

Opdracht

Opdracht 1: Neural Network

In deze opdracht gaan we een Neural Network model trainen op een dataset. We zullen daarna in opdracht 2 dit model nog verbeteren d.m.v. convolutions.

1. Download de dataset van <https://www.kaggle.com/datasets/ryanholbrook/car-or-truck>
2. Laad de dataset in met `image_dataset_from_directory` Denk er aan om een splitsing aan te houden tussen train/test/validation.
3. Defineer je model `model = keras.Sequential([...])`, in deze array moet je een aantal layers opnemen en eindigen met `layers.Dense(1, activation='sigmoid')` om een ja/nee keuze te maken op het eind.
4. Compileer je model met `model.compile(optimizer=tf.keras.optimizers.Adam(epsilon=0.01), loss='binary_crossentropy', metrics=['binary_accuracy'])`
5. Train je model met `model.fit`
6. Probeer je model aan te passen zodat hij beter voorspeld
 - E.g. meer lagen, meer nodes per laag, `DropOut` laag gebruiken.
 - Een score van 60-65% (`val_binary_accuracy`) zou haalbaar moeten zijn

Als je tijd over hebt kan je vast aan de bonus beginnen.

Opdracht 2: CNN

1. Probeer nu om een beter model te maken door middel van convoluties. Gebruik hiervoor de volgende lagen: `layers.Conv2D` en `layers.MaxPool2D`
2. Probeer het model nog op een aantal manieren te verbeteren. Een correctheis-score (`val_binary_accuracy`) van ruim 80% zou haalbaar moeten zijn.

Als je klaar bent kun je aan de onderstaande bonus beginnen.

Bonus: Opslaan, laden en model uitvoeren

1. Als je tevreden bent kun je je model opslaan met `model.save`
2. Laad je model weer in met `keras.models.load_model`
3. Voer je model uit op een sample met `model(sample)`.
 - a. Het kan zijn dat je `tensorflow.expand_dims` moet gebruiken om dimensies van een sample passend te krijgen als je met
4. Probeer bij elkaar te printen het plaatje wat je beoordeeld, het oordeel van je model en het daadwerkelijke antwoord.

Hints

Dependencies

Dependencies uit dit ecosysteem op windows installeren kan tricky zijn. De volgende set werkt begin januari 2024.

```
python = "3.11.*" # ^3.11 will not let tensorflow install;

# TENSORFLOW en KERAS
# Dep and pin needed since later versions don't have windows builds
# Hence we cannot follow the automatic choice by Poetry/Tensorflow
tensorflow-io-gcs-filesystem = "0.31.0"
tensorflow = "2.14" # Needs to be followed with `poetry run pip install tensorflow`

# JUPYTER NOTEBOOKS IN VSCODE
ipykernel = "^6.28.0" # Or the whole Jupyter package, but that is more and not
needed to run notebooks inside vscode

# STANDARD DATA SCIENCE LIBRARIES
matplotlib = "^3.8.2"
pandas = "^2.1.4"
numpy = "^1.26.2"
```

Eerste cel van je notebook

Je kunt je notebook beginnen met de volgende cel zodat je wat defaults, handige setting en imports gemeenschappelijk hebt met de rest van de groep.

```
# Imports
import os
import matplotlib.pyplot as plt
import tensorflow as tf

import numpy as np

# Reproducibility
def set_seed(seed=31415):
    np.random.seed(seed)
    tf.random.set_seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
    os.environ['TF_DETERMINISTIC_OPS'] = '1'
set_seed()

# Set Matplotlib defaults
plt.rc('figure', autolayout=True)
```

```
plt.rc('axes', labelweight='bold', labelsizes='large',  
      titleweight='bold', titlesize=18, titlepad=10)  
plt.rc('image', cmap='magma')
```

Model training time

Als je vind dat je model te lang moet trainen kun je een aantal dingen doen:

- Training set verkleinen
- Aantal parameters in het model verkleinen. (Gebruik `model.summary()` om te kijken waar de meeste zitten.)
- Aantal Epochs verminderen. Vooral als je lang doortraint, zonder dat het model beter wordt.
 - Extra mooi is het gebruik van: https://keras.io/api/callbacks/early_stopping/