IT549: Deep Learning
Academic Year 2024-25 (Winter)

Instructor: Dr. Arpit Rana (arpit_rana@daiict.ac.in)

Prerequisites: Programming in Python, Machine Learning

Slot: B.Tech. VI Semester, M.Tech. II Semester, and Ph.D. Students

Category: Technical Elective

Course Credits: 4 Credits (3-0-2)

Lectures (1 x 3): TBD

Lab Session (1 x 2): TBD

TA contact info: TBD

Course Description:

Deep Learning aims to equip students with a comprehensive understanding of the principles, algorithms, and applications of neural networks and deep learning techniques. Through theoretical knowledge and practical experience, students will gain the expertise to effectively design, implement, and deploy neural network-based solutions across various domains. The course emphasizes practical application development through course projects and lab sessions, preparing students to understand and implement deep learning algorithms and methodologies.

Suggested Books:

- Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville, The MIT Press [2016].
- Deep Learning with Python by François Chollet, Manning Publications, 2nd Edition [2021].
- Deep Learning from Scratch by Seth Weidman, O'Reilly, 1st Edition [2019]

Course Outcomes:

- Understand the advantages of deep learning.
- Gain in-depth knowledge of deep architectures such as CNNs, RNNs, LSTMs, GRUs, Attention mechanisms, Transformers, and Autoencoders.
- Theoretical and practical understanding of the architectures and insights into design choices for better model development.
- Learn essential model training concepts like regularisation, dropout, data augmentation, batch normalization, and hyperparameter tuning, which are explored for effective optimization.
- Hands-on demos using TensorFlow and PyTorch on images, text, time series, language data, etc., are included for all architectures, equipping with practical skills to excel in deep learning.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
X	X	X		X				X			X

IT549: Deep Learning
Academic Year 2024-25 (Winter)

Evaluation Scheme:

Ability to Understand and Apply Theoretical Concepts		Ability to Implement and Solving Real Problems	
In-Sem Exams (25 %)	Endterm Exam (30 %)	Challenges (15 %)	Course Project (30%)

Tentative Detailed Course Plan[1]:

Units	Topics	# Lectures
Classical Machine Learning Revisited	Algebraic, Probabilistic, and Graphical View of Data; (i) Representation: hypothesis class—bias-variance trade-off and expressiveness-model complexity trade-off; (ii) Evaluation: consistency and generalizability; data splits; and measures of model performance; and (iii) Optimization: learning as a search: hyperparameter and parameter tuning;	3
Introduction to Neural Networks	Importance of deep learning; Representation learning from large datasets; Case studies and applications. Artificial neurons, inputs, weights, and activation functions; Relationship with linear models; Layers of neurons, an analogy of matrix multiplication; Multilayer neural network: input layer, hidden layers, and output layer. The motivation behind using more layers.	6
Training a Neural Network	Hyperparameters and parameters of a neural network; training under supervised learning settings: labeled dataset, loss function, and the backpropagation algorithm: maths behind the algorithm. Challenges in training neural networks: (i) Vanishing Gradient Problem: Batch Normalization; (ii) Transfer learning; Unsupervised pre-training; Pre-training on an auxiliary task; (iii) Using faster optimizers; Learning rate scheduling; (iv) Overfitting: Reducing the network size; Weight regularization; Dropout; Early stopping	6
Convolution Neural Network (CNN)	Motivation, Convolution Operation, Image as a rank-3 tensor, convolution layer, connection between convolution layers; reducing dimensions using stride; filters, Input Image to Feature Map; Stacking Multiple Feature Maps; Pooling layer: motivation, types of pooling layer. Training a CNN: hyperparameters, memory requirements, overfitting: dropout and data augmentation; Pre-trained CNNs: Concept of Transfer learning.	6
Recurrent Neural Network (RNN)	Recurrent Neurons and Layers; Training RNNs; Forecasting a Time Series using Simple RNN, Deep RNN; Handling Long Sequences: LSTM; Encoder-Decoder Network: Bidirectional RNN, Beam Search; Attention Mechanism: The Transformer Architecture — Training of Transformer	9

Autoencoders (AEs) and Generative Adversarial Networks (GANs)	Deep learning for unsupervised learning; Architecture design of AE: Linear, Stacked, Convolutional; Recurrent; Denoising, Sparse; and Variational; Generative Learning using VAE. Generative modeling as a game-theoretic approach; Architecture design of GAN; Training methodology of GAN; Applications.	6
Introduction to Recommendation Systems and Multimodality	Definition, objectives, components, approaches, evaluation, and challenges. Models: Matrix Factorization, Neural Matrix Factorization, and Collaborative Denoising Auto-Encoders; The notion of Multimodality, Fusion Techniques, and Applications in Recommendations.	6

1 — This course plan is subject to change without notice.

Preliminary Schedule of the Course:

Week	Lecture	Lab	Project
Week-1 [1 Jan 2025]	Course Admin; Fundamentals of Predictive Analytics-I: Representation, Evaluation, and Optimization	– No lab –	–
Week-2 [6 Jan 2025]	Introduction to Neural Networks: Neurons, Activation, Layers, Architecture, and Examples	– No lab – (Group formation and domain finalization through Google form)	–
Week-3 [13 Jan 2025]	Training Deep Neural Networks: Backpropagation, Vanishing Gradient Problem: Definition, Better Weight Initialization, Non-Saturating Activation Functions, and Batch-Normalization	Linguistic Preprocessing, Text Vectorization, and Embeddings	Project Assigned
Week-4 [20 Jan 2025]	Overfitting: Reducing the network size, Weight or Max-norm regularization, Dropout, Early stopping	Pre-training and Fine-tuning	–
Week-5 [27 Jan 2025]	Applications in Natural Language Processing	100-Minute ML Development Challenge	–

Week-6 [3 Feb 2025]	**First In-Semester Examination** (7th Feb - Friday to 11th Feb - Tuesday)		–
Week-6 [10 Feb 2025]	Convolution Neural Networks (CNNs): Introduction, Motivation, Images and Rank-3 Tensors, Convolution Layers, Filters, Feature Maps, and Stacking Convolution Layers	Image Vectorization or Image Tensors, Using Pre-trained CNNs, Fine-tuning, etc.	Progress Check I
Week-7 [17 Feb 2025]	Training CNNs: Memory requirements, overfitting, data augmentation, Pre-trained CNNs, Transfer Learning,	Object Detection and Image Segmentation	–
Week-8 [24 Feb 2025]	Recurrent Neurons and Layers; Training RNNs; Forecasting a Time Series using Simple RNN, Deep RNN; Handling Long Sequences: LSTM;	Working with Sequences or Time-Series Data	–
Week-9 [3 Mar 2025]	Encoder-Decoder Network: Bidirectional RNN, Beam Search; Attention Mechanism:	100-Minute ML Development Challenge	–
Week-10 [10 Mar 2025]	**In-semester Break** (Entire week)		-
Week-11 [17 Mar 2025]	The Transformer Architecture — Training of Transformer	Training Transformers	Progress Check II
Week-12 [24 Mar 2025]	**Second In-Semester Examination** (22nd Mar Saturday to 26th Mar Wednesday)		–
Week-13 [31 Mar 2025]	Deep learning for unsupervised learning; Architecture design of AE: Linear, Stacked, Convolutional; Recurrent; Denoising, Sparse; and Variational; Generative Learning using VAE.	Unsupervised Learning using Auto-encoders: An application for demonstration	–
Week-14 [7 Apr 2025]	Generative modeling as a game-theoretic approach; Architecture design of GAN;	Training GANs: An application for demonstration	-

	Training methodology of GAN; Applications.		
Week-15 [14 Apr 2025]	Definition, objectives, components, approaches, evaluation, and challenges. Models: Matrix Factorization, Neural Matrix Factorization, and Collaborative Denoising Auto-Encoders;	100-Minute ML Development Challenge	-
Week-16 [21 Apr 2024]	Last date of Classes, Labs, and Tutorials **(23rd Apr 2025)**		Progress Check III
Week-17 [29 Apr 2024]	End-Semester Examination (24 Apr 2025 to 02 May 2025)		
Course Project Evaluation (05 May 2025 to 07 May 2025)			

Challenges and Course Project [15% + 30% = 45%]

The following points regarding the course project are to be noted.

Student Groups

- The course project will be allocated to groups of three/ four members. Each group will also be involved in 100-minute ML development challenges.
    - 45% of your marks will be based on team efforts, so choose your members wisely.
    - Teams will remain unchanged throughout the semester once registered. No requests for changes will be entertained.
- Team registration will open in the second week after classes start, and you must register your team within three days of the announcement via a Google form.
- During lab hours, there will be three machine-learning challenges. The three most proficient teams shall be acknowledged with a bonus of up to 3% on their respective scores.
- Every team member must understand the concepts, code, and claims they submit, as any member may be asked questions about their project.

Course Project [30%]

- There is only one course project, an End-to-End ML application.
- Student groups must select a thematic domain: Finance, E-commerce, Healthcare, Pharma, Sports, Entertainment, Renewable Energy, Oil & Gas, Automobile, Agriculture, FMCG, Security, Social Media, Supply Chain, or any other exciting and valuable domain.
- Each group will define the problem in their selected domain and collect dataset(s) from reliable sources, including publicly available ones (**no two groups can work on the same dataset**). You are encouraged to gather additional data to enhance your dataset and better address the problem.
- Each group must develop a multimodal machine-learning application and select a dataset with all the necessary modalities. You must add a novel contribution to your project and compare yours with the existing baselines.
- Three progress checks are scheduled to ensure incremental progress, not a last-week effort.
- The project guideline document will provide information on domain allocation, general instructions, evaluation criteria, and other protocols.

Submission

- One group member will submit the project report and the code on Google Classroom. Submission instructions will be provided in the project guideline document.
- Evaluation will primarily be online, reviewing your code. Any group member may be asked questions about anything in the assignment.
- Late submissions (up to 24 hours) will incur a 20% penalty.
- Plagiarism includes:
    - Copying any segment of code from any source.
    - Submitting code not written by you personally.
- **Suspected plagiarism will result in a ZERO for the assignment.**