Course Information

Course Code	5710796
Course Section	1
Course Title	DEEP GENERATIVE MODELS
Course Credit	3
Course ECTS	8.0
Course Catalog Description	Deep generative modeling with Autoregressive models; Energy-based models; Adversarial models; Variational models.
Prerequisites	No prerequisites
Consent of Dept./Inst.	CENG 783 or a similar course on the fundamentals of deep learning. Preferably strong background on the fundamentals of probability and statistics.
Schedule	Friday , 08:40 - 10:30, BMB2 Thursday , 11:40 - 13:30, BMB2
Course Website	https://user.ceng.metu.edu.tr/~gcinbis/courses
Learning Management System	https://odtuclass.metu.edu.tr

Instructor Information

Name/Title	Assist.Prof.Dr RAMAZAN GÖKBERK CİNBİŞ
Office Address	B205
Email	gcinbis@metu.edu.tr
Personal Website	http://user.ceng.metu.edu.tr/~gcinbis/
Office Phone	
Office Hours	By appointment -- please email.

Course Objectives

At the end of the course, the students will be expected to:

- Comprehend a variety of deep generative models.
- Apply deep generative models to several problems.
- Know the open issues in learning deep generative models, and have a grasp of the current research directions.

Course Learning Outcomes

Students who passed the course will be able:

- to understand the theory behind comtemporary deep generative models,
- to have a grasp of the open issues and trends in generative models,
- to have an understanding of the advantages and disadvantages of different types of deep generative model formulations,
- to gain hands-on experience in implementing & realizing generative models.

Program Outcomes Matrix

Master's (with thesis)

		Level of Contribution			
	Program Outcomes	**0**	**1**	**2**	**3**
1	Competence in fundamental and advanced knowledge of hardware and software Proficiency in problem solving.				
2	The ability to follow the contemporary technical development, and Initiative and aptitude for self-directed learning.				
3	They are capable of designing, and conducting experiments at advanced level.				

		Level of Contribution			
	Program Outcomes	**0**	**1**	**2**	**3**
4	The ability to design and implement systems involving hardware, software, and the interaction between the two through challenging projects.				
5	Analyze and compare relative merits of alternative software design, algorithmic approaches and computer system organization, with respect to a variety of criteria relevant to the task (e. g. efficiency, scalability, security).				
6	Strong interpersonal skills needed for working effectively in small, diverse groups on medium to large scale technical projects.				
7	Strong oral communication skills essential for effectively presenting technical material to an audience and strong written communication skills and the ability to write technical documents that include specification, design, and implementation of a major project.				

Doctoral

		Level of Contribution			
	Program Outcomes	**0**	**1**	**2**	**3**
1	Competence in fundamental and advanced knowledge of hardware and software Proficiency in problem solving.				
2	The ability to follow the contemporary technical development, and Initiative and aptitude for self-directed learning.				
3	They are capable of designing, and conducting experiments at advanced level.				
4	The ability to design and implement systems involving hardware, software, and the interaction between the two through challenging projects.				
5	Analyze and compare relative merits of alternative software design, algorithmic approaches and computer system organization, with respect to a variety of criteria relevant to the task (e. g. efficiency, scalability, security).				
6	Strong interpersonal skills needed for working effectively in small, diverse groups on medium to large scale technical projects.				
7	Strong oral communication skills essential for effectively presenting technical material to an audience and strong written communication skills and the ability to write technical documents that include specification, design, and implementation of a major project.				

Non- Thesis Master's (Evening)

		Level of Contribution			
	Program Outcomes	**0**	**1**	**2**	**3**
1	Competence in fundamental and advanced knowledge of hardware and software Proficiency in problem solving.				
2	The ability to follow the contemporary technical development, and Initiative and aptitude for self-directed learning.				

		Level of Contribution			
	Program Outcomes	**0**	**1**	**2**	**3**
3	They are capable of designing, and conducting experiments at advanced level.				
4	The ability to design and implement systems involving hardware, software, and the interaction between the two through challenging projects.				
5	Analyze and compare relative merits of alternative software design, algorithmic approaches and computer system organization, with respect to a variety of criteria relevant to the task (e. g. efficiency, scalability, security).				
6	Strong interpersonal skills needed for working effectively in small, diverse groups on medium to large scale technical projects.				
7	Strong oral communication skills essential for effectively presenting technical material to an audience and strong written communication skills and the ability to write technical documents that include specification, design, and implementation of a major project.				

0: No Contribution **1:** Little Contribution **2:** Partial Contribution **3:** Full Contribution

Instructional Methods

- Lectures
- Student paper presentations.
- Projects and project demonstrations
- Pre-recorded videos & reading materials.

For some lectures (e.g. a coding session), students may be expected to watch and/or read provided material before class. In such cases, the material may be only summarized and/or discussed without fully re-presenting all details.

Tentative Weekly Outline

Week	Topic	Relevant Reading	Assignments
1	Introduction, course logistics, deep learning review		
2	Review of probability, statistics and graphical models		
3	Autoregressive Flows & Maximum Likelihood Estimation		
4	Latent Variable Models, variational approximation		
5	Latent Variable Models, variational approximation (cont'd)		
6	Normalizing Flow Models		
7	Generative Adversarial Networks		
8	Generative Adversarial Networks		
9	Generative Adversarial Networks, evaluation metrics		

Week	Topic	Relevant Reading	Assignments
10	Energy-based Models		
11	Hybrid models & Discrete Latent Variable models		
12	Diffusion & score-matching models, vision & language models		
13	Paper presentations.		
14	Paper presentations / project demos		

Course Textbook(s)

There are no required textbooks for the course.

Course Material(s) and Reading(s)

Material(s)

Suggested Books:

- Y. Bengio, I. Goodfellow and A. Courville, "Deep Learning", MIT Press, 2016.
- C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Reading(s)

Optional / required readings will be provided throughout the semester. Course material heavily depends on recently published papers.

Assessment of Student Learning

Assessment	Dates or deadlines
Midterm: • May cover everything up to that date, including lectures & paper presentations.	

Assessment	Dates or deadlines
Projects (developer) Each student (or group) will implement one particular algorithm and contribute to the shared git repository**in PyTorch**, including documentation & demo in the format of a**jupyter notebook**. (Details will be provided in class.) - No free-form project. Each project is a proper & full-fledged implementation of one or more particular model(s) / paper(s). - There will be version-1 and version-2 submissions. First version will be evaluated by the assigned reviewer(s), who will give a written feedback. - The final grade will be a weighted average of the grade for the first version and the second version. Both versions will be graded by the instructor after submission of the final version. - The developer is expected to submit a complete and near-final version of the project in the first version, and benefit from the reviewer feedback to fix any remaining minor problems in the second version. - No code copy-paste is allowed, the submitted code + jupyter notebook should be a genuine implementation of the model. For model-independent utility functions (eg. data loading, etc), external code may be used with proper acknowledgement of the source. - Projects will be submitted to a joint git repository. The repository (including all projects) are planned to be made publicly available online at the end of the semester. - The minimum and maximum project developer group sizes will be determined within the first few weeks of the semester. Students may or may not be required to work individually. - In the last week of classes or during the final exam periods, there will project demonstrations. Project demonstration durations and guidelines are to be declared during the semester.	
Projects (reviewer) Each student will be assigned as a peer-reviewer to one or more project(s) developed by the other student(s). - The reviewer will be responsible for checking the implementation of the corresponding model and the jupyter notebook. Each reviewer will give a written feedback to the corresponding developer with a list of mistakes and missing points. - Reviewer will have at least one week to review and give feedback. - For mistakes and shortcomings denoted by the reviewer regarding the full submission material in the first version of a project, only the developer will be penalized. The developer(s) is expected to fix these problems in the second version of the project. - For mistakes and shortcomings missed by the reviewer but found by the instructor(s) in the first version, both the developer and the reviewer will be penalized. - The reviewer will additionally be evaluated in terms of the quality and coverage of his/her written review.	

Assessment	Dates or deadlines
Paper presentations: • Propose a list of candidate paper for presentation before the paper-presentation-proposal deadline. Student-to-slot assignments will be done by the instructor, within lecture hours. • Prepare well for the presentation: you will effectively be the instructor of the class during your presentation. Please prepare your presentation as a short-lecture that focuses on a particular work, rather than a dry summary of the contents of the paper. Therefore, please cover important related work (if not already covered in the class) in your presentation, in order to (i) make the presentation accessible for everyone, and, (ii) properly discuss the strengths and weaknesses of the paper compared to related work. • Each student will make one or more presentations. Each presentation should take around 20 minutes (excluding discussions). The expected duration and the expected number of presentations may be altered depending on class size. If a student makes multiple presentations, their grade averages will be taken. • Each presenter is required to send us a complete draft of the slides (as a pdf) 2 days before the presentation date, and the final slides before class on the presentation date. Slides will be published on the course webpage.	

Course Grading

Deliverable	Grade Points
Midterm	30
Paper presentation	20
Project - developer	30
Project - reviewer	10
Homework	10
Total	100

Course Policies

Class Attendance

Attendance will **not** be collected for grading purposes.

Class Participation

Students are responsible for all material covered in class and paper presentations in the midterm.

Any one of the following may result in an automatic FF / NA:

- not making the required number of applications for paper presentation with relevant papers (as described in class) before the paper presentation proposal deadline,
- not presenting a paper on his/her pre-scheduled date,
- not submitting his/her project (if the project is done individually),
- not contributing substantially to the project (if the project is done as a group),

- not writing a proper project review feedback,
- being absent in the midterm,
- being absent in his/her project presentation,
- plagiarism.

Information for Students with Disabilities

Students who experience difficulties due to their disabilities and wish to obtain academic adjustments and/or auxiliary aids must contact ODTU Disability Support Office and/or course instructor and the advisor of students with disabilities at academic departments (for the list: http://engelsiz.metu.edu.tr/en/advisor-students-disabilities) as soon as possible. For detailed information, please visit the website of Disability Support Office: https://engelsiz.metu.edu.tr/en/

Academic Honesty

The METU Honour Code is as follows: *"Every member of METU community adopts the following honour code as one of the core principles of academic life and strives to develop an academic environment where continuous adherence to this code is promoted. The members of the METU community are reliable, responsible and honourable people who embrace only the success and recognition they deserve, and act with integrity in their use, evaluation and presentation of facts, data and documents."*