RNN Introduction

Recap：word2vec

T개의 단어열로 나타난 corpus $(w_1, w_2, \cdots, w_T)$를 고려하자. CBOW는 contexts로부터 target을 추론한다. 즉, (window size가 1이라면) 분포 $\mathcal{P}(W_T|W_{t-1}, W_{t+1})$를 모델링한다. Window를 좌우 대칭이 아닌 왼쪽으로만 한정한다면 CBOW는 분포 $\mathcal{P}(W_t|W_{t-2}, W_{t-1})$를 모델링할 것이다. 다시 말해, size N의 batch에 대해 (SL 모델인) CBOW의 objective function은

$$\mathbf{L} = \frac{1}{N}\sum_{t=1}^{N} -\log p_\theta(w_t|w_{t-2}, w_{t-1})$$

이다.

학습을 통해 CBOW는 context로부터 target을 정확하게 추측하도록 θ를 최적화시킨다. 이때 CBOW가 word2vec 모델인 이유는 학습 과정에서 부산물로 단어의 의미가 encoding된 embedding vector인 단어의 분산 표현을 얻게 되기 때문이다.

Language Model

Context로부터 target을 추측한다는 CBOW의 작동은 언어 모델에 활용될 수 있다.

언어 모델은 단어의 나열에 확률을 부여하는 모델이다. 특정한 단어 sequence가 일어날 가능성이 어느 정도인지 — i.e., 얼마나 자연스러운 순서로 배열된 단어 sequence인지 — 를 출력하는 것이다.

그러므로 언어 모델은 새로운 문장을 생성하는 용도로도 이용할 수 있다. 언어 모델은 단어 sequence에 관한 분포 — 단어 순서의 자연스러움을 확률로 표현한다 — 이므로, 그러한 분포에서의 sampling으로 다음에 등장할 적절한 단어를 생성할 수 있기 때문이다.

길이 m의 단어 sequence $(w_1, \cdots, w_m)$을 고려하자. 결합 확률인 단어 sequence의 발생 확률 $\mathrm{P}(w_1, \cdots, w_m)$을 chain rule로 factorize하면

$$\mathrm{P}(w_1, \cdots, w_m) = \prod_{t=1}^{m} \mathrm{P}(w_t|w_1, \cdots, w_{t-1}) \quad (\mathrm{P}(w_1) \coloneqq \mathrm{P}(w_1|w_0))$$

이다. 고로 궁극적으로 모델링할 목표 분포 — i.e., 언어 모델 — 는 이전에 발생한 단어 sequence를 조건으로 하는 분포 $\mathcal{P}(W_T|W_1, W_2, \cdots, W_{t-1})$이다. 요 조건부 분포를 규명하면 전체 단어 sequence의 발생 확률 $\mathrm{P}(w_1, \cdots, w_m)$을 계산할 수 있으니까.

cf. 언어 모델이라 함은 원칙적으로 단어 sequence의 분포 $\mathcal{P}(W_1, \cdots, W_m)$를 말하지만, 위의 조건부 분포 $\mathcal{P}(W_T|W_1, W_2, \cdots, W_{t-1})$를 언어 모델이라 부르는 것도 일반적이다.

word2vec as Language Model

word2vec 모델인 CBOW로 언어 모델을 규명(학습)하려면, 특정 개수의 왼쪽 단어를 context로 하여 근사:

$$\begin{aligned} p(w_t|w_1, w_2, \cdots, w_{t-1}) &= \prod_{t=1}^{m} p(w_t|w_1, \cdots, w_{t-1}) \\ &\approx \prod_{t=1}^{m} p(w_t|w_{t-2}, w_{t-1}) \end{aligned}$$

하면 된다. 다시 말해, CBOW를 언어 모델로서 사용한다는 것은 (temporal process인 단어 sequence의 분포인) 언어 모델을 직전 n 개 — 여기서는 2개 — 의 timestep에 발생한 사건에만 의존하는 n-state markov chain으로 근사시켜 나타냄을 의미한다.

보다 나은 언어 모델을 얻기 위해 context가 커버하지 못하는 — i.e., 조건부에 포함되지 못한 — 앞쪽 단어들을 최소화하기 위해 window size를 키워 context에 포함되는 단어의 개수를 늘릴 수 있을 것이다. 그렇다면 단어 sequence의 자연스러움을 더 정확하게 평가할 수 있을 테니까.

그러나 CBOW는 context 속 단어들의 순서를 무시하므로, context의 길이를 늘리면 언어 모델로써의 성능은 오히려 감소할 가능성이 매우 높다.

cf. CBOW는 continous bag-of-words로, 말 그대로 단어들이 담긴 bag를 이용하는 모델이다. Bag 속 단어들에는 순서가 없으며 (순서 대신) 단어들의 bag, 즉 분포를 사용한다.

CBOW의 구조를 보면 context 속 단어들의 순서가 무시됨을 곧바로 확인할 수 있다. Batch $\begin{bmatrix} \mathrm{you}(1)\ \mathrm{say}(3) \\ \mathrm{say}(3)\ \mathrm{you}(1) \\ \vdots \end{bmatrix}_{[N,2]}$ 를 예로 들면, w_{in}:

$$w_{\mathrm{in}} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \end{bmatrix}_{[V,H]}$$

을 갖는 embedding layer로부터 각각 $\begin{bmatrix} \mathrm{you}(1) \\ \mathrm{say}(3) \\ \vdots \end{bmatrix}_{[N,]}$ 와 $\begin{bmatrix} \mathrm{say}(3) \\ \mathrm{you}(1) \\ \vdots \end{bmatrix}_{[N,]}$ 에 대응하는 $\begin{bmatrix} w_1 \\ w_3 \\ \vdots \end{bmatrix}_{[N,H]}$ 와 $\begin{bmatrix} w_3 \\ w_1 \\ \vdots \end{bmatrix}_{[N,H]}$ 를 얻고 둘을 더해 h:

$$h = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} w_2 \\ w_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} w_1 + w_2 \\ w_2 + w_1 \\ \vdots \end{bmatrix}_{[N,H]}$$

를 얻을 것이다. 고로 context $(\mathrm{you}, \mathrm{say})$와 $(\mathrm{say}, \mathrm{you})$는 모델 속에서 똑같이 다루어진다.

cf. 이 문제에 대처하기 위해 context vector들을 concatenate해 h를 구성할 수도 있지만, 그렇다면 커진 h를 처리하기 위해 w_{out}은 매우 커져야 할 것이다.

그러므로 CBOW를 언어 모델로 사용하는 것은 적절하지 않다.

cf. word2vec은 단어의 분산 표현을 얻을 목적으로 고안된 모델이므로 이를 언어 모델로 사용하는 경우는 드물다. RNN류의 모델을 이용한 언어 모델에서도 단어의 분산 표현을 얻을 순 있지만, 단어 수 증가에 대응하고 분산 표현의 질 개선을 위해 word2vec이 제안되었다.

RNN은 context가 아무리 길더라도 context의 정보를 기억하는 mechanism을 가진다. 따라서 RNN을 사용하면 아무리 긴 temporal(time series) data라 할지라도 대응할 수 있다.

cf. #TODO: 확인 필요