OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Skript

Von: Timo Baumann

Betreff: Neuronale Netze und einige ihrer Anwendungen

Datum: 8. November 2023

Dieses Skript beschreibt neuronale Netze, das Vorgehen beim datenbasierten Training und schließlich einige Anwendungen, die jeweils ganz spezifische Netzarchitekturen nahelegen: Faltungsnetze (Convolutional Neural Networks) für die Bildverarbeitung und rekurrente neuronale Netze für die Sequenzdatenverarbeitung (zum Beispiel von gesprochener Sprache und Text).

1 Neuronale Netze

Neuronale Netze orientieren sich am natürlichen Vorbild der vernetzten Nervenzellen (vor allem im Gehirn): ein Neuron nimmt die Signale von vorangegangenen Neuronen (oder Sinneszellen) mit seinen Dendriten auf, entscheidet, ob die Signale zusammengenommen dafür sorgen, dass das Neuron sich ‘aktiviert’ (dafür sind üblicherweise Signale von mehreren vorangegangenen Zellen in dichter zeitlicher Nähe nötig) und sendet in diesem Fall ein Signal (bzw. mehrere kurze, dicht beieinander liegende Signale in Form von elektrischen Impulsen) durch das eigene Axon an folgende Neuronen (oder Muskelzellen). Die Übertragung zwischen den Neuronen erfolgt durch chemische Prozesse über den ‘synaptischen Spalt’ zwischen Axon und Dendrit. Die ‘Enge’ des Spalts bestimmt letztlich das Gewicht, welches ein Neuron auf die Aktivierung eines Folgeneurons hat und unterscheidet sich zwischen den einzelnen mit einem Neuron verknüpften Vorgängern. Zusätzlich hängt die ‘Reaktivität’ eines Neurons von der Verfügbarkeit der Reagenzien für die chemischen Prozesse im synaptischen Spalt ab.

1.1 Künstliche Neuronen

Das künstliche Neuron im künstlichen neuronalen Netz abstrahiert von den meisten Eigenschaften der tatsächlichen Nervenzelle. Die Prozesse im synaptischen Spalt werden komplett ignoriert und die Zeit für die Ausbreitung der Nervenimpulse im Gehirn bleibt unberücksichtigt. Stattdessen werden ‘zeitlos’ die Aktivierungen der Eingangsneuronen gewichtet und aufsummiert um den Grad der Aktivierung eines Neurons zu bestimmen. Diese Aktivierungen werden stück-für-stück vom Anfang des Netzes und schließlich für das ganze Netz berechnet. Hierfür ist es zwingend erforderlich, dass das Netz keine Rückkopplungen enthält, also azyklisch ist.

Das künstliche Neuron ist also zunächst nicht mehr als eine ‘Stelle’: ein Platzhalter für einen Aktivierungswert mit einem bestimmten Wertebereich. Der Einfachheit halber betrachten wir im Folgenden ausschließlich Neuronen mit dem Wertebereich [0; 1]. (Dies

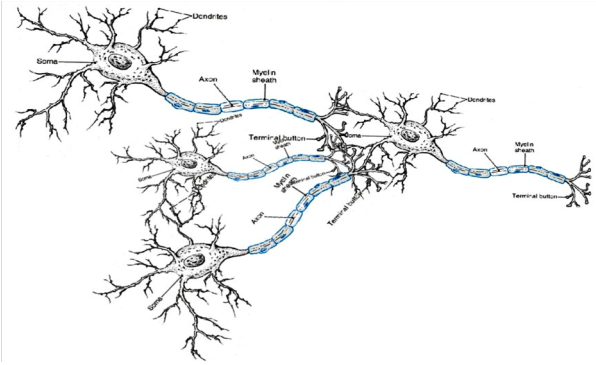

Abbildung 1: Vier natürliche Neuronen: drei links, deren Axone an die Dendriten des vierten rechts stoßen; die Informationsübertragung innerhalb der Neuronen erfolgt elektrisch, durch die Spalte mittels chemischer Reaktionen (Quelle: Chris Manning, Stanford.)

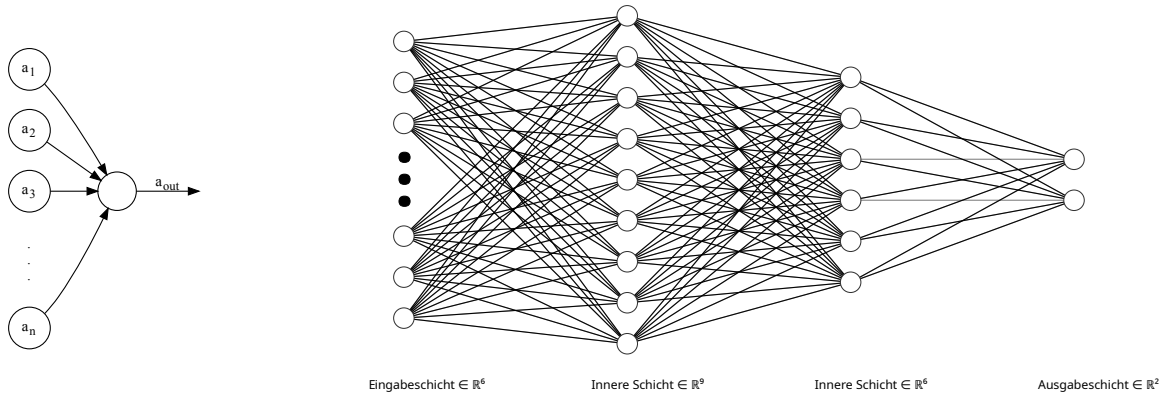

Abbildung 2: Links: Ein einzelnes künstliches Neuron mit den Eingangsaktivierungen a_i und der Ausgangsaktivierung a_{out}. Rechts: Ein in Schichten organisiertes künstliches neuronales Netz.

ist keine Einschränkung und künstliche Neuronen haben häufig auch Wertebereiche von $[-1; 1]$ oder sogar $[-\infty; \infty]$ (wobei die Erläuterungen im folgenden schlechter für den unbeschränkten Wertebereich funktionieren).

Abbildung 2 (links) zeigt ein künstliches Neuron mit Eingangsaktivierungen a_i (also die Werte der Eingangsneuronen). Diese werden durch das Neuron jeweils gewichtet (mit den zu jedem Eingang passenden Gewicht w_i) und anschließend summiert: $\Sigma_{i=0}^{n} a_i w_i$. Die resultierende Summe kann nicht direkt als Aktivierung des Neurons genutzt werden, da sie (soweit für die Gewichte w_i keine weiteren Einschränkungen gemacht werden) im Wertebereich $[-\infty; \infty]$ liegen kann und nicht auf den 'gewünschten' Bereich $[0; 1]$ beschränkt ist. Es ist dafür eine Reskalierung vom unendlichen Wertebereich in einen beschränkten nötig. Diese Stauchung wird zum Beispiel durch die *Sigmoid*-Funktion (auch als *logistische* Funktion bezeichnet) erreicht. Diese und auch alle anderen Funktionen, die diese Stauchung leisten können (und gleichzeitig überall stetig und differenzierbar sind), sind *nichtlinear*. Es zeigt sich im weiteren, dass die nichtlineare Beziehung zwischen Eingängen und Aktivierung eines Neurons für die Leistung der neuronalen Netze ausschlaggebend ist.

1.2 Nichtlinearität und Universal Approximation-Theoreme

Der tatsächliche Grund für die Nutzung nichtlinearer Aktivierungsfunktionen in neuronalen Netzen ist nicht, dass nur solche die Stauchung in einen beschränkten Wertebereich leisten können, sondern die Tatsache, dass sie die Leistung der neuronalen Netze theoretisch beweisbar erklären können.

Viele Aufgaben für maschinelles Lernen und somit auch neuronale Netze können funktional betrachtet werden als die Abbildung aus einem Merkmalsraum (der zum Beispiel die Eigenschaften einer Dateninstanz beschreibt) in einen Zielraum (für Klassifikation von Ziffern zum Beispiel in die Menge $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$). Eine typische Architektur für ein neuronales Netz, das diese Aufgabe löst würde 1 Ausgabeneuron pro Klasse nutzen und die Klasse zuweisen, deren Neuron maximal aktiviert ist (der tatsächliche Zielraum des NNs liest also in $[0; 1]^{10}$). Um jetzt alle möglichen Klassifikationen, bzw. alle möglichen Zusammenhänge zwischen Daten Zielen abbilden zu können, muss ein neuronales Netz in der Lage sein, *beliebige* funktionale Zusammenhänge zwischen diesen Räumen abzubilden, oder jedenfalls beliebig genau zu approximieren. Universal Approximation Theoreme treffen genau hierzu Aussagen, bzw. erklären die Bedingungen, die ein neuronales Netz erfüllen muss um ein *universal function approximator* zu sein.

Es zeigt sich, dass neuronale Netze mit schon einer inneren Schicht (die für beliebig genaue Approximationen dann beliebig viele Neuronen haben muss) solch eine universale Funktionsapproximierung leisten kann, wenn die Sigmoid-Funktion als Aktivierungsfunktion genutzt wird.[1]

1.3 "Deep Learning"

Im Prinzip handelt es sich bei neuronalen Netzen mit einer inneren Schicht um universal function approximators. Zusätzliche Schichten führen nicht dazu, dass das Netz insgesamt *ausdruckskräftiger* würde. Auch für ein Netz mit nur einer inneren Schicht gibt es zu jedem Problem eine Parametrisierung (Gewichte und Schwellwerte), die es löst. Dieser Existenzbeweis heißt aber nicht, dass die Parametrisierung auch bekannt ist.

Empirisch zeigt sich, dass Netze, in denen die Neuronen in vielen inneren Schichten hintereinander organisiert sind, viel leichter trainiert werden können (siehe nächste Abschnitt) als entsprechende Netze mit nur einer inneren Schicht. (Für den fairen Vergleich ist es dabei wichtig, die Anzahl der zu trainierenden Parameter zwischen den Varianten gleichzuhalten.) Diese Erkenntnis ist der Kern des *Deep Learning*. Ob jetzt bereits ein Netz mit zwei inneren Schichten "deep" ist, ist überwiegend eine Entscheidung der Marketingabteilungen der beteiligten Unternehmen.

[1] Mittlerweile gibt es auch Beweise für andere nicht-lineare Aktivierungen.

2 Training eines neuronalen Netzes

2.1 Fehlerrückführung

2.2 Gradientenabstiegsverfahren

2.3 Datenbasierte Vorgehensweise

3 Berechnungsgraph

Darstellung des neuronalen Netzes als Berechnungsgraph

3.1 Denken in Abstraktionsschichten

4 Faltungsnetze für die Bildverarbeitung

5 Rekurrente Netze für die Sprachverarbeitung