Handout

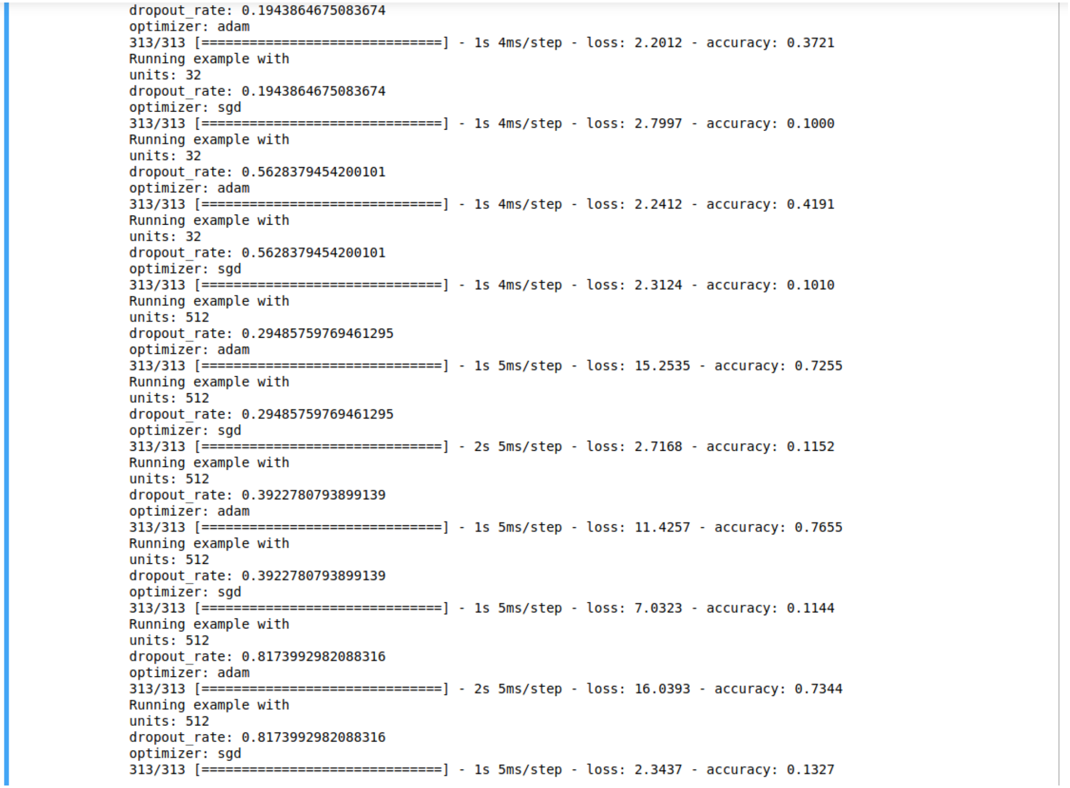
- Git
  - *git clone github.com/kschreiblehner/tensorboard_tutorial.git*
  - Ordner: material
- Tutorial und Aufgaben
  - Ordner: practice

Grundlagen in TensorBoard

Karsten Schreiblehner
11. Juni 2020

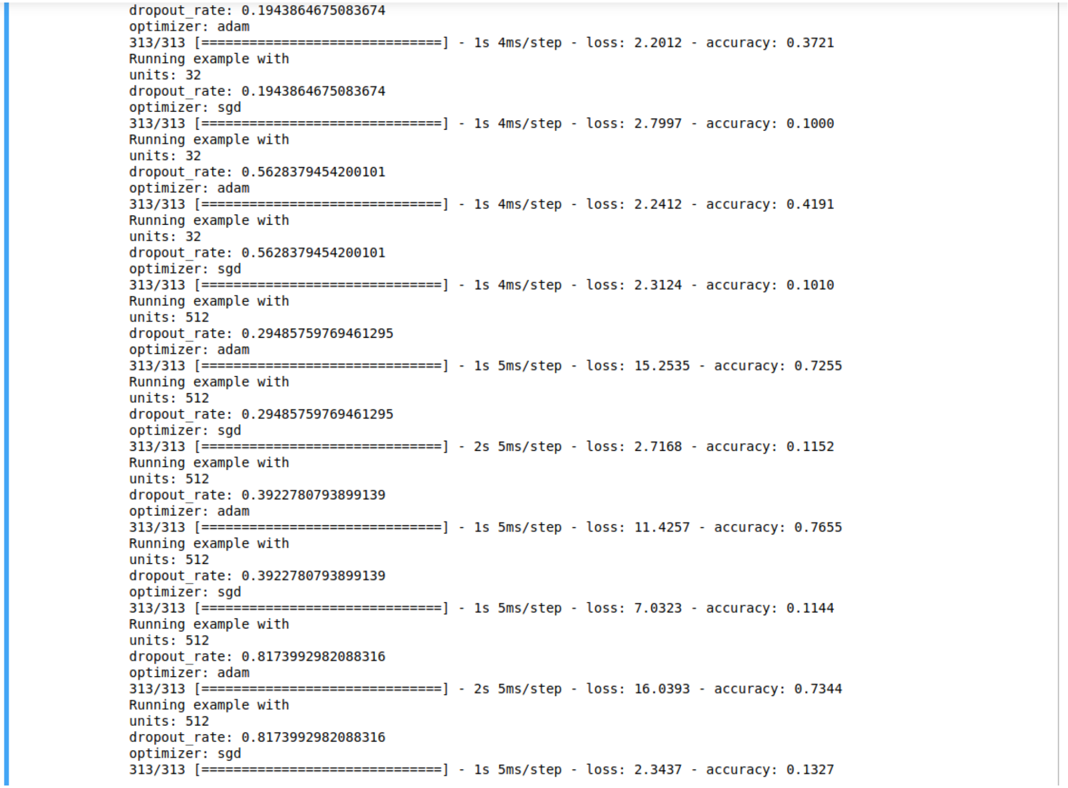
Handout

- Git
  - *git clone github.com/kschreiblehner/tensorboard_tutorial.git*
  - Ordner: material
- Tutorial und Aufgaben
  - Ordner: practice



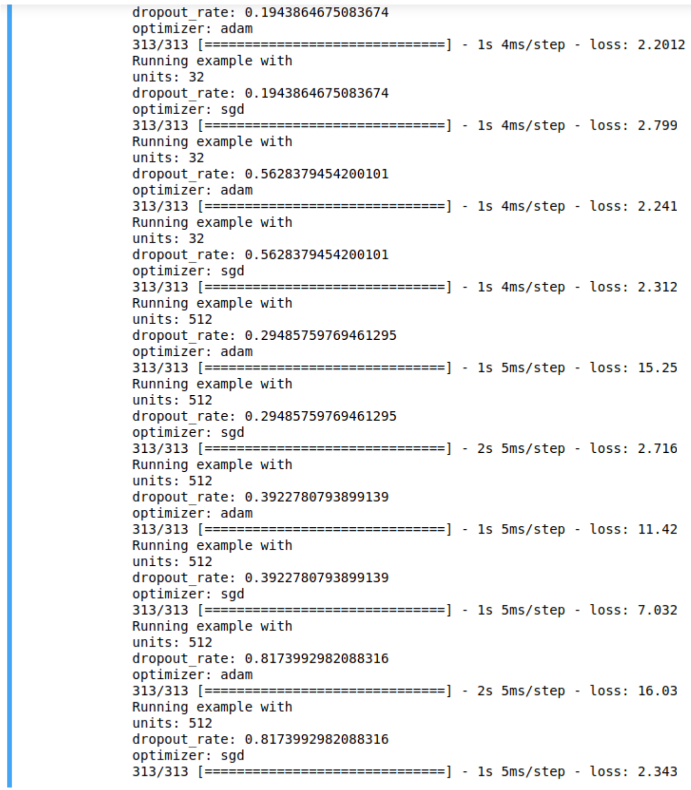
```
dropout_rate: 0.1943864675083674
optimizer: adam
313/313 [==============================] - 1s 4ms/step - loss: 2.2012 - accuracy: 0.3721
Running example with
units: 32
dropout_rate: 0.1943864675083674
optimizer: sgd
313/313 [==============================] - 1s 4ms/step - loss: 2.7997 - accuracy: 0.1000
Running example with
units: 32
dropout_rate: 0.5628379454200101
optimizer: adam
313/313 [==============================] - 1s 4ms/step - loss: 2.2412 - accuracy: 0.4191
Running example with
units: 32
dropout_rate: 0.5628379454200101
optimizer: sgd
313/313 [==============================] - 1s 4ms/step - loss: 2.3124 - accuracy: 0.1010
Running example with
units: 512
dropout_rate: 0.29485759769461295
optimizer: adam
313/313 [==============================] - 1s 5ms/step - loss: 15.2535 - accuracy: 0.7255
Running example with
units: 512
dropout_rate: 0.29485759769461295
optimizer: sgd
313/313 [==============================] - 2s 5ms/step - loss: 2.7168 - accuracy: 0.1152
Running example with
units: 512
dropout_rate: 0.3922780793899139
optimizer: adam
313/313 [==============================] - 1s 5ms/step - loss: 11.4257 - accuracy: 0.7655
Running example with
units: 512
dropout_rate: 0.3922780793899139
optimizer: sgd
313/313 [==============================] - 1s 5ms/step - loss: 7.0323 - accuracy: 0.1144
Running example with
units: 512
dropout_rate: 0.8173992982088316
optimizer: adam
313/313 [==============================] - 2s 5ms/step - loss: 16.0393 - accuracy: 0.7344
Running example with
units: 512
dropout_rate: 0.8173992982088316
optimizer: sgd
313/313 [==============================] - 1s 5ms/step - loss: 2.3437 - accuracy: 0.1327
```

- Speichern der Resultate
- Programm zum Plotten

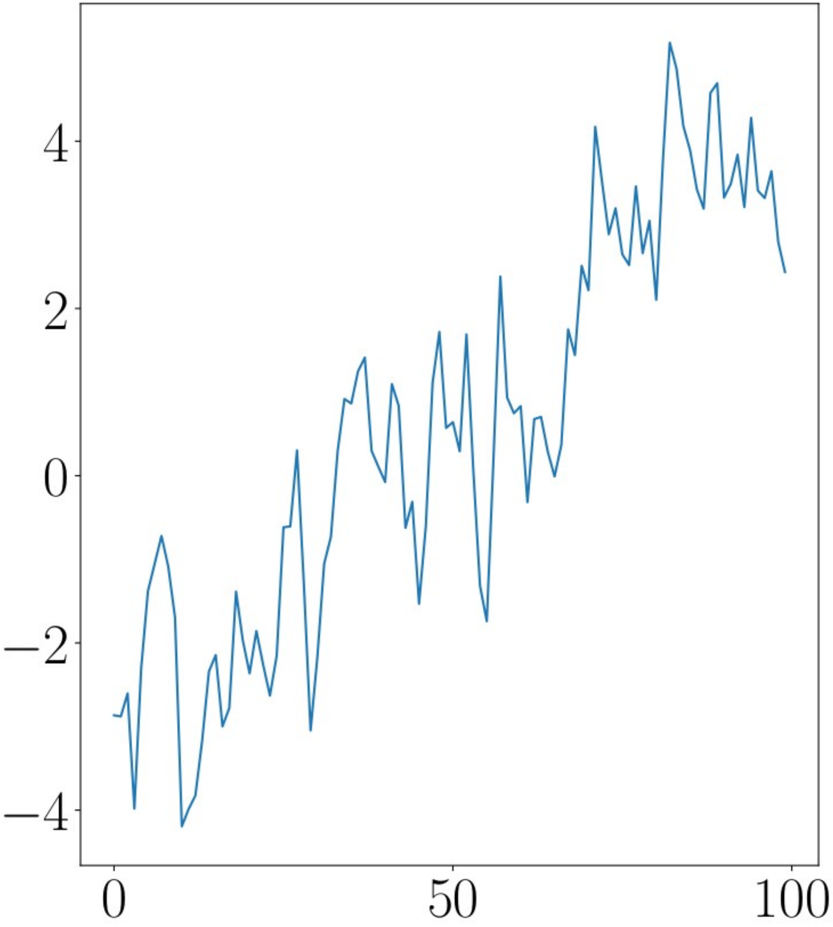


```
dropout_rate: 0.1943864675083674
optimizer: adam
313/313 [==============================] - 1s 4ms/step - loss: 2.2012 - accuracy: 0.3721
Running example with
units: 32
dropout_rate: 0.1943864675083674
optimizer: sgd
313/313 [==============================] - 1s 4ms/step - loss: 2.7997 - accuracy: 0.1000
Running example with
units: 32
dropout_rate: 0.5628379454200101
optimizer: adam
313/313 [==============================] - 1s 4ms/step - loss: 2.2412 - accuracy: 0.4191
Running example with
units: 32
dropout_rate: 0.5628379454200101
optimizer: sgd
313/313 [==============================] - 1s 4ms/step - loss: 2.3124 - accuracy: 0.1010
Running example with
units: 512
dropout_rate: 0.29485759769461295
optimizer: adam
313/313 [==============================] - 1s 5ms/step - loss: 15.2535 - accuracy: 0.7255
Running example with
units: 512
dropout_rate: 0.29485759769461295
optimizer: sgd
313/313 [==============================] - 2s 5ms/step - loss: 2.7168 - accuracy: 0.1152
Running example with
units: 512
dropout_rate: 0.3922780793899139
optimizer: adam
313/313 [==============================] - 1s 5ms/step - loss: 11.4257 - accuracy: 0.7655
Running example with
units: 512
dropout_rate: 0.3922780793899139
optimizer: sgd
313/313 [==============================] - 1s 5ms/step - loss: 7.0323 - accuracy: 0.1144
Running example with
units: 512
dropout_rate: 0.8173992982088316
optimizer: adam
313/313 [==============================] - 2s 5ms/step - loss: 16.0393 - accuracy: 0.7344
Running example with
units: 512
dropout_rate: 0.8173992982088316
optimizer: sgd
313/313 [==============================] - 1s 5ms/step - loss: 2.3437 - accuracy: 0.1327
```

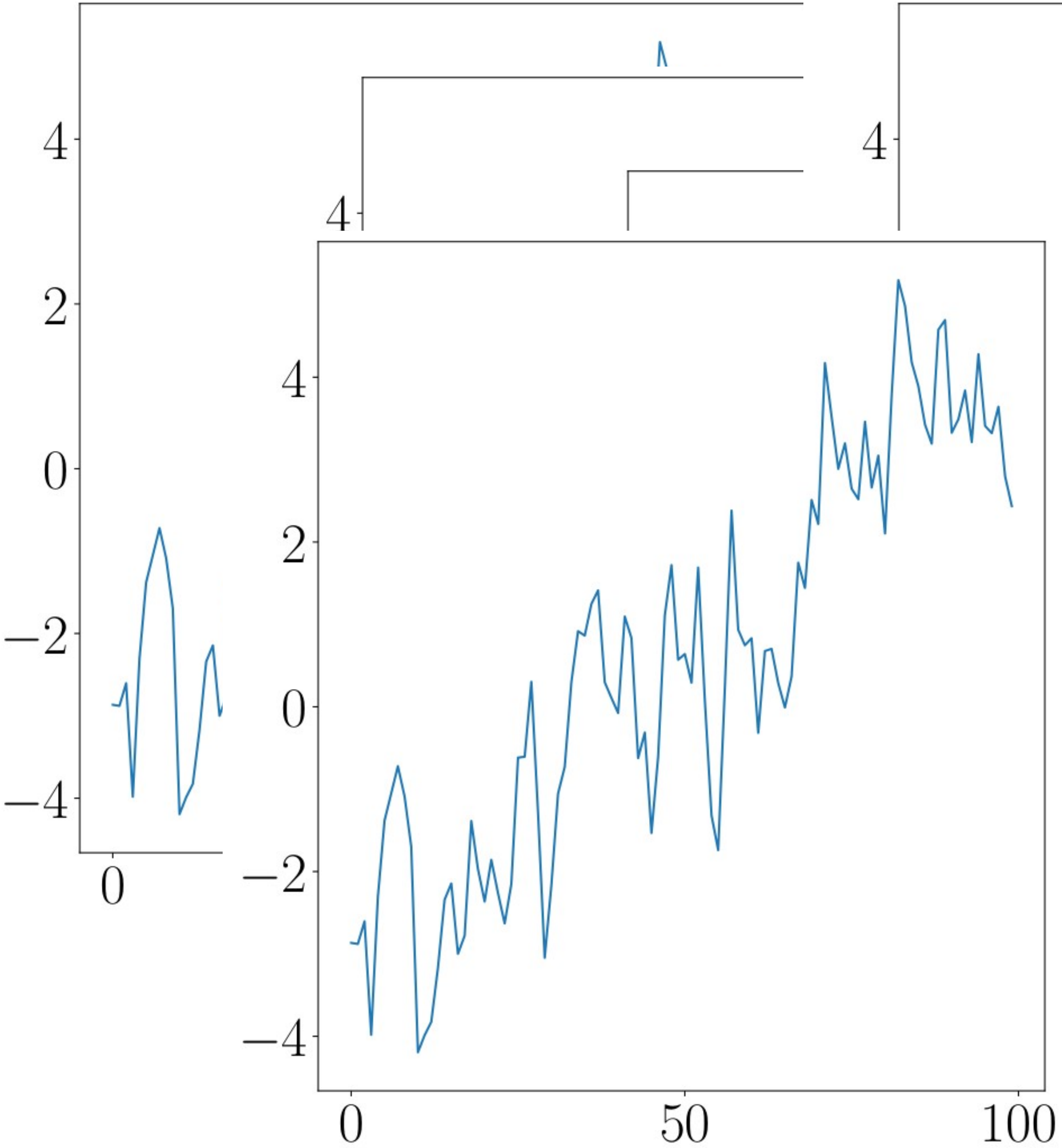
- Speichern der Resultate
- Programm zum Plotten

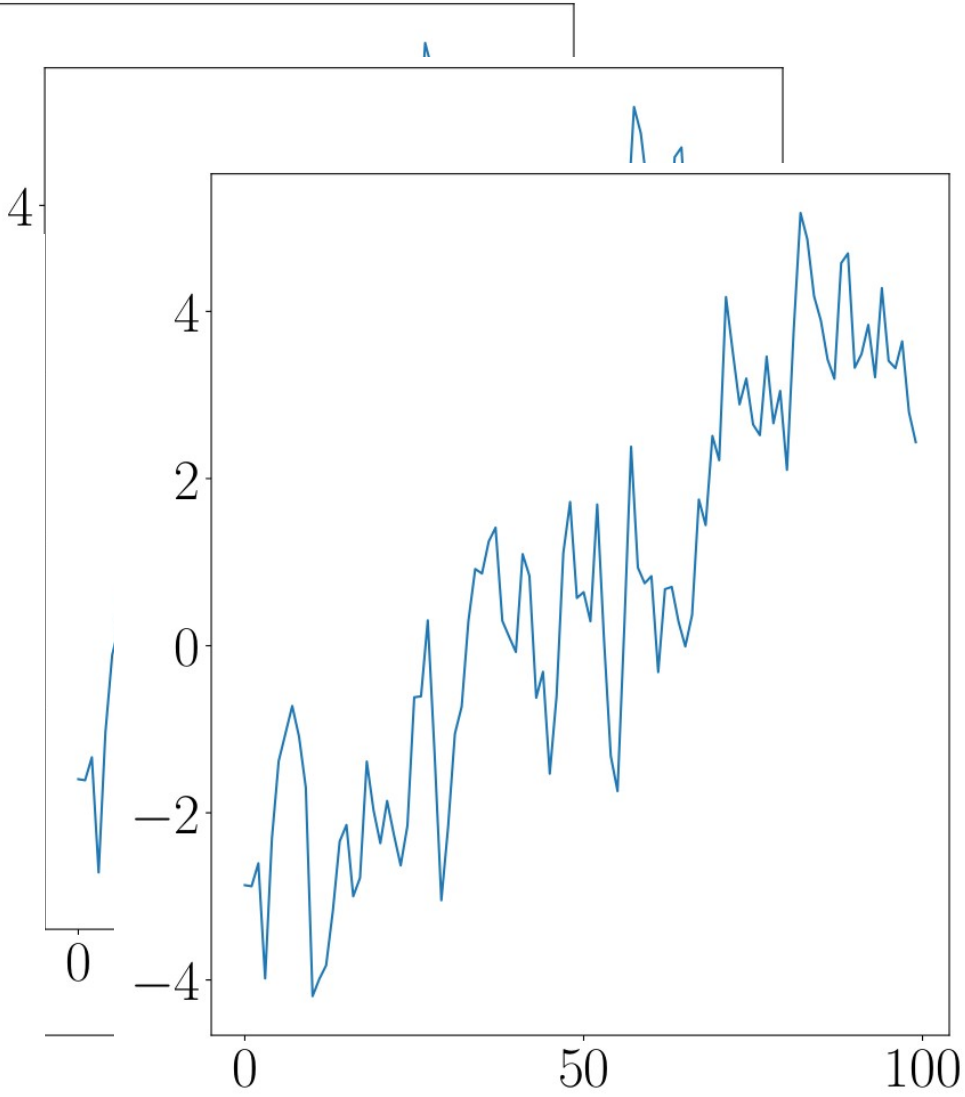


```
dropout_rate: 0.1943864675083674
optimizer: adam
313/313 [==============================] - 1s 4ms/step - loss: 2.2012 - accuracy: 0.3721
Running example with
units: 32
dropout_rate: 0.1943864675083674
optimizer: sgd
313/313 [==============================] - 1s 4ms/step - loss: 2.799
Running example with
units: 32
dropout_rate: 0.5628379454200101
optimizer: adam
313/313 [==============================] - 1s 4ms/step - loss: 2.241
Running example with
units: 32
dropout_rate: 0.5628379454200101
optimizer: sgd
313/313 [==============================] - 1s 4ms/step - loss: 2.312
Running example with
units: 512
dropout_rate: 0.29485759769461295
optimizer: adam
313/313 [==============================] - 1s 5ms/step - loss: 15.25
Running example with
units: 512
dropout_rate: 0.29485759769461295
optimizer: sgd
313/313 [==============================] - 2s 5ms/step - loss: 2.716
Running example with
units: 512
dropout_rate: 0.3922780793899139
optimizer: adam
313/313 [==============================] - 1s 5ms/step - loss: 11.42
Running example with
units: 512
dropout_rate: 0.3922780793899139
optimizer: sgd
313/313 [==============================] - 1s 5ms/step - loss: 7.032
Running example with
units: 512
dropout_rate: 0.8173992982088316
optimizer: adam
313/313 [==============================] - 2s 5ms/step - loss: 16.03
Running example with
units: 512
dropout_rate: 0.8173992982088316
optimizer: sgd
313/313 [==============================] - 1s 5ms/step - loss: 2.343
```



Können Sie mir einen Vergleich für alle Konfigurationen zeigen?


4
2
0
−2
−4
0
4
4
2
0
−2
−4
0
50
100


4
4
0
4
2
0
−2
−4
0
50
100

TensorBoard

- Webservice
- Visualisierungstool für
  - Metriken
  - Tensorflow-Berechnungsgraphen
  - Vergleich von Hyperparameter
  - ...

Installation

- Lokal
  - *pip install tensorboard*
- Docker
  - *docker pull tensorflow/tensorflow:latest*
- Standalone:
  - *https://github.com/dmlc/tensorboard*

Einbindung

- Alles ist ein Log!
- Keras
  - Über Callback *TensorBoard*
- Sonst:
  - Erstellung Summary-Writer
  - Verschiedene Funktionalitäten wie
    - *scalar, text, image*

Tensorboard Scalars

- Metriken, Selbstdefinierte Werte
- Keras-Modell
  - Callback
- Sonstige Modelle:
  - Erstellung Summary-Writer
  - *tf.summary.scalar(metric_name, value, step)*

Tensorboard Graph

- Parameter: *write_graph=True* (default)
- Layerweise
- Operationweise
- Keras (konzeptioneller Graph)
- *tf.function*

Hyperparameter Tuning

- Vergleich verschiedener Modellparameter
- Plugin: Hparams
- *hp.KerasCallback*

Weitere Funktionen

- Images
- Text
- Projector
- What if Tool
- Profiling Tool
- ...