DEEP LEARNING – WS 2019/20

LECTURE SCHEDULE

Date	Topics		Homework
25.11.	Math & ML Basics	Review of math basics ▪ linear algebra ▪ Probability ▪ information theory	
28.11.		Review of machine learning basics ▪ Supervised, unsupervised, reinforcement learning ▪ Empirical risk minimization ▪ Overfitting (05_ml:2-7) ▪ No free lunch theorem	Intro to Jupyter + tech stack Simple linear/logistic regression example in numpy ▪ (Batch) gradient descent manually, not AutoDiff ▪ Separation of training, validation & test sets ▪ Overfitting ▪ (L2) regularization **Due date: 16.12.**
16.12.	Perceptron and multi-layer perceptron	▪ Definition of perceptron and MLP ▪ Why deep? (01_intro:2-4, 06_mlp:17-20) ▪ Training deep neural networks ○ Optimization -> gradient descent (4.3, 04_numerical) ▪ Backpropagation (06_mlp) ▪ Stochastic gradient descent (5.9, 8.4+8.5) ▪ Vanilla SGD ▪ Momentum ▪ RMSProp ▪ Adam ○ Regularization ▪ Early stopping ▪ L1/L2 ▪ Dropout ○ Hyperparameter optimization: grid vs. random search	
19.12.			▪ MLP on relatively small nonlinear dataset ▪ Using PyTorch ▪ Explore different regularizers via random search ○ Early stopping ○ L2 ○ Dropout **Due date: 6.1.**

6.1.	Convolutional neural networks (CNN)	▪ Motivation and definition (9) ○ weight sharing ○ sparse connectivity ○ growing receptive fields ○ pooling for invariance ○ striding, dilation, deconvolution ○ padding ▪ Architectures of classic and modern image processing CNNs: LeNet, AlexNet, VGG, ResNet, Inception, DenseNet, NAS ▪ More on optimization ○ Initialization ○ Vanishing and exploding gradient problem ○ Batch normalization as a simple fix for CNNs ▪ Timeseries (1D) and video (3D)	▪ CNN on (fashion) MNIST, CIFAR-10 or similar ▪ Explore architecture choices like number of features, kernel sizes, pooling, striding, padding ▪ Batch normalization **Due date: 13.1.**
9.1.			
13.1.	Transfer Learning & perceptual loss functions	▪ Pre-training on ImageNet ○ DeCaf paper ○ Object detection, segmentation ○ Saliency ▪ Deep features as perceptual loss functions ○ Texture synthesis ○ Style transfer	Fine-tune or train classification layer of pre-trained ResNet on an image classification task with relatively little data available **Due date: 20.1.**
16.1.	Deep Image Synthesis	▪ Deep generative image models ○ Autoencoders ○ (Convolutional) Autoencoder ○ VAE + beta VAE + spatial broadcast VAE ▪ Autoregressive models ○ Mixture models ○ Spatial LSTM ○ PixelCNN ▪ Generative adversarial networks (GAN)	
20.1.		▪ Image-to-image models ○ Domain translation networks ○ Style transfer ○ Superresolution	▪ VAE on MNIST ▪ Visualize classes in latent space ▪ Visualize latent traversals **Due date: 27.1.**

23.1.	Recurrent neural networks (RNN) and text processing	▪ Basics/theory ○ Unfolding computational graph (10.1) ○ Simple RNNs (10.2) ○ Bidirectional RNNs (10.3) ○ Sequence-to-sequence models (10.4) ○ Deep RNNs (10.5) ▪ Recurrent architectures used in practice ○ Gated recurrent unit (GRU) ○ Long-short-term memory module (LSTM) ○ Neural Turing Machine ○ Seq2Seq: translation example ▪ Deep word and text embeddings ○ Word2Vec ○ Transformer ○ ELMo + BERT + GPT-2	
27.1.			▪ Text processing problem (topic classification, sentiment analysis) ▪ Implement solutions ○ based on LSTM ○ using transfer learning using from BERT/GPT-2 **Due date: 3.2.**
30.1.	Deep Reinforcement Learning	▪ Review Reinforcement Learning ▪ (Deep) RL approaches ○ Vanilla Policy Gradient ○ Trust Region Policy Optimization ○ Proximal Policy Optimization ○ Deep Deterministic Policy Gradient (DDPG) ○ Twin Delayed DDPG ○ Soft Actor-Critic ▪ Practical example: AlphaGo/AlphaZero	
3.2.			
6.2.	Backup/review		