**Developer’s Manual for RNN-compression Project, Ver. 1**

2020.11.10

Donghae Jang(jangdonghae@snu.ac.kr)

U Kang (ukang@snu.ac.kr)

**매뉴얼 구성**

1. Starlab RNN-compression 과제 코드 개요
2. 코드 실행 과정
3. 세부 코드 모듈 별 설명

**1. Starlab RNN-compression 과제 코드 개요**

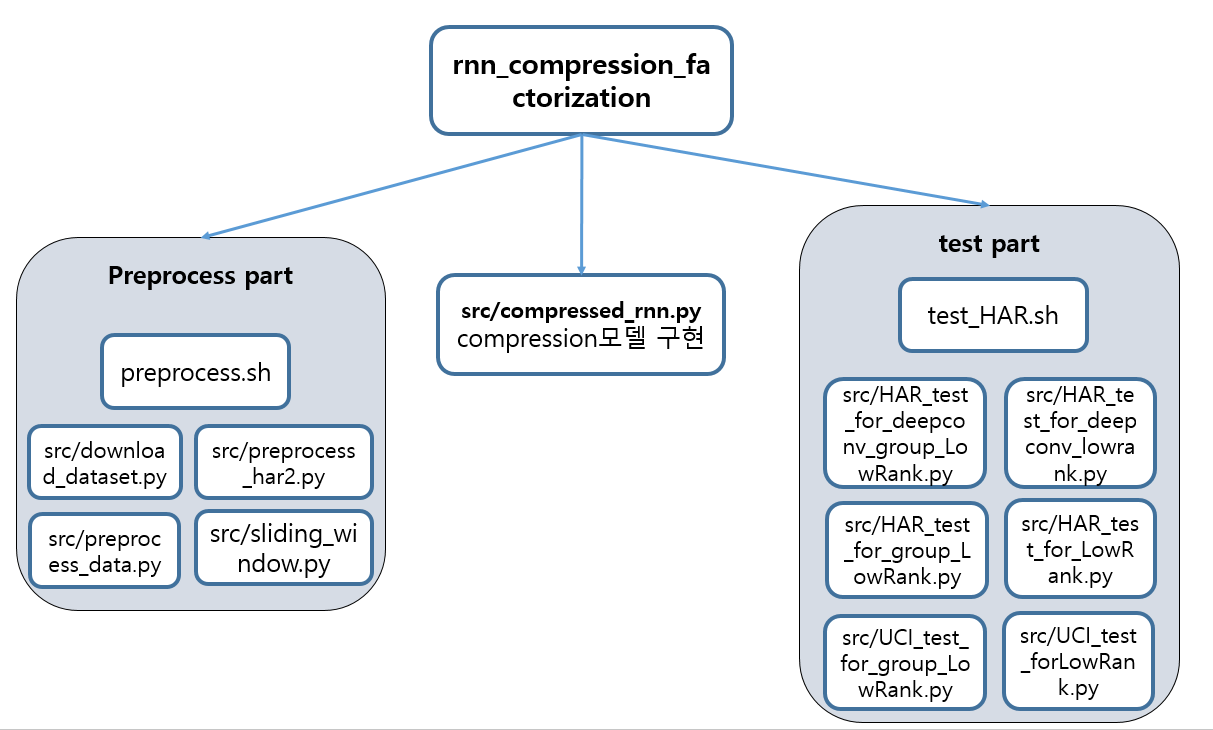


Figure 1. Starlab RNN-compression 코드의 구조도

Starlab RNN-compression 코드는 그림 1에서 확인할 수 있듯이 크게 세 부분으로 Preprocess를 담당하는 코드와, 모델 구현을 담당하는 compressed\_rnn.py 파일과, test 를 담당하는 코드들로 나누어져 있다. 일단 Preprocess part의 경우 데이터셋 다운로드, preprocessing을 하는 코드들이 담겨있고, preprocess.sh 파일을 실행함으로써 한번에 진행이 가능하다. RNN-compression을 위한 compressed rnn 모델은 compressed\_rnn.py 파일에 구현되어 있다. test part는 이를 테스트하기 위한 파트로 Opportunity dataset에서의 테스트를 위한 코드와, UCI HAR dataset에서의 테스트를 위한 코드로 구성되어 있다. Opportunity dataset의 경우 기본 LSTM 모델의 compression 후 성능을 테스트 하는 것이 가능하고, 성능이 우수한 DeepConvolutionalLSTM 모델의 compression 후 성능도 테스트가 가능하다.

**2. 코드 실행 과정**

* 실행 환경
  + shell script및 전처리 부분은 Linux 환경이 필요(Windows 사용시 코드 변경 필요)
  + pytorch, numpy, pandas, scipy가 설치되어 있는 환경
  + 사용 가능한 GPU가 존재하고, CUDA가 설치되어 있는 환경이 선호됨(CPU에서도 코드 실행 가능은 함)
* 실행 방법
  + 1. Github repo에서 코드를 받을 경우 git clone을 통해 가지고 오면 되고, 코드를 받는 경우 압축 해제
  + 2. $ bash preprocess.sh를 통해서 전처리를 하는 shell script를 실행하면 데이터 다운로드 및 전처리가 진행됨
  + 3. $ bash test\_HAR HAR\_deepconv를 통해서 코드의 메인 파트인 Opportunity 데이터셋에서의 DeepConvolutionalLSTM 모델에 대한 compression 실험이 가능
  + 4. $ bash test\_HAR HAR, $ bash test\_HAR UCI를 통해서 다른 데이터셋에 대해서 기본적인 multi-layer LSTM, GRU모델에 대한 compression 실험도 가능
  + 5. 현재 각 test 코드는 reproduce를 목적으로 seed가 고정이 되어있고, 각 test코드의 70번째 줄 부분의 seed 값을 변경함으로써 random seed를 변경할 수 있다.

**3. 세부 코드 모듈 별 설명**

1. **src/compressed\_rnn.py:** RNN compression을 위한 compressed rnn cell과 compressed rnn 모델이 구현되어 있는 구현체
   1. **myGRUCell( ), myLSTMCell( )**  
      기본적인 rnn model을 구현하기 위한 rnn cell, 추가적으로 LowRank꼴을 학습할 수 있게 wRank(non-recurrent weight), uRank(recurrent weight)를 input으로 받을 수 있음.
   2. **myGRUCell\_group( ), myLSTMCell\_group( )**  
      group LowRank rnn cell을 구현하기 위한 부분. group의 개수가 늘어남에 따라 Hardcoding을 하였기 때문에, group 개수에 따라서 Cell을 따로 구현함. wRank(non-recurrent weight)와 uRanks(recurrent weight의 각 group의 rank 수열)을 input으로 받을 수 있음.
   3. **myGRU( ), myLSTM( )**

위에서 구현한 LowRank 꼴을 학습할 수 있는 기본적인 rnn cell을 여러 층 쌓아서 만든 rnn model의 구현체. Hidden state size의 수열과 wRank(non-recurrent weight), uRank(recurrent weight)를 input으로 받을 수 있음.

* 1. **myGRU\_group( ), myLSTM\_group( )**

group LowRank rnn cell을 쌓아서 만든 rnn 모델의 구현체.

* 1. **DeepConv( )**  
     HAR 모델링을 위해 사용되는 고급 RNN 모델인 DeepConvolutionalLSTM 모델에서 LSTM 부분을 빼고 모델링 한 구현체, DeepConv 모듈과 위에서 구현한 myLSTM, myLSTM\_group을 합해서 Compression이 가능한 DeepConvLSTM 모델을 구현 가능함.

1. **Preprocess part**: 데이터 다운로드 및 전처리를 위한 부분
   1. **src/download\_dataset.py**

Opportunity dataset과 UCI HAR dataset이 경로에 없는 경우 다운로드를 하는 코드

* 1. **src/sliding\_window.py**

시계열 데이터를 전처리 하기 위해서 sliding window 기법을 구현해놓은 코드

* 1. **src/preprocess\_data.py**

Opportunity dataset에 있는 방대한 양의 데이터를 적절히 전처리 하여 Sensor 113개에 대해서 input이 들어오고, 이에 해당하는 output이 있는 학습에 용이한 데이터셋으로 변형하는 코드

* 1. **src/preprocess\_har2.py**

sliding window기법을 이용해서 RNN model 학습에 용이한 데이터 꼴로 만들어서 저장하는 코드

* 1. **preprocess.sh**

2.1 ~ 2.4 에 있는 전처리용 코드를 연속으로 실행하여 전처리를 한번에 끝내주는 shell script

1. **test part**: LowRank compress RNN 모델을 학습하고 정확도를 테스트하기 위한 코드
   1. **src/HAR\_test\_for\_deepconv\_lowrank.py**

Opportunity dataset을 DeepConvLSTM으로 모델링하고 LSTM 부분을 LowRank 형태로 학습하는 코드 구현 및 정확도 테스트하는 코드

* 1. **src/HAR\_test\_for\_deepconv\_group\_LowRank.py**

Opportunity dataset을 DeepConvLSTM으로 모델링하고 LSTM 부분을 proposed method인 group\_LowRank 형태로 학습하는 코드 구현 및 정확도 테스트하는 코드

* 1. **src/HAR\_test\_for\_LowRank.py**

Opportunity dataset을 기본적인 LSTM으로 모델링하고 LSTM 부분을 LowRank 형태로 학습하는 코드 구현 및 정확도 테스트하는 코드

* 1. **src/HAR\_test\_for\_group\_LowRank.py**

Opportunity dataset을 기본적인 LSTM으로 모델링하고 LSTM 부분을 proposed method인 group\_LowRank 형태로 학습하는 코드 구현 및 정확도 테스트하는 코드

* 1. **src/UCI\_test\_for\_LowRank.py**

UCI HAR datset을 기본적인 LSTM으로 모델링하고 LSTM 부분을 LowRank 형태로 학습하는 코드 구현 및 정확도 테스트하는 코드

* 1. src/UCI\_test\_for\_deepconv\_group\_LowRank.py

UCI HAR dataset을 기본적인 LSTM으로 모델링하고 LSTM 부분을 proposed method인 group\_LowRank 형태로 학습하는 코드 구현 및 정확도 테스트하는 코드

* 1. **test\_HAR.sh**

training과 testing을 위한 shell script.

- argument에 “HAR\_deepconv” 넣는 경우 : 코드의 메인 파트인 Opportunity 데이터셋에서의 DeepConvolutionalLSTM 모델에 대한 compression 실험이 가능

- argument에 “HAR” 넣는 경우 : Opportunity 데이터셋에서 기본적인 LSTM, GRU 모델에 대한 compression 실험이 가능

- argument에 “UCI” 넣는 경우 : UCI\_HAR 데이터셋에서 기본적인 LSTM, GRU 모델에 대한 compression 실험이 가능