# Recurrent Neural Network

Neural dependency parser

Language Model : 확률 분포 기반으로 주어진 문맥 이후 위치할 단어 예측

sequence

$$P(x^{(1)}, \dots, x^{(T)}) = \prod_{t=1}^{T} P(x^{(t)} \mid x^{(t-1)}, \dots, x^{(1)})$$ multiplication rule

자동완성 | 🔍 코딩하기 좋은 ? | $P(x^{(3)} \mid$ 코딩하기, 좋은$)$

카페

노트북

⋮

↳ 가장 높은 확률분포값 반환

조건부확률분포값 찾기

N-gram Language Model (빈도수 바탕) => sparsity problem.

Neural Language Model

- 미리 지정한 window size 이전 단어 바탕으로 예측,
- Word embedding : 단어의 distributed representation 학습
- BUT, window size 커지면 parameters 증가 : Computation↑, overfitting

두 모델 모두 한정된 길이에 예측, 모든 문맥 고려 불가

## Recurrent Neural Network

1. Input word sequence
   - 원핫 벡터 $x^{(t)} \in \mathbb{R}^{|V|}$ $x^{(1)} = (0, 0, 1, 0, \cdots, 0)$

2. Word embedding
   $e^{(t)} = E x^{(t)}$ $E : (|V| \times d_e)$

3. Hidden States

$$h^{(t)} = \sigma(W_h h^{(t-1)} + W_e e^{(t)} + b_1)$$

$\underbrace{W_h : (d_h \times d_h), W_e : (d_e \times d_h)}$

연산 처음부터
끝까지 동일가중치

4. Output distribution

$$\hat{y}^{(t)} = \text{softmax}(\underline{U} h^{(t)} + b_2) \in \mathbb{R}^{|V|}$$

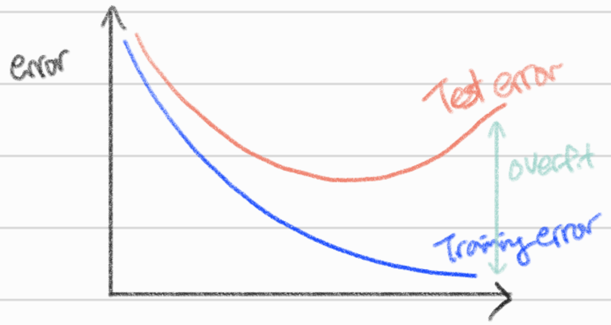
이전에 나온 모든 단어 고려 가능

매 step마다 다음 단어 예측 ⇒ many-to-many
(입력·출력 존재)

Classification 이므로, Cross Entropy Loss 사용

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

Long Term Dependency (장기의존성) 반영 어려움



Regularization 필요
- Dropout for NN

Non-linearities

logistic ("sigmoid") tanh hard tanh ReLU

Optimizers

Adagrad RMSprop Adam