

# 2022학년도 1학기 컴퓨터언어학

## 제9강 신경망 언어 모형 (3)

박수지

서울대학교 인문대학 언어학과

2022년 4월 4일 월요일

## 오늘의 목표

- 1 순방향신경망이 자연어처리에서 어떻게 사용되는지 예를 들어 설명할 수 있다.

## 중간고사 공고

**대상** 수강생 전원

**일시** 2022년 4월 18일 월요일 강의 시간 (딱 2주 남음)

**장소** 미정

**범위** 2강(2022-03-07)부터 12강(2022-04-13)까지 배운 내용

# 인공신경망(ANN: Artificial Neural Network)

## 순방향신경망(FFNN/FNN: Feed-forward Neural Network)

### 1 N층 순방향신경망

- 신경망 전체의 구조
  - 입력층 1개(층의 개수로 세지 않음)
  - 은닉층  $(N - 1)$ 개
  - 출력층 1개
- 각 층의 구조 (가중치 매개변수 행렬  $\mathbf{W}^{[i]}$ , 활성화함수  $g^{[i]}$ )
  - 현재 계층의 입력값 (=이전 계층의 활성화값)  $\vec{h}^{[i-1]}$
  - 현재 계층의 가중합 (편향은 편의상 생략함)  $\vec{z}^{[i]} = \mathbf{W}^{[i]} \vec{h}^{[i-1]}$
  - 현재 계층의 활성화값 (=다음 계층의 입력값)  $\vec{h}^{[i]} = g^{[i]}(\vec{z}^{[i]})$
- 각 층은 완전연결계층이다.

## 순방향신경망을 자연어처리의 분류 문제에 활용할 때 제일 먼저 고려해야 할 것 입력을 어떻게 표상할 것인가?

- 출력 벡터의 크기와 내용은 문제의 성격에 의해 정해진다.

**감성 분석** 입력된 텍스트의 감성 극성을 예측하는 문제.

$c_1 = \text{긍정}, c_2 = \text{부정}, c_3 = \text{중립}$

**언어 모형** 현재 입력의 다음에 나올 단어를 예측하는 문제.

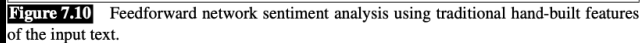
$c_1 = \text{가가멜}, \dots, c_{|V|} = \text{히타이트}$

- 은닉층은 데이터의 크기 등등을 고려하여 쌓으면 된다.

## 입력 벡터를 얻는 몇 가지 방법

- 1 손으로 만든 특성값 벡터
- 2 사전훈련된(pre-trained) 임베딩 벡터
- 3 단어들의 원-핫 인코딩 벡터

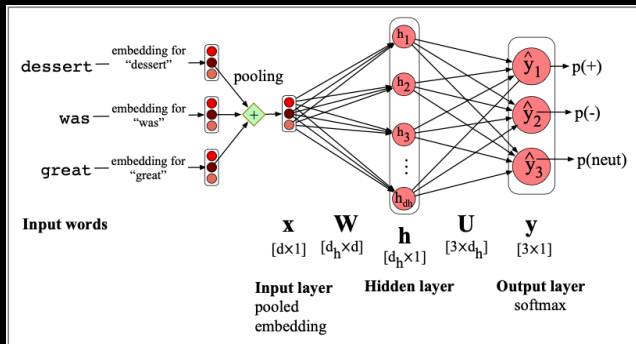
입력: 손으로 만든 특성값 벡터



은닉층 초매개변수.

# 예시: 감성 분석(Sentiment analysis)

입력: 사전훈련된 임베딩 벡터(의 풀링)



**Figure 7.11** Feedforward sentiment analysis using a pooled embedding of the input words.

## 아이디어

손으로 만든 특성값 대신 자동으로 학습한 임베딩 벡터를 사용한다.

## 구현 방법: Mean pooling

텍스트를 이루는 단어 벡터들의 성분별 평균값을 취하여 하나의 벡터를 얻는다.

# 언어 모형 벵락치기

## 언어 모형

기존 단어들의 이력  $h = w_1 w_2 \cdots w_{t-1}$  이 주어졌을 때 다음에 나올 단어  $w$  를 예측한다.  
= 조건부확률  $P(w|h) = P(w_t|w_1 w_2 \cdots w_{t-1})$  을 계산한다.

## N그램 언어 모형

단어 시퀀스의 길이를  $N$  으로 제한한다.  
= 기존 단어들의 이력을 직전  $(N - 1)$  개만 살펴본다.  
= 조건부확률을  $P(w_t|w_1 w_2 \cdots w_{t-1}) \approx P(w_t|w_{t-N+1} \cdots w_{t-1})$  로 근사한다.  
 $N = 2$  바이그램 모델  $P(w_t|w_{t-1})$   
 $N = 3$  트라이그램 모델  $P(w_t|w_{t-2} w_{t-1})$

# 언어 모형 벵락치기

## N그람 언어 모형의 확률 추정 방법

최대가능도추정법: N그람의 상대빈도로 확률을 추정한다.

$$N = 2 \text{ 바이그램 모델 } P(w_t|w_{t-1}) = \frac{\text{Count}(w_{t-1}w_t)}{\text{Count}(w_{t-1})}$$

$$N = 3 \text{ 트라이그램 모델 } P(w_t|w_{t-2}w_{t-1}) = \frac{\text{Count}(w_{t-2}w_{t-1}w_t)}{\text{Count}(w_{t-2}w_{t-1})}$$

## 상대빈도의 문제

훈련 코퍼스에 나타난 적이 없는 시퀀스의 확률은 0이 된다.

## 예시

$$P(\text{“곽가에게”} | \text{“조조는 빈 도시락통을”}) = 0$$



## 신경망 언어 모형의 해결

빈도 대신 임베딩을 도입한다.

### 예시

- 훈련 코퍼스: “I have to make sure that the cat gets fed.”
- 단어 임베딩:  $\text{sim}(\text{“cat”}, \text{“dog”}) \gg 0.7$

훈련 코퍼스에 나타나지 않은 시퀀스의 확률  $P(\text{“fed”}|\text{“the dog gets”})$ 의 값을  $P(\text{“fed”}|\text{“the cat gets”})$ 와 비슷하게 추정할 수 있다.

# 예시: 신경망 언어 모형의 순방향 추론

## 순방향 추론 (Forward inference)

입력이 주어졌을 때 다음 단어가 될 수 있는 가능한 출력들의 확률분포를 생산한다.

## 예시

입력이 “조조가 빈 도시락통을”일 때의 순방향 추론

- $P(\text{“곽가에게”} | \text{“조조가 빈 도시락통을”}) = 0.001$
- $P(\text{“순욱에게”} | \text{“조조가 빈 도시락통을”}) = 0.9$
- $P(\text{“유비에게”} | \text{“조조가 빈 도시락통을”}) = 0.01$
- ...

# 예시: 신경망 언어 모형의 순방향 추론

입력: 단어들의 원-핫 인코딩 벡터

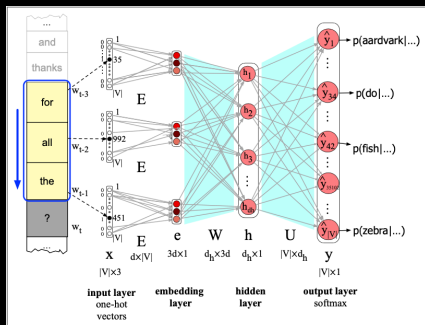
$$\begin{array}{c} |V| \\ \text{d} \end{array} \begin{array}{|c|} \hline \text{E} \\ \hline \end{array} \times \begin{array}{c} 1 \\ |V| \end{array} = \begin{array}{c} 1 \\ \text{d} \end{array} \begin{array}{|c|} \hline \text{e}_5 \\ \hline \end{array}$$

The diagram shows a matrix  $E$  with dimensions  $d \times |V|$ . A vertical green bar highlights the 5th column of  $E$ , labeled with a '5' below it. This matrix is multiplied by a one-hot vector of size  $|V| \times 1$ , which has a '1' at the 5th position and '5' next to it. The result is a vector of size  $d \times 1$ , labeled with a '1' above it and  $\text{e}_5$  below it. The vector is represented by a vertical green bar.

**Figure 7.12** Selecting the embedding vector for word  $V_5$  by multiplying the embedding matrix  $E$  with a one-hot vector with a 1 in index 5.

# 예시: 신경망 언어 모형의 순방향 추론

입력: 단어들의 원-핫 인코딩 벡터



**Figure 7.13** Forward inference in a feedforward neural language model. At each timestep  $t$  the network computes a  $d$ -dimensional embedding for each context word (by multiplying a one-hot vector by the embedding matrix  $E$ ), and concatenates the 3 resulting embeddings to get the embedding layer  $e$ . The embedding vector  $e$  is multiplied by a weight matrix  $W$  and then an activation function is applied element-wise to produce the hidden layer  $h$ , which is then multiplied by another weight matrix  $U$ . Finally, a softmax output layer predicts at each node  $i$  the probability that the next word  $w_t$  will be vocabulary word  $V_i$ .

## 아이디어

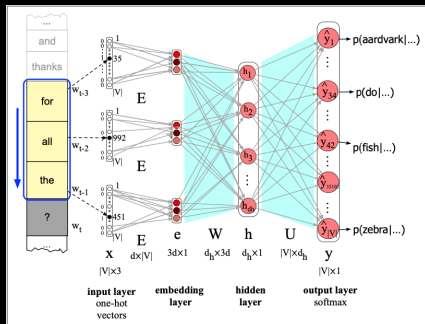
사전훈련된 임베딩 벡터가 없어도 임베딩을 이용할 수 있다.

## 구현 방법: Embedding layer

신경망 모형을 훈련시킬 때 임베딩 매개변수를 가중치로 포함시킨다.

# 예시: 신경망 언어 모형의 순방향 추론

입력: 단어들의 원-핫 인코딩 벡터



**Figure 7.13** Forward inference in a feedforward neural language model. At each timestep  $t$  the network computes a  $d$ -dimensional embedding for each context word (by multiplying a one-hot vector by the embedding matrix  $E$ ), and concatenates the 3 resulting embeddings to get the embedding layer  $e$ . The embedding vector  $e$  is multiplied by a weight matrix  $W$  and then an activation function is applied element-wise to produce the hidden layer  $h$ , which is then multiplied by another weight matrix  $U$ . Finally, a softmax output layer predicts at each node  $i$  the probability that the next word  $w_t$  will be vocabulary word  $V_i$ .

아이디어

입력으로 직전  $(N - 1)$  개 단어를 살펴본다.

구현 방법: Concatenation

$(N - 1)$  개의 임베딩 벡터들을 이어붙여 하나의 벡터를 얻는다.

## 다음 시간에 배울 것

- 순방향신경망 모형의 손실함수를 정의하고 훈련시키는 방법