Eva Bühlmann
05-058-912

MT Übung 4 – RNNs

Kurzbericht

Für die Übung habe ich das Korpus «HappyDB: A Corpus of 100,000 Crowdsourced Happy Moments[1]» verwendet. Es handelt sich dabei um eine Sammlung von rund 100'000 Aussagen von Personen zu ihrem letzten Glücksmoment. Ich habe mich für dieses Set entschieden da ich die Thematik interessant fand und gespannt darauf war, welche Aussagen zum Glück das Sprachmodell erzeugen würde. Gleichzeitig hält sich die sprachliche Komplexität des Textes in Grenzen – grösstenteils handelt es sich um kurze, einfache Sätze - wodurch ich mir bessere Resultate erhoffte, als bei einem komplexen Text. Ausserdem waren die Daten schon relativ gut strukturiert. Da jede Aussage in der Regel auf einer Zeile stand und meist nicht mehr als einen Satz umfasste, habe ich denn auch auf ein Preprocessing verzichtet.

Nach dem ersten Trainingsdurchlauf ergab sich damit eine Perplexität auf dem Dev-Set von 37.56. In einem ersten Schritt wollte ich herausfinden, ob sich ein Preprocessing der Daten doch lohnen könnte. Dazu habe ich die Input-Daten mittels NLTK-Tokenizer sowohl auf Satz- als auch auf Wortebene tokenisiert. Dadurch stand nun überall pro Zeile nur noch ein Satz mit einzelnen Tokens. Folgende Übersicht zeigt die Grösse des Textes vor bzw. nach dem Preprocessing:

	Vor Tokenisierung	Nach Tokenisierung
Anzahl Zeilen	104'436	136'695
Anzahl Tokens	1'840'730	2'021'061

Das Preprocessing führte gegenüber dem ersten Trainingsdurchlauf zu einer klaren Verbesserung des Modells, die Perplexität auf dem Test-Set konnte auf 31.96 reduziert werden.

Da ich besser verstehen wollte, welchen Einfluss einzelne Parameter auf das Resultat haben, habe ich mich entschieden einige Modifikationen an den Hyperparametern vorzunehmen. Dabei habe ich jeweils bewusst nur einzelne Wert verändert und mehrere Trainings durchgeführt, um zu beobachten was dabei passiert. Dazu habe ich die Parameter jeweils im File const.py angepasst. Da dabei jeweils nur eine oder zwei Zahlen verändert wurden, habe ich diese Files nicht im remote repository abgelegt.

Bsp. Code Veränderung.

```
NUM_EPOCHS = 12  # changed from 10 to 12

# num_steps and learning_rate are hardcoded here; at the moment,
# the only way to change them is to edit this file
NUM_STEPS = 100 # truncated backprop length
LEARNING_RATE = 0.001

HIDDEN_SIZE = 1024  # RNN state size
EMBEDDING_SIZE = 400  # embedding vector size # changed from 256 to 400
```

Die Tabelle auf der nächsten Seite gibt eine Übersicht zu den verschiedenen Anpassungen, die ich gemacht habe, den draus resultierenden Resultaten und meinen Überlegungen.

Nach mehreren Versuchen und Durchläufen gelang mir schliesslich mittels einer Vergrösserung der Word-Embeddings (von 256 auf 400) sowie einer leichten Erhöhung der Epochen (12 statt 10) eine leichte Verbesserung des Sprachmodells – die Perplexität auf dem Dev-Set konnte von 31.96 auf 30.72 verringert werden. Diese Verbesserung erscheint mir in Anbetracht der deutlichen Erhöhung der Embedding Size jedoch als relativ minim. Am meisten hat letztlich das Preprocessing gebracht.

Der mit dem letzten Modell generierte neue Text ergibt zwar inhaltlich nicht überall Sinn, wiederspiegelt aber schön den Stil und das Vokabular des Input-Textes. Mein Modell macht z.B. folgendes glücklich:

- *I met my friend today and saw that she was in Australia .*
- *This is my first love , a little bit .*
- *I love being happy and you can definitely have this almost makes me happy .*

[1] https://arxiv.org/abs/1801.07746

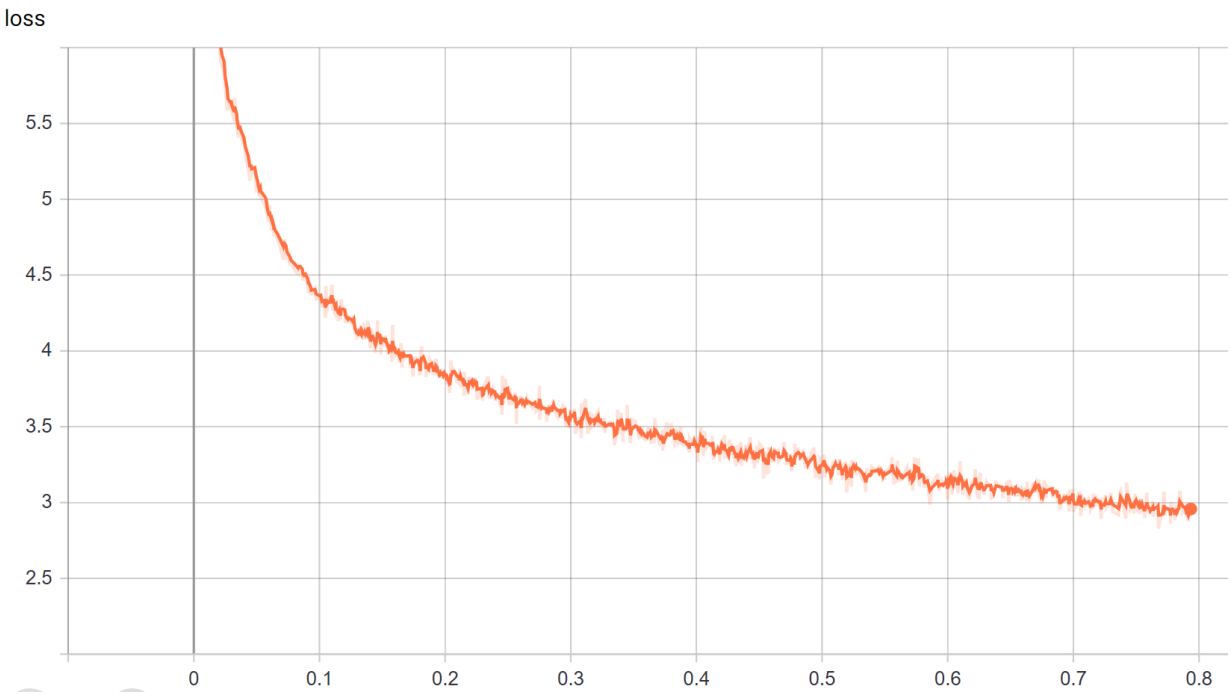
Eva Bühlmann
05-058-912

	Anpassung		Perplexity		Bemerkungen/ Überlegungen
Training	Bereich	Veränderte Parameter (im Vgl. zu Original romanesco Code)	Train-Set	Dev-Set	
1.1	-	-	?[2]	37.56	Erster Durchlauf ohne Preprocessing.
1.2	Tokenisierung	*Satz/Wort Tokenisierung*	21.87	31.96	Deutliche Verbesserung. Für weitere Schritte mit diesen Daten gearbeitet
2.1	Batch-Size	**Batch-Size erhöht:** von 64 auf 128	32.98	37.32	klare Verschlechterung, deshalb versuchen, was Verkleinerung bewirkt.
2.2		**Batch-Size verkleinert:** von 64 auf 32	17.60	31.70	Deutlich kleinere Perplexity im Train-Set, aber nur minimale Verbesserung im Dev-Set. (Overfitting?). Um mit einer Anpassten Batch-Sizes erfolgreich zu sein, müssten wohl gleichzeitig andere Parameter wie z.B. die Anzahl Epochen angepasst werden.
2.3	Embedding-Size und Epochen	**Grösse Word Embeddings: 300** (Original: 256)	23.61	31.61	In nächstem Schritt wollte ich jedoch schauen, was eine verändert Grösse der Word Embeddings bewirkt. Leichte Verbesserung → neue Grösse beibehalten. → versuchen, ob mehr Epochen das Resultat verbessern.
2.4		**Anzahl Epochen: 12** (Original: 10) **Grösse Word Embeddings:** weiterhin 300	19.10	31.18	Durch 2 zusätzliche Epochen nochmal leichte Verbesserung. →Epochenzahl weiter erhöhen.
2.5		**Anzahl Epochen: 15 Grösse Word Embeddings:** weiterhin 300	17.05	31.13	Kaum Unterschied zu 12 Epochen mit derselben Embedding Size. Deshalb nochmal Durchlauf mit 12 Epochen, aber noch grösserer Embedding Size.
2.6		**Anzahl Epochen: 12 Grösse Word Embeddings: 400**	18.31	30.72	Weitere Vergrösserung des Wort Embeddings bringt nochmals leichte Verbesserung. Allerdings fraglich, ob sich der «Aufwand» für die relativ kleine Verbesserung lohnt.

[2] Dieser Wert ging leider verloren.

Eva Bühlmann
05-058-912

Loss Übersicht letztes Training (2.6)



Eva Bühlmann
05-058-912

Gezeichnetes Modell Schema