Deep Learning

Dr. Prodromos Malakasiotis, rulller@gmail.com

Overview

Deep Learning has gained significant attention over the past few years leading to state-of-the-art methods for several tasks concerning text and image processing (e.g., machine translation, object recognition in images, etc.). The course will be a mix of theory and practice, covering the basic deep learning theory and providing examples of how to build deep neural networks in practice with popular Python tools, e.g., Keras, PyTorch, etc.

Key Outcomes

By the end of the course the students will be able to understand the basic theory of deep learning as well as the intuition behind specific deep architectures for specific tasks. They will also be able to use popular tools (Keras, PyTorch) to develop, train and evaluate complex neural networks.

Requirements and Prerequisites

This is mainly a hands-on course. Students will spend a significant amount of time on developing neural networks in Python with PyTorch and/or Keras. The course does not assume any prior experience in Python. However, basic knowledge of programming and computer science concepts is required.

Students will need to bring their laptops in class in order to try out interactively the material being presented. Laptops should have enough RAM (8 GB) to train neural networks on small datasets.

Required Course Materials

There is no required textbook. All course materials will be provided in class and available for downloading.

Books

The following books provide a good supporting material for the topics covered by the course:

- "Deep Learning (Adaptive Computation and Machine Learning)", I. Goodfellow, Y. Bengio, A. Courville, MIT Press, 2016.
- "Neural Networks and Deep Learning", M. Nielsen, 2015 (available on-line at http://neuralnetworksanddeeplearning.com/).
- "Grokking Deep Learning", A. W. Trask, Manning Publications, 2018 (to be published, the first 11 chapters are available)

Software/Computing requirements

The examples of the course will be implemented in Python 2.7, PyTorch (https://pytorch.org/) and Keras (https://keras.io/). No special hardware is required although GPU enabled hardware is preferable.

Grading

This is a practical course; students will be graded on their attendance (10%), participation (20%), and competency to develop neural networks for specific tasks (70%). The latter will be determined by a programming assignment.

Participation

In-class contribution is a significant part of your grade and an important part of our shared learning experience. Your active participation helps me to evaluate your overall performance. You can excel in this area if you come to class on time and contribute to the course by:

- Providing strong evidence of having thought through the material.
- Advancing the discussion by contributing insightful comments and questions.
- Listening attentively in class.
- Demonstrating interest in your peers' comments, questions, and presentations.
- Giving constructive feedback to your peers when appropriate.

Please arrive to class on time and stay to the end of the class period. Chronically arriving late or leaving class early is unprofessional and disruptive to the entire class. Repeated tardiness will have an impact on your grade.

Turn off all electronic devices prior to the start of class. Cell phones tablets and other electronic devices are a distraction to everyone. If the course requires you to use a laptop you will be informed so.

Late Assignments

Late assignments will either not be accepted or will incur a grade penalty unless due to documented serious illness or family emergency. Exceptions to this policy for reasons of civic obligations will only be made available when the assignment cannot reasonably be completed prior to the due date, you make suitable arrangements, and give notice for late submission in advance.

Attendance Requirements

Class attendance is essential to success in this course and is part of your grade. An excused absence can only be granted in cases of serious illness or grave family emergencies and must be documented. Job interviews and incompatible travel plans are considered unexcused absences. Where possible, please notify the instructor in advance of an excused absence.

Students are responsible for keeping up with the course material, including lectures, from the first day of this class, forward. It is the student's obligation to bring oneself up to date on any missed coursework.

Code of Ethics

Students may not work together on graded assignments unless the instructor gives express permission.

Exercise integrity in all aspects of one's academic work including, but not limited to, the preparation and completion of all other course requirements by not engaging in any method or means that provides an unfair advantage. In any case of doubt, students must be able to prove that they are the sole authors of their work by demonstrating their knowledge to the instructor.

Clearly acknowledge the work and efforts of others when submitting written work as one's own. Ideas, data, direct quotations (which should be designated with quotation marks), paraphrasing, creative expression, or any other incorporation of the work of others should be fully referenced. No plagiarism of any sort will be tolerated. This includes any material found on the internet. Reuse of material found in question and answer forums, code repositories, other lecture sites, etc., is unacceptable. You may use online material to deepen your understanding of a concept, not for finding answers.

Please report observed violations of this policy. Any violations will incur a fail grade at the course and reporting to the senate for further disciplinary action.

Course Syllabus

The course comprises five units of three hours each.

Unit 1: Machine learning brush-up, deep learning prerequisites

Theory:

- Supervised learning, unsupervised learning, reinforcement learning.
- Softmax classification, cross entropy.
- Gradient descent, stochastic gradient descent.

Practice:

- Setup and test required tools; virtual environment, Keras, PyTorch.

Unit 2: Basic deep learning concepts

Theory:

- Multi-layered Perceptrons (MLPs), back propagation, vanishing gradient, activation functions.
- Regularization, dropout, batch normalization.
- Advanced optimization algorithms, Adam, Adagrad.

Practice:

- MLPs with Keras and Pytorch.
- Best practices when training and testing deep learning models.

Unit 3: Advanced deep learning I

Theory:

- Convolutional Neural Networks (CNNs).
- CNNs for images.
- Word embeddings, CNNs for text.

Practice:

- CNN for image classification with Keras and PyTorch.
- CNN for text classification with Keras and PyTorch.

Unit 4: Advanced deep learning II

Theory:

- Recurrent Neural Networks (RNNs) for text.
- GRU cells, LSTM cells.
- Bidirectional RNNs, Stacked RNNs.

Practice:

- RNN for text classification with Keras and PyTorch.

Unit 5: Advanced deep learning III

Theory:

- Attentional models.
- Sequence to sequence models.
- Combinations of CNNs and RNNs for image caption generation.

Practice:

- Image caption generation with PyTorch and Keras.