**# Sentence Classification with CNN**

## 2013250422 김도향

1. **실험 환경**

* Google Colab(No GPU)

1. **실험 과정**
2. Preprocessing

* Text파일에 대한 UTF-8 encoding을 진행한다.
* Positive/Negative text file 각각을 NLTK를 사용해 문장별로 분리한 후, 중복 제거 및 Class label를 각 문장에 부여하고, positive/negative로 분리되어있던 두 sentence/class\_label list를 통합했다.
* 모든 문장 및 단어에 대한 indexing 작업을 진행해준다.

1. Sampling

* 편의를 위해 Training dataset을 10000개의 문장으로, 나머지를 Test dataset으로 배분하여 random하게 shuffle해준다. (9:1에 근접)

1. Hyper parameter configurations

* 제시된 조건에 따른 하이퍼 파라미터를 설정한다.

1. Define CNN model

* 각 MODE 설정에 따라 Embedding Vector 설정을 다르게 적용되도록 설정한다.
* rand MODE가 아니라면 제시된 pre-trained Vector(GoogleNews)를 사용한다.
* Convolution - ReLU - Pooling - Dropout을 거쳐 output를 return한다.

1. Define Training session and Test session

* 제시된 조건에 따라 Adam optimizer를 설정하고, learning rate, weight decay를 기입한다.
* 각 epoch마다 아래 과정을 반복한다.
  + 전체 Training dataset 중 batch\_size(=50)만큼씩 한 번에 학습한다.
  + 이 때, 문장의 최대 단어 개수를 설정해주고, 이보다 부족한 단어 수를 가진 문장에는 패딩 인덱스를 끼워넣고, 단어 수가 오히려 더 많은 경우는 그 이상의 단어는 문장 내에서 제외한다.
  + softmax를 거치며 예측된 50개 문장에 대한 scores값에 대해 softmax를 진행하고, 클래스를 예측한다.
  + 실제 라벨과 비교하여 accuracy를 측정하고, CrossEntropyLoss로 loss를 측정해 Backpropagation을 진행한다.
* 모든 epoch가 종료되면, 미리 준비해둔 Test dataset에 대해서 테스트를 진행한다. 상세 과정은 학습과 유사하나, batch size가 설정되어있지 않으므로 전체 Test dataset에 대해 진행하고 일괄적으로 정확도를 측정한다.

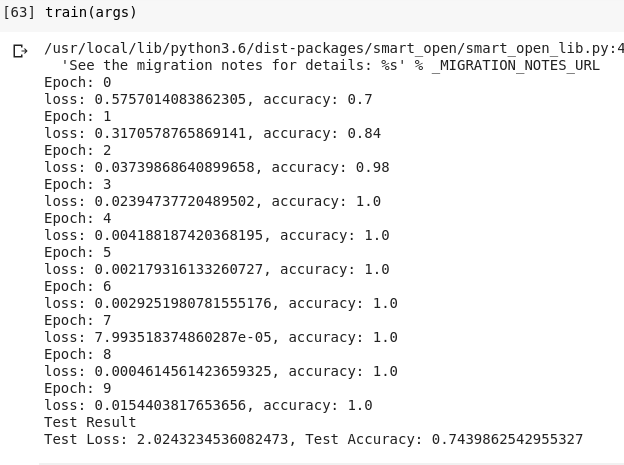
1. Test(CNN-rand/CNN-static/CNN-non-static/CNN-multichannel)

* 위 과정을 MODE를 바꿔주면서 진행하고, 값을 비교한다.

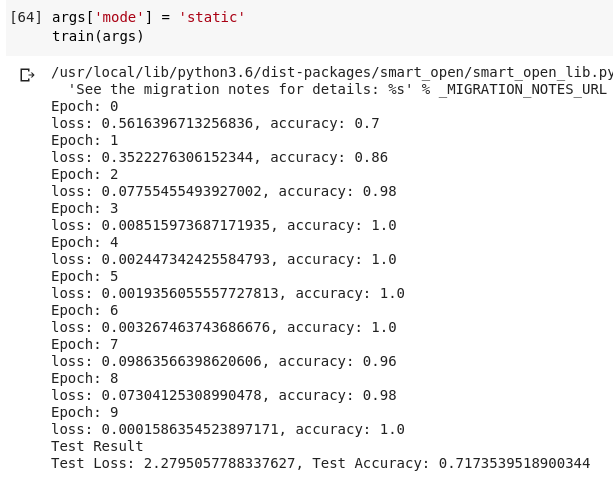
1. **실험 결과**

실험 결과는 다음과 같았다.

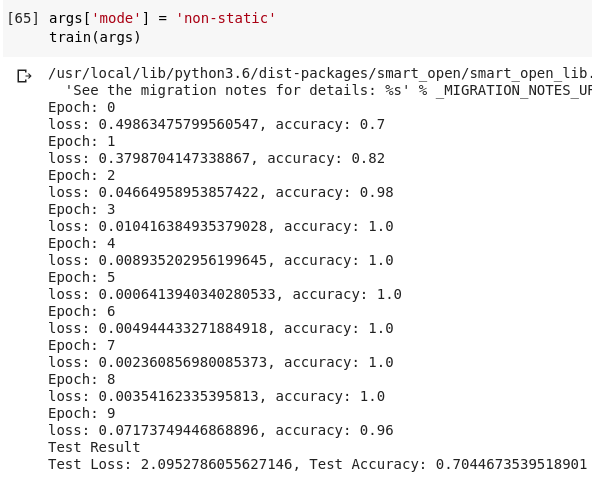
1. CNN-RAND



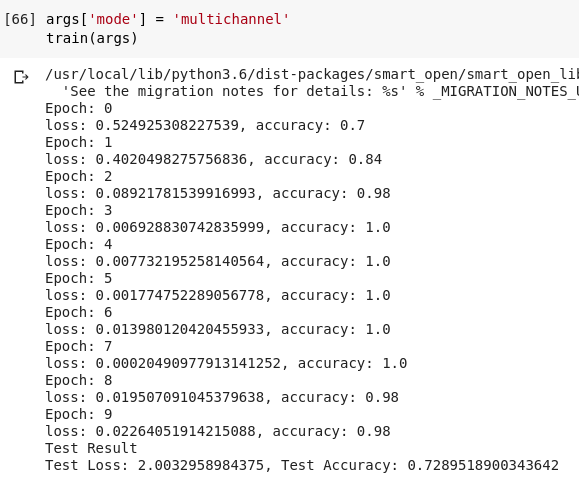
1. CNN-STATIC



1. CNN-NON-STATIC



1. CNN-MULTICHANNEL



거의 대부분의 모델들이 초기 epoch마다 급격한 예측력 상승을 보여주었다. 대체로 3 epoch에서 이미 100%에 근접한 예측력을 보이도록 학습데이터에 적응하였다.

그러나 해당 학습 데이터 예측력에 비해서 Test Accuracy는 낮은 상태임을 알 수 있다.

이를 표로 정리하면 다음과 같이 정리할 수 있다.

|  |  |
| --- | --- |
| Model | Accuracy |
| CNN-rand | 74% |
| CNN-static | 71% |
| CNN-non-static | 70% |
| CNN-multichannel | 72% |

Yoon Kim의 Convolutional Neural Network for Sentence Classification 논문에 따르면, 이들 중 절대적으로 좋은 것으로 판별된 모델은 존재하지 않았으며, 어떤 데이터셋을 사용하는지에 따라서 결과가 계속해서 달라지는 모습을 보였다.

그러나 해당 논문의 실험과 달리, 비교적 가장 낮은 예측도를 보였던 CNN-rand 모델이 본 실험에서는 가장 높은 정확도를 보였다는 점에서 차이가 나타났다.

다만 이 또한 극명한 차이를 보인 것은 아니며, 본 실험이 비교적 작은 크기의 데이터셋을 사용했고, 테스트셋의 설정과 실험 과정 또한 지나치게 간소화된 형태로 샘플링을 사용했다는 점 등을 고려했을때 나타날 수 있는 차이라고 판단된다.

아울러 본 실험에서 L2 constraints에 대한 설정이 미흡했던 것을 고려할 때, 해당 파라미터의 Configuration을 조정하면 어느정도 향상된 결과를 기대할 수 있을 것이다. 또한 본 실험에서 사용한 각 모델(Static, Non-static, Multichannel)에 대한 이해가 부족할 수 있으며, pytorch 프레임워크 숙련도가 낮아 코드 작성 결과가 제대로 반영이 안되었거나, 혹은 비효율적인 학습 과정 등이 포함되어 있을 수 있다는 점이 낮은 예측력의 이유로 제시될 수 있을 것으로 보인다.