

# Blackstories: ein Computer lernt Fragen zu beantworten

```
# tokenize dataset
```

```
def preprocess(batch):
```

## Blackstories

Ein Spieler darf zum gestellten Rätsel Fragen stellen, die vom Spiel-Partner nur mit «Ja» oder «Nein» beantwortet werden.

Meine zentrale Aufgabestellung  
**Kann ich den Spiel-Partner durch einen Computer ersetzen? Wie?**



## Lösungsansatz: Neuronales Netz (NN), genauer Deep Learning Natural Language Processing (NLP)

Konkrete Aufgabe für das NN: Entscheidung zwischen den zwei Möglichkeiten «Ja» und «Nein».

Speziell bei NLP ist: der Computer muss zuallererst Worte und Sätze in vom NN nutzbare Token IDs umwandelt. Dies ist die Aufgabe vom sog. Tokenizer:

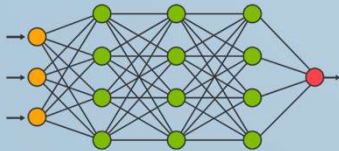
Eingabe: "This is an example sentence."



Tokens: [<s>, this, is, an, example, sentence, ., </s>]



Token IDs: [0, 713, 16, 41, 1246, 3645, 4, 2]

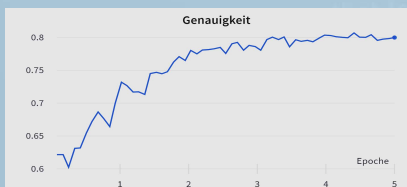


## NN-Architektur

Im Verlauf meiner Arbeit habe ich verschiedene Netz-Architekturen (verschiedene Modellgrößen und Hyperparameter) getestet, trainiert und optimiert. Insgesamt habe ich über 100 Trainings durchgeführt mit mehr als 80 Stunden Nutzungszeit von Cloud GPUs. Die von mir entwickelten Programme musste ich zum Zweck der Cloud-Nutzung entsprechend optimieren.

## Resultate meines NN-Trainings

Von mir erreichte Genauigkeit: ca. 80% (= 0.8 in der Grafik). Zum Vergleich: Mensch erreicht 90%, weltweit beste KI 92%.



Eine Epoche = ein Datensatz-Durchlauf à 12'000 Beispielen.  
D.h.: Bei 5 Epochen wurden 60'000 Beispielen durchgerechnet.

Luzian Trapletti, Maturaarbeit 2022

Kantonsschule Uster, Informatik

Betreuung: Theresa Luternauer  
und Dr. Gabriele Neyer

## Überlegungen und Hintergrund

Der Computer muss also «denken» können: er muss den Zusammenhang zwischen einer Frage und gegebener Textpassage (Rätsel-Lösung) finden. Eine Aufgabe, der nur ein Neuronales Netz gewachsen ist!

Ein NN besteht aus sehr vielen Rechenoperationen (Neuronen), die miteinander vernetzt sind und eingespeiste Informationen weiterreichen und verarbeiten, so dass schlussendlich ein Ergebnis produziert wird. Die Güte der Lösung kann beeinflusst werden durch Training des NN mit zigtausenden von Beispielen. Gleich wie beim Menschen.

## Training des NN

Anhand eines Trainingsdatensatzes, der das Schema von Ja-/Nein-Fragen hat, muss das verwendete NN zunächst trainiert werden. Ich habe hierzu BoolQ verwendet.

BoolQ beinhaltet 12'000 Fragen mit jeweiliger Textpassage und passender Antwort:

Frage	Is a wolverine the same as a badger?
Textpassage	Badgers are short-legged omnivores in the family Mustelidae, ... They belong to the caniform suborder of carnivoran mammals ...
Antwort	No

## Was passiert beim Training?

Das NN versucht vorherzusagen, ob die jeweilige Frage mit Ja oder Nein beantwortet werden muss. Dies geschieht auf der Basis von Frage und Textpassage. Die Parameter, Weights und Biases der einzelnen Neuronen, des NN werden anschliessend leicht verändert, um die richtige Antwort bei der nächsten Frage wahrscheinlicher zu machen (Gradientenabstiegsverfahren). Je mehr Beispiele so durchgerechnet wurden, desto besser kann das NN die richtige Lösung bei neuen Fragen, die es zuvor noch nie gesehen hat, vorhersagen.

Denn: Wenn ihr auf meiner Webseite eigene Fragen stellt, kennt das NN diese Frage selbstverständlich vorher nicht, kann durch dieses Training aber dennoch mit hoher Wahrscheinlichkeit die richtige Antwort geben.

## Ergebnis

Mein Chat-Bot als Spielpartner für die Blackstories:  
<https://boolq.trapletti.org>

Zur Implementierung habe ich folgende Technologien verwendet:

- Python
- PyTorch
- Dash
- Huggingface
- Wandb
- Google Cloud Server
- Vultr Server Hosting

