Instructor: Alina Vereshchaka

Final Project

Multiple deadlines, please see below

Description

The goal of the final course project is to explore advanced methods and/or applications in deep learning. You will be expected to prepare a proposal, checkpoint, final submission, and presentation. All projects should evaluate novel ideas, recent advances in deep learning, or their applications.

You are encouraged to use your ongoing research work as a project in this course, provided that this work relates to deep learning. Discuss the topic of your final project with course instructors by private message in Piazza, or during OHs. If you are unsure about the topic, we encourage you to speak with us.

There are a few directions suggested, please check the end of the description for more details.

1. Register your team (February 23)

You may work individually or in a team of up to 3 people with both undergraduate and graduate students. The evaluation will be the same for a team of any size. You can team up with students from the online section or in-person section.

Register your team at **UBLearns > Groups**.

Note: Make sure that all members join the same team on UBLearns (e.g. Team 27)

2. Project Management Tool [5 points] (March 1)

If you are working in a team of two or three people, set up a project management (PM) tool. E.g. Github Project (free plan), Trello (free plan)

1. Create a board with at least these columns: **To do, In Progress, Done**
2. Divide the project into at least 15 steps/milestones (E.g. Explore the topics, Create a proposal, Implement a basic version, Implement advanced algorithms, Submit Checkpoint, Submit Final, Prepare for the presentation, etc).
3. Add course instructor as observer or as team member to your board (amuralid@buffalo.edu)
4. There should be tracked activities every week by each team member
5. Include a few screenshots of your board activities as part of your Checkpoint and Final submissions.

3. Submit the proposal [10 points] (March 1)

The project proposal should be a one page single-spaced extended abstract motivating and outlining the project you plan to complete. Your proposal should have the following structure:

1. **Title**
2. **Short summary**. What problem statement or research question that your project aims to address? Why is it interesting?
3. **Objectives**. Outline the goals you aim to achieve.
4. **Methodology**. What deep learning techniques, models, or algorithms do you plan to use? How do you plan to improve or modify such implementations? You don't have to have an exact answer at this point, but you should have a general sense of how you will approach the problem you are working on.
5. **Evaluation**. How will you evaluate your results? Qualitatively, what kind of results do you expect (e.g. plots or figures)? Quantitatively, what kind of analysis will you use to evaluate and/or compare your results (e.g. what performance metrics or statistical tests)?
6. **Dataset**. What dataset(s) do you plan to use?
7. **References**. Include a list of the relevant literature and resources you plan to use.

4. Submit the checkpoint [20 points] (April 4)

- Submit the initial results of your model (e.g. setup the dataset for training, built baseline model and prepared a pipeline for training).
- You need to submit the code and a draft report with prior results.
- All project files need to have clear naming, e.g. **avereshc_nitinvis_checkpoint1_project.ipynb** and **avereshc_nitinvis_checkpoint1_project.pdf**
- All project files should be packed in a ZIP file named: **TEAMMATE#1_UBIT_TEAMMATE#2_UBIT_ checkpoint1 _project.zip** (e.g. **avereshc_nitinvis_checkpoint1_project.zip**).
- Submit at **UBLearns > Assessments**

5. Submit the final results [50 points] (May 2)

- Submit at **UBLearns > Assessments > Final Project**
- The code of your implementations should be written in Python. You can submit multiple files, but they all need to be labeled clearly.
- All project files need to have clear naming, e.g. **avereshc_nitinvis_final_project.ipynb** and **avereshc_nitinvis_final_project.pdf**
- All project files should be packed in a ZIP file named: **TEAMMATE#1_UBIT_TEAMMATE#2_UBIT_final_project.zip** (e.g. **avereshc_nitinvis_final_project.zip**).
- Your Jupyter notebook should be saved with the results.
- A report can be submitted with the presentation. Thus, if you discuss all the technical implementation of your project and provide the results, a separate report will not be necessary.
- Include all the references that have been used to complete the project.
- If you are working in a team, we expect equal contribution for the assignment. Each team member is expected to make a code-related

contribution. Provide a contribution summary by each team member in the form of a table below. If the contribution is highly skewed, then the scores of the team members may be scaled w.r.t the contribution.

Team Member	Project Part	Contribution (%)

7. Present your work [15 points] (May 2 – May 6)

Present your work during the presentation day. Registration slots will be available prior to dates.

- The whole team should equally present the work.
- Your presentation should represent the work you will submit.
- Submit the final presentation by **May 6, 11:59pm**.

Presentation details

Length: 15 mins + follow-up questions

Suggested Templates: UB branded templates

Suggested presentation structure:

- Project Title / Team’s Name / Course / Date [1 slide]
- Project Description [1 slide]

  Describe the problem you are working on, why it's important
- Background [max 2 slides]

  Discuss the related background and works that relates to your project. How is your approach similar or different from others?
- Dataset [max 2 slide]

  Describe the data you are working with for your project. What type of data is it? Where did it come from? How much data are you working with? Did you have to do any preprocessing, filtering, or other special treatment to use this data in your project?

- Methods [max 2 slides]

  Discuss your approach for solving the problem. Did you consider alternative approaches? Include figures, diagrams, or tables to describe your method or compare it with other methods.
- Results [max 3 pages]

  Discuss the experiments that you performed. The exact experiments will vary depending on the project, but you might compare with previously published methods, use visualization techniques to gain insight into how your model works. Include graphs, tables, or other figures to illustrate your experimental results.
- Demo (if available)
- Key Observations / Summary [1 slide]
- For Teams: Contribution Summary by Each Team Member [1 slide]
- Thank you Page [1 slide]

Extra Points [max +20 points]

Participate in Weekly Scrum Meetings [5 points]

Weekly Scrum is a regular group meeting led by Aditya Muralidharan and Wei Yang Lee starting from March 11 where at least one team member gives updates about the project progress. This is not evaluated, and you can also consider these meetings as an opportunity to get feedback on your current results. You can join any one of the scrums, either on Mon or on Wed, no need to join both the scrums per week.

Your team must take part in a scrum meeting for at least 5 weeks to be eligible for the bonus. We encourage all team members to join scrum meetings.

Real-world deep learning application [3 points]

If you are formulating and solving a real-world problem using deep learning methods, your work is eligible for this bonus. In your report, you must highlight the references to the real-world case that you are solving as well as the real-world data you use for your project.

Novelty [5 points]

If you have come up with a new model or setup, that is almost the same as the current models or even better, your work is eligible for this bonus. In your report, you must highlight the novelty.

Deploy the model locally [2 points]

- Deploy your final project model locally or on a server
- Make a 4-5 mins video recording showing how it works

CSE Demo Days [5 points]

- If you get interesting results, we encourage you to share your project with the public in terms of participating in the CSE Demo Days. CSE Demo Days is a semester event, where you can highlight your project results.
- Send us your prior results before April 27. **Selected teams** will have to prepare a poster and present it.
- No requests to take part in CSE Demo Days will be accepted after April 27.
- If you receive a winning place award for your project, **your bonus points will be +50!**

Important Information

This project can be done in a team of up to three people.

- All team members are responsible for the project files submission
- No collaboration, cheating, and plagiarism is allowed in assignments, quizzes, the midterms or final project.
- All the submissions will be checked using SafeAssign as well as other tools. SafeAssign is based on the submitted works for the past

semesters as well the current submissions. We can see all the sources, so you don't need to worry if there is a high similarity with your Checkpoint submission.

- The submissions should include all the references. Kindly note that referencing the source does not mean you can copy/paste it fully and submit it as your original work. Updating the hyperparameters or modifying the existing code is subject to plagiarism. Your work must be original. If you have any questions, send a private post on piazza to confirm.
- Submitting material that has been previously submitted, in whole or in any part is not allowed.
- All group members and parties involved in any suspicious cases will be officially reported using the Academic Dishonesty Report form. What does that mean?
    - In most cases, the grade for the assignment/quiz/final project/midterm will be 0 and all bonus points will be subject to removal from the final evaluation for all students involved.
    - A grade reduction will be applied towards the final evaluation
    - Those found violating academic integrity more than once throughout their program will receive an immediate F in the course.
- Please refer to the Academic Integrity Policy for more details.

Late Submission Policy

For the final project everyone will be provided with 3 late days per team, no matter whether you are working individually or with teammates. These late days can be applied only to final project-related due dates. Be aware that some of the project components have hard deadlines.

These cannot be combined with your individual late days.

Final Project Grading

Project Management Tool [5 points]

- Graded during the final evaluation based on 0/5 points.
    - "5" is assigned, if:
        - At least 20 steps/milestones
        - There is a tracked activities every week by each team member
    - "0" is assigned for all other cases

Proposal submission:

- Graded based on 0/10 points. Evaluated within two weeks after the due date.
    - "10" is assigned, if the proposal is complete, realistic and includes all the details following the structure suggested (see 3. Submit the proposal)
    - "0" is assigned for all other cases

Checkpoint submission:

- Graded based on 0/20 points. Evaluated within two weeks after the due date.
    - "20" is assigned, if the checkpoint clearly shows progress towards the final submission following the proposal submitted.
    - "0" is assigned for all other cases
- **Note**: it is ok to slightly adjust your initial proposal, if your initial results are not as expected. In this case, please submit your updated proposal along with your checkpoint submission.

Final submission:

- Graded based on the X out of 50 points + bonus [if applicable]
- During the final evaluation, all the parts are evaluated, so please include a final version of all the parts of the assignment in your final submission.

Present your work:

- Graded based on the 0/15 points.
    - "15" is assigned, if:
        - whole team equally present the work
        - the presentation represents the work you will submit
        - the presentation follows a structure suggested
    - "0" is assigned for all other cases

Notes:

- Only files submitted on UBlearns are considered for evaluation.

- Files from local device/GitHub/Google colab/Google docs/other places are not considered for evaluation
- We strongly recommend submitting your work in-progress on UBlearns and always recheck the submitted files, e.g. download and open them, once submitted

Important Dates

February 23, Fri, 11:59pm -- Register your team (UBLearns > Groups)

March 1, Fri, 11:59pm -- Abstract is Due and setup PM tool

April 4, Thu, 11:59pm -- Checkpoint is Due

May 2, Thu, 11:59pm -- Final Submission Deadline

May 2 - May 6 -- Presentation [hard deadline, no late days can be applied]

Final Project Type

The Final Project is an opportunity for you to apply what you have learned in class to a problem of your interest. Potential projects usually fall into these two tracks:

Application Project
Pick an application that interests you, and explore how best to apply deep learning algorithms to solve it. If you're coming with a specific background and interests (e.g. biology, engineering, physics), you are welcome to apply deep learning models learned in this class to problems related to your particular domain of interest.

Algorithmic Project
Pick a problem or family of problems, and develop a new learning algorithm, or a novel variant of an existing algorithm, to solve it.

Possible datasets

You are welcome to refer to possible datasets below or use other dataset, that fits your problem.

Image-based datasets

- ImageNet: a large-scale image dataset for visual recognition organized by WordNet hierarchy
- Microsoft COCO: a benchmark for image recognition, segmentation and captioning
- YouTube Faces DB: a face video dataset for unconstrained face recognition in videos
- Meta Pointer: A large collection organized by CV Datasets.
- Places Database: a scene-centric database with 205 scene categories and 2.5 millions of labelled images
- Flickr100M: 100 million creative commons Flickr images

NLP datasets

- WikiText: A dataset of Wikipedia articles

- BookCorpus: A large-scale dataset of book text
- SQuAD (Stanford Question Answering Dataset): A dataset where questions are asked on a set of Wikipedia articles, with corresponding answers.
- MS MARCO (Microsoft MAchine Reading COmprehension): A large-scale dataset for machine reading comprehension tasks.

Time-series datasets

- Open Data Buffalo: Open data portal for the city of Buffalo
- US Government's Data: Official data portal of the US government
- Yahoo Finance: historical stock prices, financial statements, and more
- Yahoo Webscope: Yahoo's dataset collection

Reinforcement learning environments include

- MuJoCo by DeepMind [DeepMind article, documentation]
- Atari by Gymnasium [link]: A set of Atari 2600 environments
- Safety-Gymnasium: Safe Reinforcement Learning library
- PyBullet [details]: a well-supported env for robotics simulation
- SUMO-RL: a simple interface to instantiate RL environments with SUMO for Traffic Signal Control

List of Potential Projects

Image recognition and classification

Build deep learning models to perform image recognition or classification. Experiment with at least 4 complex architectures and setups and compare their performance on various parameters.

Possible architectures include:

- VGGNet
- GoogLeNet
- ResNet
- DenseNet
- MobileNet
- Transformer-based models

Object Detection

Develop a deep learning model to detect and localize objects in images or videos. Experiment with at least 2 complex architectures and setups, evaluate their accuracy and speed.

Possible architectures include:

- R-CNN
- YOLO (You Only Look Once)
- SSD (Single Shot MultiBox Detector)
- EfficientDet

Generative Adversarial Networks (GANs)

Implement a GAN to generate realistic images. You can generate images in various domains, such as faces, landscapes, or artwork. Experiment with at least 3 complex architectures and setups.

Possible architectures include:

- DCGAN (Deep Convolutional GAN)
- CycleGAN

- StyleGAN
- BigGAN

Natural Language Processing (NLP)

Work on a text-based project. Potential tasks include:

- Sentiment analysis
- Text classification
- Document summarization
- Machine translation
- Text generation
- Text style transfer

Select your task of interest and the dataset. Experiment with at least 3 complex architectures and setups applicable for this task.

Reinforcement Learning (RL)

Build an agent using deep reinforcement learning techniques to play a game or solve a task. Experiment with at least 3 complex algorithms and setups and compare the results. You are welcome to extend your final project for CSE 4/546 RL.

Possible deep RL algorithms include:

- Deep Q-Networks (DQN)-based algorithms
- Proximal Policy Optimization (PPO)
- Deep Deterministic Policy Gradient (DDPG)
- Soft-Actor Critic (SAC)

Time Series Analysis

Apply deep learning models to predict future values in time series data. Experiment with at least 3 complex architectures and setups.

Possible architectures include:

- RNN models (e.g. GRU, LSTM)
- CNN

- Attention-based models
- Transformer-based models
- Autoeconders

Anomaly Detection

Build a deep learning model to identify anomalies or outliers in datasets. This can be applied to various domains, such as fraud detection, network intrusion detection, or manufacturing quality control. Experiment with at least 3 complex architectures and setups and compare the results.

Possible architectures include:

- Autoencoders
- RNNS
- Self-Organizing Maps (SOM)

Efficient Training of Deep Learning

Parallel processing is a good skill that can be applied not just to deep learning problems, but also to many other applications.

For the final project you are expected to implement already existing algorithms and apply the parallel training and other tricks and compare the efficiency.

Explainable Deep Learning (XAI)

XAI helps to make the decision-making process transparent and understandable. Possible directions:

- Interpretable image classification.
  Develop a deep learning model for image classification. Implement techniques such as Grad-CAM, LIME, or SHAP to explain the model's predictions.

- Textual data analysis with explanations.
  Utilize attention mechanisms or other explainability techniques to highlight important words or phrases in the text that influenced the model's decision.

- Medical diagnosis with explainable models.
  Build a deep learning model for medical diagnosis. Use techniques like attention maps or saliency maps to explain the model's reasoning behind its diagnosis.

- Explainable recommender systems.
  Develop a recommender system using deep learning and make it explainable. Utilize techniques like matrix factorization with explainability or attention-based models to provide explanations

Propose you own topic

You may come to us with your own topic proposal! We understand that it might end up being pretty challenging, if you find out you are completely stacked, you are welcome to discuss your possible switching to any other directions.

Please talk to the course instructors to ensure the project you have in mind is feasible.

Here is a list of good resources to refer for more ideas on the final project:

- https://cs229.stanford.edu/proj2019aut/
- https://paperswithcode.com/
- https://github.com/kjw0612/awesome-deep-vision

GENERAL STRATEGY

- The main motivation of the final project is to explore novel ideas (either with the problem setup or the algorithm or both) or have a comparison of existing solutions. Start with a simpler problem and build on it going forward.

- You can use your implementations in A1 or A2 as a baseline to compare against other approaches that you will use for your project. Make sure to keep a proper citation.