CPSC 636: Neural Networks
Spring 2006
Time: TR 09:35AM-10:50AM, **Room**: HRBB 126

Instructor: Ricardo Gutierrez-Osuna
Office: 520A HRBB
Office Hours: TR 10:55-11:55 AM
Phone: 979.845.2942
Email: rgutier@cs.tamu.edu
URL: http://faculty.cs.tamu.edu/rgutier/

Catalog Description: Basic concepts in neural computing; functional equivalence and convergence properties of neural network models; associative memory models; associative, competitive and adaptive resonance models of adaptation and learning; selective applications of neural networks to vision, speech, motor control and planning; neural network modeling environments.

Prerequisites: Math 304 and 308, or approval of instructor.

Textbook:
- S. Haykin. Neural Networks, a Comprehensive Foundation, 2^{nd} ed., Prentice Hall, 1999.

Recommended:
- D. Hanselman and B. Littlefield, Mastering MATLAB 7, Prentice Hall, 2005.

References:
- J. Hertz, A. Krogh and R. G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, 1991.
- J. C. Principe, N. R. Euliano and W. C. Lefebvre, Neural and Adaptive Systems. Fundamentals through simulations. Wiley, 2000.
- C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

Course Objectives: The objectives of this course are to:
- Introduce the fundamental principles of connectionist systems: network architectures, learning paradigms, and neuron models.
- Provide the students with a toolbox of connectionist methods that can be used to solve real problems

Course Outline

- Introduction (2 lectures)
  - Neural network architectures
  - Learning methods
- Single-layer networks (2 lectures)
  - The least-mean-squares algorithm
  - Perceptron learning
- Multi-layer perceptrons (4 lectures)
  - The back-propagation learning algorithm
  - Learning heuristics with MLPs
  - Advanced learning algorithms

- Temporal processing

- Radial basis functions (3 lectures)
  - Function interpolation
  - Hybrid learning methods
  - Orthogonal least squares

- Ensemble learning (2 lectures)
  - Ensemble averaging: boosting
  - Mixtures of experts

- Self-organization (2 lectures)
  - Hebbian learning and principal components
  - Learning vector quantization
  - Kohonen networks

- Information-theoretic models (2 lectures)
  - Introduction: entropy, mutual information
  - Independent components analysis

- Advanced dimensionality reduction methods (2 lectures)
  - Geometric methods: LLE, ISOMAP, NMF, MDS
  - Kernel-based methods for PCA and LDA

- Dynamic programming (2 lectures)
  - Markovian decision processes
  - Reinforcement learning and Q learning

- Neurodynamics (4 lectures)
  - Introduction to dynamical systems
  - Firing rate-models (Hopfield, Grossberg)
  - Spiking models (integrate and fire, Hodgkin and Huxley)
  - Echo state networks

Grading: The course grade will be the weighted sum of four grades. Grading will be straight scale (90-100 A, 80-89 B, 70-79 C, 60-69 D, below 60 F). These numeric thresholds may be lowered due to clustering, but will not be raised.

- **Homework**: There will be three homework assignments, distributed every 2-3 weeks during the first part of the semester. Homework assignments will emphasize the implementation of material presented in class. *Homework assignments must be done individually*.
- **Tests**: There will be a midterm exam and a final exam. All tests will be closed-books, closed--notes. One double-sided, hand-written sheet (8.5 x 11") will be allowed. Tests will have an emphasis on new material from the class notes or the reading assignments.
- **Project**: The last part of the semester will be dedicated to a term project. Students are encouraged to propose projects related to their own research. The projects can be performed individually or in groups of up to three people. Projects will be graded by their content (75%) and the quality of a classroom presentation (25%) at the end of the semester. G*rading criteria for the project presentation and final report are available in the course webpage.*

	Weight (%)
Homework	30
Project	30
Midterm	20
Final Exam	20

Homework submissions. Homework assignments are due at *09:35 AM* on the due date. Electronic material will be submitted with the "*turnin*" utility at https://csnet.cs.tamu.edu; hardcopies will be submitted directly to the instructor. Email submissions will not be accepted. Unless instructed otherwise, you are expected to write your own implementation of the algorithms; in case of doubt please consult with the instructor. Always acknowledge the source of any external libraries or toolboxes that you use for your work.

Late submissions. Late submissions (i.e., as flagged by csnet) will receive a 10% penalty on the total grade of the assignment. The penalty will increase by an additional 10% every 24 hours. Hardcopies of late submissions must be *date and time stamped* by the staff in the Computer Science main office. An assignment is considered submitted when ALL components of the assignment have been submitted; e.g., late submission of one problem in a homework will cause your entire homework to be considered as a late submission.

Missed Tests: Missed tests can only be made up in case of emergency or work conflicts, and will require supporting documentation. Whenever possible, these issues should be discussed with the instructor prior to the conflicting date.

Collaboration vs. Academic Dishonesty: Students are encouraged to exchange ideas and form study groups to discuss the course material, prepare for homework assignments and tests. However, discussions regarding homework assignments should be kept at the conceptual level. Scholastic dishonesty will not be tolerated in homework assignments, tests or projects. For a list of examples of scholastic dishonesty see Section 20 of the TAMU Student Rules (http://student-rules.tamu.edu/).

Academic Integrity Statement
"*An Aggie does not lie, cheat, or steal or tolerate those who do.*" Please review the Aggie Honor Code and Honor Council Rules and Procedures at http://www.tamu.edu/aggiehonor.

Course Schedule

Week/day		Topic	Assignments / activities
1	1/17	Introduction to neural networks	
	1/19	Learning in neural systems	
2	1/24	Least-mean-squares	
	1/26	Perceptron learning	
3	1/31	The back-propagation algorithm	Homework 1 assigned
	2/2	Learning heuristics for MLPs	
4	2/7	Advanced learning techniques	
	2/9	Temporal processing	
5	2/14	Regularization networks	Homework 1 due
	2/16	Learning in RBF networks	Homework 2 assigned
6	2/21	Orthogonal least squares	
	2/23	Ensembles: boosting	
7	2/28	Ensembles: mixture of experts	
	3/2	Hebbian learning and PCA	Homework 2 due
8	3/7	Review/catch-up	
	3/9	**MIDTERM EXAM**	**MIDTERM EXAM**
9	3/14	No class	Spring Break
	3/16	No class	Spring Break
10	3/21	Self-organizing maps	Homework 3 assigned
	3/23	Information-theoretic principles	
11	3/28	Independent components analysis	
	3/30	Kernel methods for dimensionality reduction	
12	4/4	Geometric methods for dimensionality reduction	Homework 3 due
	4/6	Introduction to dynamic programming	
13	4/11	Reinforcement learning	Project proposals due
	4/13	Introduction to dynamical systems	
14	4/18	Associative memories	
	4/20	Spiking neuron models	
15	4/25	Echo state networks	
	4/27*	Review/catch-up	
16	5/2	Final presentations	Final presentations
	5/4	No class	
	5/5	**FINAL EXAM (HRBB 126; 12:30-2:30 pm)**	**FINAL EXAM**

* *On travel*