Zhaoyan Lin

Sara Sheehan

Special Studies: Machine Learning in Game Design

May 6, 2016

List of Code: -- paths of images and files need reset

* **Src**
  + Fish.java
  + FlappyFish.java -- main application to run
  + GamePanel.java
  + GraphClassification.java
  + GraphGrid.java
  + GraphGUI.java
  + JGraph.java
  + LogisticRegressionApp.java -- ArrayUtils needs import
  + LogisticRegressionCost.java
  + Statistics.java
  + **Lbfgs**
    - Sdrive.java
  + **Optimization**
    - Minimizable.java
  + **Riso.numerical**
    - LBFGS.java
    - Mcsrch.java

Weekly Outline:

* Week1
  + Tutorial on machine learning
    - <http://cs229.stanford.edu/notes/cs229-notes1.pdf>
  + Linear & logistic regression exercises in MATLAB
  + Exercise:
    - <http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression/>
    - <http://ufldl.stanford.edu/tutorial/supervised/LinearRegression/>
* Week 2
  + Softmax regression exercise in MATLAB
  + Exercise:
    - <http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/>
* Week 3
  + Started to convert linear regression code into Java
  + Converted math equations in MATLAB into Java code
  + Used LBFGS code from Professor Sheehan
* Week 4
  + Continued converting cost function and writing the learning function
  + Started to think of how to import data file
* Week 5
  + Tried importing library dl4j, encountered problems with importing in Eclipse. So later changed to directly read from csv files
  + Mnist in CSV
    - <http://pjreddie.com/projects/mnist-in-csv/>
  + Wrote code to read csv and converted it to array
* Week 6 (after spring break)
  + Found out that Logistic regression needed binary labels, but Mnist data included 0-9. Changed to use only 0 and 1 data for learning
  + Started thinking about GUI design
* Week 7
  + Improved the part of reading data. Using ArrayList instead of array, because the size of data was unknown at beginning
  + Wrote a test function for testing the accuracy of learning
  + Designed a simple GUI for drawing 1 or 0 in a 28\*28 arrays of arrays
* Week 8 – 9
  + Added normalization process after the input of data. Wrote a Statistic class for calculating mean and std
  + Kept debugging
    - Tips
      * Check math equations carefully. Write short helping functions to calculate same equations
      * Keep in mind the difference between java and MATLAB. E.g. log function is undefined at 0 in java
      * When seeing NaN for a result, errors are likely related to math.
      * Print the results and compare with the ones in MATLAB. The results are in matrix form in MATLAB, which is easy to examine.
  + Found out the problem in taking the dot product of theta and array of pixels. Should be careful expanding arrays of arrays.
* Week 10
  + Showed the classification result in a panel next to the drawing panel
  + Wrote the result of theta in a csv file to avoid running the learning function every time
* Week 11 to the end
  + Revised the Flappy Bird code
    - Added the panel for drawing and changed the layout
    - Changed the bird to fish and deleted the gravity
    - Changed the listeners to connect to the classification result

To be improved:

The interaction could be more intuitive and interesting. The way of drawing 1 or 0 could be improved to fit the essence of a game. The game application could be tried on touch screens.