02456 Deep learning 2021 - course plan and information

Note: Not completely updated for 2021 yet.

Teaching has to be fun and safe for everyone so we have the following 2021 set up:

Covid-19: Follow DTU's general guidelines. Especially 1. and 2. are important. Unfortunately, the delta variant is a tough one so we have a high risk of seeing more infections during fall. We want to be able to stay in class so the general guidelines will be strictly followed in this class. For those of you who cannot be in class we have supervision through Zoom. See below.

Time: Monday 13-17, first session is August 30th.

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

B303A-HOEST
B303A-A042
B303A-A048
B303A-HVEST

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. The 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. A selected number of locations will have a computer with a Zoom meeting open.

We also use Slack for communication. Each location will have its own Slack channel where we also distribute the Zoom links. We will also make channels for other things (labs, software, AWS 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 brief presentations of project proposals from external supervisors in week 6.

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>`l. 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. Recurrent neural networks (RNN) + presentation of student projects internal supervisors.
6. Tricks of the trade and data science with PyTorch + presentation of student projects external supervisors.
7. Variational learning and generative adversarial networks for unsupervised and semi-supervised learning + deadline for selection of student projects.
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.
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. (Note that weeks refer to weeks in term. 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 synopsis. Deadline: Monday week 10. 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.
10. 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. Exam date is tentatively set to December 8th from 9 to 17 in SkyLab. We divide the day into half hour slots and your group will later be given the possibility to register for a slot. 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 minutes 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 don't have to use that. A limited number of you can on a first come first served basis get your poster printed at DTU Compute for free by sending your pdf to print@compute.dtu.dk with “02456” in the title. The DTU library also offers poster printing for a not too high price.
11. Final report deadline January 4rd 2021 23.59. 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.

In 2018 we moved to the deep learning framework PyTorch. Previously we used Tensorflow. The main reason for our decision to move to PyTorch is that it is easier to learn and develop code in. You can find a discussion of the two frameworks here. If you are interested in seeing how the exercises look in TensorFlow here is a link to the exercises 2017 edition. Note that for peer grading we will only accept hand-ins in PyTorch.

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. 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. These books 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 goto www.peergrade.io/join and type in course code FXCWTM or just click here. 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. That is possible but might create some confusion for you.

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 - 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.
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 Friday Oct 15th 23.59. Link to 2021 project selection sheet here.

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 2021

Not complete yet.
To get more information about select projects, you can contact supervisors.

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 and others.

2) **Unsupervised (or Active) Learning for Image Analysis:** Accurately assessing food intake is a key enable for nutritional studies in both research and clinical. The central theme is characterizing food intake though smartphone images (and metadata) and using it for predicting clinical relevant outcomes, such as overweight and glucose response. There are ~16,000 images taken by the COPSAC2000 cohort (~400 young adults) over a two-week period as part of a deep metabolism characterization. Possible project directions include,
    a) Active or online learning to for hierarchical labeling (w. feedback/use by clinical experts)
    b) Transfer learning, using existing DL image models (possibly food related)
    c) Reverse recipe generation (utilizing datasets or models from the literature)
    d) Utilizing meta-information (camera make, shutter speed, aperture) and items from the image background to estimate volume or meal composition.

    Projects will be supervised by Morten Arendt Rasmussen, KU FOOD & COPSAC (mortenr@food.ku.dk, morten.arendt@dbac.dk) and Jesper Hinrich, KU Food (jlh@food.ku.dk). For additional details, see link.
3) **Deep Reinforcement Learning**. Supervised by Peter Ebert Christensen, pebch@dtu.dk, Anders Christensen and Raul Ortega. The full project descriptions are available here (changes may occur).
    a) **Generalization in Video Games:** Teaching machines to play video games better than yourself, directly from screen pixels. We will investigate how machines react and adapt to increasingly difficult situations
    b) **Sample Efficiency in Continuous Control:** Machines tend to struggle when applied to real world systems, in which the action space is continuous. We will investigate how machines are capable of manipulating objects in 3D space.
    c) **Exploration:** In RL we often want our agents to find the best possible solution using either a simulator or utilize a readily available dataset to. Unfortunately if we commit to a solution too quickly can lead to unoptimized agents due to lack of exploration. We will investigate the effect of exploration strategies for sequential decision making.

4) **Unsupervised representation learning**. Supervised by Bo Li, blia@dtu.dk and Tommy S Alstrøm, tsal@dtu.dk. We will work to extract useful representations in an unsupervised way. Motivation can be found in the slide. Relevant papers:
    - SimCLR
    - SimSiam

    Coding skills are required. Datasets: images, time series and spectroscopy data.

5) **Audio-based out-of-distribution detection with Corti.ai**. Supervised by Jakob Havtorn, jdh@corti.ai. More information here.

6) **Image segmentation of car parts with Deloitte Consulting**. Supervised by Martin Closter Jespersen (majespersen@deloitte.dk), William Frisch Møller and Asjbørn Eller Skaarup. Link to project details.
7) **Deep Active Learning.** Supervised by Frederik Boe Hüttel (fbohy@dtu.dk) and Christoffer Riis (chrrii@dtu.dk), MLSM DTU. We want to use Bayesian neural networks (BNN) for active learning. The Bayesian framework gives uncertainty to the weights of the network, which tells us something about the model's uncertainty. The uncertainty can be exploited in active learning, to identify which data points the model is uncertain of, and use this to query labels for the data points where the model is uncertain of. However, BNN still suffers from some practical and theoretical limitations we wish to explore. We therefore have the following sub-projects proposed.

    a) **Bayesian active learning** for either spatio-temporal or structured data. (https://arxiv.org/pdf/1703.02910.pdf)
    b) **Investigate posterior collapse** in BNN using the spectral smoothing as an alternative to reparametrization. (http://proceedings.mlr.press/v139/pervez21a.html)
    c) **Deep Gaussian Processes**: We will apply an alternative to BNNs. Gaussian Processes (GPs) tend to be better at quantifying uncertainties than BNNs. On the other hand, GPs also tend to be too simple to solve real world problems, and thus we want to investigate the extensions Deep Gaussian Processes (DeepGPs), Deep Sigma Point Processes (DSPP), and GPs with Deep Kernel Learning (GPDKL). All three models have good estimation of uncertainty and provide similar flexibility as the BNNs. This project can both be on static data or in combination with active learning. The type of data is either structured or spatio-temporal.

    This project will be in the area between applied and theoretical deep learning.

8) **Deep Neural Architecture Search.** Supervised by Christoffer Riis (chrrii@dtu.dk), MLSM DTU. We will apply deep neural architecture search to speed up slow simulators. The idea is motivated by this paper**,** and we would like to test a simplified version (FNN) on a structured data set.

9) **Object detection of helmets and smartphone use.** Supervised by Christoffer Riis (chrrii@dtu.dk) and Frederik Boe Hüttel (fbohy@dtu.dk), MLSM DTU. In this project, we will use computer vision to detect safety related behaviors of bicyclists in traffic. Using an available dataset of video frames collected in the Copenhagen traffic, the task will be to annotate active cyclists in the frames and train a computer vision algorithm (e.g. RetinaNet) to reliably detect behaviors such as helmet or smartphone use.

10) **Autoencoders for image quality improvement with Omhu.** Supervised by Raluca Jalaboi (raluca@omhu.com). More information here.

11) **Guided attention for dermatological diagnosis systems with Omhu.** Supervised by Raluca Jalaboi (raluca@omhu.com). More information here.

12) **Diagnosis correction using domain knowledge with Omhu.** Supervised by Raluca Jalaboi (raluca@omhu.com). More information here.

13) **Danish speech synthesis.** Supervised by Lars Kai Hansen, lkai@dtu.dk and Martin Carsten Nielsen, martin@danspeech.io. Based upon Nvidia's Flowtron framework for multi-speaker synthesis learning we will train a Danish speech synthesis model. More information: https://arxiv.org/pdf/2005.05957.pdf, https://github.com/NVIDIA/flowtron
14) **Generating coherent text from noisy speech transcripts.** Supervised by Lars Kai Hansen lkai@dtu.dk and Martin Carsten Nielsen, martin@danspeech.io. We will attempt to build a computationally inexpensive network that takes noisy transcripts from an automatic speech recognition model and generates coherent, denoised text.
15) **Designing self-driving earbuds with augmentedhearing.io** which enhance voices based on function correlated with speech intelligibility - as one in four adults struggle to understand speech in challenging acoustics we aim to train consumer earbuds to enhance voices through back propagation using DHASP model implemented using PyTorch differentiation package validated on TIMIT speech dataset based on an objective function correlated with HASPI speech intelligibility auditory processing model available in Matlab. Supervised by Michael Kai Petersen, michaelkaipetersen@gmail.com.
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. Supervised by Allan Peter Engsig-Karup, apek@dtu.dk For more detail see: Project description PINNs and GNNs for simulation .
17) **Graph-Neural Networks for Model-based (physics-informed) simulation:** Consider a machine learning framework and model implementation techniques that can learn to simulate a wide variety of challenging physical domains, involving fluids, rigid solids, and deformable materials interacting with one another. The state of a physical system, expressed as nodes in a graph, computes dynamics via learned message-passing. Supervised by Allan Peter Engsig-Karup, apek@dtu.dk For more detail see: Project description PINNs and GNNs for simulation .
18) **Bioinformatics projects** A number of projects are available supervised by Paolo Marcatili, pamar@dtu.dk (projects b)-d)), Ole Winther, olwi@dtu.dk (projects a) and g)) and Felix Pacheco Pastor, felix.pastor@cpr.ku.dk (projects d) and e)). Project list:
    a) **Bioinformatics students come with your own project.**
    b) **Prediction of intrinsically disordered proteins by language models (dataset available)**
    c) **Prediction of crystal contacts from protein crystallographic data by graph neural networks (GNNs)**

d) **Prediction of TCR-pMHC interactions using molecular modeling and recurrent networks**
e) **Dimensionality reduction of genetic ancestry data with Variational autoencoders.** (Unsupervised Learning)
f) **Classification of sample historical era using genotype array data.** (Supervised Learning)
g) **Working with AlphaFold2 (AF2)** AF2 has been made available and a lot of researchers are reporting results also on repurposing it to other tasks like complex formation. In this project we will get acquainted with AF2 and try it on a few tasks such as prediction of protein DNA binding sites.

A detailed description of subprojects b)-d) can be found here.

19) **Deep learning for search/open domain question answering.** Deep learning works well for search as illustrated by its success in the MS Marco benchmark. In this project we will investigate dense passage retrieval methods using pre-trained transformers, with a similarity loss fine-tuned on labeled datasets. The transformers and datasets loaders can be obtained from Huggingface.co. Supervised by Ole Winther, olwi@dtu.dk.
20) **Real-time object detection and tracking.** Real world computer vision applications rely on, not just the AI algorithm, but also hardware and data pipelines. For instance, AI for autonomous driving is no good if it runs at 2 fps.
In this project, you will focus on the AI algorithms but the end goal is to run an object detection algorithm on a Jetson nano, and visualise the monitoring in grafana.
Supervised by Peter Jensen, peter.jensen@cellari.io.
21) **Colorization of old images and videos.** Supervised by Santiago Gutiérrez, s200140@student.dtu.dk. More details can be found here.
22) **Wind power prediction.** Supervised by Tomas Perez Alvarez, s183307@student.dtu.dk. More details can be found here.
23) **Neural ODEs for latent variable generative models.** Work with the neural ODE method to build hierarchical latent variable models. Supervised by Ole Winther.
24) **Learning representations for multivariate time series data with Autoencoders.** You will work on developing autoencoder based models for learning representations from car sensor data collected using the Green Mobility car fleet, within the LiRA project (more information here). Supervised by Milena Bajic, mibaj@dtu.dk.
25) **Anomaly detection in time series data using autoencoders.** You will work within the LiRA project on developing autoencoder-based models for anomaly detection in sequences of car sensor data, within the LiRA project (more information here). Supervised by Milena Bajic, mibaj@dtu.dk.

Extracurricular projects

1. CEUS.

Project list 2020

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 and others.
2) **Deep Reinforcement Learning.** Supervised by Nicklas Hansen and Mariana Monteiro. We will teach machines how to walk, play video games, and more.
   For a full description of projects, follow this link.
   a) **Generalization:** Teaching machines to play video games better than yourself, directly from screen pixels. We will investigate how machines react and adapt to increasingly difficult situations.
   b) **Sample efficiency:** Machines are notoriously slow learners. We will teach machines how to walk, balance, and manipulate objects by investigating recent methods for more efficient learning.
   c) **Offline RL:** In sequential decision making, machines usually learn by interaction, but that can be expensive or outright unsafe (e.g. self-driving cars). We will explore methods for learning without any interaction, i.e. *offline* RL.
   d) **Self-supervised keypoint learning:** Learning from low-complexity features is easier, but they can be difficult to obtain without human supervision. We will experiment with *self-supervised* learning of keypoints to automatically distill complex scenes into simple and interpretable features for RL agents.
3) **Learned Data augmentation.** Supervised by Oldouz Majidi. Augment the dataset and create new data by applying a statistical model of the transformations found within a given class. The presentation Video. More information here.
4) **Project with Hedia**. Supervised by Mads Obdrup Jakobsen, mads@hedia.co. *For a full description of projects follow this link.*
   a) Blood Glucose Forecasting based on CGM data
   b) Recognising food in photos.
5) **Image segmentation & secure synthetic data generation with Deloitte Consulting**. Supervised by Jacob Bock Axelsen (jaxelsen@deloitte.dk), Martin Closter Jespersen and William Frisch Møller. More information here.
   a) Segmenting car parts using internal dataset
   b) Synthetic data generation as an anonymization technique for data sharing
   c) Deep reinforcement learning or genetic algorithms for self-driving car

6) **Multi-label text classification for airplane inventory parts with beepanalytics.com**. Supervised by Torben Jensen, BeepAnalytics, tbj@beepanalytics.com. More information here and video presentation here.
7) **Physics Informed Neural Networks (PINNs):** Numerical simulation is of major importance in industrial design processes. Supervised by Allan Peter Engsig-Karup, DTU Compute, apek@dtu.dk More information here and presentation slides here.
8) **Projects with Corti.ai**. Text-based anomaly detection and Self-supervised representation learning on audio and transcribed text. Supervised by Lars Maaløe, lm@corti.ai and other Corti employees. More information here and video here.
9) **Real-time object detection and tracking.** Real world computer vision applications rely on, not just the AI algorithm, but also hardware and data pipelines. For instance, AI for autonomous driving is no good if it runs at 2 fps.
   In this project, you will focus on the AI algorithms but the end goal is to run an object detection algorithm on a Jetson nano, and visualise the monitoring in grafana.
   Supervised by Peter Jensen, peter.jensen@cellari.io.
10) **Multidimensional Anomaly Detection using Deep Learning.** Woodsense.dk works with predictive maintenance for wooden constructions in buildings. Supervised by Lasse Regin, lasse@woodsense.dk. More information here and video here.
    a) Anomaly detection in multivariate time series using e.g. LSTM- or Transformer Autoencoders.
    b) Generating realistic synthetic multidimensional time series data using deep learning.
    c) Object detection - detecting dormers in roof images.
11) **Speech processing with Augmented Hearing**. The project is presented in this video with slides. Projects supervised by Michael Kai Petersen, michaelkaipetersen@gmail.com. Project details: Work with pre-trained word2vec 2.0, finetune encoders to specific voices and environments, combine performance with standard iOS/android speech to text audio transcription and calculate eller phoneme error rates (PER).
12) **Bioinformatics.** Projects supervised by Ole Winther, Paolo Marcatili, pamar@dtu.dk and others. Students interested in bioinformatics can contact Ole.
13) **Deep learning for modelling urban mobility data** (Machine Learning for Smart Mobility) supervised by Filipe Rodrigues, rodr@dtu.dk and colleagues. Video here.
    a) Conditional density estimation of spatio-temporal distributions of car sharing demand (e.g. SHARENOW and Green Mobility). We use Mixture Density Networks or Normalizing Flows potentially in combination with Deep Kalman filters or Deep Gaussian processes.
    b) NeurIPS traffic4cast challenge - see: https://www.iarai.ac.at/traffic4cast/.
       Study traffic movies of entire cities. Predict future traffic flows by high-resolution forecasts. Discover rules and patterns, new insights and understanding.
    c) Smartphone Data Fusion and Deep Learning for Travel stop and mode detection. To perform classification we use GPS trajectories generated by smartphones, with data from Geographical Information Systems.

        - Multi-task classification of trip mode and purpose, using smartphone GPS data fused on OpenStreetMap
        - Semi-supervised classification of transportation mode, using smartphone GPS data fused on OpenStreetMap
        - Self-supervised learning of trip mode, using pseudo-labels extracted from smartphone-sensors fused with IoT (BLE beacons and autonomous vehicles)
    d) Spatial Neural Attention Models for mobility demand prediction. Attention mechanisms have become an integral part of sequence modeling in various tasks, allowing models to "attend to" relevant parts of the input sequence without regard to their temporal distance. In this project, we extend these concepts to deal with spatio-temporal data sources such as mobility demand (e.g. ShareNow's demand is located both in time - day&hour - and space - latitude&longitude). Information from neighboring areas could help improve estimates of future demand across different geographies, thus allowing service providers to make decisions coherently with user behavior and needs.
    e) Crowdedness prediction in public transport under Covid-19 using Smartcard demand-data (Rejsekort-data). In order to avoid full busses and trains, Movia, the public transport authority of Eastern Denmark, has a need for predicting how many passengers will be in vehicles and use this to adjust capacities (add/move vehicles) and target information under the Covid-19 pandemic. With the use of Rejsekort data (tap-ins / tap-outs) you will predict the passenger demand of tomorrow, so social distancing can be kept in the public transport.

    For additional project ideas and datasets, check out our MSc thesis catalog**.**

14) **Probabilistic Deep Generative Models**. (supervised by *Didrik Nielsen* - didni@dtu.dk, *Giorgio Giannone* - gigi@dtu.dk, *Dimitris Kalatzis* - dika@dtu.dk). We will use Deep Unsupervised Learning to find patterns in lots of unlabeled data such as images, audio, text, etc. The 3 main classes of methods we will consider are:
    a) Variational autoencoders (VAE, LadderVAE).
    b) Normalizing flows (GLOW, RealNVP).
    c) Autoregressive models (PixelCNN, WaveNet).

    After choosing one of these methods, we will design a project tailored on your interests and skills. Possible applications are Density Estimation, Image Generation, Semi-supervised learning, Language Modelling.
    - Requirements: Basics in Generative Models.
    - Slack Channel: **#project14-probabilistic-deep-generative-models**

15) **Few-shot learning**. (supervised by *Giorgio Giannone* - gigi@dtu.dk). Few-shot learning aims to mimic human ability to learn and adapt to new tasks from a handful of examples. In this project we will explore classical few-shot learning methods for supervised and unsupervised learning, understanding how the problem is formulated and focusing on approaches based on latent variable models. Relevant papers:

a) Towards a Neural Statistician - VAE-based approach that encodes datasets (instead of samples).
b) Neural Processes - VAE-based approach that encodes supervised datasets.
c) One-shot generalization in DGMs - conditional VAE+LSTM for fast inference.
d) Metric-based - MatchingNet, ProtoNet, RelationNet.
e) Meta-Learning - MAML, REPTILE.

- Requirements: Good coding skills. Basics in VAEs.
- Slack channel: **#project15-few-shot-learning**

16) **Discrete VAEs: studying the bias/variance tradeoff in gradient estimation** *(supervised by Valentin Liévin, Andrea Dittadi)* Continuous VAEs rely on the reparameterization trick for the estimation of the gradient of the inference network. For categorical latent variables (i.e. VQ-VAE), the reparameterization trick is not directly available. Two main options exist:
    - Relaxation-based methods (Gumbel-Softmax): allows for reparameterization at the cost of biasing the gradients
    - Score-function estimators (VIMCO, OVIS): unbiased gradients, the higher the computational cost

    Based on the existing implementation, design a simple experimental framework to compare the two methods. The focus will be set on quantifying the bias/variance tradeoff of the gradients in simple examples. This workshop paper will be used as a starting point (OVIS outperforms VIMCO). This work can potentially lead to a small publication (workshop).
    Requirements: Good background in Mathematics, basics in VAEs
17) **Image Super-Resolution using Hierarchical VAEs** *(supervised by Valentin Liévin, Andrea Dittadi)* Image Super-Resolution using deep convolutional networks showed one could generate high-definition images from low-definition: a science-fiction myth becoming reality. Generative models have since been leveraged to improve the perceptual quality of the generated images.
    Yet, GANs suffer from many downfalls such as mode collapse (ignoring part of the dataset) and probabilistic methods such as deep VAEs (LVAE, PixelCNN, BIVA, VQ-VAE-2, NVAE) showed impressive improvements in perceptual quality, without suffering the downsides of the GANs.
    The architecture of Deep VAE is highly compatible with Super-Resolution (the top latent variable is a feature map of lower dimension). In this project, you will
    - Briefly review the existing literature
    - Implement your own deep VAE or use the state-of-the-art pretrained NVAE model and fine-tune it for Super-Resolution.
    - Study the model and suggest improvements

    You will use either the official BIVA GitHub or the NVAE GitHub.
    Requirements: good coding skills, good development practices, a good understanding of VAEs

18) **Object-centric VAEs** *(supervised by Valentin Liévin, Andrea Dittadi)* Learning object-centric representations of complex scenes is a promising step towards enabling efficient abstract reasoning from low-level perceptual features. Yet, most deep learning approaches learn distributed representations that do not capture the compositional properties of natural scenes. Recently, object-centric unsupervised learning methods have attempted to address these limitations. Most of these are based on the VAE framework.
This project consists in:
    - briefly reviewing the literature in object-centric VAEs;
    - studying more in depth one or two papers (e.g. MONet, SPACE, Slot Attention);
    - reproducing some results from these papers and comparing them on the same benchmarks (PyTorch implementations are available for some models);
    - implementing a basic VAE (not object-centric) and comparing its performance with the more advanced models.

    Possible extensions:
    - Most of these models have discrete random variables and could benefit from better gradient estimators and tighter variational bounds. OVIS could help here.
    - Come up with minor improvements to existing models.
    - Apply these models to different benchmarks/datasets.

    Requirements: good coding skills, good development practices, a good understanding of VAEs.
19) **State-of-the-Art Language Modelling** *(supervised by Valentin Liévin, Ole Winther)* Transformers are now ubiquitous in natural language processing (GPT). Challenge the existing benchmarks in language modelling using your own implementation. How far can you go without using a billion parameters? Check the text8 benchmark
20) **State-of-the-art Neural Translation** *(supervised by Valentin Liévin, Ole Winther)* Transformers are now ubiquitous in natural language processing (GPT). Challenge the existing benchmarks machine translation using your own implementation. How far can you go without using a billion parameters
21) **Open-Domain Q&A** *(supervised by Valentin Liévin, Ole Winther)* Open-domain Question Answering aims at answering factual questions using a large collection of documents (e.g. the entire Wikipedia). A common strategy consists in using two specialized components:
    - A *knowledge retriever*: retrieve the K most relevant documents given the query
    - A machine *reader:* that reads the selected documents to answer the query

    Dense Passage Retrieval (DPR) aims at learning document representations for effective *retrieval* using learned representations (BERT) instead of classic representations such as TF-IDF.
    This project aims at reimplementing DPR based on available code and studying the method on your own dataset or on a subset of the original dataset. This project focuses on (1) analyzing and understanding the method and (2) implementation. (3) You will attempt to improve the existing method or demonstrate the effectiveness of the method based on your own experimental setup.

Requirements: basics in NLP, good coding skills, development methodology

22) **Exploring the creative use of Deep Generative Models for live musical performance** *(supervised by Valentin Liévin)* Modern music production relies on two formats: audio and MIDI. Modelling music in the audio domain is fairly complex and computationally expensive. MIDI data is a symbolic representation of the music notes, pretty much like a music sheet. Hence all methods available for natural language modelling (Transformers or VAEs) can be easily applied. For this project, you are free to use your own data. However, getting high-quality MIDI data remains an open issue. As a starting point, we will use the MAESTRO piano performance dataset.
    There is no doubt one can generate music using Transformers, but for what use? Integrating deep learning methods in a music production setting remains an open issue: how can machine learning techniques bring value to the creative process?
    This project will aim at controlling music generation based on the performer's inputs (machine learning jam). For instance,
    - Training a machine learning (LSTM/Transformer) model to play the left hand (lower octaves) given the user's right hand (higher octaves). This is similar to this project, except that we aim at generating keys, not the drums.
    - Conditioning the music generation on a particular musical style

    You can check the Google Magenta for examples of ML usage in creative applications.
    Requirements: motivation and/or playing piano, good programming skills.
23) **Graph pooling and unpooling in Graph Neural Network with its applications in graph classification, node clustering and image segmentation.** Supervised by Aasa Feragen, afhar@dtu.dk and Guan Wang, guawa@dtu.dk. Detailed project description here.
24) **Language Modeling for Protein Generation**. Controllable protein generation with an advanced RNN where the "language" are the aminoacid sequences. Keywords: Transformer architecture, RNNs. Paper: ProGen: Language Modeling for Protein Generation https://arxiv.org/pdf/2004.03497.pdf. Supervisor Yevgen Zainchkovskyy, yeza@dtu.dk.
25) **Spot Nuclei. Speed Cures.** You will create a computer model that can identify a range of nuclei (across varied conditions.Identifying the cells' nuclei is the starting point for most analyses. Identifying nuclei allows researchers to identify each individual cell in a sample, and by measuring how cells react to various treatments, the researcher can understand the underlying biological processes at work. Keywords: Computer vision, segmentation, Masking, Region based CNN, CNN. Paper: https://www.nature.com/articles/s41592-019-0612-7
    Kaggle: https://www.kaggle.com/c/data-science-bowl-2018/. Supervisor Yevgen Zainchkovskyy, yeza@dtu.dk.
26) **Algorithmic Fairness**. Supervised by Pola Schwöbel and Cilie Feldager. More information here.
27) **Danish speech synthesis.** Supervised by Lars Kai Hansen, lkai@dtu.dk and Martin Carsten Nielsen, martin@danspeech.io. Based upon Nvidia's Flowtron framework for

multi-speaker synthesis learning we will train a danish speech synthesis model. More information: https://arxiv.org/pdf/2005.05957.pdf, https://github.com/NVIDIA/flowtron

28) **Generating coherent text from noisy speech transcripts.** Supervised by Lars Kai Hansen lkai@dtu.dk and Martin Carsten Nielsen, martin@danspeech.io. We will attempt to build a computationally cheap network that takes noisy transcripts from an automatic speech recognition model and generates coherent, denoised text. As the project doesn't rely on any implemented framework, some fluency in tensorflow or pytorch is beneficial as we will be building the models from scratch.

Project plan

To appear.

Deep learning student jobs

To appear.

Deep learning student PhD positions and internships

2021: To appear.

2020:

1. https://eee.epfl.ch/
2. Oticon

Deep learning potential master projects

2020 edition:

1. Demant.
2. Worms Safety AI
3. DL for chemistry

2019 edition and beyond;

1. Coloplast project description here.
2. Usertribe project description here.
3. Criterion.ai project descriptions here, here and here.

4. Projects on machine learning for material science jointly supervised by DTU Energy and DTU Compute here.
5. Raffle.ai. Please contact Ole ow@raffle.ai.

Deep learning PhD scholarship

1. Chalmers PhD1 and PhD2.