

Übungsblatt 3: Deep Learning mit **keras** und **tensorflow**

In diesem Lab werden wir uns mit Deep Learning in Python beschäftigen. Wir werden dazu auf die Bibliotheken **keras** und **tensorflow** zurückgreifen. Diese beiden Bibliotheken haben sich aktuell zum Quasi-Standard in diesem Bereich herausgebildet. **keras** ist dabei ein High-Level-Framework, das auf **tensorflow** aufbaut und dessen Anwendung anwendungsfreundlicher macht. Weiterführende Informationen zu beiden Bibliotheken findet ihr natürlich online, z.B. hier:

- **Offizielle Website von **keras**** [Hier](#) (April 2019)
- **Offizielle Website von **tensorflow**** [Hier](#) (April 2019)
- **Cheatsheet für **keras**** [Hier](#) (April 2019)

Unsere Ziele für dieses Lab

Wir haben uns für diesen Lab Block folgenden Ziele gesteckt:

- Nach diesem Block kennt ihr grundlegende Möglichkeiten um eure Projekte mit Hilfe von **Deep Learning** zu erweitern
- Nach diesem Block kennt ihr die **wichtigsten Python Werkzeuge** um Deep Learning Pipelines in euren Projekten umzusetzen.

Mega Tutorium

Unsere Aufgabe im Mega Tutorium ist es, (a) uns mit künstlichen neuronalen Netzwerken vertraut zu machen und (b) ein solches für ein Vorhersagemodell zu entwickeln um die Zielklasse zu schätzen.

Szenario: Wir verwenden das [UCI Wine Quality Data Set](#) ([data/winequality-red.csv](#) und [data/winequality-white.csv](#)), der uns eine Reihe von Features angibt. Wir werden versuchen eine künstliches neuronale Netzwerk zu bauen, das uns erlaubt, basierend auf den Input Features zu unterscheiden, ob es sich bei einem Wein und Weißwein oder Rotwein handelt. Es handelt sich dabei also wieder um ein Klassifizierungsproblem mit zwei Klassen.

Künstliche Neuronale Netze können nicht nur zur Verarbeitung von strukturierten Daten verwendet werden, sondern eignen sich auch zur Bildverarbeitung. In einem zweiten Beispiel wollen wir (a) ein Bildverarbeitungsnetz entwickeln und dieses (b) verwenden um geschriebene Ziffern zu erkennen.

Szenario: Wir verwenden hierzu das [MNIST Datensatz](#) um ein künstliches neuronales Netzwerk zu erstellen, das hand geschriebene Ziffern erkennt. Ausgangsbasis dafür sind 10.000 Bilder mit 28 x 28 Pixel in Grauskala.

Individualaufgaben

Aufgabe 3a: Weinqualität (**Einfach**: 10 Punkte)

Verwendet den Datensatz aus dem Mega Tutorium und erstellt ein künstliches neuronales Netzwerk um die Qualität der Weine vorherzusagen. Diese Aufgabe kann entweder als Klassifizierung oder als Regression gelöst

werden. Je nach Ansatz muss die Netzarchitektur (Output-Nodes) und die Loss-Funktion richtig angepasst werden.

Aufgabe 3b: Bildklassifizierung (Mittel: 15 Punkte)

Verwendet den [CIFAR-10 Datensatz](#) und entwickelt ein künstliches Neuronales Netz zur Klassifizierung von Bildern (es gibt 10 Bildklassen im Datenset). Ihr könnt euch hier direkt die Python-Version des Datensets herunterladen und diese mit Hilfe von [pickle](#) importieren.

Aufgabe 3c: Verkehrszeichenerkennung (Herausfordernd: 20 Punkte)

Verwendet den Datensatz zu [Verkehrszeichen in Belgien](#) (BelgiumTSC_Training und BelgiumTSC_Testing) um ein künstliches neuronales Netzwerk zu trainieren, das den Typ von Verkehrszeichen erkennt. Ausgangsbasis dafür sind RGB Bilder von Verkehrszeichen mit unterschiedlicher Größe.

Aufgabe 3d: Titanic (Einfach: 10 Punkte)

Verwendet den Titanic Datensatz aus Lab 2 ([data/titanic.csv](#)). Dieses Mal verwendet aber ein künstliches neuronales Netzwerk um einen Klassifizierer zu entwickeln, der voraussagt, ob eine Person das Schiffsunglück überlebt hat, oder dabei verstorben ist. Vergleicht die Qualität des neuronalen Netzes mit eurem Klassifizierer aus Lab 2.

Aufgabe 3e: Immobilienpreise (Einfach: 10 Punkte)

Verwendet den Datensatz des Maklerbüros über Immobilien in Kalifornien aus Lab 1 ([data/housing.csv](#)). Dieses Mal verwendet aber ein künstliches neuronales Netzwerk um einen Schätzer zu entwickeln, der die Immobilienpreise vorhersagt. Vergleicht die Qualität des neuronalen Netzes mit eurem Regressor aus Lab 1.

Aufgabe 3f: Unsupervised Learning (Mittel: 15 Punkte)

Verwendet den California Housing Datensatz und clustert die dort enthaltenen Datensätze mit Hilfe von unterschiedlichen unsupervised learning Ansätzen (z.B. Agglomerativ Hierarchisch, kmeans usw.). Analysiert, wie gut die erzeugten Cluster bestehende Klassifizierungen (z.B. `ocean_proximity`) widerspiegeln. Experimentiert mit unterschiedlichen Einstellungen (single linkage, complete linkage usw.) und Parametern (k). Stellt eure Ergebnisse wenn möglich grafisch dar (z.B. als Dendrogram oder colorierter Scatterplot).