Chair of Visual Computing and Artificial Intelligence
School of CIT
Technical University of Munich

Esolution
Place student sticker here

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

Introduction to Deep Learning

Exam:	IN2346 / endterm	**Date:**	Tuesday 13th February, 2024
Examiner:	Prof. Dr. Matthias Nießner	**Time:**	10:30 – 12:00

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8
I								

Working instructions

- This exam consists of **20 pages** with a total of **8 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 95 credits.
- Detaching pages from the exam is prohibited.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- Do not write outside of the solution boxes.
- Physically turn off all electronic devices, put them into your bag and close the bag.

Left room from __________ to __________ / Early submission at __________

Problem 1 Multiple Choice (18 credits)

Mark correct answers with a cross ☒
To undo a cross, completely fill out the answer option ■
To re-mark an option, use a human-readable marking ×■

Please note:

- For all multiple choice questions any number of answers, i.e. either zero (!), one or multiple answers can be correct.
- **For each question, you'll receive 2 points if all boxes are answered correctly (i.e. correct answers are checked, wrong answers are not checked) and 0 otherwise.**

1.1 Which of the following statements are true about 2D Average Pooling?

☐ It does not have any learnable parameters.

☐ It always performs better than 2D Max Pooling, since it does not ignore any input features.

☐ It reduces the spatial dimensions of the input.

☐ It reduces the channel dimension of the input.

1.2 Which of the following statements are true about deep learning optimizers?

☐ RMSProp is equivalent to SGD with momentum and learning rate divided by an exponentially weighted gradients average.

☐ RMSProp employs a local estimation of the Hessian.

☐ RMSProp requires less memory than Adam.

☐ Adam optimizer uses 1st and 2nd moment of gradients.

1.3 Which of the following statements are true about transformers?

☐ The attention mechanism itself is invariant to order. Equivariant

☐ The concept of transformer can only be applied to text data.

☐ Transformers utilize convolutional layers to gather context information.

☐ Due to masked attention, the decoder output only depends on the previous outputs and the encoder.

1.4 Which of the following statements are true about autoencoders?

☐ The hidden layer should ideally be larger than the input layer.

☐ The encoder and the decoder have the same number of layers.

☐ Any autoencoder can perfectly learn the identity function.

☐ An autoencoder can be used as a lossy compressor.

1.5 Which of the following statements are true about upsampling in neural networks?

☐ Applying a convolution followed by a transposed convolution with the identical filter weights is equivalent to an identity operation.

☐ Transposed convolution adds trainable parameters to the model, whereas interpolation does not.

☐ Removed

☐ Transposed convolution is equivalent to bilinear interpolation (upsampling) followed by applying a standard convolution.

1.6 Given a regression task where the labels are in the range $[-1, 1]$, which of the following final activation functions $f(x)$ could be used?

- ☐ $f(x) = 2\sigma(x) - 1$, where σ corresponds to the sigmoid function.
- ☐ $f(x) = \tanh(10x)$
- ☐ $f(x) = \max(-1, x)$
- ☐ $f(x) = \min(\max(-1, x), 1)$

1.7 Which of the following are true about Generative Adversarial Networks (GANs)?

- ☐ The generator learns by maximizing the probability that a fake image will be classified 'real' by the discriminator.
- ☐ The discriminator aims to learn the distribution of input images, but the generator does not.
- ☐ Removed
- ☐ A fully trained discriminator can be used as a sampling mechanism.

1.8 Given a convolutional layer (kernel size k, padding 1, stride 1, m convolution filters), which of the following might affect the output shape of the convolutional layer (input is a tensor of shape $C \times N \times N$, where C is the number of channels)?

- ☐ N
- ☐ k
- ☐ m
- ☐ C

1.9 Which of the following statements are true about backpropagation?

- ☐ The backpropagated gradient through a tanh activation function is always smaller or equal in magnitude than the upstream gradient.
- ☐ During backpropagation, any of sigmoid/tanh/Leaky ReLU activation functions won't change the sign of gradient.
- ☐ Vanishing gradient causes deeper layers to learn more slowly than earlier layers.
- ☐ The derivative of the loss with respect to a specific weight in your network is negative (e.g. -6). That means that decreasing this weight (by a tiny amount) would decrease the loss.

Problem 2 **Short Questions (20 credits)**

2.1 Consider a binary classification task and two trained classifiers. Classifier A achieves 50% accuracy and classifier B achieves 5% accuracy. With no further training possible, which classifier is more useful (1p) and why (1p)?

2.2 Explain briefly why positional encoding is used in the transformer architecture, particularly for sequential data such as text (1p). Demonstrate what problem would arise if it were not used (2p).

2.3 Are the following two functions appropriate to be activation functions in a backpropagation neural network? If not, why?

- a) $y = \cos(x)$, where $y' = -\sin(x)$
- b) $y = 0.5 \cdot \mathrm{sgn}(x)$, where $\mathrm{sgn}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases}$

2.4 Describe the concept of gradient clipping in training neural networks (2p) and explain why it benefits the training of recurrent neural networks particularly (1p).

0 1 2 3

2.5 During training of a neural network, you noticed that the validation loss is smaller than the training loss. Name two possible explanations for this.

0 1 2

2.6 Show mathematically that sequentially applying two linear layers with biases (W_1, b_1) and (W_2, b_2) is equivalent to using one linear layer if no non-linearities are applied between them (1p). What will be the weight and bias of this new layer (2p)? You may use x to denote the input.

0 1 2 3

2.7 Explain the β_1 and β_2 hyperparameters in the Adam optimizer.

0 1 2

2.8 Explain the Inception layer and list its internal elements.

0 1 2

Problem 3 Backpropagation (11 credits)

Consider the following 1D Convolutional Neural Network:

Conv1D Layer:
(in_channels = 1, out_channels = 1, kernel_size = 3, stride = 1, padding = 0)
with weights (c_1, c_2, c_3)
$\downarrow$
MaxPool1D Layer:
(kernel_size = 2, stride = 1, padding = 0)
$\downarrow$
Fully Connected Layer (FC):
(in_features = n, out_features = 1)
with weights $(w_1, w_2, \ldots, w_n)$
$\downarrow$
Loss Function:
L_2 loss: $L(y, y^*) = (y - y^*)^2$

0 1

3.1 Consider a **7-dimensional vector**: $\vec{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]$ as input to the network. Write down the output dimensions of the Conv1D and MaxPool1D layers, as well as the input dimension n of the FC layer.

For the next questions, we change the network to handle a **5-dimensional input** $\vec{x} = [x_1, x_2, x_3, x_4, x_5]$. Assume the following setting: $\vec{x} = [2, 3, 1, 2, -1]$, convolutional weights $[0.5, 1.5, 0.5]$, FC weights $\vec{w} = [-1, +3]$, and ground-truth label $y^* = 2$.

0 1 2 3

3.2 Calculate the forward pass through each of the network's layers, including the output y and the loss value $L(y, y^*)$. Suggested notation: $\vec{z}$ for the output of the convolutional network, $\vec{m}$ for the maxpool layer.

3.3 Construct a matrix that can be used to express the convolutional layer as a matrix-vector multiplication between the matrix and the input $\vec{x}$.

0
1
2

3.4 Given the loss function $L = (y - y^*)^2$, calculate the derivative $\frac{dL}{dy}$. Also, evaluate its explicit value.

0
1

0
1

3.5 Denote $\vec{m} = [m_1, m_2]$ the outputs of the MaxPool1D layer. Given $\frac{dL}{dy} = 2$, calculate $\frac{dL}{d\vec{m}}$.

0
1
2

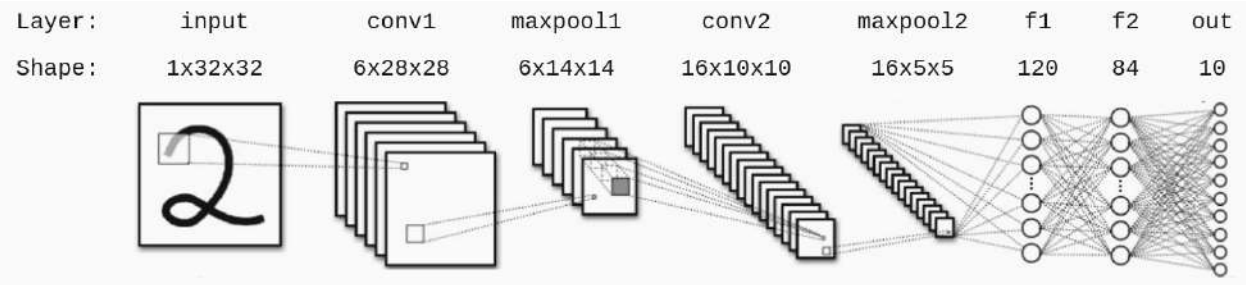
3.6 Denote $\vec{z} = [z_1, z_2, z_3]$ the outputs of the Conv1D layer. Given $\frac{dL}{dm} = [2, 1]$, calculate $\frac{dL}{d\vec{z}}$.

0
1

3.7 Given $\frac{dL}{d\vec{z}} = [1, 2, 3]$, calculate $\frac{dL}{dc_2}$.

Problem 4 CNN (15 credits)

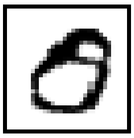
You are contemplating design choices for a convolutional neural network for the classification of digits. LeCun et. al suggest the following network architecture:



For clarification: the shape **after** having applied the operation 'conv1' (the first convolutional layer in the network) is $6 \times 28 \times 28$. No padding is being used. Convolutions use a stride of 1, Maxpool uses a stride of 2 and kernel size of 2.

4.1 Explain the term 'receptive field' (1p). What are the receptive fields of a pixel in the feature maps after the operations 'conv1' and 'maxpool1'? (0.5p each) What is the receptive field of a neuron in layer 'f1' (1p)?

4.2 You and your classmates have each trained their own softmax classifier with a multiclass cross-entropy loss. You test your models with the following image of the digit 8, and obtain the outputs as depicted in the table below:



Model \ Digit	0	1	2	3	4	5	6	7	8	9
Model A	0.45	0.00	0.00	0.00	0.05	0.00	0.05	0.05	0.40	0.00
Model B	0.47	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.41	0.10
Model C	0.30	0.00	0.05	0.00	0.00	0.00	0.20	0.00	0.35	0.10

Table 4.1: Vector of class probabilities, as outputted by models A, B, and C

Which model/models would predict the class correctly for this sample (1p)?
On this sample, which model would yield the largest loss value (1p)? What is its value (1p)? *Note*: You may use elementary functions like log and exp in your answer.

0
1

4.3 Instead of recognizing MNIST digits, you now want to be able to classify whether an image depicts an even or odd digit. What is a single change to the network architecture needed to support this? Choose a solution that is efficient in terms of number of trainable weights.

0
1
2

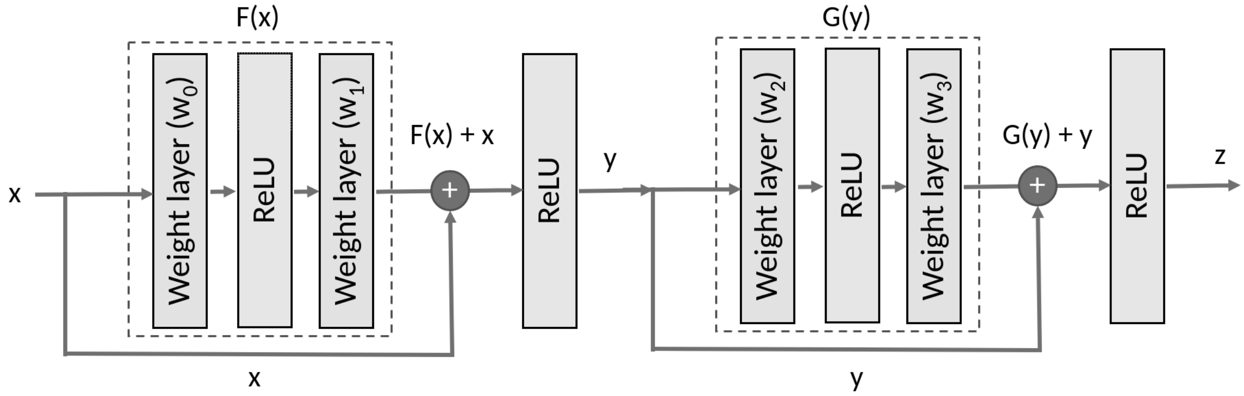
4.4 Instead of taking 32×32 images, you now want to train the network to classify images of size 68×68. Describe two possible architecture changes to support this.

0
1
2

4.5 Your architecture works and you manage to classify digits fairly well. After reading many online blogs, you decide to try out a much deeper network to boost the network's capacity. Describe two problems that you might encounter when training very deep networks.

4.6 You read that skip connections are beneficial for training deep networks. The following image shows a segment of a very deep architecture that uses skip connections. How are skip connections helpful (1p)? Demonstrate this mathematically by finding the derivative of w0 (3p). Hint: consider the chain rule, up-stream gradients.

0 1 2 3 4



Problem 5 Convolutional Kernel (7 credits)

0 1 2

5.1 Compute the forward pass for the given input and convolution kernel as follows. Note that in this case we have a stride of 1 and no padding.

$$\text{Input: } \begin{bmatrix} 1 & 2 & -1 & -3 \\ -1 & 3 & 0 & 2 \\ 1 & 1 & 1 & -2 \\ 0 & 2 & 0 & -3 \end{bmatrix}$$

$$\text{Kernel: } \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

0 1

5.2 Actually, the kernel we used above has a particular function. Explain what kind of effect this kernel has when applying it on an image.

0 1 2 3 4

5.3 Compute the gradient with respect to the input, using the above kernel and the following up-stream gradient.

Up-stream gradient:

$$\begin{bmatrix} 3 & -1 \\ 1 & 2 \end{bmatrix}$$

Problem 6 Activation Function (9 credits)

In November 2015, researchers proposed the activation function "Exponential Linear Unit" (ELU):

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha \cdot (e^x - 1) & \text{if } x < 0 \end{cases} \tag{6.1}$$

6.1 Sketch the function ELU(x) as a graph. Consider the effect of α in your graph. Describe the behavior of ELU(x) for the following cases:

0 1 2 3

- positive values of x
- large negative values of x
- small negative values of x

6.2 Write down the derivative of the ELU function (2p). Express the derivative in terms of the ELU function itself (1p).

0 1 2 3

6.3 Describe 2 advantages and 1 disadvantage of ELU(x) compared to ReLU(x)?

0 1 2 3

Problem 7 Sparsity (7 credits)

0
1

7.1 Name a regularization method that encourages weights to be sparse. Write down its explicit formula (assume it is applied on a weights matrix $W_{m\times n}$).

0
1

7.2 Name one non-learning based approach and one learning based approach to reduce the dimensionality of data.

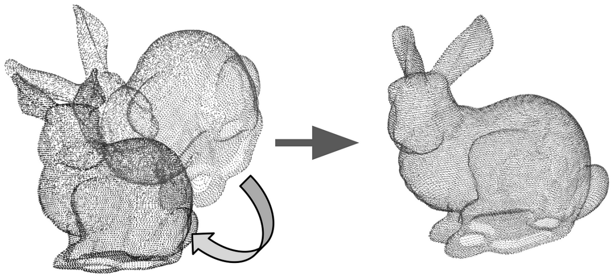
0
1
2

7.3 You trained a classification network on 100 classes, which achieves 95% accuracy on your train and 90% on your validation set. You have deployed your model and noticed that it performs poorly on certain classes. When examining the class your model performs worst on, you realize that your model only achieves less than 32% accuracy on that class (both in train and validation). Assuming the data labels are correct, explain a potential reason, why this can happen and how to solve this problem.

0
1
2
3

7.4 You trained a very large model consisting of many linear layers. You have examined the trained model's weights and noticed many of the weights are very small compared to others in the same layer. Suggest how sparsity in your weight matrices could be leveraged for a lossy model compression (2p). What are the practical advantages of doing that (1p)?

Problem 8 3D Pointclouds (8 credits)



Pointclouds are a simple data structure to represent 3D objects as an unordered set of points.

$$X = \{[x_{i1}, x_{i2}, x_{i3}] \in \mathbb{R}^3 | i = 1, ..., N\}$$

A batch of pointclouds is then stored in a tensor of shape (B, N, D), where B - batch size, N - number of points (assume it is the same for all pointclouds), D - dimension (usually 3).
Rigid transformations, like rotation and translation, are common operations on pointclouds. In 3D, a rotation can be represented by a 3×3 rotation matrix (R) and translation by a 1×3 vector (t), similar to an affine layer. The rotated pointcloud X' can be obtained by $X' = X \cdot R + t$.

$$X' = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & x_{N3} \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} + \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}$$

Consider the bunny shaped pointcloud as depicted in the image above, and a dataset consisting of copies of the pointcloud, each pointcloud randomly rotated and translated in space.
Your task is to design a neural network to align the pointclouds: given any such pointcloud of the bunny, your network will predict the rotation and translation, that applying them would transform the pointcloud back to its canonical position $(0, 0, 0)$ and orientation.
Assume a supervised setting where the dataset contains ground truth rotation matrices and translation vectors for each pointclud.

8.1 Network architecture. Explain why a Fully Connected layer (FC) would be a better choice to use on the pointclouds rather than a convolutional layer.

0
1
2

8.2 Reshaping input data. How can you reshape the input batch of shape (B, N, 3) so that the first FC layer extracts global information directly from all points of each pointcloud?

0
1

0
1
2

8.3 Input and output layer sizes. Given a batch size of $B = 8$ pointclouds, with $N = 100$ points each, what are the sizes of the *input layer* and the *output layer* of your fully connected network?

0
1
2
3

8.4 Loss function for rotation matrices. Rotation matrices belong to a special group of matrices called Orthogonal Matrices, which observe the following property: its transpose is also its inverse.
Once you have estimated a rotation matrix with your network, you notice that applying R_{pred} to the pointcloud causes skewing to its shape. Your colleague told you it was happening because the network did not output an orthogonal matrix, making R_{pred} an invalid rotation matrix.
Suggest a regularization term $Reg(R_{\text{pred}}, R_{\text{GT}})$ that would also take into consideration this crucial characteristic of valid rotation matrices and encourage your output matrix to be orthogonal.
Hint: Avoid matrix inversion, because they are costly and not numerically stable.

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.