**Instructor: Alina Vereshchaka**

**Final Project**

**Multiple deadlines, please see below**

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 include complex analysis and recent advances in deep learning. Consider this as an opportunity to build a project that you can add to your portfolio piece that demonstrates your skills.

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.

We have prepared a list of possible directions, please check the end of the description for more details.

1. Register your team (March 3)

You may work individually or in a team of up to 3 people. The evaluation will be the same for a team of any size. You can work with your teammates from A1 and/or A2. However, you cannot work with your partner from Paper&Code Presentation.

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

2. Project Management Tool [5 points] (March 7)

If you are working in a team of two or three people, set up a project management tool. E.g. [Github Project](https://github.com/features/issues) (free plan), [Trello](https://trello.com/pricing) (free plan).

1. Create a board with at least these columns: **To do, In Progress, Done**
2. Divide the project into at least 20 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. There should be tracked activities every week by each team member
4. Include at least 10 screenshots of your board activities and the link to your board as part of your Checkpoint and Final submissions. Submissions without the link won’t receive points.

If you are working individually, this step is optional, e.g. 5 pts will be applied automatically.

3. Submit the proposal [10 points] (March 7)

The project proposal is max 2 pages, single-spaced extended abstract outlining your project. 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.

Name as project\_proposal\_ TEAMMATE1\_TEAMMATE2.pdf

Submit at UBLearns > Assignments

4. Submit the checkpoint [20 points] (April 3)

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.
* Weights and datasets:
  + Go to [UBbox](https://www.buffalo.edu/ubit/ubbox.html) > CSE 676-B Final Project by TEAMMATE 1 & TEAMMATE 2 & TEAMMATE 3
  + Upload your chosen datasets and weights into this folder that you used for this assignment.
  + Copy a shared link to the UBbox folder, so it can be viewed by people in your company & it can be viewed and downloaded.

A screenshot of a computer

Description automatically generated

* + Add the link to the txt file, named as

project\_datasets\_TEAMMATE1\_ TEAMMATE2\_TEAMMATE3.txt

project\_weights\_TEAMMATE1\_ TEAMMATE2\_TEAMMATE3.txt

* + Submit this txt file as part of your submission on UBlearns
* All project files need to have clear naming, e.g.

project\_checkpoint\_TEAMMATE1\_TEAMMATE2.ipynb

project\_report\_TEAMMATE1\_TEAMMATE2.pdf

* All project files should be packed in a ZIP file named:

project\_checkpoint\_TEAMMATE1\_TEAMMATE2

* Submit at **UBLearns > Assessments**
* Suggested file structure:

project\_checkpoint\_TEAMMATE1\_ TEAMMATE2.zip

* + project\_checkpoint\_ TEAMMATE1\_ TEAMMATE2.ipynb
  + project\_report\_ TEAMMATE1\_ TEAMMATE2.pdf
  + project\_weights\_TEAMMATE1\_ TEAMMATE2.txt
  + project\_datasets\_TEAMMATE1\_ TEAMMATE2.txt
  + project\_pm\_tool.pdf

For the checkpoint, the primary goal is for us to assess your progress, your understanding of the chosen approach, and the initial results you've obtained. It's also an opportunity for you to demonstrate that your methodology is sound and that you are on track for the final project.

5. Submit the final results [50 points] (May 1)

* 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.

project\_TEAMMATE1\_TEAMMATE2.ipynb

project\_report\_TEAMMATE1\_TEAMMATE2.pdf

* All project files should be packed in a ZIP file named:

project\_TEAMMATE1\_TEAMMATE2\_TEAMMATE3

* Submit at **UBLearns > Assessments**
* Suggested file structure:

project\_TEAMMATE1\_ TEAMMATE2.zip

* + project\_TEAMMATE1\_ TEAMMATE2.ipynb
  + project\_report\_TEAMMATE1\_ TEAMMATE2.pdf
  + project\_weights\_TEAMMATE1\_ TEAMMATE2.txt
  + project\_datasets\_TEAMMATE1\_ TEAMMATE2.txt
  + project\_pm\_tool.pdf
* 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 1 – May 5)

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 5, 11:59pm**.

**Presentation details**

**Length:** 15 mins + follow-up questions

**Suggested Templates:** [UB branded templates](http://www.buffalo.edu/brand/resources-tools/ub-templates-and-tools/powerpoint-templates.html)

**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 +25 points]**

**Participate in weekly scrum meetings [5 points]**

Weekly Scrum is a regular group meeting led by our TAs starting from March 28 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, Wed or Fri, no need to join both the scrums per week.

**Requirements:** Your team must take part in a scrum meeting for at least 4 weeks to be eligible for the bonus. We encourage all team members to join scrum meetings.

**Deploy the model locally [5 points]**

Deploy your final project model locally or on a server. Demonstrating deployment skills shows you understand the full lifecycle of a DL project, from model building to practical application.

**Requirements:**

* Deploy your final project results locally on your machine or on a server (e.g., using Flask, Streamlit, or similar tools)
* Use an advanced UX/UI design for your project
* Record a 4-5 minute video showcasing your deployed model in action. Explain how it works and demonstrate its functionality.
* Upload your recording to UBBox and submit a txt file with a link to your recording as part of your submission.

**CSE Demo Days [5 points]**

The CSE Demo Days is a semesterly event where you can present your projects to a wider audience, including faculty, industry professionals, and fellow students. If you get interesting results, we encourage you to share your project with the public in terms of participating in the [CSE Demo Days](https://engineering.buffalo.edu/computer-science-engineering/news-and-events/events/cse-demo-days.html).

**Requirements:**

* Make a private piazza post and share your prior results before April 24.
* **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 24.
* If you receive a winning place award for project, **your bonus points will be +50!**

**Conference Paper Submission [10 points]**

If your project yields novel results or introduces an interesting approach to solve a real-world problem, we encourage you to convert it into a conference paper for submission to a research conference. You can get experience in scientific writing and further enhance your portfolio!

**Requirements:**

* Your project must present novel results or a novel approach and have a real-world application or present a novel theoretical concept.
* Before March 24, complete the form (<https://forms.office.com/r/f2faWAsdcD>)and include the details:
  1. What is the problem that you are addressing using AI methods?
  2. What are the existing solutions in this domain? What are the gaps and limitations in current approaches that your work can address.
  3. Share a high-level concept for your proposed solution. You can use flowchart or other tools to share a flow.
  4. What are the key AI methods you plan to use.
  5. Show any results that you have already obtained.
* Obtain confirmation from Prof. Alina that your project has paper potential. She will share all the further guidance.

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 past semesters' submitted works and 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](https://catalog.buffalo.edu/policies/integrity.html) 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
    - A link to PM tool & screenshots are submitted as part of the final submission
    - If the project is completed individually
  + “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

**March 3, Fri,** 11:59pm -- Register your team (UBLearns > Groups)

**March 7, Fri,** 11:59pm -- Abstract is Due and setup PM tool

**April 3, Thu,** 11:59pm -- Checkpoint is Due

**May 1, Thu,** 11:59pm -- Final Submission Deadline

**May 1 - May 5** -- Presentation[hard deadline, no late days can be applied]

**Final Project Directions**

For the final project you can deeply explore a specific DL area that excites you and showcase your skills.

**Possible datasets**

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

**Image-based datasets**

* [ImageNet](http://http/image-net.org/): a large-scale image dataset for visual recognition organized by [WordNet](http://wordnet.princeton.edu/) hierarchy
* [Microsoft COCO](http://mscoco.org/): a benchmark for image recognition, segmentation and captioning
* [YouTube Faces DB](http://www.cs.tau.ac.il/~wolf/ytfaces/): a face video dataset for unconstrained face recognition in videos
* [Meta Pointer: A large collection organized by CV Datasets.](http://www.cvpapers.com/datasets.html)
* [Places Database](http://places.csail.mit.edu/): a scene-centric database with 205 scene categories and 2.5 millions of labelled images
* [Flickr100M](http://yahoolabs.tumblr.com/post/89783581601/one-hundred-million-creative-commons-flickr-images): 100 million creative commons Flickr images

**NLP datasets**

* [WikiText](https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/): A dataset of Wikipedia articles
* [BookCorpus](http://yknzhu.wixsite.com/mbweb): A large-scale dataset of book text
* [SQuAD](https://rajpurkar.github.io/SQuAD-explorer/) (Stanford Question Answering Dataset): A dataset where questions are asked on a set of Wikipedia articles, with corresponding answers.
* [MS MARCO](https://microsoft.github.io/msmarco/) (Microsoft MAchine Reading COmprehension): A large-scale dataset for machine reading comprehension tasks.

**Time-series datasets**

* [Open Data Buffalo](https://data.buffalony.gov/): Open data portal for the city of Buffalo
* [US Government's Data](https://www.data.gov/): Official data portal of the US government
* [Yahoo Finance](https://finance.yahoo.com/): historical stock prices, financial statements, and more
* [Yahoo Webscope](https://webscope.sandbox.yahoo.com/): Yahoo's dataset collection

**Reinforcement learning environments include**

* MuJoCo by DeepMind [[DeepMind article](https://deepmind.com/blog/announcements/mujoco), [documentation](https://deepmind-mujoco.readthedocs-hosted.com/en/latest/overview.html)]
* [Safety-Gymnasium](https://github.com/PKU-Alignment/safety-gymnasium): Safe Reinforcement Learning library
* PyBullet [[details](https://pybullet.org/wordpress/)]: a well-supported env for robotics simulation
* [SUMO-RL](https://github.com/LucasAlegre/sumo-rl): a simple interface to instantiate RL environments with SUMO for Traffic Signal Control

**List of Potential Projects**

**Category 1: Computer Vision Applications**

* + - 1. **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

Real-world applications: autonomous vehicles, robotics, and surveillance, Medical Image Analysis

* + - 1. **Object Detection**

Develop a deep learning model to detect and localize objects in images or videos. Experiment with at least 3 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
* DETR
  + - 1. **Generative Adversarial Networks (GANs)**

Implement and investigate GANs for the task of realistic image synthesis within a defined domain (e.g., faces, animals, landscapes). Experiment with at least 3 GAN architectures.

**Possible architectures include:**

* DCGAN (Deep Convolutional GAN)
* CycleGAN
* StyleGAN
* BigGAN

**Category 2: Natural Language Processing (NLP)**

* + - 1. **Advanced Sentiment Analysis**

Develop a sentiment analysis system capable of discerning emotions, sarcasm, and context-dependent sentiment within textual data, such as social media discourse or customer feedback.

Employ Transformer-based architectures like BERT, RoBERTa, or DistilBERT and evaluate performance on datasets like Sentiment140 or Twitter US Airline Sentiment.

* + - 1. **Question Answering Systems**

Construct a Question Answering (QA) system that integrates knowledge graphs to provide contextually enriched and more informative answers. Implement architectures such as BERT or Transformer-based QA models in conjunction with a knowledge graph (e.g., Wikidata, DBpedia, or a custom-built graph). Train and evaluate using datasets like SQuAD or MS MARCO.

* + - 1. **Text Summarization**

Develop a text summarization system focused on abstractive summarization methodologies, aiming to generate concise summaries of lengthy and complex documents (e.g., legal or financial texts).

Apply architectures such as Transformer-based models (BART, T5) or Pointer-Generator Networks and evaluate their efficacy on relevant datasets (e.g., CNN/DailyMail dataset or specialized legal/financial document corpora).

* + - 1. **Dialogue Systems**

Design and implement a dialogue system capable of engaging in multi-turn conversations and providing contextually appropriate and engaging responses.

Explore architectures such as Transformer-based dialogue models (DialoGPT, BlenderBot) or Recurrent Neural Network-based models with attention mechanisms. Evaluate performance on dialogue datasets like DailyDialog or Persona-Chat.

**Category 3: Deep Learning for Time Series Forecasting and Anomaly Detection**

* + - 1. **Deep Learning for Industrial Equipment Failure Prediction**

Develop a time series forecasting model for predictive maintenance in manufacturing, aimed at forecasting machine failures using sensor data. Implement architectures such as LSTMs, GRUs, or Transformer-based time series models and evaluate their robustness to noisy data and their efficacy in providing early and reliable failure predictions.

* + - 1. **Financial Time Series Forecasting**

Construct a financial time series forecasting model that predicts market trends and also incorporates risk assessment and uncertainty quantification.

Investigate architectures such as LSTMs, GRUs, or Transformer-based models and integrate risk metrics like Value at Risk (VaR) or Expected Shortfall. You can use financial datasets such as Yahoo Finance data and explore methods for quantifying forecast uncertainty.

* + - 1. **DL for Network Traffic and Cybersecurity Analysis**

Create an anomaly detection system using Deep Learning to identify anomalous patterns in network traffic or cybersecurity logs, indicative of potential security threats or network malfunctions.

Investigate techniques such as Autoencoders, RNN-based anomaly detection, or Transformer-based anomaly detection and evaluate performance on network traffic or cybersecurity log datasets (e.g., KDD Cup '99, NSL-KDD).

* + - 1. **DL for Supply Chain Optimization in Retail**

Develop a demand forecasting model using Deep Learning to optimize inventory management and supply chain operations, particularly within the retail sector.

Implement architectures such as LSTMs, GRUs, Transformer-based time series models, or hybrid CNN-RNN models. Address challenges related to seasonality and external factors influencing demand.

**Category 4: Novel DL Architectures**

* + - 1. **Benchmarking State-of-the-Art Models**

Conduct an in-depth evaluation of a recently published Transformer architecture variant (within the last two years, e.g., Perceiver, Longformer, Reformer).

Implement the chosen architecture and benchmark its performance against established models on a demanding task within NLP or Computer Vision (e.g., long document classification, image generation, few-shot learning).

* + - 1. **Self-Supervised Learning**

Investigate and implement a self-supervised learning methodology (e.g., Contrastive Learning methods such as SimCLR, MoCo, BYOL; Masked Autoencoders - MAE) on an appropriate dataset (image or text).

Evaluate the effectiveness of the learned representations by pre-training a model and subsequently fine-tuning it for a downstream supervised task using limited labeled data.

* + - 1. **Novel Regularization Techniques**

Conduct a research-oriented project exploring recent advances in regularization techniques for DL. Implement and experimentally evaluate at least 5 or more novel regularization methods (beyond standard L1/L2 regularization, Dropout) to improve the generalization, robustness, or training efficiency of a DL model in a chosen application.

* + - 1. **Deep Learning for Niche Applications**

Identify a real-world problem within a less commonly addressed domain (e.g., agriculture, environmental science, social welfare, assistive technologies, specialized scientific domains).

Develop a Deep Learning-based solution tailored to the specific challenges and data characteristics of this niche application. This may involve adapting existing architectures or proposing novel model designs.

* + - 1. **Reinforcement Learning (RL)**

Build an agent using deep reinforcement learning techniques to solve a real-world problem. Experiment with at least 3 complex algorithms and setups and compare the results. You can 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)
  + - 1. **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 your 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>
* <https://introtodeeplearning.com/2019/materials/2019_6S191_Projects.pdf>

**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.