Core Courses Syllabi

AI702 - Deep Learning

Title	Deep Learning
Code	AI702
Loading	4 Credit-hours
Prerequisites	• Basics of Linear Algebra, Calculus, Probability and Statistics • Proficiency in Python
Catalog Description	This course provides a comprehensive overview of different concepts and methods related to deep learning. Students will first learn the foundations of deep learning, after which they will be introduced to a series deep models: convolutional neural networks, autoencoders, recurrent neural network, and deep generative models. Students will work on case studies of deep learning in different fields such as computer vision, medical imaging, natural language processing, etc.
Goal	This course aims to familiarize students with the foundations of deep learning and its application domains. Specifically, students will be able to build, train, evaluate, and improve appropriate deep learning models for different problems.
Contents	This course covers the following major modules: **(I)** neural networks, **(II)** convolutional neural networks, **(III)** recurrent neural networks, **(IV)** reinforcement learning, and **(V)** generative adversarial networks.
Recommended Textbooks	I. Goodfellow, Y. Bengio, A. C. Courville, *Deep Learning*, MIT Press, 2016.
Recommended References & Supplemental Material	Relevant research papers, tech reports, and surveys for each topic, where needed, are identified in the teaching plan ahead. In addition, the following textbook may be useful: P. J. Gibson, A. Deep learning: A practitioner's approach, O'Reilly Media, Inc., 2017.

Teaching Week	Topics
1	**Applied Math and Machine Learning Basics** **Lectures** • A refresher on linear algebra and probability • Machine learning basics **Lab** • Getting started with Python • Practice problems to gain familiarity with Pytorch and Jupyter Notebook.
2	**Introduction to Deep Learning and Neural Networks Basics** **Lecture** • History and cognitive basis of neural computation. • The neural network as a universal function approximator **Lab** • Practice writing vectorized code with NumPy • Implementation and application of K-nearest neighbour algorithm on a toy dataset
3	**Deep Feedforward Networks** **Lecture** • Artificial neural architecture design • Gradient-based learning • Optimization by gradient descent **Lab** • Instructor-led demonstration related to topics of the week
4	**Regularizing Neural Networks** **Lecture** • Regularization and under-constrained problems • Data augmentation • Early stopping • Dropout **Lab** • Implement a deep fully connected neural network in the given python code template • Train it on a given toy dataset • Implement Dropout in the given python code template to regularize the deep fully connected network

Teaching Week	Topics
5	**Optimizers for Training Deep Neural Networks** **Lecture** • Momentum, Nesterov momentum • Second order methods • Algorithms bearing adaptive learning rates • Hyperparameter optimization **Lab** • Implement AdaGrad, RMSprop and ADAM optimizers in the given code skeleton
6	**Convolutional Neural Networks (CNNs) - Part I** **Lecture** • Overview of CNNs • CNN basics – layer, kernel, stride, pooling, deconvolution, etc. • Convolution and pooling as an infinitely strong prior • Structured outputs **Lab** • Instructor-led demonstration related to topics of the week
7	**Convolutional Neural Networks (CNNs) - Part II** **Lecture** • Efficient convolution algorithms • Unsupervised features • Popular Image Classification CNNs **Lab** • Implement the forward and backward passes of convolution, pooling, and batch normalization in the given code template
8	**Recurrent Neural Networks (RNNs) - Part I** **Lecture** • Motivation for recurrent neural networks • Backpropagation through time (BPTT) • Architecture overview: fixed-size to sequence, sequence to sequence, etc. **Lab** • Implement and train a vanilla RNN character-level language model

Teaching Week	Topics
9	**Recurrent Neural Networks (RNNs) - Part II** **Lecture** • Long-term dependencies - challenges • Long short term memory units (LSTMs) • Gated recurrent units (GRUs) • Optimization for long-term dependencies **Lab** • Implement and train LSTM character-level language model
10	**Improving Deep Neural Networks** **Lecture** • Performance metrics • Default baseline models • Importance of data • Debugging strategies **Lab** • Assignment # 3 due
11	**Prevalent Deep Learning Fields** **Lecture** • Computer vision • Speech recognition, and natural language processing • Health care • Financial data analysis **Lab** • Run any three state-of-the-art face detection algorithms and compare performance using standard metrics
12	**Autoencoders** **Lecture** • Undercomplete autoencoders • Denoising autoencoders • Learning manifolds with autoencoders **Lab** • Prepare a short report on the effectiveness of the U-Net architecture for both medical imaging applications and various mainstream computer vision tasks

Teaching Week	Topics
13	**Generative Models** **Lecture** • Linear generative models • Generative adversarial networks (GANs) **Lab** • Implement a GAN to generate photo-realistic (facial or natural) images
14	**Deep Reinforcement Learning** **Lecture** • Introduction to reinforcement learning • Markov process • Value and policy iterations • Temporal difference (TD) learning • State-action-reward-state-action (SARSA) **Lab** • Assignment # 4 due
15	**Guest Lecture & Review** **Lecture** • Guest Lecture on Trending Topic **Lab** • Review and Exam Preparation