ECE 6960-26 Deep Learning for Image Analysis (Spring 2023)

Learning outcomes

- Learn theoretical and practical aspects of deep neural networks and be able to include these tools in formulating solutions to engineering problems [ABET 1] (All assignments and course project)
    1. Learn to build simple neural networks from scratch using Numpy / Python
    2. Design and experiment with complex neural networks using Pytorch / Python
- Develop and conduct experimentation, analyze and interpret data, and use engineering judgment to draw conclusions [ABET 6] (All assignments and course project)
- Acquire new knowledge about challenging data science problems in computer vision and image analysis , design, implement and experiment with solutions using deep learning [ABET 6,7] (Course project)

Course content

- Introduction to machine learning
- Linear models and multilayer neural networks
- Error backpropagation algorithm and neural network training
- Convolution and max-pooling operations; image classification
- Training CNNs and modern architectures
- Object detection, localization, regression
- Upsampling and transpose convolution operations
- Image segmentation
- Unsupervised learning and autoencoders
- Semi-supervised and self-supervised learning
- Transformer models
- Vision/Language models
- Generative adversarial networks

Required background

- Undergraduate level coursework in i) Linear algebra, ii) Calculus, and iii) Probability.
- Working knowledge of Python.
- We will use Numpy, Pytorch, Mayplotlib, Jupyter notebook for assignments. Previous knowledge useful but not required.

Course information:

- Lectures: MW 1:25- 2:45
- Instructor: Tolga Tasdizen, Email: u0301982@utah.edu
- Teaching Assistant: TBA

Resources

- Canvas course page: We will post lecture our slides / notes, links to additional resources and important notifications. Online discussion and questions to instructor and TA. All assignments will be handled via Canvas.
- Textbook: Deep Learning by Goodfellow, Bengio and Courville, MIT Press 2016. Freely available online at https://www.deeplearningbook.org.
- Other resources: Stanford CS231N website http://cs231n.stanford.edu.
  Pytorch website: https://pytorch.org. Tutorials and documentation.

Grading

- Assignments (70 points)
  - You will be expected to implement algorithms in Python using Numpy and Pytorch, analyze your results visually and quantitatively, and produce written summary and analysis.
  - All assignments will be submitted via Canvas.
  - Late assignment policy: You have 6 days grace period in total that you can use as you need (spread them around assignments as you need). 25% of points earned will be deducted for each day after the deadline once your grace period is used. Exemptions will be made only for real emergency situations. Assignments will be due at midnight, submitting past 12 am will be considered late and use a grace day if available.
- Individual Project (30 points)
  - Choose from a list of projects or come up with your own project proposal
  - Python implementation (can use Pytorch or Tensorflow)
  - Project report

Academic Honesty and Student Conduct:

- Copying someone else's work whole or in part on an assignment or project is considered cheating.
- You can discuss assignments with each other, but sharing code is not allowed. Anything turned in must be your own work.
- Copying text, code, image, etc. from a webpage or other source without giving a reference is considered plagiarism which is a form of cheating.
- First instance of cheating will result in a 0 in assignment/project. Second instance will result in failing the course.

College of Engineering guidelines: https://www.coe.utah.edu/students/current/semester-guidelines/
Information about appeals, Americans with Disabilities Act, and other important topics