EE/CS 228: Introduction to Deep Learning

Winter 2026

Lecture Times: 9:30 am – 10:50 am on Mondays and Wednesdays
Location: Materials Science and Engineering Room 103

Instructor: Yinglun Zhu
Contact Info: yzhu@ucr.edu
Office hour: By appointment

TA: Bowen Zuo (bzuo002@ucr.edu)
Office Hours:
    4:00 pm – 5:00 pm on Thursdays via Zoom
    1:00 pm – 2:00 pm on Fridays via Zoom
    Zoom link: https://ucr.zoom.us/j/8090428745

Course purpose: This course covers the fundamental ideas behind deep learning and provides an exposition to contemporary machine learning algorithms.

Credits and type:
4 Units
Lecture: 3 hours
Research (outside): 3 hours

Course Description:
Explores fundamentals of deep neural networks and their applications in various machine learning tasks. Includes the fundamentals of perception, approximation, neural network architectures, loss functions, and generalization. Addresses optimization methods including backpropagation, automatic differentiation, and regularization. Covers non-standard problems including autoencoders, weak supervision and probabilistic models. Presents applications in machine learning/computer vision/natural language processing.

Prerequisite(s): Understanding of linear algebra, multivariate analysis, and probability, experience with Python.

Tentative Class Schedule (subject to change):

week	Tuesday	Thursday	assigned	due
1	Introduction	Perceptron		
2	Training	Classification	PS1	
3	Optimization I	Optimization II	PS2	
4	Convnets	Architectures	PS3	PS1
5	Optimization III	Overparameterization		PS2, Proposal
6	Generalization	Regularization	PS4	PS3, Midterm
7	Transformers	NLP applications	PS5	
8	Self-supervision	Pre-training		PS4

9	Foundation Models	Recent LLM advances		PS5
10	Presentation I	Presentation II		Final project

Problem Sets (PS) contain technical problems or coding projects related to class content. Due dates will be specified in Canvas.

Course Work:
a) Five assignments will be due during the quarter, each due on Friday through Canvas (eLearn). **See Canvas for instructions of assignment submissions.**
b) The class has no final but one midterm to assess your understanding of course content.
c) You will complete a final project applying the knowledge you acquired in this course.

Grading:
a) Assignments 30%
b) Midterm exam 30%
c) Final project 40% (GPT help allowed, **but need to specify**)
- Proposal 5%
- Presentation + Demo 15%
- Report 20%

Final project:
The final project is expected to demonstrate a non-trivial application of deep learning techniques. The project should be done in groups; **the possible group sizes are 3-5 students**. The project topics are flexible however project must include software implementation and demo. For instance

- Students can come up with a new application idea and develop the associated software product by designing a deep learning model.
- Students can design a new algorithm improving on state-of-the-art techniques and demonstrate its benefit on machine learning tasks.

For both directions, you need to compare against existing methods/baselines!

The project proposal is 1 page. Instructor will provide feedback during proposal phase. Week 10 will be dedicated to presentations. Each presentation is expected to be around 10 to 15 minutes depending on the number of groups. Grade will be a function of presentation quality as well as how well demo works. Project report will describe the design of the project, and detail the intellectual and practical merit; **final project should be 6-8 pages long (excluding appendices).**

- Proposal and final report **must follow NeurIPS format** (see here https://neurips.cc/Conferences/2024/CallForPapers).
  - o Educate yourself about how to use LaTex (e.g., via overleaf tutorial https://www.overleaf.com/learn/latex/Tutorials)!
- **There should be two dedicated sections describe (i) where GPT is used, and (ii) the contribution of each group member.**
- If you grab a pre-existing project (e.g. GitHub repo) and pass it as your own, **you will fail the class.**

- You are allowed to use external code helpers such as CoPilot or ChatGPT. Note that the final grades are curved and everyone has access to same resources.

Late policy:
Late days are rounded up, e.g., 28 hours late accumulated 2 late days.

- Assignments: Each late day leads to 20% grade off on top of your received grade. You will receive a 0 grade if your submission is late for 3 or more days.
- Proposals/final reports: **Late submissions are not allowed.**

Python:
All software development will be in Python, a programming language which is commonly used for artificial intelligence projects with the help of libraries like PyTorch and Tensorflow. Competency in general programming and debugging is a pre-requisite.

We strongly encourage you to use **Python 3** (versus Python 2) so that if code is provided on an assignment there are no compatibility issues. This also helps with debugging if everyone is on the same version.

Python tutorial: https://www.learnpython.org/
PyTorch tutorial: https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
Tensorflow tutorial: https://www.tensorflow.org/tutorials

Academic Integrity: Please carefully review the UCR academic integrity policies and procedures (https://conduct.ucr.edu/policies/academic-integrity-policies-and-procedures). Violations of these policies and procedures can lead to **failure in the course**.

Textbooks and Related Materials:
Textbook:
Deep learning: Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT press, 2016. [free version available online at https://www.deeplearningbook.org/]

Recommended sources:
a) *Elements of Statistical Learning:* J.H. Friedman, R. Tibshirani, and T. Hastie, [available as a free pdf from http://statweb.stanford.edu/~tibs/ElemStatLearn]
b) *All of Statistics:* L.A. Wasserman
c) *Foundations of machine learning*: M. Mohri, A. Rostamizadeh, and A. Talwalkar, MIT press, 2018.

Tentative Syllabus (Related chapters from the Deep Learning book are listed.)
Week 1
introduction to deep learning, history, multilayer perceptron, universal approximation
Chapters 5.1, 5.2, 6.1, 6.3

Week 2
neural network training, empirical risk minimization, loss functions, multiclass classification, gradient descent
Chapters 5.4, 5.7, 5.11, 6.2

Week 3
backpropagation, chain rule, vector formulation, stochastic gradient descent, minibatch, software packages, automatic differentiation
Chapters 5.9, 6.3, 6.5, 8.1

Week 4
convolutional neural networks, weight sharing, CNN architectures, residual networks, vision applications
Chapters 9, 12.2

Week 5
adaptive optimization techniques, initialization, over-parameterization, loss landscape
Chapter 8

Week 6
generalization, bias/variance tradeoff, regularization techniques, early stopping, dropout, batch normalization
Chapter 5.3, 5.4, 7.1-7.4, 7.8, 7.12

Week 7
recurrent neural networks, LSTM, Exploding/vanishing gradients, attention, applications
Chapter 10, 12.3

Week 8
pre-training, transfer learning, multi-task learning, semi-supervised learning, applications
Chapter 7.5, 7.6, 7.7, 15.2

Week 9
autoencoders, dimensionality reduction, probabilistic approaches, generative adversarial networks
Chapter 14, 20.1, 20.2, 20.3, research papers on generative adversarial networks.

Week 10
final project presentations