Ein Vergleich zwischen N-Gram Sprachmodellen und Neuronalen Sprachmodellen

Robin Labryga

August 2020

Inhaltsverzeichnis

1 Einführung **1**

2 Vergleich zwischen N-Gram Sprachmodellen und Neuronalen Sprachmodellen **2**

2.1 Aufwand 2

2.2 Qualität 3

2.3 Verwendung bei Sprachverarbeitung 4

3 Ergebnis und Ausblick **6**

Einführung

Bei Sprachmodellen werden verschiedenen Wortsequenzen Wahrscheinlichkeiten zugeteilt, die dann auf die Wahrscheinlichkeit von Sätzen schließen lassen. Zwei Möglichkeiten diese Wahrscheinlichkeitsverteilungen zu berechnen sind N-Gram Sprachmodelle und Neuronale Sprachmodelle.
In dieser Arbeit werden wir uns mit den Unterscheiden und Gemeinsamkeiten von N-Gram Modellen und Neuronalen Sprachmodellen beschäftigen und dabei die Frage beantworten: "Was sind Vor- und Nach- teile von Neuronalen Sprachmodellen gegenüber N-Gram Modellen bei der Bestimmung von Wortsequenzwahrscheinlichkeiten?".
Um die Leitfrage zu beantworten, schauen wir uns zuerst den Aufwand an, der bei der Bestimmung von Wahrscheinlichkeitsverteilungen auftritt. Dann werden wir uns die Qualität der Wahrscheinlichkeitsverteilungen anschauen und dabei auf das Embedding und die Bestimmung von Satzwahrscheinlichkeiten eingehen. Zum Schluss werden wir dann schaue, inwiefern man die beiden arten von Modellen bei der Verarbeitung gesprochener Sprache einsetzen kann und ein Fazit fassen welches Modell nun wann das bessere ist.

Vergleich zwischen N-Gram Sprachmodellen und Neuronalen Sprachmodellen

2.1 Aufwand

Wir schauen uns zuerst an, wie die Wahrscheinlichkeiten bei einem N-Gram Modell berechnet werden und wie ein Neuronales Netzwerk trainiert wird. Im Anschluss, Vergleichen wir, ob das N-Gram Sprachmodell oder das Neuronale Sprachmodell den größeren Aufwand verursacht.
Bei N-Gram Modellen, schaut man sich einen Korpus von Sätzen an und bestimmt über diesem Korpus Wahrscheinlichkeiten dafür, dass ein bestimmtes Wort auf einen Kontext von $N-1$ Wörtern folgt. Die Wahrscheinlichkeit, dass ein Wort auf einen Kontext folgt, kann als relative Häufigkeit bestimmt werden. Es gilt: $P(Wort|Kontext) = \frac{C(Kontext+Wort)}{C(Kontext)}$, wobei $C(Wortsquenz)$ die Häufigkeit dieser Wortsequenz in dem gegebenen Korpus ist.[JM19, p. 31]
Bei Sprachmodellen auf Basis von Neuronalen Netzwerken (Neuronalen Sprachmodellen), trainiert man ein Neuronales Netzwerk, indem man verschiedene $N-lange$ Wortsequenzen nimmt (wie bei den N-Gram Modellen) und dann den Kontext in dieser Wortsequenz als Eingabe des Neuronalen Netzwerkes verwendet. Die Wahrscheinlichkeitsverteilung, welche das Neuronale Netzwerk als Ausgabe übergibt, vergleicht man dann mit dem Wort, welches tatsächlich auf den eingegebenen Kontext folgt und passt die Einheiten (eng. Unit) des Neuronalen Netzwerkes iterativ an. Dies wiederholt man über dem gesamten Korpus. Am Ende dieser Lernphase, hat man dann ein Neuronales Netzwerk trainiert, welches auf Basis des gegebenen Korpus Wahr-

scheinlichkeiten für ein Folgewort in Abhängigkeit von einem Kontext berechnet.[JM19, p. 129-132 und p. 141]
Wir können feststellen, dass der Aufwand, der benötigt wird, um ein Neuronales Netzwerk zu trainieren, deutlich höher ist, als der, der benötigt wird, um die Wahrscheinlichkeiten eines N-Gram Modells zu berechnen, da man beim Trainieren eines Neuronalen Netzwerkes nicht einfach nur für jedes Wort im Korpus die Wahrscheinlichkeit berechnen muss, nach der es in einem bestimmten Kontext auftritt, sondern jede Kombination einmal durch das Neuronale Netzwerk berechnen lassen muss.
Der Aufwand, der für das finden der Korpora benötigt wird, ist bei N-Gram Sprachmodellen und Neuronalen Sprachmodellen gleich, wobei ein größerer Korpus sowohl bei N-Gram Sprachmodellen, als auch bei Neuronalen Sprachmodellen zu einem besseren Ergebnis führt.

2.2 Qualität

Als nächstes schauen wir uns die resultierenden Wahrscheinlichkeiten an, die ein N-Gram Modell und ein Neuronales Sprachmodell erzeugt. Dazu betrachten wir erst einmal genauer wie ein Neuronales Netzwerk Eingabewörter modelliert, um das Embedding zu verstehen. Daraufhin werden wir uns mit der Bestimmung von Satzwahrscheinlichkeiten beschäftigen, um zu sehen, warum die Wahrscheinlichkeitsverteilung eines Neuronalen Sprachmodells in dieser Hinsicht besser ist als eine Wahrscheinlichkeitsverteilung eines N-Gram Modells.
In einem Neuronalen Netzwerk eines Neuronalen Sprachmodells, werden Eingabewörter in einen Vektor umgewandelt, auf welchen anschließend Berechnungen durchgeführt werden. Diese Umwandlung von Wörtern zu Vektoren, wird während dem Training des Neuronalen Netzwerks iterativ angepasst. Die Umwandlung in einen Vektor, führt dazu, dass Wörter die im Korpus in ähnlichen Kontexten auftreten, im Vektorraum näher beieinander sind, als jene Wörter, die nicht in ähnlichen Kontexten auftreten. Dadurch kommen für Wörter, die in ähnlichen Kontexten auftreten auch ähnliche Wahrscheinlichkeitsverteilungen als Ausgabe des Neuronalen Netzwerkes raus. Diesen Prozess, dass Wörter die in ähnlichen Kontexten auftreten nähere Vektoren erhalten, nennt man Embedding. Das Embedding, führt also dazu, dass wenn zwei Wörter in ähnlichen Kontexten auftreten, aber eines davon im Korpus nie auf einen bestimmten Kontext folgt, es trotzdem eine etwas größere

Wahrscheinlichkeit erhält, dass es auf diesen Korpus folgen könnte[JM19, p. 138-140].
Das Embedding kommt in N-Gram Modellen nicht vor, da Ähnlichkeiten zwischen auftretenden Kontexten in der Berechnung der Wahrscheinlichkeiten keinen Einfluss haben. Wenn ein Wortsequenz im Korpus nicht, beziehungsweise sehr selten vorkommt, hat sie nach dem N-Gram Modell, eine geringe Wahrscheinlichkeit.
Mit den Wahrscheinlichkeiten, die ein N-Gram Sprachmodell und ein Neuronales Sprachmodell berechnet, kann man ganzen Sätzen Wahrscheinlichkeiten zuteilen, indem man die Wahrscheinlichkeit eines Satzes $P(Satz) = P(w_1^n)$, als Produkt der Wahrscheinlichkeiten der Wörter des Satzes darstellt. Es gilt also: $P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_1^{k-1})$[JM19, p. 33]. Wie man an dieser Formel sieht, ist die Wahrscheinlichkeit eines Satzes Null, wenn nur ein einziges Wort im Kontext des Satzes die Wahrscheinlichkeit Null hat, selbst wenn alle anderen Wörter eine sehr hohe Wahrscheinlichkeit haben. Bei N-Gram Modellen, kommen Wahrscheinlichkeiten vom Wert Null vor, sobald ein Wort im Korpus kein einziges mal in einem bestimmten Kontext auftritt. Die Wahrscheinlichkeiten von Null bei den N-Gram Sprachmodellen, können durch das Smoothing verhindert werden. Das Smoothing verteilt allerdings das Wahrscheinlichkeitsgewicht ungleich über die Wörter[JM19, p. 42-49]. Bei Neuronalen Sprachmodellen, kommt eine Wahrscheinlichkeit von Null nicht vor, da die Einheiten des Neuronalen Netzwerkes mit zufälligen kleinen Zahlen, am Anfang des Trainieren initialisiert werden und somit keine Nullwahrscheinlichkeiten als Ausgabe raus kommen können[JM19, p. 137]. Deshalb sind Neuronale Netzwerke im Hinblick auf Bestimmung von Satzwahrscheinlichkeiten den N-Gram Modellen überlegen.
Bei der Qualität der Wahrscheinlichkeitsverteilung von N-Gram Sprachmodellen und Neuronalen Sprachmodellen, haben wir festgestellt, dass die Wahrscheinlichkeitsverteilungen eines Neuronalen Sprachmodells, sowohl aus Sicht auf das Embedding, als auch im Hinblick auf Bestimmung von Satzwahrscheinlichkeiten ein besseres Ergebnis liefern.

2.3 Verwendung bei Sprachverarbeitung

Als letztes betrachten wir, inwiefern N-Gram Modelle und Neuronale Sprachmodelle bei der Verarbeitung von Gesprochener Sprache eine Rolle spielen beziehungsweise Verwendung finden.

Wir haben bereits festgestellt, dass der Aufwand, der benötigt wir um ein N-Gram Sprachmodelle zu berechnen geringer ist als der Aufwand, um ein Neuronales Sprachmodell zu trainieren. Dementsprechend ist ein N-Gram Sprachmodell für die Sprachverarbeitung besser geeignet als ein Neuronales Sprachmodell, wenn mehr Wert auf Einfachheit, als auf Qualität gelegt wird. Ein Weiterer Vorteil von N-Gram Modellen gegenüber Neuronalen Sprachmodellen bei der Verarbeitung gesprochener Sprache, ist das sie weniger Rechenleistung erfordern[Raj+19].
Ein Vorteil eines Neuronalen Sprachmodells gegenüber einem N-Gram Sprachmodell, ist der, dass ein Neuronales Sprachmodell, insbesondere eine rekursives Neuronales Sprachmodell, welches sehr lange Kontexte erlaubt[JM19, chapter 9], zusammenhänge zwischen in einer Wortsequenz sehr weit entfernten Wörtern erkennen kann, welche ein N-Gram Sprachmodell nicht erkennen würde. Außerdem kann die Fehlerrate die bei der Spracherkennung auftritt durch Neuronale Sprachmodelle reduziert werde[Raj+19].
Es scheint allerdings so, als wäre die beste Lösung eine Kombination aus N-Gram Sprachmodellen und Neuronalen Sprachmodellen. Dabei wird das N-Gram Sprachmodell als Basis verwendet und das Neuronale Sprachmodell wird verwendet um die Schwächen des N-Gram Modells zu beheben[HKN14][Raj+19].

Ergebnis und Ausblick

In dieser Arbeit, haben wir herausgestellt, dass es sowohl Vorteile eines N-Gram Sprachmodells gegenüber einem Neuronalen Sprachmodell gibt, als auch andersherum. N-Gram Sprachmodelle sind einfacher zu trainieren/berechnen, da nur alle Wörter mit allen Kontexten ausgerechnet werden müssen, währen bei einem Neuronalen Sprachmodell alle Wortsequenzen des Korpus durchgegangen werden müssen. Neuronale Sprachmodelle bieten dafür eine bessere Qualität der Wahrscheinlichkeitsverteilung im Hinblick auf Embedding und der Abwesenheit von Nullwahrscheinlichkeiten. Wenn man sich die Verwendung von Sprachmodellen bei der Verarbeitung gesprochener Sprache anschaut, kann man feststellen, dass eine Kombination aus N-Gram Sprachmodell und Neuronalen Sprachmodell die beste Lösung ist, da sie Stärken von beiden Sprachmodellen vereint.
In Zukunft, wird es interessant sein zu sehen, inwieweit die Integration von Neuronalen Sprachmodelle in die Spracherkennung die Fehlerrate bei der Erkennung von Wörtern weiter reduziert und wie der höhere Rechenaufwand, der benötigt wird, um ein Neuronales Sprachmodell zu benutzen, umgangen wird, damit die Integration Neuronaler Sprachmodelle, in die Spracherkennung, effizient ist.

Quellen

[HKN14] Takaaki Hori, Yotaro Kubo und Atsushi Nakamura. *Real-time one-pass decoding with recurrent neural network language model for speech recognition*. 2014. URL: https://core.ac.uk/download/pdf/24061415.pdf.

[JM19] Daniel Jurafsky und James H. Martin. *Speech and Language Processing*. Okt. 2019. URL: https://web.stanford.edu/~jurafsky/slp3/.

[Raj+19] Anirudh Raju u. a. *Scalable Multi Corpora Neural Language Models for ASR*. Juli 2019. URL: https://arxiv.org/pdf/1907.01677.pdf.