MT Übung 4

Nicolas Spring

May 3, 2018

1 Dateien

- Die Preprocessing-Skripte können im Ordner ***my_preprocessing*** gefunden werden.
- Das Datenset (inklusive der Zwischenschritte) befindet sich im Ordner ***my_dataset***.
- Der Bericht, das Modell-Schema, der generierte Text und der Screenshot der Loss-Übersicht können im Ordner ***Abgabe*** gefunden werden.

2 Datenset und Preprocessing

2.1 Datenset

Als Datenset habe ich Leo Tolstois *Krieg und Frieden* in der Englischen Übersetzung verwendet, dies einerseits, weil mir das Buch sehr gut gefällt, und andererseits, weil ich mir erhoffte, im generierten Text eventuell den einen oder anderen Namen zu finden und so zu sehen, wie das RNN damit umgeht. Auf Grund des grossen Umfangs des Buches, der vereinzelten Französischen Textstellen und der vielfältigen Sprache arbeitete ich mit einer Vokabulargrösse von 50'000, um möglichst viel davon mitzunehmen. In dieser Hinsicht war ich mit dem Sampling-Text sehr zufrieden. Er enthielt viele Personen- und Eigennamen und zumindest subjektiv auch eine recht vielfältige Sprache.

2.2 Preprocessing

Als ich das Buch von Project Gutenberg heruntergeladen hatte, enthielt es noch massenhaft newlines, wahrscheinlich wegen OCR. Glücklicherweise hatte ich für eine frühere PCL-II Übung ein Skirpt geschrieben, das genau für die Restrukturierung solcher Textdateien gedacht ist (*dehyphenate.py*). Dieses konnte ich mit minimalen Änderungen wiederverwenden und so die unnötigen newlines loswerden. In einem zweiten Schritt musste ich nur noch die Sätze und Wörter tokenisieren, um den Text in die im README beschriebene word-level Input Form zu bringen. Hierzu verwendete ich NLTK. Siehe *preprocessing.py*.
Die Datei *build_sets.sh* enhät die Kommandos aus der Aufgabenstellung zur schnellen Aufteilung des Datensets.

3 Hyperparameter beim Training

Ich passte folgende zwei Hyperparameter an, der Rest entsprach den Standardwerten:

- Vokabulargrösse: 50'000
  Wie oben bereits angesprochen erhoffte ich mir durch die Erhöhung der Vokabulargrösse eine Verbesserung der Sprachvielfalt im generierten Text.
- Epochen: 20
  Ich erhöhte die Anzahl Epochen auf 20, um durch längeres Training den zusätzlichen LSTM-Zellen, die ich einbaute gerecht zu werden.

4 Code-Veränderungen

In einem Tensorflow-Tutorial stiess ich auf die interessante Möglichkeit, LSTM-Zellen zu stapeln. Dies wollte ich ausprobieren, denn ich erhoffte mir durch die zusätzlichen Zellen, die potentiell angepasst werden können, eine tiefere Perplexität. Ich entschied mich, drei LSTMs zu stapeln. Die Überlegung war, dass man durch das Stapeln mehrerer LSTMs ein viel mächtigeres Netz bauen kann, als das mit nur einem LSTM möglich wäre.
Da dies codemässig eine eher kleine Veränderung am RNN war, hatte ich, sobald ich wusste, wie dies umzusetzen war, nicht wirklich Probleme bei der Implementation. Hier war die TensorFlow-Dokumentation sehr hilfreich.

5 Perplexität

Beim Scoring auf dem Dev-Set mit meinem Modell erreichte ich eine Perplexität von **201.22**.