

## CUAI CS224N 스터디 1팀

2025.03.18

발표자 : 민세희

## 스터디원 소개 및 만남 인증

스터디원 1 : 정인혁

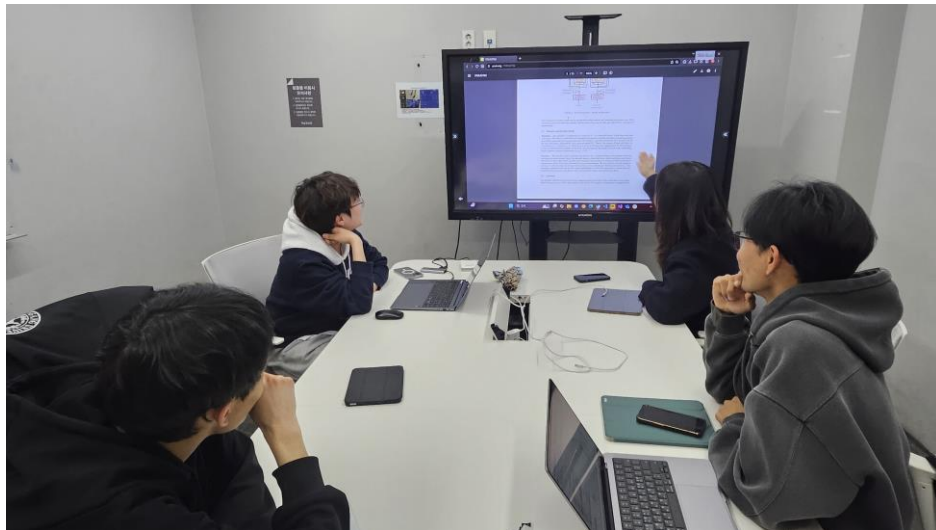
스터디원 2 : 태아카

스터디원 3 : 민세희

스터디원 4 : 김지호

스터디원 5 : 민유안

스터디원 6 : 양희원



## 스터디 소개

Stanford CS224N: NLP with Deep Learning | Spring 2024 | Lecture 1 - Intro and Word Vectors

Stanford Online

1:20:17

2 Stanford CS224N: NLP with Deep Learning | Spring 2024 | Lecture 2 - Word Vectors and Language Models

Stanford Online

1:19:12

3 Stanford CS224N: NLP with Deep Learning | Spring 2024 | Lecture 3 - Backpropagation, Neural Network

Stanford Online

1:13:27

4 Stanford CS224N: NLP with Deep Learning | Spring 2024 | Lecture 4 - Dependency Parsing

Stanford Online

1:18:57

5 Stanford CS224N: NLP with Deep Learning | Spring 2024 | Lecture 5 - Recurrent Neural Networks

Stanford Online

1:18:52

6 Stanford CS224N: NLP with Deep Learning | Spring 2024 | Lecture 6 - Sequence to Sequence Models

Stanford Online

Stanford CS224N: NLP with Deep Learning | Spring 2024 | Lecture 1 - Intro and Word Vectors

Stanford Online

구독자 73.8만명

410

공유

저장

Date	Description	Course Materials	Events	Deadlines
<b>Week 1</b>	Word Vectors <a href="#">[slides]</a> <a href="#">[notes]</a>	Suggested Readings: 1. Efficient Estimation of Word Representations in Vector Space (original word2vec paper) 2. Distributed Representations of Words and Phrases and their Compositionality (negative sampling paper)	Assignment 1 <b>out</b> <a href="#">[code]</a>	
Tue Jan 7				
Thu Jan 9	Word Vectors and Language Models <a href="#">[slides]</a> <a href="#">[notes]</a> <a href="#">[code]</a>	Suggested Readings: 1. GloVe: Global Vectors for Word Representation (original GloVe paper) 2. Improving Distributional Similarity with Lessons Learned from Word Embeddings 3. Evaluation methods for unsupervised word embeddings  Additional Readings: 1. A Latent Variable Model Approach to PMI-based Word Embeddings 2. Linear Algebraic Structure of Word Senses, with Applications to Polysemy 3. On the Dimensionality of Word Embedding		

## CS224N: Natural Language Processing with Deep Learning

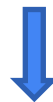
매 주 유튜브 강의 영상 + 강의 노트 + Suggested Readings 논문 리뷰 스터디 진행

## 스터디 진행 방향

NLP Study Schedule		
수행날짜	주	비고
03/11~03/17	1	Lecture 1,2
03/18~03/24	2	Lecture 3,4 + <u>논문 리딩</u>
03/25~03/31	3	Lecture 5,6
04/01~04/07	4	Lecture 7,8
04/08~04/28	-	중간 고사
04/29~05/05	5	추후 논의
05/06~05/12	6	
05/13~05/19	7	
05/20~05/26	8	팀별 프로젝트 선정    발표 준비
05/27	9	발표
05/29~	-	기말 고사

스터디 방식(중간 이전)

(개인 학습) 매 주 2강 학습  
+  
관련 논문 1개 리딩



(대면 스터디) 3명씩 조를 나누어  
2명-강의 리뷰 & 질의응답 진행  
1명-논문 리뷰 & 질의응답 진행  
Ex) 3강 리뷰 - 민세희  
4강 리뷰 - 민유안  
논문 리뷰 - 정인혁  
다음주 - 남은 3명이 진행

## 1주차 스터디 내용

어떻게 인간의 언어를 컴퓨터가 이해할 수 있게 만들까?

전통적 NLP: 단순히 단어를 One-Hot vector로 표기

문제점

- 단어 간 유사성을 표현할 수 없음
- Ex): Seattle motel을 검색할 때 당연히 Seattle hotel도 포함하여 검색하고 싶지만, 단순히 원-핫 벡터로 표현했을 경우 컴퓨터는 이를 이해할 수 없음

But:

motel = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

hotel = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]

These two vectors are orthogonal

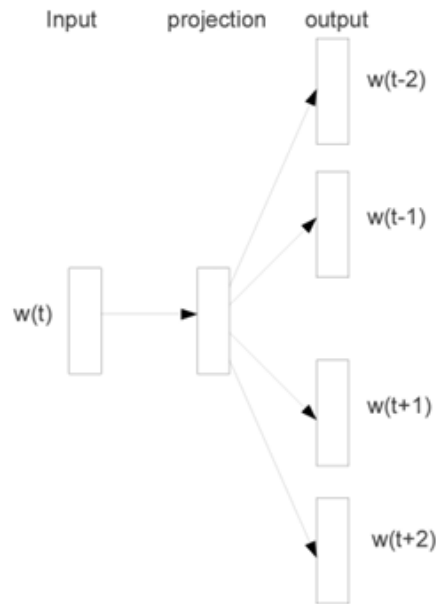
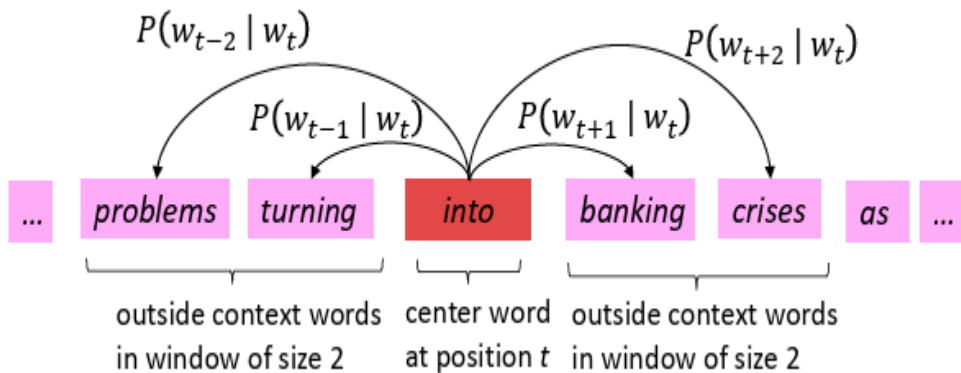
There is no natural notion of similarity for one-hot vectors!

## 1주차 스터디 내용

**Word2vec:** 중심 단어와 주변 단어의 관계성을 이용하여 단어의 임베딩 표현을 구함

Skip-gram: 중심단어->주변단어

CBOW: 주변단어->중심단어



Skip-gram model  
(Mikolov et al. 2013)

## 1주차 스터디 내용

SoftMax: Word2vec에서 주변 단어 예측 시 사용, 즉 다중 클래스 분류 문제  
But 연산량이 너무 많다는 문제점 발생(분모)

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

← A big sum over words

## 1주차 스터디 내용

**Negative Sampling:** SoftMax의 연산량 문제를 해결하기 위해 사용

(중심 단어, **실제 주변 단어**) 쌍이 진짜라는 것을 잘 맞히게,  
(중심 단어, **가짜 주변 단어**(negative sample)) 쌍이 가짜인 것을 잘 맞히게 학습  
보통 5~20개의 negative sample 사용

$$J_{neg-sample}(\mathbf{u}_o, \mathbf{v}_c, U) = - \log \sigma(\mathbf{u}_o^T \mathbf{v}_c) - \sum_{k \in \{K \text{ sampled indices}\}} \log \sigma(-\mathbf{u}_k^T \mathbf{v}_c)$$

/   
 sigmoid rather than softmax

즉 SoftMax 대신 시그모이드 함수를 활용하여  
이진분류 문제로 변환

-> 연산량을 획기적으로 감소시키면서 성능 유지



# 1주차 스터디 내용

## GloVe

단어의 동시 등장 확률을 벡터 내적을 통해 단어의 의미를 벡터공간에서 자연스럽게 표현  
Corpus 전체 단어쌍의 동시 출현 정보를 활용  
-> 윈도우(주변) 단어들만 보는 Word2vec과의 차이점

손실함수 **Loss:** 
$$J = \sum_{i,j=1}^V f(X_{ij}) (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

$w_i^T \tilde{w}_j$  -> 단어 임베딩 벡터의 내적, 이 값이  $\log X_{ij}$ 와 가까워지도록 학습

$\log X_{ij}$  -> 단어  $i,j$ 가 얼마나 자주 등장하는가에 대한 로그값

$f(X_{ij})$  -> 빈도에 따른 편향 완화를 위한 가중치 함수

## 다음 스터디 계획

03.24 18:00 대면 스터디

3강: Backpropagation and Neural Network Basics

4강: Dependency Parsing

관련 논문 리뷰:

A Fast and Accurate Dependency  
Parser using Neural Networks

감사합니다