

# III Übung zur Vorlesung Deep Learning

Lineare Algebra und Einführung in PyTorch

## Übungsaufgaben – Lineare Algebra

### Aufgabe 1

Betrachte die drei Matrizen  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{n \times p}$  und  $C \in \mathbb{R}^{p \times q}$ . Verifiziere die Assoziativität der Matrixmultiplikation  $(AB)C = A(BC)$ . Zeige dazu, dass das  $(i, j)$ -te Element der Matrix  $(AB)C$  gleich dem  $(i, j)$ -ten Element der Matrix  $A(BC)$ . **Hinweis:** Nutze die Definition der Matrixmultiplikation aus der Vorlesung, Folie 11.

### Aufgabe 2

Betrachte die zwei Matrizen  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ , sodass  $AB \in \mathbb{R}^{m \times m}$  und  $BA \in \mathbb{R}^{n \times n}$  quadratische Matrizen sind. Zeige, dass  $\text{tr}(AB) = \text{tr}(BA)$ . **Hinweis:** Nutze die Definition der Spur aus der Vorlesung, Folie 14.

## Übungsaufgaben – Einführung in PyTorch

PyTorch ist ein Deep Learning Framework zum Erstellen und Trainieren von neuronalen Netzen. Für eine Anleitung zur Installation mit Anaconda findest du alle Informationen auf der offiziellen Website <https://pytorch.org/get-started/locally/>.

### Aufgabe 3

Im weiteren Verlauf wirst du ein einfaches neuronales Netz darauf trainieren, handgeschriebene Ziffern zu erkennen. Die Offizielle Website von PyTorch liefert eine Reihe an hervorragenden Tutorials für den Einstieg. Zur Vorbereitung für die folgende Aufgabe empfiehlt es sich, die folgenden kurzen Tutorials durchzuarbeiten:

- 1) [https://pytorch.org/tutorials/beginner/blitz/tensor\\_tutorial.html#sphx-glr-beginner-blitz-tensor-tutorial-py](https://pytorch.org/tutorials/beginner/blitz/tensor_tutorial.html#sphx-glr-beginner-blitz-tensor-tutorial-py)
- 2) [https://pytorch.org/tutorials/beginner/blitz/autograd\\_tutorial.html#sphx-glr-beginner-blitz-autograd-tutorial-py](https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html#sphx-glr-beginner-blitz-autograd-tutorial-py)

Die Übungsaufgabe orientiert sich an dem Tutorial der Website

([https://pytorch.org/tutorials/beginner/blitz/cifar10\\_tutorial.html#sphx-glr-beginner-blitz-cifar10-tutorial-py](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html#sphx-glr-beginner-blitz-cifar10-tutorial-py)) und ist für den Stand der Inhalte aus der Vorlesung angepasst. Für die Bearbeitung der folgenden Teilaufgaben kannst du dich an dem Tutorial orientieren und den vorgegebenen Code für die Teilaufgaben anpassen.

- a) Lade den MNIST Datensatz, der direkt in PyTorch verfügbar ist, und untersuche die Objekte, in dem die Daten gespeichert sind. Beantworte folgende Fragen mit ein paar Zeilen Code.
  - Wie viele Trainingsbilder gibt es?
  - Wie viele Testbilder gibt es?
  - Wie viele Klassen gibt es?
  - Wie viele Features hat ein Bild?
  - Wie viele Datenpunkte pro Klasse gibt es jeweils im Trainings- und Testdatensatz?
- b) Visualisiere ein paar der Bilder aus dem Trainingsdatensatz und verschaffe dir einen Überblick über die Varianz der Bilder.
- c) Erstelle ein einfaches neuronales Netz mit zwei Hidden Layers und einem Output Layer. Beide Hidden Layer sollen jeweils 50 Neuronen haben und das Output Layer 10 Neuronen, eins für

jede Klasse. Die beiden Hidden Layers sollen mit ReLU Aktivierung erstellt werden, und die Output Layer mit der aus der Vorlesung bekannten Softmax Aktivierung.

- d) Erstelle einen Optimizer, der die Gewichte des Netzes mit dem Gradientenabstiegsverfahren anpasst und ein Bewertungskriterium, welches zu dem Lernproblem passt (Siehe Vorlesung, Folie 35+36). Trainiere dein Netz auf den Trainingsdaten und evaluiere es auf den Testdaten. Tune die Hyperparameter deines Netzes (Anzahl an Layer, Anzahl an Neuronen pro Layer) des Optimizers (Lernrate) und die sonstigen Parameter (Batch Size, Anzahl Epochen, etc...) um eine Genauigkeit von > 95% zu erreichen.