

딥러닝 개론 PA 2

20201133 윤현서

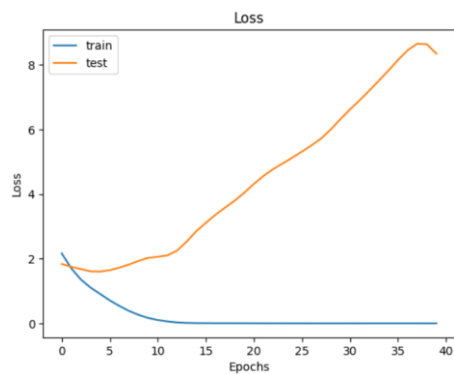
1. Result

(a) RNN+SGD+50d

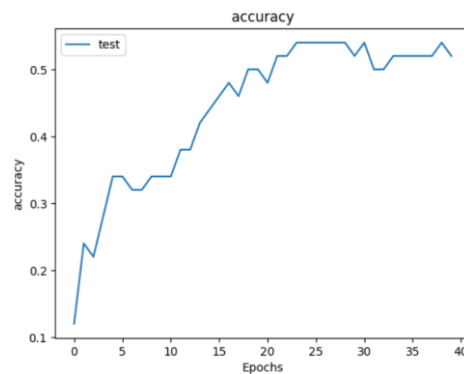
Final Training Loss: 0.0004

Final Test Loss: 8.3475, Final Test Accuracy: 52.00%

(1) Loss graph



(2) Accuracy graph



(3)all emojis for test set

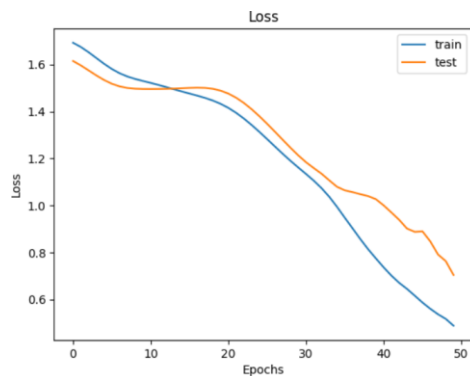
정답(1~26): 🍷😞😞😞😞😞😞😞🍷😞🚫😞😞😞🚫😞😞😞🚫😞😞😞🍷❤️😞🍷😞🚫❤️
예측(1~26): 🚫😞😞🚫😞😞🚫❤️😞😞❤️🚫😞🚫😞🚫😞🚫😞🚫🍷🍷🚫🚫🚫🚫😞🚫🚫🚫

정답(27~52): 🚫😞❤️🚫😞😞🚫😞🍷🍷😞🚫❤️❤️🚫❤️❤️😞❤️😞😞😞❤️😞🍷🍷
예측(27~52): 🚫🚫🚫😞😞😞🚫🚫🚫🚫❤️🚫😞🚫❤️❤️😞❤️🚫😞😞😞😞😞🚫😞🍷🍷

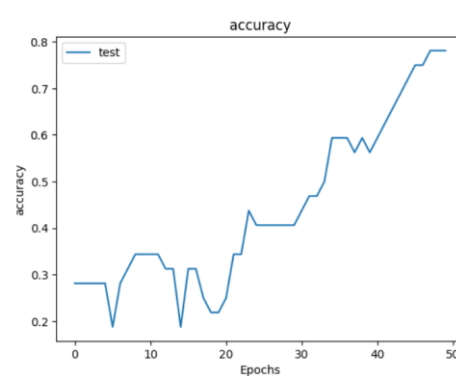
(b) LSTM+SGD+50d

Epoch 50/50 Training Loss: 0.4882 Test Loss: 0.7041, Test Accuracy: 78.12%

(1) Loss graph



(2) Accuracy graph



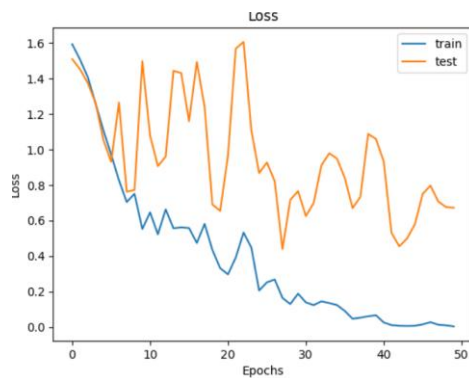
(3)all emojis for test set

[illegible]

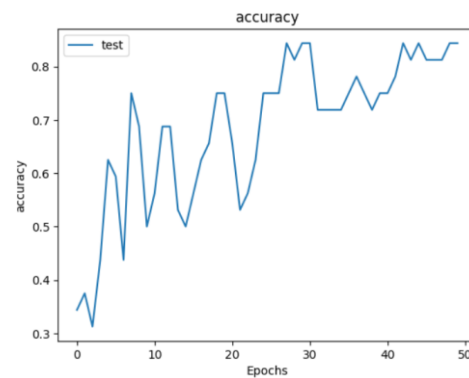
(c) LSTM+ADAM+50d

Epoch 50/50 Training Loss: 0.0029 Test Loss: 0.6715, Test Accuracy: 84.38%

(1) Loss graph



(2) Accuracy graph



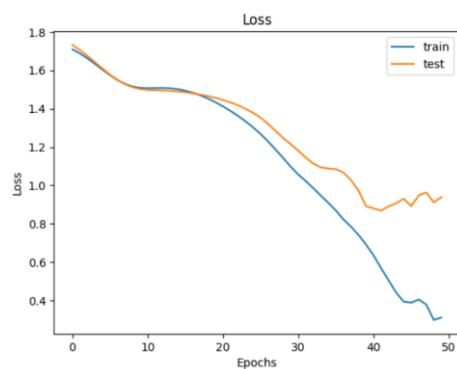
(3)all emojis for test set

[illegible]

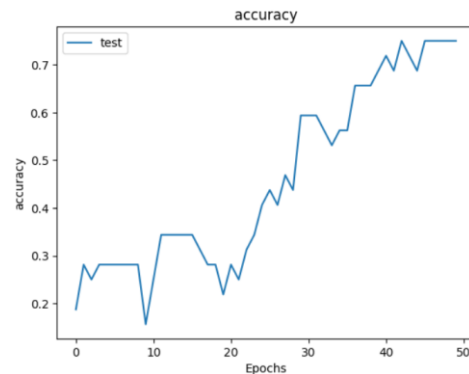
(d) LSTM+SGD+100d

Epoch 50/50 Training Loss: 0.3110 Test Loss: 0.9382, Test Accuracy: 75.00%

(1) Loss graph



(2) Accuracy graph



(3)all emojis for test set

[illegible]

보다 test loss가 더 낮으며 Accuracy가 더 높게 나타난 것을 알 수 있다.

RNN은 long term 메모리를 잘 처리하지 못하는 반면 LSTM은 추가적인 gate를 통해 long term 메모리를 잘 처리할 수 있기 때문에 나타난 결과라고 해석할 수 있다.

(3) Glove 벡터 차원 비교(50d vs 100d)

(b)와 (d)를 비교해보면 동일한 모델과 optimizer에 대해서 입력 벡터의 차원의 크기를 늘렸을 때(50d->500d), 모델의 성능은 오히려 감소하는 경향을 보였다. Embedding 벡터의 차원을 늘리면 더 많은 정보를 담을 수 있지만, 학습 데이터가 충분하지 않을 경우 overfitting이 일어날 수 있기 때문에 나타난 결과라고 해석할 수 있다.

(4) Dropout 적용 비교

(b)와 (e)를 비교해보면 동일한 조건에서 (e)에서는 dropout을 추가한 것이다. Dropout은 모델이 학습 데이터에 과적합 되는 것을 방지하기 위함이지만 오히려 dropout을 추가한 (e)에서 정확도가 더 떨어진 것을 확인할 수 있다. 이는 dropout으로 인해 모델이 충분히 학습되지 못하여 성능이 저하된 것으로 해석할 수 있다.

(5) Word2vec와 Glove의 차이점

Word2vec는 신경망 기반으로 단어의 문맥적 관계를 예측한다. 학습방식은 로컬 문맥 정보를 이용하며 중심 단어와 주변 단어 간의 확률적 관계를 동적으로 학습한다. 학습 데이터는 주변 단어의 문맥적 관계이고 결과 벡터는 신경망의 학습된 가중치이다. 대표적으로 skip-gram과 CBOW가 있다.

Glove는 행렬 분해 기반으로 동시 등장 빈도 행렬에서 단어 간의 전체적인 관계를 학습한다. 학습방식은 전역 통계 정보를 이용하며 동시 등장 빈도 로그 값을 이용하여 행렬을 분해한다. 학습데이터는 모든 단어 쌍의 동시 등장 빈도 정보이며 결과 벡터는 행렬 분해 결과로 생성된 벡터이다.

위 프로젝트에서 glove 데이터를 이용한 이유는 glove 동시 등장 빈도(co-occurrence matrix)를 기반으로 학습되며, 단어와 단어 사이의 전역적 통계 정보를 포함한다. 예를 들어 "happy"와 "joyful" 또는 "sad"나 "miserable"같은 단어는 동시 등장 패턴을 통해 비슷한 벡터로 매핑된다. 이런 벡터는 감정 분류 작업에서 유사한 감정을 표현하는 단어들을 구분하는데 효과적이다. 또한 glove는 대규모 데이터셋을 통해 미리 학습된 벡터이므로 추가적인 학습이 필요 없어서 학습 시간이 단축되고, 더 적은 계산으로도 좋은 성능을 낼 수 있다.