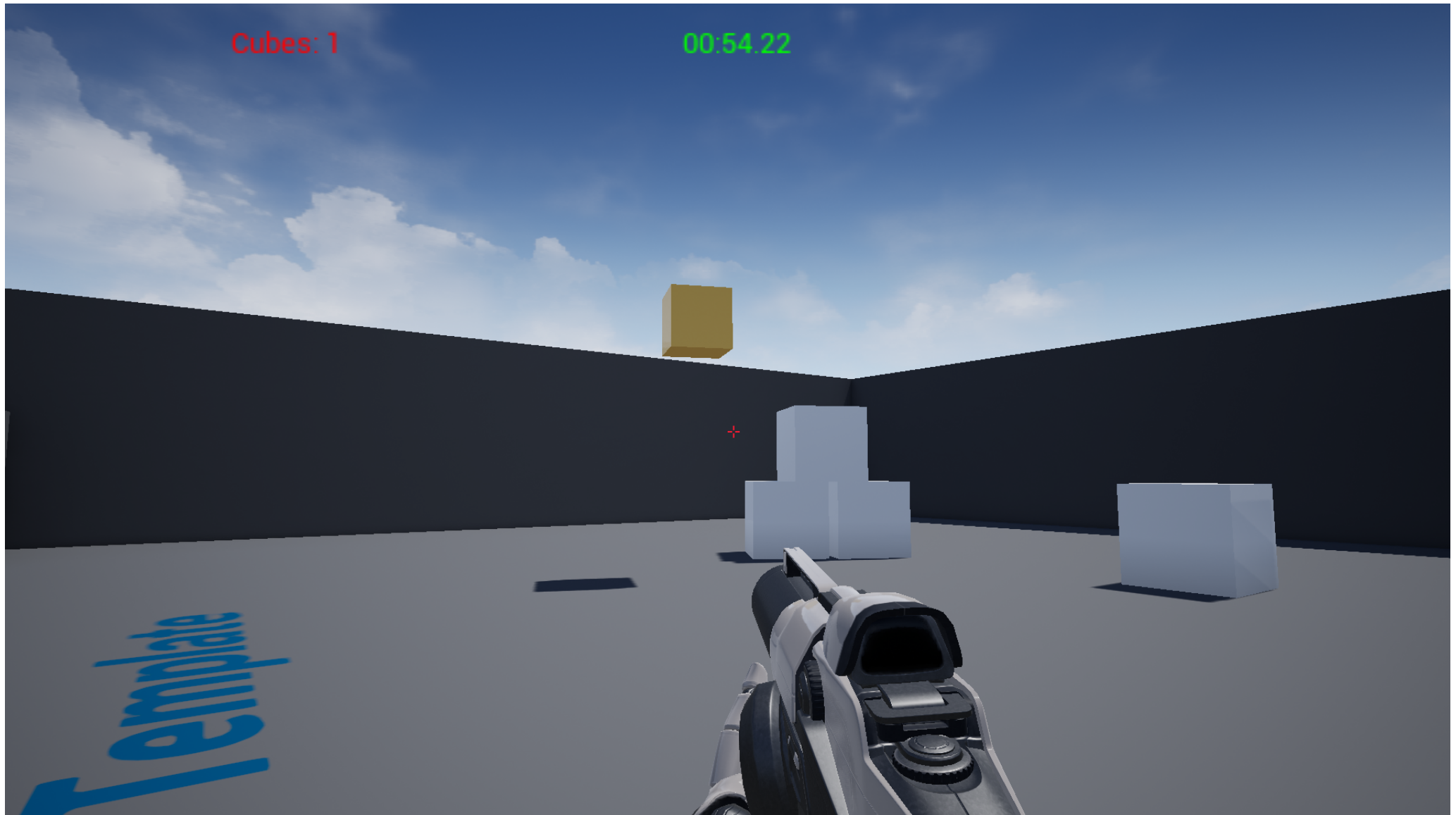


Unreal Tensor

Projektaufgabe Mustererkennung Gruppe – Team #3

Dennis Renz
Raphael Drechsler
David Neumair

Die Idee...



Die Handschrift der Spieler

→ Welche Informationen abtasten?

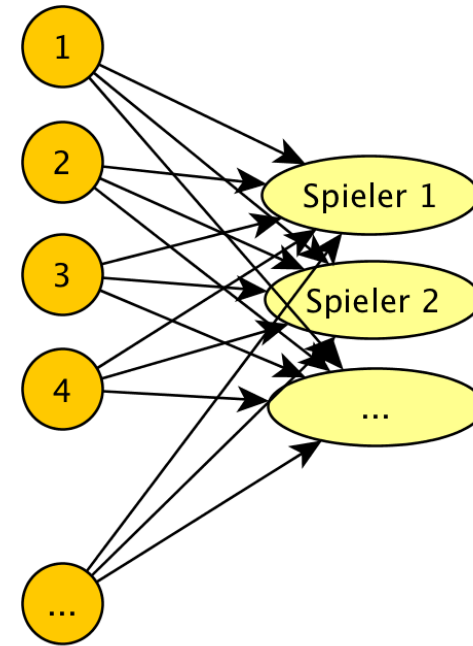
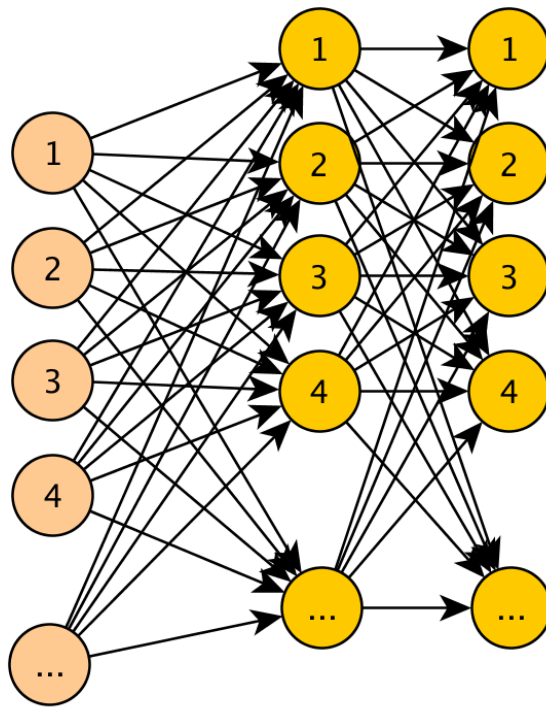
- Mausposition bzw. Mausbewegung
- Klick-Information
- Abstand zu den Zielen
- Zeit-Informationen

→ Einfacher Ansatz:

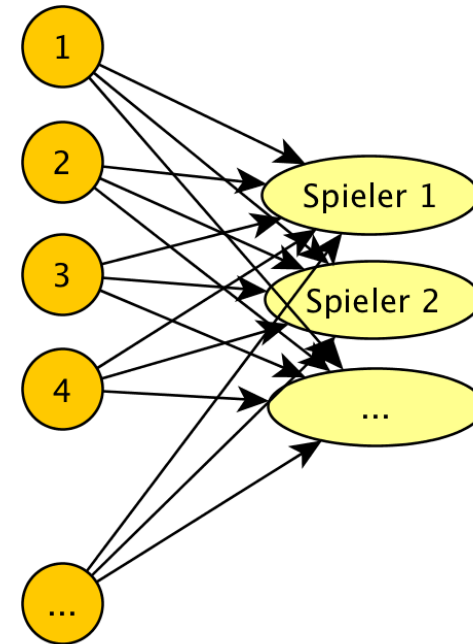
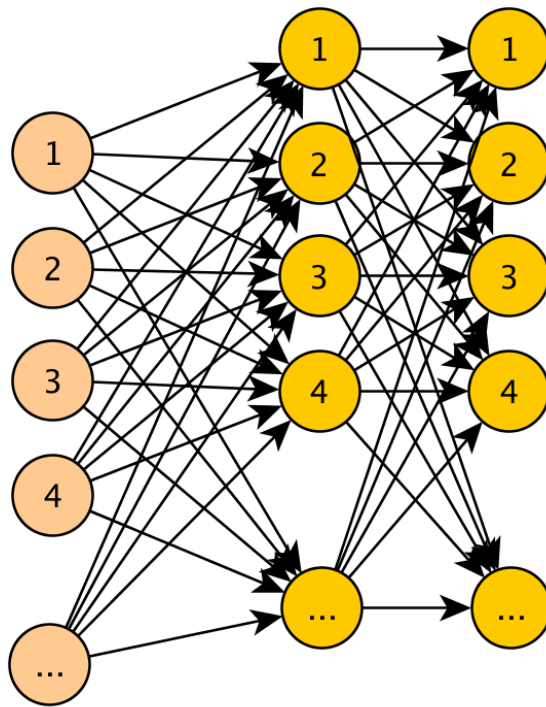
Mausbewegung als Delta pro Frame

Das Netz

Das Netz

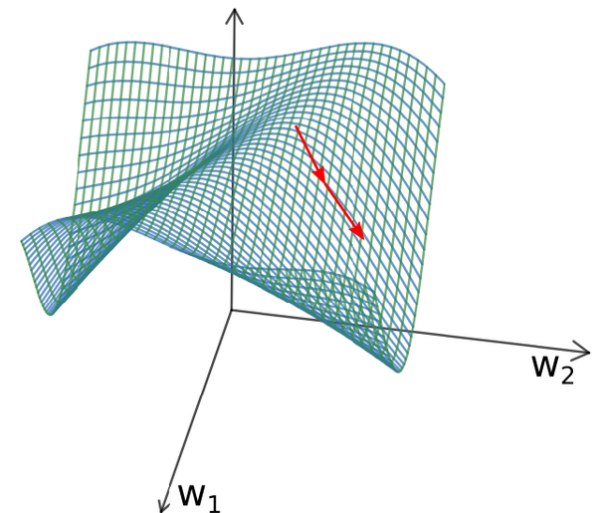


Das Netz



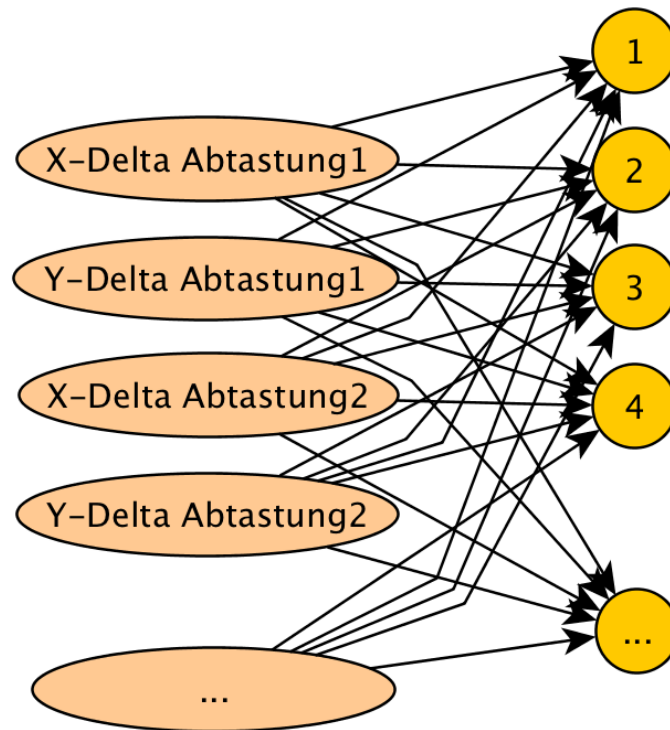
→ Variable Parameter des Netzes

- Genutzte Loss-Funktion
- Genutzte Optimierungs-Funktion
- Aktivierungsfunktionen der Neuronen der Schichten
- Anzahl der Schichten
- Anzahl der Neuronen pro Schicht



Input-Größe des Netzes

→ Zerschneiden der abgetasteten Maus-Deltas



→ Zusätzliche Variable: Größe der Input-Tensoren

Input CSVs aufbereiten

Input CSVs aufbereiten

→ Durchführen von mehreren 60s Runs

```
0.0 : 0.0  
0.0 : 0.0  
0.0 : 0.0  
0.0 : 0.0  
-0.944906 : 0.139986  
-2.05408 : 0.581344  
-3.234226 : 0.828277  
-4.124402 : 0.281791  
-4.496261 : -0.19895  
-4.041295 : -0.68702
```

ca. 2600 Zeilen

Input CSVs aufbereiten

→ Durchführen von mehreren 60s Runs

```
0.0 : 0.0  
0.0 : 0.0  
0.0 : 0.0  
0.0 : 0.0  
-0.944906 : 0.139986  
-2.05408 : 0.581344  
-3.234226 : 0.828277  
-4.124402 : 0.281791  
-4.496261 : -0.19895  
-4.041295 : -0.68702
```

ca. 2600 Zeilen

→ Zuschneiden auf Input-Tensor-Größe und Label pro Spieler anfügen

```
0.0, 0.0, -0.210898, -0.2530245, -0.69993, ... , -1.599471, -0.2412782, -0.047125, 0  
...  
-0.260842, -1.223579, -0.186589, -0.074636,..., -0.139986, -0.069993, 0.0, 1  
...  
0.073833, -0.034997, 0.07453, -0.07453, ... , -0.06729, 0.0, 0.0, 2  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -2.174874, ... , -1.94709, 0.0, 0.0, 2  
0.0, 0.0, -1.210898, -0.330245, -0.69993, ... , -1.699471, -0.242782, -0.037125, 2  
-0.260842, -0.223579, -0.186589, -0.074636,..., -0.139986, -0.069993, 0.0, 2
```

Input CSVs aufbereiten

→ Durchführen von mehreren 60s Runs

```
0.0 : 0.0
0.0 : 0.0
0.0 : 0.0
0.0 : 0.0
-0.944906 : 0.139986
-2.05408 : 0.581344
-3.234226 : 0.828277
-4.124402 : 0.281791
-4.496261 : -0.19895
-4.041295 : -0.68702
```

ca. 2600 Zeilen

→ Zuschneiden auf Input-Tensor-Größe und Label pro Spieler anfügen

```
0.0, 0.0, -0.210898, -0.2530245, -0.69993, ... , -1.599471, -0.2412782, -0.047125, 0
...
-0.260842, -1.223579, -0.186589, -0.074636,..., -0.139986, -0.069993, 0.0, 1
...
0.073833, -0.034997, 0.07453, -0.07453, ... , -0.06729, 0.0, 0.0, 2
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -2.174874, ... , -1.94709, 0.0, 0.0, 2
0.0, 0.0, -1.210898, -0.330245, -0.69993, ... , -1.699471, -0.242782, -0.037125, 2
-0.260842, -0.223579, -0.186589, -0.074636,..., -0.139986, -0.069993, 0.0, 2
```

→ Run 1..4 Training-Daten und Run 5..6 Test-Daten

Ergebnisse:

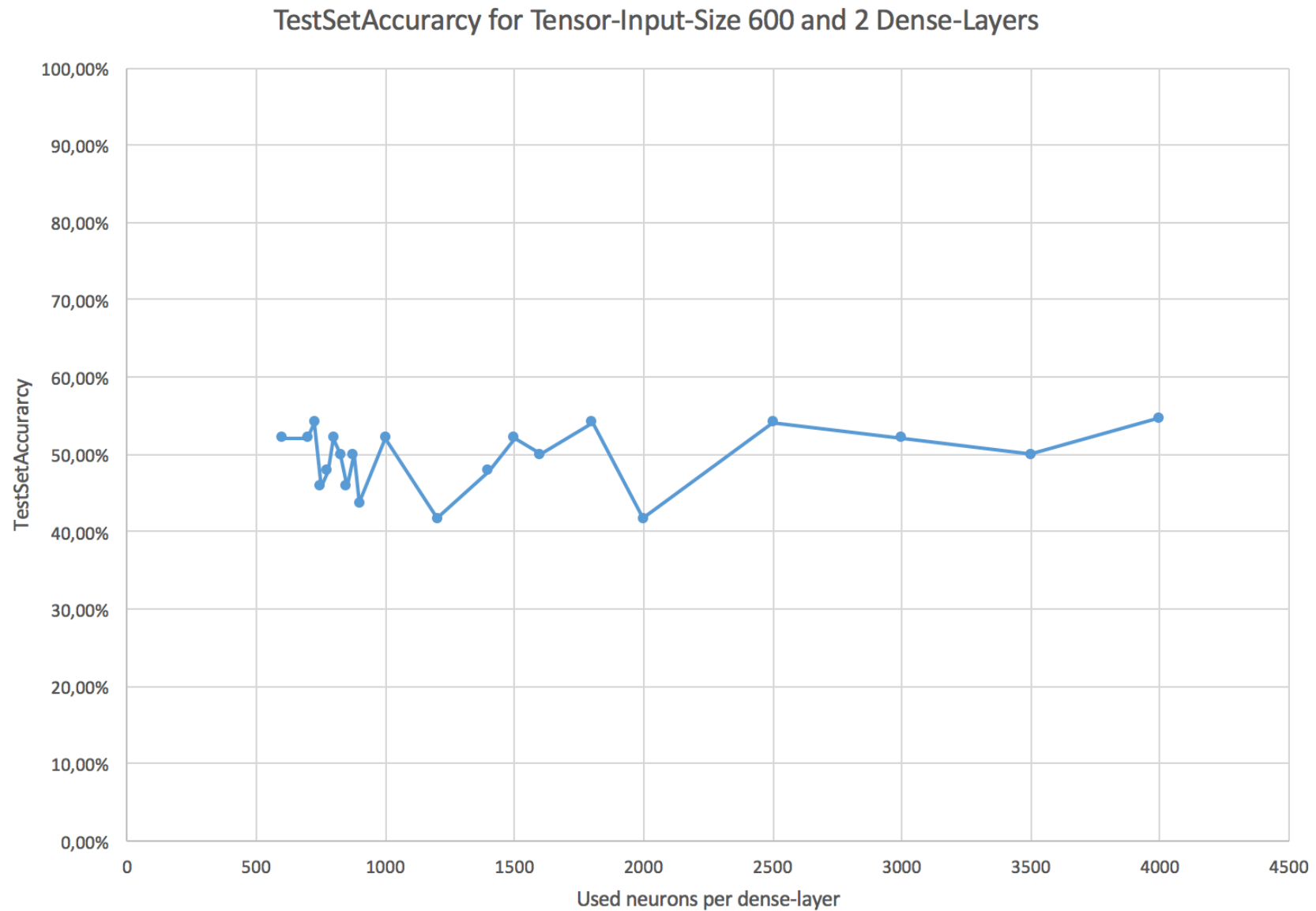
Ergebnisse: < 60% Testset-Accuracy

```
python3 train_and_test.py training_300.csv test_300.csv 700 700 300  
>Epoch 200: Loss: 0.579, Accuracy: 98.438%  
>Test set accuracy: 52.083%
```

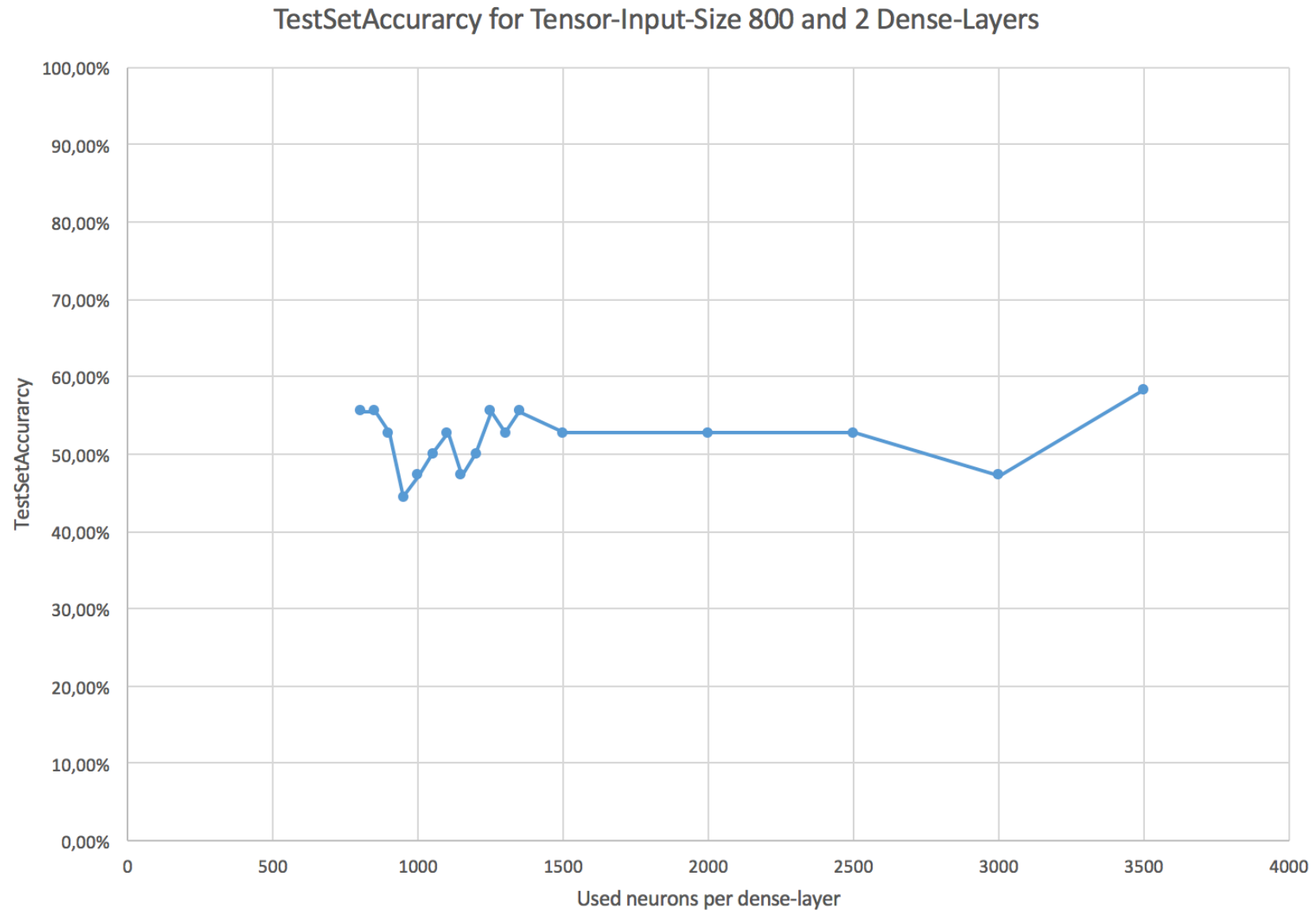
```
python3 train_and_test.py training_300.csv test_300.csv 725 725 300  
>Epoch 200: Loss: 0.566, Accuracy: 100.000%  
>Test set accuracy: 54.167%
```

```
python3 train_and_test.py training_300.csv test_300.csv 750 750 300  
>Epoch 200: Loss: 0.571, Accuracy: 99.219%  
>Test set accuracy: 45.833%
```

Ergebnisse < 60%



Ergebnisse < 60%



Ergebnisse < 60%

=> Problem gefunden!

Let's play!

Mögliche Fehlerquellen

- Abtastrate und Genauigkeit der Maus
- FrameRate
- Boxen spawnen zufällig verteilt
 - Mehr Daten!

Fazit

Fragen?