RNN

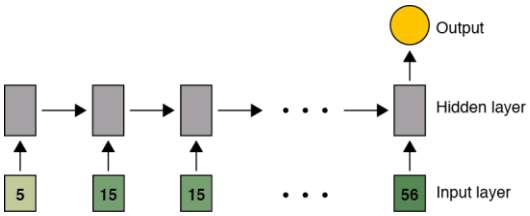
- 1990 년 기온 예측 -

2017-11-23

홍만수

1. RNN 네트워크 구조

Many-to-One



2. 인자 탐색 과정 및 결과

<1 회>

seq_length = 6 <-- 6 일간 기온을 기준으로 7 일째 기온을 예측한다.

data_dim = 1

learning_rate = 0.001

forget_bias = 1.0 <-- 기온은 보통 격하게 변하지 않으므로 이전 메모리를 잊어선 안된다고 판단했다.

output_keep_prob = 0.8

num_hidden = 512

training_steps = 2000

▸ RMSE 2.6676

▹ 훈련 부족으로 인한 것일 가능성이 있기 때문에 training_step 을 높여보기로 했다.

<2 회>

training_steps = 3000

▸ RMSE 3.1742

▹ 오버피팅은 커녕 역으로 prediction 과 실제값이 더 큰 차이가 나버렸다. training_step 을 첫회보다 더 낮춰보기로 했다.

<3 회>

training_steps = 1000

▸ RMSE 2.3663

▹ training_step 을 낮추자 RMSE 도 떨어졌다. 데이터가 복잡하지 않아서 많이 돌릴 필요가 없는 것을 거라고 생각하고 training_step 을 더 낮추어 보기로 했다.

<4 회>

training_step = 500

▸ RMSE 2.3635

▹ 반으로 줄여도 큰 차이가 나지 않는 것을 보고 다른 hidden layer 수를 줄여보기로 했다. 역시 데이터가 복잡하지 않아서 단순한 모델이 더 적합한 것으로 추정된다.

<5 회>

num_hidden = 256

▸ RMSE 2.3974

▹ 큰 차이가 없다. 혹시 모르니 training_step 을 높여보기로 한다.

<6 회>

training_step = 2000

▸ RMSE 2.4854

▹ training_step 은 500 가량이 적합해 보인다. 다음은 실험적으로 hidden layer 수와 training_step 수를 극단적으로 줄여보기로 한다.

<7 회>

num_hidden = 64

training_step = 100

▸ RMSE 3.8977

▹ 심각한 언더피팅이 드러났다. 가장 적합한 training_step 인 500 과 가장 성공적이었던 hidden layer 수였던 512 와 64 사이의 2 의 승수 중 사용해보지 않은 128 을 각각 적용해 해보기로 한다.

<8 회>

num_hidden = 128

training_step = 500

▸ RMSE 2.4629

▹ 보다 단순한 학습 모델 구현을 위해 learning_rate 를 낮춰보기로 했다.

<9 회>

learning_rate = 0.01

num_hidden = 512

training_step = 1000

▸ RMSE 3.0561

▹ 역시 제일 적합한 training_step 은 500 으로 보인다.

<10 회>

training_step = 500

▸ RMSE 2.4255

▹ learning_rate 0.001 일 때와 큰 차이가 나지는 않지만 미묘하게 더 크다. 마지막 단순화로 drop-out wrapper 를 제거해보기로 했다.

<11 회>

▸ RMSE 2.3436

<결과>

11 회차에 사용된 hidden layer 512 와 training_step 500 이 가장 낮은 RMSE 2.3436 를 뽑아냈다. 물론 data dimension 이나 forget bias 등을 조작할 수도 있지만 특별히 그래야할 이유를 찾지 못한채로 섣불리 손대는 일은 피하고 싶었고, 무엇보다 레포트를 쓸 자리가 모자르다. train set 과 validation set 의 분배를 바꾸는 것도 영향을 끼칠 것이다.

결론은, feature 가 몇 개 없고 record 수도 약 3200 여개였던만큼 복잡하고 깊은 모델보다 단순한 모델이 더 좋은 결과를 내는 것으로 보인다.

3. 데이터 조작 과정 및 결과

우선 train.csv 를 pandas dataframe 으로 열어, 각 컬럼을 확인했다. 인덱스인 Date 와 Y 인 Label 을 제거한 F1~F6 만을 추출하여 numpy array 인 train 에 사용할 X 로 구현했다. X 를 다시

70%만 training 에 쓰고 나머지 30%는 validation 에 쓰기 위해 Xtr 과 Xval 로 나누었다. Y 도 마찬가지로 Label 을 Ytr 과 Yval 로 나누었다.

이후, 몇 차례 다른 인자를 적용한 training 을 거쳐 실제 Label 과 validation set 을 prediction 한 결과를 비교했을 때 root mean square error(RMSE)가 낮은 쪽을 채용하였다. validation 의 최종 RMSE 결과는 2.3635 를 기록했다.

이제 test.csv 를 pandas dataframe 으로 열어, 각 컬럼을 확인한 후, 인덱스인 Date 를 뺀 나머지를 Xte 란 변수명의 numpy array 로 받아온 후 1990 년 기온 prediction 을 뽑아냈다.