Introduction

Dr. Alex Williams

COSC 425: Introduction to Machine Learning
Fall 2020 (CRN: 44874)
Welcome!
Interest over time

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitve Science, C-045, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurons-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure.

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the desired state vector of the output units for each state vector of the input units. If the input units are directly connected to the output units it is relatively easy to find learning rules that iteratively adjust the relative strengths of the connections so as to progressively reduce the difference between the actual and desired output vectors. Learning becomes more interesting but more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input-output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer at the top. Connections within a layer or from layers are forbidden, but connections can span layers. An input vector is presented to the input units, then the states of all the intermediate layers are determined by applying equations containing lower layers. All intermediate layers are by their states set in parallel, but different states set sequentially, starting at the bottom upwards until the states of the output units are calculated.

The total input, \( x_j \), to unit \( j \) is a linear function of the units that are connected to \( j \) and on these connections:

\[ x_j = \sum \frac{x_i w_{ij}}{1 + e^{-x_i}} \]

Units can be given biases by introducing an unit which always has a value of 1. The \( w \) input is called the bias and is equivalent to the opposite sign. It can be treated just like any other.

A unit has a real-valued output, \( y_j \), which is the function of its total input:

\[ y_j = \frac{1}{1 + e^{-x_j}} \]

© 1986 Nature Publishing Group

---

WHY IS SO MUCH MEMORY NEEDED FOR DEEP NEURAL NETWORKS?

Memory is one of the biggest challenges in deep neural networks (DNNs) today. Researchers are struggling with the limited memory bandwidth of the DRAM devices that have to be used by today's systems to store the huge amounts of weights and activations in DNNs. DRAM capacity appears to be a limitation too. But these challenges are not quite as they seem.

Computer architectures have developed with processor speeds continuing to gain an edge over memory performance, while the disparity between processor and memory speeds has been steadily increasing over the years. This has led to the introduction of novel optimization strategies. Computer architects have developed tricks to overcome the memory bottleneck and improve efficiency. Standard optimizations include caching, prefetching, and data compression.
Today’s Objectives

By the end of this video, you should:

1. Have a clear vision of expectations for COSC 425.
2. Understand the breadth of COSC 425’s syllabus.
Who am I?

Some time ago ...
B.S.
M.S.
Ph.D
Today

COSC 425: Intro. to Machine Learning
What do I do?

How can we build interactive systems that fundamentally make the nature of work better?
What do I do?

Conversational Agents for Work

Supporting Workplace Detachment and Reattachment with Conversational Intelligence

What’s the first step of this task?

What do you want to work on?

Prepare for Work

Enter Work

Task-Centric

Prepare to Leave

Work

Home

Emotion-Centric

Emotion-Centric

SwitchBot (CHI ’18)

Additional Information is available online.
What do I do?

Mobile Programming

Mercury (UIST ‘19)

Additional Information is available online.
What do I do?

The Future of Work

Amazon Mechanical Turk
What do I do?

The Future of Work

COSC 425: Intro. to Machine Learning
A Future of Intelligence

1. How can SwitchBot disengage proactively? 
   ➔ Sensing the user getting in a vehicle.

2. How can Mercury route a relevant microtask? 
   ➔ Understanding IDE context at the desktop.

3. How can HIT Ranger identify “happy” work? 
   ➔ Knowing work that makes the user happy.

Solution: Collect data ➔ Learn trends.
COSC 425: Introduction to Machine Learning
The University of Tennessee, Knoxville - Fall 2020 (CRN: 44874)

Instructor: Dr. Alex Williams
E-mail: acbottel@utk.edu
Office Hours: By appointment.

Teaching Assistant: Zhihong Li
E-mail: zl19@vols.utk.edu
Office Hours: By appointment.

I. Course Description
Machine learning is concerned with computer programs that automatically improve their performance through experience. This course covers the theory and practice of machine learning from a variety of perspectives. We cover topics such as clustering, decision trees, neural network learning, statistical learning methods, Bayesian learning methods, dimension reduction, kernel methods, and reinforcement learning. Programming assignments include implementation and hands-on experiments with various learning algorithms.

N.B.: COSC 425 is not in the Graduate Catalog. Graduate students are therefore unable to take COSC 425 for graduate credit. (See COSC 522: Machine Learning for graduate credit).

II. Required Materials
This course does not have a required textbook. However, this course does have two general requirements: (1) Students should have access to a computer for completing the course assignments; and (2) Students should have access to a working Internet connection that allows you to engage with the course’s material. Students should contact the instructor immediately if they are unable to satisfy either of these requirements.

III. Course Prerequisites
To enroll in the course, students must have completed: Electrical and Computer Engineering 213 or 217 or Mathematics 223; Mathematics 251 or 257. This is a 400-level computer science course, and it is taught at a level appropriate for seniors in computer science. You will be expected to have the background knowledge of senior CS students and, of course, to be competent, efficient, and effective programmers. In this course, you will be expected to complete programming assignments in the Python programming language.

Course Website + Syllabus
https://tiny.utk.edu/cosc425
Course Objectives

This course has three main objectives:

1. to develop an understanding of the fundamental concepts of modern machine learning;

2. to provide first-hand experience with implementing a breadth of learning algorithms; and

3. to empower students with the ability to evaluate, deploy, and critique learning algorithms.
Course Structure

Lectures

→ Asynchronous unless specified otherwise.
→ Available at 12:01am each “class day”.

Office Hours

→ Reserved Time: 2:00 - 4:00pm T/R)
→ Book online via Course Website.
Course Structure

50% Programming Assignments
→ Up to 5 assignments related to implementation.
→ Must use Python.

20% Homework
→ Up to 6 assignments related to ML articles.
→ Written responses, brief questions, etc.

30% Final Team Project
→ Create a machine learning solution for a “real” problem.
  Final report → Purpose, Approach, Evaluation, Limits
Course Topics

- Decision Trees
- Instance-based Learning
- Perceptron
- Linear Regression
- Probabilistic Modelling
- Linear Methods of Classification
- Support Vector Machines
- Performance Evaluation
- Feature Construction / Selection
- Ensemble Learning
- Interpretable Machine Learning
- Active Learning
- Semi-Supervised Learning
- Unsupervised Learning
- Neural Networks
- Deep Learning
- Adversarial Generative Models
- Structured Prediction
- Reinforcement Learning
- Human-Centered Machine Learning
- The Future of Machine Learning
Course Scope

- COSC 420: Biologically-Inspired Computation
- COSC 425: Introduction to Machine Learning
- COSC 421: Computational Cognitive Neuroscience
- COSC 522: Machine Learning

COSC 425: Intro. to Machine Learning
Spring 2021

Special Topics: Human-AI Interaction

What is “Interaction”? → Where HCI meets AI.

Designing Intelligent Interactivity. → Challenges and opportunities.

Evaluating Intelligent Interactivity. → User studies, walkthroughs, etc.
Course Policies

Communication Policy *
- E-mail + Canvas

Attendance Policy *
- No policy. (You are responsible adults!)

Assignment Submission Policy *
- Hand-in via Canvas by 11:59pm on the due date.

Late Submission Policy*
- 1 Day Late = 10%; 2 Days Late = 20%, etc.

*See Syllabus for details.
Course Policies

Academic Integrity & Plagiarism *
→ See Hilltopics Student Handbook.
→ Do not plagiarize, copy, etc.

Civility Policy *
→ I reserve the right to dismiss you from Zoom.

Accommodations for Disabilities *
→ Contact Office of Disability Services
→ (865-974-6087)

*See Syllabus for details.
Course Policies

Accommodations for Change Related to COVID-19

→ **Personal Change:** Consult with me immediately.*

→ **University Change:** I will try to keep you informed.

*See Syllabus for details.

Other concerns (e.g. “absences”): Consult the Dean of Students (https://dos.utk.edu).
Next Time

What is Machine Learning?

Friday, 2:15pm. @ https://tennessee.zoom.us/j/6294683300