



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES



inovex

AI Labor - Wintersemester 2019/20

Computer Vision
3. Termin

Robin Baumann, Matthias Richter

Karlsruhe, 18. Oktober 2019

Agenda für Heute

- Datensatz CIFAR-10
- Keras
- Praktischer Teil: Bildklassifizierung
 - Keras
 - MLP
 - CNN

CIFAR-10

Allgemeines

Datensatz vom Canadian Institute for Advanced Research

- 60k 32x32 Pixel RGB Bilder
 - 50k training, 10k testing
- 10 Klassen
 - airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck
 - 6k Beispiele pro Klasse
- CIFAR-10 und CIFAR-100 sind gelabelte Teilmengen des “[80 million tiny images](#)” Datensatzes

airplane



automobile



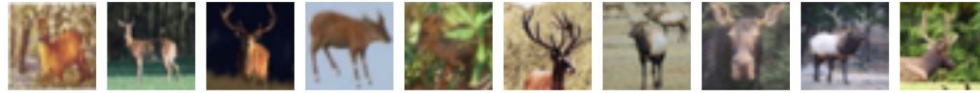
bird



cat



deer



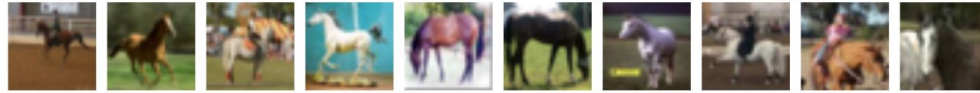
dog



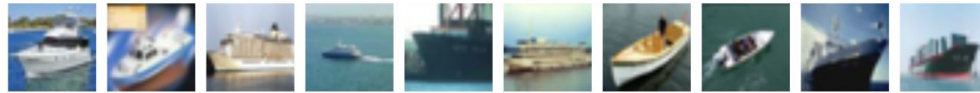
frog



horse



ship



truck



Struktur von CIFAR-10

Batchen

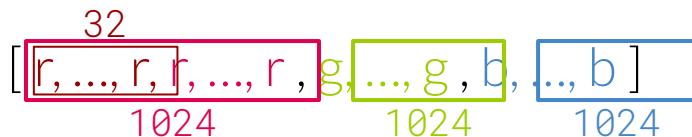
- Aufgeteilt in 6 batches (je 10k Bilder):
 - 5 train batches
 - 1 test batch
- test batch besteht aus jeweils 1k zufällig ausgewählten Bildern pro Klasse
- Jeder batch wird als [pickle](#) File geliefert

Struktur von CIFAR-10

Daten und Labels

- Jeder batch besteht aus einem *dictionary* mit:

- **data**: 10000x3072 numpy array aus uint8
 - jede Zeile speichert ein 32x32 Pixel RGB Bild
 - row-major: Erst Kanal, dann Zeile, dann Pixel



- **labels**: Liste aus 10000 Zahlen im Bereich 0-9
 - **labels[i]** beschreibt die Klasse von **data[i]**

Struktur von CIFAR-10

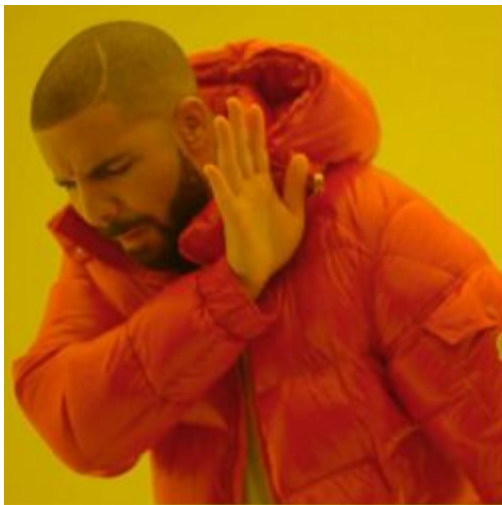
Metadaten

- Zusätzlich gibt es noch eine File `batches.metadata`
 - Enthält ein *dictionary* mit den Label-Namen, z.B.:
 - `label_names[0] == "airplane"`
 - `label_names[1] == "automobile"`

CIFAR-10 ist gelöst

Research Paper	Error rate (%)	Publication Date
Convolutional Deep Belief Networks on CIFAR-10 ^[6]	21.1	August, 2010
Maxout Networks ^[7]	9.38	February 13, 2013
Wide Residual Networks ^[8]	4.0	May 23, 2016
Neural Architecture Search with Reinforcement Learning ^[9]	3.65	November 4, 2016
...
...	3.47	December 18, 2014
...
Search ^[10]
AutoAugment: Learning Augmentation Policies from Data ^[17]	1.48	May 24, 2018
A Survey on Neural Architecture Search ^[18]	1.33	May 4, 2019
GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism ^[19]	1.00	Nov 16, 2018

Keras



tensorflow



Keras

Keras

“Keras is a [high-level neural networks API](#), written in Python and capable of [running on top of TensorFlow](#), CNTK, or Theano. It was developed with a focus on [enabling fast experimentation](#).”

Keras

```
# Modell definieren  
model = Sequential(layers)  
  
# Training vorbereiten  
model.compile(...)  
  
# Modell trainieren  
model.fit(...)  
  
# Modell ausführen  
predictions = model.predict(...)
```

Keras

mit der Sequential API

```
# Modell = Liste verschiedener Layer
model = Sequential([
    Dense(12, input_shape=(345,)),
    Activation('relu'),
    Dense(78, activation='sigmoid')
])
```

```
# Layer anfügen
model.add(Dense(9, activation='softmax'))
```

Keras

... bringt alle üblichen Layer mit

LAYERS

About Keras layers

Core Layers

Convolutional Layers

Pooling Layers

Locally-connected Layers

Recurrent Layers

Embedding Layers

Merge Layers

Advanced Activations Layers

Normalization Layers

Noise layers

Layer wrappers

Writing your own Keras layers

- Wichtige Layer:
 - `Flatten()`, `Reshape(target_shape)`
 - `Dense(num_units)`
 - `Dropout(rate)`
 - `Conv2D(num_filters, kernel_size, ...)`
 - `MaxPooling2D(pool_size, ...)`
- Operationen auf allen Layern:
 - `l.get_weights()`, `l.set_weights(w)`
 - `l.input`, `l.output`, `l.input_shape`, ...

Praktischer Teil

Aufgaben

TODOs implementieren:

1. MLP trainieren (Ziel: >40% test accuracy)
 - › Flatten, Dense, Dropout, ...
2. TensorBoard summaries für train und test
3. CNN trainieren (Ziel: >70% test accuracy)
 - › Flatten, Dense, Dropout, Conv2D, MaxPooling2D, ...

Setup

- Anmeldung am Rechner
 - User: `i-lfm-docker`
 - Passwort: `garten::obst`
- Ethernet auf hskaopen umstellen
- Wired Settings > Security:
 - ☒ No CA certificate is required
 - `testXYZ` entfernen und mit Studentenkürzel ersetzen
 - Passwort eingeben

Setup

- Terminal öffnen
 - `git clone https://github.com/hskaailabcv/source.git`
 - `cd source`
 - `docker-compose up`
- Jupyter: <http://localhost:8888>

Feedback

- Google Forms

<https://forms.gle/5yafxDPP1beLQ4g2A>



Vielen Dank

Robin Baumann

rbaumann@inovex.de

Matthias Richter

mrichter@inovex.de

inovex GmbH

Ludwig-Erhard-Allee 6
76131 Karlsruhe

