COMP 4/6740: Introduction to Neurocomputing

Objectives

The course aims to describe models and methods of this alternative to conventional computing that was was re-born in the 1980s, once conventional computers had acquired sufficient power to enable their massively parallel models. The models are based on machine learning algorithms based on data rather than traditional analysis and programming. Applications will be described to a variety of practical problems in data analysis, forecasting, decision making and, more recently, modeling complex phenomena. Students will get hands on experience on software packages to solve a significant problem in a term project of their choice.

Course Description

4/6996. Introduction to Neurocomputing. (3).

Topics include connectionist data-driven AI; Learning algorithms; Least-mean squares; Supervised learning algorithms (perceptrons. backpropagation and its variants, recurrent neural nets); Unsupervised methods (Hebbian, competitive and reinforcement learning); Deep Learning; Computing platforms for neural nets; case studies.
PREREQUISITES: COMP 2150 and MATH 3242, or permission of instructor.

Syllabus

Week 1	Introduction
Week 2	Evolution of neurocomputing
Week 3	Fundamentals of Machine Learning
Week 4	Computational Platforms (e.g., R, Python)
Week 5	Supervised Learning: Perceptrons
Week 6	Feed-Forward Networks
Week 8	Case Studies
Week 9	Recurrent Neural Nets
Week 10	Computational Power
Week 11	Unsupervised Learning: Hebbian and SOMs
Week 12	Unsupervised Learning: Reinforcement Learning
Week 13	Deep Learning
Week 15	Term Project Presentations

Textbooks

Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles. G. Ciaburro, B. Ventakeswaran. Packt Publishing, 2017.

Alternate

Neural Networks: A systematic Introduction, Raul Rojas. Springer-Verlag, 1996.

Deep Learning, I. Goodfellow, Y. Bengio, A. Courville. MIT Press, 2016.

Evaluation

Four homeworks (40%), Term Project Reports (40%) and Progress Reports (20%). Students taking the class at the 6000 level are expected to engage in more encompassing projects, as well as to turn in solutions of a higher quality. Letter grades will be assigned on total scores as +**A-** ≥ 85% ≥ +**B-** ≥ 73% ≥ **+C-** ≥ 60% ≥ **D** ≥ 54% ≥ **F**. Plus/minus grading will be used based on significant performance differences on the project within each range.

Plagiarism (cheating behavior) in any form is unethical and detrimental to proper education and will not be tolerated. All work submitted by you the student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. Plagiarism is incurred when any part of anybody else's work Is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but appropriate references must be included for the materials consulted, and appropriate citations made when the material is taken verbatim. By taking this course, you agree that any assignment turned in may undergo a review process and that the assignment may be included as a source document in Turnitin.com's restricted access database solely for the purpose of detecting plagiarism in such documents. Any assignment not submitted according to the procedures given by the instructor may be penalized or may not be accepted at all.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor's discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the University Judicial Affairs Office for further disciplinary action. For further information on U of M code of student conduct and academic discipline procedures, please refer to http://www.people.memphis.edu/~jaffairs/">http://www.people.memphis.edu/~jaffairs/.