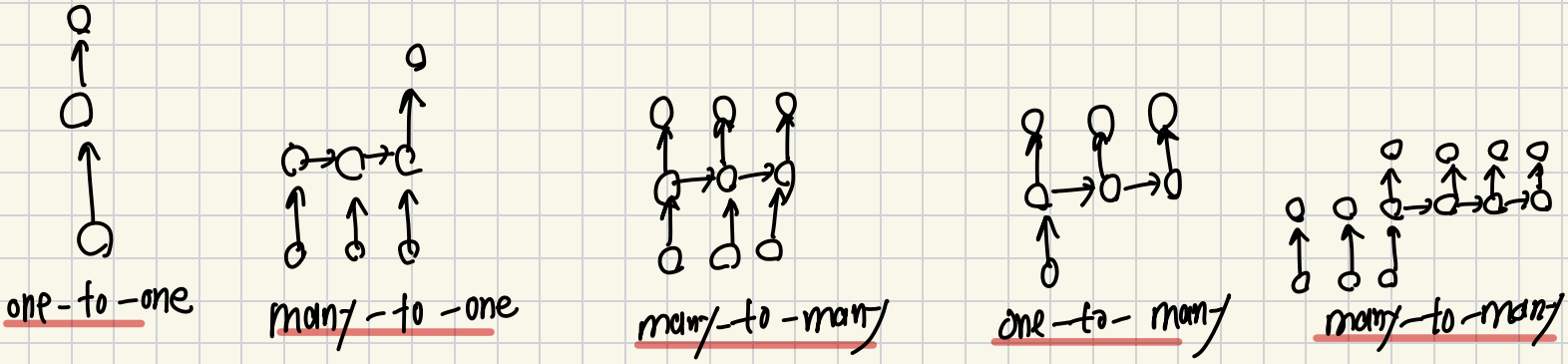
Sequential Data

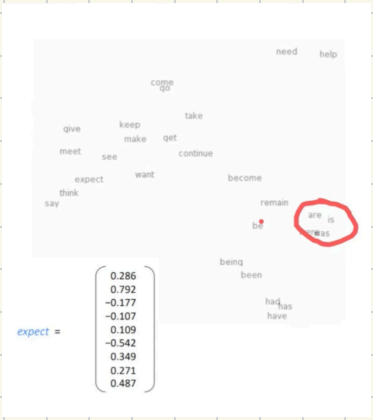
순서가 중요한 data들



Word embedding

data를 vector로 만들어준다.

embedding : N-d를 유클리드 dimension의 의미를 가지고 좌표(벡터)가도록



비슷한 의미들은 distance가 ↓

다른 의미들은 distance가 ↑ 가 되도록

word를 vector로 embedding 한다.

word2vec : 의미있는 최소 단위

data driven approach ⇒ 사람적 정의 사용X 어느 단어와 같이 사용되는지 통계적 추론.

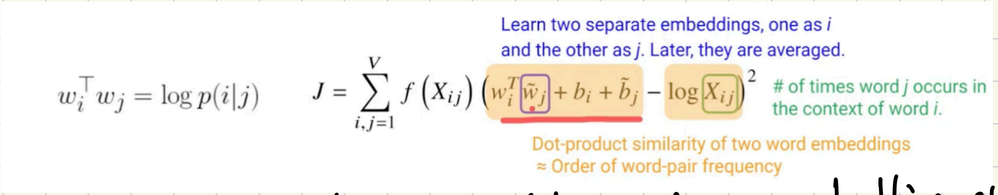
〈 word2vec 〉

⇒결국 Sentence에서 앞과 뒤를 likelihood $\left(L(\theta)=\prod_{t=1}^{T}\prod_{-m\le j\le m}P(w_{t+j}|w_t;\theta)\right)$ 를 maximize.

x_{t-2} x_{t-1} x_t x_{t+1} x_{t+2}

$P(x_t|x_{t-2})$ $P(x_{t+1}|x_t)$

< Glove >

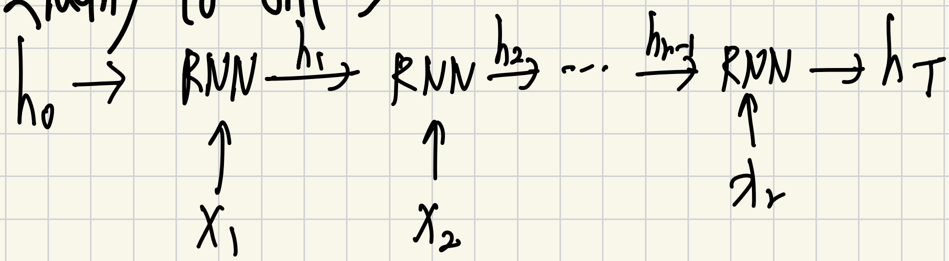


: 자주 동시에 나오는 단어들을 비슷한 vector로 embedding 하자라는 Idea이다.

RNN for sequence classification (T/F를 맞추기)

* Image와 다르게 임의의 길이를 갖는 data를 처리할 수 있어야한다.

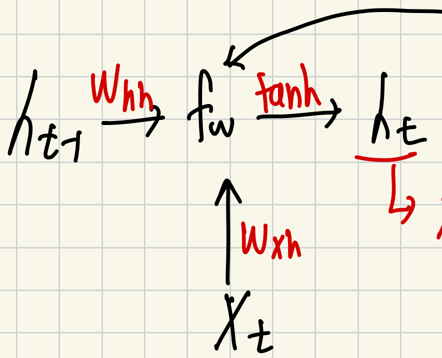
<Many to one>



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

이전 h와 입력을 이용.

→ f_w는 어느 step에서도 same이다.
Because 길이에 영향을 받으면 안되기때문

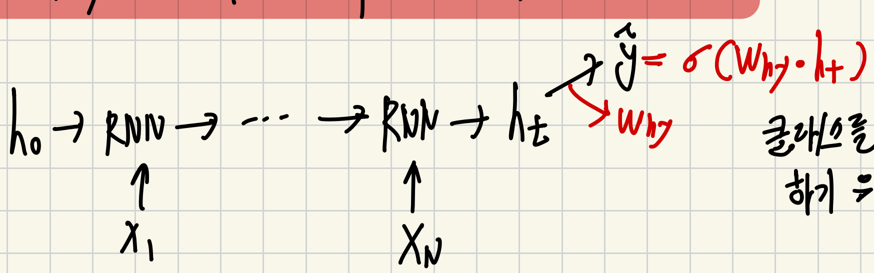


• $f_w = \tanh(x_t \cdot W_{xh} + h_{t-1} \cdot W_{hh})$

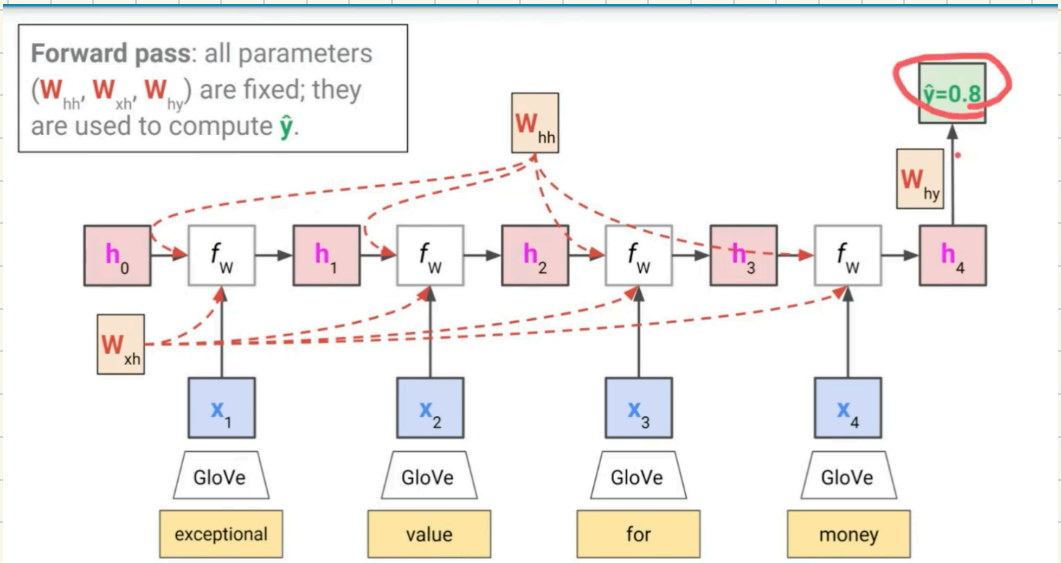
↳ $x_1 \sim x_t$까지 data를 보고 결정된 feature.

: 모든 weight (W_{hh}, W_{xh})는 모든 step에 대해 same.

최종 result를 얻기 위해서



클래스를 P로 change 하기 위함.



이때 Backprop을 하면 초기 단계는 계속 곱해져 W3 gradient를 구해주어야한다.

< Many to Many >

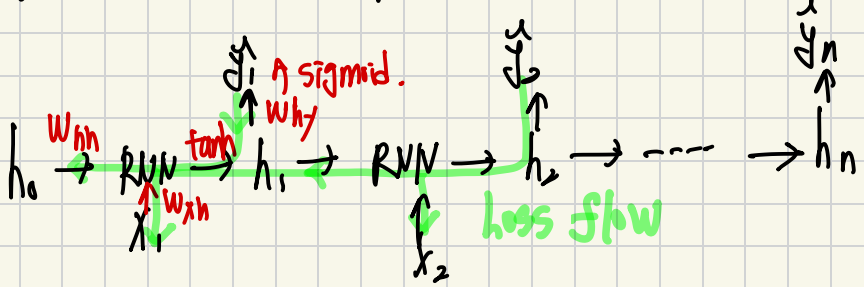
언어 = 단어들을 특정 규칙에 맞게 나열하는 것.

Traditional : n-grams.

모든 data에 대해 조건부 확률로 A라는 단어가 나올때 B가 나올 확률을 모두 사전으로등록

⇒ 결국 정보량↑ 사전의 크기가 ↑ 되게 된다!.

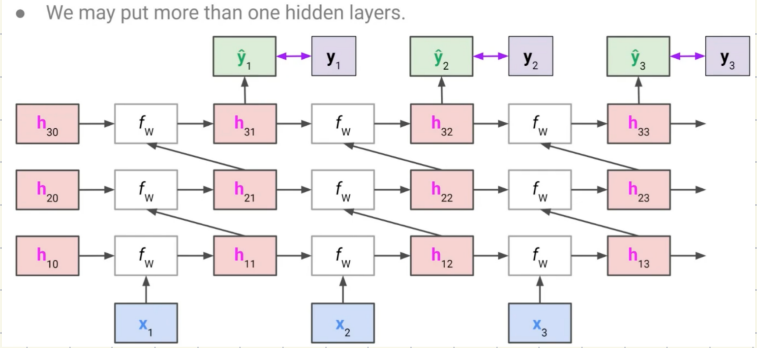
X_t가 들어오자 마자 output을 출력해보자.



X는 사전에 존재하는 모든 word들 中 뭐가 가장 높은 단어로 예측.

⇒ train은 cross entropy 사용.

Multi-Layer RNN (RNN을 여러층 쌓아올린다)



장) Input sequence 길이에 제약 X.

→ 매 step 가중치들을 일일이 하여 해결.

단) 느리다 (순차적 처리)

vanishing gradient & long range dependence