chapter 3 - word2vec

- 앞 장에서는 [통계 기반 기법]으로 단어의 분산 표현을 얻었는데, 이번 장에서는 더 강력한 기법인 [추론 기반 기법]을 살펴보겠습니다.

3.1 추론 기반 기법과 신경망

3.1.1 통계 기반 기법의 문제점

- [통계 기반 기법]에서는 주변 단어의 빈도를 기초로 하는 단어를 표현했습니다. 구체적으로는 단어의 동시발생 행렬을 만들고 그 행렬에 SVD를 적용하여 밀집벡터를 얻었습니다. 그러나 이 방식은 대규모 말뭉치를 다룰 때 문제가 발생합니다.
- 한편, [추론 기반 기법]에서는, 예컨대 신경망을 이용하는 경우에는 미니배치로 학습하는 것이 일반적입니다. 미니배치 학습에서는 신경망이 한번에 소량(미니배치)의 학습 샘플씩 반복해서 학습하며 가중치를 갱신해갑니다.

3.1.2 추론 기반 기법 개요

- 추론 기반 기법에서는 당연히 '추론'이 주된 작업입니다. 추론이란 [그림 3-2]처럼 주변 단어(맥락)가 주어졌을 때 '?'에 무슨 단어가 들어가는지를 추측하는 작업입니다.
- 이러한 추론 문제를 반복해서 풀면서 단어의 출현 패턴을 학습하는 것이죠.
- 모델은 맥락 정보를 입력받아 (출현할 수 있는) 각 단어의 출현 확률을 출력합니다. 이러한 틀 안에서 말뭉치를 사용해 모델이 올바른 추측을 내놓도록 학습시킵니다. 그리고 그 학습의 결과로 단어의 분산 표현을 얻는 것이 추론 기반 기법의 전체 그림입니다.

3.1.3 신경망에서의 단어 처리

- 우선, 단어를 '고정 길이의 벡터'로 변환합니다.
- 이때 사용하는 대표적인 기법은 [원핫 표현(또는 원핫 인코딩)]입니다.
- 단어를 벡터로 표현하고, 신경망을 구성하는 계층들은 벡터를 처리할 수 있습니다.
- 이 화살표에는 가중치(매개변수)가 존재하여, 입력층 뉴런과의 가중합이 은닉층 뉴런이 됩니다.
- 그래서 이 책의 경우, 완전연결계층은 1장에서 구현한 MatMul 계층과 같아지죠. 참고로 딥런이 프레임워크들은 일반적으로 완전연결계층을 생성할 때 편향을 이용할지 선택할 수 있도록 제시합니다.

3.2 단순한 word2vec

- 그리고 이번 절에서 사용할 신경망은 word2vec에서 제안하는 CBOW(continuous bag of words) 모델입니다/

3.2.1 CBOW 모델의 추론 처리

- CBOW 모델은 맥락으로부터 타깃을 추측하는 용도의 신경망입니다.
- CBOW 모델의 입력은 맥락입니다. 이 맥락을 원핫 표현으로 변환하여 CBOW 모델이 처리할 수 있도록 준비합니다.

- 이 그림에서 입력층이 2개인 이유는 맥락으로 고려할 단어를 2개로 정했기 때문입니다. 즉, 맥락에 포함시킬 단어가 N개라면 입력층도 N개가 됩니다.
- 은닉층의 뉴런은 입력층의 완전연결계층에 의해 변환된 값이 되는데, 입력층이 여러 개이면 전체를 '평균'하면 됩니다.
- 그리고 출력층 뉴런은 각 단어의 '점수'를 뜻하며, 값이 높을수록 대응 단어의 출현 확률도 높아집니다.
- 점수를 Softmax 계층에 통합시킨 후의 뉴런을 '출력층'이라고도 합니다.
- (미리 밝히자면) 이 가중치가 바로 단어의 분산 표현의 정체입니다.
- 따라서 학습을 진행할수록 맥락에서 출현하는 단어를 잘 추측하는 방향으로 이 분산표현들이 갱신될 것입니다.
- 은닉층의 뉴런 수를 입력층의 뉴런 수보다 적게하는 것이 중요한 핵심입니다. 은닉층에는 단어 예측에 필요한 정보를 '간결하게' 담게 되며, 결과적으로 밀집벡터 표현을 얻을 수 있습니다. 이때 그 은닉층의 정보는 우리 인간은 이해할 수 없는 코드로 쓰여 있습니다. 바로 [인코딩]에 해당하죠. 한편, 은닉층의 정보로부터 원하는 결과를 얻는 작업은 [디코딩]이라고 합니다.
- CBOW 모델은 활성화 함수를 사용하지 않는 간단한 구성의 신경망입니다. 입력층이 여러 개 있고, 그 입력층들이 가중치를 공유한다는 점을 제외하면 어려운 부분은 없습니다.

3.2.2 CBOW 모델의 학습

- 단어에 소프트맥스 함수를 적용하면, '확률'을 얻을 수 있었는데요, 이 확률은 맥락(전후 단어)이 주어졌을 때 그 중앙에 어떤 단어가 출현하는지를 나타냅니다.
- 이때 '가중치가 적절히 설정된' 신경망이라면 '확률'을 나타내는 뉴런들 중 정답에 해당하는 뉴런의 값이 클 것이라 기대할 수 있습니다.
- 우리가 다루고 있는 모델은 다중 클래스 분류를 수행하는 신경망입니다. 따라서 이 신경망을 학습하려면 소프트맥스와 교차 엔트로피 오차만 이용하면 됩니다. 여기에서는 소프트맥스 함수를 이용해 점수를 확률로 변환하고, 그 확률과 정답 레이블로부터 교차 엔트로피 오차를 구한 후, 그 값을 손실로 사용해 학습을 진행합니다.
- 덧붙여 [그림 3-13]에서는 Softmax 계층과 Cross Entropy Error 계층을 사용했습니다만, 우리는 이 두 계층을 Softmax with Loss라는 하나의 계층으로 구현할 겁니다. 따라서 우리가 앞으로 구현할 신경망의 정확한 모습은 [그림 3-14]처럼 생겼습니다.

3.2.3 word2vec의 가중치와 분산 표현

- 입력 측 가중치 $W_{\text{인}}$의 각 행이 각 단어의 분산 표현에 해당합니다.
- 또한 출력 층 가중치 $W_{\text{아웃}}$에도 단어의 의미가 인코딩된 벡터가 저장되고 있다고 생각할 수 있습니다. 다만, 출력 측 가중치는 [그림 3-15]에서 보듯 각 단어의 분산 표현이 열 방향(수직 방향)으로 저장됩니다.
- 그러면 최종적으로 이용하는 단어의 분산 표현으로는 어느 쪽 가중치를 선택하면 좋을까요?

1) A : 입력 측의 가중치만 이용한다. 2) 출력 측의 가중치만 이용한다. 3) 양쪽 가중치를 모두 이용

- word2vec(특히 skip-gram 모델)에서는 A 안의 '입력 측의 가중치만 이용한다'가 가장 대중적인

선택입니다. 많은 연구에서 출력 측 가중치는 버리고 입력 측 가중치 $W_{\text{입}}$만을 최종 단어의 분산 표현으로서 이용합니다.

3.3 학습 데이터 준비

3.3.1 맥락과 타깃

- word2vec에서 이용하는 신경망의 입력은 [맥락]입니다.
- 그리고 그 정답 레이블은 맥락에 둘러싸인 중앙의 단어, 즉 '타깃'입니다.
- 다시 말해, 우리가 해야 할 일은 신경망에 '맥락'을 입력했을 때 '타깃'이 출현할 확률을 높이는 것.
- 그러기 위해선, 우선, 말뭉치 텍스트를 단어 ID로 변환해야 합니다.
- ID의 배열인 [corpus]로부터 맥락과 타깃을 만들어냅니다.

3.3.2 원핫 표현으로 변환

3.5 word2vec 보충

- 이번 절에서는 지금까지 말하지 못한 word2vec에 관한 중요한 주제 몇 개를 보충하겠습니다.
- 우선은 CBOW 모델을 '확률' 관점에서 다시 살펴보겠습니다.

3.5.1 CBOW 모델과 확률

- CBOW 모델의 학습이 수행하는 일은 이 손실 함수의 값을 가능한 한 작게 만드는 것입니다.
- 그리고 이때의 가중치 매개변수가 우리가 얻고자 하는 단어의 분산 표현인 것입니다.

3.5.2 skip - gram 모델

- skip-gram은 CBOW에서 다루는 맥락과 타깃을 역전시킨 모델입니다.
- CBOW 모델은 맥락이 여러 개 있고, 그 여러 맥락으로부터 중앙의 단어(타깃)을 추측합니다.
- 한편, skip-gram 모델은 중앙의 단어(타깃)으로부터 주변의 여러 단어(맥락)을 추측합니다.
- 보다시피, skip-gram 모델의 입력층은 하나입니다. 한편 출력층은 맥락의 수만큼 존재합니다.
- 따라서 각 출력층에서는 (Softmax with Loss 계층 등을 이용해) 개별적으로 손실을 구하고, 이 개별 손실들을 모두 더한 값을 최종 손실로 합니다.
- 여기서 skip-gram 모델에서는 맥락의 단어들 사이에 관련성이 없다고 가정하고 다음과 같이 분해합니다. (정확하게는 '조건부 독립'이라고 가정합니다.)
- 그럼 CBOW 모델과 skip-gram 모델 중 어느 것을 사용해야 할까요? 그 대답은 skip-gram 모델이라고 할 수 있습니다. 단어 분산 표현의 정밀도 면에서 skip-gram 모델의 결과가 더 좋은 경우가 많기 때문이죠. 특히 말뭉치가 커질수록 저빈도 단어나 유추 문제의 성능 면에서 skip-gram 모델이 더 뛰어난 경향이 있습니다. 반면, 학습 속도 면에서는 CBOW 모델이 더 빠릅니다.
- 이런 점에서 skp-gram 모델 쪽이 '더 어려운' 문제에 도전한다고 말할 수 있습니다. 그리고 더 어려운 상황에서 단련하는 만큼 skip-gram 모델이 내어 주는 단어의 분산 표현이 더 뛰어날 가능성

이 더 커지는 것입니다.

3.5.3 통계 기반 vs 추론 기반

- [통계 기반 기법]은 말뭉치의 전체 통계로부터 1회 학습하여 단어의 분산 표현을 얻었습니다.
- [추론 기반 기법]에서는 말뭉치를 일부분씩 여러 번 보면서 학습했습니다. (미니배치 학습)
- 어휘에 추가할 새 단어가 생겨서 단어의 분산 표현을 갱신해야 하는 상황을 생각해봅시다. [통계 기반 기법]에서는 계산을 처음부터 다시 해야 합니다. 그에 반해 [추론 기반 기법]은 매개변수를 다시 학습할 수 있습니다. 이런 특성 덕분에 기존에 학습한 경험을 해치지 않으면서 단어의 분산 표현을 효율적으로 갱신할 수 있습니다.
- 두 기법으로 얻는 단어의 분산 표현의 성격이나 정밀도 면에서는 어떨까요? 분산 표현의 성격에 대해 논하자면, [통계 기반 기법]에서는 주로 단어의 유사성이 인코딩됩니다. 한편 word2vec(특히 skip-gram 모델)에서는 단어의 유사성은 물론, 한층 복잡한 단어 사이의 패턴까지도 파악되어 인코딩되죠.
- 이런 이유로 추론 기반 기법이 통계 기반 기법보다 정확하다고 흔히들 오해하곤 합니다. 그러나 실제로 단어의 유사성을 정량 평가해본 결과, 의외로 추론 기반과 통계 기반 기법의 우열을 가릴 수 없었다고 합니다.
- 또 다른 중요한 사실로, 추론 기반 기법과 통계 기반 기법은 서로 관련되어 있다고 합니다. 구체적으로 skip-gram과 네거티브 샘플링을 이용한 모델은 모두 말뭉치 전체의 동시발생 행렬에 특수한 행렬 분해를 적용한 것과 같습니다. 다시 말해, 두 세계는 '서로 연결되어 있다'고 할 수 있습니다.