The following is a list of questions about the content of the 13 lectures on Advanced Machine Learning taught in Winter Semester 2024/2025. These are not the exam problems, some questions are rather deep, some are too simple. The questions should aid you in better accessing the contents of the lecture.

Lecture 1: Introduction & Basics

- What do artificial neurons have in common with real neurons?
- What is an SVM?
- What is an activation function?
- How are these used in context of SVM?
- How would you group the different activation functions? Why?
- Which ones are used where? (This extends to later lectures!)
- Why don't we use the linear function as an activation function for all layers?
- What is an artificial neural network (ANN)?
- What are weights and biases?
- What are (input, hidden, output) layers?
- What are the dimensions of the weights and biases?
- Is a neural network a (continuous, differentiable, analytic) function?
- What is a feedforward neural network (FNN)?
- How are FNN evaluated?
- How many parameters does an FNN have?
- How would you implement a FNN? (Extends to later lectures.)
- How could you change a FNN into other neural networks (NN)?
- Why are NN successfully used nowadays?
- How would you speed up the computations in an NN?
- What is universal approximation (UA)?
- Are NN universal approximators?

Lecture 2: Finding the Network Parameters

- What is the universal approximation theorem (UAT)?
- How many neurons/layers do we need?
- Is that true for any activation function?
- Can you give a pictorial proof of the UAT for special dimensions and/or activation functions?
- How do you approximate/interpolate/construct a function by an NN?
- What is a cost function? How do you choose it?
- What is the method of steepest descent/gradient descent (GD)? What are the mathematical preliminaries that you need for the method?
- What is Newton's method? What are the mathematical preliminaries that you need for the method?
- How do you determine the step length in these methods?
- What is the difference between the differential and the gradient?
- Prove that the negative gradient is the direction of steepest descent.
- Are there variants that take other directions of descent? How are these determined?
- What is the relation of Newton's method close to a minimum and GD?
- What is SGD? Why do we use it?
- What is a minibatch?
- What is backpropagation? How do we derive it in case of a FNN?
- How does the multi-dimensional chain rule look like?
- How did we change the chain rule to fit our needs?
- What is the meaning of the notation $\partial C/\partial \mathbf{v}$ for some vector $\mathbf{v}$?
- What is the meaning of the notation $\partial C/\partial \mathbf{A}$ for some matrix $\mathbf{A}$?

- How do we compute the derivative of the cost function c with respect to $\mathbf{W}$, $\mathbf{x}$, and $\mathbf{b}$, when we know the derivative of the cost function C with respect to the vector $\mathbf{z} = \mathbf{W}\mathbf{a} + \mathbf{b}$?
- How do we compute the derivatives of the cost function C with respect to the affine linear combinations $\mathbf{z}_i$ in the ith layer? In which order?
- How do we efficiently implement feedforward and backpropagation with minibatches in a FNN?

Lecture 3: Variants of SGD & Regularization

- How do we initialize the weights and the biases in a FNN? (This extends to later lectures.)
- What happens when we initialize everything by zero? By a constant?
- Are the properties of random matrices random?
- What is the circular law, what do Marčenko and Pastur say about singular values of random matrices?
- How do Xavier Glorot and Kaiming He initialize the weights?
- What is orthogonal initialization?
- How would you initialize the weights?
- Do you know variations of SGD? Which ones? How are these motivated?
- Which SGD variant would you prefer? For what reason?
- What do under- and overfitting mean?
- What is the capacity of a neural network?
- What groups of regularization techniques do you know?
- What is the problem of vanishing/exploding gradients?
- How does using ReLU help?
- What is the problem of dying neurons?
- Can you reactivate dead neurons?
- How does gradient clipping work?
- Can you describe L1 and L2 regularization? Which parts of the backpropagation change? How?
- What is weight decay?
- What is batch normalization?
- What is Dropout? How would you implement it?
- What is DropConnect? How would you implement it?
- How do Dropout and DropConnect relate to each other?

Lecture 4: Convolutional Neural Networks (CNN)

- What is a CNN?
- What is the difference between convolution and cross-correlation?
- What is a filter/kernel?
- Do you know examples of convolutions?
- How do the weight matrices look like in a CNN with 1D convolutions?
- What is a Toeplitz matrix?
- Is convolution a linear operation? If so, why?
- What is the difference between full and valid convolution?
- How do convolutions work in 1D/2D/dD?
- What are padding/striding?
- How does the weight matrix of a 2D convolution look like?
- What is weight sharing in context of CNN?
- What is pooling? Which forms of pooling do you know?
- How would you combine convolutions and dropping for reasons of efficiency?
- What are the changes in feedforward/backpropagation/update of the parameters when using a CNN in place of a FNN?

- Where do we need to flip the kernels?
- Why is the kernel for 2D rotated?
- How does backpropagation work for pooling layers?

Lecture 5: Implementation of CNN, History, & DFT

- Which CNN mark important steps in the development? Why?
- What is data augmentation? Why do we need this?
- Which types of data chunks do we have to combine in the training of a CNN when using minibatches and filter banks? 1D/2D/3D/4D/5D/...?
- What is a circulant matrix?
- What is the relation between convolution and multiplication by a (block) circulant?
- What are the eigenvalues and eigenvectors of a circulant?
- What is the DFT/IDFT?
- What is the relation between convolutions and the DFT/IDFT?
- Why do we obtain a block circulant with circulant blocks in the 2D convolution case?
- What are the eigenvalues and eigenvectors of such a matrix?
- How do DFT/IDFT change in 2D? dD?

Lecture 6: FFT, im2col, & Winograd

- What is a separable kernel?
- Can you describe the DFT/IDFT for dD?
- What is the FFT?
- How would you derive an FFT for $n = n_1 \cdot n_2$?
- How can you implement the FFT recursively? How many operations do you need?
- How do the indices in the radix2 DIT FFT change?
- What is the Cooley-Tuckey Factorization?
- What is the scrambled DFT space? Why is this useful for convolutions?
- How do you perform convolution using padding, FFT, and IFFT?
- What is the NCHW format?
- What is a filter bank?
- How does convolution in the NCHW format look like?
- What is im2col? Why do we use it? What is the disadvantage of im2col?
- How can we implement im2col in Python?
- How does Winograd's minimal algorithm look like?
- Are there similarities with the FFT?
- Why do Winograd convolutions speed up CNN? When?
- How would you implement Winograd convolutions?

Lecture 7: Understanding CNN & Adversarial Attacks

- How does the first layer of a trained CNN look like?
- What can be said about higher layers?
- What is attribution?
- What is a saliency map?
- How do CAM and Grad-CAM work? When?
- What is a deconvolutional network?
- What is guided backpropagation?
- Can you describe the differences between backpropagation, deconvolution, and guided backpropagation?

- What is feature visualization?
- Which things are detected by which layers of a CNN?
- What does Deep Dream do?
- What is an activation atlas?
- What is an adversarial example?
- How do adversarial attacks work?
- What is BFGS, L-BFGS?
- What is the curvature condition?
- What is the secant condition?
- What is the fast gradient sign method? How does it work?
- How do images look like that are classified with high confidence?
- What kinds of adversarial attacks do you know?
- Do adversarial attacks work in the real world?
- Do we need knowledge about weights, biases, and filters of the CNN for an adversarial attack?
- What is adversarial training?
- What is XAI and why should you use it?
- Are humans susceptible to adversarial attacks?

Lecture 8: Recurrent Neural Networks

- What is an RNN?
- For what kind of data do we use RNN?
- What is the difference between an RNN and a FNN?
- What is unfolding? How is it used?
- Are there different types of RNN concerning input and output?
- What is the difference between a Jordan and an Elman RNN?
- How many parameters does a vanilla Elman RNN have?
- What is the relation between RNN and dynamical systems?
- How do eigenvalues of a certain weight matrix and stability relate to each other?
- What is backpropagation through time? How does it work?
- How does the update of the weights and biases look like?
- In which order can these updates be computed?
- How would you implement RNN?

Lecture 9: LSTM & GRU

- What is the vanishing/exploding gradient problem in RNN?
- What activation functions should be used in an RNN? Why?
- What is the problem of long-term dependencies?
- What is an LSTM?
- What are gates? How many does an LSTM have? What do they do?
- What are peephole connections?
- What does "coupling the input and forget gates" mean?
- What is a GRU?
- How does BPTT for LSTM work?
- What does weight sharing in context of RNN mean?

Lecture 10: Residual Neural Networks & Variants

- When is a network considered a deep network? (Philosophical question, it's like a sparse matrix, so tell me your thoughts on this one.)

- Why should a trained deeper net be at least as good as a smaller one where some layers are omitted?
- What do you observe in practice when training the network structures treated thus far?
- What is the idea underlying a Highway Network?
- How is a Highway Network implemented?
- Why are these termed "Highway" Networks?
- What is lesioning a layer? How does this look like for Highway Networks?
- What is the idea underlying a Residual Network (ResNet)?
- How do these look like?
- What happens in training a ResNet compared to standard deep nets?
- What variants of ResNet exist? Which one is the best method? Do you have an idea why?
- How many paths connecting input to output do we have in a ResNet?
- What happens when lesioning a layer in a ResNet?
- What is a DenseNet?
- What is training with stochastic depth?

Lecture 11: Autoencoder & GAN

- What is the difference between supervised and unsupervised learning?
- What is an autoencoder? What is the decoder, latent space, encoder?
- What do we need for a CNN-Autoencoder?
- What is the manifold hypothesis?
- What applications do autoencoders have?
- How does U-Net look like?
- What is the relation between a linear autoencoder and the PCA?
- Why can we cluster data using the latent space of an autoencoder? What did we observe?
- How does interpolation in the latent space work?
- What happens in regions where no data gets mapped to?
- What is a Bregman divergence?
- How does Kullback-Leibler divergence look like? What does it measure?
- What is a variational autoencoder? How does the cost function look like?
- What does the reparameterization trick do?
- How can we implement a variational autoencoder?
- What is the Bayesian interpretation of an autoencoder?
- What is a posterior-collapse?
- What is a GAN? What is a generator and what is a discriminator?
- What cost functions are used in a GAN, how do they relate to each other?
- How does the training of the generator work?
- In which sense does a GAN converge?
- What happens in the loss curves? When do we stop?
- What is the recipe for a Deep Convolutional GAN (DCGAN)?
- What are Wasserstein distances? What kind of objects do the compare?
- How can we rewrite Wasserstein-1?
- What is a Wasserstein GAN (WGAN)?
- What is a critic?
- What is a WGAN with gradient penalty (WGAN-GP)?

(Wasserstein GANs are not part of the Curriculum in Winter Semester 2024/2025.)

Lecture 12: Attention & Transformer

- What is attention?
- What is hard and soft attention? In which domain is it used? Which of these is differentiable?

- What is local and global attention? In which domain is it used?
- What is additive and multiplicative attention?
- How are the retrieval system terminology terms “query”, “key”, and “value” used in attention mechanisms?
- What is the transformer?
- What is positional embedding?
- What is (masked) scaled dot-product attention?
- What is multi-headed attention?
- What is BERT?
- How is BERT trained? What tasks are used in the pre-training?
- How many parameters does BERT have?
- What is Masked Language Model (MLM)?
- What is Next Sentence Prediction (NSP)?
- What kinds of embeddings are used in BERT? How are these combined?
- What is GPT?
- What does “zero-shot” or “few-shot” learner mean?
- Describe the three training phases of ChatGPT.
- What is the Vision Transformer?
- What is CLIP? WHich loss function is used there?
- What is a diffusion model?
- What is a latent diffusion model?

Lecture 13: LLM, Finetuning, Fast Inference

- What are neural scaling laws? Why are they useful?
- What is the Chinchilla Scaling Law?
- What are RASP and Tracr? How do they help to understand the transformer architecture?
- What is parameter-efficient finetuning (PEFT)?
- What is LoRA? How are the matrices initialized? How many parameters are saved?
- What is VeRA? How are the matrices initialized? How many parameters are saved?
- What is QLoRA? What is the relation to LoRA?
- What are Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT)?
- What is linear quantization? What is symmetric absmax quantization? What is zero-point quantization?
- What are normal floats (nf4)?
- What is double quantization?
- What is LLM.int8()?
- What are GPTQ and OBQ?
- What is the KV cache? What elements are stored? How does this speed up the inference?
- What is Flash Attention? What Flash Attention 2? What is the difference?