ICT이노베이션스퀘어 AI복합교육 고급 언어과정

자연어처리를 위한 Language Model

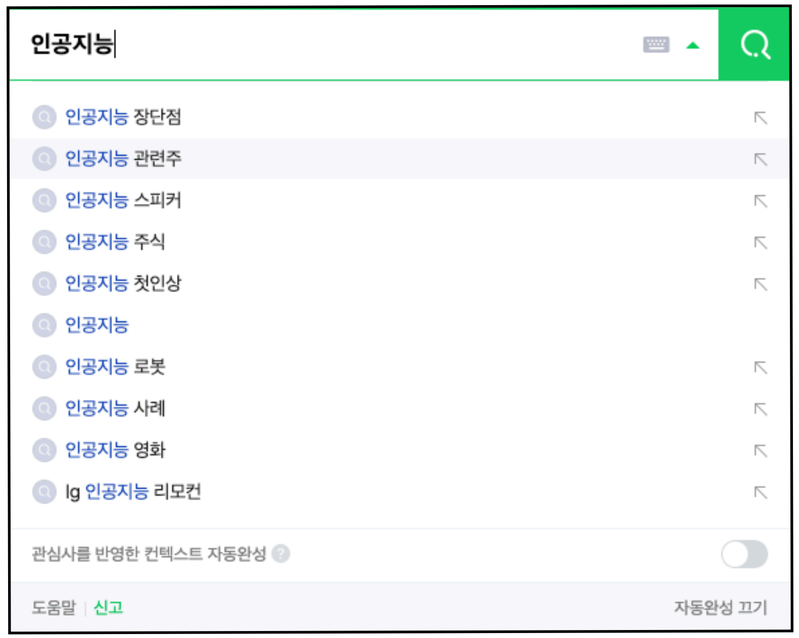
현청천

2021.04.19

What is Language Model

Language Model은 언어의 확률분포를 추정하는 것

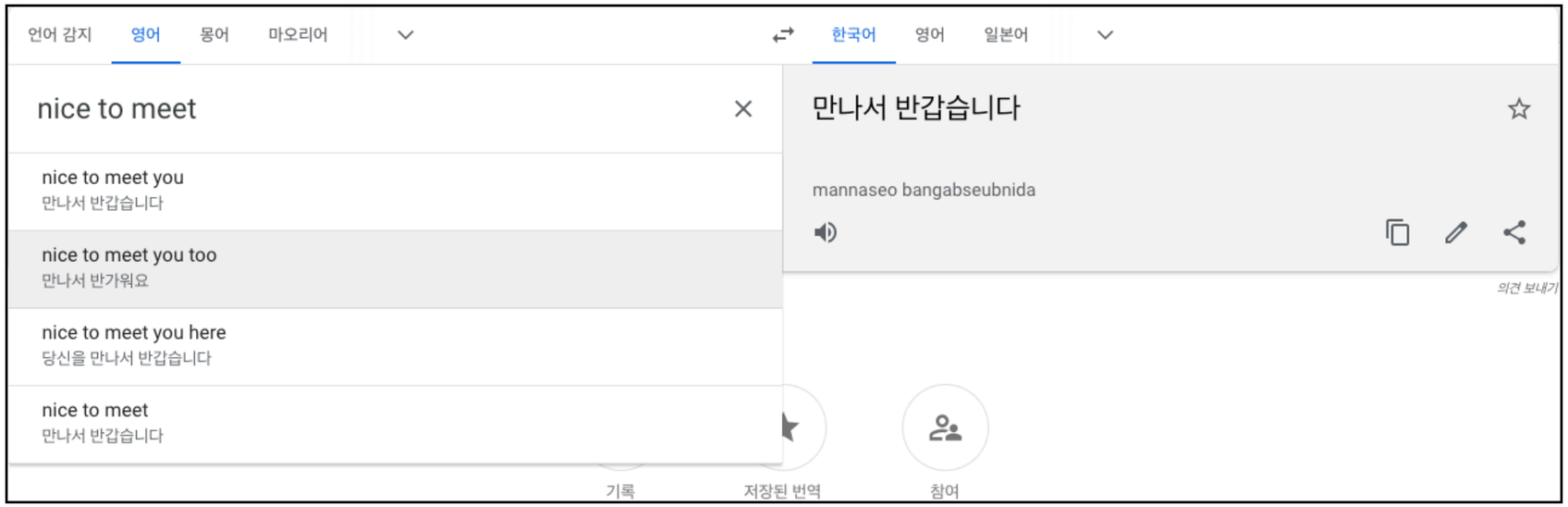
What is Language Model



○ 자동완성기능

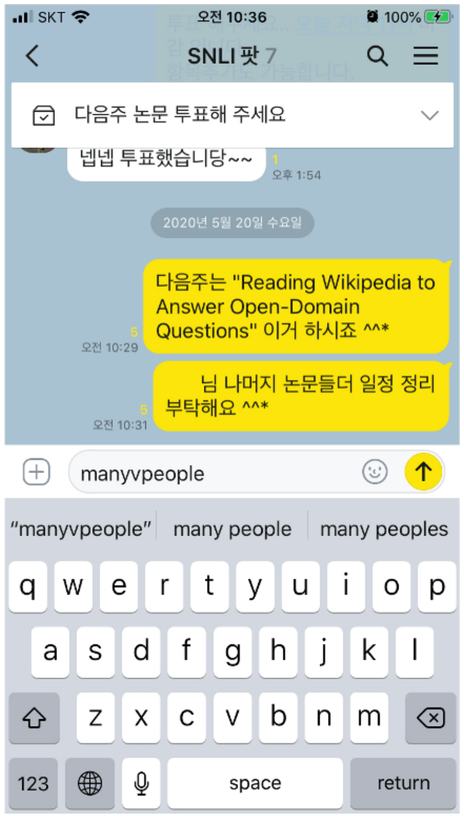
자동완성

What is Language Model



자동완성

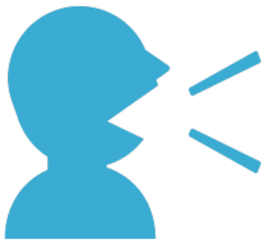
What is Language Model



오타

후보군

What is Language Model



아버지가방에들어가신다

아버지가 방에 들어가신다

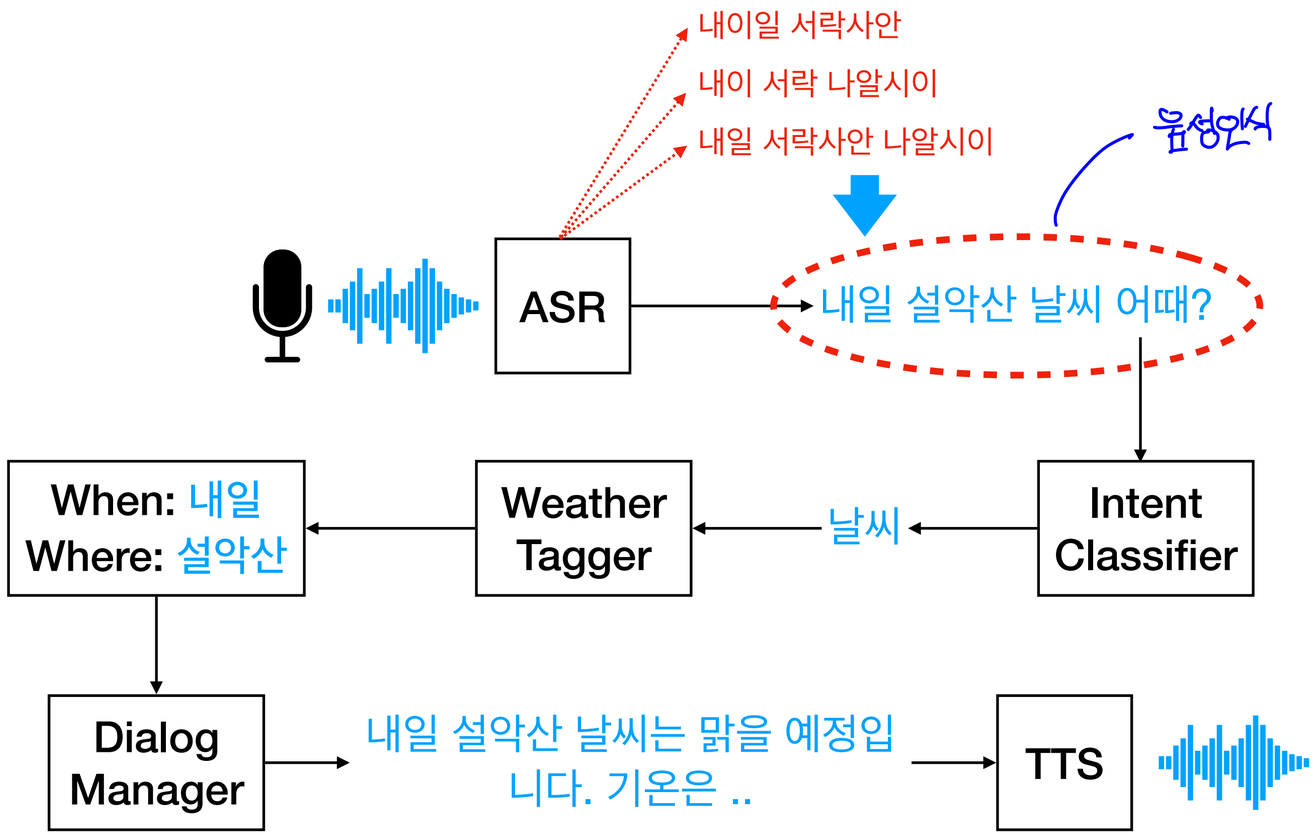
아버지 가방에 들어가신다

아버지가방 들어가신다

아버지가 방에 들어가신다

사람간의 대화

What is Language Model



음성인식에서 Language Model을 이용해 음성을 문자로 변환

What is Language Model

자연어에서 발생할 확률

p(그는 사과를 보자 배고픔을 느꼈다) > p(그는 사과를 보자 외로움을 느꼈다)

p(그녀는 운동을 열심히 한다) > p(그녀는 운동을 떳떳이 한다)

확률된 확률표를 사용하는 것.

What is Language Model

실제 언어의 확률분포를 아는 것은 어려움

좋은 근사치를 제공하는 Language Model을 정의 할 수 있음

만드는게 목적이겠지.

N-gram Language Model

48봉지 2620개의 M&M의 컬러 분포

372 + 544 + 369 + 483 + 481 + 371 = N

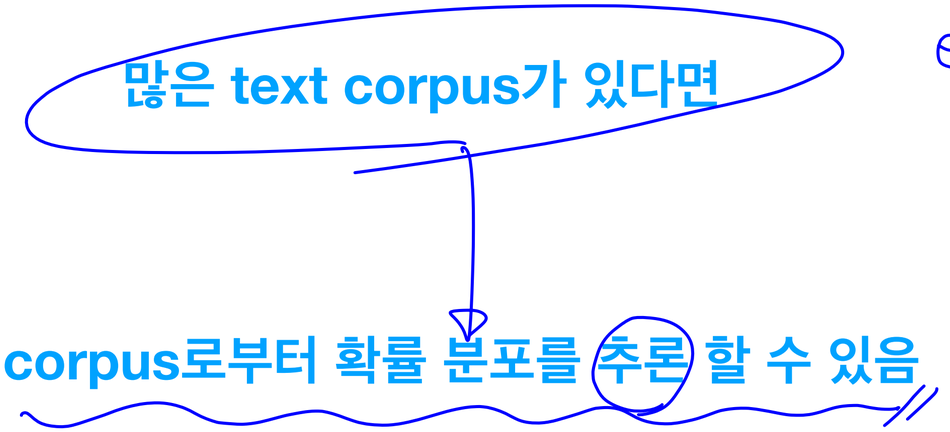
이 데이터로부터 확률 분포를 추론하는 방법은?

$$p(color) = \frac{count(color)}{N}, \quad N = \sum_{color} count(color)$$

N-gram Language Model

Word sequence로부터 확률 분포를 추론하는 방법

$$s = (w^{(1)}, w^{(2)}, \ldots, w^{(n)})$$ 이 단어들의 순서의 배열 = 문장



ex) 법률, 화학.. 각각에는 bias가 있어..

N-gram Language Model

$p(s = w^{(1)}, w^{(2)}, \ldots, w^{(n)})$

$p(s = the\ cat\ slept\ quietly)$

$p(w^{(1)} = the, w^{(2)} = cat, w^{(3)} = slept, w^{(4)} = quietly)$

$p(quietly \mid the\ cat\ slept) \cdot p(slept \mid the\ cat) \cdot p(cat \mid the) \cdot p(the)$

$$p(w^{(1)}, w^{(2)}, \ldots, w^{(n)}) = \prod_{i=1}^{n} p(w^{(i)} \mid w^{(1)}, \ldots, w^{(i-1)})$$

the i=1

cat i=2

slept i=3

w^1, w^2

N-gram Language Model

너무 어려우니까 전체를 보는건

몇개만 보자!

Independent Assumption

단어의 분포는 고정된 몇 개의 이전 단어에 의존함

너무

$$p(w^{(i)} \mid w^{(i)}, w^{(2)}, \ldots, w^{(i-1)}) \dashrightarrow p(w^{(i)} \mid w^{(i-n+1)}, w^{(i-n+2)}, \ldots, w^{(i-1)})$$

n=3 i-3+1 = i-2 까지

Trigram: $p(w^{(i)} \mid w^{(1)}, w^{(2)}, \ldots, w^{(i-1)}) \approx p(w^{(i)} \mid w^{(i-2)}, w^{(i-1)})$

n=2 i-2+1 = i-1 까지

bigram: $p(w^{(i)} \mid w^{(1)}, w^{(2)}, \ldots, w^{(i-1)}) \approx p(w^{(i)} \mid w^{(i-1)})$

unigram: $p(w^{(i)} \mid w^{(1)}, w^{(2)}, \ldots, w^{(i-1)}) \approx p(w^{(i)})$ n=1

N-gram Language Model

$$p(w^{(i)} \mid w^{(1)}, w^{(2)}, \ldots, w^{(i-1)}) \dashrightarrow p(w^{(i)} \mid w^{(i-n+1)}, w^{(i-n+2)}, \ldots, w^{(i-1)})$$

$$= \frac{p(w^{(i-n+1)}, w^{(i-n+2)}, \ldots, w^{(i-1)}, w^{(i)})}{p(w^{(i-n+1)}, w^{(i-n+2)}, \ldots, w^{(i-1)})}$$

Q 독립시행이니까?

N-gram과 (N-1)-gram의 확률 분포를 어떻게 구할 것인가?

➡ **큰 text corpus에서 개수를 세면 분포를 구할 수 있음**

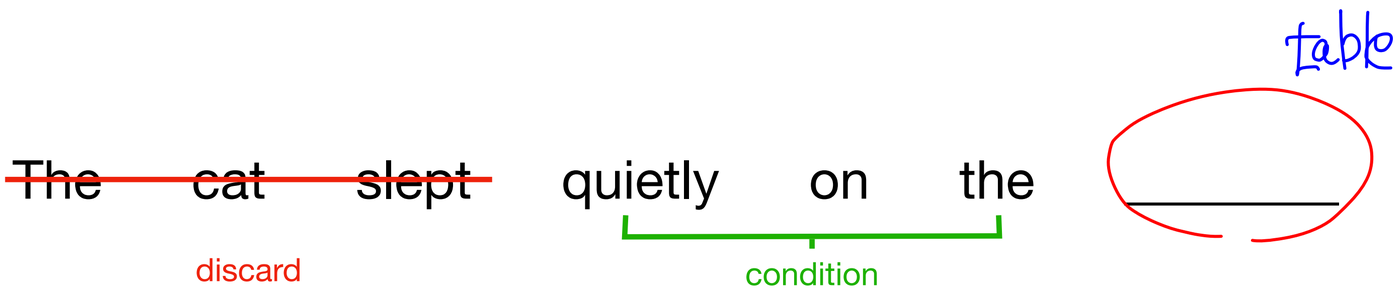
$$\frac{count(w^{(i-n+1)}, w^{(i-n+2)}, \ldots, w^{(i-1)}, w^{(i)})}{count(w^{(i-n+1)}, w^{(i-n+2)}, \ldots, w^{(i-1)})}$$

(Statistical approximation)

n개의 단어배열이 발생한 횟수

단어 배열 발생 횟수.

N-gram Language Model



$$p(w^{(i)} \mid The\ cat\ slept\ quietly\ on\ the) \approx \frac{count(quietly\ on\ the\ w^{(i)})}{count(quietly\ on\ the)}$$

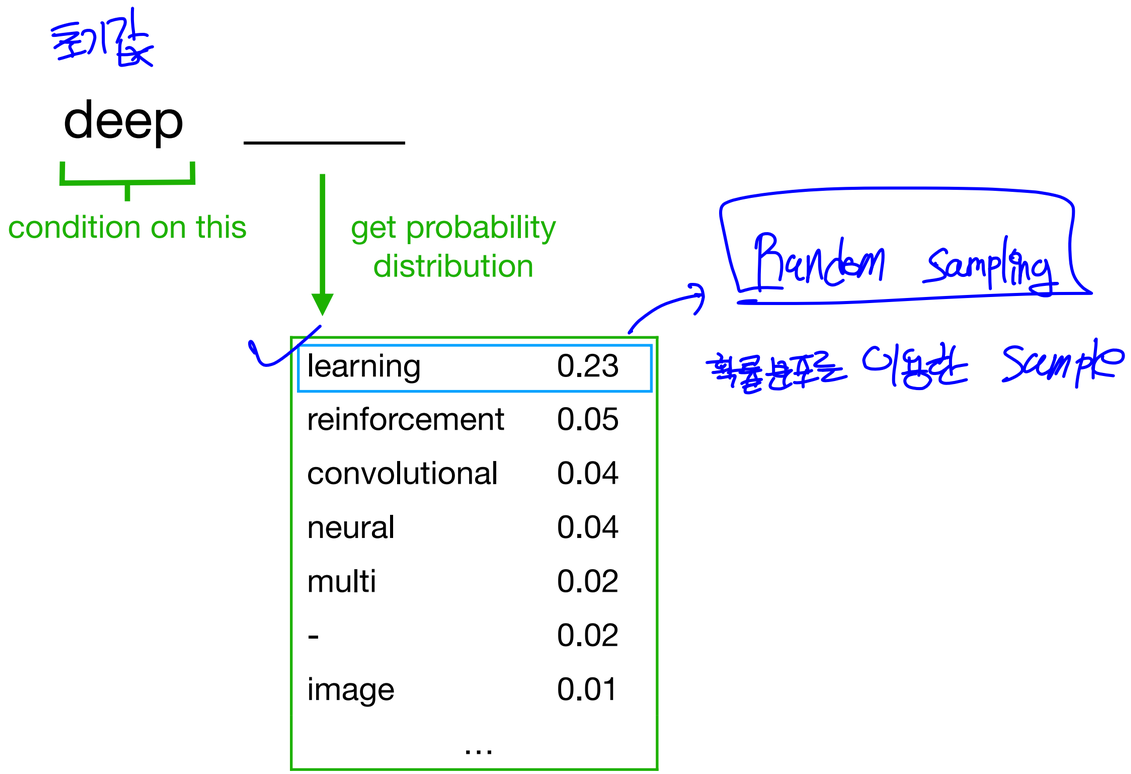
table의 개수 = 36 / 100

ex) 아름다운 — 꽃무궁화 = 20/100 %

— 나리 = 10/100 %

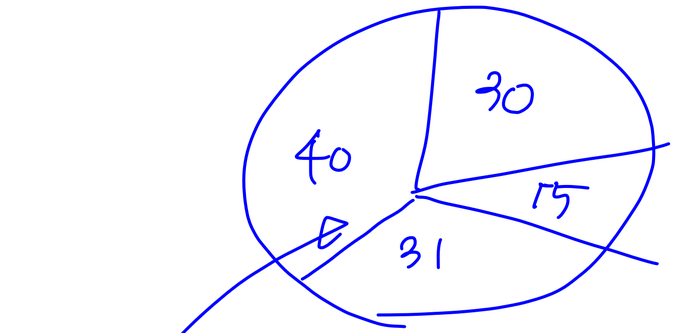
지금까지 count 기반

N-gram Language Model (Text Generation 3-gram)



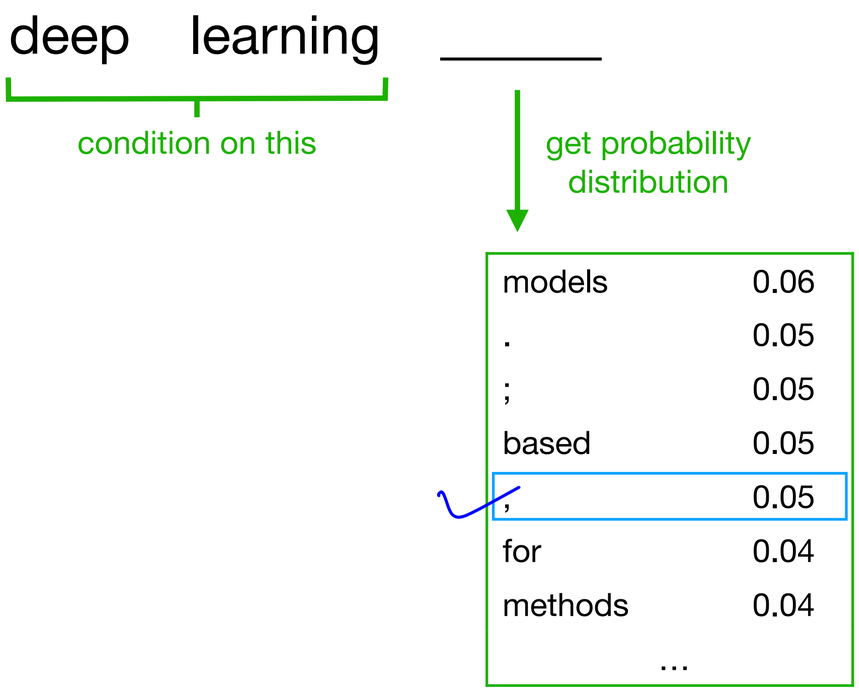
Random sampling

확률분포를 이용한 Sample

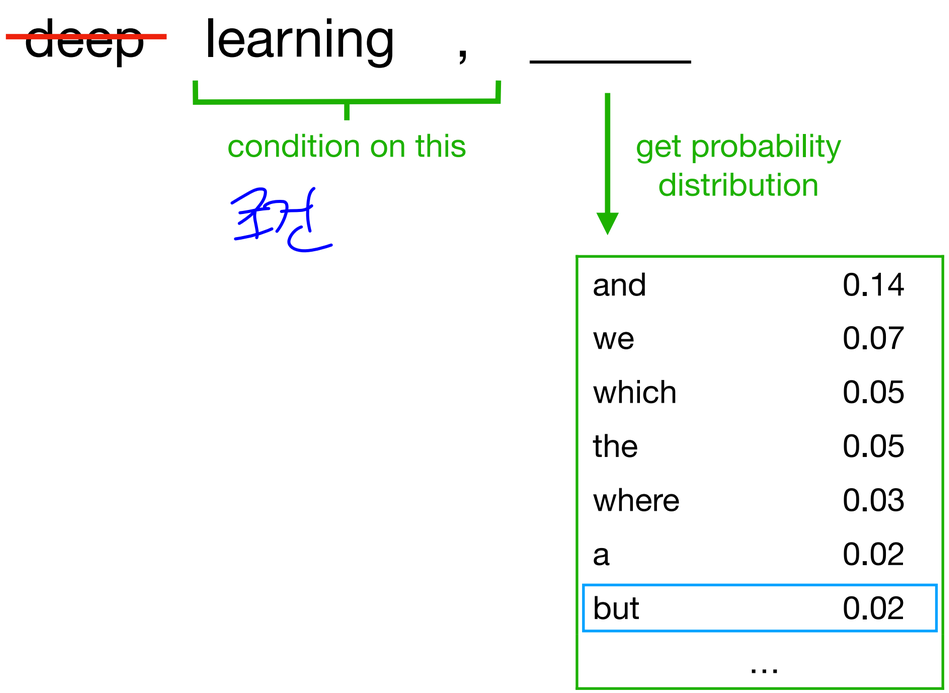


best만 뽑는게 아니라
그중에 확률로 뽑는거지.

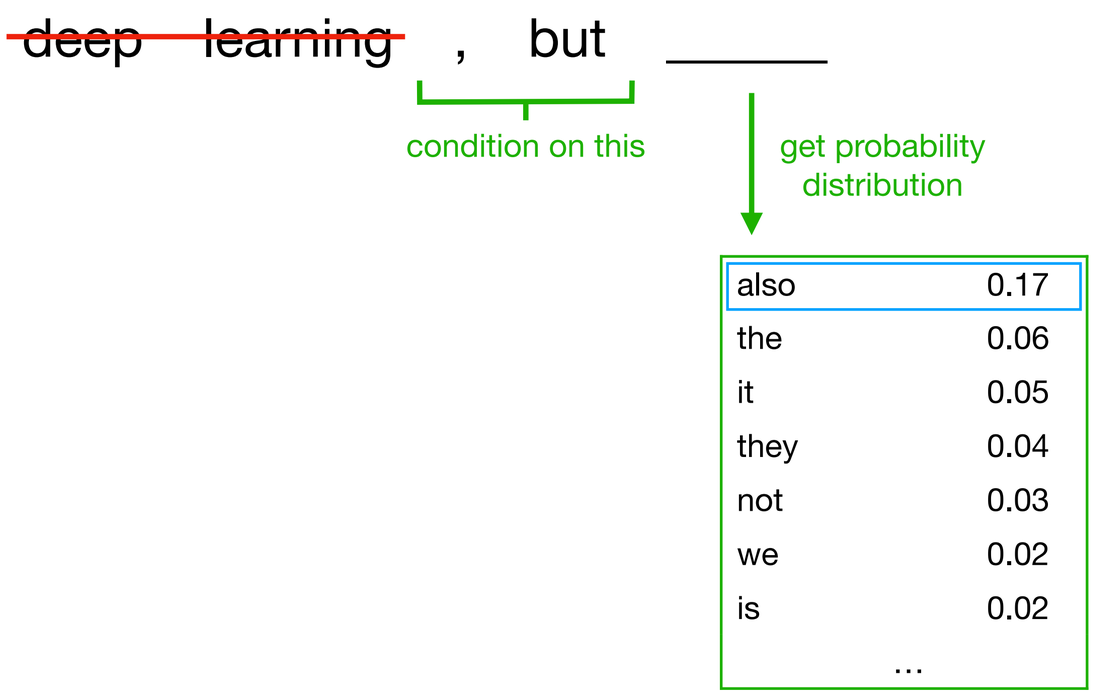
N-gram Language Model (Text Generation 3-gram)



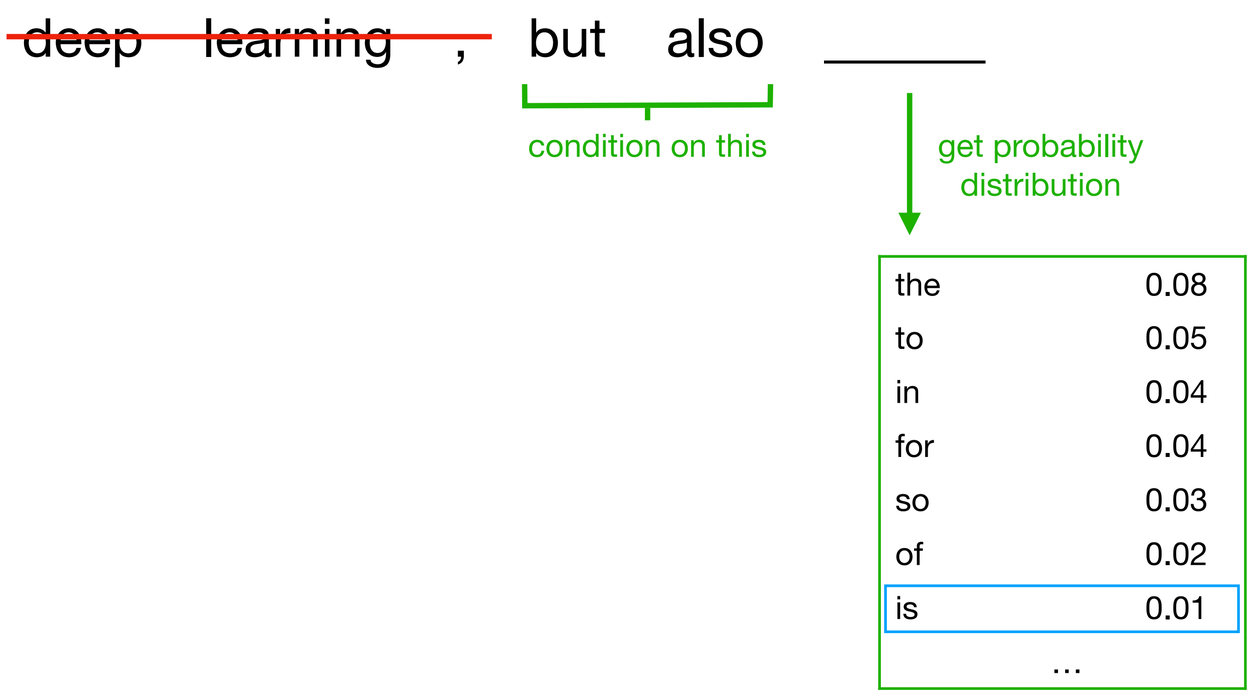
N-gram Language Model (Text Generation 3-gram)



N-gram Language Model (Text Generation 3-gram)

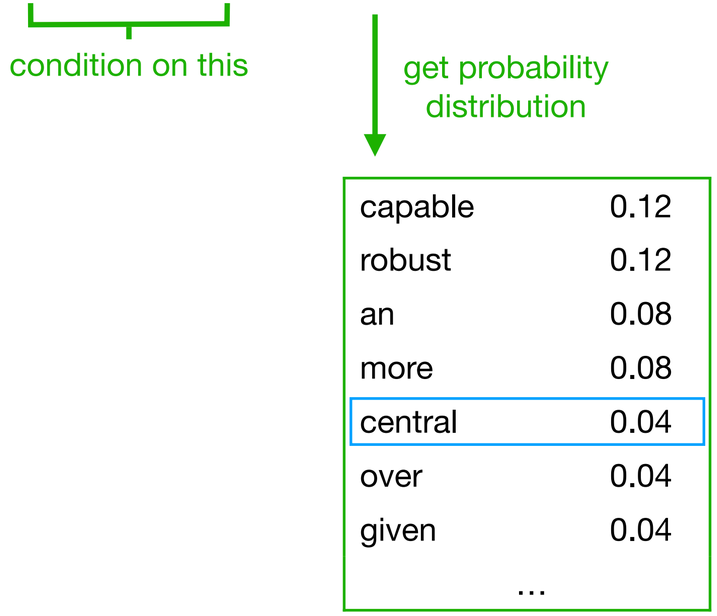


N-gram Language Model (Text Generation 3-gram)



N-gram Language Model (Text Generation 3-gram)

~~deep learning , but~~ also is _____



N-gram Language Model (Text Generation 3-gram)

deep learning , but also is central to human. performance . however , using structural similarity index measure than other partitioned sampling schemes , while making the approach with empirical data has the effect of phonetics has received little attention within the context of information on …

내용의 일관성이 전혀 없음

Neural Language Model (Fixed Window)

Output distribution

$$\hat{y} = softmax(Uh + b_2) \in \mathbb{R}^{|V|}$$

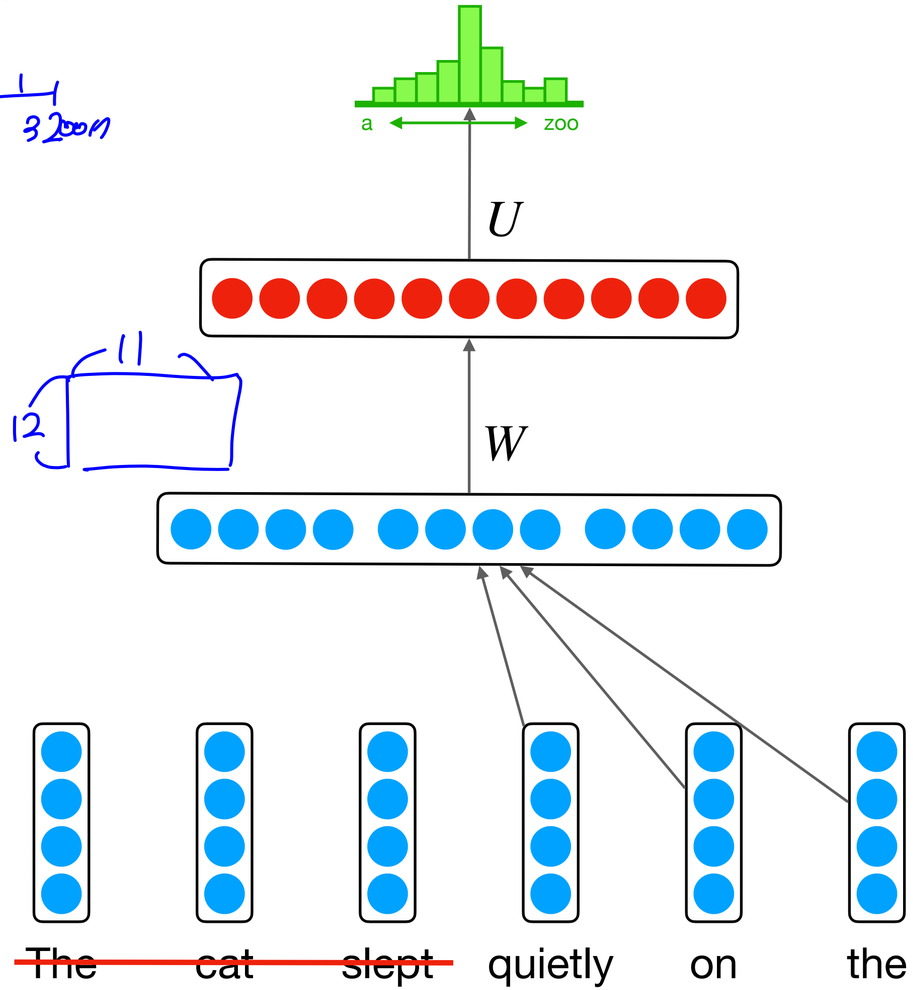
Hidden layer

$$h = f(Wx + b_1)$$

Concatenate word Embedding

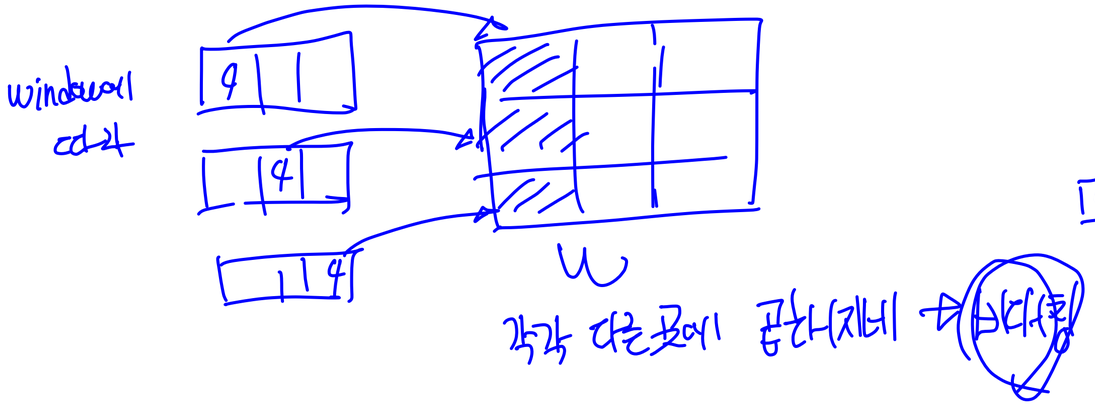
$$x = (x^{(i-3)}; x^{(i-2)}; x^{(i-1)})$$

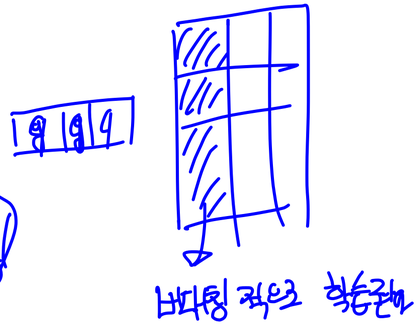
Word Embedding

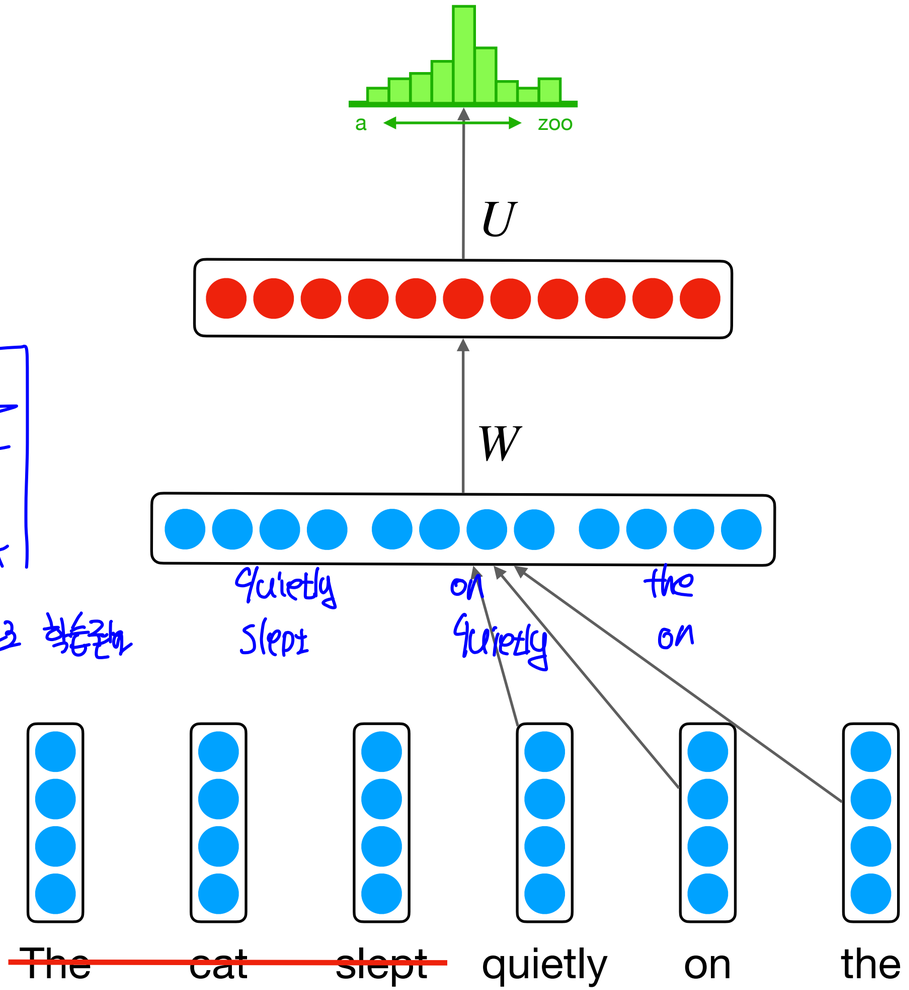


Neural Language Model (Fixed Window)

- 고정된 Window는 자연어를 처리하는데 크기가 부족함
- $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$은 window 위치에 따라 다른 weight를 사용 함 (비 대칭)







길이에 상관없이 처리 가능한 Neural Network가 필요 함

Neural Language Model (RNN)

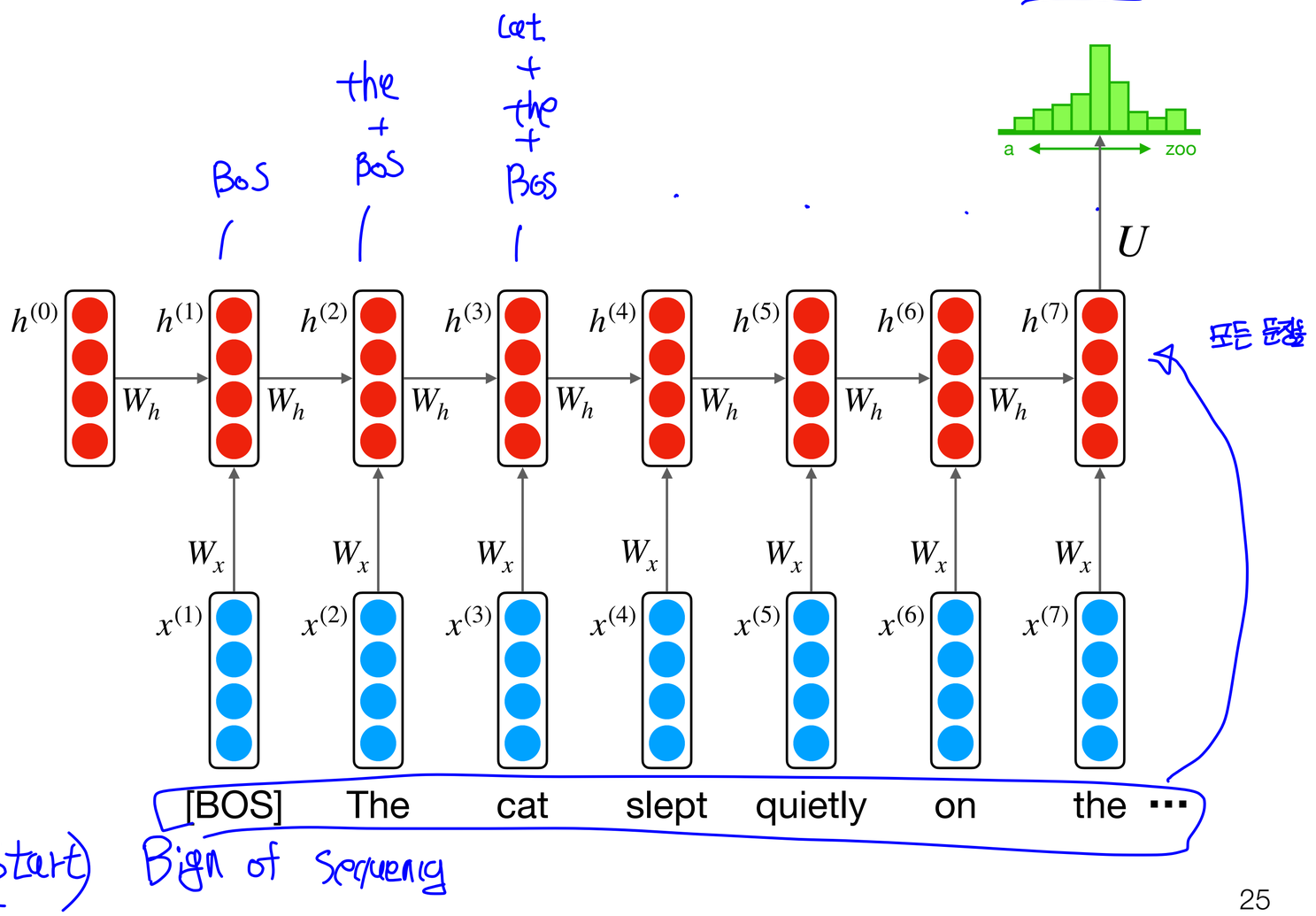
Output distribution

$$\hat{y} = softmax(Uh + b_2) \in \mathbb{R}^{|V|}$$

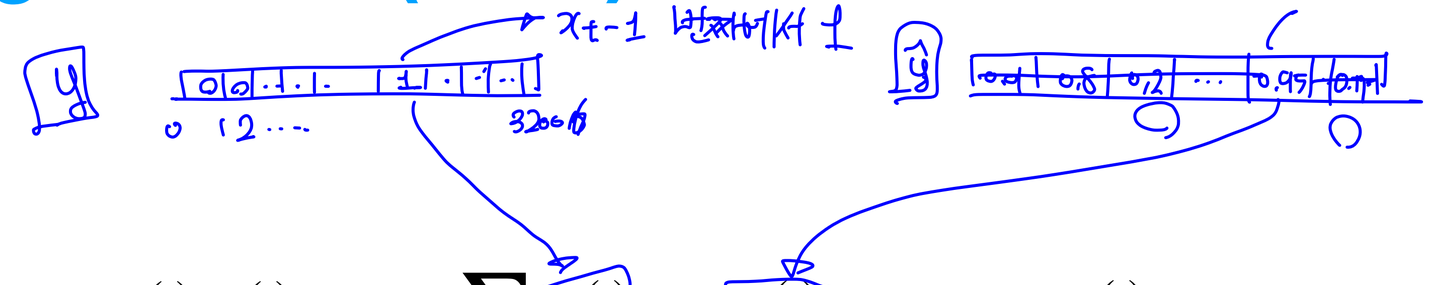
Hidden state

$$h^{(t)} = tanh(W_h h^{(t-1)} + W_x x^{(t)} + b_1)$$

Word Embedding



Neural Language Model (RNN)



$$J^{(t)}(\theta) = CE(y^{(t)}, \hat{y}^{(t)}) = -\sum_{w \in V} y_w^{(t)} \log(\hat{y}_w^{(t)}) = -\log \hat{y}_{x_{t+1}}^{(t)}$$

$\hat{y}_{x_{t+1}}^{(t)}$ ↑ 다음 단어를 예측한 확률을 크게하게 학습이다.

$\propto \lg \hat{y}_{x_{t+1}}^{(t)}$ ↑

$\frac{1}{\propto} -\lg \hat{y}_{x_{t+1}}^{(t)}$ ↓

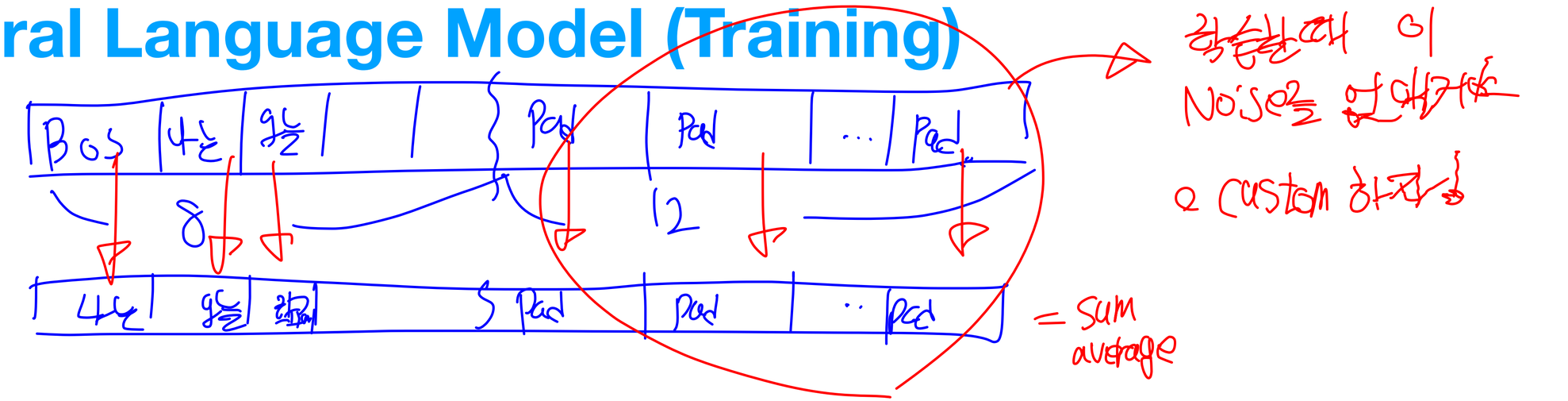
$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} -\log \hat{y}_{x_{t+1}}^{(t)}$$

negative log likelihood를 줄여준다.

= MLE

= Negative log likelihood

Neural Language Model (Training)

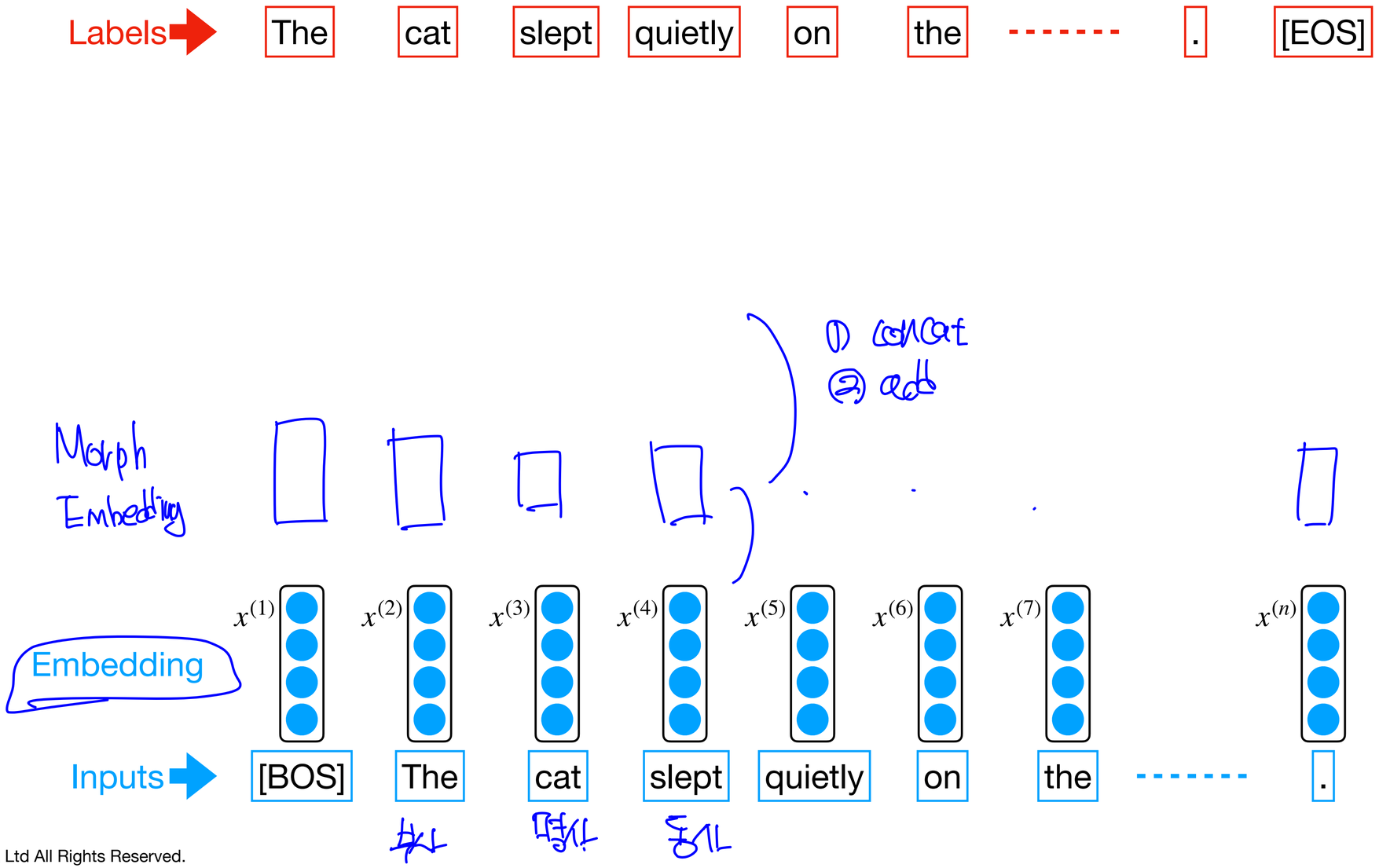


End of sequence

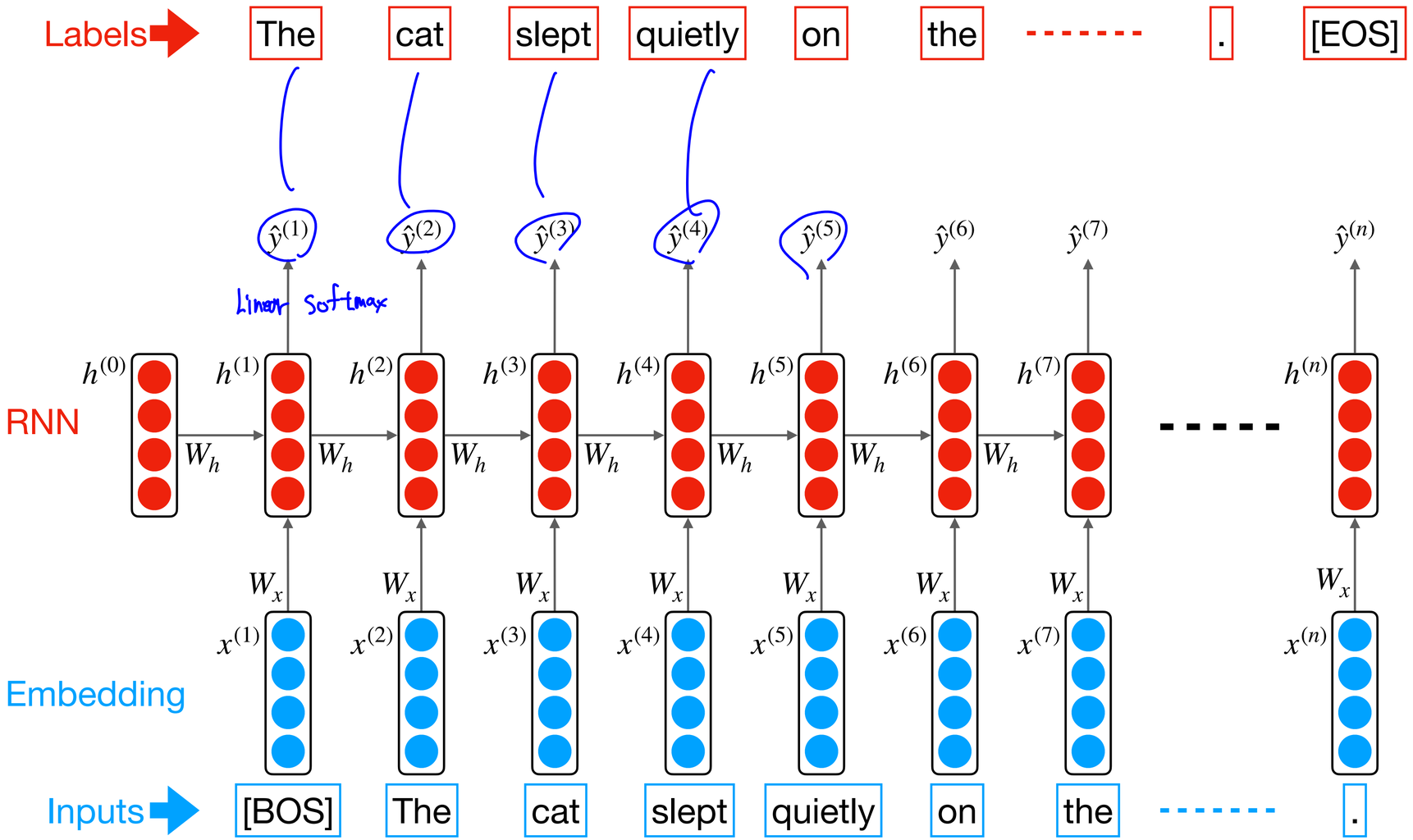
Labels →	The	cat	slept	quietly	on	the			.	[EOS]
Inputs →	[BOS]	The	cat	slept	quietly	on	the			.

Neural Language Model (Training)

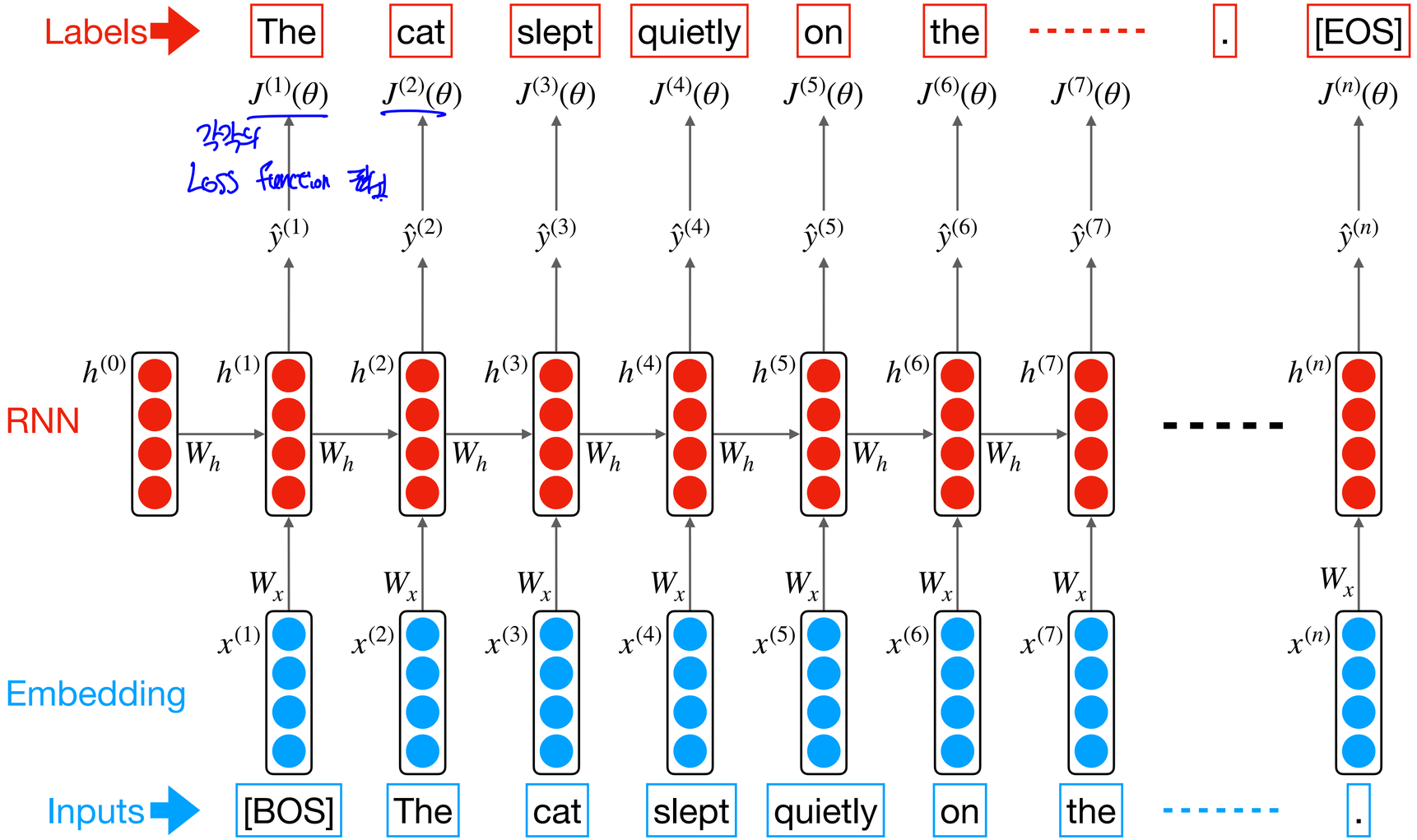
Neural Language Model (Training)



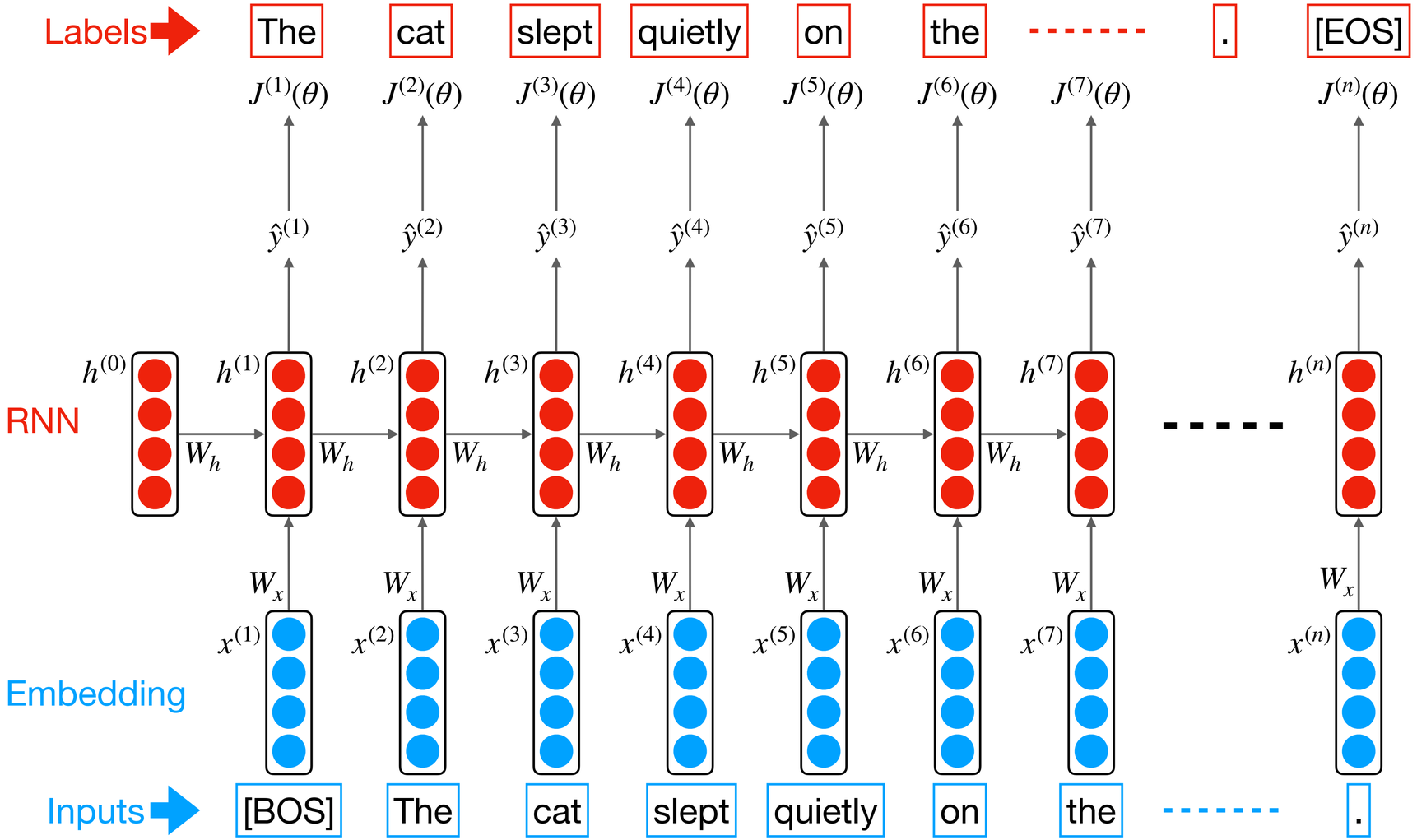
Neural Language Model (Training)



Neural Language Model (Training)



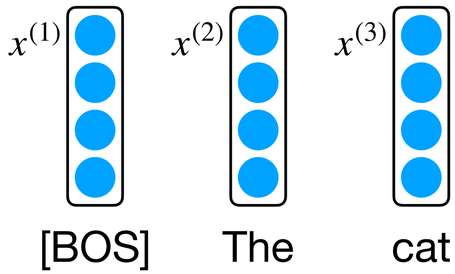
Neural Language Model (Training)



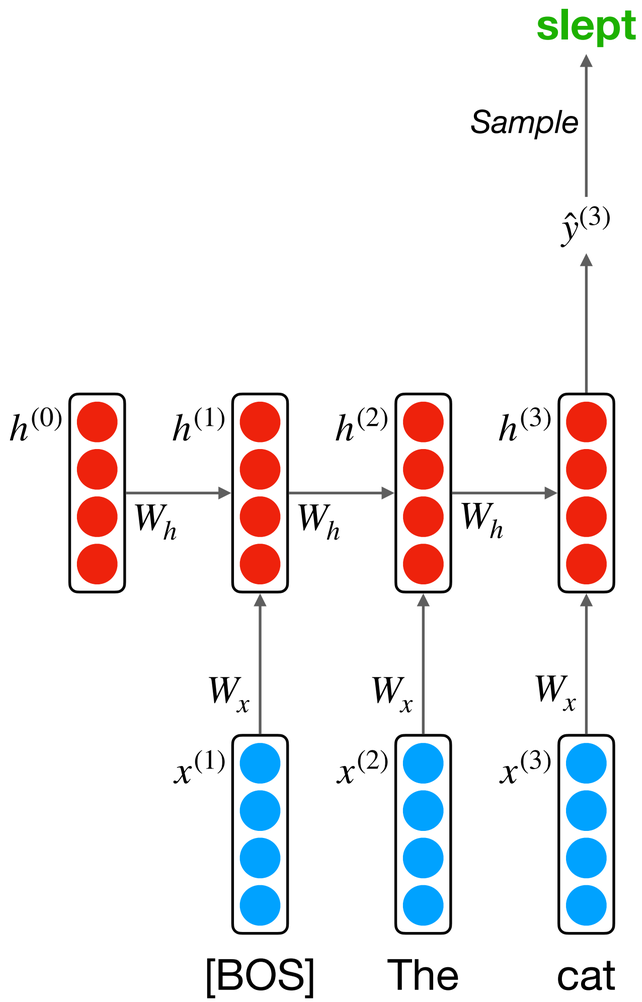
$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} J^{t}(\theta)$$

Mean of negative log prob

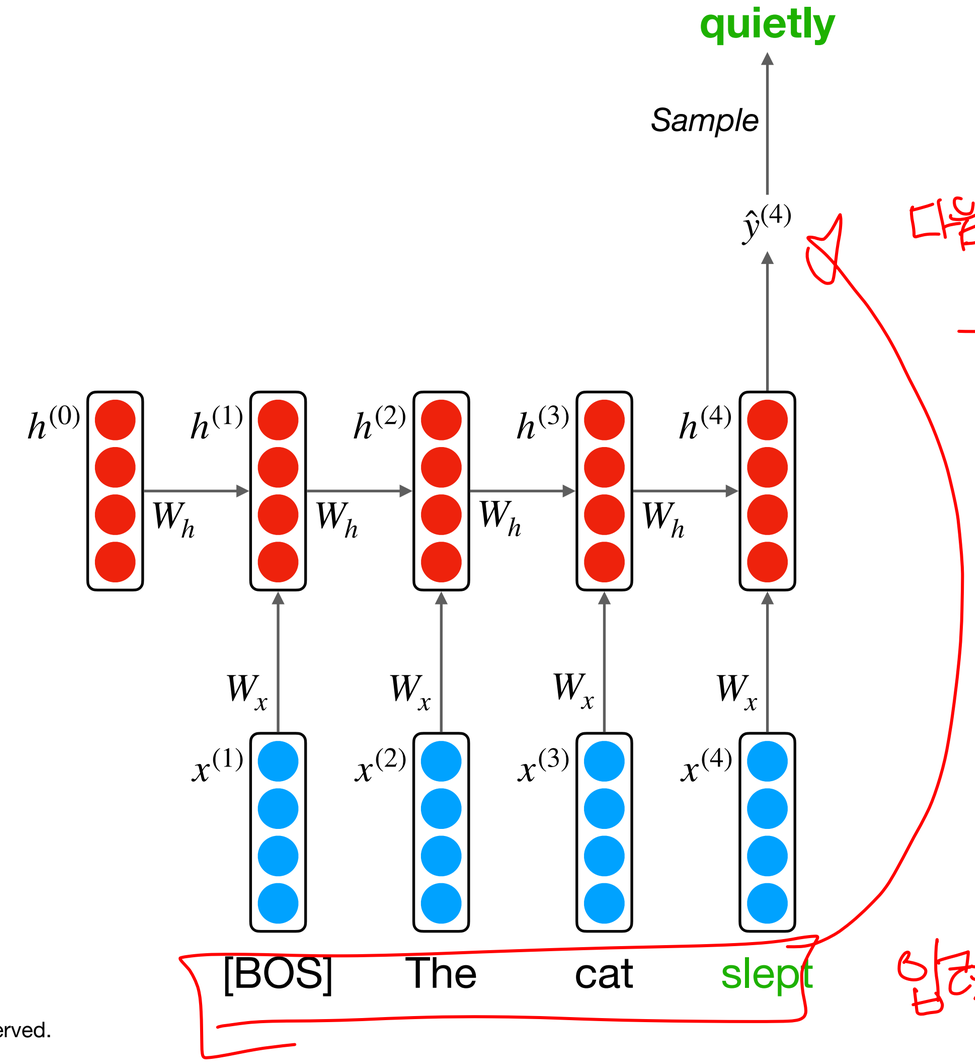
Neural Language Model (Generate Text)



Neural Language Model (Generate Text)



Neural Language Model (Generate Text)



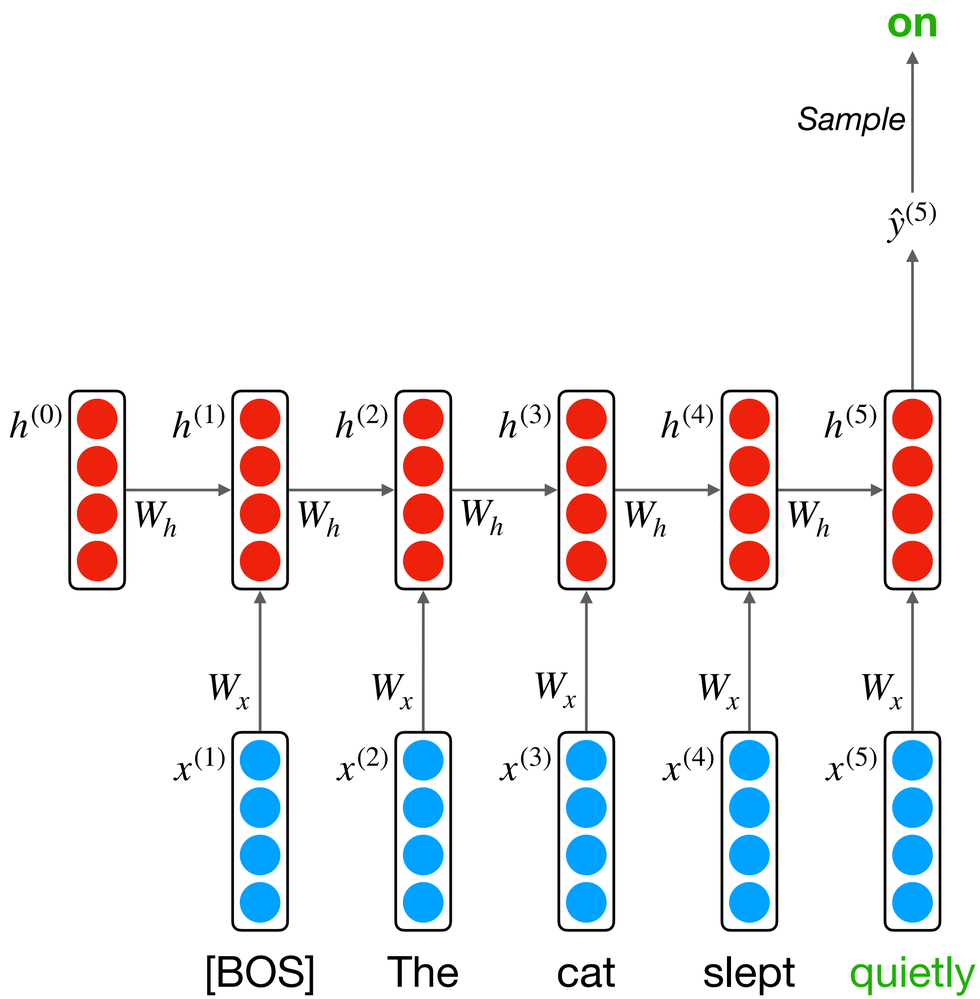
다음 단어 예측을 학습한지

→ 그것을에 Sample 통해 넣어주기

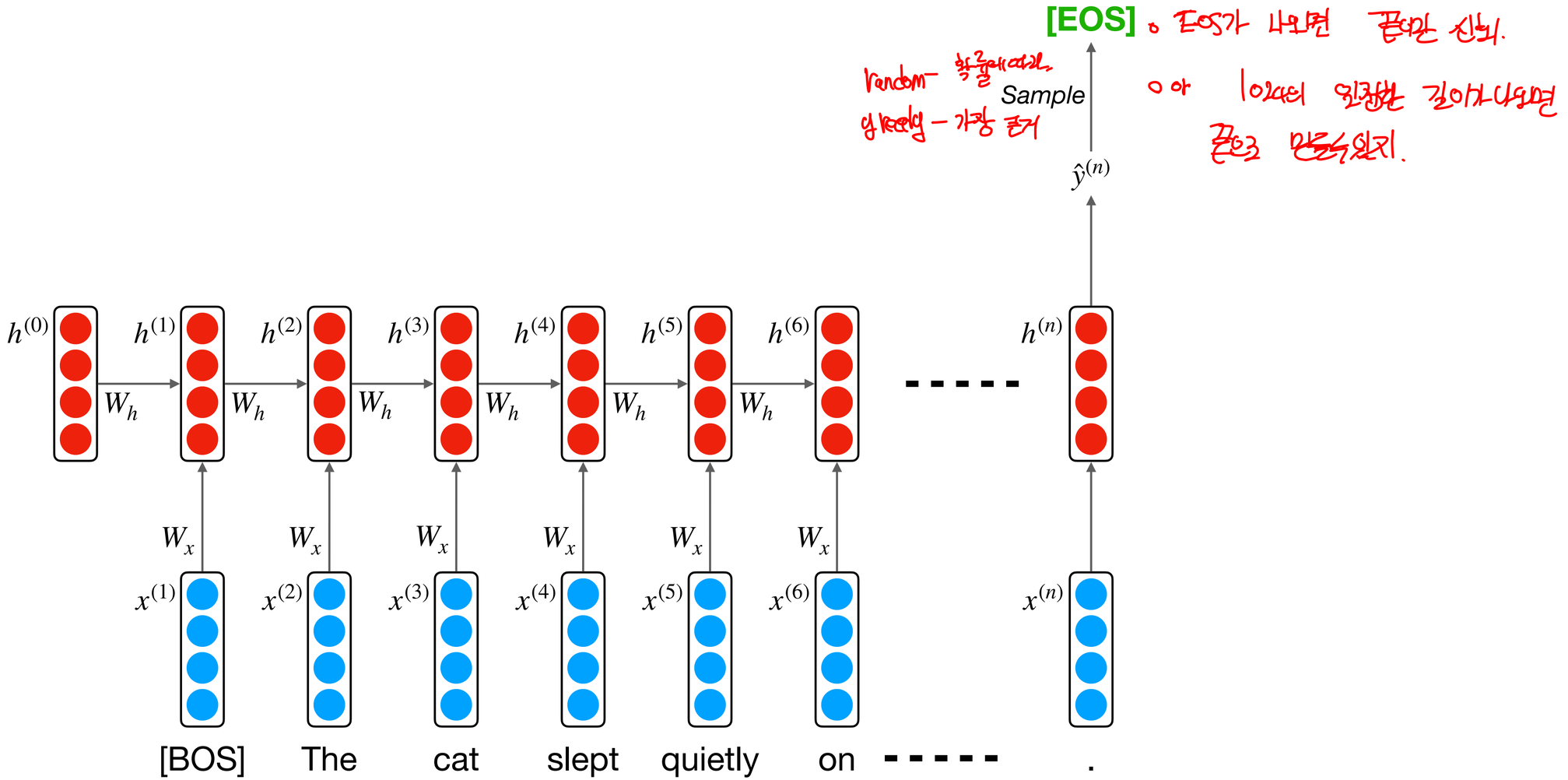
⇒ 이때 random sample 인자

입력 4개.

Neural Language Model (Generate Text)



Neural Language Model (Generate Text)



Neural Language Model (Subcomponent)

- Predicting Typing
- Speech recognition
- Handwriting recognition
- Spelling/grammar correction
- Authorship identification
- Machine Translation
- Summarization
- Dialog
- etc.

Neural Language Model (Metric)

$$perplexity = \prod_{t=1}^{T} \left(\frac{1}{P_{LM}\left(x^{(t+1)} \mid x^{(1)}, x^{(2)}, \ldots, x^{(t)}\right)} \right)^{\frac{1}{T}}$$

Normalize by number of words

Inverse probability of corpus

Neural Language Model (Metric)

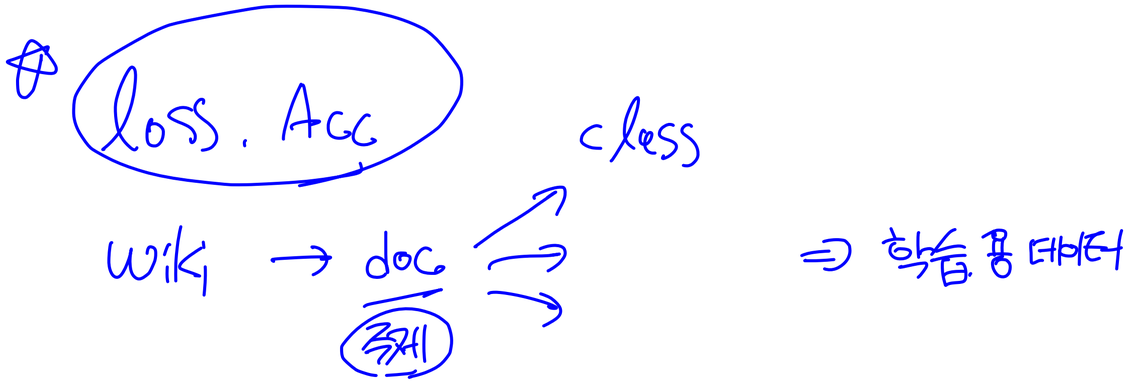
$$perplexity = \prod_{t=1}^{T} \left(\frac{1}{P_{LM}\left(x^{(t+1)} \mid x^{(1)}, x^{(2)}, \ldots, x^{(t)}\right)} \right)^{\frac{1}{T}}$$

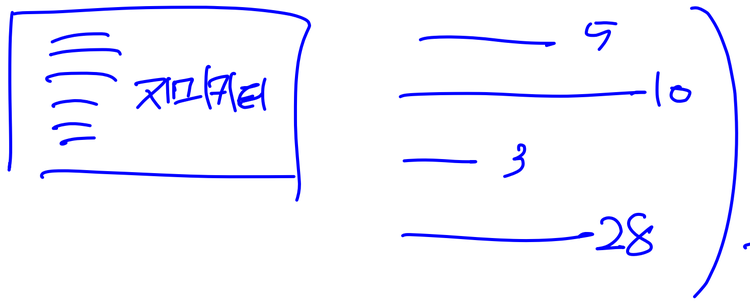
$$= \prod_{t=1}^{T} \left(\frac{1}{\hat{y}^{(t)}_{x_{t+1}}} \right)^{\frac{1}{T}} = \exp\left(\log\left[\prod_{t=1}^{T} \left(\frac{1}{\hat{y}^{(t)}_{x_{t+1}}} \right)^{\frac{1}{T}} \right] \right) = \exp\left(\frac{1}{T} \sum_{t=1}^{T} -\log \hat{y}^{(t)}_{x_{t+1}} \right)$$

$$= \exp(J(\theta))$$

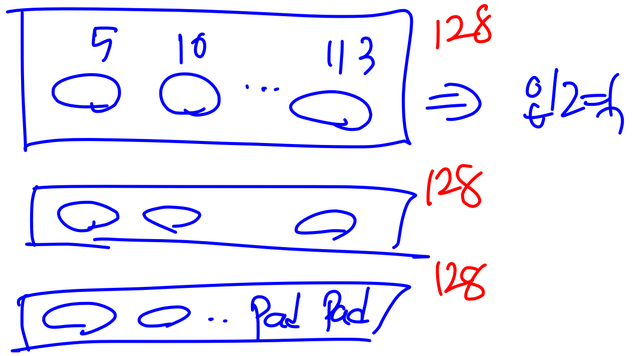
Low perplexity is better !!!

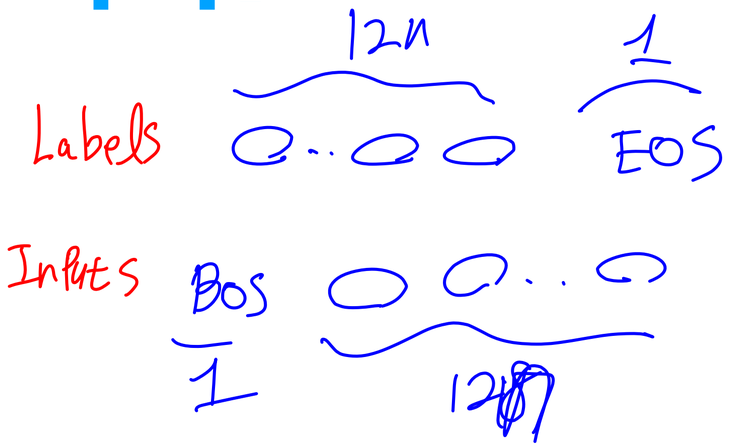
감사합니다.





감사합니다.





학습 데이터

다운스트림 태스크

3층 병렬학습 과정

batch-size=1024

1GPU 128

- capacity limit

big model 만들 때 write full

- 한국어에 적합한 tokenizer

- subword tokenize → 적절한 토큰

BPE

- 좋은 지표의 변화

- Morpheme

- 유창한가?
- 얼마나 유창한지 성능을 확인하기 어렵지.

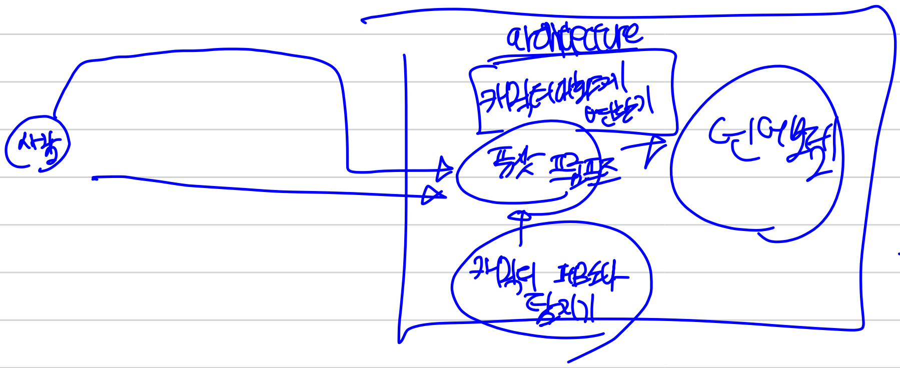
Bule score 로도 확인 어렵지.

- Vocab이 다른거는 perplexity로 확인이 어려워지

- real vs fake .

- 일관된 캐릭터의 대화에 유지 + 유창성

- 캐릭터 세계관



P-turning

Prompt-tuning

Prompt-control-unit