Natural Language Processing with DeepLearning

week 1

1. The course
   - 딥러닝을 위한 효과적인 방법 이해
   - 인간의 언어에 대한 이해 및 생산에 있어서 어려움 이해
   - PyTorch로 대부분의 NLP 문제를 해결할 수 있는 능력

2.Human language and word meaning
   - 의미 : “Denotational semantics” - 의미를 사물이 나타내는 것으로 생각한다
   - How do we have usable meaning in a computer? => Wordnet (synonym sets, hypernyms)
   - Wordnet problems : missing nuance, missing new meanings, subjective, can’t compute accurate word similarity

<Representing words as discrete symbols>

1) One-hot vectors

ex) motel =[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]

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

Problems? 수학적으로는 orthogonal => 단어의 유사성 파악 X

Solution: try to rely on WordNet’s list -> fail, Instead: learn to encode similarity

2) Distributional semantics

ex) banking = {0.286, 0.792, -0.177, -0.107, 0.109, -0.542, 0.349, 0.271}

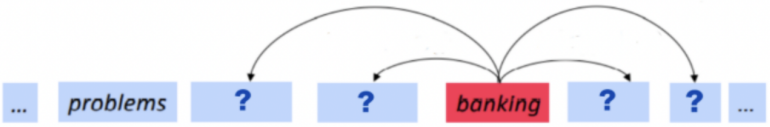
모든 숫자가 0이 아닌 벡터로, 의미를 숫자로 표현

=> Word Embedding(단어의 유사도 벡터화)

<유사한 문맥에서 나타나는 비슷한 형태의 벡터>

3) word2vec

: 워드 벡터 학습하기 위한 프레임워크 (center word 주어졌을때, 주변 맥락에 무슨 단어 들어가는지 추측)



3. Word2vec Intro

<Idea>

- large corpus of text
- every word in a fixed vocabulary is represented by vector
- text 내 position t를 지나면서 center words (c), context words (o)
- use the similarity of the word vectors & calculate the probability of o given c
- keep adjusting the word vectors to maximize this probability

4. Word2vec objective function gradients

Word2vec: objective function

For each position $t = 1, \ldots, T$, predict context words within a window of fixed size m, given center word w_j.

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \neq 0}} P(w_{t+j} \mid w_t; \theta)$$

θ is all variables to be optimized

sometimes called *cost* or *loss* function

The objective function $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} \log P(w_{t+j} \mid w_t; \theta)$$

Minimizing objective function ⟺ Maximizing predictive accuracy

(1) center word 주어졌을때, context word 예측하는 LikeliHood
(2) objective fuction : (1)에 negative log likelihood

<Question> : 아래 어떻게 계산?

$$P(W_{t+j}|W_t; \theta)$$

=> word vector : 단어 하나당 두 벡터의 평균 (w 가 center word일때, context word일때)

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

(내적: 유사도 계산, softmax의 꼴, 항상 Positive로 만들기 위해 exponential)

- loss 최소화하도록 파라미터 조정 => Optimization을 통해 objective function을 최소화하는 파라미터 θ, 즉 u와 v(word를 나타내는 두 벡터)를 찾음
- θ: 모델의 모든 파라미터, V개의 단어,

$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zcbra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

5. Optimization basics
(Gradient Descent), (Stochastic Gradient Descent)

6. Looking at word vectors