Übung 03
CNN mit **Keras** in **Python**: Erkennung mit eigenen Bildern
INFI-IS
5xHWII

November 14, 2025

Abgabetermin: lt. Moodle
Übungsleiter: Albert Greinöcker

CCA – COMPETENCE CENTRE
HTL Anichstraße

Ziel der Übung:

- Kennenlernen `RNN` und Erweiterungen wie `LSTM` und `GRU` in `Keras`
- Vorbereiten von eigenen Texten für die Verwendung von `RNN`
- Interpretation der Ergebnisse

1 Vorbereitung

Installation

- KERAS sollte schon für Übung 02,03 installiert sein

Folgende Skripte aus unserem github-Projekt `machine_learning_examples` könnten dabei hilfreich sein:

- `ex_05_rnn_imdb.py`: Ein Beispiel für eine Sentimentanalyse mit Filmkritiken (positiv/negativ) der Internet Movie Database
- `ex_05_rnn_custom_data.py`: Wie kann man die Texte so vorbereiten dass sie für eine RNN geeignet sind.
- `ex_05_rnn_reuters_categories.py`: Grundsätzliches Laden der Daten und die Kategorien als Liste

2 REUTERS-Kategorien

Ein RNN-Modell (z. B. LSTM oder GRU) soll entwickelt werden, dass Nachrichtenartikel aus dem Reuters-Datensatz automatisch einer Themenkategorie zuordnet. Der Datensatz ist schon vorhanden und in KERAS aufbereitet (siehe `ex_05_rnn_reuters_categories.py`):

```
from tensorflow.keras.datasets import reuters

# Nur die 10.000 häufigsten Wörter verwenden
(x_train, y_train), (x_test, y_test) = reuters.load_data(num_words=10000)

print(len(x_train), "Trainingsbeispiele")
print(len(x_test), "Testbeispiele")
print(max(y_train), "Kategorien")
```

Er hat den folgenden Aufbau:

```
8982 Trainingsbeispiele
2246 Testbeispiele
45 Kategorien
```

KERAS bietet leider keine Möglichkeit die Kategorie-Namen auszulesen, deshalb hier eine Liste:

```
label_names = [
"cocoa", "grain", "veg-oil", "earn", "acq", "wheat", "corn", "crude",
"money-fx", "interest", "ship", "trade", "reserves", "cotton", "coffee",
"sugar", "gold", "tin", "strategic-metal", "livestock", "retail", "ipi",
"iron-steel", "rubber", "heat", "jobs", "lei", "bop", "carcass",
"money-supply", "alum", "oilseed", "meal-feed", "cpi", "housing",
"rubber", "zinc", "nickel", "orange", "pet-chem", "dlr", "gas", "silver",
"wpi", "strategic-reserves", "wheat-germ"
]

print(f"Beispielklasse: {y_train[0]} {label_names[y_train[0]]}")
```

Wie hat sich die Qualität der Klassifikation durch die Erhöhung der Kategorien verändert?

3 Aufbereiten von bestehenden Texten

Kaggle (kaggle.com) bietet eine Menge von Datensätzen an, die für solche Klassifikationen gut geeignet sind. Hier ein paar Beispiele:

- Text `Document Classification Dataset`: `https://www.kaggle.com/datasets/sunilthite/text-document-classification-dataset`. Enthält ca. 2.225 Textdokumente in 5 Kategorien (Politik, Sport, Technik
- Ecommerce Text Classification `https://www.kaggle.com/datasets/saurabhshahane/ecommerce-text-cl`
- ArXiv Multi-Label Text Classification Datasets (63MB) `https://www.kaggle.com/datasets/kelixirr/arxiv-multi-label-text-classification-datasets`. Wissenschaftliche Papers

Hinweis: Für den Download lässt sich kagglehub gut verwenden, weil gleich in Python integriert (code wird beim Download angeboten)