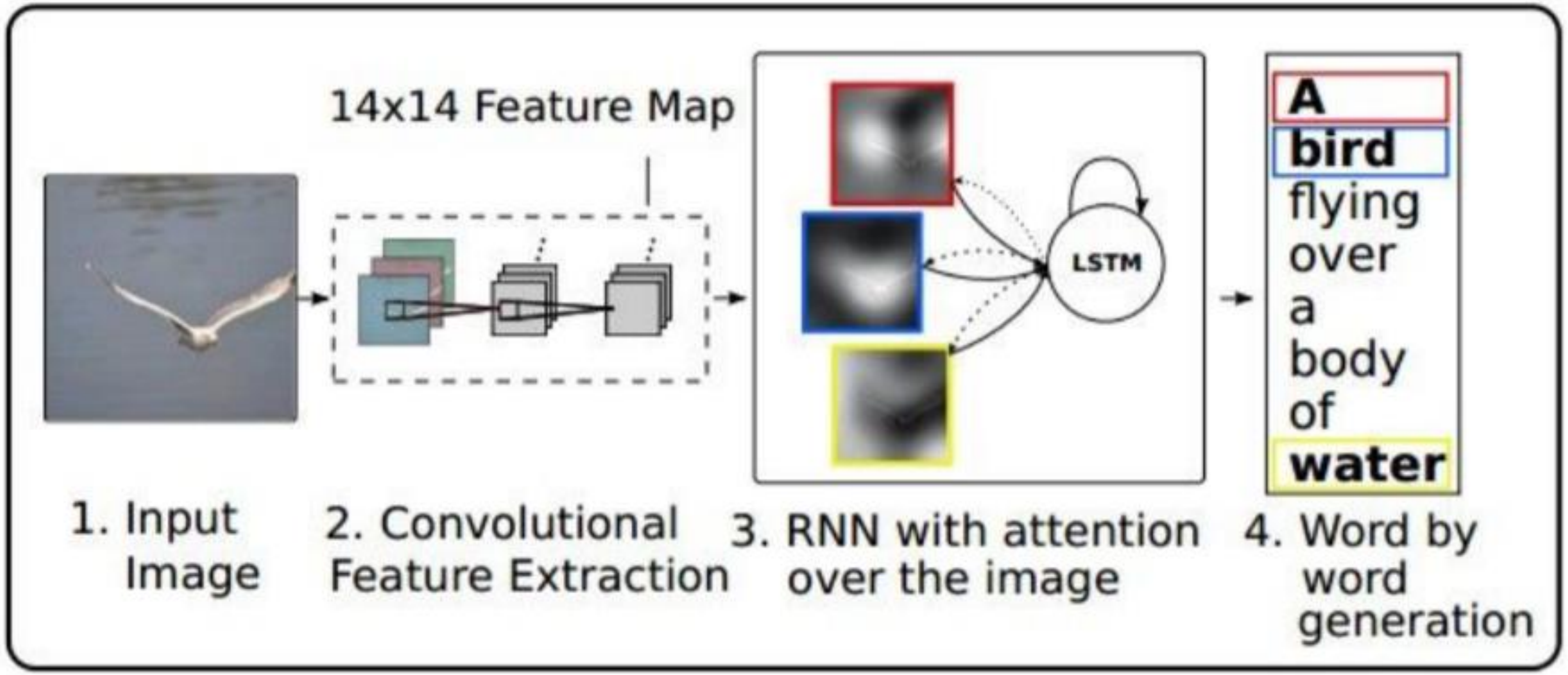
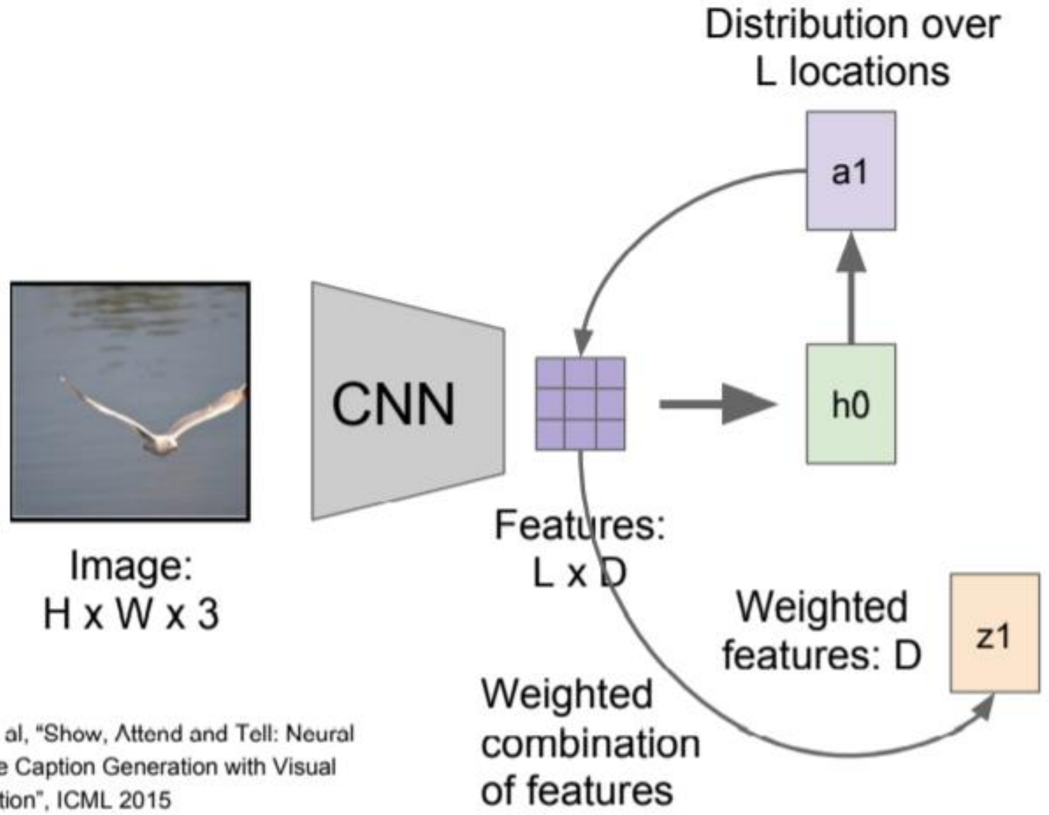

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Image Captioning with Attention


Distribution over L locations
a1
h0
CNN
Image: H x W x 3
Features: L x D
Weighted features: D
z1
Weighted combination of features

$$z = \sum_{i=1}^{L} p_i v_i$$

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

<방법>

1. CNN으로 공간정보를 가지고 있는 grid of vector를 만들어낸다.
2. h_0 는 이미지 위치에 대한 분포 계산
3. a_1 는 벡터 집합(L*D)와 연산하여 이미지 z_1을 생성한다.

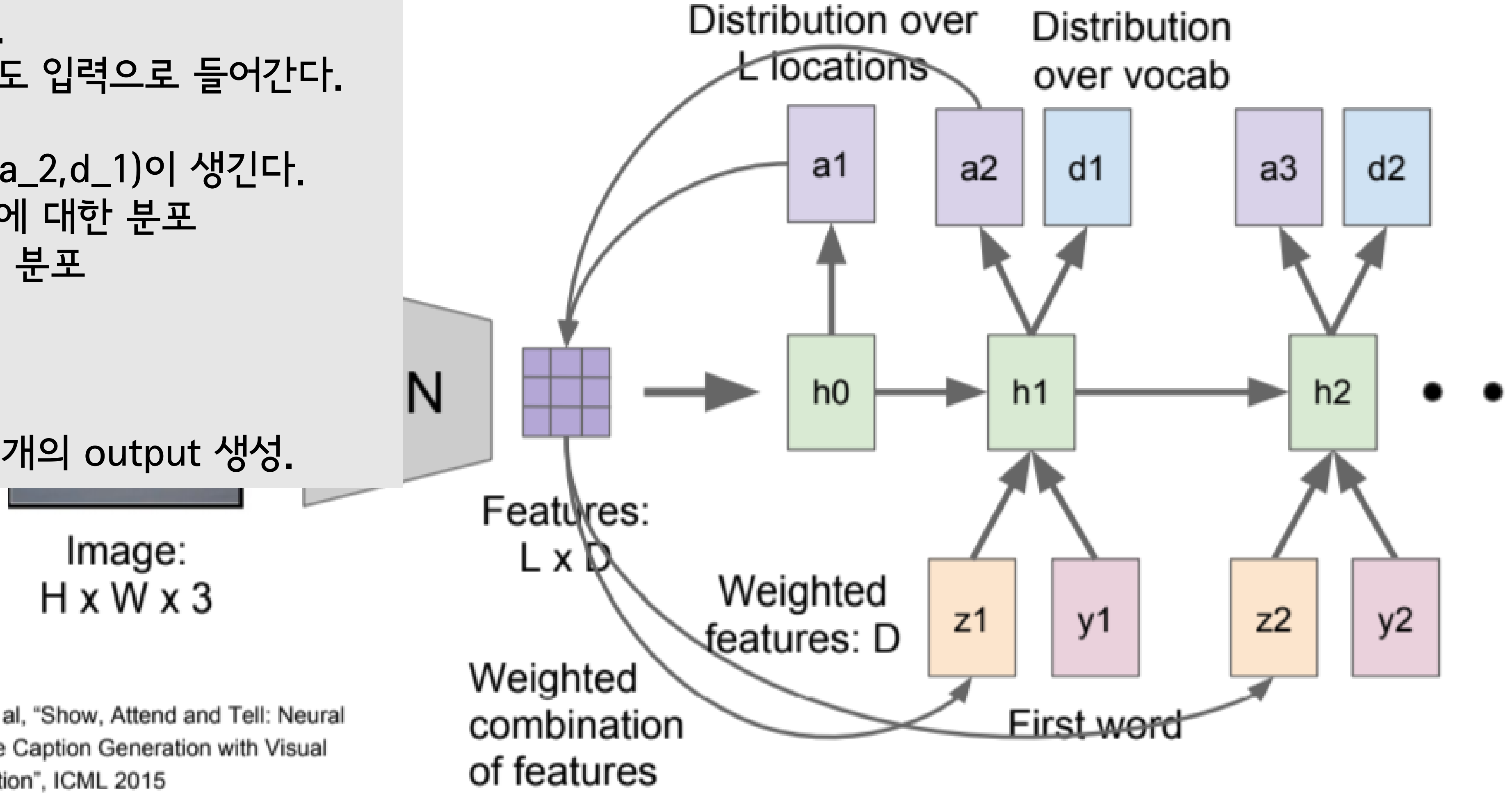
4 Image captioning

4. z_1은 Neural Network의 다음 스텝의 입력으로 들어간다.
첫번째 단어인 y_1도 입력으로 들어간다.

5. 두개의 output(a_2,d_1)이 생긴다.
a_2 = 이미지 위치에 대한 분포
d_1 = 각 단어들의 분포

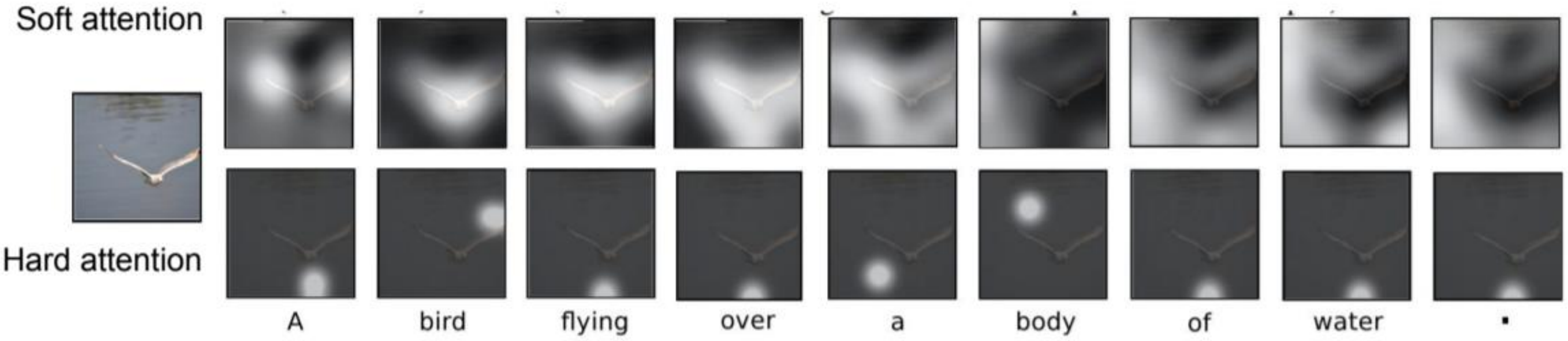
6. 반복

7. 매 스텝 마다 두개의 output 생성.



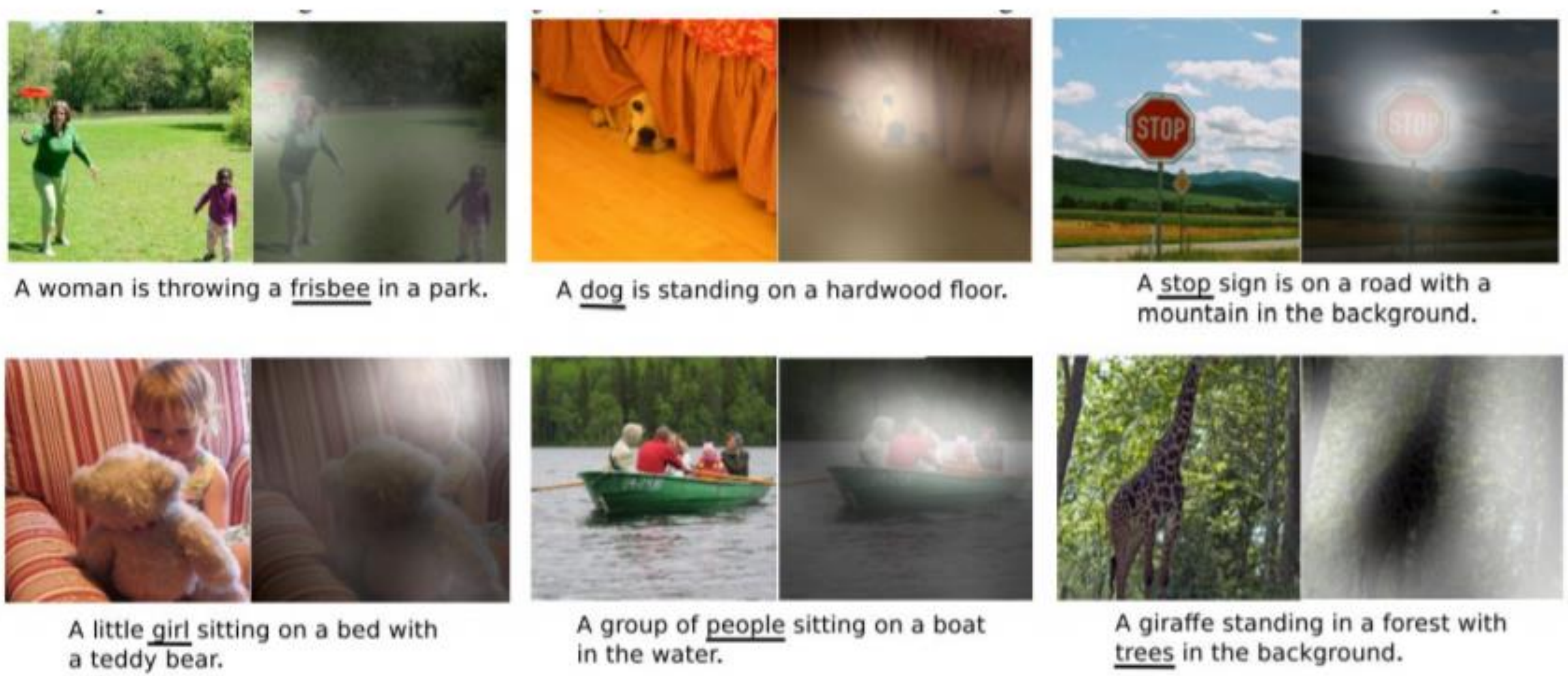
Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

4 Image captioning



→ Train 끝마치면, 모델이 caption 생성하기 위해서 이미지의 attention 이동시키는 모습을 볼 수 있다.

4 Image captioning



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

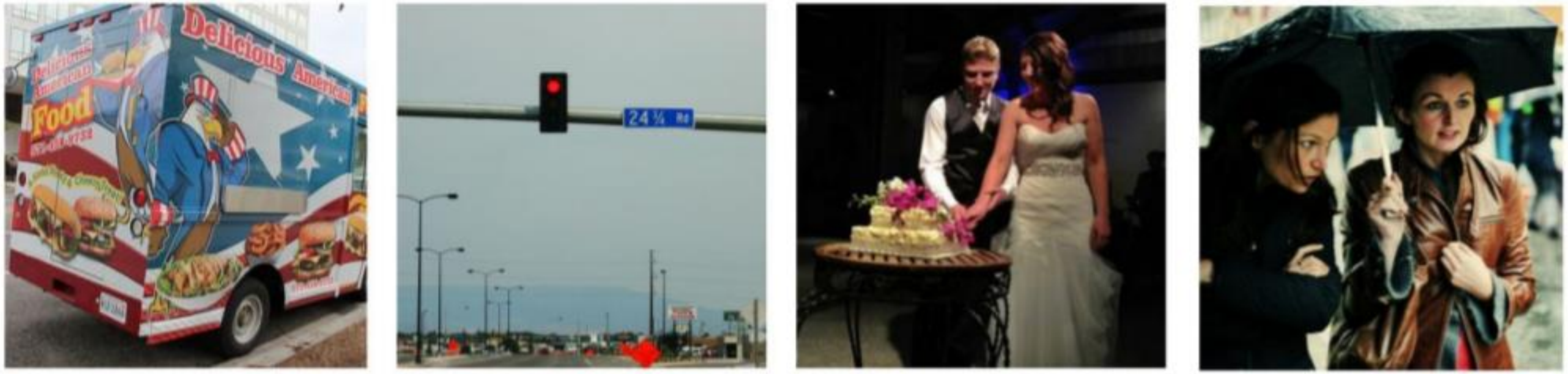
A giraffe standing in a forest with trees in the background.

→ Attention 모델 학습시키고 나서 caption 생성해보면 실제로 모델이 caption을 생성할 때 의미 있는 부분에 attention을 집중한다는 것을 알 수 있다.

Visual Question Answering

VQA = RNN + CNN

Input 두개 = 이미지 & 이미지 관련 질문(자연어 문장->”many to one”



Q: What endangered animal is featured on the truck?

A: A bald eagle.
A: A sparrow.
A: A humming bird.
A: A raven.

Q: Where will the driver go if turning right?

A: Onto 24 ¾ Rd.
A: Onto 25 ¾ Rd.
A: Onto 23 ¾ Rd.
A: Onto Main Street.

Q: When was the picture taken?

A: During a wedding.
A: During a bar mitzvah.
A: During a funeral.
A: During a Sunday church service

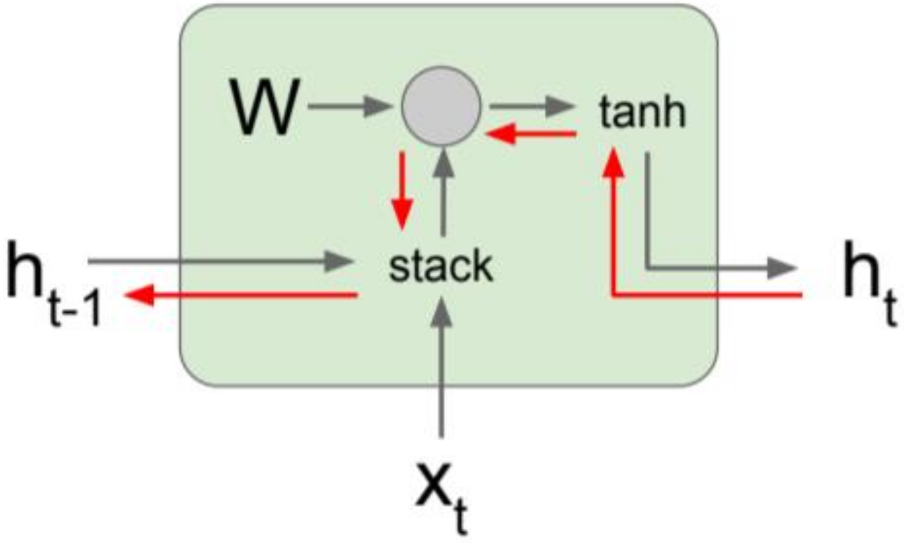
Q: Who is under the umbrella?

A: Two women.
A: A child.
A: An old man.
A: A husband and a wife.

Vanilla RNN Gradient Flow

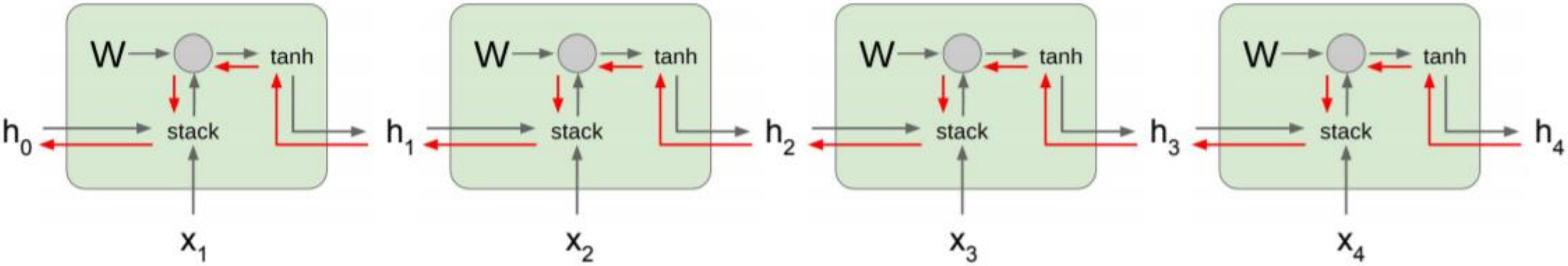
Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013

Backpropagation from h_t to h_{t-1} multiplies by W (actually W_{hh}^T)



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

5 LSTM



h_0 gradient 구하려면
모든 RNN Cells 거쳐야 한다. & Vanilla 통과할 때마다 각 cell의 W transpose을 곱하게 된다.

"Exploding gradient problem" (Largest singular value > 1)
back prop 시 레이어가 깊어질수록 gradient가 기하급수적으로 증가하는 현상
➔ Gradient clipping : gradient 의 L2 norm이 임계값보다 크면 최대 임계값을 넘지 못하도록 조정하는 것.
"Vanishing gradient problem" (Largest singular value < 1)
back prop 시 레이어가 깊어질수록 gradient가 기하급수적으로 감소하는 현상

➔ LSTM : RNN의 fancier 버전

Long Short Term Memory (LSTM)

Vanilla RNN

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

-매 스텝당 hidden state를 재귀적 방법으로 업데이트
-두 입력(h_t-1, x_t)를 concat하고 행렬 곱 연산하여
hidden state구하기

LSTM

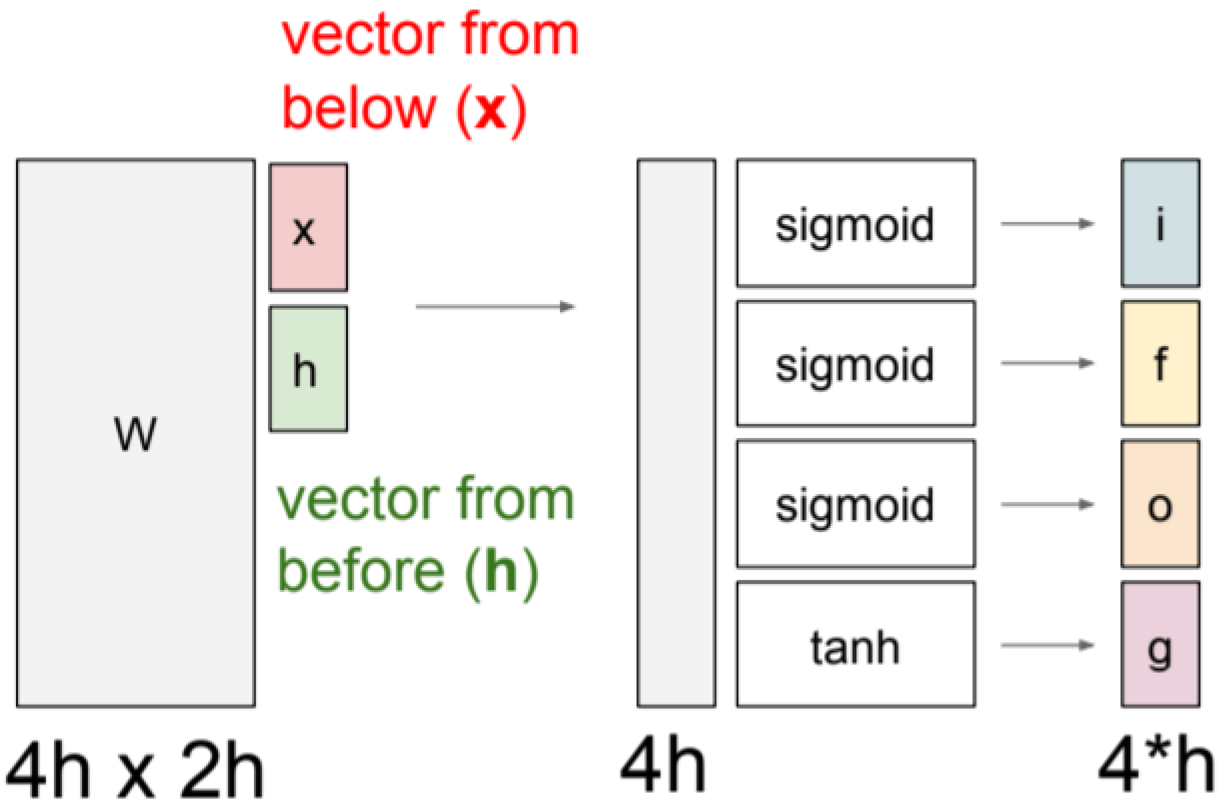
$$\begin{pmatrix}i\\f\\o\\g\end{pmatrix} = \begin{pmatrix}\sigma\\\sigma\\\sigma\\\tanh\end{pmatrix} W\begin{pmatrix}h_{t-1}\\x_t\end{pmatrix}$$

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

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

-한 cell 당 두개의 hidden state(h_t-1,c_t)

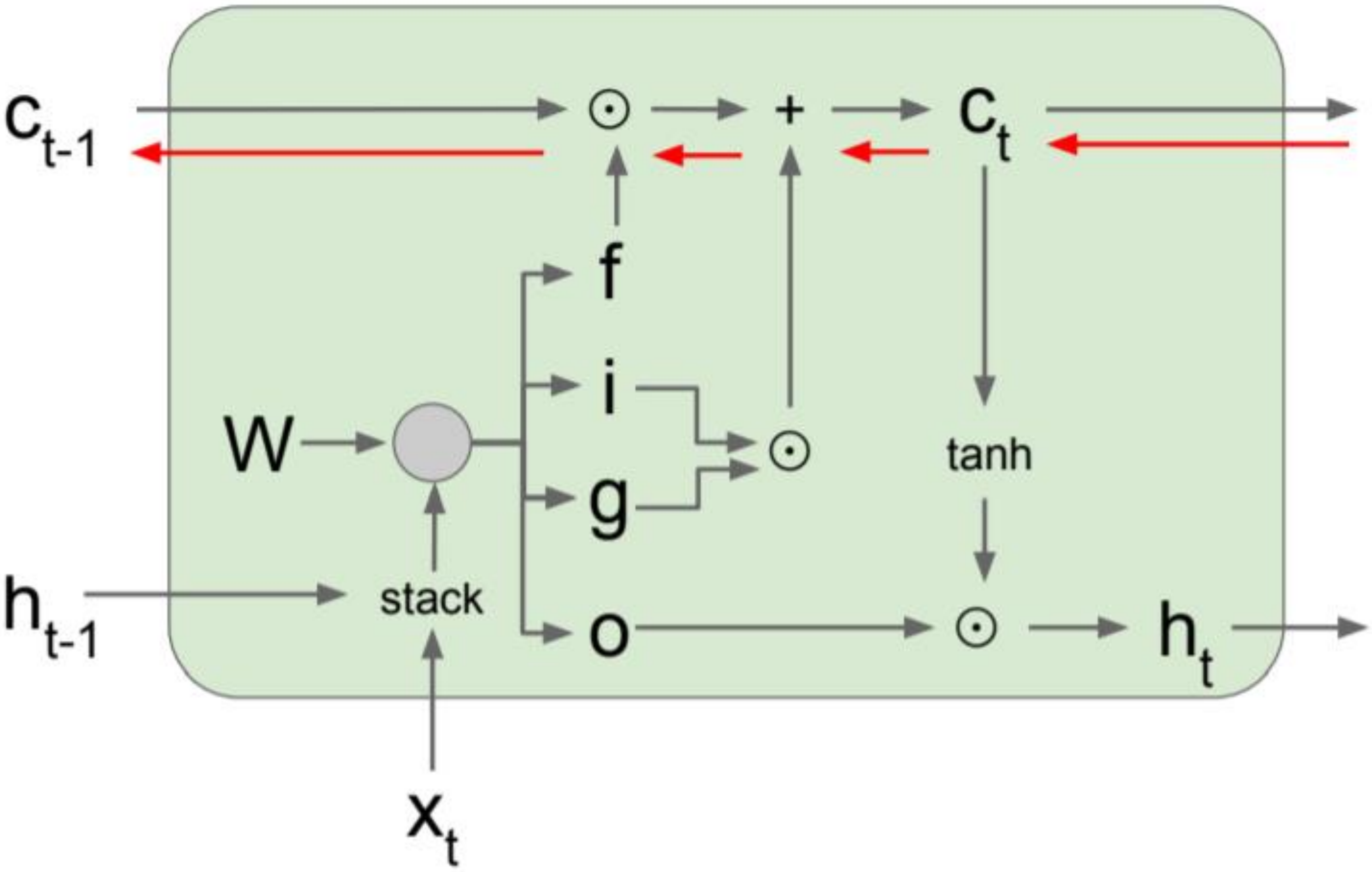
Long Short Term Memory (LSTM)



I = input gate cell에서의 입력에 대한 가중치
F = forget gate 이전 스텝의 cell 정보를 얼마나 forget 할지에 대한 가중치
O = Output gate cell state(c_t)를 얼마나 밖에 드러내 보일지에 대한 가중치
G = gate gate input cell을 얼마나 포함시킬지 결정하는 가중치

Sigmoid (0~1 사이값)
Tanh (-1~1 사이값)

5 LSTM



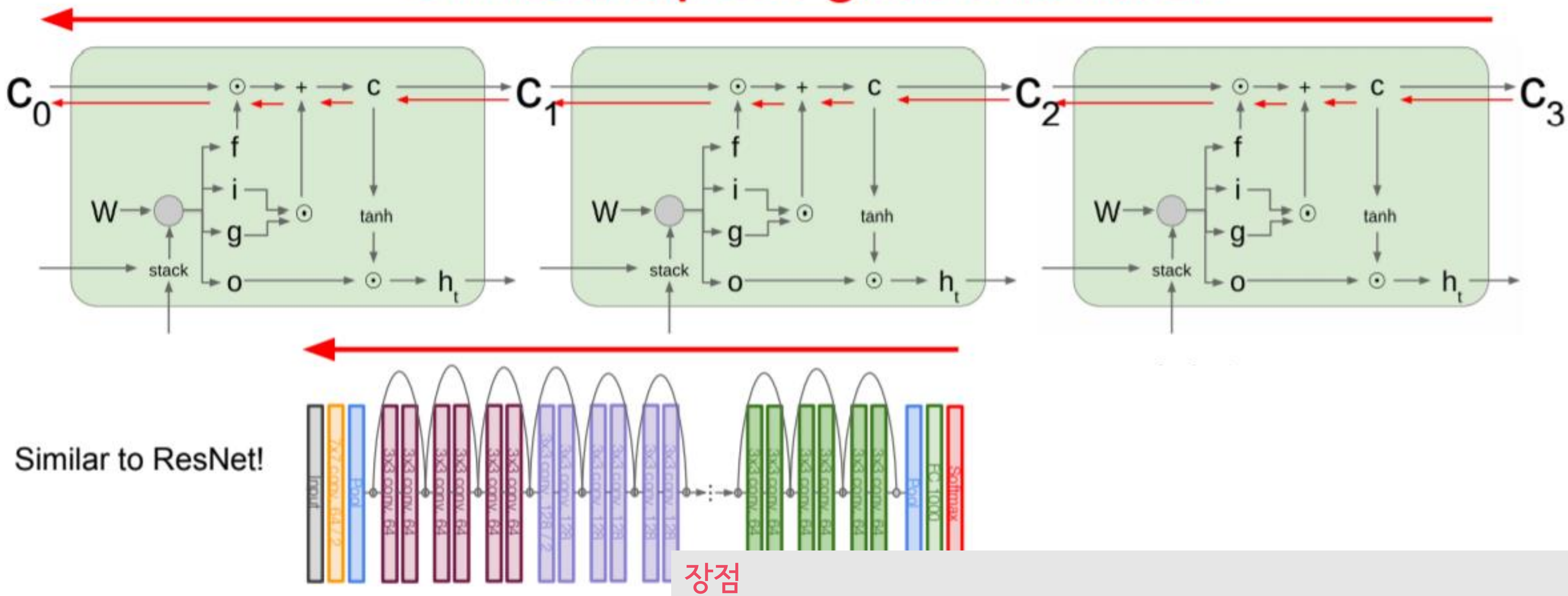
Backpropagation from c_t to c_{t-1} only elementwise multiplication by f, no matrix multiply by W

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

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

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

Uninterrupted gradient flow!



장점

1. Forget와 곱해지는 연산이 matix가 아니라element-wise 곱이다.
2. 매 스텝마다 다른 값의 forget gate와 곱해진다.
3. Backprop할 때 매 스텝마다 tanh 거치지않고 단 한번만 거친다.

➔Cell state를 사용하면 backprop은 gradient을 위한 고속도로!!

감사합니다!

발표 들어주셔서 감사합니다