

**CSE 5693 Machine Learning, HW1**  
**Due 6:30pm, Feb 5**  
**Canvas: HW1**

1. Written assignment (from textbook) [pdf file]:
  - (a) 1.2 (tic-tac-toe)
  - (b) 1.4
  - (c) from your programming assignment:
    - i. state the learned weight values without a teacher
    - ii. state the learned weight values with a teacher
    - iii. discuss why each of the weight values makes sense or not.
2. Programming assignment: Tic-tac-toe with LMS weight update (Ch1)
  - (a) Use the design from above (1a)
  - (b) Two modes for selecting experience:
    - i. with teacher
    - ii. without teacher (“self-teaching”)
  - (c) Weak opponent (if both players are expert, the game generally ends in a tie):
    - i. When I evaluate your program, I do not try to win using the middle spot unless I need to block to not lose (ie, try to win in the rows/columns in the perimeter)
  - (d) Initialize each weight to be 0.1; use 0.1 (or smaller) as the learning rate
  - (e) Train on at least 20 games
  - (f) Test on at least 5 games for performance evaluation
  - (g) Provide scripts/programs:
    - i. testTeacher:
      - A. train from an input file of games selected by the teacher,
      - B. display the learned weights,
      - C. allow the user to select going first/second to play with the computer until the user stops,
      - D. report win/loss/tie of the user and the computer
    - ii. testNoTeacher:
      - A. train from games generated by the program,
      - B. same items B to D in testTeacher
  - (h) For a human to enter a move, use row (0-2) and column (0-2) numbers:

```
    0 1 2
    0 | |
      -+-+
    1 | |
      -+-+
    2 | |
```
  - (i) Implementation:
    - i. Use C (GNU gcc), C++ (GNU g++), Java, LISP (CLISP), or Python. If you don't have a preference, use Java since it's more portable.
    - ii. Your program should run on code01.fit.edu (linux) \*without\* non-standard packages/libraries (no additional installation of libraries/packages)
    - iii. You might have these modules:
      - A. Experience: select experience (teacher and no-teacher modes)
      - B. Learner: use experience to gain knowledge
      - C. Player: use knowledge and board to decide a move
      - D. Game: ask who to start, display board, allow moves, output win/loss/tie at the end of a game
  - (j) Submission:
    - i. README.txt: what are the different files, how to compile and run your program on code01.fit.edu (linux).
    - ii. source code files
    - iii. input game/data file for the Teacher mode