

Instructor: Robert R. Snapp, 353 Votey, 656-0735, snapp@cs.uvm.edu.

Office Hours: TBA; or by appointment.

Lectures: MWF, 3:35 - 4:25 p.m. in 367 Votey.

Web Page: www.cs.uvm.edu/~snapp/CS256/

Description: Artificial neural networks, their computational capacity and limitations, and the algorithms used to train them. Statistical capacity, convergence theorems, reinforcement learning, generalization.

Prerequisites: Math 124 (or Math 271), Stat 151.

Readings: There is no official textbook for this course. Instead we will read from a variety of books and articles that are on reserve at the Bailey-Howe Library. Required and recommended readings appear on the *Course Outline*.

Grading Policy: The course grade will be based on assigned homework (30%), two take-home midterm exams (30%), a term project (20%), and class participation (20%). Students taking this course for graduate credit will be required to achieve higher quantitative scores than undergraduates to receive corresponding grades. (See page 34 of the 2000-2002 *Graduate Catalogue*, or the URL, <http://www.uvm.edu/academics/catalogue/?Page=courses/about.html&SM=coursemenu>.) Thus, letter grades will be assigned as follows:

Undergraduate	Graduate	Letter Grade
80 - 100	90 - 100	A
70 - 79	80 - 89	B
60 - 69	70 - 79	C
50 - 59		D
0 - 49	0 - 69	F

Homework: There will be approximately six homework assignments. No credit will be given for late homework, but the lowest homework grade will be dropped. Some homework assignments may include programming exercises. Programs will be graded on correctness and style, i.e., clarity, robustness, and efficiency, and should run in a UNIX environment (see below).

Midterm Exams: Two take-home midterm exams will be given during the semester. The first exam will be given March 4-11; the second, from April 22-26. During these periods, students can only discuss the exams with the instructor. However, students can consult any published references (in print, or on line), and their own course notes (including their own graded homework assignments), and can use any computing device and software.

Course Projects: Each student is required to complete a term project that consists of a critical review of one (or more) scholarly publications related to course topic. Each project contains three parts:

1. On March 4, a 1-2 page, *proposal* is due. The proposal should be typeset using \LaTeX , an essential typesetting program in the public domain for scholarly publications. (\LaTeX and \TeX are available on most EMCF computers, and are distributed with Linux. Free versions are available for Macintosh and Windows computers.) Your proposal should summarize what you have accomplished to date; describe what you intend to accomplish during the remaining nine weeks and any software that you intend to create. (10%);
2. On April 22, a 10-25 page *report* is due. The report should also be typeset using \LaTeX . It should describe your project in detail. The report should be clearly written, and will be graded on organization, clarity, originality, effort, and correctness (including spelling and grammar). (50%)
3. During the final exam period (Thursday, May 12, 12 noon - 3:30 p.m) each student is required to deliver a 20-30 minute *oral presentation* to the class. (40%)

Late Assignments: Late homework and project proposals will be penalized 10% credit each calendar day. (Note: this includes Saturdays and Sundays.) Take-home exams, project reports and

oral presentations will not be accepted after the due date. *Students entitled to special accommodation should notify the instructor by the second week of the semester.*

Computer Accounts: Each student should have an EMCF computer account. Programming assignments should be turned in by electronic mail. Because your assignment may employ several files (and perhaps even subdirectories), you may wish to use the *tar* program to assemble all of your work into a single tar file for submission. Assigned programs will be untarred, recompiled and tested as part of the grading. Please include

a *makefile* (and/or a *read.me* file that describes how to compile and link your programs). Programs should be written in C or C++. All programs should run on a Sun Ultra or Sun Blade workstation. (Sun Ultra 10s are available in Votey 229, and Sun Blade 100s in Votey 369.)

Collaboration: You are *encouraged* to share your knowledge, discoveries, and ideas with other students, even as it pertains to assigned work. However, the work that you turn in, should be your own. *Strong similarity between the solutions or programs of two or more students will be treated as a major violation of academic honesty.*