Final Project

This sheet has the missing formalities for the final project and has a list of possible publications that you can work on.

The idea of the project is to show that you can **read** and **understand** a **deep learning research paper**, **extract the important parts** and **reimplement the model/experiments.**

Submission Format: You are supposed to submit **the code** (via a GitHub repository) and **a written report/tutorial** on your project (as a pdf via LaTex). The report should explain the important parts of the paper that you worked on and also explain the corresponding important parts of the implementation (I recommend to do that in turns, e.g. first explain how the loss is computed, then give the code that implements this loss). Please be as concise as possible. In the best case I don't have to look at the actual code (although I might) and understand what you did and how you did it from your report. If you have any questions about the format don't hesitate to contact me.

Project Topic: As already mentioned I recommend to choose an existing paper and focus on understanding and reimplementing. If you want you can also design your own project or choose a Kaggle competition that you want to solve. But in both cases I see the problem that you have to focus on understanding how to realize your goal and then actually achieve it, which might be more complex than you think. In the following you find a list of possible publications that you could work on. They are only an inspiration, you are also free to choose other publications.

In any case please write me an E-Mail, explaining which publication/project you would like to implement and also give some detailed info on what exactly you plan to do. Deadline: 14.02.2020!

Visual Object Recognition
Multiple Object Recognition with Visual Attention (https://arxiv.org/pdf/1412.7755.pdf)
Dynamic Routing between Capsules (https://arxiv.org/abs/1710.09829)
Matrix Capsules with EM-Routing (https://openreview.net/forum?id=HJWLfGWRb)
Stacked Capsule Autoencoders (https://arxiv.org/pdf/1906.06818.pdf)
Adversarial Examples Are Not Bugs, They Are Features (https://arxiv.org/abs/1905.02175)
Towards Deep Compositional Networks (https://arxiv.org/abs/1609.03795)
Attend, Infer, Repeat (https://arxiv.org/abs/1603.08575)

Visual & Creative
Colorful Image Colorization (https://arxiv.org/abs/1603.08511)
A Neural Algorithm of Artistic Style (https://arxiv.org/abs/1508.06576)
Compositional Pattern Producing Networks (https://distill.pub/2018/differentiable-parameterizations/)

Technical
Deep Complex Networks (https://arxiv.org/pdf/1705.09792.pdf)
Training and Inference with Integers in Deep Neural Networks (https://arxiv.org/abs/1802.04680)
Spherical CNNs (https://arxiv.org/pdf/1801.10130.pdf)
The Lottery Ticket Hypothesis (https://arxiv.org/abs/1803.03635)
Weight Agnostic Neural Networks (https://arxiv.org/abs/1906.04358)
The wake-sleep Algorithm for Unsupervised Neural Networks (https://www.cs.toronto.edu/~hinton/csc2535/readings/ws.pdf)

Generative Models
StyleGAN (https://arxiv.org/pdf/1812.04948.pdf)
CycleGAN (https://arxiv.org/abs/1703.10593)
Generative Face Completion (https://arxiv.org/pdf/1704.05838.pdf)
Glow: Generative Flow with Invertible 1x1 Convolutions (https://arxiv.org/abs/1807.03039)
BEGAN (https://arxiv.org/abs/1703.10717)

Progressive Growing GANs (https://arxiv.org/abs/1710.10196)
Pix2Pix (https://arxiv.org/abs/1611.07004)
beta-VAE (https://openreview.net/forum?id=Sy2fzU9gl)

Continual Learning and Out-of-Distribution Detection
iCaRL: Incremental Classifier and Representation Learning (https://arxiv.org/abs/1611.07725)
Overcoming Catastrophic Forgetting in Neural Networks (https://arxiv.org/abs/1612.00796)
Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples (https://arxiv.org/abs/1711.09325)

One-Shot Learning
Prototypical Networks for Few-Shot Learning (https://arxiv.org/pdf/1703.05175.pdf)
Siamese Neural Networks for One-Shot Image Recognition (https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf)
Low-Shot Learning with Imprinted Weights (https://arxiv.org/abs/1712.07136)
Matching Networks for One-Shot Learning (https://arxiv.org/abs/1606.04080)

Reinforcement Learning: Ask Leon Schmid.

Natural Language Processing
Keep in mind that most NLP approaches are computationally expensive. In the end I don't expect you to achieve awesome results. If one can see that the network is training this will be sufficient.
Seq2seq learning (https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf)
Transformer (quite a lot): https://arxiv.org/abs/1706.03762
Convolutional Seq2Seq https://arxiv.org/abs/1705.03122