Gymnasium Bäumlihof, 5Bb

MATURAARBEIT

Kann der Computer Werbung erkennen?

Bilderkennung mit einem Neuronalen Netzwerk

Georg Schwan

Betreuungsperson
Test1
Korreferent
Test2

Datum

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

1.2 Aufbau der Arbeit

Kapitel 2

Neuronales Netzwerk

2.1 Konzept

Wenn man ein normales Programm schreiben will muss man das Problem in viele kleiner aufteilen, bis der Computer fähig ist es zu lösen. In einem Neuronale Netzwerk sagen wir dem Computer nicht wie er das Problem lösen kann, sondern ein Neuronales Netzwerk versteht das Problem, indem wir es Beispieldaten geben und es daran lernen kann, bis es seine eigene Lösung gefunden hat. Zum Beispiel, wir wollen einem Netzwerk beibringen ob in einem Bild ein Auto vorkommt, dazu geben wir dem Neuronale Netzwerk viele Bilder, mit und ohne Auto. Mit jedem Bild, dass das Neuronale Netzwerk bekommt, lernt es besser wie ein Auto ausschaut.

Das Konzept eines Neuronales Netzwerk ist nicht etwas neuses. Im Jahre 1957 hat Frank Rosenblatt ein erste Idee eines Neuronales Netzwerk vorgestellt, aber erst in den letzten Jahren ist der grosse Hype des Neuronale Netzwerkes ausgebrochen, dies liegt daran, dass man früher nicht die Daten und Rechenleistung hatte, wie sie heute zur verfügung steht.

2.2 Neuron

User Gehirn kann Entscheidungen treffe, da wir billionen von Neuronen haben, die miteinander verbunden sind und sich verständigen können. Aber ein Neuron an sich ist praktisch nutzlos, aber in grosser Anzahl können sie komplexeste Probleme lösen.

Nach dem gleichen Prinzip funktioniert ein Neuronales Netzwerk Es besteht aus vielen Neuronen (daher der Name) die miteinander verbunden sind.

Ein Neuron in einem Neuronales Netzwerk wird als Mathematische funktion definiert wie Abbildung 2.1 verdeutlicht. Ein Neuron hat n verschiedene Eingaben, die als x_j bezeichnet werden und mit einem spezifischen Gewicht w_j multipliziert werden. Die Ausgabe erfolgt indem man alle gewichteten Eingaben, mit einem bias b, addiert und durch eine so genannte Aktivierungsfunktion durchlaufen läst. Als Formel:

$$y = f\left(\sum_{j=1}^{n} x_j w_j + b\right)$$

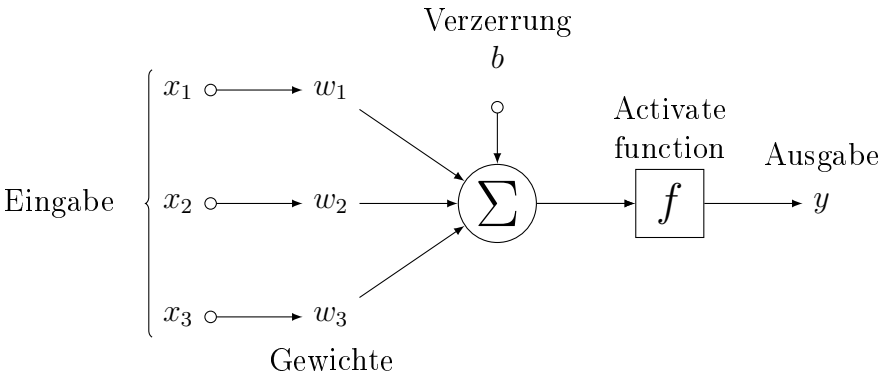

Abbildung 2.1: Einzelner Neuron in einem Neuronalen Netzwerks

Die Gewichte w_j und die Verzerrung b des Neurons sind die Parameter, die angepasst werden und somit das Neuron lernfähig machen.

Quelle: https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f

Eine Aktivierungsfunktion ist nötig, da ohne eine wäre ein Neuronales Netzwerk eine komplett lineare funktion, welches nur lineare Probleme lösen könnte. Die meisten Probleme sind viel komplexer als das man sie linear darstellen könnte und deswegen ist eine aktivierungsfunktion von nötig.

Ein Beispiel für eine Aktivierungsfunktion ist die Sigmoidfunktion, wie sie auf Abbildung 2.2 zu sehen ist. Besonders an dieser Funktion ist, dass sie den Ausgabewert zwischen 0 und 1

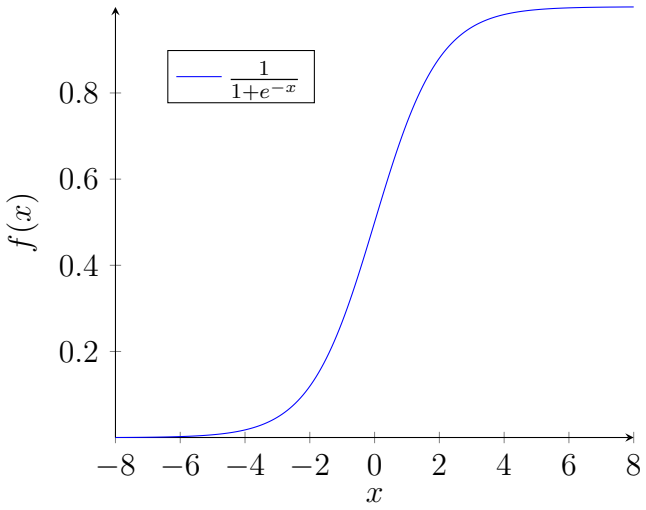

Abbildung 2.2: Sigmoidfunktion

eingrenzt, was uns erlaubt den Ausgabewert des ganzen Netzwerkes besser zu deuten, als wenn der Wert von $-\infty$ bis ∞ gross sein könnte.

Jedes Neuron in einem Netzwerk kann eine andere Aktivierungsfunkton haben.

2.3 Architektur

Wie auch im biologischen Gehirn ist ein Neuron allein nutzlos. Erst wenn man die Neuronen miteinander verbindet kann es komplexe Zusammenhänge modellieren.

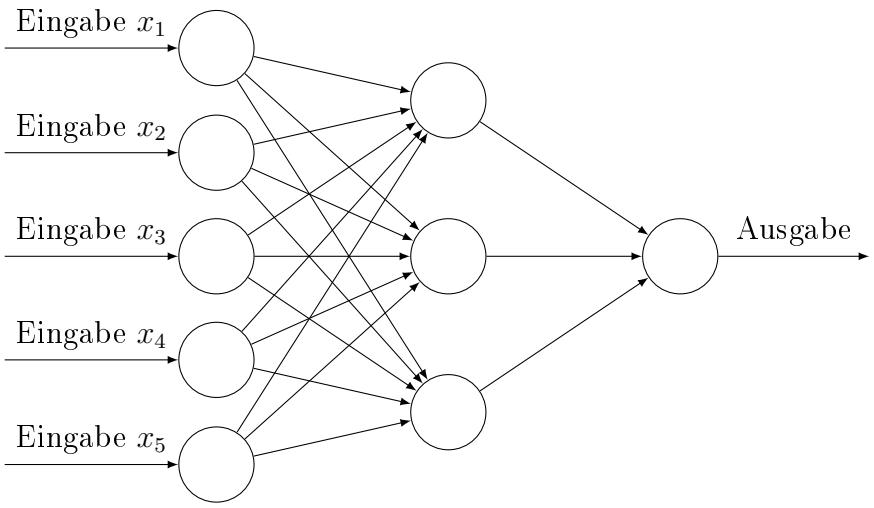

Abbildung 2.3: Mögliche architektur eines Neuronalen Netzwerk

Eine mögliche Architektur kann wie in Abbildung 2.3 ausschauen. Ein Netzwerk wird generell immer in verschiedene Schichten unterteilt. Die linke Schicht wird als eingabe Schicht bezeichnet und die Neuronen in dieser Schicht werden eingabe Neuronen genannt. Analog da zu wird die rechte Schicht ausgabe Schicht genannt, die die ausgabe Neuronen beinhaltet. Die mittleren Schichten, die von der Anzahl variieren können, werden versteckte Schichten genannt. Die Anzahl der Neuronen in jeder Schicht kann auch variieren. Abbildung 2.4 zeigt eine andere mögliche Architektur für ein Netzwerk, welches 2 versteckte Schichten hat. Jeder Neuron

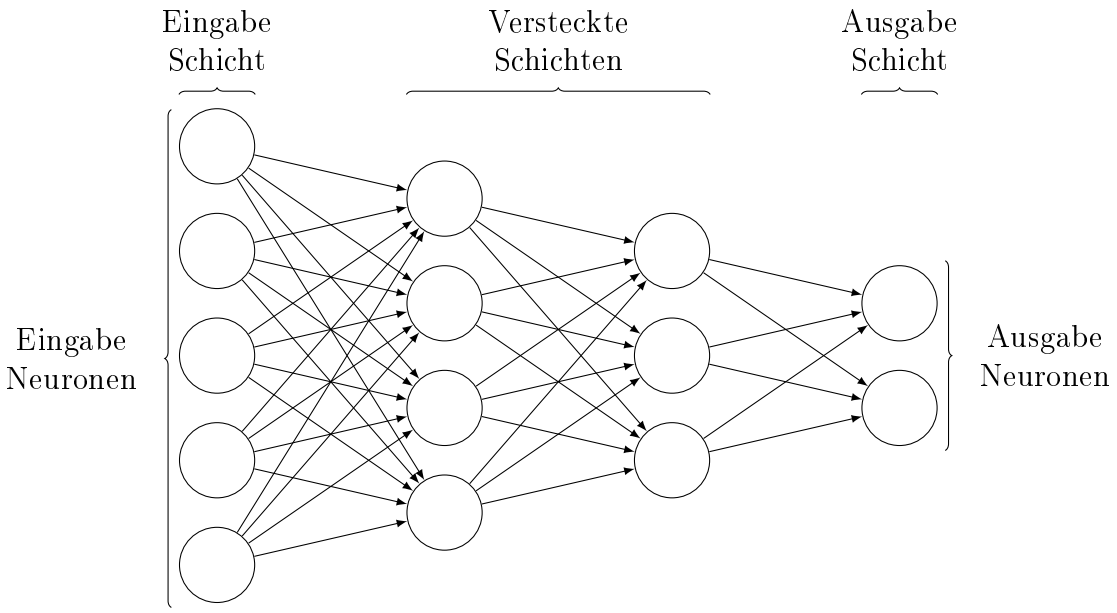

Abbildung 2.4: Neuronales Netzwerk mit 2 versteckten Schichten

von der vorigen Schicht ist mit jedem Neuron der nachfolgenden Schicht verbunden. Dies ist ein klassisches vorwärtsgekoppeltes Netzwerk (im englischen feed forward network). Wichtig zu beachten ist, dass keine Schleifen vorkommen. Es gibt Architekturen in denen Schleifen vorkommen, aber auf diese wird nicht näher eingegangen, da sie für die Bilderkennung nicht relevant sind.

2.3.1 Formel

Um eine allgemeine Formel bestimmen zu können, muss man zuerst die Elemente des Netzwerks benennen. Wir bezeichnen das Gewicht $w^l_{k,j}$ für die Verbindung des k^{ten} Neuron der $(l-1)^{ten}$ Schicht zu dem j^{ten} Neuron der l^{ten} Schicht. Ähnlich dazu bezeichnen wir die Ausgabe des Neurons als a^l_j und der bias des Neurons als b^l_j, Abbildung 2.5 verdeutlicht diese Notation.

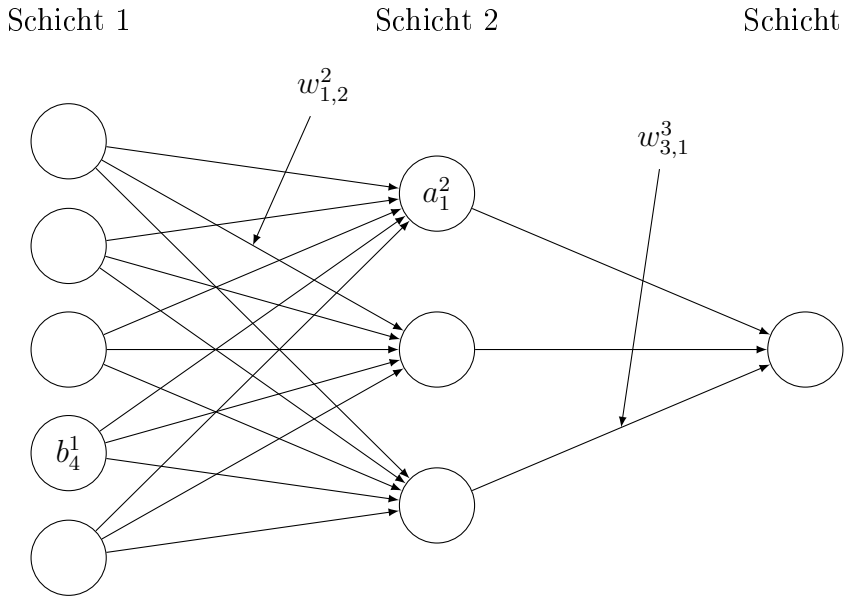

Abbildung 2.5: Bezeichnung der Parameter

Mit dieser Notation kann eine Formel für das Netzwerk aufgestellt werden, welche sehr ähnlich zu der Formel vom Abschnitt 2.2 ist.

$$a^l_j = f\left(\sum_k a^{l-1}_k w^l_{k,j} + b^l_j\right)$$

2.4 Lernen

Bis jetzt ging es nur darum wie ein Neuronales Netzwerk aufgebaut ist. In dem Abschnitt geht wie ein Neuronales Netzwerk, anhand von Daten, lernen kann

2.4.1 Kostenfunktion

Damit ein Netzwerk lernen kann muss man dem Netzwerk zuerst sagen können wie gut oder wie schlecht es gerade ist. Dazu definieren wir eine Kostenfunktion C, die von allen Gewichten w und allen biases b abhängig ist. Das Netzwerk wird als Funktion $y(x)$ bezeichnet. Den Eingabewert wird als x bezeichnet mit dem dazugehörige Label l. Beachte, dass x und l Vektoren sind. Zum Beispiel würde ein Bild, das 10x10 Pixel gross ist, einen $(10 * 10 = 100)$ 100-dimensionaler Vektor bekommen und die Label, wenn es 3 verschiedene Ergebnisse gibt, einem 3-dimensional Vektor, der z.B so ausschaut: $(0, 1, 0)$

$$C(w, b) = |y(x) - l|^2$$

Das Ziel des Netzwerkes ist diese Kostenfunktion zu minimieren, bis idealerweise $C \approx 0$, dies geschieht wenn die Ausgabe des Netzwerks und des Label sehr ähnlich sind.

2.4.2 Gradient Descent

Um diese Kostenfunktion zu minimieren wird ein Algorithmus names Gradient Descent benutzt. Das Konzept basiert darauf, dass man eine Funktion, in abhängigkeit einer Variablen, ableiten kann und so die Steigung and diesem Punkt berechnen kann und die Variable richtung Minimum anpasst.

Bild Einfügen

Zum Beispiel hat man eine Kostenfunktion $C(a, b)$ die von a und b abhängig ist. Wenn man diese Graphisch abbildet, sieht es wie auf Abbildung **ref**.

Die Variablen a und b werden am Anfang zufällige Werte zugeteilt und das Ziel ist sie so anzupassen, dass man das Minimum der Kostenfunktion findet. Um das Minimum zu finden kann man sich einen Ball vorstellen, der in das Minimum herunterrollt. Um dies zu berechnen muss man die Steigung, mithilfe einer Ableitung, herausfinden und die Variable in die gegensätzliche Richtung bewegen, wie auf Abbildung **ref** zu sehen ist.

Bild Einfügen

Die Bewegung als Formel sieht so aus:

$$\begin{aligned} a \to a' &= a - \mu \frac{\partial C}{\partial a} \\ b \to b' &= b - \mu \frac{\partial C}{\partial b} \end{aligned}$$

wobei μ eine kleine positive Zahl (learning rate genannt) ist, die die Geschwindigkeit der Bewegung steuert. Ausserdem beachte, dass der Ball keine Beschleunigung hat. Wenn man diese Formel nun immer wieder anwendend gelangt man zum Minimum der Kostenfunktion.

Der Algorithmus funktioniert auch bei mehr als nur 2 Variablen und sieht fürs Neuronale

Netzwerk identisch aus.

$$w_{k,j}^{l} \to w_{k,j}^{l'} = w_{k,j}^{l} - \mu \frac{\partial C}{\partial w_{k,j}^{l}}$$
$$b_{j}^{l} \to b_{j}^{l'} = b_{j}^{l} - \mu \frac{\partial C}{\partial b_{j}^{l}}$$

2.4.3 Backpropagation

Kapitel 3

Lösungsansatz

Kapitel 4

Umsetzung

Kapitel 5

Reflexion

Abbildungsverzeichnis

Ehrlichkeitserklärung

Die eingereichte Arbeit ist das Resultat meiner persönlichen, selbstständigen Beschäftigung mit dem Thema. Ich habe für sie keine anderen Quellen benutzt als die in den Verzeichnissen aufgeführten. Sämtliche wörtlich übernommenen Texte (Sätze) sind als Zitate gekennzeichnet.

Insert Datum