Word2Vec

Introduction

- **기존의 문제점:** 단어를 one hot encoding → 단어 간 유사도 판단 불가, vocab size 가 클수록 vector의 차원이 증가 + sparse matrix 문제
- **Skip-gram + CBOW** → 연산량을 최소화하여 분산 표현 학습
- **N-gram**: Count 기반, 이때 일부 몇개 단어를 결정하는지 → n

  if) 4-gram: n-1 에 해당하는 앞의 3개의 단어만 고려
  - 이때의 문제점 → sparsity, n을 선택해야 하는 문제
- **NNLM** (Neuarl Network Net Language Model)
  - RNNLM

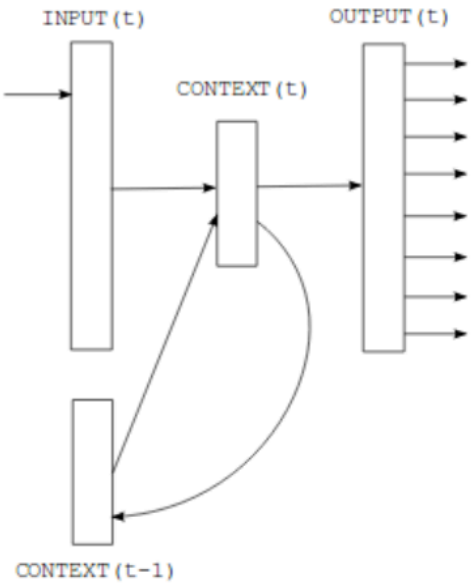


Figure 1: *Simple recurrent neural network.*

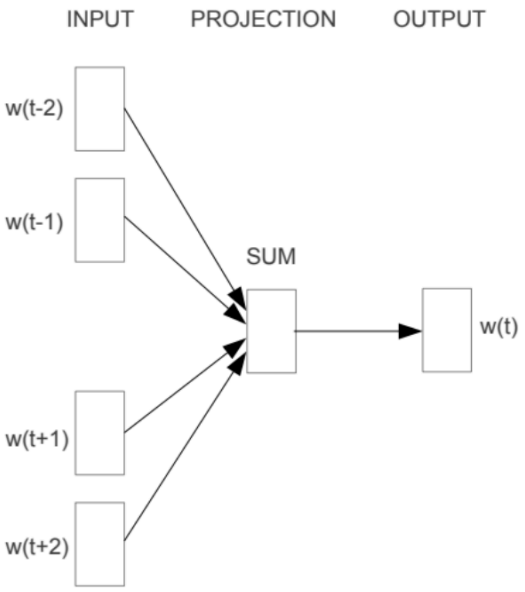
  - 이때의 문제점 → 고정된 window size

- 따라서 기존의 문제점은 Sparsity + 고연산 비용 → 대규모 corpus로부터 단어의 분산 표현(Distributed Representation)을 학습하는 **Word2Vec** 제안

Model Architecture

Continuous Bag-of Words Models(CBoW)

: 문맥(주변 단어들)을 보고 중심 단어를 예측하는 모델



CBOW

- **기존의 NNLM과 유사 but Hidden layer가 제거됨 → 연산량이 적어짐**
- 또한 기존의 NNLM과 달리 임베딩 행렬의 연결이 아닌 평균으로 → 모든 단어가 같은 위치에 projection 될 수 있도록 (**모든 단어는 동일한 행렬으로 projection되는 것)**

$$Q = N \times D + D \times log_2(V).$$

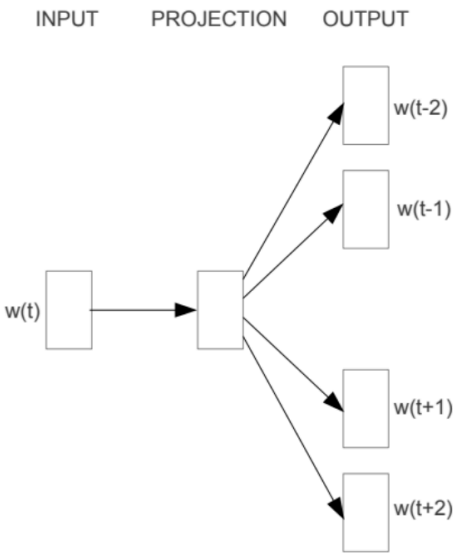
이때의 훈련 복잡도로 O(logV)가 됨

- 기존의 softmax가 아닌 Hierarchical Softmax로 계산량이 V→ logV가 됨

Skip-gram

: 중심 단어를 보고 문맥(주변 단어들)을 예측하는 모델

- 예측하는 주변 단어의 범위를 늘릴 수록 연산량이 증가함 → 현재 단어로부터 거리가 있는 단어들에 더 sampling 빈도를 낮춰 가중치를 줄임



Skip-gram

- 훈련 복잡도의 C는 예측을 위한 단어의 최대거리로 단어의 앞뒤에서 무작위로 R개를 선택해서 사용함

Results

- Word2Vec은 기존 모델보다 훨씬 빠른 속도로 학습 가능
- Word2Vec으로 학습된 단어 임베딩이 실제 단어 간 의미적 관계를 얼마나 잘 포착하는지 평가하기 위해 단어 유사도를 측정(cosine 유사도로 실험) 하는 실험을 진행
- Word2Vec의 또 다른 강력한 특징은 **벡터 연산을 통해 단어 관계를 추론할 수 있다는 점**

Future

- Word2Vec은 **단어의 다의성을 처리하지 못함**
- **문맥을 직접 반영**하지 못함
    - 단어 간 관계를 학습하지만, 전체 문맥을 반영하는 모델이 아님

→ **GloVe**: Co-occurrence matrix 기반으로 단어 간 관계를 더 잘 학습하도록 개선

→ **FastText**: Subword 정보를 활용하여 OOV 문제 (out of vocan)를 해결