

11장. LSTM 과제

부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공
201724579 정현모

1. 개요

LSTM은 Long Short-Term Memory의 약자로 기존 RNN 신경망에서 나타나는 순환 벡터나 손실 기울기 정보의 소멸 및 폭주 현상을 해소한 신경망이다. RNN의 고질적인 문제인 오래 된 정보를 잊는다는 점을 해결하여 문제 처리 능력이 우수하다. 이번 과제에선 LSTM의 원도우 크기에 따른 실험 결과를 알아보고, RNN과 LSTM의 성능 차이에 대해서도 알아보겠다.

2. 학습 결과

2.1. 도시 소음 분류 신경망 학습 비교

2.1.1. Epoch 에 따른 순환 신경망 모델 비교

Model us_basic_10_10

구분	내용	값
조건	Epoch	10
	hidden layer	RNN
최종 결과	Accuaracy	34.8%
	Train time	46 secs
결과 화면	Model us_basic_10_10 train started: Epoch 2: cost=2.120, accuracy=0.240/0.180 (10/10 secs) Epoch 4: cost=2.012, accuracy=0.290/0.230 (9/19 secs) Epoch 6: cost=1.937, accuracy=0.325/0.290 (10/29 secs) Epoch 8: cost=1.889, accuracy=0.328/0.210 (9/38 secs) Epoch 10: cost=1.846, accuracy=0.345/0.230 (8/46 secs) Model us_basic_10_10 train ended in 46 secs: Model us_basic_10_10 test report: accuracy = 0.348, (0 secs)	

11장. LSTM 과제

Model us_lstm_10_10

구분	내용	값
조건	Epoch	10
	hidden layer	LSTM + 순환벡터
최종 결과	Accuarcy	32.5%
	Train time	151 secs
결과 화면	Model us_lstm_10_10 train started: Epoch 2: cost=2.130, accuracy=0.249/0.230 (30/30 secs) Epoch 4: cost=2.099, accuracy=0.250/0.240 (30/60 secs) Epoch 6: cost=2.018, accuracy=0.287/0.290 (31/91 secs) Epoch 8: cost=1.974, accuracy=0.288/0.220 (30/121 secs) Epoch 10: cost=1.983, accuracy=0.293/0.250 (30/151 secs) Model us_lstm_10_10 train ended in 151 secs: Model us_lstm_10_10 test report: accuracy = 0.325, (1 secs)	

Model us_basic_10_10

구분	내용	값
조건	Epoch	10
	hidden layer	LSTM + 상태벡터
최종 결과	Accuarcy	35.3%
	Train time	151 secs
결과 화면	Model us_state_10_10 train started: Epoch 2: cost=2.060, accuracy=0.252/0.200 (30/30 secs) Epoch 4: cost=1.962, accuracy=0.284/0.270 (30/60 secs) Epoch 6: cost=1.903, accuracy=0.305/0.300 (30/90 secs) Epoch 8: cost=1.834, accuracy=0.345/0.300 (30/120 secs) Epoch 10: cost=1.764, accuracy=0.367/0.380 (31/151 secs) Model us_state_10_10 train ended in 151 secs: Model us_state_10_10 test report: accuracy = 0.353, (0 secs)	

11장. LSTM 과제

2.2.1. Epoch 에 따른 순환 신경망 모델 비교

Model us_basic_10_10

구분	내용	값
조건	Epoch	100
	hidden layer	RNN
최종 결과	Accurarcy	37.4%
	Train time	410 secs
결과 화면	Model us_basic_10_10 train started: Epoch 20: cost=1.627, accuracy=0.414/0.260 (78/78 secs) Epoch 40: cost=1.491, accuracy=0.448/0.340 (80/158 secs) Epoch 60: cost=1.429, accuracy=0.465/0.340 (85/243 secs) Epoch 80: cost=1.414, accuracy=0.481/0.270 (84/327 secs) Epoch 100: cost=1.319, accuracy=0.497/0.330 (83/410 secs) Model us_basic_10_10 train ended in 410 secs: Model us_basic_10_10 test report: accuracy = 0.374, (0 secs)	

Model us_lstm_10_10

구분	내용	값
조건	Epoch	100
	hidden layer	LSTM + 순환벡터
최종 결과	Accurarcy	48.6%
	Train time	1569 secs
결과 화면	Model us_lstm_10_10 train started: Epoch 20: cost=1.749, accuracy=0.378/0.290 (311/311 secs) Epoch 40: cost=1.561, accuracy=0.448/0.340 (313/624 secs) Epoch 60: cost=1.425, accuracy=0.492/0.430 (318/942 secs) Epoch 80: cost=1.383, accuracy=0.539/0.380 (321/1263 secs) Epoch 100: cost=1.275, accuracy=0.582/0.390 (306/1569 secs) Model us_lstm_10_10 train ended in 1569 secs: Model us_lstm_10_10 test report: accuracy = 0.486, (1 secs)	

11장. LSTM 과제

Model us_basic_10_10

구분	내용	값
조건	Epoch	100
	hidden layer	LSTM + 상태벡터
최종 결과	Accuarcy	57.7%
	Train time	1532 secs
결과 화면	Model us_state_10_10 train started: Epoch 20: cost=1.538, accuracy=0.450/0.460 (300/300 secs) Epoch 40: cost=1.247, accuracy=0.547/0.520 (303/603 secs) Epoch 60: cost=1.119, accuracy=0.600/0.490 (308/911 secs) Epoch 80: cost=0.991, accuracy=0.646/0.430 (310/1221 secs) Epoch 100: cost=0.889, accuracy=0.678/0.530 (311/1532 secs) Model us_state_10_10 train ended in 1532 secs: Model us_state_10_10 test report: accuracy = 0.577, (1 secs)	

2.2.2. 윈도우 크기에 따른 순환 신경망 비교

Model us_basic_10_100

구분	내용	값
조건	Epoch	100
	hidden layer	RNN
최종 결과	Accuarcy	59.0%
	Train time	387 secs
결과 화면	Model us_basic_10_100 train started: Epoch 20: cost=1.138, accuracy=0.602/0.580 (91/91 secs) Epoch 40: cost=1.015, accuracy=0.649/0.550 (75/166 secs) Epoch 60: cost=0.978, accuracy=0.667/0.590 (74/240 secs) Epoch 80: cost=0.784, accuracy=0.711/0.690 (73/313 secs) Epoch 100: cost=0.845, accuracy=0.700/0.650 (74/387 secs) Model us_basic_10_100 train ended in 387 secs: Model us_basic_10_100 test report: accuracy = 0.590, (0 secs)	

11장. LSTM 과제

Model us_lstm_10_100

구분	내용	값
조건	Epoch	100
	hidden layer	LSTM + 순환벡터
최종 결과	Accuarcy	72.2%
	Train time	1463 secs
결과 화면	Model us_lstm_10_100 train started: Epoch 20: cost=1.229, accuracy=0.593/0.520 (290/290 secs) Epoch 40: cost=0.910, accuracy=0.690/0.650 (291/581 secs) Epoch 60: cost=0.746, accuracy=0.756/0.680 (293/874 secs) Epoch 80: cost=0.704, accuracy=0.772/0.660 (294/1168 secs) Epoch 100: cost=0.618, accuracy=0.806/0.680 (295/1463 secs) Model us_lstm_10_100 train ended in 1463 secs: Model us_lstm_10_100 test report: accuracy = 0.722, (1 secs)	

Model us_state_10_100

구분	내용	값
조건	Epoch	100
	hidden layer	LSTM + 상태벡터
최종 결과	Accuarcy	75.8%
	Train time	1459 secs
결과 화면	Model us_state_10_100 train started: Epoch 20: cost=0.979, accuracy=0.682/0.640 (288/288 secs) Epoch 40: cost=0.746, accuracy=0.747/0.690 (290/578 secs) Epoch 60: cost=0.609, accuracy=0.794/0.710 (292/870 secs) Epoch 80: cost=0.482, accuracy=0.837/0.740 (294/1164 secs) Epoch 100: cost=0.461, accuracy=0.848/0.730 (295/1459 secs) Model us_state_10_100 train ended in 1459 secs: Model us_state_10_100 test report: accuracy = 0.758, (1 secs)	

11장. LSTM 과제

3. 결론

3.1. RNN

RNN은 Hidden Layer가 순환 구조를 이루는 신경망의 한 종류이다. 과거의 정보를 기억하여 다시 학습에 사용한다는 점이 음성, 문자처럼 순차적으로 등장하는 데이터 처리에 적합한 모델로 작용한다.

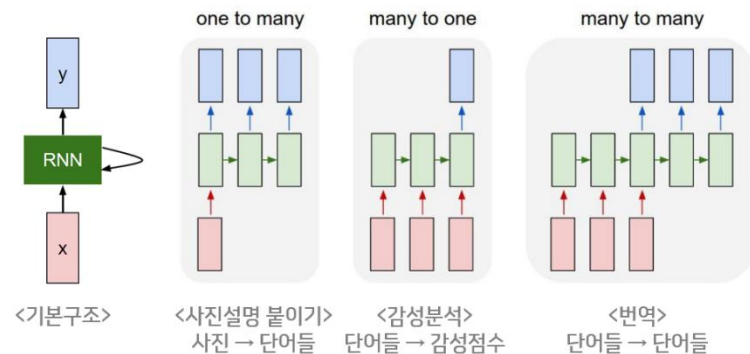


그림 1. RNN

RNN은 위 그림과 같이 과거에 학습한 정보가 다음 학습 Node로 넘어가는 형식이다. 위의 그림에서 알 수 있듯 시퀀스 길이에 상관없이 인풋과 아웃풋을 받아들일 수 있는 네트워크 구조이기 때문에 필요에 따라 다양하고 유연하게 구조를 만들 수 있다는 점이 RNN의 가장 큰 장점이다.

하지만 RNN의 가장 큰 문제점은 장기 의존성 문제점(Long-Term Dependency Problem)이다. 일반적인 RNN의 경우 짧은 시퀀스를 처리할 때 유리하지만, 관련 정보와 그 정보를 사용하는 지점이 멀어지면 학습 능력이 현저하게 떨어진다. 해당 문제의 원인은 시퀀스가 길어지면 RNN의 역전파 알고리즘이 장기 의존성 학습에 어려움이 있기 때문인데, 가중치를 업데이트 하는 과정에서 1보다 작은 값들이 계속 곱해지면서 기울기가 사라지는 기울기 소실 문제(Vanishing Gradient Problem)가 발생하기 때문이다.

기울기가 사라지게 되면 과거 정보가 현재의 학습에 영향을 미치지 못하게 되면서 RNN의 퍼포먼스가 크게 저하되기 때문에 RNN의 문제점을 해결하기 위해 여러 알고리즘들이 나타났는데, 대표적으로 LSTM이 있다.

11장. LSTM 과제

3.2. LSTM

RNN의 문제를 극복하기 위해 고안된 것이 바로 LSTM인데, LSTM은 RNN의 히든 state에 컨베이어 벨트 역할을 하는 cell state를 추가해준 모델이다.

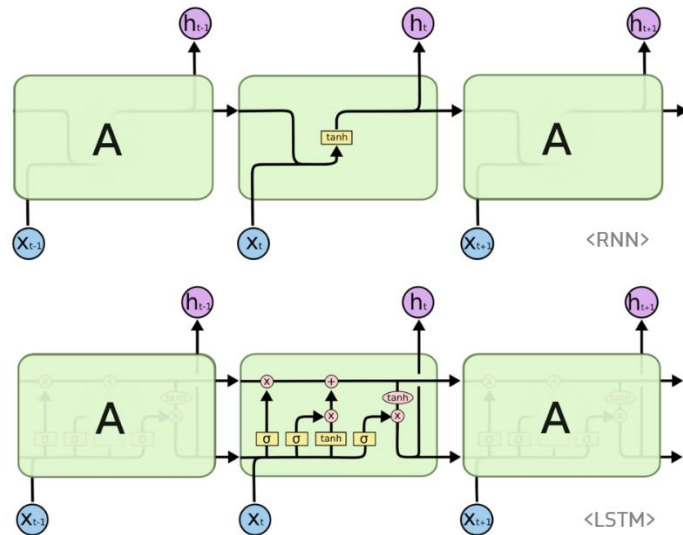


그림 2. RNN과 LSTM의 차이

위의 그림을 보면 한 state안에 여러 수식이 적혀있는데, forget gate, input gate등 과거 정보를 잊기 위한 게이트, 현재 정보를 기억하기 위한 게이트등 중요한 정보는 기억하고, 상대적으로 중요하지 않은 정보는 잊으려 하는 것이 특징이다.

3.3. 실험 결과

실험은 RNN, LSTM + 순환벡터, LSTM + 상태벡터로 총 3개의 모델을 사용하였다. 작은 윈도우 객체에서 10 epoch 학습 결과는 역시 LSTM + 상태벡터가 가장 정확도가 좋았지만, RNN이 LSTM + 순환벡터보다 성능이 약간 더 우세한 것으로 나왔다. 하지만 epoch가 커질 수록 LSTM의 장기 기억 능력 때문에 성능이 더 좋아졌다. 또한 윈도우 크기를 증가시켰을 때 LSTM과 RNN의 차이가 확연하게 드러났다. 하지만 LSTM이 RNN에서 여러 cell state가 추가되었기 때문에 상대적으로 학습속도가 4~5배 정도 난다는 단점이 있었다.