Xiaowen (Angel) Zhang
Prof. Sable
ECE 469 AI-IS

Neural Network Project Report

Link to the project repository: https://github.com/angelzxw/Neural-Network

Instructions and Explanations

- There are three runnable programs created in this project, and they are:
  1. a program converting raw data to training/testing sets named, and the source code file is located at /NeuralNet/src/ data_gen_main.cpp.
     The runnable file is located at /NeuralNet/data
     This program takes the raw data and normalized each entry by dividing each original value in the column by the maximum value in the column if the original data range is greater than 1.
  2. a program creating initial neural network profile according to the given number of hidden nodes, and the source code file is located at /NeuralNet/src/init_gen_main.cpp
     The initial weights are generated pseudorandomly between 0 and 1.
     The runnable file is located at /NeuralNet/init
  3. a program enables both training and testing functions for a given neural network and a set of training/trained data. All code files are located at /NeuralNet/src/program
     The runnable file is located at /NeuralNet/NeuralNet

*Note: Since these executable run on my Mac, if there is problem regarding with running these files in Windows, please let me know so that I would write a makefile for you.

- All files related to the training and testing my dataset are located at /NeuralNet/my_dataset.
  This folder would include the following files:
  - student_names.txt //a description of the data set by its creators
  - student-mat.csv //the raw data for Mathematics (mat) student
  - student-por.csv //the raw data for Portuguese language (por) student
  - student_5.init //my initial neural network for the dataset with 5 hidden nodes
  - student_10.init // my initial neural network for the dataset with 10 hidden nodes
  - student_20.init //my initial neural network for the dataset with 20 hidden nodes
  - student_mat.data //pre-pre-process training set, converts data from student-mat.csv file
  - student_por.data //pre-pre-process training set, converts data from student-por.csv file
  - student_mat.train //my pre-processed training set file by my data_gen program
  - student_mat.test //my pre-processed testing set file by my data_gen program
  - student_por.train //my pre-processed training set file by my data_gen program
  - student_por.test //my pre-processed testing set file by my data_gen program

  The following files are the training and testing output files, and they are named acoording to the formats:
  student_NameOfDataset_NumOfHiddenNodes_Epochs_LearningRate.trained or
  student_NameOfDataset_NumOfHiddenNodes_Epochs_LearningRate_result.txt
  - student_mat_20_5000_0.2.trained
  - student_mat_20_5000_0.2_result.txt
  - student_mat_20_1000_0.2.trained

- student_mat_20_1000_0.2_result.txt
- student_mat_20_1000_0.1.trained
- student_mat_20_1000_0.1_result.txt
- student_mat_10_5000_0.2.trained
- student_mat_10_5000_0.2_result.txt
- student_mat_10_1000_0.2.trained
- student_mat_10_1000_0.2_result.txt
- student_mat_10_1000_0.1.trained
- student_mat_10_1000_0.1_result.txt
- student_mat_5_1000_0.2.trained
- student_mat_5_1000_0.2_result.txt
- student_mat_5_1000_0.175.trained
- student_mat_5_1000_0.175_result.txt
- student_mat_5_1000_0.1.trained
- student_mat_5_1000_0.1_result.txt
- student_por_10_1000_0.2.trained
- student_por_10_1000_0.2_result.txt
- All_results.txt //includes all the results from the tests above

Data set Description

- The raw data is from the website: http://archive.ics.uci.edu/ml/datasets/Student+Performance
- **Abstract:** This data set is used for predict students' performance in high school of two Portuguese schools, and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). The total number of instances is 395 for Mathematics and is 649 for Portuguese language. There are 32 input attributes, and one output. As for math, we take 198 examples to be training dataset, and take the rest 198 examples to be testing set. Similarly, for Portuguese language, I take 325 examples to form the training set, and the rest of 324 examples to form the testing set.
- **Attributes information**: There are 33 attributes in total, which include binary, numeric and nominal types. I consider all these attributes to be relevant to a student's academic performace, so I keep all of them in my training and testing sets.
  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 - 5th to 9th grade, 3 - secondary education, or 4 - higher education)
  8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education, or 4 - 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)
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)

- Pre-process the raw data: For binary attributes, I assign with 0 or 1 for different types; for three-level-classification attributes, I assign values of 0, 0.5, and 1; for four-level-classification attributes, I assign values of 0, 0.5, 1, and 1.5; for five-level-classification attributes, I assign values of 0, 0.5, 1, 1.5, and 2. At this point, every attributes have a numeric value.
- To create the training and testing test, I take the pre-processed data into my data_gen program. So the program will take all even entries to be the training set, and the rest will be the testing set. Also the program normalizes the data by dividing the value by the maximum value that column and round the results to 3 decimal places. To be noted, the original output, G3, is a numerical value from 0 to 20, but I changed the data to be a binary classification: I'd like to consider the student to be honored, if the final grade is greater than 10, and then G3 gets reassigned a value of 1; otherwise, it is reassigned to be 0.
- Different parameter sets for the training and the testing results are listed in the file located at /NeuralNet/my_dataset/All_results.txt