Tasks: Internship - Reserach work Deep Learning

Autoren: Jennifer Wendland and Prof. Dr. Maik Kschischo

SS 2025

Literature

1 Bishop, C. M. and Bishop, H. (2024) Deep Learning: Foundations and Concepts. New York, NY, USA: Springer US.

2 Prince, S. J. (2023) Understanding Deep Learning. sThe MIT Press.

3 Zhang, A., Lipton, Z. C. , Li, M. and Smola, A. (2023) Dive into Deep Learning. A. J. Cambridge University Press.

Date		
16.04.25	First Meeting	
23.04.25	Task 2	Preliminaries
30.04.25	Task 3	Linear Neural Networks
07.05.25	Task 4	Multilayer perceptrons
14.05.25	Task 5	Convolutional Neural Networks
21.05.25	Task 6	Recurrent Neural Networks
28.05.25	Task text + exam	
04./11./25.6 and 9.7.25	Q&A session (hybrid)	
18.6 and 2.7	Self study	
23.7	Final presentation	

Table 1: Schedule Overview

1 Introduction

Task 0 Read **Python_install.pdf** and install python/pytorch.

Overall Task 1 Download the pytorch notebooks. Open the notebooks via google colab or jupyter notebooks. Read and run the code of the notebooks. Do the exercises.

Overall Task 2 Read further literature on the subject e.g. in book Dive into Deep Learning [3].

Overall Task 3 Upload the solutions to the tasks to the cloud
`https://cloud.uni-koblenz.de/f/140752612`

2 Preliminaries

Task 1 You can find these exercises at the end of the notebook **Data_Manipulation_exercises.ipynb**.

a Create a tensor X with the numbers (0, ..., 23) of the shape (3,4,2)

b Compare the .shape und the .numel() function by applying them to X. What is the difference between them?

c Get all elements of the first two indices of the second axis of X

d Compute the square root of all elements of X by using the sqrt() function

Task 2 You can find these exercises at the end of the notebook **Linear_Algebra_exercises.ipynb**.

a Compute the following tensor X = torch.arange(24).reshape(2, 3, 4). What is the output of len(X)?

b For a tensor X of arbitrary shape, does len(X) always correspond to the length of a certain axis of X? What is that axis? Can you compute an alternative approach to get the value of len(X)?

c Run A / A.sum(axis=1) and see what happens. Can you analyze the reason and provide corrected code? (A is defined in the notebook before)

d Compute the (sample) standard deviation of all entries of A with the .std(unbiased=True) function (the default value of unbiased is True)

e Compute the (sample) standard deviation of all columns of A

f Assume that we have a vector of 6 observations of a variable Y=(0,1,-1,3,2.5,-0.3) and 6 predicted values (e.g. from a least-squared fit) Ypred=(0.2,1,-1.1,3,2.6,-0.4). Compute the Mean Squared Error of the predictions.

3 Linear Neural Networks

Task 1 You can find these exercises at the end of the notebook **Lineara_Regression_Exercises.ipynb**

a Is it a good idea to initialize the weights to zero? You can set the values of the network to zero by using net[0].weight.data.fill_(0)

b Review the PyTorch documentation to see what loss functions are provided. Replace (in the concise version) the loss by Huber's loss.

c Review the PyTorch documentation to see what initialization functions are provided. Specify a linear layer and check whether an initialization is made automatically or not.

d How do you access the gradient of 'net[0].weight'?

Task 2 You can find these exercises at the end of the notebook **image_classification_dataset.ipynb**.

a Change the 'batch size' in the data loader. What can you observe?

b Check out the framework's online API documentation. Which other datasets are available?

c What does the reshape function in the call of the show_images function? Can you name 2 alternatives to compute the effect of the reshape function here?

Task 3 You can find these exercises at the end of the notebook **softmax_regression_scratch.ipynb**

a In this section, we directly implemented the softmax function based on the mathematical definition of the softmax operation. What problems might this cause? Hint: try to calculate the size of exp(50). What can you do to solve this problem?

b The function 'cross_entropy' in this section was implemented according to the definition of the crossentropy loss function. What could be the problem with this implementation? Hint: consider the domain of the logarithm. Why is it a bit problematic to add a small constant to the class with probability zero?

c WAssume that we want to use softmax regression to predict the next word based on some features. What are some problems that might arise from a large vocabulary?

Task 4 You can find these exercises at the end of the notebook **softmax_regression_concise.ipynb**.

a Try adjusting the hyperparameters, such as the batch size, number of epochs, and learning rate, to see what the results are.

b Increase the number of epochs for training. Why might the test accuracy decrease after a while? How could we fix this?

Task 5 You can find these exercises at the end of the notebook **weight_decay_Exercises.ipynb**

a How could you get a good value of λ?

b What might be advantages to use L1 regularization?

4 Multilayer perceptrons

Task 1 You can find this exercise at the end of the notebook **mlp_scratch_Exercises.ipynb**. Try adding an additional hidden layer with a different number of neurons than the previous one. Use the tangens hyperbolicus activation function after the second hidden layer and see how it affects the results.

Task 2 You can find this exercise at the end of the notebook **mlp_concise_Exercises.ipynb** Add a second hidden layer with 100 output neurons and use the tanh activation function.
Task 3 You can find this exercise at the end of the notebook **underfit_overfit_Exercises.ipynb**. Name some ideas on how you could perform model selection

Task 4 You can find these exercises at the end of the notebook **dropout_Exercises.ipynb**

a Why is dropout not typically used at test time?

b Is it a good idea to use dropout and weight decay together?

c What happens if we apply dropout to the individual weights of the weight matrix rather than the activations?

5 Convolutional Neural Networks

Task 1 You can find these exercises at the end of the notebook **conv_layer_Exercises.ipynb**

a Construct an image X with diagonal edges
   - What happens if you apply the kernel K in this section to it?
   - What happens if you transpose X?
   - What happens if you transpose K?

b Do some literature research about edge detection and the Sobel Operator

c Apply the Sobel Filter to X

Task 2 You can find this exercise at the end of the notebook **padding_and_strides_Exercises.ipynb**. For the last example in this section, use mathematics to calculate the output shape to see if it is consistent with the experimental result.

Task 3 You can find this exercise at the end of the notebook **pooling_Exercises.ipynb**. Do we need a separate minimum pooling layer? Can you replace it with another operation?

Task 4 You can find this exercise at the end of the notebook **lenet_Exercises.ipynb**. Replace the average pooling with maximum pooling. What happens?

Task 5 You can find these exercises at the end of the notebook **batch_norm_Exercises.ipynb**

a Can we remove the bias parameter from the fully-connected layer or the convolutional layer before the batch normalization? Why?

b Do we need batch normalization in every layer? Experiment with it?

c Is it possible to use dropout and batch normalization?

6 Recurrent Neural Networks

Task 1 You can find these exercises at the end of the notebook **text_preprocessing_Exercises.ipynb**

a Tokenization is a key preprocessing step. It varies for different languages. Try to find another three commonly used methods to tokenize text.

b In the experiment of this section, tokenize the text into words and vary the min_freq argument of the Vocab instance. How does this affect the vocabulary size?

Task 2 You can find these exercises at the end of the notebook **language_models_and_dataset_Exercises.ipyn**
If we want a sequence example to be a complete sentence, what kind of problem does this introduce in minibatch sampling? How can we fix the problem?

Task 3 You can find these exercises at the end of the notebook **sequence_Exercises.ipynb**.

a Improve the model in the experiment of this section.
   - Incorporate more than the past 4 observations? What do you observe?
   - Change the neural network architecture and evaluate the performance.

b Give an example for when a latent autoregressive model might be needed to capture the dynamic of the data.

Task 4 You can find these exercises at the end of the notebook **rnn_scratch_Exercises.ipynb**

a Run the code in this section without clipping the gradient. What happens?

b Change the structure of the RNN and try to improve the results (or get earlier convergence)

Task 5 You can find these exercises at the end of the notebook **rnn_concise_Exercises**.
What happens if you stack two RNNs together?

Task 6 You can find these exercises at the end of the notebook **geru_Exercises.ipynb**.
Compare runtime, perplexity, and the output strings for rnn.RNN and rnn.GRU implementations with each other (just look at the results of the concise implementation of gru and rnn)

Task 7 You can find these exercises at the end of the notebook **lstm_Exercises.ipynb**

a Compare the computational cost and the performance of GRUs, LSTMs, and regular RNNs for a given hidden dimension (use the concise implementation. We use in all of our examples a hidden dimension of 256)

b Compare the computational cost and the performance of GRUs, LSTMs, and regular RNNs for a given hidden dimension (use the concise implementation. We use in all of our examples a hidden dimension of 256)