

Grundlagen in TensorBoard

Karsten Schreiblehner
Leipzig, 11. Juni 2020

Handout

- Git
 - *git clone*
github.com/kschreiblehner/tensorboard_tutorial.git
- Handout, Installation, Live
 - Ordner: *material*
- Aufgaben und Lösungen
 - Ordner: *practice*

```

accuracy: 0.7418
Epoch 9/50
32/32 [=====] - 2s 49ms/step - loss: 1.0538 - accuracy: 0.8240 - val_loss: 2.9378 - val_
accuracy: 0.7497
Epoch 10/50
32/32 [=====] - 2s 49ms/step - loss: 0.8670 - accuracy: 0.8400 - val_loss: 2.9800 - val_
accuracy: 0.7581
Epoch 11/50
32/32 [=====] - 2s 49ms/step - loss: 0.7464 - accuracy: 0.8530 - val_loss: 3.5558 - val_
accuracy: 0.7413
Epoch 12/50
32/32 [=====] - 2s 48ms/step - loss: 0.6638 - accuracy: 0.8560 - val_loss: 2.9108 - val_
accuracy: 0.7631
Epoch 13/50
32/32 [=====] - 2s 49ms/step - loss: 0.4751 - accuracy: 0.8690 - val_loss: 2.6289 - val_
accuracy: 0.7543
Epoch 14/50
32/32 [=====] - 2s 51ms/step - loss: 0.7479 - accuracy: 0.8370 - val_loss: 2.4456 - val_
accuracy: 0.7607

```

- Speichern der Resultate
- Programm zum Plotten

```

accuracy: 0.7418
Epoch 9/50
32/32 [=====] - 2s 49ms/step - loss: 1.0538 - accuracy: 0.8240 - val_loss: 2.9378 - val_
accuracy: 0.7497
Epoch 10/50
32/32 [=====] - 2s 49ms/step - loss: 0.8670 - accuracy: 0.8400 - val_loss: 2.9800 - val_
accuracy: 0.7581
Epoch 11/50
32/32 [=====] - 2s 49ms/step - loss: 0.7464 - accuracy: 0.8530 - val_loss: 3.5558 - val_
accuracy: 0.7413
Epoch 12/50
32/32 [=====] - 2s 48ms/step - loss: 0.6638 - accuracy: 0.8560 - val_loss: 2.9108 - val_
accuracy: 0.7631
Epoch 13/50
32/32 [=====] - 2s 49ms/step - loss: 0.4751 - accuracy: 0.8690 - val_loss: 2.6289 - val_
accuracy: 0.7543
Epoch 14/50
32/32 [=====] - 2s 51ms/step - loss: 0.7479 - accuracy: 0.8370 - val_loss: 2.4456 - val_
accuracy: 0.7607

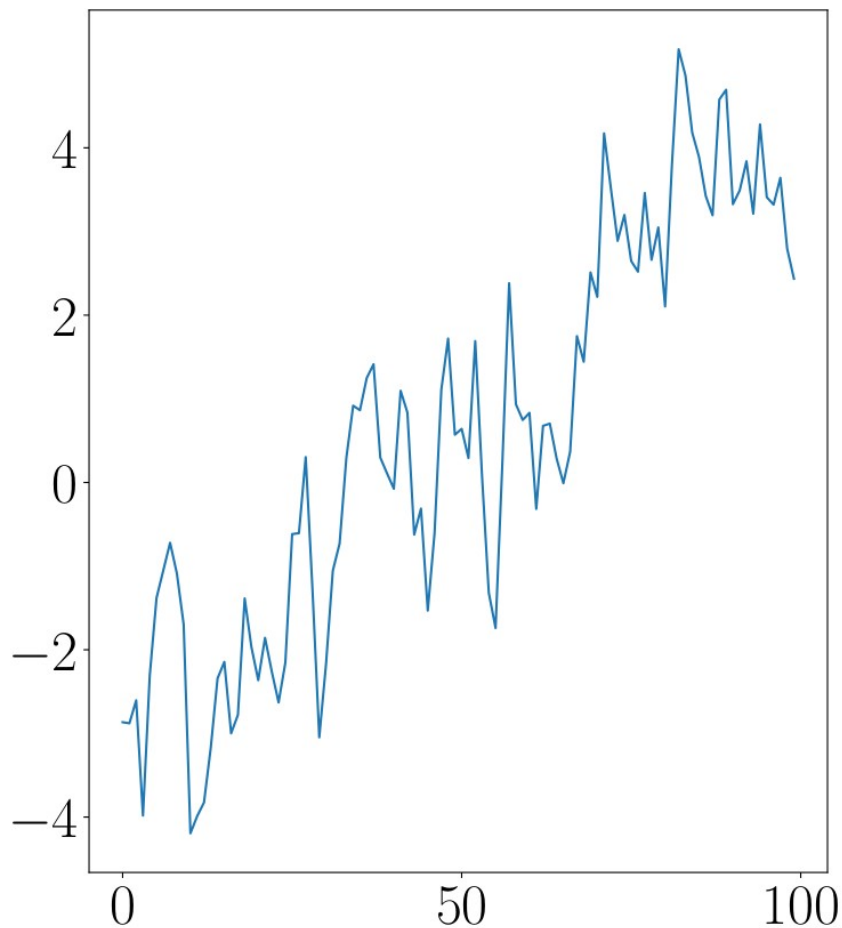
```

- Speichern der Resultate
- Programm zum Plotten

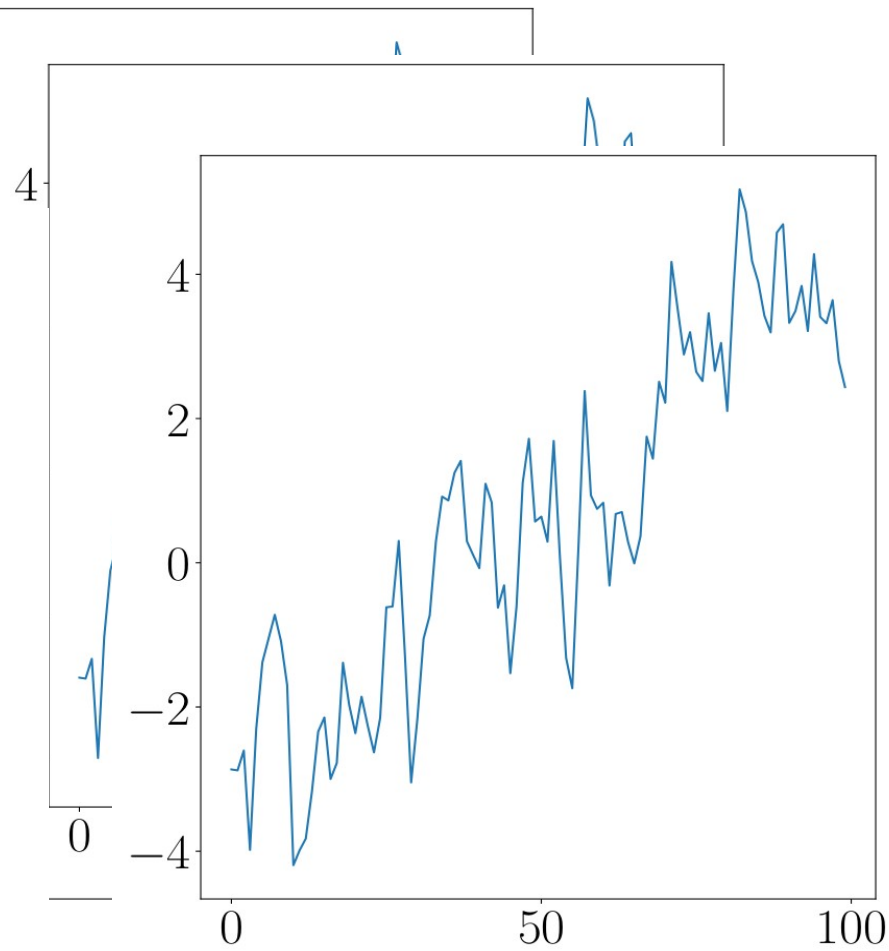
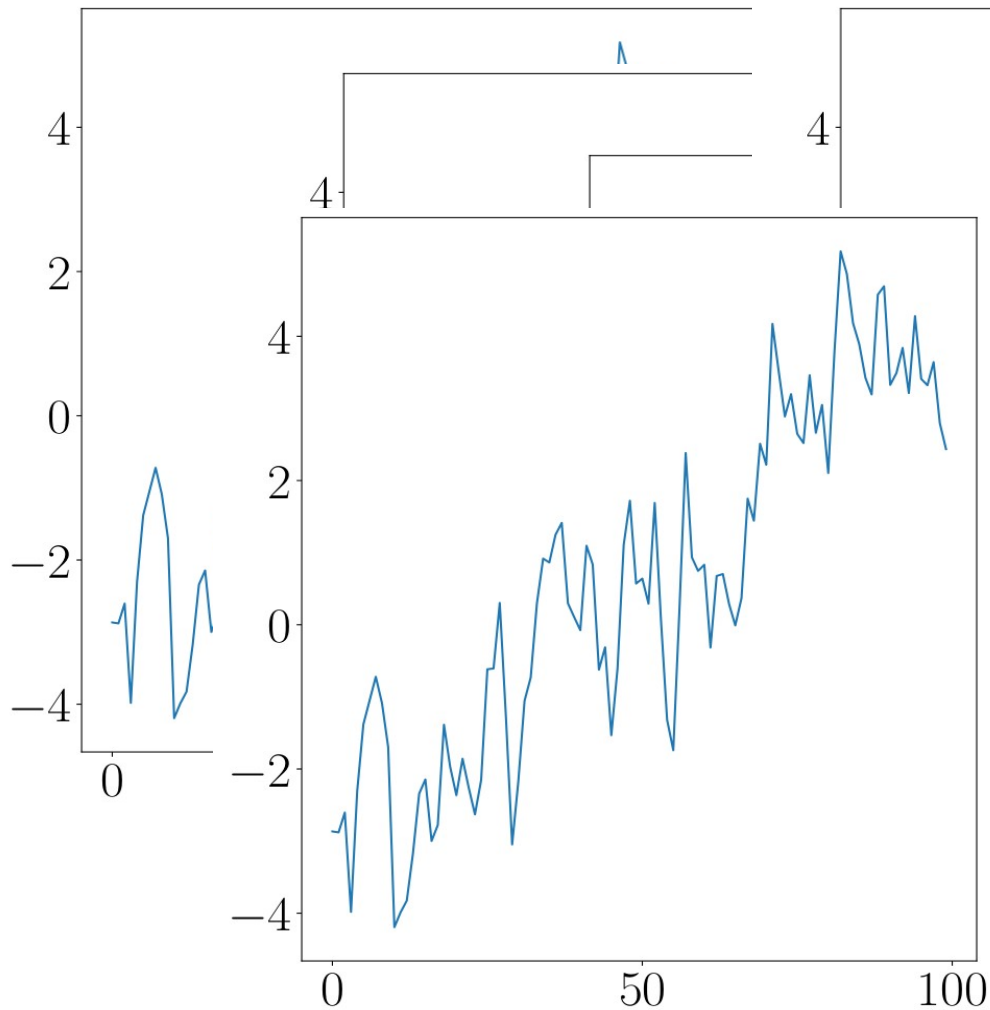
```

Epoch 9/50
32/32 [=====]
accuracy: 0.7418
Epoch 10/50
32/32 [=====]
accuracy: 0.7497
Epoch 11/50
32/32 [=====]
accuracy: 0.7581
Epoch 12/50
32/32 [=====]
accuracy: 0.7413
Epoch 13/50
32/32 [=====]
accuracy: 0.7631
Epoch 14/50
32/32 [=====]
accuracy: 0.7543
Epoch 15/50
32/32 [=====]
accuracy: 0.7607

```



Können Sie mir einen Vergleich für
alle Konfigurationen zeigen?



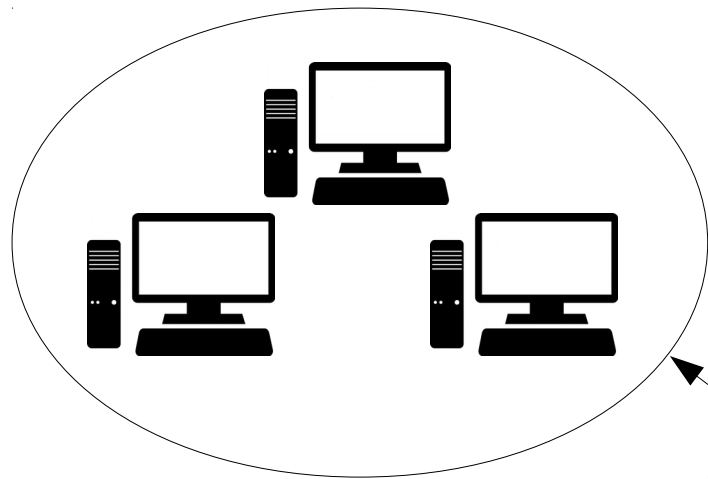
TensorBoard

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

Installation

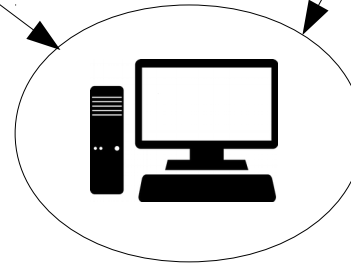
- Lokal
 - *pip install tensorboard*
- Docker
 - *docker pull tensorflow/tensorflow:latest*
- Standalone
 - *<https://github.com/dmlc/tensorboard>*

Installation



TensorFlow

- Eigenständiges Cluster
- Ausführung Berechnungen
- Generierung Log-Dateien

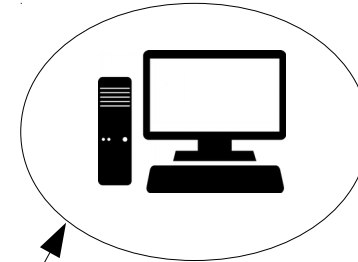


Host

- Fungiert als Master
- Speichert Log-Dateien
- Darstellung

TensorBoard

- Fungiert als Slave
- Webservice
- Auswertung Log-Dateien



Starten der Oberfläche

- Jupyter-Notebook
 - Laden der Extension *%load_ext tensorboard*
 - *%tensorboard --logdir logdir --host host --port port*
- Konsole
 - *tensorboard --logdir logdir --host host --port port*

Einbindung

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

TensorBoard Scalars

- Metriken, selbstdefinierte Werte
- Keras-Modell
 - *Callback*
- Sonst
 - *tf.summary.scalar(metric_name, value, step)*

TensorBoard Graph

- Callback-Parameter *write_graph=True* (default)
- Ansichten
 - Layer
 - Operation
 - Konzeptioneller Graph (Keras)
 - *tf.function*

Hyperparameter Tuning

- Noch in Vorschau-Phase!
- Modellvergleich
- Hparams Plugin
- Keras *hp.KerasCallback*
- Sonst Summary-Writer

Weitere Funktionen

- Images
- Text
- Projector
- What-If Tool
- Profiling Tool
- ... viele mehr in der Entwicklung