

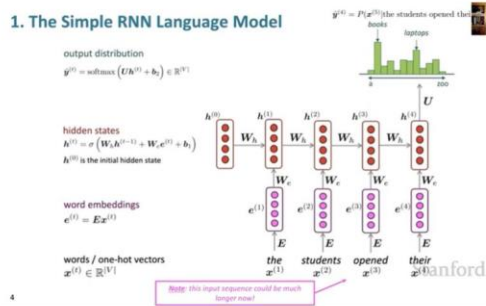
## ▼ 6주차 예습과제

### + RNN

input sequence 의 각 word embeddings과 hidden states 을 입력으로 사용하고 output vector를 softmax에 통과하여 다음 단어 예측

다음과 같이 t가 4일 때, output distribution은 다음과 같이 산출된다.

#### 1. The Simple RNN Language Model



RNN language model이 특정 t시점일 때 loss function은 다음과 같이 나타낼 수 있다.

#### Training a RNN Language Model

- However: Computing loss and gradients across entire corpus  $x^{(1)}, \dots, x^{(T)}$  is too expensive!
- $$J(\theta) = \frac{1}{T} \sum_{t=1}^T J^{(t)}(\theta)$$
- In practice, consider  $x^{(1)}, \dots, x^{(T)}$  as a sentence (or a document)
  - Recall: Stochastic Gradient Descent allows us to compute loss and gradients for small chunk of data, and update.
  - Compute loss  $J(\theta)$  for a sentence (actually, a batch of sentences), compute gradients and update weights. Repeat.

Stanford

하지만, 전체 corpus를 분석하기 위해 loss와 gradients를 따로 계산하는 것은 많은 시간이 소요되기 때문에 문장이나 문서 등의 큰 단위로 나누어 입력

#### Evaluating Language Models

- The standard evaluation metric for Language Models is perplexity.

$$\text{perplexity} = \prod_{t=1}^T \left( \frac{1}{P_{\text{LM}}(x^{(t+1)} | x^{(1)}, \dots, x^{(t)})} \right)^{1/T}$$

Normalized by number of words

Inverse probability of corpus, according to Language Model

- This is equal to the exponential of the cross-entropy loss  $J(\theta)$ :

$$= \prod_{t=1}^T \left( \frac{1}{\hat{y}_{t+1}^{(t)}} \right)^{1/T} = \exp \left( \frac{1}{T} \sum_{t=1}^T -\log \hat{y}_{t+1}^{(t)} \right) = \exp(J(\theta))$$

Lower perplexity is better!

Stanford

RNN의 장점

- 입력의 길이에 제한이 없음
- 입력에 따른 모델 크기에 변화가 없음
- 각 과정에서 동일한 가중치를 적용하므로 대칭적임

RNN의 단점

- 단계의 진행을 위해서 해당 단계의 전 단계의 계산이 완료되어야 하므로 최종 계산까지 시간이 많이 소요
- vanishing gradient problem의 문제로 context가 모두 반영되지 않는 문제
- 장기 메모리 불가, 초기 데이터 유치가 어려기 때문

## LSTR(Long Short-Term Memory RNNs)

텍스트 초기 부분의 정보를 유지할 수 있기 때문에 RNN의 short term memory 문제 해결이 가능하다.

LSTR은 각 t마다 다음 cell로 정보를 보내기 전에 정보들을 유지할지 제거할지 결정한다. 필요없다고 여겨지는 정보는 longtermmemory로 가져가지 않기 위함이다. 이때 필요성에 대한 판단의 정확도를 높이기 위해 training이 필요하다.

### input

새로운 정보를 long term memory에 저장할지 결정

첫번째 layer: 새로운 정보의 통과여부 필터링

두번째 layer: short-term memory와 input을 tanh 활성화 함수에 필터링

### output

output에서는 다음 cell에 전달될 short term memory와 hidden state를 만들 input 값을 생성한다.

### [절차]

forget layer를 지날 때 이전 특징 별로 통과 여부 결정 → input gate를 통해 해당 t 시점에서의 중요성을 판단 → 중요성을 바탕으로 업데이트할 정보 결정 → cell state에 저장 → t 시점의 forget gate, input gate에 의해 결정되어진 현시점의 cell state 값을 다음 layer로의 출력 여부 결정