Neural Networks and Deep Learning for Scientists

38-616 & 09-616, Spring 2023

Olexandr (or Oles for short) Isayev

Department of Chemistry, CMU

olexandr@cmu.edu

Intro!

- Introduce yourself in a few sentences.

- What do you expect from this course?

- Fact about you: What if your favorite dish?

Outline

- Course logistics
- Course objectives
- Course overview
- Anaconda Python& Jupyter notebook ecosystem
- Access to PSC resources & GPU computing

Class schedule

Mon & Wed 11:00- 12:30 AM

3SC Room 110@PSC Building

Canvas: https://canvas.cmu.edu/courses/33629

Slack or Discord channel ?

Course requirements

Designed for STEM master students & MCS recent graduates

Open to senior–year undergraduate students (with permission)

This is **practical**, application-oriented course, requiring skill in algorithmic problem solving.

We will use **Python & PyTorch** based tools and libraries. Prior programming experience with Python is needed.

Prerequisites: probability, linear algebra, statistical thermodynamics and intro ML or related quantitative courses

Programming experience

Jupyter Notebook / Lab

ML/DL experience

PyTorch or other DL framework experience

M.S. in Data Analytics for Science (MS-DAS) program

- Fresh brand-new program
- New course! Sorry for possible hiccups.
- Please gave your feedback. olexandr@cmu.edu
- Job market is tough, but we are here to help!

Learning objectives

- Be proficient in using modern computing technologies (Python, Jupyter notebooks and PyTorch, etc.)
- Know how to explore and classify large scientific data set using neural networks and other deep learning tools.
- Understand core components of a data analytics pipeline: EDA, classification, regressions, prediction, etc.
- Implement and analyze well-known existing ML and AI algorithms.
- Integrate multiple components of practical machine learning and deep neural network methods in a single system: data preprocessing, learning, regularization, model selection and be familiar with programming tools to accomplish it.
- Hands on experience with real-world cases on how neural networks and deep learning
- could address challenges in science.

Course Outline

- Basic concepts: Model accuracy, prediction accuracy, interpretability, supervised and unsupervised training, regularization.
- Artificial neural networks, feed-forward, activation functions, loss functions.
- Non-linear optimization, gradient descent, back-propagation
- Deep Learning tools: PyTorch, PyTorch Geometric, Hugging Face, etc
- Autoencoders, dense embedding, dimensionality reduction
- Convolutional networks, transfer learning, applications in image processing and sciences
- Recurrent networks, LSTM, GRU, Transformers and their applications in NLP
- Graph-based models, flow- and diffusion models
- Other topics: GANs, Reinforcement Learning, Multitask Learning, advanced applications of deep learning in chemical and biological sciences.

Course Structure

Monday

Lecture materials

Wednesday

- Recital/practice
- Tutorials
- Lab discussions

Reading

No textbook

Readings, **mostly DL papers** will be provided on Canvas portal and lectures.

The readings for this course are required.

We recommend you read them **before** the lecture.

Course Grades

5% for attendance

5% for class participation

50% for Lab assignments

40% for final open-ended class project

Bonus points for Top Kaggle leaderboard score

Final Project

- You should work in teams of 2-3
- Open-ended project!
- Solve a science related problem with deep neural networks! Use your domain expertise
- Encourage to use your data
- Jupyter notebook, which mixes together written markdown and code portions or python script and report.
  - ~1000 words (2 pages of text)
  - ~200-300 lines of code
- All text and code must be your own work, but you can adapt and built on existing models.

Final Project

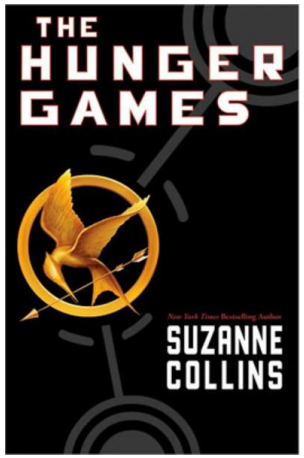
- Submit one paragraph project proposal (~February)
- Short project talk (Pitch! 1-2 slides) (~Early March)
- Presentation during last week and write short final report as a paper
- You will be graded by the course instructor and other students taking the course (peers)

Lab/Home Exercises

- Fully connected NN
- CNN
- RNN/Transformers
- Graph-based NN

Lab auto-grading

Brought to you by



- Five Labs per semester
- Short solution discussions after each Lab
- Assignments will include 2 parts: *programmi*

  kaggle component (autoscoring)
- Sometimes you will compete

  with each other…

Dashboard | Public Leaderboard - Heritage Health Prize

This leaderboard is calculated on approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

#	Δ1w	Team Name * in the money	Score	Entries
1	–	EXL Analytics *	0.443793	555
2	–	POWERDOT	0.447651	671
3	–	Dolphin	0.450403	555
4	↑1	jack3	0.451425	455
5	↓1	Hopkins Biostat	0.451569	444
6	–	Xing Zhao	0.453081	161
7	–	Old Dogs With New Tricks	0.454096	370
8	–	Areté Associates	0.454424	112
9	–	Alice Sasandr	0.454670	376
10	↑9	J.A. Guerrero	0.454728	173

ChatGPT Policy

ChatGPT, the buzzy chatbot developed by OpenAI that is capable of writing cogent essays, solving science and math problems and producing working computer code.

This is (potentially) the future! Embrace ChatGPT, if you find it useful for this class.

Learn from it, but please clearly attribute ChatGPT (if used) to comply with university policies on Ethics and Academic Integrity.

January Class Plan

January 23	ML and neural networks history
January 25	Bridges2 introduction & GPU computing
January 30	Deep neural networks & their “anatomy”
February 1	PyTorch tutorial

Questions?

Anaconda Python

- For the class, we strongly recommend you use Anaconda Python

- This distribution of Python, includes most libraries and tools

https://www.anaconda.com/download/

Installing additional packages

There are two general ways to install additional packages

```
conda install <package name>

     conda search <package name>
     conda list


pip install <package name>
```

PyTorch

www.pytorch.org

Jupyter notebook

- Notebook documents (are documents produced by the Jupyter Notebook App, which contain both computer code (python) and rich text elements (paragraph, equations, figures, links, etc…).

- Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc..) as well as executable documents which can be run to perform data analysis.

- More info about Jupyter here: http://www.jupyter.org

Jupyter notebook

Launch jupyter via the command:

```
jupyter notebook
```

Open in browser: http://localhost:8888

New (alternative) environment: `jupyter lab`

Access to GPU resources

- Google Colab

Pittsburgh Supercomputing Center (www.psc.edu)

- PSC Bridges-2 GPU-AI -> Nvidia Tesla V100 (DGX boxes)

If you are NOT in MS-DAS program, but taking this course:

- Please go to https://identity.access-ci.org/new-user and register.
- Please send me your username. I will add you to the allocation.