



[AI-United](#) » [LSTM](#) » LSTM als neuronales Netzwerk mit langem Kurzzeitgedächtnis

LSTM als neuronales Netzwerk mit langem Kurzzeitgedächtnis



Die Menschen fangen mit der Denktätigkeit nicht jedes Mal neu an. Beim Lesen eines Artikels versteht man die Bedeutung jedes Wortes aufgrund der Bedeutungen der vorigen Wörter. Den Gedanken ist es eigen, sich anzusammeln und gegenseitig zu beeinflussen. Dieses Prinzip verwenden die LSTM-Netzwerke.

Einfache neuronale Netzwerke können dies nicht tun, was ihr großer Nachteil ist. Stellen Sie sich folgende Situation vor: Es sollten in Echtzeit die Ereignisse klassifiziert werden, die in einem Film geschehen. Es ist eine sehr große Frage, wie ein gewöhnliches neuronales

Netzwerk das Wissen über vergangene Ereignisse nutzen kann, um nachfolgende zu erforschen.

Rekurrentes neuronales Netz (RNN)

Rekurrente neuronale Netze sind imstande, dieses Problem zu lösen, da sie aus Zyklen bestehen, die die Informationen speichern.

*Zyklen
in
RNN*

Das Diagramm oben zeigt, dass ein Teil des neuronalen Netzwerks **A** das Eingangssignal **x** annimmt und den Wert **h** ausgibt. Dieser Zyklus ermöglicht, die Informationen von einem Netzwerkschritt zu dem anderen zu übergeben.

Wegen des Vorhandenseins von Zyklen sehen RNN etwas geheimnisvoll aus. Wenn man sie aber genauer betrachtet, wird es klar, dass sie sich nicht von einem einfachen neuronalen Netzwerk unterscheiden. Ein RNN kann man sich als mehrere Kopien ein und desselben Netzwerks vorstellen, wobei jede Kopie eine Nachricht an einen Nachfolger sendet. Betracht... wir, was passiert, wenn ein Zyklus abgetastet wird:

*Zyklusabtastung
in RNN*

Die abgetastete Struktur des RNNs zeigt, dass sich wiederholende neuronale Netzwerke eng mit Sequenzen und Listen verbunden sind. Dies ist die natürliche Architektur eines neuronalen Netzwerkes, die für diesen Datentyp verwendet wird.

Zweifellos werden sie heutzutage sehr aktiv verwendet! Einen unglaublichen Erfolg erreichten die RNN in den letzten Jahren bei der Anwendung in solchen Forschungsgebieten wie Spracherkennung, linguistische Modellierung, Übersetzung, Bildbeschreibung... Die Liste ist endlos. Zu einem wesentlichen Hilfsmittel bei der Lösung der angeführten Aufgaben wurden

LSTM. Sie sind rekurrente neuronale Netze eines spezifischen Typs, die einzelne Aufgaben wesentlich effizienter als Standardmethoden lösen. Mit der Verwendung von LSTM-Netzen sind alle bedeutenden Ergebnisse verbunden, die auf RNN basieren.

Das Problem langfristiger Beziehungen

Das rekurrente neuronale Netzwerk verwendet die zuvor erhaltenen Informationen, um die nachfolgenden Aufgaben zu lösen, z.B. können die nachfolgenden Videofragmente aufgrund der früher erhaltenen analysiert werden.

Manchmal ist zur Lösung einer Aufgabe nur die letzte Information erforderlich. Beispielsweise wird ein Sprachmodell erstellt, das versucht, das nächste Wort basierend auf den vorherigen Wörtern vorherzusagen. Um das letzte Wort in der Wortverbindung "Wolken am Himmel" vorherzusagen, wird keinen zusätzlichen Kontext gebraucht: Es ist offenbar, dass das nächste Wort "Himmel" sein wird. Wenn die Lücke zwischen den vorherigen Informationen und dem Ort, an dem sie benötigt wird, klein ist, wird der RNN die Aufgabe bewältigen. RNN löst erfolgreich diese Aufgabe nur in den Fällen, wenn die Lücke zwischen den vorherigen Informationen und dem Ort, an dem sie notwendig ist, klein ist.



Manchmal wird aber mehr Kontext gebraucht. Betrachten wir den Versuch, das letzte Wort im folgenden Text vorherzusagen: "Ich bin in Deutschland aufgewachsen... Ich spreche fließend Deutsch." Aus den vorangegangenen Wörtern ist klar, dass das nächste Wort wahrscheinlich der Name der Sprache ist. Um die richtige Sprache zu nennen, muss die Erwähnung von Deutschland berücksichtigt werden. Der Abstand zwischen der Information, die für die Erfassung erforderlich ist, und dem Punkt, an dem sie benötigt wird, wird größer.

Leider **verliert das RNN bei vergrößertem Abstand die Verbindung zwischen den Informationen.**

Rein theoretisch kann RNN solche "langfristigen Abhängigkeiten" bewältigen. Um dieses Problem zu lösen, kann der Forscher die Einstellungen des Netzwerkes sorgfältig zusammenstellen. Leider ist RNN in der Praxis unfähig, dieses Problem zu lösen. Dieses

Thema wurde in Studien [[Hochreiter \(1991\)](#)] und [[Bengio et al. \(1994\)](#)] erforscht, in denen die grundlegenden Einschränkungen von RNN beschrieben wurden. Zum Glück **hat LSTM dieses Problem nicht!**

LSTM-Netzwerke

Unter LSTM (*Long Short Term Memory, langes Kurzzeitgedächtnis*) wird eine Art rekurrentes neuronales Netzwerks gemeint, das langfristige Abhängigkeiten erlernen kann. Zuerst wurde LSTM in der Forschung [Hochreiter & Schmidhuber \(1997\)](#) vorgestellt, später von anderen Forschern verbessert und populär gemacht. Es bewältigt erfolgreich viele Aufgaben und wird immer noch weit verwendet.

LSTM wurde mit dem Ziel erarbeitet, Probleme der langfristigen Abhängigkeit zu beseitigen. Ihre Spezialität ist das Speichern von Informationen für längere Zeitperioden, dementsprechend müssen sie praktisch nicht trainiert werden!

Alle rekurrenten neuronalen Netzwerke haben die Form einer Kette wiederkehrender Module eines neuronalen Netzwerks. Für dieses wiederkehrende Modul in Standard-RNN ist eine einfache Struktur typisch, z.B. eine **Tanh**-Schicht.



*Das
wiederkehrende
Modul des
Standard-RNNs
enthält eine
Schicht*

*Das
wiederkehrende
LSTM-Modul
enthält vier
zusammenwirkende
Schichten*

Die Details sind im Moment unwichtig. Betrachten wir nun die Begriffe, die weiter verwendet werden.

Jede Zeile bedeutet einen Vektor. Rosa Kreis dient der Bezeichnung von punktweisen Operationen, zum Beispiel der Summierung von Vektoren. Unter gelben Zellen werden die Schichten eines neuronalen Netzwerks gemeint. Die Linienverknüpfung ist die Vereinigung von Vektoren, und das Zeichen der Gabelung bedeutet das Kopieren des Vektors mit weiterer Speicherung an verschiedenen Stellen.

Arbeitsweise des LSTM-Netzwerks

Der Schlüsselbegriff von LSTM ist der Zellzustand: Dies ist eine horizontale Linie, die durch den oberen Teil des Diagramms durchgeht.

Den Zellzustand kann man mit einem Förderband vergleichen, da er die gesamte Kette durchläuft und kleinen linearen Veränderungen unterliegt.

Von Bedürfnissen hängt es ab, ob LSTM die Informationsmenge in dem Zellzustand verringert oder erhöht. Zu diesem Zweck werden sorgfältig eingestellte Strukturen verwendet, die man auch Gates nennt.



Ein Gate ist ein "Tor", das die Informationen entweder durchlässt oder nicht. Die zwei wichtigsten Bestandteile von Gates sind eine Sigmoidschicht eines neuronalen Netzwerks und eine punktweise Multiplikationsoperation.

Um zu bestimmen, wie viel Prozent jeder Informationseinheit weiter durchgelassen werden soll, werden am Ausgang der Sigmoidschicht Zahlen von Null bis Eins ausgegeben. Der Wert "0" bedeutet "Nichts durchlassen", der Wert "1" bedeutet "Alles durchlassen".

Schrittweises Arbeitsschema des LSTM-Netzwerks

In LSTM unterscheidet man drei Gates zur Kontrolle des Zellzustandes, die weiter genauer betrachtet werden.

Verlustschicht

Im ersten Schritt muss es entschieden werden, welche Informationen aus dem Zellzustand ausgeworfen werden. Diese Entscheidung wird von einer Sigmoidschicht getroffen, die als "Forget Gate-Schicht" bezeichnet wird. Sie bekommt die Eingabe \mathbf{h} und \mathbf{x} und gibt die Zahl von 0 bis 1 für jede Zahl im Zellzustand \mathbf{C} an. **1** steht für "vollständig speichern" und **0** steht für "vollständig löschen".

Betrachten wir unser Beispiel des Sprachmodells erneut. Jetzt wird versucht, das nächste Wort vorherzusagen, das auf allen vorherigen Wörtern basiert. Bei solcher Aufgabe enthält der Zellzustand das Geschlecht des Subjekts, um die richtigen Pronomen zu verwenden. Beim neuen Subjekt muss das Geschlecht des vorigen Subjekts vergessen werden.

Speicherschicht

Im nächsten Schritt muss entschieden werden, welche neuen Informationen im Zellzustand gespeichert werden. Dieser Prozess läuft in zwei Etappen durch. Zuerst entscheidet die Sigmoidschicht, die auch "Schicht des Eingangsgates" genannt wird, welche Werte aktualisiert werden müssen. Dann erstellt die **Tanh**-Schicht den Vektor neuer Werte \mathbf{C} , die dem Zellzustand hinzugefügt werden. Als nächstes werden diese beiden Werte zur Aktualisierung des Zustandes verbunden.



Im Beispiel unseres Sprachmodells wird der Zellzustand vom Geschlecht des neuen Subjekts ergänzt, um das Geschlecht des alten Subjekts zu ersetzen.

Neuer Zustand

Jetzt wird der vorige Zellzustand aktualisiert, um den neuen Zustand \mathbf{C} zu bekommen. Nachdem Aktualisierungsmethode ausgewählt wurde, wird die Aktualisierung selbst durchgeführt.

Zuerst wird der alte Zustand mit \mathbf{f} multipliziert, indem die Informationen verloren werden, die zu vergessen waren. Dann wird $\mathbf{i} * \mathbf{C}$ hinzugefügt. Dies sind neue Kandidatenwerte. Sie werden dann abhängig davon skaliert, wie jeder Wert des Zustandes aktualisiert wurde.

Im Fall des Sprachmodells werden die Informationen über das Geschlecht des alten Subjekts abgelehnt und neue Informationen hinzugefügt.

Zum Schluss muss entschieden werden, was am Ausgang erhalten werden soll. Das Ergebnis ist der gefilterte Zellzustand. Zuerst wird die Sigmoidschicht gestartet, die entscheidet, welche Teile des Zellzustands auszugeben sind. Dann wird der Zellzustand durch **tanh** durchgelassen (um alle Werte im Intervall $[-1, 1]$ anzuordnen) und mit dem Ausgangssignal des Sigmoid-Gates multipliziert.

Da das Netzwerk nur mit dem Subjekt arbeitete, kann es für das Sprachmodell die auf das Verb bezogenen Informationen ausgeben. Zum Beispiel gibt das Netzwerk Informationen über den Numerus des Subjekts (Singular oder Plural) aus, um das Verb richtig zu konjugieren.

LSTM-Beispiele



Das oben beschriebene Schema ist traditionell für LSTM. Nicht alle LSTM sind aber gleich. In der Tat werden fast in jedem Artikel verschiedene Versionen verwendet. Obwohl die Unterschiede geringfügig sind, lohnt es sich, einige davon zu erwähnen.



In der Forschung [[Gers & Schmidhuber \(2000\)](#)] wurde die beliebte Version von LSTM dargestellt, in der sich Gate-Schichten den Zellzustand anschauen können.

In obigem Diagramm haben alle Gates ein "Äuglein", aber in vielen Artikeln ist es nur in einigen der Gates vorhanden.

Die zweite Variante ist die Verwendung von verbundenen Verlust- und Eingangsgates. Anstatt getrennt zu entscheiden, was zu vergessen ist und wozu neue Informationen hinzuzufügen sind, werden diese Entscheidungen gleichzeitig getroffen. Die Informationen werden nur dann vergessen, wenn etwas Neues an derselben Stelle angeordnet werden muss. Neue Werte werden erst dann in den Zustand eingetragen, wenn etwas Älteres vergessen wird.

Ein etwas anderes LSTM-Beispiel ist Gated Recurrent Unit oder kurz GRU, das in [Cho, et al. al. \(2014\)](#) eingeführt wurde. Hier werden der Zellzustand und der verborgene Zustand miteinander verbunden und einige andere Änderungen eingetragen. Da das resultierende Modell einfacher als Standard-LSTM-Typen ist, wird es immer beliebter.

Dies sind nur einige der bekanntesten LSTM-Beispiele. Es gibt viele andere, beispielsweise wird in [Yao, et al. \(2015\)](#) ein RNN mit einem Tiefe-Gate dargestellt und in Clockwork RNN von [Koutnik et al. \(2014\)](#) ein völlig anderer Ansatz zur Lösung langfristiger Abhängigkeiten beschrieben.

Welches der folgenden Modelle ist das Beste? Sind die Unterschiede sehr wichtig? In [Greff et al. \(2015\)](#) werden die beliebten LSTM-Versionen ausführlich verglichen und gezeigt, dass sie alle ungefähr gleich sind. In [Jozefowicz et al. \(2015\)](#) wurden mehr als zehntausend RNN-Architekturen getestet. Bei der Lösung bestimmter Probleme sind einige von ihnen besser als LSTM.



Fazit



Wenn LSTM als ein Satz von Gleichungen betrachtet wird, erscheint es schrecklich. Hoffentlich der schrittweise Ablauf des gesamten Schemas in diesem Beitrag, es zugänglicher zu machen.

LSTM war ein großer Schritt in der Entwicklung von RNN. Natürlich entsteht eine weitere Frage: Kann man weitergehen? Die allgemeine Meinung aller Forscher lautet: "Ja! Der nächste Schritt besteht in der Verwendung des Aufmerksamkeitsmechanismus!" Wie funktioniert er? Bei jedem Schritt wählt RNN die Informationen zum Anschauen aus jeder großen Datenmenge aus. Beispielsweise wird mithilfe von RNN eine Bildbeschreibung erstellt, so kann das Netzwerk einen Teil des Bildes zum Anschauen jedes Ausgangswortes auswählen. Genau dies wurde in [Xu, et al. \(2015\)](#) gemacht und wird der Ausgangspunkt für die Forschung des Aufmerksamkeitsmechanismus sein.

Wenn Sie mehr über LSTM-Netzwerke erfahren möchten, können Sie sich an das Team von AI-United.de per Mail oder Q&A wenden.

[Quelle](#)

AI-United-Redaktion

[Follow](#)**KI CONSULTING**

Name (Pflichtfeld) *

E-Mail-Adresse (Pflichtfeld) *

Senden



#Algorithms

#lstm

#Neural Networks

#Recurrent Neural Networks

#Training

Kommentar hinzufügen

Kommentar schreiben...

Name

E-mail address

Webseite Link

Senden

