

TensorFlow RNN + TensorBoard

Jan Popko

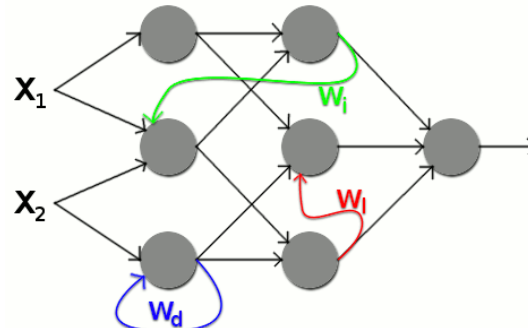
Python Advanced

Recurrent Neuronale Netze

- rückgekoppelte neuronale Netze
- Anwendung bei Problemstellungen, die das Verarbeiten von Sequenzen erfordern

Mögliche Rückkopplungen:

- direkte Rückkopplung (w_d):
 - der eigene Ausgang eines Neurons wird als weiterer Eingang genutzt.
- indirekte Rückkopplung (w_i):
 - Ausgang eines Neurons wird mit einem Neuron der vorhergehenden Schichten verbunden
- seitliche Rückkopplung (w_l):
 - Ausgang eines Neurons wird mit einem anderen Neuron derselben Schicht verbunden
- vollständige Verbindung:
 - jeder Neuronenausgang hat eine Verbindung zu jedem anderen Neuron



Jan Popko

Python Advanced

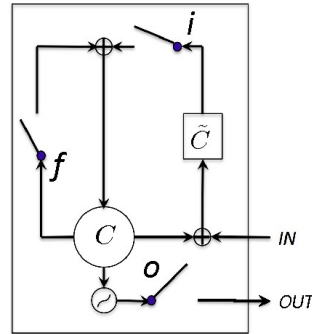
LSTM + GRU

Long Short Term Memory Neuron (LSTM):

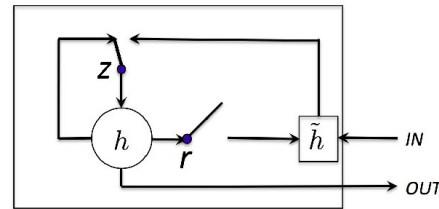
- Neuron mit einer self-recurrent Verbindung
- Input Gate: steuert ob ein neuer Wert in die Zelle fließt
- Forget Gate: steuert ob ein Wert in der Zelle bleibt/vergessen wird
- Output Gate: steuert ob der Wert in der Zelle zur Berechnung für das nächste Neuron verwendet wird

Gated Recurrent Units (GRU):

- GRU ist eine Alternative zum LSTM
- wird meist in der Musikmodellierung genutzt
- verbindet forget und input Gate zu update Gate
- weniger Komplex (theoretisch etwas schneller aber ungenauer)



(a) Long Short Term Memory Unit.
 i : Input Gate, f : Forget Gate, o : Output Gate.



(b) Gated Recurrent Unit.
 z : update Gate, r : Reset Gate.

Jan Popko

Python Advanced

RNN Layers

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, CuDNNLSTM, GRU, CuDNNGRU
```

SimpleRNN:

- bekommt eine Serie (Array)
- return_sequences: wenn True, gibt Sequenz aller Outputs, sonst nur den Output
 - muß True sein, wenn RNN/LSTM/GRU Layer folgt

LSTM:

- ähnlich SimpleRNN

CuDNNLSTM:

- LSTM Layer speziell für GPU (sonst gleich)

GRU:

- ähnlich SimpleRNN

CuDNNGRU:

- GRU Layer speziell für GPU (sonst gleich)

ConvLSTM2D:

- zum verarbeiten von Sequenzen von Bildern (Videos)

Jan Popko

Python Advanced

TensorFlow GPU

- TensorFlow hat zwei Versionen CPU und GPU
- GPU Berechnungen sind um ein vielfaches schneller

CPU install:

```
pip install tensorflow
```

GPU install:

```
pip install tensorflow-gpu
```

- CUDA Toolkit: <https://developer.nvidia.com/cuda-downloads>
 - installieren
- cuDNN: <https://developer.nvidia.com/rdp/cudnn-download>
 - entpacken und Dateien in cuda Ordner verschieben

Jan Popko

Python Advanced

TensorBoard

erstellt die Logdaten im Ordner "logspath/Name/":

```
NAME = "Name"
```

```
tensorboard = TensorBoard(log_dir = f"logspath/{NAME}")
```

Daten werden als Liste in der `.fit()` Methode über `callbacks` gegeben:

```
model.fit(X, y, .. callbacks = [tensorboard])
```

Aufruf im Terminal:

```
tensorboard --logdir=logspath/
```