02456 Deep learning 2023 - course plan and information

Time: Mondays at 13:00-17:00 (first session is August 28th, 2023)

Locations: We will use the following rooms - building/room - (Campus map):

B303A-A042
B303A-046
B303A-047
B303A-048
B303A-HOEST
Zoom (You need to sign-in with you DTU account)

We use flipped classroom teaching. During the weeks with labs, the teachers and teaching assistants will circulate between the rooms so there will be opportunity to meet all. Any short lectures/instructions will be repeated in all rooms. You are free to choose whatever room you prefer of course respecting the limits on room capacity. During the weeks with project work each room will cover specific topics.

If you are not able to be on campus or prefer to work remotely you will be able to participate through Zoom. One teaching assistant will be dedicated to the Zoom channel: Zoom.

We also use Slack for communication: We will make dedicated channels for labs and projects. Here is a Slack invite link. (In Slack you can add channels from the list of channels by clicking the "+" next to Channels in the left panel and click "Browse channels" to choose.)

Bring a laptop.

The first eight weeks of the course will be dedicated to lab work. There will be a brief introduction to the course at the first session and a number of dedicated meetings online or in person with project supervisors.

Teachers

- Ole Winther
- Jes Frellsen

Teaching assistants

- Aleksander Nagaj
- Anders Christensen
- Anna Maria Clara Schibelle
- Anshuk Uppal
- Beatrix Miranda Ginn Nielsen
- Bo Li
- Kenny Olsen
- Marco Miani
- Nina Weng
- Paul Jeha
- Pawel Tomasz Pieta
- Raul Ortega Ochoa
- Teresa Karen Scheidt
- Thea Brüsch

Google CoLab

Google CoLab is a free cloud based Jupyter notebook platform with collaboration functionality. It even has GPUs and you don't need any credits, just log in with your Google account. To start, import a notebook using a github link or upload it from your pc: https://colab.research.google.com/. Setting up is quite straightforward. If you need to install libraries you can add that in a code cell with `` `! pip install <library name>` ``. You can upload some extra files (such as additional py scripts) that your jupyter notebook will use.

Other free GPU compute resources

It might be that Google CoLab will start putting restrictions if you use it too much. But there are alternatives:

DTU HPC

Nicklas Hansen has made this guide.

Google cloud platform (GCP)

You can also sign up to Google cloud (GCP) and get free credits there. Here are instructions on how to set up GPU computation on Google cloud. Follow the instructions until Optional.

Topics in the first eight weeks

1. Introduction to statistical machine learning, feed-forward neural networks (FFNN) and error back-propagation. Part I do it yourself on pen and paper.
2. Introduction to statistical machine learning, feed-forward neural networks (FFNN) and error back-propagation. Part II do it yourself in Python.
3. Introduction to statistical machine learning, feed-forward neural networks (FFNN) and error back-propagation. Part III PyTorch.
4. Convolutional neural networks (CNN)
5. Transformers and recurrent neural networks (RNN)
6. Tricks of the trade and data science with PyTorch
7. Variational learning and generative adversarial networks for unsupervised and semi-supervised learning + deadline for selection of student projects on Friday, Oct 13th 2023 at 23.59.
8. Reinforcement learning - policy gradient and deep Q-learning + start of student projects.

Week 9-13 will be only project work

In the seven project weeks we will still meet on Mondays for project work and supervision.

Evaluation and peer grading during the course

Evaluation:

1. The course is graded using the 7-step scale.
2. The final grade is based solely on the evaluation of the final project, which starts in the 7th week of the course. The project group should consist of 3-4 students. In special circumstances we can also accept groups of 1 or 2 students. (In the course catalogue it says 1-3 students.

We will correct that for next year but cannot change it now.)
3. The evaluation of the final project is based on two parts, both of which are done in groups but evaluated individually:
    a. a poster exam presentation, where the project groups document the results of their project in a poster and present to two or more teachers acting as examiners and
    b. a report in which the project groups document their solution. The report should be a maximum of 6 pages plus references using this conference paper format.

    More details are given below.
4. The student gains access to the final project by passing 6 out of 8 lab sessions that precede it.
5. A lab session is passed by:
    a. grading the reports from lab sessions of 3 other students on Peergrade and
    b. passing the lab as judged by the teacher. More details given below.

More details on peer grading: The 8 lab sessions are evaluated using peer grading. We use peer grading to ensure more accurate evaluation and better feedback. Graders get 3 reports at each deadline and have one week to carry out the feedback. If you forget to perform your peer grading it is not nice to your fellow students, but you can still pass that lab for which you forgot to grade.

Handing in and peer grading six of the eight labs reports is required for being able to execute the project and eventually pass the course. If a hand-in is not passed you will be contacted with the option of re-submitting the lab directly to the teacher so if you hear nothing assume that you have passed the lab. You can also contact the teacher directly on Slack if something went wrong with the submission of the lab. Peergrade deadlines are strict so no need to write about getting an extension.

The following reports should be handed in jupyter notebook format. *The weeks refer to weeks in term, and the fall break week is not counted.*

1. Week 1 computer exercise. Deadline: Monday week 2.
2. Week 2 computer exercise. Deadline: Monday week 3.
3. Week 3 computer exercise and 1 exercise of your own choice from course material week 1. Deadline: Monday week 4
4. Week 4 computer exercise and 1 exercise of your own choice from course material week 1-2. Deadline: Monday week 5.
5. Week 5 computer exercise. Deadline: Monday week 6.
6. Week 6 computer exercise. Deadline: Monday week 7.
7. Week 7 computer exercise and 1 exercise of your own choice from course material week 1-3. Deadline: Monday week 8
8. Week 8 computer exercise and 1 exercise of your own choice from course material week 1-3. Deadline: Monday week 9.
9. **Project selection. Deadline Friday, Oct 20th 2023 at 23.59**.
   Link to 2023 project selection sheet
10. **Project synopsis. Deadline: Monday week 9 at 23:59**. The synopsis should be approximately half a page and maximum one page with a project title, motivation, background, milestones and references. It is important that the plan is realistic. The main purposes of the synopsis are to make sure the project size is well-calibrated and is concrete enough to start working from day one. The synopsis will not be used in the evaluation. The synopsis should be sent to your project supervisor.
11. Project poster session. PhD students taking the course as part of their PhD *will not have to make a poster and take part of the poster session*. In mixed groups of PhD and non-PhD students, only the non-PhD students have to take part in the poster session. The exam date is December 7th from 9 to 17. We divide the day into half hour slots and your group will later be given the possibility to register for a slot. A link to sign up for the poster session will appear here in due time. So having another exam on the same day should not be a problem. We will also organise an extra exam date for those of you who cannot make it on the date. It will be group poster presentations. We will invite outside guests and

we will walk around and ask questions to all groups. We will make a schedule for when the teachers visit your poster. Plan for a 2 minute presentation per group member and 1-2 minutes for questions. The remainder of the time you can either present your poster to other students and guests or go visit other posters. Remember that it is important for the overall impression that you divide the presentation and answering of the questions more or less equally between you. The poster should be in A1 format. Remember to put both your names and student numbers under title. Here and here are links to examples using the latex template and here is one in powerpoint. You do not have to use that. The DTU library offers poster printing for a not too high price.

12. **Final report deadline December 21st at 23:59**. [Note this was earlier set to a later date but according to DTU rules, the latest allowed deadline is December 21st.] The report should be a maximum 6 pages plus references using this conference paper format. The report should also contain a link to your project code Github repository. Among the files in the repository should be a jupyter notebook that ideally should recreate the main results of your report. If some of your data is confidential then use some shareable data instead. For MSc students, please also include your poster in the submission.

Detailed content

Links to individual video lectures and lecture slides are given below. Here is a link to all 2016 video lectures as a playlist and a Google doc folder with all the lecture slides. More videos have been added over the years. They are all linked below. A very good alternative video resource is Hugo Larochelle's YouTube playlist.

Week 1 - Feed-forward neural networks - do it yourself pen and paper

1. During *this week and the following two weeks* watch video lectures:
    a. Part 0 Overview
    b. Part 1 Deep learning
    c. Part 2.1 Feed-forward neural networks
    d. Part 2.2 Feed-forward neural networks
    e. Part 3 Error Backpropagation
    f. Part 4 Optimization

    and take notes for at least 3 questions to ask. Link to lecture slides is here.
2. During *this week and the following two weeks* read Michael Nielsen, Neural networks and deep learning http://neuralnetworksanddeeplearning.com/ Chapters 1-3 (stop when reaching the section called Overfitting and regularization) and browse Chapter 4. Note that this is reading material for the first three weeks of the course. Also, in total six exercises of your own choice will be homework later in the course.
3. Alternative textbooks: All topics are also covered in the deep learning book that may be read as a supplement. The book can also be bought from the DTU bookstore. You will get 10% discount with this link. Feed-forward neural networks are covered in this chapter. Chapter 1 gives an introduction to deep learning and Part II gives the necessary background on linear algebra, probability, numerical computation and machine learning. Alternative textbook 2: Chris Bishop, Pattern recognition and machine learning. If you need to up your game in mathematics, the book Mathematics for machine learning is an excellent resource and the note Mathematics for Machine Learning offers a concise and compressed collection of the mathematical concepts used in deep learning. These resources are freely available online and are very valuable sources of information.
4. Install software on your laptop or go directly to Google CoLab (see above). Installation guide for laptop and cloud may be found here.

5. Carry out computer exercises week 1. It is encouraged to work together with other students. Type in everything yourself. Code answers are fine not to differ much within the group and text answers should be in your own words. Note that the computer exercises may experience minor change up to 3 days before the actual session. The hand-in is the notebook with your modifications. It is only allowed to hand in .ipynb files. Each week you should only hand in one file. It is the file with EXE in its name. You hand in on peergrade.io. In order to be able to hand in on peergrade you can use this invitation link https://app.peergrade.io/join/JFFJBF or login into peergrade and use the invitation code JFFJBF. There you will receive information on handing in exercises and deadlines for activities. Some students have previously by accident signed up twice with different emails or forgotten the "student" in their DTU student mail address. If you then submit and check with different email addresses it will look as though you have not handed in.
6. Peergrade exercise from three other students through peergrade.io.

Week 2 - Feed-forward neural networks - do it yourself in NumPy

1. See 1. and 2. from Week 1.
2. Carry out computer exercises week 2.
3. Peergrade exercise from three other students through peergrade.io.

Week 3 - Feed-forward neural networks in PyTorch

1. See 1. and 2. from Week 1.
2. Carry out computer exercises week 3.
3. Peergrade exercise from three other students through peergrade.io.
4. Hand in the notebook marked with EXE on peergrade.io. It should contain your added code in the exercises and the answer of one exercise from Michael Nielsen's book (see point 3. above). The answer to the book exercise should be in a markdown cell at the bottom of the notebook.
5. Peergrade exercise from three other students through peergrade.io.

Week 4 - Convolutional neural networks

1. Watch week 2 video lectures
    a. Part 1 Introduction to CNNs (PART 1/2)
    b. Part 1 Introduction to CNNs (PART 2/2)
    c. Part 2 CNNs the details (PART 1/2)
    d. Part 2 CNNs the details (PART 2/2)
    e. 2017 CNN update
    f. 2017 Activation functions update
    g. 2017 Image segmentation

    and take notes for at least 3 questions to ask. Link to lecture slides is here and here for 2017 updates.
2. Reading material Michael Nielsen, Neural networks and deep learning http://neuralnetworksanddeeplearning.com/ Chapter 6 (stop when reaching section called Other approaches to deep neural nets).
3. Alternative textbook chapter in the deep learning book.
4. One exercise from the book chapters.
5. Carry out computer exercises week 4.
6. Hand in the notebook marked with EXE on peergrade.io.
7. Peergrade exercise from three other students through peergrade.io. You will receive instructions about this

from peergrade.io.

Week 5 - Transformers and recurrent neural networks

1. Watch week 3 video lectures
    a. 02456week3 1 RNN (PART 1 of 3)
    b. 02456week3 1 RNN (PART 2 of 3)
    c. 02456week3 1 RNN (PART 3 of 3)
    d. 02456week3.2_RNN_training (PART 1 of 3)
    e. 02456week3.2_RNN_training (PART 2 of 3)
    f. 02456week3 2 RNN training (PART 3 of 3)
    g. 02456week3 3 Attention (PART 1 of 2)
    h. 02456week3 3 Attention (PART 2 of 2)
    i. 02456week3 4 Supervised learning recap
    j. 2017 Quasi RNN
    k. 2017 Non-recurrent sequence to sequence models
    l. 2017 Text summarization
    m. 2020 Transformers (PART 1 of 2)
    n. 2020 Transformers (PART 2 of 2)
    o. 2020 Language modelling - GPT-2 and 3
    p. 2020 BERT

    and take notes for at least 3 questions to ask. Link to: 2016 lectures, 2017 lecture updates and 2020 lecture updates.
2. Reading material Alex Graves book, Supervised Sequence Labelling with Recurrent Neural Networks Chapters 3.1, 3.2 and 4. Browse Michael Nielsen, Neural networks and deep learning Chapter 6 section Other approaches to deep neural nets) and onwards. A good introduction to Transformers is The Illustrated Transformer. New tutorial on Transformers https://aman.ai/primers/ai/transformers/#one-hot-encoding
3. Alternative textbook chapter in the deep learning book. Andrej Karpathy has a nice blogpost that gives a good flavour of the whats and hows of RNNs.
4. Carry out computer exercises week 5
5. Hand in and peergrade on peergrade.io like in previous week.

Week 6 - Tricks of the trade and data science challenge

1. Watch week 4 video lectures
    a. 02456week4 1 1 Initialization and gradient clipping
    b. 02456week4 1 2 batch normalization
    c. 02456week4 2 1 regularization
    d. 02456week4 2 2 regularization methods
    e. 02456week4 2 3 data augmentation
    f. 02456week4 2 4 ensemble methods and dropout
    g. 02456week4 3 recap
    h. 2017 37 reasons your nn working (part 1 of 2) Walk through of the 37 reasons why your neural network is not working blog post.
    i. 2017 37 reasons you not working (part 2 of 2)
    j. 2020 Recipe to training neural networks - become one with data (part 1 of 3).
    k. 2020 Recipe to training neural networks - baselines (part 2 of 3).
    l. 2020 Recipe to training neural networks - overfit, tune and tune some more (part 3 of 3).

    and take notes for at least 3 questions to ask. Link to lecture slides 2016 lecture slides, 2017 blog post and 2020 lecture slides.
2. Reading material Michael Nielsen, Neural networks and deep learning http://neuralnetworksanddeeplearning.com/ Chapter 3 from section Overfitting and regularization and Chapter 5.
3. Alternative textbook chapters on regularization, optimization, deep learning practice and applications from the deep learning book.

4. Additional material: Andrei Karpathy blogpost on how to approach a data science problem with deep learning, blogpost on things that can go wrong in neural network training and interactive initialization demo.
5. Computer exercises week 6 using PyTorch on the Kaggle competition leaf classification. Hand in and peergrade on peergrade.io like in previous weeks.

Week 7 - Un- and semi-supervised learning

1. Watch week 5 video lectures
    a. 02456week5 1 1 unsupervised learning
    b. 02456week5 1 2 unsupervised learning latent variables
    c. 02456week5 2 1 autoencoders
    d. 02456week5 2 2 autoencoders layerwise pretraining
    e. 02456week5 3 1 variational autoencoders
    f. 02456week5 3 2 semi-supervised variational autoencoders
    g. 2017 Generative adversarial networks
    h. 2020 Flows
    i. 2020 Self-supervised learning
    j. 2020 Self-training/noisy student
    k. 2020 Distribution Augmentation
    l. 2020 Flat minima

    and take notes for at least 3 questions to ask. Link to lecture slides 2016 slides and 2017 slides and 2020 slides.
2. Reading material DL Chapter 14 and 20.10.3. (Further learning a course dedicated to generative modelling.)
3. One exercise from the book chapters.
4. Carry out computer exercises week 7 on autoencoder un- and semi-supervised. Hand in and peergrade on peergrade.io like in previous weeks.
5. Project selection deadline is this week (see above).

Week 8 - Reinforcement learning

1. Watch week 6 video lectures
    a. 02456week6 1 1 reinforcement learning
    b. 02456week6 1 2 reinforcement learning approaches
    c. 02456week6 2 1 AlphaGo policy and value networks
    d. 02456week6 2 2 AlphaGo steps 1 to 4
    e. 02456week6 3 policy gradients
    f. 02456week6 4 a few last words
    g. 2017 Deep Q learning
    h. 2017 Evolutionary strategies

    and take notes for at least 3 questions to ask. Link to lectures here and here for 2017 update.
2. Reading: another nice blog post by Andrei Karpathy. Optional reading material on the connection between variational and reinforcement learning.
3. One exercise from the book chapters.
4. Computer exercises on reinforcement learning methods (policy gradient, deep Q learning, evolutionary strategies) in the openAI Gym. Carry out exercises week 8. Hand in and peergrade on peergrade.io like in previous weeks.
5. Project work.

Project list 2023

To get more information about select projects, you can contact supervisors. Potential supervisors will organise meetings presenting their projects to interested students.

1) **Come with your own project.** Requirements: data and problem statement are in place so that time will be spent on modelling. Ideally, team up with other students but one student teams may be accepted in special

circumstances. Supervised by Ole Winther, olwi@dtu.dk, Jes Frellsen, jefr@dtu.dk and others.

2) **Generative AI for teaching and learning.** gnrtive.ai is a startup designing AI digital twins that retain memories of past learning experiences to link current abilities with new knowledge, and personalize teaching by enabling OpenAI's GPT 3.5 turbo to interact with existing course material and adapt to individual learning styles. Going beyond Q&A bots the aim is to turn it into a simulation engine, that based on RAG retrieval augmented generation initiates multiple collaborating professor agents. Agents are divided into tutor, mentor and coaching roles dependent on the context by integrating vector store memories into prompt templates using CoT chain of thought reasoning. The aim of the project is to prototype a LangChain framework https://learn.deeplearning.ai/langchain/lesson/1/introduction combining aspects of prompt engineering with retrieval augmented generation https://learn.deeplearning.ai/langchain-chat-with-your-data/lesson/1/introduction extended into a multi-agent architecture resembling the architecture of MetaGPT https://github.com/geekan/MetaGPT. Supervisor Michael Kai Petersen, (michaelkaipetersen@gmail.com).

3) **Generative modelling of magnetic fields and potentials.** Magnets are critical for energy and infrastructure (e.g. wind turbines, electric vehicles) but our ability to design novel magnetic materials and configurations is limited if we use non-probabilistic simulations. Taking clever advantage of the inductive biases built into physical laws, we can design and implement novel architectures that accurately predict fields around magnets. In this project, the student(s) might choose to: i) build a generative architecture for modelling two- and three-dimensional magnetic systems; ii) implement a Bayesian architecture that allows for uncertainty or missing data; or iii) consider the use of magnetic potentials, i.e. scalar functions from which the fields can be derived through (automatic) differentiation.

   This project will be exciting for anyone looking to develop deeper ability with the JAX ecosystem, a curiosity about the deep relationships between physics, symmetry and machine-learning, and looking to build on a topic with industrial and social impact. Supervised by Berian James (berjo@dtu.dk) and Stefan Pollok (spol@dtu.dk).
4) **Neural audio coding for speech enhancement.** Supervisor: Kenny Falkær Olsen (kfaol@dtu.dk). Neural audio codecs consist of encoder/decoder neural networks, which encode audio streams into highly compressed latent codes from which the decoder can approximately recover the original audio with high fidelity. Current state-of-the-art codecs are based on residual vector-quantizing variational autoencoders (e.g. https://arxiv.org/abs/2306.06546), which encode and quantize audio at a fixed target bitrate. In this project we will investigate how to specialize neural audio codecs for speech enhancement, with an eye towards subjective quality and real-time applicability.

   Possible project directions include:
   - Fine-tuning existing codecs for speech enhancement.
   - Performing speech enhancement in the compressed latent domain.
5) **Input Privatization for Safely Using Foundation Model APIs.** Supervisor Mi Jung Park (mjupa@dtu.dk). More information here (project 1)
6) **Differentially private sampler guidance for diffusion models.** Supervisor Mi Jung Park (mjupa@dtu.dk). More information here (project 2)
7) **Does well-calibrated uncertainty come for free with federated learning?** This project aims to investigate if we can obtain a good uncertainty estimate in a federated learning framework and what factors in FL can improve the quality of the estimated uncertainties. Project details can be found at https://github.com/lyn1874/project_ideas/blob/main/does_well_calibrated_uncertainty_come_for_free_w Supervisor: Bo Li (blia@dtu.dk).

8) **Predicting transmembrane protein topology from 3D structure.** Use geometric deep learning/ graph neural

network techniques to improve performance on a protein property prediction task using 3D structure data. Details at https://docs.google.com/document/d/1kIJNC9K82o9T-NpzwTbW-iGXsugsUVgOlzNsQDtSVgU/edit?usp=sharing Supervisor Felix Teufel felix.teufel@gmail.com and co-supervisor Ole Winther (olwi@dtu.dk).

9) **Deep learning projects with Turbulence Research at DTU.** Supervised by Clara Marika Velte (cmve@dtu.dk) and Simon Lautrup Ribergård (silari@dtu.dk):

    a) **Improvements to particle identification in 4D-LPTV data.** Lagrangian Particle Tracking Velocimetry is an evolving method for recording velocity fields in fluid dynamics. By tracking individual particles and fitting paths to their position in time, that adheres to conservation laws, a very good representation of a given flow is achieved. However, the current methods for determining whether what is seen in the image is indeed a particle, a very naïve approach is taken that simply looks for an intensity threshold. This method is not nearly as good as the human eye at detecting particles, since their shape and brightness vary quite a bit. In this project, the student would attempt to produce a new method for particle detection in a 2D-image in such a way that the method produces more positive identifications of particles in a more reliable manner than the existing method. Shake-the-box (4D-LPTV): https://link.springer.com/article/10.1007/s00348-016-2157-1

    b) **Improvements to particle reconstruction in 4D-LPTV data.** Similarly to the project above, this would be an improvement to a method used for Lagrangian Particle Tracking Velocimetry in which details of complex flows can be investigated. Initially, particles are identified on 2D-images and their position in space extended through an epipolar line from multiple cameras from different spatial perspectives. Combined with information on calibration and positioning of the viewpoints, this reconstructs a particle position in 3D. This project would allow the student to investigate if it is possible to increase the number of successfully reconstructed particles while lowering the number of so-called ghost particles (particles seem like they should exist, but do not). Shake-the-box (4D-LPTV): https://link.springer.com/article/10.1007/s00348-016-2157-1 Advanced iterative particle reconstruction: https://link.springer.com/article/10.1007/s00348-021-03276-7

    c) **Improvements to track generation in 4D-LPTV data.** Once particles have been identified and reconstructed, tracks must be fitted to their positions. This is often done by the use of correlation methods, however Shake-the-box (STB) has sped the tracking up by only using correlation on the first few time steps, after which the fitted track is used to extrapolate a guess as to where the same particle should be found in the next time step. Adding a constraint for the fitted tracks to make sure they adhere to physical conservation laws while successfully generating at least the same number of tracks as STB, preferably faster, would be a challenge, but a big improvement.

    d) **Improvements to image pre-processing for 4D-LPTV data.** The first step when conducting 4D-LPTV experiments is always to clean up the recorded images to lower noise and improve the desired signal. This is done by enforcing a background with a grey-scale intensity count of 0, while retaining the information for each particle in the image regarding size, shape and intensity. The naïve approach is to simply subtract a flat amount of intensity from every pixel in the image, but an ML-approach could be to train a NN to identify what parts of the image are particles and what is background and treat these differently.

    e) **Eulerian gradients from 4D-LPTV data.** The result of an experiment using Lagrangian Particle Tracking Velocimetry is typically a cloud of time-

dependent tracks passing through the volume of interest. It would be of interest to be able to infer the spatial and temporal gradients of a point inside this volume of interest, say the centroid, without having to go through first interpolating on to an Eulerian grid. This should reduce the amount of interpolation and discretization errors introduced and would be a lot faster. Velocity gradients from LPTs: https://scholarworks.calstate.edu/downloads/dv140167c

10) **Molecular Property Prediction using Graph Neural Networks.** Supervised by Jonas Vestergaard (jovje@dtu.dk). Molecular property prediction plays an important role in materials science and drug discovery, and accurate predictions can improve the efficiency of research and development (R&D) pipelines. While accurate predictions traditionally require resource-intensive quantum mechanical simulations, recent advancements introduce a promising alternative - Graph Neural Networks (GNNs). Not only do GNNs demonstrate high accuracy in predicting molecular properties, but they also offer substantial computational speed-ups, bypassing the limitations associated with quantum simulations.

    Project Overview: The primary objective of this project is to implement the state-of-the-art Polarizable Atom Interaction Neural Network (PaiNN) for molecular property prediction. We will train and test the model(s) on the QM9 dataset, a benchmark dataset in this domain. Additionally, we will explore if PaiNN can be improved with Stochastic Weight Averaging.

    Potential Extensions (if time permits):

    1. Bayesian modeling: Extend PaiNN to a Bayesian model, employing techniques such as SWA-Gaussian (SWAG) or Gaussian Mean-Field Variational Inference.

    2. Layer optimization: Investigate the impact of varying the number of message-passing layers in PaiNN, with a particular focus on addressing potential over-smoothing - a common challenge in GNNs.

    3. Comparison with other architectures: Implement and evaluate other state-of-the-art GNN architectures designed for molecular property prediction.

    4. Wider evaluation: Assess the generalizability of the model(s) by evaluating performance on additional datasets beyond the QM9 dataset.

    Introductory resources on GNNs:

    - Chapters 5 and 6 in Graph Representation Learning.

    - Stanford's lecture slides on Equivariant GNNs.

11) **Bioinformatics projects** A number of projects are available supervised by Ole Winther, olwi@dtu.dk
    a) **Bioinformatics students come with your own project.**
    b) **Work with AlphaFold DB and FoldSeek** to understand the protein structure universe. A major clustering analysis was carried out by the FoldSeek group in Barrio-Hernandez et al, 2023 ideas are welcome. In this project you will carry out a more detailed analysis for particular types of proteins (e.g. membrane proteins and carbo-hydrate active enzymes provided by Ole) using the clustering tool from the paper and fine-tuned protein language models.
    c) **Predicting protein-ligand interactions.** We will work from this dataset and benchmarking paper with the accompanying Github page found

here. In the project you will run validations for trained models both on the validation set provided and new (small) ones deposited to PDB after the paper cut-off. You will also run your own training on simpler models to investigate the effect of model size.

d) **Neural-ODE for pharmacokinetics (PK) modelling.** Use the implementation from LU et al, 2021 found here to model pharmacokinetics datasets available from the nlmixr2 R PK package for example theo_sd.

12) **Denoising diffusion probabilistic models for time-series and images** supervised by Ole Winther (olwi@dtu.dk). Background reading material: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/, https://arxiv.org/abs/2006.11239 and https://arxiv.org/abs/2201.11793. Implementations: Stable diffusion, Dalle 2, https://iterative-refinem$ent.github.io/ and https://github.com/google-research/vdm. You first task will be to perform a literature study to find relevant references and resources published since those listed above.

13) **Natural language processing (NLP)**. A number of projects in NLP continuing the topics from the Transformer lab (week 5) is available. Supervised by Ole Winther (olwi@dtu.dk) and potential projects include: To be updated!
    a) **NLP students come with your own project.**
    b) **Large language models for teaching-learning systems** The paper Assigning AI discusses how to use large language models for different teaching roles (tutor, mentor, …). The goal of this project is to make LLM agents that can solve different tasks in a teaching-learning platform. Teaching material such as texts and/or transcribed lectures are available and the system should for example be able to assist the teacher with making test questions and assist the student with helping understanding the material, finding answers and summarising, for example questions like “summarize the lecture from minute 5 to 15”, “make three questions to test the students understanding of the lecture note” and “explain the steps to prove the theorem…”. We will use the Instructor framework with more detail here and LangChain.
    Steps in the projects:

        i) Find some teaching material online you want to build the system on.

        ii) Get to know langchain that we can use for indexing and search in data and perhaps also some of the interaction with the language model.

        iii) Set up an OpenAI account and play with completion.

        iv) Play with the instructor framework and build functionality for specific tasks.

        v) Make a simple user interface to run the solution.

14) **Implementation of DDPM.** The project is to re-implement the Denoising Diffusion Probabilistic Model in PyTorch or Jax and reproduce their results at least on MNIST and ideally on CIFAR-10. This paper is the one that kicked off the diffusion movement, it is a great way to learn what diffusion is all about and have hands-on experience. Link to paper: https://arxiv.org/abs/2006.11239. Supervised by Paul Jeha (pauje@dtu.dk).

15) **Implementation of Score Based Model.** This project is to re-implement Generative Modeling by Estimating Gradients of the Data Distribution in PyTorch of Jax and

reproduce their results at least on MNIST and ideally on CIFAR-10. This paper is an alternative but very popular view on diffusion models that will challenge your understanding of what a diffusion model is and provide you with hands-on experience. Link to paper: https://arxiv.org/abs/1907.05600. Supervised by Paul Jeha (pauje@dtu.dk)

16) **Physics Informed Neural Networks (PINNs).** Numerical simulation is of major importance in industrial design processes. Data-driven PINNs are emerging as maturing techniques within scientific machine learning (SciML) that utilize model-based approaches in digital twins concepts, simulation, optimization and control problems, etc. PINN utilize mathematical models (instead of / combined labelled data) together with automatic differentiation (in open source neural network software) stated in terms of differential equations to find accurate approximate solutions of differential equations, discover hidden models (system identification and parameter estimation), solution techniques for inverse problems, construct low fidelity / surrogate / reduced order models, enable parametrized solutions, improve efficiency of numerical solvers, etc. In this project, we will understand the theoretical foundations and explore the practical application of PINNs for data-driven solutions and data-driven discoveries. The recommended projects should focus on the experience level and interests of the participants. (Contact: **Assoc. Prof. Allan P. Engsig-Karup, apek@dtu.dk**): *Select a topic and test one or more techniques on a problem(s) of interest*

    1. Data-driven solution. Find approximate numerical solutions of selected ordinary and partial differential equations using PINNs. Examine different network types such as FNN, ResNet, CNN, etc.
    2. Data-driven discovery. Estimate unknown parameters of selected ordinary and partial differential equations using PINNs.
    3. Neural Operators. Find approximate numerical solutions of selected ordinary and partial differential equations using emerging neural operator architectures such as DeepONet, Fourier Neural Operators, etc.
    4. Uncertainty Quantification for Neural Networks. Enable making predictions and quantifying uncertainty (fx via prediction intervals) in such predictions at the same time.
    5. Exploration / Application on a topic of interest. Apply the learned insights to, e.g., assess properties of PINNs through solving a new differential equation problem, evaluate pros/cons of advanced applications, etc.

References:

- c) https://arxiv.org/abs/2109.11313 (Borrel-Jensen, Engsig-Karup & Jeong, September, 2021)
- d) https://arxiv.org/abs/2308.05141 (Borrel-Jensen et al. 2023)
- e) https://arxiv.org/abs/1910.03193
- f) https://arxiv.org/abs/2105.09506
- g) https://www.preprints.org/manuscript/202006.0258/v1
- h) DeepXDE: https://arxiv.org/abs/1907.04502 , code: https://github.com/lululxvi/deepxde
- i) https://www.mdpi.com/2076-3417/10/17/5917/htm
- j) https://arxiv.org/abs/2003.10208 (Application, CFD)
- k) https://github.com/maziarraissi/PINNs
    - i) https://arxiv.org/abs/1711.10561
    - ii) https://arxiv.org/abs/1711.10566
    - iii) https://www.sciencedirect.com/science/article/pii/S0021999118307125

17) **Reduced Order Modelling Techniques for Digitial Twin applications**. We consider new data-driven algorithmic strategies that combines dimensionality reduction with multi-fidelity neural network surrogates for solving time-dependent problems efficiently and accurately. The data used for the training is generated using PDE-based simulators. The topic aims to be a small research project where the focus is on understand

the concepts behind new techniques and then apply them based on data from PDE based models that are also used for comparison. The PDE-based models are of interest, since they allow for descriptions of broad ranges of physical phenomena of engineering interest, however, often suffer from curse of dimensionality, i.e. the computational expense grows with the degrees of freedom in the numerical schemes. This can be circumvented through proper design of reduced order models with potential gains of up to three order of magnitude for a range of applications. This potential enable certain applications to become real-time, i.e. interactive with users. Multi-fidelity technique are proposed, however, other types of reduced order techniques based on data, e.g. POD-type techniques, may also be considered. Discuss with teacher what is the level of experience and interest. Contact: **Assoc. Prof. Allan P. Engsig-Karup, (**apek@dtu.dk): *Select a topic and test one or more techniques on a problem(s) of interest*
References:

l) Multi-fidelity surrogate modeling using long short-term memory networks, https://arxiv.org/abs/2208.03115
m) 2. Multi-fidelity reduced-order surrogate modeling, https://www.researchgate.net/publication/373579998_Multi-fidelity_reduced-order_surrogate_modeling

18) **Implementation of Conditional Flow Matching for data generation.** Flow Matching uses that a continuous normalizing flow can be defined using a vector field and that this vector field also defines a probability path. Conditional Flow Matching uses that we can construct conditional vector fields to approximate the probability path. This project is about defining the conditional vector fields such that the conditional probability paths are optimal transport paths between two distributions and training a model using the Conditional Flow Matching objective.
Recommended dataset: CIFAR-10.
Recommended reading: https://openreview.net/pdf?id=PqvMRDCJT9t, https://arxiv.org/pdf/2209.15571.pdf, https://arxiv.org/pdf/2209.03003.pdf Supervised by Beatrix M. G. Nielsen (bmgi@dtu.dk).

19) **Using Sentence BERT for information retrieval.** Sentence BERT (www.sbert.net) is a way of converting sentences or short pieces of text into fixed length vectors. This can be used to search for similar paragraphs or sentences among a number of records. This project is about implementing a sentence BERT model and using it for information retrieval.
Recommended dataset: Something from https://microsoft.github.io/msmarco/TREC-Deep-Learning-2019
Recommended reading: https://arxiv.org/pdf/1908.10084.pdf, https://www.sbert.net/examples/applications/information-retrieval/README.html, https://www.sbert.net/examples/applications/semantic-search/README.html
Supervised by Beatrix M. G. Nielsen (bmgi@dtu.dk).

20) **Exploring Explainability for Time Series Representations Learned through Self-Supervised Learning.** Self-supervised learning (SSL) has emerged as a potent technique for deriving meaningful representations from data in scenarios where labeled samples are scarce. However, the conventional evaluation of SSL models predominantly relies on downstream classification performance, lacking comprehensive methods for evaluating or interpreting the intrinsic learned representations. The Representation Learning Explainability (RELAX) method, as introduced in https://arxiv.org/abs/2112.10161, has recently demonstrated the potential to explain image representations.
Time series data, ubiquitous across numerous domains, remains less explored compared to images and natural language in the realm of deep learning applications. An intriguing example of such data is electroencephalography (EEG), representing electrical

measurements of brain activity, with applications ranging from diagnosing epilepsy to cognitive neuroscience. Additionally, time series encompass audio and speech representations, opening up a myriad of opportunities for advancing our understanding and using of these data modalities.
In this project, we aim to extend the RELAX method to time series representations, specifically targeting models trained through self-supervised learning on spectrograms of time series data. This project offers you the opportunity to choose the specific domain of interest within the time series realm, and together, we will define the scope and direction of the project. As a potential avenue, we propose using a pretrained model to assess the efficacy of the extended RELAX method.
The project is supervised by Thea Brüsch (theb@dtu.dk) and Teresa Scheidt (tksc@dtu.dk). Feel free to contact us if you have any questions.

21) **Image segmentation of car parts with Deloitte Consulting**. Supervised by Martin Closter Jespersen (majespersen@deloitte.dk), Mikkel Sikker Sørensen and Asbjørn Eller Skaarup. Link to project details here.

22) **Taylor Approximation of Neural Networks and their predictive posterior:** In this project, we will approximate the neural network with its n-th order taylor expansion using the method outlined in this paper: https://openreview.net/pdf?id=SkxEF3FNPH. We will then study how they transform normal distributions in the parameter space that approximate the posterior of the neural network(Obtained through variational inference or laplace approximation). Typically the pushforward of posterior distribution in the parameter space to the logit space is not tractable because of the highly nonlinear behavior of the neural network. However, by considering simple approximations of it we can obtain tractable distributions in the output space which we can then study to understand the behavior in the general case. Supervised by Hrittik Roy (hroy@dtu.dk).

23) **Understanding the Loss Landscape of Large Neural networks:** In this project, we will analyse the training dynamics of large neural networks. We will do so by analysing the eigenspectrum of loss function curvature similar to the analysis in this paper:

    https://browse.arxiv.org/pdf/1912.07145.pdf.

    Typical measures of curvature include the Fisher Information Matrix, Generalized Gauss-Newton approximation of the Hessian and the Hessian of the Neural Network. To obtain access to the eigenspectrum of these large(parameters x parameters) matrices we will implement Matrix-vector products with these matrices and then use standard matrix-free algorithms to compute the eigenspectrum, such as the Lanczos algorithm. Supervised by Hrittik Roy (hroy@dtu.dk).

24) **Change detection + HMR** (see also dtu.jobteaser.com)
    tl;dr. Detect changes in dynamic human poses over long periods of type.
    About the project: In this project, we will try to detect changes in human poses and shapes over long periods of time. It is possible to accurately model the human shape and pose using the SMPL model. This can be done automatically from a single image. In collaboration with Teton.ai, we have anonymized poses and shapes from patients at nursery homes over a long period of time. We are interested in detecting changes in their behavior or the way they walk as we expect this can be an indicator of increased risk of falling or need for special attention by the health care staff. In this project, you will develop an unsupervised clustering algorithm to detect changes in behavior. We will first create a large dataset of human poses over long periods of time with associated actions, e.g. getting up from bed or walking. We will then compare the human poses for each action

over long periods of time. We will build an automatic system to detect changes. Hopefully, this system can become a preventive healthcare tool that can help nurses pay special attention to patients with an increased risk of falling.
Relevant links: I recommend that you read the following papers:
*
SMPL: https://files.is.tue.mpg.de/black/papers/SMPL2015.pdf
* 4D humans: https://shubham-goel.github.io/4dhumans/
*
LART: https://people.eecs.berkeley.edu/~jathushan/LART/
Supervisor: Frederik Warburg <frewar1905@gmail.com>

25) **Diffusion + HMR** (see also dtu.jobteaser.com)
tl;dr. Use diffusion models to generate realistic human poses.
About the project: In this project, we will try to learn the underlying distribution of human poses. Once we have learned the distribution, we will generate new poses. We will parameterize a human pose with an SMPL model. We can use the 4D human model [see link below] to automatically generate training data. The diffusion model is trained by iteratively adding Gaussian noise to the SMPL parameters and denoising them. The project will be composed of the following steps: 1) generate training data using 4D human model, 2) train an unconditional diffusion model and generate a novel human pose, 3) train a conditional diffusion model that is conditioned on human actions, 3) train a dynamic diffusion model that generates a sequence of poses.
Relevant links: I recommend that you read the following papers:
* 4D humans: https://shubham-goel.github.io/4dhumans/
* DDIM: https://arxiv.org/abs/2010.02502
* BUDDI: https://muelea.github.io/buddi/
Supervisor: Frederik Warburg <frewar1905@gmail.com>

26) **NeRF + HMR (see also** dtu.jobteaser.com)
tl;dr. Reconstruct static 3D environments with dynamic 3D human poses.
About the project: In this project, we want to build a 3D training simulator using NeRFs or 3D Gaussian Splatting. In the project, we will first reconstruct a static hospital room scene. We have already collected the data for you! :) We will then create a dynamic reconstruction of a human that performs some actions, e.g. falls out of the bed. We can compose the static hospital scene and the dynamic fall scene into a new reconstruction. This will result in novel training data for downstream models. Lastly, we will explore SMPL models together with a static 3D reconstruction. The advantage of the SMPL models is that they allow for more control of the human subject, thus, we do not need to reenact a fall scene, but can rather just program the fall.

Relevant links: I recommend that you read the following papers:
* NeRF: https://www.matthewtancik.com/nerf
* 3DGS: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
* Dynamic 3DGS: https://dynamic3dgaussians.github.io/

Supervisor: Frederik Warburg <frewar1905@gmail.com>

27) **Project Title: Comparative Study of Deterministic and Rank-1 Bayesian Neural Networks**

In this project, you'll have the opportunity to work with advanced neural network architectures by implementing both a deterministic and a Bayesian version. Your work will be based on the research paper "Efficient and Scalable Bayesian Neural Nets with Rank-1 Factors" (Arxiv link). This paper introduces an innovative parameterization technique for Bayesian Neural Networks (BNNs) to improve their scalability and efficiency.

For the implementation, you can opt to replicate one of the architectures used in the paper, such as ResNet-50 or Wide ResNet 28-10. If you're feeling adventurous and have some experience, you could also choose to implement a Graph Neural Network (GNN) and train it on a public molecule dataset. The Bayesian version will use Variational Inference for training, and you'll be guided by a specific loss function as outlined in the paper.

The aim of this project is to give you a deep understanding of the trade-offs involved in using deterministic versus Bayesian approaches, particularly in the context of neural network efficiency and uncertainty quantification.

For any questions or additional information, please contact **Laurits Fredsgaard Larsen <laula@dtu.dk>**

28) **Automating the Segmentation of X-ray Images with Deep Neural Networks** In an era of rapid advancements in X-ray physics and the growing capabilities of X-ray synchrotron sources, the analysis of tomographic X-ray datasets has become increasingly critical in various scientific, medical, and industrial applications. However, once the raw data are collected, the manual segmentation of these images is a time-consuming and error-prone process, often plagued by uncertainties and subjectivity. Automating the segmentation process becomes imperative to keep pace with data acquisition rates and to ensure timely scientific discoveries and industrial insights. To address this challenge, in this project, we plan to leverage the power of deep neural networks to automate the segmentation of ptychographic X-ray images, removing the need for human intervention and significantly expediting the analysis process.

The primary objective is the development and training of a deep neural network, based on existing architectures, commonly used for other computer vision tasks (e.g. UNet, VGGnet, etc.). The training dataset consists of real-world X-ray images (raw and segmented), and the results will be benchmarked against manually labeled datasets. The project will be supervised by Salvatore De Angelis (sdea@dtu.dk) and Peter Stanley Jørgensen (psjq@dtu.dk).

More info can be found in the Project's GitHub repository:
https://github.com/sdea/AIStudentProjects/edit/main/README.md

29) **Prediction of protein isoforms using semi-supervised learning.** Through alternative splicing each gene in the human genome can produce multiple protein isoforms with distinct biological function. While gene-expression data is abundant, isoform expression is not. We have recently shown isoforms are understudied and present a huge opportunity in modern science. In many instances it is however not possible to re-quantify transcriptomics data with isoform resolution due to technical, practical and privacy issues. Therefore, we would like you to create a ML method which predicts isoform expression from gene expression. Specifically, you would use vast amounts of RNA-seq data as input to a VAE to extract informative gene-expression representations. Next you will train a DNN to predict isoform expression using these representations of a smaller training data as input. The project can, optionally, be extended using scGPT for gene

representation. The project will be co-supervised by Jes Frellsen (jefr@dtu.dk) and Kristoffer Vitting-Seerup (krivi@dtu.dk).

30) **Deep Learning-based Image Coding Enhancement.** Consider a scenario where we want to send an image from one device to another. We will encode the image on the transmitter, send the data through a noisy channel, and decode it on the receiver side. This encoding and decoding are traditionally done by mathematical-based algorithms. Starting a couple of years ago, researchers started using DNNs to do the task. A novel that got a lot of attention in this area is described in E. Bourtsoulatze, D. B. Kurka and D. Gunduz, Deep joint source-channel coding for wireless image transmission, IEEE Transactions on Cognitive Communications and Networking, 2019., where a simple autoencoder is used to do the encoding and decoding. The images that the autoencoder trained on are of size 32x32 pixels, and the performance drops down as we increase the size. One approach is to partition a larger image into blocks of 32x32 and send them one by one and at the end, put them back together. But the idea here is to identify which parts of the image contain more information and which parts just have a few (like a solid background part of an image). Thereafter, we can send the parts containing more information with more data and the parts having less information with less data to get a better overall performance in total. Supervised by Amin Hasanpour, moam@dtu.dk.

31) **Shrinking AlexNet - First Steps Toward TinyML and Efficient Deep Learning**. The goal is to familiarize students with the concepts used in TinyML and model optimization. Many devices around us are not as powerful as our PCs. As examples, we can name different microcontrollers, mobile phones, FPGAs, Raspberry Pi, etc. These devices are very limited in terms of memory, computation power, and in some cases energy. In this project, we are going to learn the techniques that can be used to make a model small. Please take a look at the lecture Dr. Song Han had at Stanford [2] to get a better understanding of this research field. We will start with 8-bit quantization on AlexNet, which is simply using 8-bit data instead of 32-bit data that we always use. Thereafter, the path is set by the students' demands. We can investigate other types of quantization, pruning, clustering, checking the tools that are available for this task (famously TensorFlow Lite Micro), implementing the resulting network on a tiny ARM-based microcontroller, or even go further by training the network on the end-device! Illustrations of experiments. Supervised by Amin Hasanpour, moam@dtu.dk.

32)