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	6	8	10	14	9	6	7	(6)	60
Achieved									

1 Single Choice Questions (6P)

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 statement about the Softmax function as the last layer in a neural network for a classification task is **false**?

- ☐ It assigns a probability to each class
- ☐ The individual outputs for one sample are dependent on each other
- ☐ It cannot be used during testing
- ☐ Generally produces a vector per sample for multi-class problems

Question 1.2 1 P.

Which statement about initialization is **false**?

- ☐ Initialization is important for convex problems
- ☐ Using a small positive constant for the bias is helpful against the dying ReLU problem
- ☐ The gradient with respect to the weights is zero when the weights are initialized with zero
- ☐ Xavier initialization takes the number of input features into account

Question 1.3 1 P.

In which scenario would you use under- or oversampling?

- ☐ When evaluating a segmentation task
- ☐ In the dropout layer to boost the influence of certain weights
- ☐ To counter class imbalance during training
- ☐ To create artificial data with variational autoencoders

Question 1.4 1 P.

Which of the following techniques **cannot** be used to visualize important image areas for a neural network with a classification task?

- ☐ Saliency maps
- ☐ Occlusion
- ☐ Guided backpropagation
- ☐ Inception

Question 1.5 1 P.

Which statement about reinforcement learning is **correct**?

- ☐ During training it is sufficient to exploit a known good action
- ☐ Following the exploitation principle we enforce new action sequences during the whole training process
- ☐ Exploration should be used during training to try out new action sequences
- ☐ Exploration should be used during testing

Question 1.6 1 P.

Which statement about segmentation is correct?

- ☐ Instance segmentation can only find one instance of an object in the image
- ☐ Semantic segmentation can differentiate between different classes in an image
- ☐ Segmentation tries to find bounding boxes around the object of interest
- ☐ Semantic segmentation treats multiple objects of the same class as multiple entities

2 Multiple Choice Questions (8P)

For each of the following questions, mark **all choices** that apply. Each question has **at least** one correct option unless otherwise stated. No explanation is required. You will only get points if all correct options are marked.

Question 2.1 2 P.

What is a suitable measure to evaluate the final performance of your fine-tuned neural network?

- ☐ Accuracy on the validation set
- ☐ Mean precision on a combination of training, validation and test set
- ☐ Receiver operating characteristics curve on the test set
- ☐ Recall on the training set
- ☐ F1-Score on the validation set

Question 2.2 2 P.

What are advantages of making a neural network deeper?

- ☐ Exponential feature reuse
- ☐ Less memory consumption
- ☐ Increasingly abstract features
- ☐ Enable training with larger batch size
- ☐ Ability to approximate increasingly complex functions

Question 2.3 2 P.

Which network architectures contain convolutional layers and skip connections?

- ☐ U-Net
- ☐ LeNet
- ☐ VGG16
- ☐ LSTM
- ☐ ResNet

Question 2.4 2 P.

What are problems associated with the simple Elman cell?

- ☐ Small weights are multiplied multiple times and may lead to a vanishing gradient
- ☐ Long term memory loss
- ☐ Many to many mapping is not possible
- ☐ Skip connections lead to short- and long-term memory loss
- ☐ Bounded non-linear activation functions

3 Short Answers (10 points)

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

Figure 1: Example images of a dog and a cat

Andy and Kristine want to work on an image classification task to gather some experience in machine learning. They want to classify whether there is a dog or a cat in an image. They were told to use a Fully Connected Network since it classifies the IRIS flower dataset well. To increase model capacity, they use two hidden layers with many hidden neurons. They trained the model on 800 images of different dogs and 200 images of different cats. Even though they used all the images for training, they could only reach a training accuracy of 80%.

Question 3.1 1 P.

What problem might have occured considering the accuracy of 80%?

Question 3.2 3 P.

Identify two problems with your current experimental setup that limit your training accuracy. How would you solve them?

Question 3.3 2 P.

With the suggestions you gave, they successfully achieved 95 % training accuracy, but they found out that their model does not work well on newly taken images. Name the problem and provide a possible solution to tackle the problem.

Question 3.4 2 P.

By looking at the code, you find out that they used the sigmoid function as activation functions for each layer. What is the purpose of using activation functions? What is the limitation of this activation function?

Question 3.5 2 P.

So far you have used the Stochastic Gradient Descent (SGD) optimization scheme but it is known to have problems with slow convergence and local minima. What term can you add to your optimization scheme to counter these issues. Explain its concept.

4 Recurrent Networks and Backpropagation (14 P)

Given is the following modified recurrent network cell $f(x_t, h_{t-1}) = \hat{y}_t$. It receives an input $x_t \in \mathbb{R}$ and a hidden state $h_t \in \mathbb{R}$ to compute a prediction $\hat{y}_t$. It only contains one weight $w \in \mathbb{R}$ which is multiplied with its input. The states s_t and o_t are highlighted in the figure as intermediate results. The + operation is a simple addition. Using the label $y_t \in \{0, 1\}$ and the loss function $L(y_t, \hat{y}_t) = ||\hat{y}_t - y_t||_2^2$ one can compute the loss and backpropagate it through the network for an input sequence of length $T \in \mathbb{N}^+$ with $t \in [1, T]$ being the current time state. The network is visualized in the following figure:

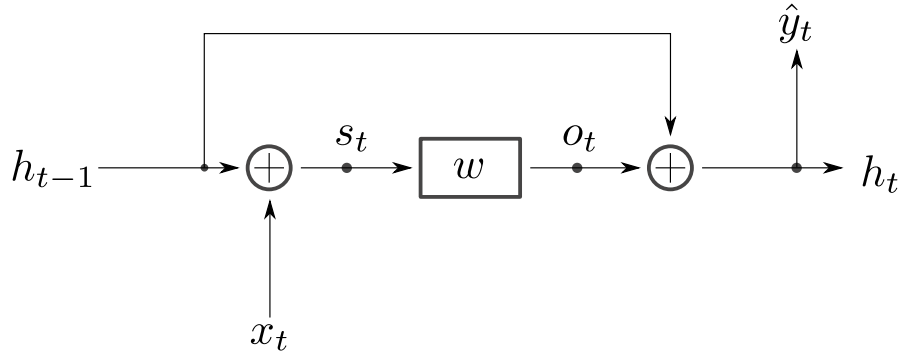


Figure 2: Schematic of a recurrent network cell.

Question 4.1 2 P.

Define the network f as well as the two intermediate states s_t and o_t as functions depending on the values h_{t-1}, x_t and w.

Question 4.2 8 P.

Derive the partial derivatives for the network f 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 gradient $\frac{\partial L}{\partial h_t}$ is given. $\frac{\partial L}{\partial w_t}$ describes the gradient of the weight w_t for one time step t while $\frac{\partial L}{\partial w}$ is the gradient over all time steps used for the update later.

$$\frac{\partial L}{\partial \hat{y}_t} =$$
$$\frac{\partial L}{\partial o_t} =$$
$$\frac{\partial L}{\partial w_t} =$$
$$\frac{\partial L}{\partial w} =$$
$$\frac{\partial L}{\partial s_t} =$$
$$\frac{\partial L}{\partial x_t} =$$
$$\frac{\partial L}{\partial h_{t-1}} =$$

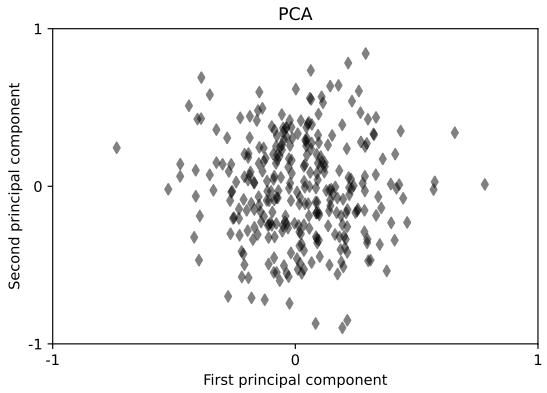
Question 4.3 1 P.

For updating the weights w of your recurrent cell you are using $\frac{\partial L}{\partial w}$ and not $\frac{\partial L}{\partial w_t}$, meaning that you update only after you have processed the whole batch. Why is it not correct to update w after each time step t?

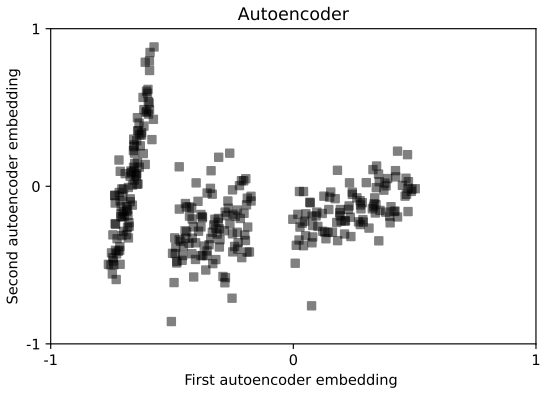
Question 4.4 3 P.

Imagine a text with 10000 characters/letters, where each character would be one input x_t. Name a problem that would occur during the update of a single parameter if you would process the whole text in one batch ($T = 10000$)? Then name and briefly explain a possible solution to overcome this problem.

5 Unsupervised learning (9P)



(a) Two-dimensional PCA embedding of each data sample



(b) Two-dimensional autoencoder embedding of each data sample

Figure 3: Dimensionality reduction on the dataset to two dimensions.

You are given a large unannotated image dataset (40000 images with 256x256 pixel resolution) with the task of measuring the similarity between sample images and identifying clusters in the data. For this purpose, you seek a lower-dimensional representation of the data samples to compare and visualize the samples and clusters in 2D. Conventional dimensionality reduction methods like Principal Component Analysis (PCA) have failed on this task, see Figure 3a. So you decide to try autoencoders, a well-known deep learning-based approach for dimensionality reduction. The autoencoder manages to identify clusters and map similar samples close to each other in the space.

Question 5.1 3 P.

Make a simplified sketch of a classical **autoencoder** architecture and label the two sub-networks. Additionally, label where you would extract the embedding of the autoencoder with the highest dimensionality reduction. Further, define input, output, and a suitable loss function to train an autoencoder.

Question 5.2 2 P.

What is the key difference between autoencoders and PCA? Which component would you have to change in your autoencoder network to learn a generalized PCA?

Question 5.3 2 P.

How can you use an autoencoder for the task of denoising?

Question 5.4 2 P.

Explain the general concept of a variational autoencoder and its key difference compared to a classical autoencoder.

6 Coding: Framework (6P)

Question 6.1 6 P.

You are given the framework of the exercises 1, 2 and 3. Implement a variant of the mean squared error (MSE-V) loss using only numpy and basic python operations. Using a a network prediction vector $\mathbf{y}$ and its according label $\hat{\mathbf{y}}$ which both have the length of the batch size N, this variant is defined as

$$\text{MSE-V}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{\sqrt{N+1}} \sum_{i=0}^{N} (\hat{y}_i - y_i)^2$$

Your loss class must contain a constructor **__init__**, a method **forward(prediction_tensor, label_tensor)** and a method **backward()**. Note: The exact recall of numpy functions is not required to pass the task.

```python
import numpy as np

class MSELoss:
    def __init__(self):
        pass

    def forward(self, prediction_tensor, label_tensor):
    # insert your code here




    def backward(self):
    # insert your code here



# end of file
```

7 Coding: Pytorch (7P)

You want to implement the following modified inception block in PyTorch. It consists of multiple convolutional layers with different strides and padding (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.

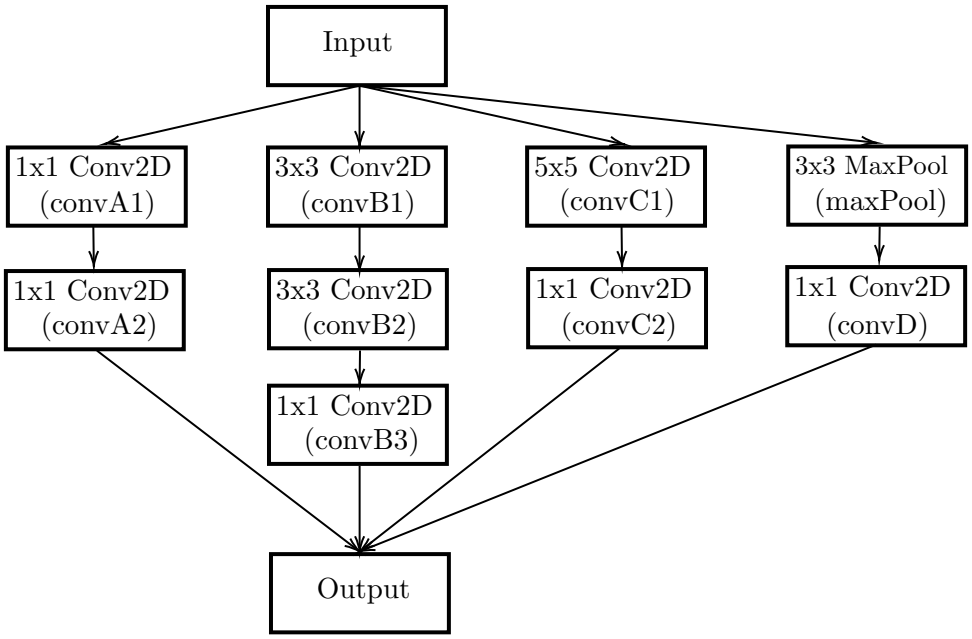


Figure 4: structure of a modified inception block

Question 7.1 7 P.

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

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

class Inception_block(nn.Module):
    def __init__(self):

        super(Inception, self).__init__()
        self.convA1 = nn.Conv2d(in_channels=3,
                                out_channels=64,
                                kernel_size=1,
                                stride=2,
                                padding=0)
```

```
        self.convA2 = nn.Conv2d(64, 128, 1, stride=1, padding=0)
        self.convB1 = nn.Conv2d([A], 128, 3, stride=1, padding=[B])
        self.convB2 = nn.Conv2d(128, [C], [D], stride=[E], padding=1)
        self.convB3 = nn.Conv2d(128, 128, 1, stride=1, padding=0)
        self.convC1 = nn.Conv2d([F], [G], [H], stride=[I], padding=2)
        self.convC2 = nn.Conv2d(64, 128, 1, stride=1, padding=0)
        self.maxPool = nn.MaxPool2d(kernel_size=3, stride=2,
                                    padding=[J])
        self.convD = nn.Conv2d(3, 128, 1, stride=1, padding=0)

    #### please write your answers below:

    # A. _____.  B. _____.  C. _____.  D. _____.  E. _____.

    # F. _____.  G. _____.  H. _____.  I. _____.  J. _____.


    def forward(self, x):
        convA1 = F.relu(self.convA1(x))
        convA2 = F.relu(self.convA2(convA1))
        # Implement the forward function below

        convB1 =


        convB2 =


        convB3 = F.relu(self.convB3(convB2))


        convC1 =


        convC2 = F.relu(self.convC2(convC1))


        maxPool =


        convD = F.relu(self.convD(maxPool))


        output = torch.cat([convA2, convB3, convC2, convD], dim=1)
        return output
```

8 Notes