



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES



AI Labor - Wintersemester 2019/20

Computer Vision
4. Termin

Robin Baumann, Matthias Richter

Karlsruhe, 4. November 2019

Agenda für Heute

- Theorie:
 - Keras Sequence
 - Keras Functional API
 - Transfer Learning
- Praxis: Wie löse ich ein Problem?
 - PointNet: Paper nachimplementieren
 - Benchmark: reduced ModelNet10
 - Praxis: Retraining mit neuen Klassen

Keras Sequence

Lazy, paralleles Laden von Datensätzen

- Große Datensätze passen nicht immer in den RAM
- Beim Training wird immer nur ein Batch benötigt
- **Sequence** abstrahiert Zugriff auf Datensatz:
 - Lädt nur den benötigten Batch
 - Kann eigenes shuffling implementieren
 - Augmentierung und Vorverarbeitung “on the fly”
- Hier:
 - Tracken der File-IDs
 - Auswahl und Shuffling auf Index

Keras Sequence

```
0 class ModelNetProvider(Sequence):  
1     """  
2     Lazily load point clouds and annotations from filesystem and prepare it for model training  
3     """  
4  
5     def __init__(self, dataset, batch_size, n_classes, sample_size):...  
6  
7     def __len__(self):... Länge := Anzahl an "steps" in Epoche  
8  
9     def __getitem__(self, index):... index := Step in Epoche  
10  
11     def __generate_data(self, batch_samples):... Ausgliederung des Loading & Preprocessing  
12  
13     def on_epoch_end(self):... Am Ende jeder Epoche ausgeführt
```

Keras Functional API

- Mehr Flexibilität beim Erstellen der Modell-Architektur
 - z.B. Multi-Input/-Output Modelle/ Shared Layer etc.
- Layer-Instanzen sind als Funktion aufrufbar
 - erlaubt beliebige Verschachtelung
- Modell wird über inputs/outputs definiert
 - `Model(inputs=[...], outputs=[...])`

Keras Functional API

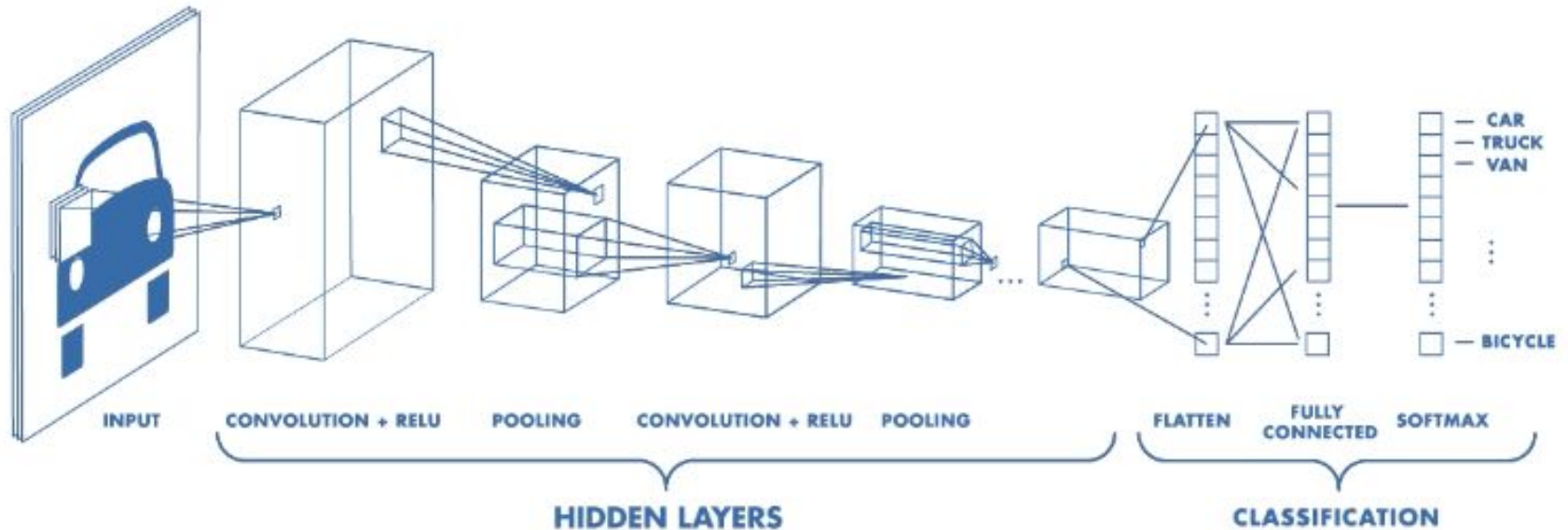
```
from keras.layers import Input, Dense
from keras.models import Model

# This returns a tensor
inputs = Input(shape=(784,))

# a layer instance is callable on a tensor, and returns a tensor
output_1 = Dense(64, activation='relu')(inputs)
output_2 = Dense(64, activation='relu')(output_1)
predictions = Dense(10, activation='softmax')(output_2)

# This creates a model that includes
# the Input layer and three Dense layers
model = Model(inputs=inputs, outputs=predictions)
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(data, labels) # starts training
```

Transfer learning



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

PointNet

Idee

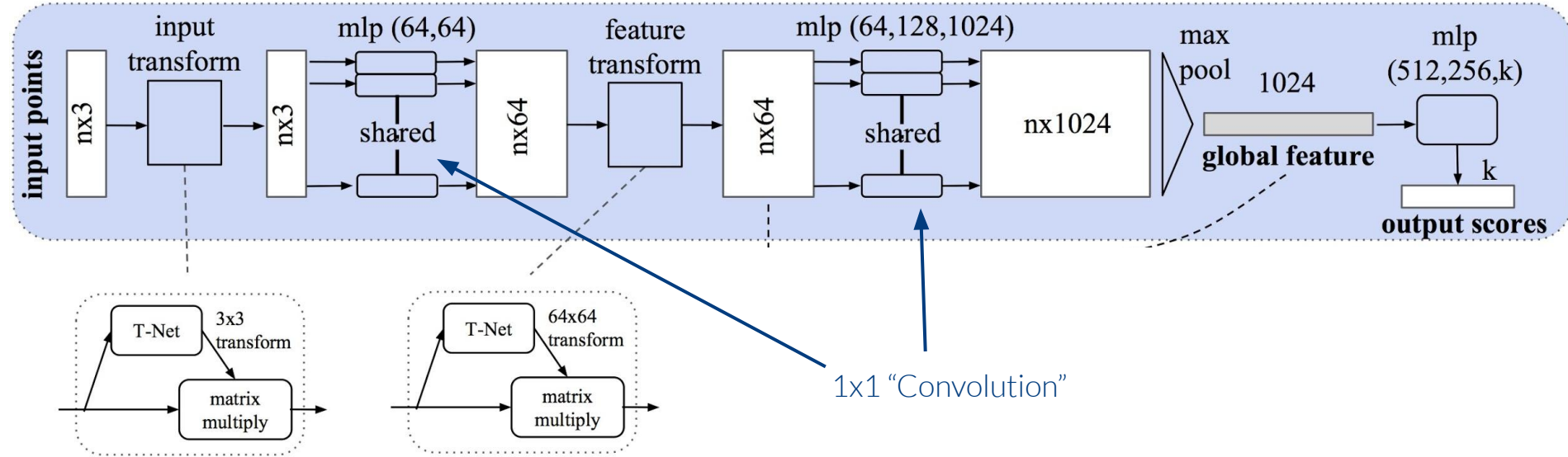
- Typische Anforderungen von Deep Learning Modellen
 - feste Größe / Dimensionalität der Eingaben
 - feste Ordnung / Reihenfolge der Werte
 - Gilt für Punktwolken nicht immer
- Hotfix: 3D Occupancy bzw. Voxel-Grid sampeln
 - → Input hat feste Größe und Ordnung
- Problem:
 - Wie Quantisieren (Informationsverlust vs Speicher)
 - Unnötiger Rechenaufwand

PointNet

Idee

- Lösungsansatz
 - Permutations- und Dimensionsinvariante Netzarchitektur
- Paper (mit Code): <http://stanford.edu/~rqi/pointnet/>

Classification Network

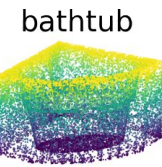
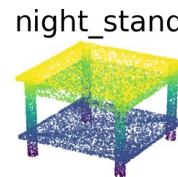
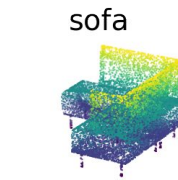
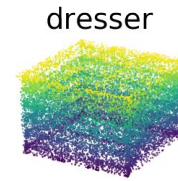
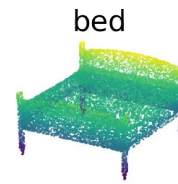


PointNet Architektur

Datensatz:

ModelNet10

- 3D Meshes
- 10 unterschiedliche Klassen
- Training / Testing Split:
 - 3991 Training Samples
 - 908 Testing Samples
- <https://modelnet.cs.princeton.edu>



Aufgaben

TODOs implementieren:

1. PointNet-Architektur implementieren
2. Vortrainierte Gewichte laden
3. Merkmalsextraktor freeze und Classifier neu trainieren
4. Ergebnisse validieren

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/CZwaNJFtQhRdgt7fA>



Vielen Dank

Robin Baumann

rbaumann@inovex.de

Matthias Richter

mrichter@inovex.de

inovex GmbH

Ludwig-Erhard-Allee 6
76131 Karlsruhe

