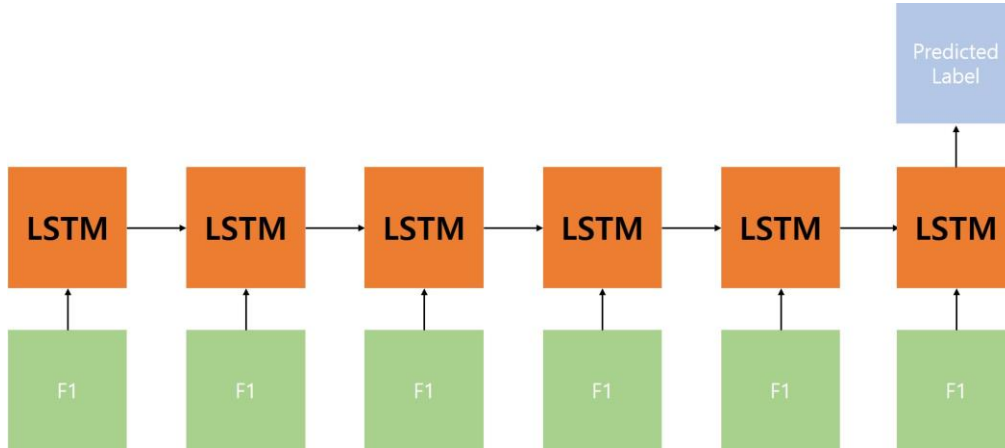


HW3 RNN

빅데이터플랫폼 객민지

1. RNN 네트워크 구조



2. 코딩

- * 주어진 train.csv 의 전반부 200 행을 test 셋으로 하였고 나머지를 train 셋으로 설정하였다.
- * hidden dimension, learning rate, iteration 횟수, cell layer 층의 개수를 다르게 하여 학습을 실시하도록 설정하였다.
- * test.csv.에 대한 prediction 은 주어진 train.csv 의 모든 행에 대해 학습을 실시한 후 그 모델로 예측을 함.

3. 인자 탐색 과정

`samplemain(hiddendim,learningrate,iterations,layernum,printyes)`

이 함수의 4 가지 parameter 를 hyperparameter 로 간주하여 탐색하였다.

- * hiddendim: Hidden layer 의 차원수를 의미한다.
- * learning rate
- * iteration: training set 을 학습하는 횟수이다. 너무 많이 학습하면 overfitting 이 발생할 것이다.
- * layernum: 쌓아올린 layer 의 층수이다.

2-1) hidden layer 의 dimension

hidden layer 6 으로 했을 때(2.47)가 10(2.56) 또는 2(2.50)로 했을 때에 비해 에러가 더 작았다.

2-1) learning rate

0.05, 0.01, 0.005, 0.001 로 실행해 본 결과

0.005, 0.001 때(2.60)에 비해 0.05, 0.01 일 때(2.55)의 에러가 더 작았다.

2-2) optimizer

GradientDescentOptimizer 를 사용시(2.67) adamoptimizer 를 사용했을 때(2.47)보다 크게 상승하였다

2-3) iteration

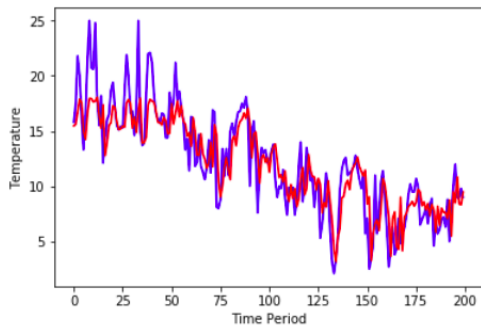
* 1000 번 : rmse 가 2.55 였다.

* 2000 번 : rmse 가 2.50 으로 하락하였다.

* 3000 번: rmse 가 2.47 까지 하락하였다.

```
pred=samplemain(6,0.01,3000,1,False)##
```

RMSE: 2.4725711345672607



* 10000 번: rmse 가 2.67 정도였다. overfitting 이 발생하였다.

2-4) multi cell 사용 여부

multi cell 을 사용하였더니(2.60) 사용하지 않았을때보다(2.55) 오히려 error 가 상승하였다.

3. 결론

* hiddendim = 6

* learning rate = 0.01

* iteration = 3000

* layernum = 1

일 때 가장 만족스러운 예측력을 보여주었다.