CPE 486/586: Deep Learning for Engineering Applications

01 Introduction to Deep Learning: Paving pathways to AI

Spring 2026

Rahul Bhadani

About Me

Rahul Bhadani
Assistant Professor
Electrical and Computer Engineering
The University of Alabama in Huntsville
Office: 217-H
Email: rahul.bhadani@uah.edu
Web: https://rahulbhadani.github.io/

Research Interests

Cyber-physical Systems, Intelligent Transportation, Connected-and-Autonomous Driving, Applied machine learning, Quantum Information Science

Course Meeting and Office Hours

Lecture:
Tue/Thur 4:20 PM - 5:40 PM

Location:
ENG 240

Office Hours:

- Mon-Thur 2:00 PM - 3:30 PM

Instructor Email: rahul.bhadani@uah.edu

Textbooks

Reference(s)

The Matrix Calculus You Need For Deep Learning. Terence Parr, Jeremy Howard.
https://arxiv.org/abs/1802.01528.

Mathematical theory of deep learning . Philipp Petersen, Jakob Zech.
https://arxiv.org/abs/2407.18384.

Learning Deep Learning. Magnus Ekman.
Addison-Wesley Professional.
ISBN-10: 0-13-747035-5, ISBN-13: 978-0-13-747035-8

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Aurelien Géron.
O'Reilly Media, 2nd Edition, 2019.
ISBN-10: 3031046471, ISBN-13: 978-3031046476

Neural Networks Theory. Alexander I. Galushkin .
Springer, 2007.
ISBN-10: 9783540481249, ISBN-13: 978-3540481249

Optimal Transport: A Comprehensive Introduction to Modeling, Analysis, Simulation, Applications. Gero Friesecke .
SIAM Publication, 2024.
ISBN:978-1-61197-808-7

Additional textbook references will be provided as we proceed.

Grading

Grading Scale	
Percentage	**Grade**
90% - 100%	A
75% - 89%	B
60% - 74%	C
45% - 59%	D
0% - 44%	F

The percent score will be rounded to the nearest integer before assigning the final grade.

Score Breakdown

CPE 487

Homework:	30%
Quizzes:	5%
Attendance/In-Class Participation:	5%
Mid-term Exam:	15%
Scribe:	10%
Three-page Paper Commentary:	5%
Final Exam:	30%

CPE 587

Homework:	25%
Quizzes:	5%
Attendance/In-Class Participation:	5%
Mid-term Exam:	15%
Scribe:	10%
Three-page Paper Commentary:	5%
Final Exam:	30%
Project Report:	5%

Homework Policy

- Each late submission will be penalized by 10% per day for up to 5 days maximum, thereafter, if later, one will receive 0 credit.
- Solution to homework will be posted 10 days after the due date.

Classwork

- Each lecture may be followed by in-class problem-solving that students will turn in the next lecture day. If you miss the lecture day (either the day it is handed to you, or the day you need to turn in), you will not receive any credit.
- There will be intermediate small tests to assess your skills based on lectures and the classwork. This portion will count towards your classwork credits.

Scribe

Scribe will include one student preparing a chapter-style lecture notes in Latex on specific topic(s) that is a part of the syllabus and submit to the Canvas by the second last week of the semester.

The instructor will provide a Latex template using which you will be writing the chapter. It should include appropriate references to any textbook or paper referred. Please refrain from using AI-generated texts for writing your scribe.

Three-page Paper Commentary

You will be assigned research papers to read frequently throughout the semester. You are required to submit a three-page commentary on the paper in IEEE conference format (template will be provided). The commentary should entirely be your own and not an AI-summary as the purpose is to broaden your own understanding and not an AI's understanding. Commentary is supposed to include mathematical formulation, model structure, and other scientific notation. Your commentary can be supplemented with python scripts, jupyter notebook, additional running examples and dataset for an extra credit of 5%.

Project Report

You will choose a deep learning method of your choice to solve an engineering problem in your domain with in-depth analysis and scientifically conclusive results supported by code, and plots and submit a report in 6-10 pages in NeurIPS format along with codebase and dataset used.

Attendance Policy

- Must attend all lectures.
- Two unexcused absences permitted.
- No option to make up for classwork.

Exam Schedule

Exam Dates		
Exam	**Date**	**Time**
Mid-Term 1	Tuesday, March 03, 2026	4:20 PM to 5:40 PM
Final Exam	Thursday, April 30, 2026	3:00 PM to 5:30 PM

Tentative Absence

1. Jan 13, Jan 15
2. April 14 (likely may not happen, I will update on that)

Tentative Topics

- Crash Course on Machine Learning: Numpy, Pandas, Scikit-learn, PyTorch, Regression, Logistics Regression, Dimensionality Reduction, Support Vector Machine, Clustering, Setting Up Development Environment **(1 Lecture)**
- Introducing Deep Learning – Neural Networks: Perceptron, Multi-layer Perceptron, Nonlinear Activation Functions, Backpropagation Algorithms, Vectorization and Batch Techniques, Neural Network Layers, Training Neural Networks, Learning Rates and Optimization Techniques in Neural Network, Implementation using PyTorch, Neural Network Systems **(3 Lectures)**
- Deep Learning for Sequence Data: Convolution, Pooling, Convolutional Neural Network (CNN), CNN architectures, Recurrent Neural Network (RNN), Long Short-term Memory (LSTM), Gated recurrent unit (GRU), xLSTM, Transformers, Implementation using PyTorch **(3 Lectures)**
- Generative Adversarial Network (GAN): Adversarial Learning, Generator and Discriminator, Minimax Loss, Wasserstein Loss, GAN Training, Image and Video Synthesis using GAN, Deep Convolution GAN, Cycle GAN, Wasserstein GAN, StackGAN, StyleGAN, Issues in GAN: Model Collapse and Training Instability, Interpretability, Data Privacy and Security, GAN for Voice and Music, Implementation using PyTorch, Applications of GAN in industrial setttings **(1 Lecture)**
- Generative Deep Learning: Variational Autoencoder, Diffusion Model, UNet, Normalizing Flow Models, Energy-Based Models, Multimodal Models **(2 Lectures)**
- Graph Neural Network (GNN) and Geometric Learning: Representation Learning with Graphs, Message Passing Techniques in Graph Neural Network, Graph Convolutional Network, Graph Representation Learning, GNN for Node Classification, Graph Classification, Link Prediction, Application of GNN, Implementation in PyTorch, Geometric Learning and Topological Deep Learning **(2 Lectures)**
- Deep Reinforcement Learning (DRL): Sequential Decision Problems, Markov Decision Process, SARSA, Q-Learning, Policy-based RL, Value-based RL, Actor-Critic Models, Gym Environment, Simulation Environment for RL Training **(2 Lectures)**
- Physics-Informed Machine Learning, Neural ODE/PDE, Operator Learning: Physics-Informed Neural Networks , Loss Function Design, Weak and Variational Formulations, Operator Learning, Training Challenges & Optimization, Discretization-Informed Architectures, Neural Controlled Differential Equations, Hamiltonian & Lagrangian Neural Networks, Dynamical Systems Identification, Observability and Controllability **(3 Lectures)**

Tentative Topics

- Bayesian Deep Learning and Uncertainty Quantification: Bayesian Statistics, Bayesian Deep Neural Network, Markov Chain Monte Carlo , Evidential Deep Learning, Gaussian Processes (GPs) & Neural Kernels, Types of Uncertainty, Conformal Prediction, Conformal Correlation, Optimal Transport **(2 Lectures)**
- Kolmogorov-Arnold Network: Kolmogorov-Arnold Representation Theorem, Network Architecture and Topology, B-splines **(1 Lecture)**
- Federated Learning: Distributed Learning, Privacy-Preserving Techniques and Differential Privacy, Algorithm in Federated Learning, Federated Neural Architecture Search, **(1 Lecture)**
- Adversarial Training, Non-stationarity and Concept Drift: Adversarial Attacks, Adversarial Defenses, Adversarial Distributional Shift, Non-Stationary Environments, Concept Shift, Full Distribution Shift, Statistical Change Detection, Data Distribution Monitoring, Active Learning, IID Assumption Violation, Drift Characterization, Incremental Learners **(1 Lecture)**
- NeuroSymbolic AI: Neural Symbolic Integration, Symbolic Neural Networks, Logical Neural Networks , Differentiable Theorem Provers, Neurosymbolic Architectures & Systems, Symbolic Reasoning, Neural-Guided Symbolic Reasoning, Knowledge Graph Embeddings, Neural Network Verification within Symbolic Frameworks **(2 Lectures)**
- Quantum Machine Learning Quantum Deep Neural Network: Qubits & Quantum States, Quantum Gates & Circuits, Noisy Intermediate-Scale Quantum (NISQ) Era, Quantum Approximate Optimization Algorithm, Quantum Annealing, Quantum Kernel Methods, Parameterized Quantum Circuits, Ansatz Design, Quantum Neuron Models, Quantum Deep Learning Architectures **(2 Lectures)**

Getting ML Specific Help with Python

Where can I get resources to help with Python programming?

- Python for Everybody by Dr. Charles Severance (https://www.youtube.com/watch?v=PKrC027wIUU&list=PLAtoCfxWRIErTozMKGdHmUieMuQDGEoEJ) is a great place to refresh your Python
- Sklearn has some extremely helpful documentation pages (https://scikit-learn.org/stable/index.html)
- Matplotlib: the most used visualization tool in Python: (https://matplotlib.org/stable/tutorials/index.html)
- Polars for Data Analysis: https://docs.pola.rs/user-guide/getting-started/
- Learning PyTorch with Examples: (https://pytorch.org/tutorials/beginner/pytorch_with_examples.html)

How to get help for this course?

- Ask questions in the class without hesitation. No question is silly.
- Utilize office hours to the maximum extent.
- Start your homework as soon as it is posted. The more you delay, the chance of your success will diminish.
- Do additional self-reading related to topics covered in the class.

Remember, you are here to learn the material in this course, and not just pass it.

In-Class Activity

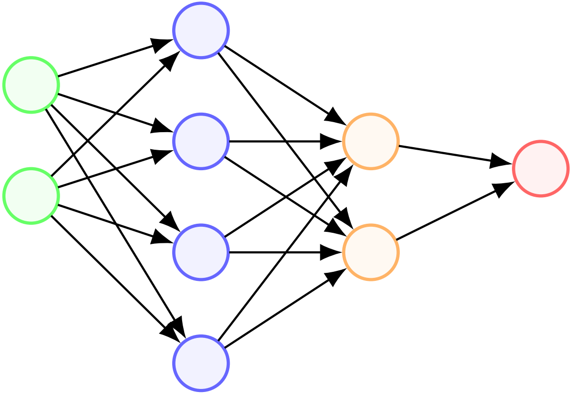
Introduce Yourself

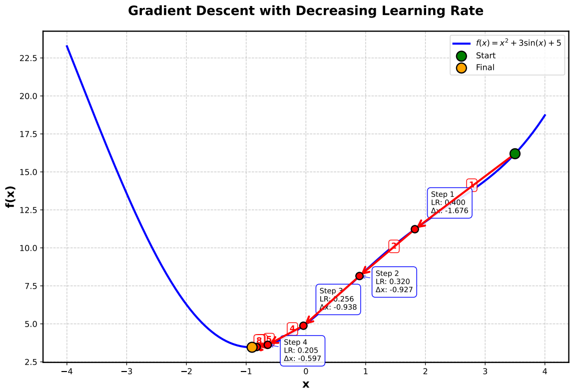
- Why do you want to take this course?
- What are your research intersts? How are you planning to utilize Deep Learning in your research/career?
- Pick a domain of your interest where you will apply deep learning algorithms. Personalized homework problems will be assigned based on your domain. You may take a week to decide, if not today.

Introduction to the Course

What is Deep Learning?

Learning algorithms that rely on neural network architectures with several layers and are primarily based on gradient descent optimization.





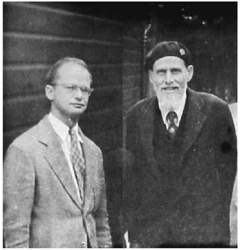
History of Neural Network

1943

Warren McCulloch, Walter Pitts: How neurons work.

McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5, no. 4 (1943): 115-133.

https://jontalle.web.engr.illinois.edu/uploads/410-NS.F22/McCulloch-Pitts-1943-neural-networks-ocr.pdf



McCulloch (right) and Pitts (left) in 1949

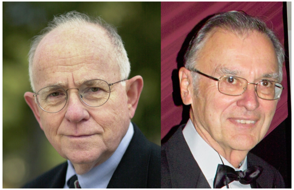
https://www.historyofinformation.com/detail.php?entryid=782

History of Neural Network

1959/1962

Bernard Widrow and Marcian Hoff: ADALINE, and MADALINE: (Multiple) Adaptive Linear Elements

"Associative storage and retrieval of digital information in networks of adaptive neurons" In Biological Prototypes and Synthetic Systems: Volume 1 Proceedings of the Second Annual Bionics Symposium, Springer US, 1962.

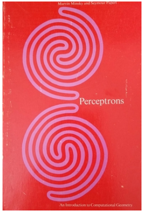


Bernard Widrow (Left) and Marcian Hoff (Right)

1969: Beginning of the first AI Winter

Marvin Minsky and Seymour Papert suggested problem could not be solved using a single-layer perceptron if classes were not linearly separable. They suggested there could not be an extension from the single layered neural network to a multiple layered neural network.

Minsky, Marvin, and Seymour Papert. "An introduction to computational geometry." Cambridge tiass., HIT 479, no. 480 (1969): 104.



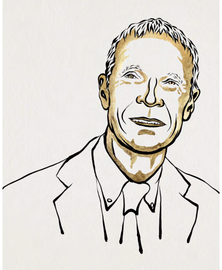
History of Neural Network

1. **1982:** John Hopfield's approach was to create more useful machines by using bidirectional lines. Previously, the connections between neurons was only one way. *Hopfield, John J. "Neural networks and physical systems with emergent collective computational abilities." Proceedings of the national academy of sciences 79, no. 8 (1982): 2554-2558.*
2. **1982:** Reilly and Cooper used a "Hybrid network" with multiple layers, each layer using a different problem-solving strategy. *Reilly, Douglas L., Leon N. Cooper, and Charles Elbaum. "A neural model for category learning." Biological cybernetics 45, no. 1 (1982): 35-41.*
3. **1982:** Japan announced fifth-gen effort on AI, US got FOMO and increased funding that renewed interest in AI.
4. **1986:** Backpropagation by David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams. *Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323, no. 6088 (1986): 533-536.*

The 2024 physics laureates

"for foundational discoveries and inventions that enable machine learning with artificial neural networks"

The laureates built computer systems that lay the foundation that made it possible to train computers to do things like chatting or driving a car.



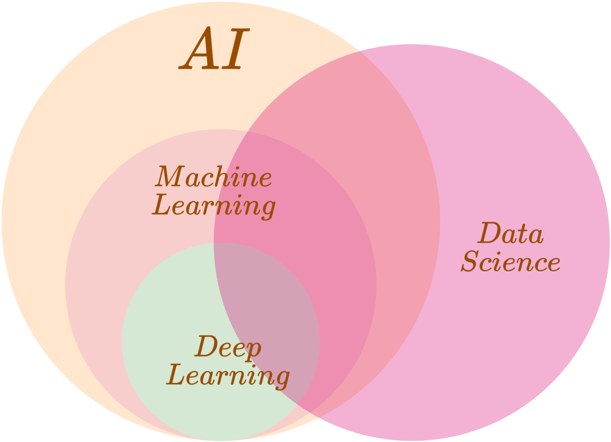
John Hopfield
Born: 1933, USA

Geoffrey Hinton
Born: 1947, United Kingdom

Source: https://www.nobelprize.org/uploads/2024/12/Slideshow_All_NobelPrizes_2024_NobelPrizeLessons.pdf

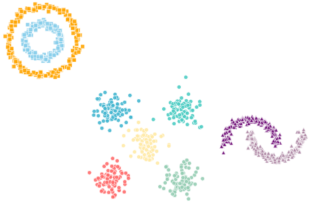
Tasks with Deep Learning

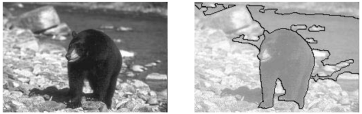
Deep Learning, Data Science, Artificial Intelligence



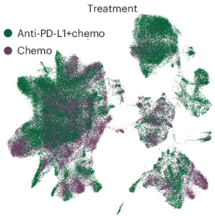
Classification, Clustering, Segmentation

Usually done in unsupervised fashion, but can be semi-supervised or supervised as well.

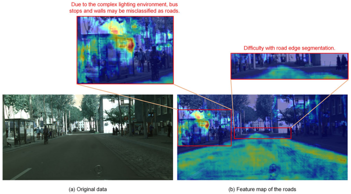




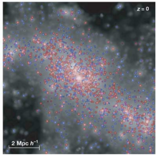
Learning a Classification Model for Segmentation, Xiaofeng Ren and Jitendra Malik



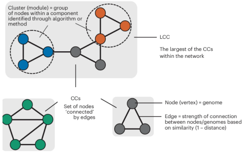
Pertpy: an end-to-end framework for perturbation analysis, Lukas Heumos et al.



A segmentation network for enhancing



Simulations of the formation, evolution and clustering of galaxies and quasars, Volker Springel



Machine learning enables scalable and systematic hierarchical virus taxonomy, Benjamin Bolduc et al.

Regression Problems: predicting continuous values or next token

1. Machine learning-based framework for wall-perching prediction of flying robot https://doi.org/10.1038/s41467-025-67386-0
2. Machine learning models based wear performance prediction of AZ31/TiC composites https://www.nature.com/articles/s41598-025-33417-5
3. Deep learning based real-time prediction of depth of penetration during activated tungsten inert gas welding of 10 mm thick 316LN stainless steel https://www.nature.com/articles/s41598-025-31414-2
4. Comparative analysis of deep learning architectures in solar power prediction https://www.nature.com/articles/s41598-025-14908-x
5. Explainable judgment prediction and article-violation analysis using deep LexFaith hierarchical BERT model https://www.nature.com/articles/s41598-025-32833-x

Types of Learning

Learning Modalities

- **Supervised learning**: Given training data with previously labeled classes, learn the mapping between the data and their correct classes.
- **Unsupervised learning**: Given unlabeled data obtained from an unknown number of categories, learn how to group such data into meaningful clusters based on some measure of similarity
- **Reinforcement learning**: Given a sequence of outputs, learn a policy to obtain the desired output game-playing problems.

What do you need to be successful in this course?

- **Linear Algebra**: Data are represented as vectors, vectors lie in a vector space, . . . , you get the point!
- **Calculus**: A little bit of calculus is needed to understand how optimization problem works!
- **Probability and Statistics**: We need a way to capture uncertainty in our data and models. Probability theory provides us a way to capture and harness uncertainty.
- **Coding/Software**: Not only are you going to be talk the talk in machine learning, but with software you're going to be able to walk the walk.

Prerequisite Slides

1. https://github.com/rahulbhadani/CPE486586_FA25/blob/main/Lectures/Chapter_01_Introduction.pdf
2. https://github.com/rahulbhadani/CPE486586_FA25/blob/main/Lectures/Chapter_02_Tools_For_Machine_Learning.pdf

Setting up Development Environment

Command Line Tools

Required for mastering automation, file operations and system interaction.

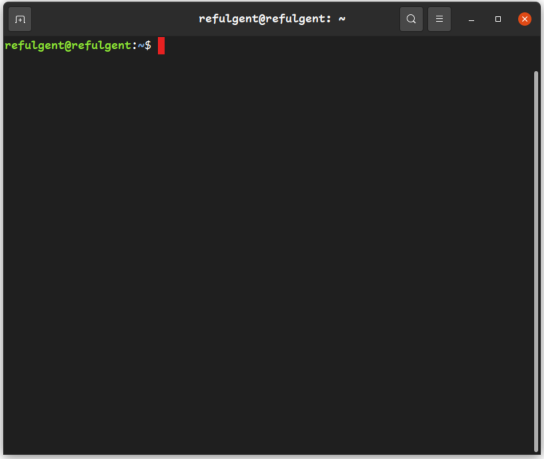


Figure: Linux Command Line Tool

Using Command Line Tools

1. Installed by default on Linux and Mac
2. Windows: recommended option: use Windows Subsystem for Linux. Installation guide: 1. https://youtu.be/GMhV5Uqd8R8, 2. https://youtu.be/NPuIUT_6NeM

Connecting Remotely to a Linux Machine for Development

```
ssh rkb0022@blackhawk.ece.uah.edu
```

Blackhawk machine is a placeholder. From there you can enter into Euler or Gauss machine:

```
ssh rkb0022@euler.ece.uah.edu
```

or

```
ssh rkb0022@gauss.ece.uah.edu
```

Default password (if not already changed by you) is the user's lowercase first and last initial followed by the last 6 digits of their A#.
Using these remote machines are recommended if you don't have GPU-based machine available to you.

Installing UV

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Creating Python Virtual Environment

```
mkdir python_project
cd python_project
uv init .
# install python
uv python install 3.12
# create a virtual environment
uv venv --python 3.12
# activate virtual environment
source .venv/bin/activate
```

Now, you can install some packages

```
uv add numpy matplotlib  scikit-learn  polars torch
```

Running your Python code

```
python example.py
```

VS Code

Many IDEs are available such PyCharm, Atom, VS Code.

For this course, I recommend VS Code.

VS Code supports several extensions for Python, and other necessary tools.

Installing VS Code

Download: https://code.visualstudio.com/download

On Linux, downloaded deb file can be install at command line:

```
sudo dpkg -i code_1.103.1-1755017277_amd64.deb
```

VS Code Extension

Recommended VS Code Extension:

1. Jupyter (Microsoft)
2. Jupyter Notebook Renderers (Microsoft)
3. Python (Microsoft)
4. Pylance (Microsoft)
5. Python Debugger (Microsoft)
6. Python Environments (Microsoft)
7. Remote – SSH (Microsoft)
8. Remote – SSH: Editing Configuration Files (Microsoft)
9. Remote Explorer (Microsoft)

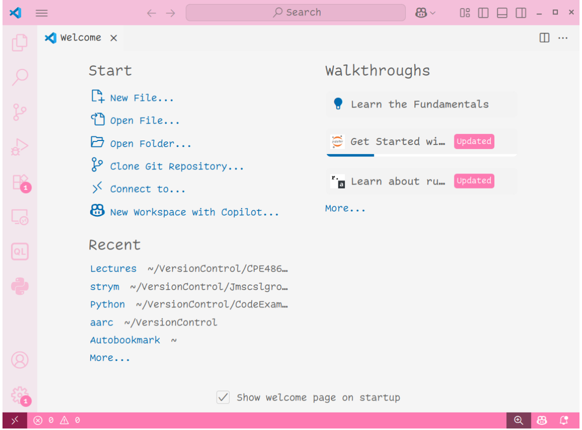
Connecting to Remote Machine in VS Code

When connecting to remote machine on VS Code, environment variables of the remote machine takes into effect.

To connect with `blackhawk.ece.uah.edu` from outside the Engineering building, if need VPN, download from:

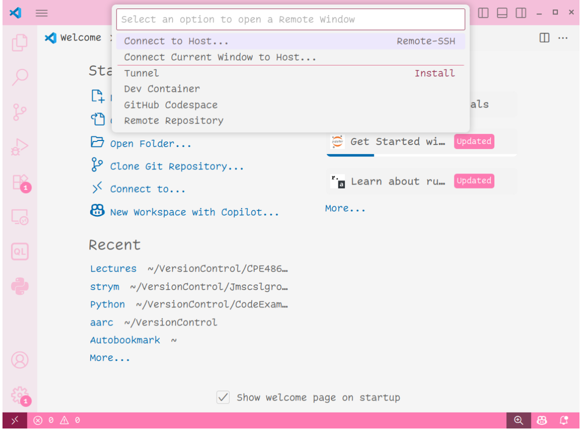
Download VPN: https://chargerware.uah.edu/all-software/secure-access-vpn

Connecting to Remote Machine in VS Code



After connecting to campus VPN, click **Connect to ...**

Connecting to Remote Machine in VS Code



Click **Connect to Host ...**, enter the address `username@blackhawk.ece.uah.edu`. Use your own user name.

The End