Homework #4

[ECE30021/ITP30002] Operating Systems

Mission

- Develop a predictor for student dataset using pytorch.
  - Achieve your best performance using MLP.
- Submission
  - Submit an .ipynb file on HISNet report board
    - Filename: hw4_<student_id>.ipynb
- Due date: PM 11:00, Nov. 16th
  - This assignment is for individual students
  - Group discussion is allowed only after Nov. 17th

Honor Code Guidelines

- “과제”
  - 과제는 교과과정의 내용을 소화하여 실질적인 활용 능력을 갖추기 위한 교육활동이다. 학생은 모든 과제를 정직하고 성실하게 수행함으로써 과제에 의도된 지식과 기술을 얻기 위해 최선을 다해야 한다.
  - 제출된 과제물은 성적 평가에 반영되므로 공식적으로 허용되지 않은 자료나 도움을 획득, 활용, 요구, 제공하는 것을 포함하여 평가의 공정성에 영향을 미치는 모든 형태의 부정행위는 단호히 거부해야 한다.
  - 수업 내용, 공지된 지식 및 정보, 또는 과제의 요구를 이해하기 위하여 동료의 도움을 받는 것은 부정행위에 포함되지 않는다. 그러나, 과제를 해결하기 위한 모든 과정은 반드시 스스로의 힘으로 수행해야 한다.
  - 담당교수가 명시적으로 허락한 경우를 제외하고 다른 사람이 작성하였거나 인터넷 등에서 획득한 과제물, 또는 프로그램 코드의 일부, 또는 전체를 이용하는 것은 부정행위에 해당한다.
  - 자신의 과제물을 타인에게 보여주거나 빌려주는 것은 공정한 평가를 방해하고, 해당 학생의 학업 성취를 저해하는 부정행위에 해당한다.
  - 팀 과제가 아닌 경우 두 명 이상이 함께 과제를 수행하여 이를 개별적으로 제출하는 것은 부정행위에 해당한다.
  - 스스로 많은 노력을 한 후에도 버그나 문제점을 파악하지 못하여 동료의 도움을 받는 경우도 단순한 문법적 오류에 그쳐야 한다. 과제가 요구하는 design, logic, algorithm의 작성에 있어서 담당교수, TA, tutor 이외에 다른 사람의 도움을 받는 것은 부정행위에 해당한다.
  - 서로 다른 학생이 제출한 제출물간 유사도가 통상적으로 발생할 수 있는 정도를 크게 넘어서는 경우, 또는 자신이 제출한 과제물에 대하여 구체적인 설명을 하지 못하는 경우에는 부정행위로 의심받거나 판정될 수 있다.

UCI Student Performance Dataset

- Homepage
  - https://archive.ics.uci.edu/ml/datasets/Student+Performance

Student Performance Data Set

Download: Data Folder, Data Set Description

Abstract: Predict student performance in secondary education (high school).

Data Set Characteristics:	Multivariate	**Number of Instances:**	649	**Area:**	Social
Attribute Characteristics:	Integer	**Number of Attributes:**	33	**Date Donated**	2014-11-27
Associated Tasks:	Classification, Regression	**Missing Values?**	N/A	**Number of Web Hits:**	803230

- Input columns
  - All columns except for the target columns (G1, G2, and G3)
- Target column: G3

UCI Student Performance Dataset

- Input attributes
  - 1 school – student's school (binary: 'GP' – Gabriel Pereira or 'MS' – Mousinho da Silveira)
  - 2 sex – student's sex (binary: 'F' – female or 'M' – male)
  - 3 age – student's age (numeric: from 15 to 22)
  - 4 address – student's home address type (binary: 'U' – urban or 'R' – rural)
  - 5 famsize – family size (binary: 'LE3' – less or equal to 3 or 'GT3' – greater than 3)
  - 6 Pstatus – parent's cohabitation status (binary: 'T' – living together or 'A' – apart)
  - 7 Medu – mother's education (numeric: 0 – none, 1 – primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
  - 8 Fedu – father's education (numeric: 0 – none, 1 – primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
  - 9 Mjob – mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
  - 10 Fjob – father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
  - 11 reason – reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
  - 12 guardian – student's guardian (nominal: 'mother', 'father' or 'other')
  - 13 traveltime – home to school travel time (numeric: 1 – <15 min., 2 – 15 to 30 min., 3 – 30 min. to 1 hour, or 4 – >1 hour)
  - 14 studytime – weekly study time (numeric: 1 – <2 hours, 2 – 2 to 5 hours, 3 – 5 to 10 hours, or 4 – >10 hours)
  - 15 failures – number of past class failures (numeric: n if 1<=n<3, else 4)
  - 16 schoolsup – extra educational support (binary: yes or no)
  - 17 famsup – family educational support (binary: yes or no)
  - 18 paid – extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
  - 19 activities – extra-curricular activities (binary: yes or no)
  - 20 nursery – attended nursery school (binary: yes or no)
  - 21 higher – wants to take higher education (binary: yes or no)
  - 22 internet – Internet access at home (binary: yes or no)
  - 23 romantic – with a romantic relationship (binary: yes or no)
  - 24 famrel – quality of family relationships (numeric: from 1 – very bad to 5 – excellent)
  - 25 freetime – free time after school (numeric: from 1 – very low to 5 – very high)
  - 26 goout – going out with friends (numeric: from 1 – very low to 5 – very high)
  - 27 Dalc – workday alcohol consumption (numeric: from 1 – very low to 5 – very high)
  - 28 Walc – weekend alcohol consumption (numeric: from 1 – very low to 5 – very high)
  - 29 health – current health status (numeric: from 1 – very bad to 5 – very good)
  - 30 absences – number of school absences (numeric: from 0 to 93)
- Target attributes
  - 31 G1 – first period grade (numeric: from 0 to 20)
  - 31 G2 – second period grade (numeric: from 0 to 20)
  - 32 G3 – final grade (numeric: from 0 to 20, output target)

Loading and Preprocessing

- Upload the csv files into colab environment
  - student-mat.csv and student-mat-por
- Load ‘student-mat.csv’
  - pandas.read_csv()

Uploading Files to Colab

- Uploading from local computer

```
from google.colab import files


uploaded = files.upload()
for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length}
bytes'.format(name=fn, length=len(uploaded[fn])))
```

- Mounting google drive

```
from google.colab import drive


drive.mount('/content/gdrive')
```

Preprocessing

- Convert the categorical attributes into numerical attributes by one-hot encoding

  Ex) 'Male'/'Female'/'Unknown' // categorical values

  ➔ (1, 0, 0) / (0, 1, 0) / (0, 0, 1) // numerical values

  - pandas.DataFrame.get_dummies()

- Outlier handling
  - Check outliers by pandas.DataFrame.boxplot()
    - Use the *figsize* parameter to display the boxplot charts in appropriate size
  - Remove outliers
    - https://datascience.stackexchange.com/questions/54808/how-to-remove-outliers-using-box-plot

- Handle missing values (NaN)
  - Randomly choose 5 samples and replace their 'traveltime' by np.nan.
    - Check the result by displaying the data.
  - Apply one of the techniques introduced in the following document.
  - https://m.blog.naver.com/youji4ever/221791455668

Predict G3 using PyTorch

- Predict math score from the input attributes
  - Separate the data into a training set (80%) and a test set (20%)
  - Use the *test_size* parameter of train_test_split()
- Design choices for the neural networks (MLP)
  - # of layers
  - # of nodes on each layer
  - Activation functions
- Design choices for training
  - Batch-size
  - # of steps or epochs
  - Loss function
  - Optimizers, learning rates, other hyper-parameters

Using Neural Networks on PyTorch

1. Define a network model
   - Define a network class inheriting Module
   - Override two methods __init__() and forward()
2. Prepare data
   - Use DataLoader
3. Train the model
   - Repeat
     - Forward propagation // computes output
     - Backward propagation // computes gradient
     - Update weights // update weights using gradient
4. Evaluate / use the model