**서울대학교 4 차 산업혁명 아카데미**

**빅데이터 플랫폼**

**딥러닝 (강유 교수님) 숙제 3**

**출제:** 2017 년 11 월 16 일 목요일

**제출:** 2017 년 11 월 23 일 목요일

본 숙제의 목표는 TensorFlow 를 이용하여 시계열(time-series) 데이터 예측을 위한 RNN 모델을 만들어 보는 것입니다. 다른 라이브러리도 자유롭게 사용할 수 있으나 모델 부분은 TensorFlow 로만 이루어져 있어야 합니다. LSTM 등 기존 RNN 구조에서 발전한 형태를 사용하는 것도 허용됩니다. 숙제에 사용할 데이터와 뼈대 코드가 제공됩니다.

## 1. 데이터 정보

숙제에 사용되는 데이터는 호주 멜버른에서 측정된 10 년 동안의 일별 기온입니다. 하루 중에서 최저 기온만 기록되며 하루에 한 개씩 총 3650 개의 측정치가 존재합니다. 기온 값은 섭씨로 주어지며 모든 값은 -0.8 부터 26.3 사이의 범위를 가지고 있습니다. 주어진 데이터는 단순한 시계열 데이터로서 피처와 클래스를 따로 포함하지 않습니다.

## 2. 문제 정의

본 숙제에서 풀어야 할 문제는 6 일 동안의 기온이 주어졌을 때 7 일 째의 최저 기온을 예측하는 것입니다. 예를 들어 2017 년 11 월 6 일부터 11 월 12 일까지의 기온이 주어졌을 때 11 월 13 일의 기온을 예측해야 합니다. 이와 같이 부분적인 시계열이 주어졌을 때 다음에 나타날 값을 예측하는 문제를 시계열 예측 문제(time-series prediction problem)라고 합니다. 정해진 클래스 중 하나로 분류하는 것이 아니라 값을 예측하는 것이 목표이므로 분류(classification) 문제가 아니라 회귀(regression) 문제라고 할 수 있습니다. 따라서 실습에서와는 달리 딥 러닝 네트워크의 마지막 레이어를 소프트맥스(softmax)가 아닌 형태로 사용해야 합니다. 소프트맥스는 주어진 인스턴스가 각 라벨에 속할 확률을 계산하는 함수이기 때문입니다.

## 3. 평가 방법

모델의 평가는 예측한 값과 실제 값 사이의 평균 제곱 오차(mean squared error)에 기반해 이루어집니다. 이는 회귀 문제에서 널리 쓰이는 방식으로 두 벡터의 L2 거리를 계산하는 것과 같은 방식입니다.  평균 제곱 오차에 기반한 로스(loss)는 다음과 같이 계산됩니다.

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

여기서 $\mathbf{y}$는 원래 기온, $\hat{\mathbf{y}}$는 예측한 기온, $n$은 인스턴스의 수를 나타냅니다. 예측한 값이 실제 값과 가까울수록 로스의 값은 낮아지며, 두 벡터가 완전히 일치할 경우 0 이 됩니다.

### 4. 제공 데이터

본 숙제에는 세 개의 데이터 파일이 제공됩니다. 첫 번째 파일은 별도로 처리하지 않은 원래 데이터이고, 두 번째와 세 번째 파일은 훈련과 테스트에 사용할 파일입니다. 자세한 정보는 다음과 같습니다. 본 숙제에서는 따로 검증(validation) 데이터를 제공하지 않으므로 훈련 데이터를 나누어 하이퍼 파라미터 검증을 위해 사용해야 합니다.

- **시계열 데이터(raw.csv):** 1981 년부터 1989 년까지의 시계열 데이터입니다. 본 파일을 이용하여 데이터가 가지고 있는 특징(패턴이나 범위 등)을 분석할 수 있습니다.
- **훈련 데이터(train.csv):** 1981 년부터 1989 년까지의 시계열 데이터를 피처(feature)와 라벨(label) 형태로 가공한 데이터입니다. 피처로는 매 6 일 동안의 기온과 날짜가, 라벨로는 7 일 째의 기온이 들어갑니다. 훈련에 사용하길 권장합니다. 이 데이터를 사용하여 훈련과 검증을 모두 진행해야 합니다.
- **테스트 데이터(test.csv):** 테스트를 위해 사용할 1990 년의 데이터입니다. 월요일부터 토요일까지의 기온이 피처로 주어지며 매주 일요일의 값이 라벨이 됩니다. 모든 라벨이 가려져 있습니다.

### 5. 코드 구현

여러분의 코드(main.py)는 훈련 데이터(train.csv)를 읽어서 모델을 만든 다음 테스트 데이터(test.csv)를 읽어서 예측한 라벨을 파일(result.csv)로 출력해야 합니다. 코드 구현에 대해서는 뼈대 코드를 참고하기 바랍니다.

## 6. 보고서 작성

여러분은 2 페이지 이내의 간단한 보고서를 작성하여 제출해야 합니다. 본 숙제는 모델의 객관적인 성능 뿐 아니라 여러분이 좋은 모델을 찾기 위해 노력한 과정을 기반으로 채점됩니다. 따라서 여러분의 보고서는 다음과 같은 내용을 필수적으로 포함해야 합니다.

- **RNN 네트워크 구조:** 구현한 RNN 네트워크의 구조를 명확하게 서술해야 합니다. 그림이나 도표 등을 사용하면 좋습니다.
- **인자 탐색 과정 및 결과:** 여러분이 사용한 모델의 최적 인자를 탐색하는 과정, 방법, 그로 인해 찾은 최적 인자를 서술해야 합니다.
- **데이터 조작 과정 및 결과:** 주어진 데이터를 조작해서 사용한다면, 데이터 조작 과정과 그 결과에 대해 서술해야 합니다.

## 7. 제출 방법

완성된 숙제는 압축하여 유재민 조교(jaeminyoo@snu.ac.kr)에게 보내면 됩니다. 압축 파일의 이름에는 제출자의 이름이 반드시 포함되어 있어야 합니다. 압축 파일에 포함되어야 하는 파일의 목록은 다음과 같습니다. 이중 코드 파일(main.py)은 Python 3 환경에서 실행 가능해야 하며, 데이터 파일(test.csv)이 주어졌을 때 결과 파일(result.csv)을 만들어야 합니다.

- HW1_{이름}.zip (예: HW1_유재민.zip)
    - **report.pdf:** 보고서 파일입니다. PDF 형식이어야 합니다.
    - **main.py:** 결과 코드입니다.
    - **result.csv:** 여러분의 코드를 실행한 예측 결과입니다.
    - **README.txt:** (선택사항) 추가적으로 언급할 내용을 적으면 됩니다.

## 8. 주의 사항

다음과 같은 주의 사항을 지키지 않을 경우 감점될 수 있습니다.

- 보고서는 2 페이지 이내여야 합니다.
- 보고서는 코드를 포함하지 않는 편이 낫습니다.
- 제출한 코드는 하이퍼 파라미터 탐색 과정을 포함하지 않는 상태여야 합니다. 채점 시 코드를 돌려볼 때 지나치게 오랜 시간이 걸리기 때문입니다.