Recurrent Neural Networks

Marcel Canclini, marcel.canclini@gmail.com

5 Oktober 2016

Abstract

"A Critical Review of Recurrent Neural Networks for Sequence Learning" (Lipton, Berkowitz und Elkan 2015) gibt eine Übersicht von Modellen, welche zeitliche Zusammenhänge erkennen. Dabei wird neben der Geschichte und aktuellen Applikationen speziell auf den Long Short Term Memory Baustein (LSTM) sowie die Bidirectional Recurrent Neural Network (BRNN) Architektur eingegangen. Es gibt zudem einen Überblick über Feedforward Netzwerke, deren Architekturen sowie Beschreibungen der wichtigsten damit verbunderen Begriffe.

Dieses Dokument beschreibt die wichtigsten Punkte zum allgemeinem Verständnis von Neuronalen Netzen anhand des Berichts von Lipton und setzt sich danach detailierter mit Recurrent Neural Networks (RNN) auseinandersetzen um die Möglichkeiten zur Anwendung in der Klassifikation und Metadatenextraktion von Texten aufzuzeigen.

Warum das Thema Recurrent Neural Networks?

Im Rahmen des Machine Learing Moduls (CAS Data Science Applications) wurden die Grundlagen von Neuronalen Netzen sowie ein Einblick in Convolutional Neural Networks (CNN) vermittelt.

CNN sind bekannt, im Bereich der Bild- und Spracherkennung sehr gute Resultate zu erzielen. Standard Feedforward Netzwerke haben die Problematik keine zeitlichen Abhängigkeiten zu kennen und sind somit nicht in der Lage von vergangenen oder zukünftigen Ereignissen auf ein Resultat zu schliessen. Um die Übersicht von Neuronalen Netzen zu vervollständigen fehlen aus meiner Sicht die Recurent Neural Networks (RNN).

Im Bereich Gesundheitswesen ist durch eHealth die Digitalisierung und Strukturierung von medizinischen Dokumente ein grosses Thema. Der Grossteil der Dokumente wird heute unstrukturiert erfasst. Mittels Machine Learning, speziell RNN besteht die Möglichkeit aus diesen Dokumenten strukturierte Medatdaten zu extrahieren. Beispiel sind Behandlungsdatum, Diagnosen, Medikation, etc.

Zusammenfassung

Neuronale Netzwerke

Neuronale Netzwerke sind biologisch inspirierte Berechnungsmodelle. Ein solches Netz besteht aus mehreren künstlichen Neuronen häufig als *Knoten* j bezeichnet. Die Knoten werden mit gerichteten *Kanten*, verbunden. Jedes Neuron hat eine Aktivierungsfunktion l_j. Gängige Aktivierungsfunktionen sind sigmoid, Rectifier Linear Unit (ReLU) sowie tanh. Jeder Kante ist ein Gewicht w zugeteilt. Das Gewicht wird gerichtet von Knoten j' zu j als w_{jj} bezeichnet.

Somit berechnet sich der Wert eines Neurons indem auf die Summe der gewichteten Eingänge des Neurons die Aktivierungsfunktion angewendet wird:

$$v_j = l_j \left(\sum_{j'} w_{jj'} \cdot v'_{j'} \right)$$

Feedforward Neural Networks

Diese Klasse von Neuronalen Netzen lassen keine Zirkelbezüge zu. Die Knoten werden in einzelnen Layer angeordnet, wobei zwischen Eingabe-, Hidden- und Ausgabe-Layer unterschieden wird. Jedes Netz hat einen Eingabe- und einen Ausgabe-Layer, kann aber mehrere Hidden Layers haben. Die Layers sind aufeinandergeschichtet, wobei die Ausgabe eines Layers auf Basis der Ausgabe des vorangegangenen Layers berechnet wird.

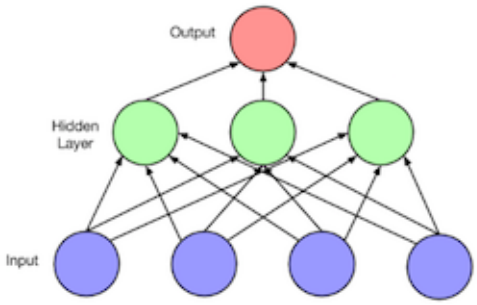

Figure 1: Einfaches Feedforward Netzwerk mit jeweils einem Input, Hidden und Output Layer (Lipton, Berkowitz und Elkan 2015, p. 8)

Nach jeder Verarbeitung einer Eingabe wird der Status des Netz zurückgesetzt. Es verliert also die Information über die soeben verarbeiteten Daten. Ein solches Netz ist geeignet bei voneinander unabhägigen Eingabewerten. Bei Applikationen, bei denen eine zeitliche Abfolge vorhanden ist (video, audio, text, etc.) und eine Abhängikeit der Eingabedaten gegeben ist, ergibt sich ein Problem.

Backpropagation Algorithmus

Der Backpropagation Algorithmus ist die meist verbreiteste Art ein Neuronales Netz zu trainieren. Dabei wird der Ausgabefehler mittels Differenzierung der Loss Funktion im Netz züruckgespielt. Jedes einzelne Gewicht w im Netzwerk wird mittels Gradient Descent so angepasst, dass bei erneutem Anlegen der selben Eingabewerte eine Annäherung der Ausgabe $\hat{y}$ an den realen Wert y erfolgt.

Recurrent Neural Network

RNN sind eine Unterart der Feedforward Netze. Im Unterschied zur herkömmlichen Architektur, wird bei RNN im Hidden Layer ein Zirkelbezug eines Neurons mit sich selbst erstellt. Dies ermöglicht den Vorhergehenden Wert in den nächste Zeitschritt zu übernehmen. Ein mit sich selbst verbundener Hidden Layer lässt sich über mehrere Zeitschritte "auffalten" und entspricht dann einem Deep Network mit einem Hidden Layer pro Zeitschritt. Ein solches Netz kann wiederum mittels Backpropagation trainiert werden.

So lässt sich eine zeitliche Abfolge in einem neuronalen Netz modellieren und zeitliche Abhängigkeiten von Eingabedaten berücksichtigen.

Eine Schwierigkeit bei RNN ist das **Trainieren der Modelle** da sich das Modell über die Zeitschritte optimieren soll. Dabei kommt es zu verschwindenden oder exponentiell explodierenden Gradienten je nach Anzahl Zeitschritte und Aktivierungsfunktion. Details sind (Lipton, Berkowitz und Elkan 2015, chap. 3.2) zu entnehmen. Diesem Problem kann man mittels Regularisierung entgegenwirken. **Truncated backpropagation through time (TBPTT)** ist eine Variante, bei welcher die Anzahl der Zeitschritte eingeschränkt wird.

Die **Optimierung** eines RNN ist komplexer und kann nicht einfach durch Anpassung der Netzwerkarchitektur gelöst werden. Neuere empirische und theoretische Versuche zeigen jedoch, dass das Verhältnis von Sattelpunkten zu wahren lokalen Minima exponentiell mit der Grösse des Netzwerks steigt. Je grösser ein Netzwerk ist, desto wahrscheinlicher ist es, ein wahres lokales Minimum bei der Optimierung mittels Gradient Descent zu finden.

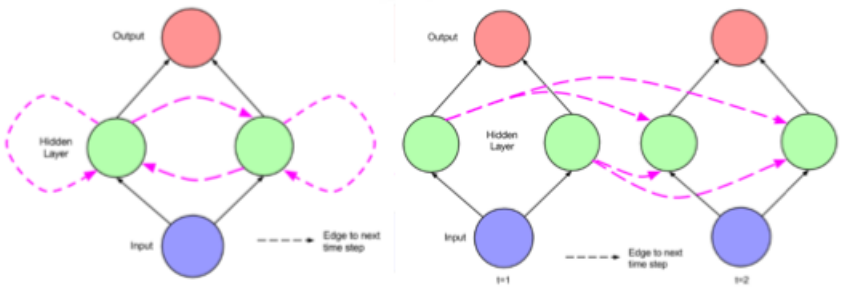

Figure 2: RN: links als Architektur und rechts aufgefalltet um 2 Zeitschritte zu visualisieren.(Lipton, Berkowitz und Elkan 2015, fig. 3 und 4)

Long Short Term Memory (LSTM)

Um den verschwindenden Gradienten entgegenzuwirken haben Hochreiter und Schmidhuber das Long Short Term Memory (LSTM) Modell entwickelt (Hochreiter und Schmidhuber 1997). Dieses führt eine Memory Zelle anstelle eines künstlichen Neurons ein. Die Memory Zelle besteht aus 3 Gates (Input, Forget und Output Gate) sowie einem internen Status (Memory Zelle). Durch diese Konstellation wird verhindert, dass bei der Backpropagation der Gradient weder verschwindet, noch explodiert und das Netzwerk in der Lage ist auch über längere Zeitabschnitte zu lernen.

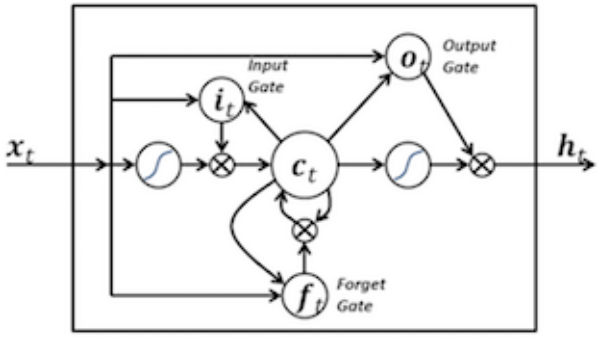

Figure 3: LSTM Modul inklusive der Peephole Connections welche durch Gers und Schmidhuber vorgestellt wurden (Gers und Schmidhuber 2000)

Bidirectional Recurrent Neural Network (BRNN)

Durch Schuster und Paliwal (Schuster und Paliwal 1997) vorgestellte Architektur. Hierbei werden neben den vergangenen auch die zukünftigen Ereignisse miteinbezogen. Dies ermöglicht eine Aussage zum Zeitpunkt t, welche auf Daten beruht die erst in einem zukünfitgen Zeischritt bekannt sind.

Ein BRNN umfasst 2 Hidden Layers, wobei beide mit der Eingabe und der Ausgabe verbunden sind. Der Unterschied der beiden Layers besteht in der zeitlichen Verbindung. Der erste Layer hat Verbindungen vom vorhergehenden Zeitschritt, wobei im zweiten Layer die Richtung der Verbindungen gedreht wird und die Aktivierung abhängig vom zukünftigen Zeitschritt ist.

Ein Bidirectional Recurrent Neural Networks bedingt eine **fixe Länge** der Eingabesequenz um mittels Backpropagation trainiert zu werden. Dies ist die **Limitation** der BRNN. Sie können nicht mit kontinuierlichen Sequenzen umgehen. Es braucht einen klar definierten zeitlichen Endpunkt, sowohl in der Zukunft als auch in der Vergangenheit. Eine Vorhersage von zukünftigen Ereignissen kann nicht gemacht.

Ein guter, anschaulicher Anwendungsfall ist part-of-speech tagging (POS Tagging). Dabei soll die Wortart eines Wortes innerhalb eines Satzes bestimmt werden. Die Sequenz der Wörter kann dabei als zeitliche Abfolge betrachtet

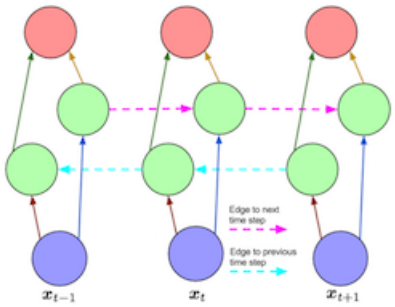

Figure 4: Bidirectional Recurrent Neural Network mit den 2 Hidden Layers, welches sowohl die vergangenen als auch die zukünftigen Werte berücksichtigt. (Lipton, Berkowitz und Elkan 2015, fig. 12)

werden. Für die Bestimmung eines Wortes in der Mitte des Satzes sind nicht nur die vorhergehenden, sonder auch die folgenden Wörter relevant.

Kombination und Anwendungen von LSTM und BRNN

Die beiden Ansätze lassen sich sehr gut kombinieren. LSTM ist ein neuer Baustein, welcher innerhalb eines Hidden Layers verwendet werden kann. BRNN verbindet die vorhandenen Hidden Layers auf eine neue Art. Diese Kombination wurde als **BLSTM** bei der Handschriftenerkennung und Phonem Klassifikation eingesetzt.

Aktuelle Einsatzgebiete von LSTM und BRNN sind im Bereich von Text Processing (z.B. Übersetzung von Texten) und Image Captioning (Generierung von Bildunterschriften) zu finden.

Schlussfolgerung

"A Critical Review of Recurrent Neural Networks for Sequence Learning" erklärt nicht nur anschaulich wie Recurrent Neural Networks funktionieren, sonder gibt auch eine gut strukturierte Einführung in Neuronale Netze und die wichtigsten Begriffe in diesem Bereich. Auch die kurze Erklärung der verwendeten Nomenklatur hilft beim weiteren Studium von Material in diesem Fachgebiet.

Applikationen wie Übesetzung von Texten oder Part-Of-Speech Tagging zeigen, dass eine Anwendung zur Metadatenextraktion auf unstrukturierten, medizinischen Dokumente ein Anwendungsfall sein kann.

Referenzen

Gers, F. A. und J. Schmidhuber. 2000. Recurrent Nets that Time and Count. In: *IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000*, 3:189–194 vol.3. doi:10.1109/IJCNN.2000.861302,.

Hochreiter, S. und J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, Nr. 8: 1735–1780. doi:10.1162/neco.1997.9.8.1735,.

Lipton, Zachary C., John Berkowitz und Charles Elkan. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv:1506.00019 [cs]*. http://arxiv.org/abs/1506.00019 (zugegriffen: 2. Oktober 2016).

Schuster, M. und K. K. Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing* 45, Nr. 11: 2673–2681. doi:10.1109/78.650093,.