

---

## Übung 5

---

### HYPERPARAMETERS:

Wir haben für das Training folgende Hyperparameter gesetzt:

- SOURCE\_VOCAB\_SIZE = 40000  
TARGET\_VOCAB\_SIZE = 40000
- EMBEDDING\_SIZE = 256  
HIDDEN\_SIZE = 512
- Epochs = 6

Hierbei wurde vor allem reduziert, da es sonst zeitlich sehr knapp geworden wäre. Wir haben ein Training gestartet, dieses hätte jedoch nur schon für eine Epoche 1.5 Tage gedauert. Es wäre wahrscheinlich aber besser geworden (SOURCE\_VOCAB\_SIZE = 75000, TARGET\_VOCAB\_SIZE = 75000, EMBEDDING\_SIZE = 1048, HIDDEN\_SIZE = 2048).

### CODE-VERÄNDERUNGEN

Wir haben folgende Veränderungen eingebaut:

- Early-stopping
- Input-inverse
- Dropout (rate=0.3)

**Early-stopping:** Immer am Ende der Epoche wird verglichen, ob die Perplexity grösser wurde als in der vorherigen Epoche. Wenn ja wird ein Counter um 1 erhöht. Wenn nicht, wird der Counter um 1 reduziert, bis er wieder auf 0 ist. So versuchen wir zu verhindern, dass wir nicht zu früh stoppen, falls es zuerst schlechter und dann doch wieder besser wird. Es muss also effektiv 2x hintereinander schlechter werden, damit das Programm terminiert.

**Input-inverse:** In reader.py werden nach dem einlesen der Dateien die ID's alle rückwärts sortiert, also auch umgekehrt zurückgegeben. So sollte das RNN sich besser an das erste Wort erinnern können.

**Dropout:** Dropout sorgt dafür, dass immer wieder ein Neuron auf 0 gesetzt wird. Diese Methode verhindert Overfitting. Wir haben eine Rate von 0.3 verwendet. Das heisst, dass eine 30 prozentige Chance besteht, dass ein Neuron auf 0 gesetzt wird.

Bleu-Score: 19.888

Wir haben noch ein zweites Modell trainiert mit: dropout 0.5, learning-rate 0.001 anstatt 0.0001. Dabei gab es einen Bleu-Score von 20.212. Wenn wir dieses noch länger trainiert hätten könnte ich mir vorstellen, dass dieser Wert noch mehr gestiegen wäre. Dies liegt wahrscheinlich an der hohen Dropoutrate, denn diese verhindert Overfitting.

### PROBLEME

Probleme hatten wir eigentlich nur beim Starten des Trainings, dort fehlte ein Ordner und das bereitete uns Probleme bis wir es gefunden haben. Ausserdem mussten wir auf dem Windows-Laptop `export CUDA_VISIBLE_DEVICES=xx` schreiben, damit es funktionierte... Jedoch war alles (auch die Code-Veränderungen) irgendwie lösbar dank Google und nachfragen.

Das Early-stopping hat nichts gebracht, nach 6 Epochen wurde das Ergebnis immer noch besser.