ECE 613 Computational Intelligence Methods for Data Analysis
Design Elective
Spring Semester 2014
Tue, Thu 9:30-10:45am, Lutz 306, Ph. 852-6314
Instructor Jacek M. Zurada, Lutz 405, jacek.zurada@louisville.edu,

ECE 613 Computational Intelligence Methods for Data Analysis, Credit 3. Prerequisite: Graduate/Professional Standing. Foundations of neural processing, fuzzy systems, genetic algorithms, and machine learning. Theory and application of Principal and Independent Component Analysis. Applications for classification, clustering, visualization, data mining, rule generation, and expert systems.

Prerequisites (by Topic):

Matrix calculus, introduction to optimization

Textbooks:

P.N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Pearson 2006 (not required to purchase)

References:

M. Berthold, D. J. Hand (Eds.), *Intelligent Data Analysis: An Introduction*, Springer 1999
D. H. Ballard, *An Introduction to Natural Computation*, MIT Press 1997
J. M. Zurada, *Introduction to Artificial Neural Systems*, PWS, 1992
I.H. Witten, E. Frank, *Data Mining,* Morgan Kaufmann, 2000
Hand-outs, papers, tutorials

Objectives:

Course will provide students with foundations of neural processing algorithms, fuzzy logic-based reasoning, evolutionary techniques and elements of machine learning for data analysis. It includes design assignments as part of homework and of class project involving student-written software, or use of existing tools such as MATLAB toolboxes (see Computer Use for details).

Course Learning Outcomes: Students who complete this course will be able to:

1. Analyze multilayer perceptron networks
2. Design a perceptron-based pattern classifier or regression analyzer
3. Perform clustering of data with unsupervised training, such as Winner-Takes-All (WTA)
4. Design multidimensional data reduction through Principal Component (PC) learning, or SOFM or Hebbian learning software
5. Design data projection/clustering/visualization software interface
6. Design a decision tree for machine learning-based classification tasks
7. Apply fuzzy sets approach to approximation and logic rules extraction
8. Apply linguistic descriptors as examples of fuzzy variables and design a simple fuzzy controller
9. Design a tool that solves a minimization problem with a genetic algorithm
10. Perform Independent Component Analysis
11. Understand and apply Nonnegative Matrix Factorization
12. Design a Particle Swarm-based solution of an optimization problem
13. Design a software tool for a problem-solving engineering task with one of the computational intelligence algorithms
14. Respond to need of life-long learning and use educational resources to master a selected topic of interest

Topics Covered by Class Schedule (class count in parentheses):

1. Spatial and temporal data representation, basic statistics, statistical inference (2)
2. Learning of a single neuron (perceptron, delta, Hebbian, WTA learning) (2)
3. Supervised learning of multilayer networks: EBP algorithm and its application for classification, function approximation, prediction, expert systems, logic rule extraction (3)

4. Design of data dimensionality reduction through Hebbian or PC learning (2)
5. Competitive learning for clustering, learning vector quantization (1)
6. WTA learning with geometrical neighborhoods, Kohonen maps (2)
7. Design of decision trees for classification and regression (8)
8. Independent Component Analysis and Nonnegative Matrix Factorization (3)
9. Design concepts of fuzzy sets theory and fuzzy logic: membership functions, unions, intersections, complements, level sets and the decomposition theorem, application to multi-criteria decision making (2)
10. Linguistic variables, fuzzy rules, and inference in fuzzy rule, fuzzy models from data (3)
11. Design of classical genetic optimizer including crossover and mutation operators (5)
12. Design of particle swarm-solver for optimization (2)
13. Hybrid approaches (2 classes)
14. Two quizzes and presentations (6 classes)

Topics Covered by Laboratory Schedule: none

Computer Use:

Work with MATLAB neural networks and fuzzy systems toolboxes, code algorithms in high-level language, generate decision trees for data at hand. Examples of computer-based design projects include design of classifiers, regression, decision trees and customization of heuristic and learning-based optimization tools (as specified by CLOs). Most homeworks and all design projects, including the class design project have computer use component.

Evaluation:

1. Two quizzes, 21% each (42%)
2. Homework/design projects assigned weekly or bi-weekly (22%)
3. Class design project presented/demonstrated (5-10 mins) to the class (14%)
4. Seminar based on literature reading focused on a topic of class project 3) (20 mins), PPT, MM or software presentation to class will be graded, no paper necessary (14%)
5. Class attendance (8%), 1% subtracted for each unexcused absence up to 8 points

Grading Scale:

[96, 100] = A+	[92, 96) = A	[88, 92) = A-	[84, 88) = B+	[80, 84) = B	[76, 80) = B-
[72, 76) = C+	[68, 72) = C	[64, 68) = C-	[60, 64) = D+	[56, 60) = D	[52, 56) = D-
[00, 52) = F					

Contribution of Course to Meeting the Professional Component:

Engineering Science: 1 credit or 33%
Engineering Design: 2 credits or 67%

Relation of Course to Program Outcomes:

This course supports the attainment of Program Outcomes 1 (apply science, math & engineering), 2 (design & conduct experiments), 3 (design components, devices, & system), 5 (identify and define problems in EE and solve them), 7a&b (effective oral & written communication), 9 (life long-learning), and 11 (use of techniques, skills, & tools of modern engineering).

Academic Integrity

Students are reminded of academic integrity. Collaboration and team efforts are only permitted in 3), other assignments are to be done as individual efforts. Cheating will not be tolerated, and academic penalties will be imposed if cheating is detected.

Fore more information of general nature please refer to the Code of Student Rights and Responsibilities available at http://louisville.edu/dos/policies-and-procedures/code-of-student-rights-and-responsibilities.html

Prepar**ed by:** J. M. Zurada **Date:** September 2006, rev. Feb 14,
Apr.1, 2009, Nov 14, 2009, Jan 8, 2010, October 17, 2011