인공지능기초와활용

**Project #2 Natural Language Processing**

|  |  |
| --- | --- |
| 담당교수 | 이종환 |
| 학번 | 2018930012 |
| 학과 | 전자전기컴퓨터공학부 |
| 이름 | 박수영 |
| 제출일자 | 2022.06.14 |

**Competition**

자연어 처리 경진대회 보고서

|  |
| --- |
| 목차 |

1. 소개

2. 수업요약

3. 프로젝트 알고리즘 설계과정

4. 학습기록

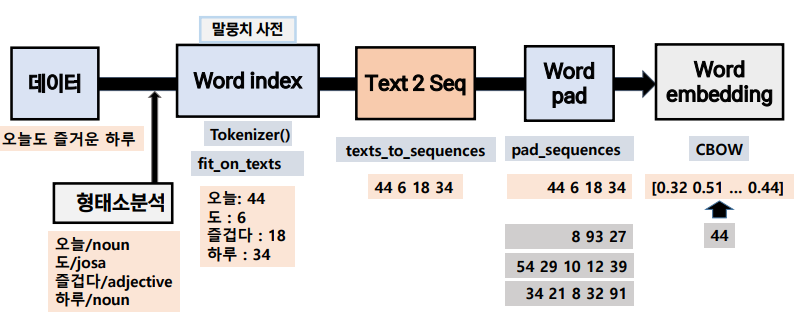
5. 참고문헌

1. 소개

본 보고서는 서울시립대학교 2022-1학기에 개설된 공학소양 교양교과목 ‘인공지능 기초와 활용’의 자연어 처리 경진대회 실험보고서이다.

1. 수업요약

* 자연어 처리: Natural language processing 은 머신러닝을 사용해 텍스트와 데이터를 처리 및 해석하는 기법입니다. 이 기법은 구조화 되지 않은 텍스트 기반의 데이터에서 유용한 정보를 파악하기 위해 사용됩니다.



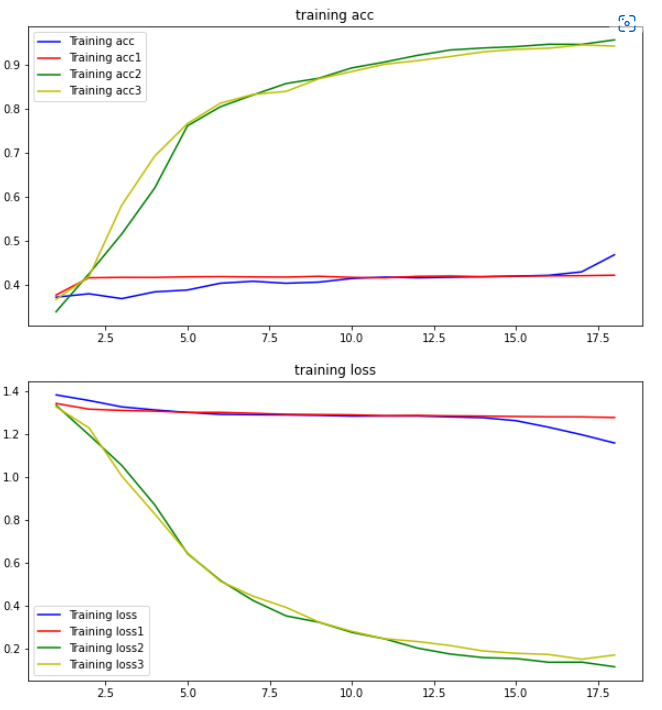
* Word2Vec: 단어를 vector로 표현하여 단어 사이의 유사성과 차이점을 계산한다. 계산된 결과를 바탕으로 그 주변단어(context word)와의 관계를 통해 단어가 표현(representation)하는 바를 직접적으로 예측한다.
* One-Hot Encoding: 각 단어에 고유한 인덱스를 부여하는 기법 결과
* Embedding 벡터: One-hot의 단점을 해소하기 위해 단어의 의미를 고려해 더 밀집된 차원의 벡터로 표현한다.
* CBOW: Continuous Bag Of Word 주변 단어를 이용해 중심 단어를 예측하는 방법이다.
* Skip-Gram: 중심 단어를 이용해 주변 단어를 예측하는 기법이다.
* Model
  + - RNN: Recurrent Neural Networks 은닉층 노드가 방향을 가진 엣지로 연결되어 순환하는 구조를 이루는 인공 신경망이다.
    - LSTM: Long Short Term Memory RNN모델에서 정보와 그 정보를 사용하는 지점의 거리가 멀 경우 학습 능력이 저하되는 문제가 있어 이를 보완하기 위해 고안되었다. RNN모델에 cell을 추가한다. input gate, forget gate, output gate, update gate를 사용해 정보가 흐를지, 버릴지, 저장할지, 업데이트 할지를 결정한다.
    - GRU: LSTM과 유사하며, cell을 단순화시킨 형태로 저사양에서 높은 효율을 보인다.

3. 프로젝트 알고리즘

1. **데이터 분석:**  자연어 처리를 위한 자료로 teat와 train 데이터가 csv파일로 구분되어 있다. Train 데이터는 ID, TITLE, ABSTRACT, label로 구성되며 label은 0,1,2,3 4개로 분류된다. Train 데이터의 크기는 15472개 이다. Test 데이터는 ID, TITLE, ABSTRACT로 구성되며, 데이터의 크기는 4844개이다.
2. **데이터 불러오가:** /input/nlp-sci-topic에 있는 train, test csv 파일을 불러와 train 과 test 변수에 저장했다. 각 변수의 TITLE, ABSTRACT 부분을 분리해 train\_sum\_data, test\_sum\_data에 저장한 뒤, 병합하였다.
3. **Toknizer:** 낱말분석 fit\_on\_texts를 사용해 낱말의 빈도를 분석합니다.
4. **Text\_to\_sequence:** 텍스트 안의 단어들을 숫자의 시퀀스의 형태로 변환한다.
5. **pad\_sequences:** 앞서 변환된 시퀸스를 숫자 0을 이용해서 같은 길이의 시퀀스로 변환한다. ( NumPy 어레이로 변환)
6. **데이터 분리 train test:** padding을 한 sequence들을 train과 test로 나눈다.
7. **Target 설정:** train 데이터의 label을 target으로 설정
8. **모델 설정:** Bidirectional LSTM.Word embedding: 입력 길이를 200으로 정하고 출력의 차원을 300으로 한다. Bidirectional 을 사용해 재귀적 신경모델 LTSM을 감싸준다. Dense층으로 Layer을 구성한다. 출력은 4개의 매개변수를 softmax로 출력한다.
9. **모델 컴파일**: 최적화를 RMSProp을 사용하였다.
10. **모델 fit:** train 데이터를 epochs=20, BATCH\_SIZE=8000에서 fitting한다.
11. **시각화:** matplotlib를 사용해 loss, val\_loss, acc, val\_acc를 시각화한다.
12. **내보내기:**  model에 test 데이터를 predict한값이 2차원 array로 저장되는데 np.max()를 사용해 각 array의 최대값에 해당하는 label을 출력용 sub.csv에 저장한다.

4. 학습기록

(1) SimpleRNN, LSTM, Bidirectional, GRU 4개 모델의 성능 비교



4개의 모델을 모두 사용한 결과 bidirectional LTSM과 GRU의 성능이 우수하였다. Bidiretional LSTM모델을 사용해 실습 모델을 설계했다.

(2) 실습모델 설계

Train data에 대해서는 0.98의 높은 정확도를 얻지만, test data에서는 0.77 정도의 정확도를 얻었다. Bidirectional LSTM층을 추가하고, batch\_size를 늘려 8.1131의 정확도를 얻었다.

5. 참고문헌

1. 인공지능 기초와 활용 수업자료
2. 사이토 고키, “밑바닥부터 시작하는 딥러닝”