



Software-Projektpraktikum Maschinelle Übersetzung

3. Übung

Thema:

Im Rahmen dieses Aufgabenblattes werden Sie ein Übersetzungsmodell implementieren. Das Modell, sein Aufbau und Training bilden den Kern eines automatischen Übersetzungssystems. Das in der Vorlesung vorgestellte Feedforward Modell nimmt einige starke Vereinfachungen an, führt aber zahlreiche Methoden ein, die auch in fortgeschrittene Architekturen verwendet werden.

Anmerkungen:

Für eine effiziente Nutzung sollten alle Ihre Programme in der Lage sein sinnvolle Modellteile (wie ein Subwort-Zerlegungsmodell oder ein Vokabular) zu speichern und zu laden. Es ist nicht notwendig die Batches, wie auf dem letzten Aufgabenblatt geschehen, explizit zu speichern. Diese können im Training on-the-fly berechnet werden. Zentrale Hyperparameter sollen nicht im Programmcode spezifiziert sein, sondern vom Nutzer bei Bedarf festgelegt werden können (z.B. per Befehlszeilenargument oder einer Konfigurations-Datei).

Arbeiten Sie auf folgenden Daten, die auf der Seminarwebseite zu finden sind:

- Trainingsdaten: `multi30k.de.gz` und `multi30k.en.gz`
- Dev-/Entwicklungsdaten: `multi30k.dev.de` und `multi30k.dev.en`

Aufgabe:

1. Schreiben Sie ein Programm, welches das Feedforward Übersetzungsmodell aus der Vorlesung implementiert. Ihr Programm sollte in der Lage sein

- ein Training mittels SGD (bzw. einer Erweiterung davon), inklusive Batches aus ausgewählten Trainingsdaten durchzuführen.
- Modelle zu speichern und zu laden (zu unterschiedlichen Checkpoints nach Userauswahl), auch mehrere Speicherpunkte pro Epoche sollen möglich sein.
- Metriken wie Accuracy und Perplexity in regelmäßigen Abständen auszugeben.
- auf einem gegebenen Entwicklungsdaten alle n Updates das Modell auszuwerten.
- vor Beginn des Trainings die Gesamtzahl der trainierbaren Parameter sowie die verwendeten shapes von allen verwendeten `tf.Tensor` Objekten (also die Dimensionalität deren Ausgabe) zusammen mit dem entsprechenden Namen auszugeben und damit die Netzwerkstruktur deutlich machen.

- auf einem Rechner ohne GPU lauffähig zu sein.
- optional beim Start auswählbar: Die Lernrate halbieren, wenn die Performance des Modells auf den Entwicklungsdaten stagniert.

Verwenden Sie in Ihrer Implementierung das Python Interface von tensorflow. Nutzen Sie dabei grundlegende Bausteine wie beispielsweise die in der Vorlesung vorgestellten Methoden.

Neben diesem Grundmodell können Sie gerne selbstgewählte Verbesserungen am Modell implementieren.

2. Testen Sie Hyperparameter auf den genannten Trainingsdaten. Verwenden sie dazu BPE mit 7k Zusammenzugoperationen. Was ist der niedrigste Perplexity Wert den ihr Modell auf dem Entwicklungsdaten erreicht? Variieren Sie die in der Vorlesung vorgeschlagenen Hyperparameter um ein möglichst starkes System zu trainieren.

Abnahmetermin: Montag, 31. Mai, Uhrzeit nach Absprache

Schriftliche Ausarbeitungen werden nicht verlangt. Schicken Sie bitte die commit-ID Ihrer Abgabe (kommentierten Quelltexte und die kurzen Ausgabewerte der geforderten Ergebnisse) bis Sonntag, (30. Mai, 23:59 Uhr) an

mtprak21assi@i6.informatik.rwth-aachen.de

Am Montag erläutern Sie uns dann Ihre Lösungen und demonstrieren Ihre Programme.