DL Exam

Full Name*	
Matriculation Number*	
Course of Studies*	

- You have 90 minutes to finish the exam.
- You are not allowed to use any electronic auxiliaries including calculators. If you have complex mathematical expressions, you may leave the fractions, logarithms, exponentials, etc. as is without having to calculate the exact numerical value.
- You are allowed to use exactly one DinA4 sheet of notes. (Back and front handwritten)
- The space below each question should be sufficient to write down your answer (more paper is available on demand).
- Please keep your handwriting legible and stick to the number of answers asked for. Illegible, ambiguous and multiple answers will be not graded.
  Use a permanent marker!
- Students who registered with "MeinCampus" can check their results after grading there. All others will be notified by the e-mail address linked with the StudOn course access.

☐ You can send me e-mails for upcoming events and open positions to the following e-mail address**:

I have visited the Deep Learning exercise in the following semester ***:	
I have read all the information above and entered required data truthfully:	
Signature	

*This data is required to identify you for the grading process.

**This entry is optional and has no effect on the exam whatsoever. Only fill it in if you want to be put on a mailing list from our lab.

*** This addresses in particular students who did the exercise in a previous semester. We want to ensure bonus points are transferred correctly from previous semesters.

Question	1	2	3	4	5	6	7	Exercise Bonus	Total
Points	10	9	9	9	9	7	7	(6)	60
Achieved									

1 Single Choice Questions (10P)

For each of the following questions, mark the **one correct choice**. Each question has **only one** correct option. No explanation is required.

Question 1.1 1 P.

Which of the following techniques is NOT a common practice to tackle class imbalance?

- ☐ Weighting loss with inverse class frequency
- ☐ Data augmentation
- ☐ Oversampling
- ☐ Batch normalization

Question 1.2 1 P.

What is ensembling?

- ☐ A visualization technique where different areas in the input image are occluded
- ☐ A network with multiple outputs for different tasks
- ☐ Parallel paths in a network each including a different kernel size respectively to let the network decide on the best kernel parameters
- ☐ Multiple networks trained on the same task where the prediction is a majority vote

Question 1.3 1 P.

Which option can be used to create synthetic data samples?

- ☐ The discriminator part of a Generative Adverserial Network
- ☐ The encoder part of a Variational Autoencoder
- ☐ The decoder part of a Variational Autoencoder
- ☐ The encoder part of a Generative Adverserial Network

Question 1.4 1 P.

Imagine that you want to generate a sequence of words with your RNN after you finished the training. Which of the following sampling strategies does not exist for that purpose?

- ☐ Greedy sampling
- ☐ Beam sampling
- ☐ Recognition sampling
- ☐ Random sampling

Question 1.5 1 P.

Which statement is false when implementing a Convolutional layer?

- ☐ The number of kernels determines the number of output channels
- ☐ Stride bigger than one reduces the image size
- ☐ A "valid" convolution outputs the image of the same size as its input
- ☐ The number of parameters/weights inside the kernel is independent from the image size

Question 1.6 1 P.

Why does initialization matter for the optimization of deep neural networks?

- ☐ Because the problem is often non-convex
- ☐ To avoid too many parameters
- ☐ So we always end up in the same local minimum
- ☐ To quickly find the steepest gradient

Question 1.7 1 P.

What's the main tasks of Object Detection

- ☐ Finding Bounding boxes and Classification
- ☐ Search for boundaries between different objects
- ☐ Increase the image resolution of the regions of interest
- ☐ Fill out the missing part of the image

Question 1.8 1 P.

Which statement about YOLO (You Only Look Once) and Fast R-CNN (Regional Convolutional Neural Network) is True?

- ☐ YOLO produces many object candidate suggestions that are passed to the CNN
- ☐ YOLO combines bounding box prediction and classification in one network
- ☐ Fast R-CNN is in general faster under the same hardware conditions
- ☐ Fast R-CNN can do real time detection while YOLO can not

Question 1.9 1 P.

It is important to create your data set properly to avoid misclassifications during the testing. Which of the following examples is not a confound?

- ☐ Food classification task: The food images were taken with two different cameras regardless of their label
- ☐ Traffic object detection task: training images of cars were taken in daylight, while images of pedestrians were taken on rainy days
- ☐ Clothing classification task: T-shirts are worn only be females while hoodies are only worn by males
- ☐ Language classification task: Spanish voice samples were recorded with a different microphone than the Italian samples

Question 1.10 1 P.

When handling an image classification task, you might only have a few options to tackle the problem but luckily, you have a model that was already trained on a similar task. Which of the following methods could be used to make use of this trained model?

- ☐ Freeze all feature extraction layers and retrain the classification layers at the end
- ☐ Use the whole trained model and only retrain the input layer
- ☐ Combine the original classification layer of the network with a classical (non-trainable) feature extraction approach
- ☐ Assess each layers and pick some layers to form a new model

2 Short Answers (9P)

For each of the following questions, answer briefly in 1-2 sentences.

Question 2.1 1 P.

What's the relation between convolution and cross-correlation?

Question 2.2 1 P.

What does model capacity mean?

Question 2.3 2 P.

Name two possible reasons which could lead to Internal Covariate Shift.

Question 2.4 1 P.

Why are Recurrent Neural Networks suitable for problems with time-series?

Question 2.5 1 P.

What's the benefit of using Long Short-Term Memory Units compared to using Elman Cells ?

Question 2.6 1 P.

Name two measures which can only evaluate segmentation tasks.

Question 2.7 1 P.

As the kernel weights are difficult to interpret, when you want to visualize the effect of a convolutional layer, what would you visualize instead?

Question 2.8 1 P.

What benefit could a 1x1 convolution bring?

3 Feed Forward Network (9P)

You have been assigned to a medical image classification task and you need to identify whether a patient is healthy or not. You created a network with 10 fully connected layers, each followed by a ReLU activation function.

Question 3.1 1 P.

Why do you use ReLU activations instead of Sigmoid activations after each fully connected layer?

Question 3.2 2 P.

Imagine an image size of (X, X).

(a) How many weights does one fully connected layer with a number of Z output neurons need to feed through this image? For simplicity we ignore potential bias terms.

(b) How many weights do we need if we would use a convolutional layer with N kernels of size (K, K)? For simplicity we ignore potential bias terms.

Question 3.3 2 P.

Aside from the number of the weights, state a problem that is associated with using fully connected networks for an image classification task. Further, state one advantage that convolutional layers offer compared to fully connected layers?

Question 3.4 1 P.

Name an option how you would connect the output of a convolutional layer to the input of a fully connected layer?

Question 3.5 1 P.

For simplicity we have ignored a bias term for the layers until now. Why is it useful to learn a bias term in your training?

Question 3.6 2 P.

Now you want to classify the different kind of disease in your "unhealthy" patients. In this case, it could happen that same patient suffer from different diseases at the same time. Which final activation function is suitable in this classification case, Sigmoid activation with Cross Entropy Loss or Softmax activation with Cross Entropy Loss? Why?

4 Recurrent Networks and Backpropagation (9P)

Given is the following network which receives an input $x \in \mathbb{R}$ to predict two outputs $f_1(x) = \hat{y}_1$ and $f_2(x) = \hat{y}_2$ where $\hat{y}_{1,2} \in \mathbb{R}$. It only contains one weight $w \in \mathbb{R}$ which is multiplied with its input and a bias $b \in \mathbb{R}$ which is added to its input. At the end a sigmoid function σ is applied. The states k, l and m are highlighted in the figure as intermediate results. The + operation for the skip connection is a simple addition. The network is visualized in the following figure:

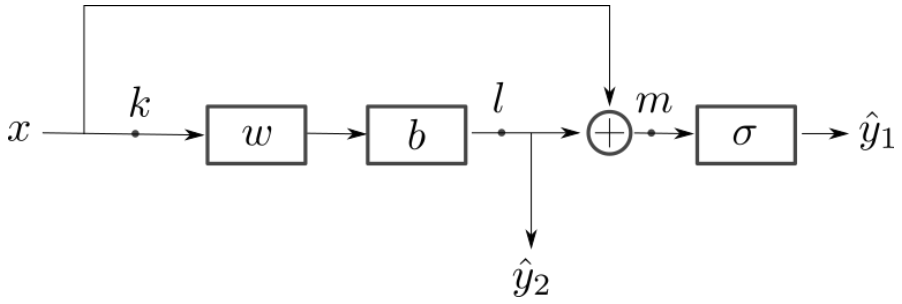

Figure 1: Schematic of a multi-headed residual network.

Question 4.1 3 P.

Define the network f_1 and f_2 as well as the intermediate states l and m as functions depending on the values x, w, b and σ. Furthermore, define the sigmoid function $\sigma(x)$ and its derivative $\sigma'(x)$.

Question 4.2 6 P.

Derive the partial derivatives for the network listed below as general formulas depending on the above-defined variables. You may substitute already computed derivatives in the following derivations. Furthermore, you can assume that the gradients $\frac{\partial L}{\partial \hat{y}_1}$ and $\frac{\partial L}{\partial \hat{y}_2}$ are given.

$$\frac{\partial L}{\partial m} =$$

$$\frac{\partial L}{\partial l} =$$

$$\frac{\partial L}{\partial b} =$$

$$\frac{\partial L}{\partial w} =$$

$$\frac{\partial L}{\partial k} =$$

$$\frac{\partial L}{\partial x} =$$

5 Reinforcement learning (9P)

A few years ago, the Deep Learning tutors wanted to determine who is the best and most efficient tutor, so they decided to play a game during the Friday exercise: Answering a question gives 1 point, taking a submission for the first exercise gives 2 points and taking a submission for the second exercise gives 3 points. However, the tutors are only allowed to move from one student to his/her neighbor in the CIP pool (they cannot move diagonally!). Please refer to Fig. 2 for the distribution of points/requests. Each cell represents one student. To always get the best pathway, tutor Florian (at position F) wants to design a Reinforcement algorithm.

0	1	0	0	2
3	2	1	0	0
0	1	0	2	0
1	F	2	1	1
0	0	0	1	0

Figure 2: Each cell is one seat in the CIP Pool. The number indicates the points a tutor could get by processing the student's request.

Question 5.1 1 P.

At first, Florian has to formalize the problem to apply reinforcement learning. Explain: what would be an action in this example? What would be a reward?

Question 5.2 2 P.

Florian uses a greedy action selection algorithm which only takes the points for the surrounding students of its current position into account for each step, not for the whole CIP pool. Mark the pathway in Fig. 3 for three steps which a greedy algorithm would take. Also mark the best possible pathway for three steps to get the maximum number of points. Please indicate clearly which of the drawn pathway is the optimal and which is the greedy one.

	F			

Figure 3

Question 5.3 3 P.

Using his greedy action selection algorithm, Florian achieves only the second place. What is the main problem of a greedy algorithm? Explain by also defining the concepts of exploration and exploitation. Additionally, explain the epsilon greedy policy.

Question 5.4 3 P.

For a better problem description and to design a trainable algorithm, it is often necessary to not only model the reward of the action but also the next state that you are in. This can be done by modelling a Markov Decision Process. Draw a sketch how the Agent and the Environment play together in such a process and briefly explain the idea.

6 Coding: Optimization (7P)

You want to implement the Nesterov Accelerated Gradient (NAG) optimizer also known as Nesterov Momentum, which is similar to the Stochastic Gradient Descent method with momentum. The goal for this optimizer is to look ahead and compute the gradient in the direction we are currently moving. $v^{(k)}$ is the momentum gradient at iteration k, μ is the momentum, η is the learning rate and $\Delta L(w^{(k)})$ is our gradient with respect to weights. In the case of the first iteration we initialize our momentum gradient with zeros.

$$v^{(k)} = \mu v^{(k-1)} - \eta \nabla L(w^{(k)}) \tag{1}$$

$$w^{(k+1)} = w^{(k)} - \mu v^{(k-1)} + (1+\mu) v^{(k)} \tag{2}$$

Question 6.1 7 P.

Fill the missing parts of the following code to implement the Nestrov Accelerated Gradient (NAG) optimizer in python.

```
import numpy as np
class NAG():
    def __init__(self, learning_rate, momentum=0.9):
        #TODO




    def calculate_update(self, weight_tensor, gradient_tensor):
        #TODO

# end of file
```

7 Coding: Pytorch (7P)

You want to implement the following modified simplified UNet in PyTorch. It consists of multiple convolutional layers (specified in the Model constructor below), each followed by a ReLU activation.
Remark: All convolutional layers and max-pooling layers in PyTorch are per default in "valid" mode, which means no padding at the borders is applied (padding value of 0). If the padding value is larger than 0, padding is applied before the convolution in height and width dimension on both sides of the image. The padding value (in the constructor) defines the padding width of one side of the image. Stride size is by default 1.

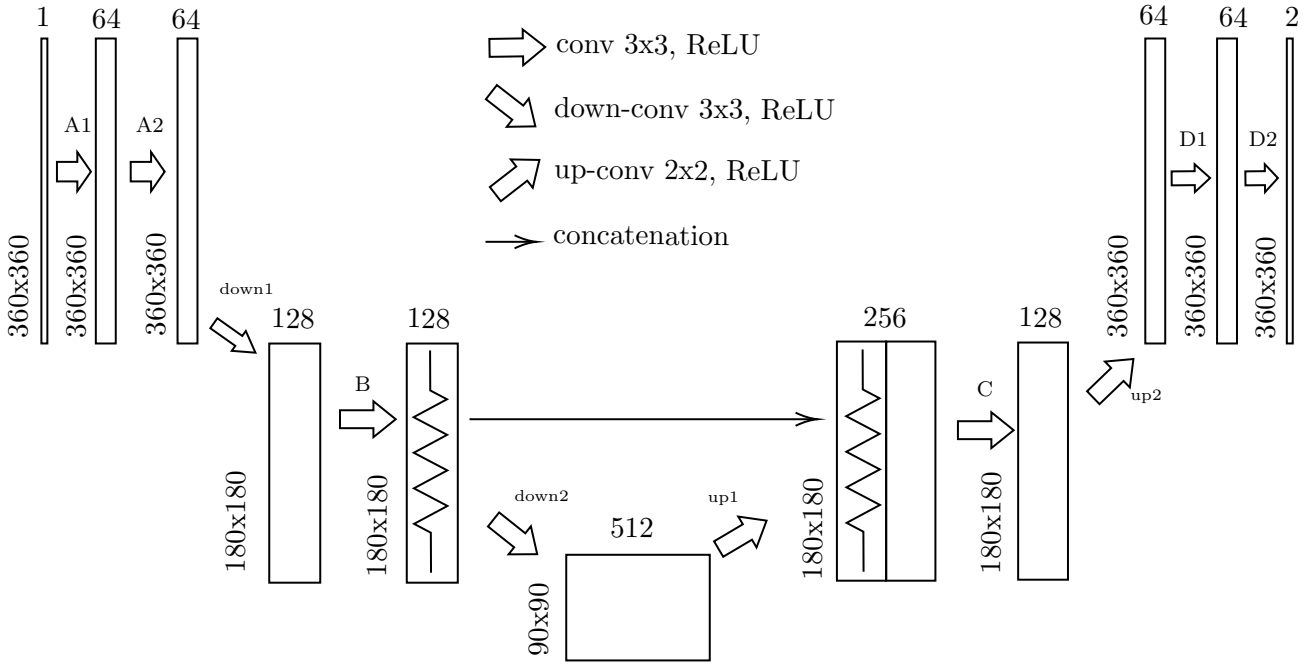

Figure 4: structure of a simplified UNet

Question 7.1 7 P.

Fill the missing parts of the following code to implement the given architecture in python using PyTorch

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class Simplified_UNet(nn.Module):
    def __init__(self):     (2p)

        super(Simplified_UNet, self).__init__()
        self.convA1 = nn.Conv2d(in_channels=1,
                                out_channels=64,
                                kernel_size=3,
                                stride=1
                                padding=1)
        # TODO ######
        self.convA2 = nn.Conv2d(___, 64, kernel_size=3, padding=1)
        self.down1 = nn.Conv2d(___, ___, kernel_size=3, padding=1,
           stride=2)

        self.convB = nn.Conv2d(128, 128, kernel_size=3, padding=1)
        self.down2 = nn.Conv2d(128, 512, kernel_size=3, padding=1,
           stride=2)

        self.conv_up1 = nn.ConvTranspose2d(512, 128, kernel_size=2,
           stride=2)

        # TODO ######
        self.convC = nn.Conv2d(___, 128, kernel_size=3, padding=1)

        self.conv_up2 = nn.ConvTranspose2d(128, 64, kernel_size=2,
           stride=2)
        self.convD1 = nn.Conv2d(64, 64, kernel_size=3, padding=1)
        self.convD2 = nn.Conv2d(64, 2, kernel_size=3, padding=1)
```

```
    def forward(self, x):    (5p)

        convA1 = F.relu(self.convA1(x))

        # TODO
        convA2 =
        down1 =

        convB =
        down2 =

        conv_up1 =
        conv_all = torch.cat([___, ___], dim=1)
        convC =

        conv_up2 =
        convD1 =
        convD2 =



        output = convD2
        return output
```

8 Notes