

Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Fachbereich Informatik
Arbeitsbereich Informationsdienste

Bachelorarbeit Informatik

Event Segmentation mit Hilfe eines Rekurrenten Neuronalen Netzes mit LSTM

Steffen Schnürer

14.06.2017

Gutachter

Prof. Dr. Martin Butz
Fachbereich Informatik
Arbeitsbereich Kognitive Modellierung
Universität Tübingen

Betreuer

Sebastian Otte

Steffen Schnürer:

*Event Segmentation mit Hilfe eines Rekurrenten Neuronalen
Netzes mit LSTM*

Bachelorarbeit Informatik

Eberhard Karls Universität Tübingen

Bearbeitungszeitraum: 14.02.2017 - 14.06.2017

Erklärung

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbstständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe.

Tübingen, den 14.06.2017

(Steffen Schnürer)

Inhaltsverzeichnis

1	Einführung	1
2	Grundlagen	2
2.1	Neuronale Netze	2
2.2	Rekurrente Neuronale Netze	4
2.3	Das Vanishing Gradient Problem	4
	Abbildungsverzeichnis	6

1 Einführung

Bishier noch bisschen leer...

2 Grundlagen

In diesem Kapitel möchte ich einige Grundlagen für die weiteren Kapitel dieser Arbeit legen. Zunächst erläutere ich den Aufbau eines neuronalen Netzes im allgemeinen, die Bedeutung von rekurrenten neuronalen Netzen (RNN), zeige am Bouncing Ball Szenario das Vanishing Gradient Problem auf und mache so die Notwendigkeit der LSTM-Technik deutlich.

2.1 Neuronale Netze

Als neuronales Netz bezeichnet man eine verwobene Struktur zwischen vielen einzelnen Zellen von meistens gleichem - aber keineswegs darauf beschränktem - Aufbau, den sogenannten Neuronen. Eine solche Zelle hat immer die Eigenschaft, dass sie Signale von anderen Zellen empfängt, diese gewichtet aufakkumuliert und abhängig von einer internen Aktivierungsfunktion ein entsprechendes Signal an andere Zellen weitergibt, die damit ihrerseits diesselbe Prozedur durchlaufen. Ein neuronales Netz im Gehirn einer Ameise hat ca. 250.000 Neuronen, ein menschliches 86 Milliarden (1) und wir haben lediglich eine wage Vorstellung, wozu diese imstande sind. In der Informatik werden solche Strukturen als künstliche neuronale Netze nachgebildet und je nach Problemstellung abgewandelt. Zur Vereinfachung meinen wir ab sofort, sofern nicht explizit anders angegeben, mit neuronalen Netzen künstliche neuronale Netze. Es gibt viele verschiedene Arten von neuronalen Netzen, die einfachste von Ihnen ist das Multilayer Perceptron (MLP). Die Zellen werden zu mehreren Ebenen(Layer) zusammengefügt und es gibt eine geordnete Datenflussrichtung, eine Zelle erhält ihren Input von jeder Zelle der vorigenen Ebene und gibt ihren Output entsprechend an jede Zelle der folgenden Ebene weiter, Feedforward genannt (Vgl Bild 2). Bild 3 Künstliches Neuron Gl 1: $x_h = \phi(h(\text{net}(h)))$

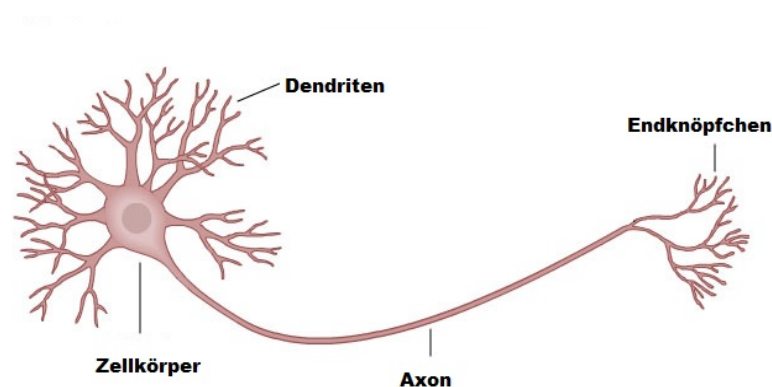


Abbildung 2.1: Schematische Darstellung eines biologischen Neuron [1]

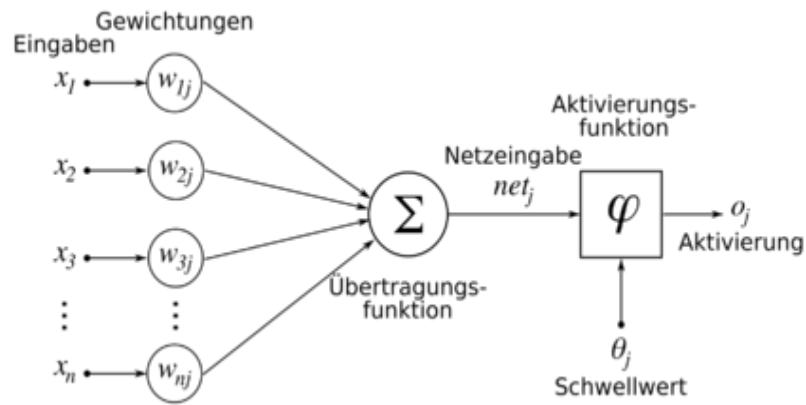


Abbildung 2.2: Schematische Darstellung eines künstlichen Neuron [2]

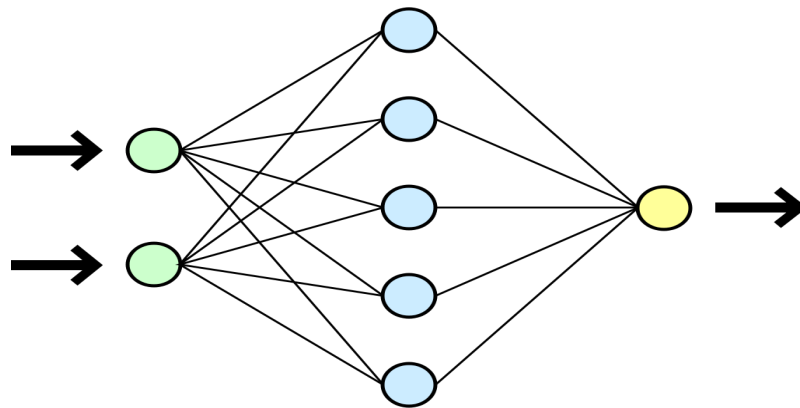


Abbildung 2.3: Ein Multilayer Perceptron mit 2 Inputneuronen(grün), einem Hidden Layer mit 5 Neuronen(blau) und einem einzelnen Outputneuron(gelb). [3]

= $\phi(\sum_i w_{ij} x_i)$ In jeder Zelle werden die Inputsignale, also die Aktivierungen der vorigen Zellen aufsummiert, aber unterschiedlich gewichtet vgl GL 1. Eben diese Gewichte sind der Kern des maschinellen Lernens, da bei einmal geschickt gefundenen Gewichten, komplexe Aufgabenstellungen und Probleme mit (vergleichsweise) wenig Rechen- sowie Programmieraufwand gelöst werden können. Im Wesentlichen versucht ein MLP immer eine Funktion zu berechnen, welche einen Inputvektor der Größe n auf einen Outputvektor der Größe m abbildet. FM: $R^n \rightarrow R^m$ (Seite 11). Ein Beispiel für eine solche Funktion wäre freigegebenes Bild zu unterscheiden, ob daraus Layerentsprechende Inputdaten existieren, berechnet die Aktivierungen aller Zellen, vergleicht die Aktivierung mit der Zielaktivierung $E(z, y) = \frac{1}{2} \sum_i (z_i - y_i)^2$. Die Herausforderung ist es nun, diejenigen Gewichte zu finden, die die Aktivierung der Outputzellen mit der Zielaktivierung y übereinstimmen. Gelte f für die Aktivierungsfunktion. Formuliere den Algorithmus. Code einfügen?

2.2 Rekurrente Neuronale Netze

Neuronale Netzwerke haben einen für einen 1-Dimensionalen Input einen klar definierten 1-Dimensionalen Output und für die meisten Problemstellungen ist auch genau diese Kompetenz gefordert. Es gibt aber auch Probleme, bei denen der Output nicht nur von diesem, sondern auch von allen beziehungsweise einigen vorigen Inputs / Zuständen abhängt. Ein anschauliches Beispiel wäre hier die Klassifizierung von Videoausschnitten, wo die Bedeutung eines einzelnen Frames vom Kontext der vorigen Frames abhängt, hingegen ein einfacheres Beispiel wäre das in dieser Arbeit betrachtete Bouncing Ball Szenario. Befindet sich ein Ball im Punkt (0/0) und der nächste Ort soll vorhergesagt werden, ist es natürlich entscheidend ob der Ball sich zuvor beispielsweise im Punkt (0.1/0.1) oder im Punkt (-0.1/-0.1) befand. Hierzu werden den Zellen des neuronalen Netzes, außer den Input und Output Neuronen rekurrente (von lat. badidadam) Verbindungen hinzugefügt, wie z.B. In Abbildung 43. Bild eines rekurrenten Netzes. Eine Zelle bekommt nun seinen Input immernoch von allen Zellen des unteren Layers, zusätzlich nimmt sie aber auch noch als Input den Output aller Zellen desselben Layers, aber aus dem vorigen Zeitschritt. Formel für Aktivierungen in RNN Da diese neuen rekurrenten Inputs auch wieder gewichtet verarbeitet werden, müssen diese erst noch geschickt gefunden werden. Dies macht man mit Backpropagation through time (Rückpropagierung durch die Zeit, BPTT). Dies passiert analog zur Backpropagation im MLP, jedoch werden die einzelnen Zeitschritte aufgeklappt und (entnehme Formulierung aus Literatur und zitiere sie hier) Formel für BPTT einfügen.

2.3 Das Vanishing Gradient Problem

Literaturverzeichnis

- [1] https://online.science.psu.edu/bisc004_activewd001/node/1907
(abgerufen am 10.05.2017)
- [2] https://de.wikipedia.org/wiki/K%C3%BCnstliches_Neuron (abgerufen am 10.05.2017)
- [3] https://de.wikipedia.org/wiki/Neuronales_Netz#/media/File:Neural_network.svg (abgerufen am 10.05.2017)
- [4] (abgerufen am 10.05.2017)
- [5] (abgerufen am 10.05.2017)
- [6] (abgerufen am 10.05.2017)
- [7] (abgerufen am 10.05.2017)
- [8] (abgerufen am 10.05.2017)
- [9] (abgerufen am 10.05.2017)

Abbildungsverzeichnis

2.1	Schematische Darstellung eines biologischen Neuron [1]	2
2.2	Schematische Darstellung eines künstlichen Neuron [2]	3
2.3	Ein einfaches Neuronales Netz [3]	3