Negative Sampling

박주찬

DICE Lab
School of Computer Science and Engineering
KOREATECH
green261535@gmail.com

Introduction

Negative sampling은 Word2vec의 training방법중 하나이다. Negative sampling이 무엇인지 살펴보기 전에 먼저 Word2vec란 무엇인지 간단하게 알아보고, Word2vec의 두가지 방식인 CBOW, skip-gram에 대해서 간단히 알아보고, 마지막으로 단어가 벡터로 잘 바뀔 수 있도록 training시키는 방법중 하나인 Negative sampling에 대해서 자세히 다루겠다.

Word2vec란

Word2vec란 무엇인지 먼저 살펴 보자. 텍스트 기반의 모델을 만들고 input으로 텍스트를 넣어주기 위해서는 텍스트를 숫자 즉 벡터로 바꿔야한다. one-hot-encoding이라는 간단한 방법이 있었는데 이 방법의 단점은 벡터 표현에 단어와 단어 간의 관계가 전혀 드러나지 않는다는 점이다. 그래서 단어를 벡터로 바꿀 때, 벡터에 단어의 의미를 담을 수 있도록 단어를 벡터로 바꿔주는 모델을 단어 임베딩 모델 이라고 한다. Word2vec은 단어 임베딩 모델들 중 대표적인 모델이다.

CBOW, Skip-gram

CBOW모델은 Continuous Bag of Words의 약어로 context word(주변 단어)로 center word를 예측하는 방식이다. Context word란 보통 center word의 직전 몇 단어와 직후 몇 단어를 뜻한다. Context word(주변 단어)의 범위를 window라고

부르며 사이즈를 조정할 수 있다. Skip-gram모델은 CBOW와는 반대 방향의 모델이라고 볼 수 있다. Context word로 center word를 예측하는 방식이다.

Context word로 center word를 예측하는 CBOW에 비해 Skip-gram의 성능이 더 좋다. 성능이 더 좋은 이유는 CBOW일 경우 center word는 단 한번의 업데이트를 한다. 하지만 window size가 2인 Skip-gram의 경우에는 center word는 업데이트를 총 4번이나 한다. 그러므로 학습량에서 4배 차이가 나며 CBOW보다 Skip-gram이 성능이 더 좋아서 Word2vec을 수행할 때 skip-gram을 주로 사용한다.

Negative Sampling

먼저 Word2Vec의 학습과정을 살펴보자. Word2Vec의 Skip-gram은 아래 식을 최대화하는 방향으로 학습을 진행한다. 아래 식 좌변은 center word(c)가 주어졌을 때 context word(o)가 나타날 확률을 뜻한다. 우변의 v는 입력층-은닉층을 잇는 가중치 행렬W의 행벡터, u는 은닉층-출력층을 잇는 가중치 행렬W'의 열벡터 이다.

$$P(o|c) = \frac{\exp(u_o^T \mathrm{v_c})}{\sum_{w=1}^{W} \exp(u_w^T \mathrm{v_c})}$$

Word2vec은 출력층이 내놓은 스코어값에 softmax를 적용해 확률값으로 변환한 후 이를 정답과 비교해 backpopagation하는 구조이다. 하지만 위 식을 보면 분모에 해당하는 값, 즉 center word와 나머지 모든 단어의 내적을 한 뒤, 이를 다시 exp를 취해줘야 한다. 보통 전체 단어의 개수가 10만개라고 했을 때 위 식의 계산량은 어마어마해진다. 이 때문에 softmax 확률을 구할 때 전체 단어를 대상으로 구하지 않고, 일부 단어만 뽑아서 계산을 하게 된다. 이것이 바로 negative sampling이다.

Mikolov et al.은 “Distributied Representation of Words and Phrases and their Compositionality”라는 논문에서 Negative Sampling을 제안했다. Neagative sampling의 절차는 먼저 사용자가 지정한 window size내에 등장하지 않는 단어(neagative sample)을 5~20개 정도 뽑는다. 이를 정답 단어와 합쳐 전체 단어처럼 softmax 확률을 구하는 것이다. 이때 window내에 등장하지 않는 어떤 단어 w_i가 negative sample로 뽑힐 확률은 아래와 같이 정의한다.

$$P(\mathrm{w_i}) = \frac{f(\mathrm{w_i})^{3/4}}{\sum_{j=1}^{n} f(\mathrm{w_i})^{3/4}}$$

Center word와 context word의 쌍을 (w, c)라고 하자. 이 쌍이 corpus 데이터로부터 추출 되었을 확률을 $P(B = 1|w, c)$로 표기하자. (w, c)쌍이 corpus 데이터로부터 추출되지 않았을 확률을 $P(B = 0|w, c)$ 로 표기하자. $P(B = 1|w, c)$을 sigmoid function을 이용해서 모델링을 하면 다음과 같다.

$$P(B = 1|w, c, \theta) = \sigma(u_w^T \mathrm{v_c}) = \frac{1}{1 + \exp(-u_w^T \mathrm{v_c})}$$

Center word와 context word가 실제로 corpus data안에 있다면 corpus data에 있을 확률을 최대화 하고, center word와 context word가 실제로 corpus data안에 없다면 corpus data에 없을 확률을 최대화하는 새로운 objective function을 만들자. 이 두 확률에 대해 간단한 maximum likelihood 방법을 취한다. Corpus data안에 있는 (w,c)쌍의 집합을 D라고 하고, Corpus data안에 없는 (w,c)쌍의 집합을 $\tilde{D}$라고 하면 objective function은 아래와 같이 표기할 수 있다. ($\tilde{D}$은 negative sampling한 쌍의 집합)

$$
\begin{aligned}
J_t(\theta) &= argmax_\theta \prod_{(w,c)\in D} P(B=1|w,c,\theta) \prod_{(w,c)\in \tilde{D}} P(B=0|w,c,\theta) \\
&= argmax_\theta \prod_{(w,c)\in D} P(B=1|w,c,\theta) \prod_{(w,c)\in \tilde{D}} (1-P(B=1|w,c,\theta)) \\
&= argmax_\theta \sum_{(w,c)\in D} \log P(B=1|w,c,\theta) + \sum_{(w,c)\in \tilde{D}} (1-P(B=1|w,c,\theta)) \\
&= argmax_\theta \sum_{(w,c)\in D} \log \frac{1}{1+\exp(-u_w^T v_c)} + \sum_{(w,c)\in \tilde{D}} (1-\frac{1}{1+\exp(-u_w^T v_c)}) \\
&= argmax_\theta \sum_{(w,c)\in D} \log \frac{1}{1+\exp(-u_w^T v_c)} + \sum_{(w,c)\in \tilde{D}} (\frac{\exp(-u_w^T v_c)}{1+\exp(-u_w^T v_c)}) \\
&= argmax_\theta \sum_{(w,c)\in D} \log \frac{1}{1+\exp(-u_w^T v_c)} + \sum_{(w,c)\in \tilde{D}} (\frac{\frac{1}{\exp(u_w^T v_c)}}{\frac{\exp(u_w^T v_c)+1}{\exp(u_w^T v_c)}}) \\
&= argmax_\theta \sum_{(w,c)\in D} \log \frac{1}{1+\exp(-u_w^T v_c)} + \sum_{(w,c)\in \tilde{D}} (\frac{1}{1+\exp(u_w^T v_c)}) \\
&= \log \sigma(u_o^T v_c) + \sum_{j=1}^{k} \mathbb{E}_{j\sim P(w)} [\log \sigma(-u_j^T v_c)]
\end{aligned}
$$

$$
\begin{aligned}
J(\theta) &= \frac{1}{T} \sum_{t=1}^{T} J_t(\theta) \\
&= \frac{1}{T} \sum_{t=1}^{T} \left(\log \sigma(u_o^T v_c) + \sum_{j=1}^{k} \mathbb{E}_{j\sim P(w)} [\log \sigma(-u_j^T v_c)] \right)
\end{aligned}
$$

T는 V matrix의 총 행의 개수이다. $j\sim P(w)$의 의미는 negative sampling한다는 의미이다. J(θ)를 maximize 하도록 θ를 업데이트 하면 된다.

Conclusion

Word2vec를 training 할 때, negative sampling을 한다면 성능이 향상되고 계산량이 낮아지므로 좋은 training 기법이라고 할 수 있다.

References

1. Y. Goldberg, et al., word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method. https://brunch.co.kr/
2. https://ratsgo.github.io/
3. http://solarisailab.com/