CSCI 4701: Deep Learning

Course Description

Deep Learning focuses on artificial neural networks and how they are trained and optimized. This course covers core concepts starting with backpropagation, including regularization and optimization techniques. Students will gain practical skills using the PyTorch framework and learn neural architectures used in both Computer Vision (CV) and Natural Language Processing (NLP), including Convolutional Neural Networks (CNNs) and Transformer-based models. The course introduces generative modeling with Variational Autoencoders (VAEs) and briefly covers current developments in deep learning, such as Latent Diffusion Models (LDMs) and Large Language Models (LLMs).

Prerequisite(s)

- CSCI 4734: Machine Learning (official)
- Fundamental knowledge in Calculus, Linear Algebra, & Probability Theory
- Experience with Python & numpy (see this tutorial)

Contacts

Instructor(s)		
Name	**Email**	**Office**
Ismayil Shahaliyev	ishahaliyev@ada.edu.az	B318

Academic Advisors[1]		
Name	**Email**	**Program**
Turkan Huseynova	thuseynova@ada.edu.az	BSCS
Kamala Hashimova	khashimova@ada.edu.az	BSIT & BSMATH
Aysun Mustafazada	amustafazada@ada.edu.az	BSCE & BSEEE

Assessments

Item	**Info**	**100%**
Team Project	Students will apply the theoretical knowledge they gained to a practical semester-long team project.	25
Quizzes (8x)	Students will test their knowledge as they progress throughout the course.	25
Exam [Midterm]	Students will test their knowledge gained in the first half of the semester.	25
Exam [Final]	Students will test their knowledge gained throughout the semester.	25

Technological Requirements

Students are expected to have Github and Google (for accessing Google Colab) accounts, and install the git version control system, Python (3.9 or later), and (optionally) Jupyter. IDEs such as VSCode or Cursor are recommended for local development.

[1] Office: B building 1st floor. In case of **excused absences,** you are required to bring a document of proof to your academic advisor.

Deadline Policy

Deadlines will never be extended. When submitting, a student must consider possible internet connection issues, uploaded file size, etc. **Late submissions** of one minute until thirty minutes will receive a 10% penalty, late submissions of thirty minutes until one day will receive a 25% penalty with no claim for bonuses (if there are any), late submissions of more than one day will receive zero. All deadlines are at 11 PM (23:00) Baku time (UTC+4), unless explicitly stated otherwise.

Team Project

You will form teams of up to three members at the start of the semester. Students who are not part of a team one week before the project proposal deadline will be automatically assigned to a team at random. A **project proposal** that *passes* receives 10%. A proposal marked *revise* receives a provisional 0% until the required changes are submitted and approved, after which full credit is awarded. A proposal that is *rejected* and resubmitted as a substantially different proposal is eligible for a maximum of 5%. In addition to the proposal, there will be two further **project milestones**, scheduled before the midterm and final exam periods, each accounting for 45% of the total project grade. Detailed rubrics will be shared with you on blackboard.

Quizzes

There will be **nine** computer-based quizzes assessing understanding of the material covered in the preceding lectures. The quiz grade will be calculated as the arithmetic mean of the **best eight** quizzes, with the lowest score automatically discarded (e.g. missing one quiz will not affect your grade). Each quiz will consist of **10 questions** with a **30-minute** time limit, including **4x** scenario-based multiple-choice questions (testing conceptual understanding and reasoning), **3x** short-answer questions (requiring precise explanations), and **3x** code-focused questions (e.g. identifying errors, predicting outputs, or completing small code fragments). One sample question per type will be shared with you before the first quiz.

Exams

You will have two exams: midterm and final. Both exams will be computer-based (unless noted otherwise), consist of **20 questions** of the same types (**8x**, **6x**, **6x**) used in the quizzes, and have a **60-minute** time limit.[2] More specific rules and details will be communicated to you before the exam[3].

Suggested Reading

- Goodfellow, I., Bengio, Y., and Courville, A. Deep Learning. MIT Press, 2016. https://www.deeplearningbook.org/
- Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. Dive into Deep Learning. https://d2l.ai
- Study materials will be shared with you on blackboard throughout the semester.

Course Outline

The content is subject to change. Please consistently check the course page on blackboard and the ADA Academic Calendar for modifications as well. The last day of the add/drop period, holidays, etc. are noted in the calendar.

---

[2] Quiz and exam dates, as well as project deadlines are noted in the course outline section.

[3] You are expected to be aware of the Exam Rules and Regulations. If you have special needs or health issues, you are strongly recommended to contact the University's Student Academic Support Services well ahead of the examination date. It is your responsibility to manage conflicts in your schedule and notify your instructors about it at least two weeks before the examination date.

#	Topic	Learning Outcomes	Assessment
1	Deep Learning (DL) overview / Course structure	Describe the scope of DL and the course syllabus. Fulfill **technological requirements**.	
2	Mathematics of DL: Linear Algebra / Calculus	Work with vectors, matrices, and tensors, and perform basic **matrix operations**. Apply **norms** and inner products. Compute partial derivatives and apply the **chain rule** to multivariable functions. *Optional*: Understand intuition behind eigenvectors and SVD.	
3	Gradient Descent / Backpropagation I	Compute gradients efficiently using the chain rule on **computational graphs.** Perform forward and backward passes. Understand how **gradient descent** updates parameters to reduce a loss. Understand **automatic differentiation** of the PyTorch autograd engine (with micrograd).	
4	Gradient Descent / Backpropagation II	**Implement** full backpropagation.	**Feb 3:** Quiz 1 [1-3] Last day to submit team member details
5	Activation Functions / Neuron	Implement **activation functions** and understand *non-linearity*. Backpropagate over an **N-dimensional neuron**.	
6	Multilayer Perceptron (MLP)	Construct an MLP from stacked neurons. **Train a simple MLP classifier** on a small dataset.	**Feb 10:** Project proposal deadline
7	Images as Tensors / MLP on MNIST / Batching & Cross-Entropy	Understand **image representations**, tensor shapes, batching. Use torchvision datasets and dataloaders. Train an MLP on MNIST with **SGD** + **Cross-Entropy**.	**Feb 12:** Quiz 2 [5-6]
8	Convolutional Neural Networks (CNN)	Define and implement 2D **cross-correlation** (convolution) and **pooling** with kernels, including padding and stride. **Train a LeNet-style CNN** on MNIST. Compare MLP with CNN.	
9	Mathematics of DL: Probability Theory	Describe **random variables**; distinguish discrete and continuous distributions, work with **PMF/PDF**. Compute **expectation**, **variance**, and **covariance**. Understand **joint/marginal distributions**, **conditional probability**, **independence**, the **chain rule**, and **Bayes' rule**. Recognize the behavior of common **distributions** (e.g. Bernoulli, Multinoulli, Normal).	**Feb 19:** Quiz 3 [7-8]
10	Regularization	Apply **weight decay** and **dropout** to reduce overfitting. Handle **exploding** and **vanishing gradients**. Use **Xavier** and **He initialization** to improve gradient flow. Distinguish **local minima** from saddle points when reasoning about training dynamics.	
11	Optimization	Adjust **learning rate** and apply schedules. Use SGD with **momentum** to accelerate convergence. Apply **RMSProp** and **Adam** to handle noisy gradients. Compare optimizers based on convergence behavior and practical performance.	
12	Regularization / Optimization	**Train a regularized CNN on CIFAR-10** using optimizers. Apply **hyperparameter tuning.**	**Mar 3:** Quiz 4 [10-11]
13	Paper: AlexNet	Discuss AlexNet paper, its key ideas. Understand what is outdated. Discuss the paper structure.	
14	Bigram Model / Negative Log-Likelihood / Softmax	Build a character-level **bigram model** from text data and **sample** from it. Understand difference between probability and **likelihood**. Compute average **negative log-likelihood** as a loss. Explain the purpose of **softmax**.	
15	Neural Network N-gram Model / Mini-Batch Training	Construct a **neural N-gram model**. Train the model with mini-batch updates.	**Mar 12:** Quiz 5 [14] Project milestone 1 deadline
16	Midterm Exam [1-15] TUESDAY, **MARCH 17**		
17	Midterm Exam Review	Half-semester overview.	
HOLIDAYS, **MARCH 20-30**			

18	Batch Normalization / Layer Normalization	Explain why normalization helps training deep networks. Implement **batch normalization** in a simple neural network and understand training vs evaluation behavior with **running statistics**. Understand the impact of **batch size** on batch normalization and when to prefer **layer normalization** instead.	
19	Residual Blocks / Residual Network for NLP	Understand the idea of **residual (skip) connections** and why they help very deep networks train. Modify a feed-forward N-gram model to include a **residual block** with appropriate dimensions. Interpret residual blocks as extending the function class while still allowing identity mappings, and connect them to vanishing gradients and regularization.	
20	Sequence Modeling: Autoregressive Models and RNN/LSTM	Understand **autoregressive sequence modeling** beyond fixed context windows. Explain how RNNs maintain state across time. Identify limitations of RNNs and LSTM/GRU (sequential computation, difficulty scaling, long-range dependency).	**Apr 7:** Quiz 6 [18-19]
21	Attention Mechanism	Understand **attention** as weighted information selection over a sequence. Derive and explain **queries**, **keys**, and **values** at the tensor level. **Implement attention** explicitly using matrix operations and verify shape consistency and normalization behavior.	
22	Transformer Architecture / Self-Attention	Understand **self-attention** as attention within a single sequence. Explain how **Transformer blocks** are constructed from self-attention, feed-forward layers, residual connections, and layer normalization. Explain why Transformers scale.	**Apr 14:** Quiz 7 [20-21]
23	Transformer Architecture / Transformer Blocks	**Assemble** a complete **Transformer block** from self-attention and feed-forward sublayers. Trace the forward and backward signal flow through the block. Analyze training stability, gradient propagation, and sensitivity to initialization and learning rate.	
24	Paper Reading: Transformer, Vision Transformer, Swin-Transformer	Extract the core architectural ideas from the Transformer, Vision Transformer, and Swin Transformer papers. Compare how attention is applied to sequences versus images and how architectural constraints affect scalability and efficiency.	**Apr 21:** Quiz 8 [22-23]
25	Mathematics of DL: Information Theory and Probabilistic Modeling	Compute entropy, cross-entropy, and **Kullback–Leibler (KL) divergence** for discrete and continuous distributions. Derive cross-entropy loss from **maximum likelihood** principles. Interpret common deep learning losses as probabilistic objectives.	
26	Variational Autoencoders I	Introduce **latent-variable generative models.** Understand the idea of latent representations and probabilistic **encoders/decoders**. Explain the role of **approximate inference** and why variational methods are needed. Analyze how VAEs learn structured latent spaces and how they differ from deterministic **autoencoders**.	**April 28:** Project milestone 2 deadline
27	Variational Autoencoders II	Understand the VAE training objective (**ELBO** optimization). **Implement VAE.** Interpret **reconstruction** and **regularization** terms and their trade-off.	
28	Diffusion Models / Score Matching	Formulate diffusion models via **forward noising** and learned **reverse denoising** processes. Derive the **training objective** and interpret it as denoising **score matching**. Explain sampling as iterative probabilistic inference.	**May 5:** Quiz 9 [25-27]
29	Foundation Models and Modern Trends	Explain large-scale pretraining and **transfer learning** as the basis of **foundation models**. Examine GPT, BERT, and CLIP as representative architectures for language and multimodal learning, and **latent diffusion models** (LDMs). Examine their scaling behavior and practical limitations.	
Final Exam [1-29] TUESDAY, **MAY 12**			

Revisions

v0

- No revisions yet