Fall 2025 (Aug 25 - Dec 18) CRN: 1660 MCS 5993 Topics in CS:

Evolutionary Computation and Deep Learning

Updated: 8-25-2025 V1.0

Class Day/Time/Room: MW 7:10pm-8:25pm in **S214**

Credit Hours: 3

Prerequisite: CS2 and Data Structures (This class requires in-depth programming experiences. If you are not confident in programming and problem solving with Python, it is strongly recommended to take this class after gaining more programming experiences with Python.)

Instructor: Chan-Jin "CJ" Chung, PhD

- Office: J355
- Phone: (248) 204-3504 Dept.: (248) 204-3560
- Email: cchung@LTU.edu
- Office Hours in J355: Mon 3pm-4pm, Wed 3pm-4pm, and by appointment.

Teaching Assistant: N/A

Course Description

This course explores how evolutionary computation enhances deep learning, addressing key challenges such as adaptability and learning from limited data, enabling neural network models to generalize well with only a few training examples (few-shot learning).

Students will learn bio-inspired algorithms that optimize neural network architectures, hyperparameters, and connection weights, minimizing model size while improving or at least preserving generalization ability.

Through hands-on projects, students will apply genetic algorithms and evolutionary strategies to real-world problems, advance deep learning models, and contribute to research.

Course Objectives

- Learn how the principles of evolutionary computation overcome deep learning's common pitfalls and deliver adaptable model upgrades without constant *manual* adjustment.
- Explore how biology-inspired algorithms and intuitions amplify the power of neural networks to solve complex problems
- Improve your deep learning models using the principles of biological evolution
- Solve challenge problems in DL, discover new knowledge and publish research articles on Research Day in April 2026 and/or conferences

Recommended Textbook 1: Evolutionary Deep Learning: Genetic algorithms and neural networks

- Author: Micheal Lanham
- https://www.google.com/books/edition/Evolutionary_Deep_Learning/1oDGEAAAQBAJ
- Textbook Code: https://github.com/cxbxmxcx/EvolutionaryDeepLearning

Recommended Textbook 2:

- Author: François Chollet
- Deep Learning with Python, 2nd edition
- https://www.manning.com/books/deep-learning-with-python-second-edition

Recommended Textbook 3:

- Author: Frances Buontempo:
- Genetic Algorithms and Machine Learning for Programmers: Create AI Models and Evolve Solutions (Pragmatic Programmers) 1st Edition
- https://www.amazon.com/Genetic-Algorithms-Machine-Learning-Programmers/dp/168050620X/

Course Main Topics (Tentative Schedule)

Week 1. Intro to evolutionary deep learning (EDL) (**HW1** out)
Week 2. Intro to Google Colab and function optimization
Week 3. Intro to evolutionary computation and ES (1+1) with 1/5 rule (**HW2** out)
Week 4. Review of ML, NN (Perceptrons, Deep Feedforward)
Week 5. Connection weight optimization instead of BP (**HW3** out)
Week 6. Review of Keras and DL (CNN and Finetuning) (**Project** out)
Week 7. Covariance Matrix Adaptation and ES (**HW4** out)
Week 8. Hyperparameter Optimization (**HW5** out) (**Project** must be decided)
Week 9. Evolving weights, hyperparameters, and topology (**HW6** out)
Week 10. Reinforcement learning, Deep RL, and EC
Week 11. Review DL: Regression, Time-series models
Week 12. Evolutionary computation with DEAP
Week 13. Evolutionary Meta, Few-Shot, Continual learning
Week 14. Evolutionary Discovery of Lightweight models: Evolutionary Quantization, Evolving BitNets
Week 15. Topics: MoE (Mixture of Experts), Agentic AI, Feature Engineering and EC; Evolving HP/Architecture of LLMs/Transformers (lighter, faster, more adaptable LLMs)
Week 16. Project Demo

Important Dates (See also Grading Section)

Sep 1	Labor Day. No class
Sunday, Sep 7	Last day to drop courses with refund. HW1 due by Sep 3, 7pm.
Sep 22, 24	Async. online class; No in-person class; No in-person office hours
Friday, Nov 21	Last day for "W" – Friday
Nov 26 - Nov 30	Thanksgiving break
Dec 15-18 **Wed., Dec 17**	Final Exam Week; Final Exam, 20:00-21:50, LTU Final Exam Schedule

Grading

- 6 homework assignments (12%) – 2 percent for each
- 3+ written closed book quizzes (33%) – Questions are mainly from homework assignments
- Final written closed book exam, comprehensive (30%)
- Term Project (25%)

Total % score will be translated into a letter grade based upon the percentages given below. **F will be given to Graduate students, if under 70%.** "Canvas" course management system (CMS) is used for recording scores.

A	91-100%	**C**	71% -
A-	90% -	**C-**	70% -
B+	85% -	**D+**	65% -
B	81% -	**D**	61% -
B-	80% -	**D-**	60% -
C+	75% -	**F**	0% -

Python File Type that must be used in class

- Everyone is required to use Jupyter Notebook, *.ipynb
- Why Jupyter Notebook?
  - Multiple code cells: Each cell can be executed individually. Some algorithms might take a very long time (like model training). Some algorithms (like model evaluation) need to be run independently multiple times. Multiple cells enable incremental development. Good for debugging too.
  - Multiple text cells for documentation in Markdown, a lightweight markup language

Suggested Computing Platforms for Programming

- Google Colab with free GPU. Now with generative AI "Gemini" assistance
- Your laptop with GPU using MS Code, PyCharm, etc. Need to install Anaconda and TensorFlow, etc. on your laptop. LTU's laptop for CS and Engineering has RTX A2000 8GB
- If your laptop has GPU, you can also use Google Colab's runtime from cloud to locally running Jupyter Notebook. You also need to install Anaconda, TensorFlow, etc.

Class Policies

- Attendance is essential to doing well in the course. Canvas is collecting data about your activities; Attendance will be checked ***at the start of every meeting*** by the instructor (Now it is required by LTU). Federal regulations require to collect & keep the class attendance data.
- **Student Athletes:** Notify the instructor in advance of excused absences. Games are excused; practices are not.
- Students who anticipate missing class due to family events, travel, or work obligations must inform the instructor in advance. For unexpected illness, appropriate documentation (e.g., a doctor's note) may be required.
- It is strongly recommended to use *Canvas Discussions* instead of using emails for technical questions**. Q & A can be shared with classmates.**

- At the end of the course, you will be invited to participate in a University evaluation of this course. Your feedback is important to the University, and to me as an instructor.
- *Class events may be photographed and/or videotaped. Students are expected to give permission for this material to be printed, published, posted on the websites, and/or televised in the public forum*.

Quizzes & Final Exam Policies

- **Quiz dates may not be announced.**
- The exams will focus primarily (but not exclusively) on material presented in the lectures and assignments.
- No makeup exam without prior communication.

Homework Programming Assignment Policies

- Homework programming assignments must be done individually
- Write your own code. Collaborating on programming assignments is not allowed. If too similar codes are submitted, (both giver and copiers) or (all collaborators) will be penalized
- When you use codes found on the Internet, appropriate *credits* as comments in your source code are required.
- **Policy on the Generative AI Tool Usage**: it is allowed to use AI coding assistant tools as long as you totally understand the generated code. **Please note that quizzes will be based directly on the assignments.**
- Every ipynb file submitted must have a markdown text cell at the very top as shown below:

> **HW *n* Assignment Disclaimer**
> - **Author Name:** First Last
> - **LTU ID:** 000000123
> - **Work Ownership:** This work is my own. It is not copied from classmates (Yes/No):
> - **AI Assistance:** Percentage of code generated with AI tools: xx%
> (If greater than 0%, briefly describe how AI was used)
> - **Understanding:** I understand every part of this code (Yes/No):
> - **Confidence:** I am confident that I can modify, adapt, and extend this code on my own (Yes/No):

- Source codes must follow good programming standards such as (1) good modular design, for example, introducing factored functions instead of a one huge main function. (2) commenting, (3) indentation, and (4) meaningful names. Note that OO design is NOT required
- Read the submission instruction carefully for each homework.
- Make sure to RERUN your code before submission to check if any errors.

- Submitted source code files (*.ipynb) that contain run results inducing graphs must be uploaded to the Canvas by the due date/time. Note that Canvas system records the exact time of your file uploading.
- Number of submission attempts: 2 (Instructor will grade the latest submission). If you request instructor's testing/feedback before the deadline, you must request it by email at least 72 hours (3 days) in advance. If your 1st submission is not near perfect, the 2nd submission will receive at least 5% deduction.
- Late penalty: 10% deduction **per day** up to 7 days. No credit will be given for works later than one week.

Term Project

- Group/Team projects can have up to 2 or 3 members based on *the project size and complexity*.
- Projects must be decided and approved by **Oct 15.**
- Group projects are required to submit peer evaluation form. See the form on Canvas.
- You must submit *well documented* source code files, presentation documents/slides/poster.
- Every ipynb file submitted must have a markdown text cell at the very top as the homework programming assignment shown above in a box.
- A rubric to grade the term project will be provided later on Canvas.

Communication

- All communication from LTU to students, faculty, and staff will be sent to Lawrence Tech email addresses and will not be sent to addresses in any other domain
- Class announcement will be sent out through Canvas to your LTU email address

Intellectual Property and Copyrights

All the deliverables may be reused/modified/upgraded by another students and/or instructor later on for educational purposes. The instructor will make sure to give appropriate credits and acknowledgements to the student in that case. The instructor believes that the student has the intellectual property rights of the software student wrote. However, since it is done in a class at LTU, it is also requested that the student should give appropriate credits and acknowledgements to the University as well as the instructor, if the software developed in class is used or commercialized after the class.

LTU Student Code of Conduct

The Student Code of Conduct outlines the rights and responsibilities and expected levels of conduct of students in the University community.