Introduction to Deep Learning: Bridging Theory and Practice using PyTorch

Sina Moghimi

2024-01-08

Contents

Chapter 1

Syllabus

1.1 Course Description

This course provides a comprehensive introduction to deep learning, guiding you through fundamental theoretical concepts, neural network architectures, and hands-on implementation in PyTorch. Through clear explanations supplemented with practical examples and coding exercises, you will gain expertise in constructing deep neural networks, implementing cutting-edge architectures, training and optimizing models, and deploying deep learning systems. The curriculum covers convolutional neural networks, sequence models, generative adversarial networks, and other key architectures. You will master techniques for training robust models at scale and troubleshooting common pitfalls. By the end of this course, you will have the skills to independently implement, debug, and deploy production-ready deep learning models in PyTorch.

1.2 Learning Objectives

Upon Completion of this course, students will be able to:

- Explain foundational principles of deep learning including gradient descent and backpropagation

- Implement standard feedforward, convolutional, recurrent, and other neural network architectures in PyTorch

- Train deep learning models at scale using GPU acceleration and distributed strategies

- Optimize model training using techniques like regularization, normalization, and dropout

- Troubleshoot issues like over/underfitting, imbalance, and vanishing/exploding gradients

- Leverage pretrained models and transfer learning to boost performance

- Construct custom datasets from scratch and prepare them for use

- Apply deep learning techniques to computer vision, NLP, speech recognition, and other domains [1]

The course will consist of three primary components:

- *Lectures* will focus on theoretical aspects of deep learning.
- *Practical sessions* will involve hands-on demonstrations of best practices in deep learning.
- *Programming assignments and defenses* will help reinforce the learned concepts and provide practical implementation skills.

1.3 Course Prerequisites

- Calculus, Linear Algebra, Statistics, Probability Theory

1.4 Course Instructor

Sina Moghimi

Email: mogimi.s@phystech.edu

Office Hours: Available by request.

1.5 Communication

This page function as the main hub for the course. Discussion threads are available at discussion page for questions related to the course and for addressing coding problems or bugs.

1.6 Lectures

Lectures will be held once a week on Mondays. Lectures notes will be linked on the Lecture Notes after the lecture.

---

[1]This objective will be provided upon successful completion of the prior objectives.

1.7 Practical Sessions

Practical Sessions will be held once a week on Wednesdays.

1.8 Grading

Main Tasks	Points
1. Assignment Neural Network Components	10
2. Assignment Image Classification	10
3. Assignment Music Genre Identification	10
4. Assignment Image Reconstruction	15
5. Assignment Pop Music Generation	15
6. Assignment Image Super-Resolution	15
7. Assignment Music Composition	15
8. Final Exam	10
Class Total	100

Auxillary Tasks	Points
Capstone Project	40
Attendance	10
Auxillary Total	50

1.9 Capstone Project

As an additional task, students are encouraged to begin contemplating their project ideas during the initial weeks of the course. The capstone project serves as an opportunity for students to explore the application of deep learning techniques to interesting topics, such as Computer Vision, NLP, Biology, and Robotics Control. Students may choose to work individually or in groups.

The project for this course will require students to write a substantial amount of code to implement a deep learning model. The official programming language and package for this class are Python and PyTorch. Students are free to choose to work with another language or package, but no support will be provided for that.

In the context of the course, students have the option to utilize their own data for their project, or they may choose from any publicly available dataset.

1.10   Late Assignments

Students are encouraged to submit their assignments promptly. Late assignments will be penalized by deducting 10% of the total score for each day of delay. However, extension requests may be considered on a case-by-case basis, taking into account extenuating circumstances. It is advisable to reach out and discuss any concerns or challenges before the deadline.

1.11   Policies

1.11.1   Collaboration

Students are encouraged to collaborate on homework assignments and may seek guidance from external sources such as reference materials, peers, or the instructor. However, any material obtained from external sources must be properly cited. It is expected that all submitted solutions are individually written and reflect the student's own comprehension of the subject matter at the time of writing. Additionally, Python scripts and plots are considered part of the individual write-up and should be completed independently, allowing for the sharing of ideas but not code.

1.11.2   Academic Integrity and Plagerism

The comprehension and prevention of plagiarism are of paramount importance. Plagiarism, defined as the unauthorized use of another individual's ideas, processes, results, or language without proper acknowledgment, represents a fundamental violation of academic integrity.[2]

By submitting any form of work, whether in print or electronically, individuals are required to adhere to the regulations on plagiarism established by MIPT. Furthermore, by submitting the work, individuals grant MIPT the authority to take measures to verify the authenticity of the submitted material, which may include, but is not limited to, employing a plagiarism detection service and sharing the work with other faculty members.[3]

---

[2]https://www.ox.ac.uk/students/academic/guidance/skills/plagiarism
[3]Elena Bazanova, Academic Writing for Research Purposes, MIPT

Chapter 2

Assignments

2.1 Neural Network Components

- **Purpose**: Understanding the fundamentals of neural networks by building components from scratch.

- **Components**: Layers, activation functions, loss functions, optimizers.

2.2 Image Classification

- **Purpose**: Apply knowledge to design an effective CNN for a real-world task.

- **Challenge**: Achieve at least 85% validation accuracy on CIFAR-10 through architecture design and hyperparameter tuning.

2.3 Music Genre Identification

- **Purpose**: Apply image classification techniques to the musical domain using PyTorch.

- **Challenge**: Design a CNN to identify the genre of a song based on its mel-spectrogram representation.

2.4 Image Reconstruction

- **Purpose**: Understand supervised learning and image reconstruction.

- ***Challenge***: Successfully reconstruct images from the Fashion MNIST dataset using an autoencoder.

2.5   Pop Music Generation

- ***Purpose***: Explore sequence generation in the context of music using PyTorch.

- ***Challenge***: Train an LSTM model on a dataset of MIDI music files and experiment with different loss functions to generate pop music.

2.6   Image Super-Resolution

- ***Purpose***: Explore the use of Generative Adversarial Networks (GANs) and Super-Resolution GANs (SRGANs) for image super resolution, which can enhance the quality of low-resolution images.

- ***Challenge***: Train the GAN and SRGAN models effectively, as they require large amounts of high-quality training data and careful tuning of hyper parameters to achieve good results.

2.7   Music Composition

- ***Purpose***: Generate new musical compositions based on a learned latent representation using PyTorch.

- ***Challenge***: Train a VAE on piano music and experiment with generating novel compositions.

Chapter 3

Capstone Project

The capstone project is a crucial aspect of the course, providing students with a valuable opportunity to apply their deep learning skills to real-world issues. This project serves as a culmination, allowing students to demonstrate their proficiency gained throughout the course and showcase their ability to independently implement, debug, and deploy production-ready deep learning models using PyTorch. The project's comprehensive nature involves the application of deep learning techniques to diverse domains like computer vision, NLP, speech recognition, or music generation. By challenging students to think critically and creatively, the project encourages the development of innovative solutions for complex problems. Functioning as a vital assessment method, the capstone project enables students to exhibit their problem-solving and troubleshooting skills, affirming their readiness to apply deep learning in real-world scenarios. Students have the flexibility to suggest project topics, and depending on class size, collaborative teamwork may be encouraged to better prepare them for the job market. The capstone project, therefore, not only assesses academic knowledge but also emphasizes practical application and the ability to address challenges in a professional context.

Chapter 4

Course Materials

For those who love reading books

- Deep Learning (MIT Press, Ian Goodfellow)
- Deep Learning with PyTorch (Manning Publications, Eli Stevens)

For those who like reading online

- Deep Learning: Link
- Dive into Deep Learning: Link

For those who hate reading but enjoy watching

- MIT Introduction to Deep Learning: Link
- Pytorch Tutorials: Link

You would also love to check

- Explainable AI: Link
- Physics-Informed Neural Networks: Link
- Understanding Deep Learning: Link

Chapter 5

Lecture Notes

Will be updated ...