CH. 1 Probability & Language model

Def 1.1 Language Model

단어 시퀀스에 대해 확률을 부여, 즉 해당 시퀀스가 얼마나 자연스러운가를 평가한다.

$w_1, w_2, \dots w_{t-1}, \boxed{w_t}, \dots w_m$

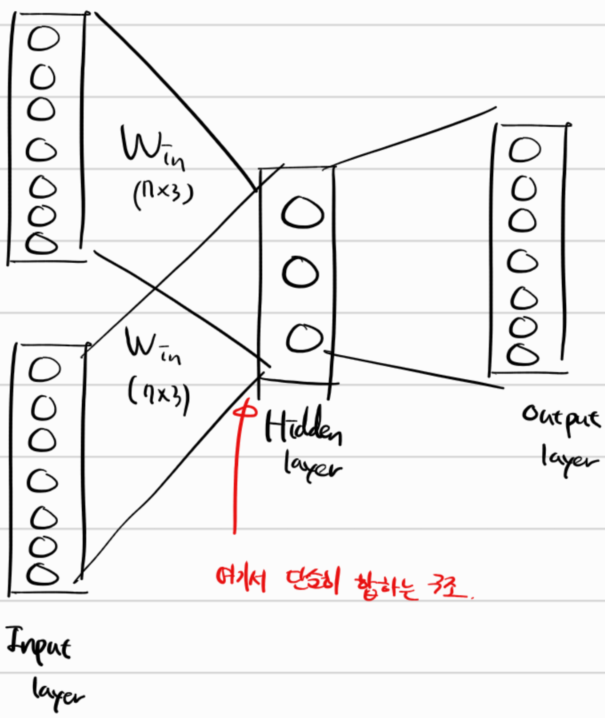
Goal : $P(w_t \mid w_1, w_2, \dots w_{t-1})$

이 표현 때문에 Conditional Language Model 이라고도 함!

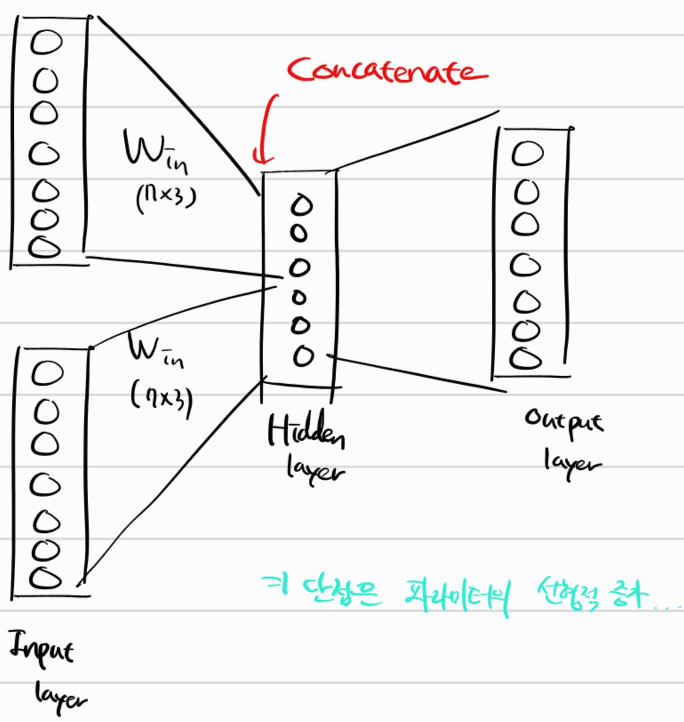
$$P(w_1, w_2, \dots w_m) = \prod_{t=1}^{m} P(w_t \mid w_1, \dots w_{t-1}) \approx \prod_{t=1}^{m} P(w_t \mid w_{t-2}, w_{t-1})$$

2층 마르코프 연쇄

Note 1.2 CBOW는 맥락에 순서가 없다!



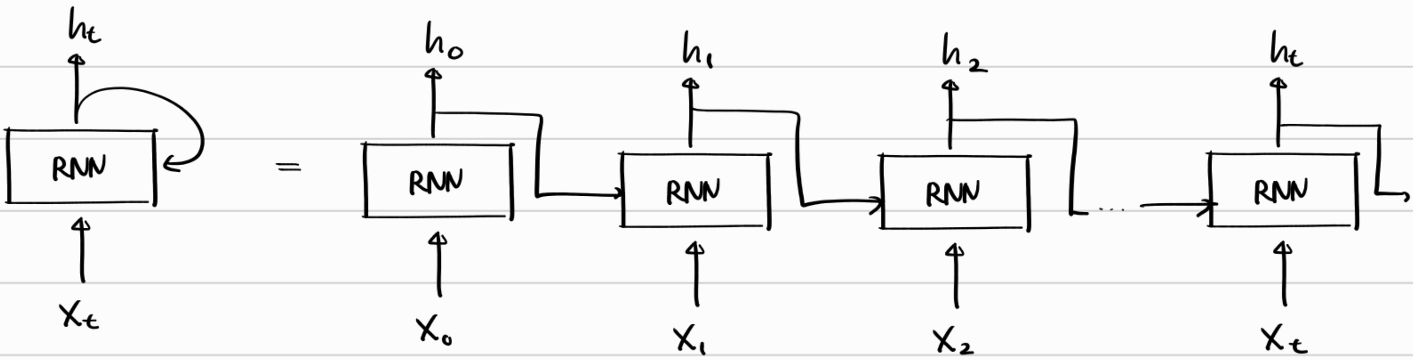
Neural Probabilistic Language Model



즉 CBOW는 언어모델로 쓰기는 어렵다. 그저 단어의 분산 표현을 얻는 것!

CH. 2 Recurrent Neural Network

Def 2.1 Recurrent Architecture

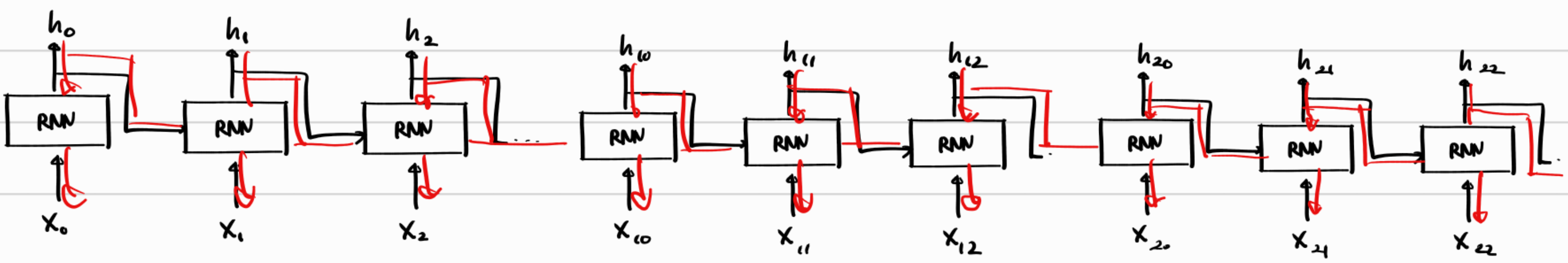


이건 feed forward와 유사.

$$h_t = \tanh(h_{t-1} W_h + x_t W_x + b)$$

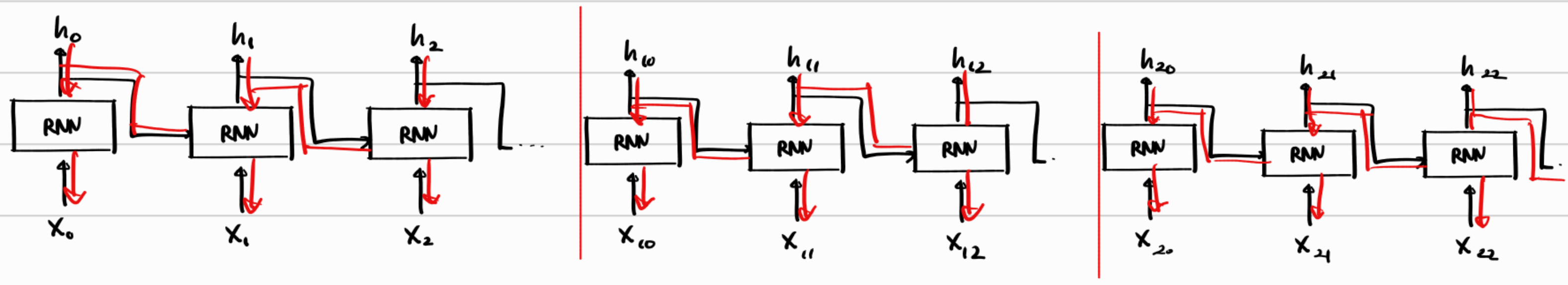
hidden State (vector)

Def 2.2 (Truncated) Back Propagation Through Time (BPTT)



↳ too long...

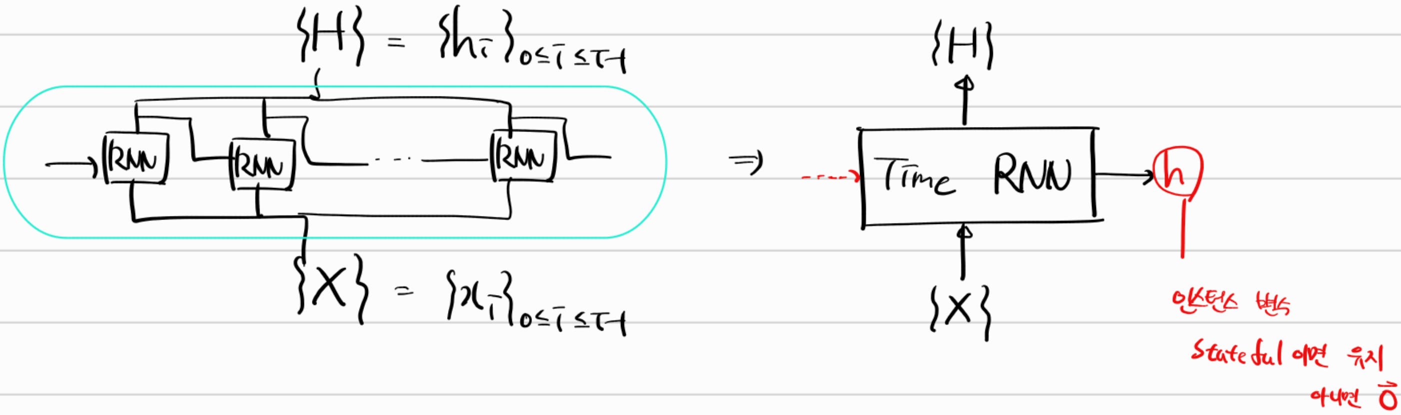
⟨Truncated⟩



Note 2.3 ① RNN은 반드시 데이터를 순서대로 제공해야한다!
② 미니 배치별로 시작 위치를 옮겨 줘야한다.

시계열 데이터 한 방에 처리!

Def 2.4 Time RNN

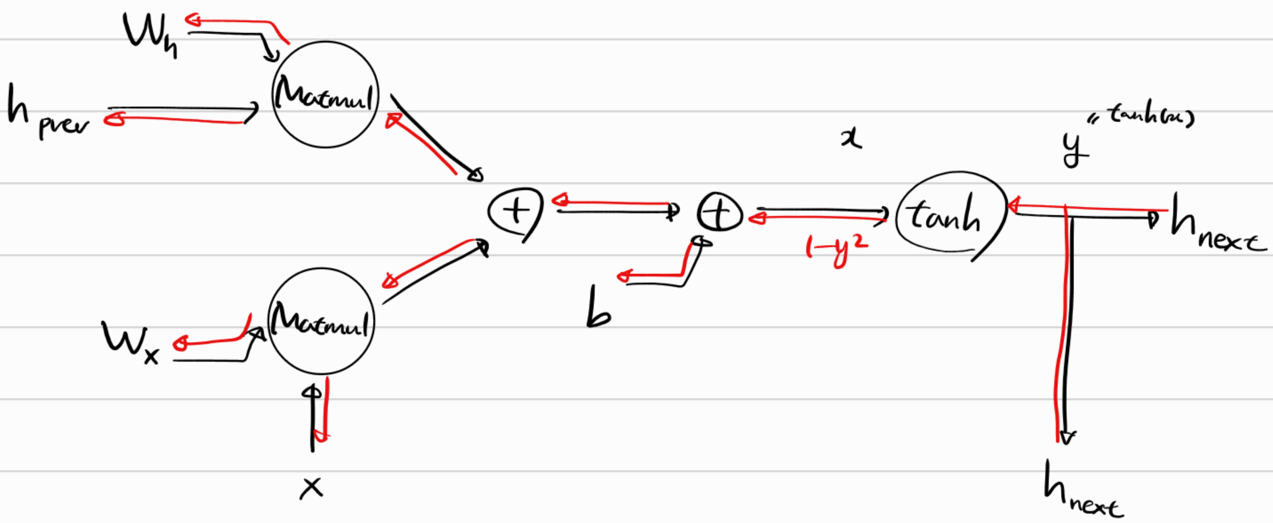


Thm 2.5 RNN matrix size

Let $\dim(h) = H$ and $\dim(x) = D$, batch size $= N$

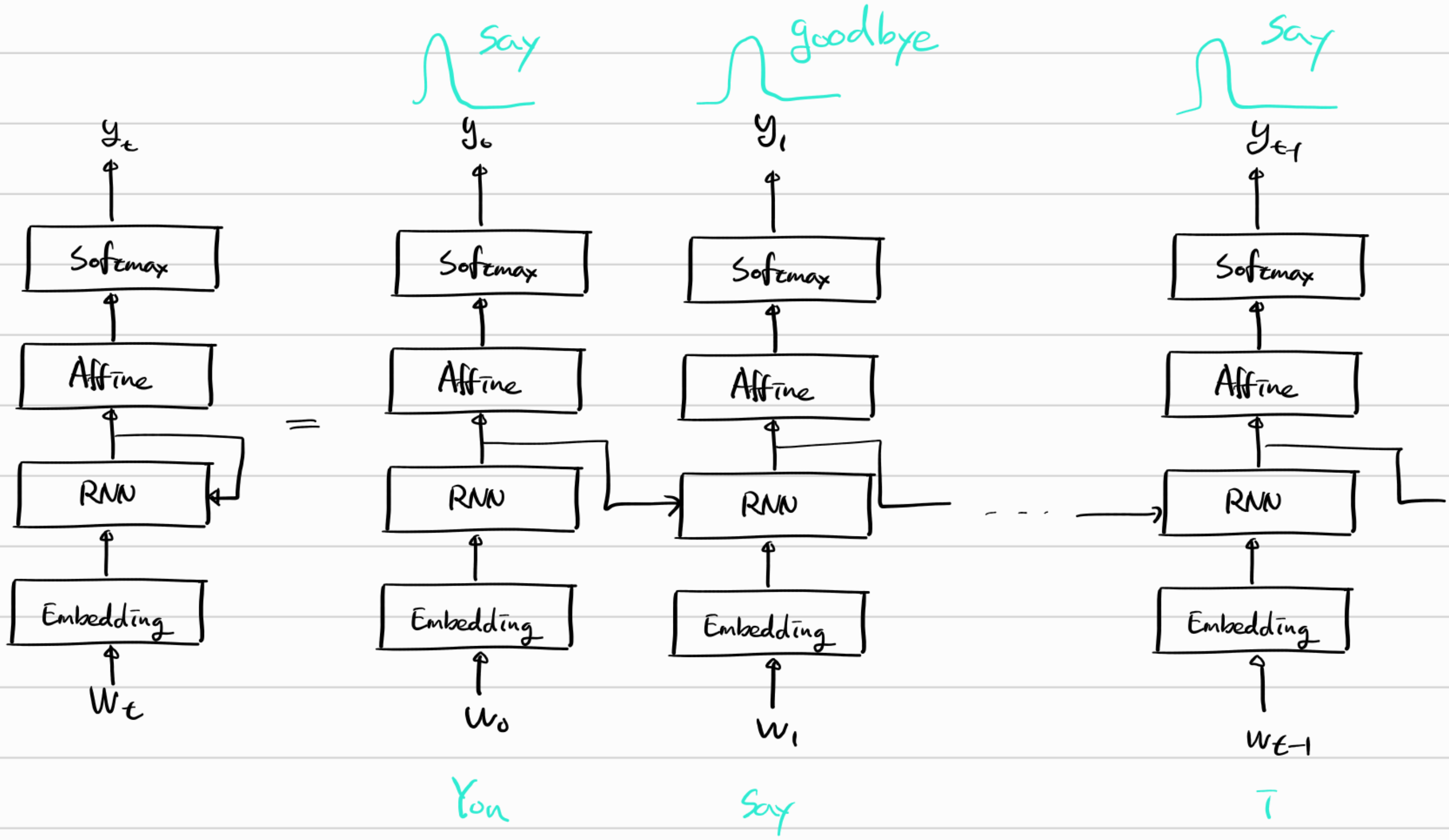
Then, $\underset{N \times H}{\underline{h_{t-1}}} \times \underset{H \times H}{\underline{W_h}} + \underset{N \times D}{\underline{x_{t-1}}} \times \underset{D \times H}{\underline{W_x}} + \underset{N \times H}{\underline{b}}$

Thm 2.6 RNN's Computational Graph

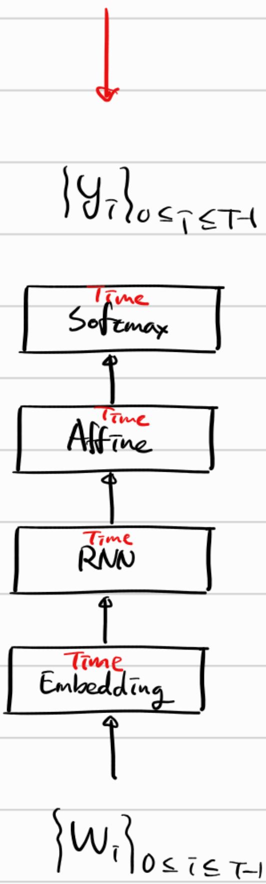


CH.3 RNNLM

Def 3.1 RNNLM Architecture



Thm 3.2 Time RNN LM



Def 3.3 Perplexity : Performance metric for LM

단순하게 확률의 역수임. 의도는 # of branches, 즉 확률 분포로 다음 단어로 출현할 후보의 개수를 의미

ex) $P(\text{say}) = 0.8 \Rightarrow \text{Perplexity}(\text{say}) = 1.25$

$P(\text{say}) = 0.2 \Rightarrow \text{Perplexity}(\text{say}) = 5$

정교하게는,

Let $L = -\frac{1}{N}\sum_n \sum_k t_{nk} \log y_{nk}$

then, $\boxed{\text{Perplexity} = e^L}$