



Implementierung eines neuronalen Netzwerkes zur Zeichenerkennung in SetIX

Studienarbeit

Studiengang Angewandte Informatik

Duale Hochschule Baden-Württemberg Mannheim

von

Lucas Heuser und Johannes Hill

Bearbeitungszeitraum:	05.09.2016 - 29.05.2017
Matrikelnummer, Kurs:	-, TINF14AI-BI
Matrikelnummer, Kurs:	-, TINF14AI-BI
Ausbildungsfirma:	Roche Diagnostics GmbH, Mannheim
Abteilung:	Scientific Information Services
Betreuer der DHBW-Mannheim:	Prof. Dr. Karl Stroetmann

UNTERSCHRIFT DES BETREUERS

Eidesstattliche Erklärung

Hiermit erklären wir, dass wir die vorliegende Arbeit mit dem Thema

Implementierung eines neuronalen Netzwerkes zur Zeichenerkennung in SetIX

selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Mannheim, den 17. März 2017

LUCAS HEUSER

JOHANNES HILL

Inhaltsverzeichnis

1	Einleitung	1
1.1	Studienarbeit and DHWB	1
1.2	GitHub Link	1
1.3	Was ist künstliche Intelligenz	1
1.4	Aktuelle Relevanz/Themen von neuronalen Netzen	1
1.5	Ziel der Arbeit	1
1.6	Aufbau der Arbeit	1
2	Theorie	2
2.1	SetlX	2
2.2	MNIST	2
2.3	Perceptrons	2
2.4	Sigmoid Neurons	3

Abbildungsverzeichnis

2.1	Percetron mit den Eingaben x_1, x_2, x_3 und der Ausgabe <i>output</i>	2
2.2	Unterschiedliche Möglichkeiten der Entscheidungsfindung.	3

Kapitel 1

Einleitung

1.1 Studienarbeit and DHWB

1.2 GitHub Link

1.3 Was ist künstliche Intelligenz

1.4 Aktuelle Relevanz/Themen von neuronalen Netzen

1.5 Ziel der Arbeit

Die menschliche Wahrnehmung ist

1.6 Aufbau der Arbeit

Kapitel 2

Theorie

2.1 SetIX

2.2 MNIST

2.3 Perceptrons

Ein Perceptron ist ein mathematisches Modell zur Abbildung eines künstlichen Neurons in einem Netzwerk. Es wird für die Entscheidungsfindung herangezogen, indem verschiedene Aussagen abgewägt werden. Hierbei nimmt das Perceptron eine Menge von Eingaben x_n mit $n \in \{1, \dots, n\}$ und berechnet einen einzigen binären Ausgabewert (siehe Abb. 2.1). Für die Berechnung der Ausgabe werden

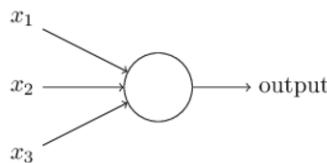


Abbildung 2.1: Perceptron mit den Eingaben x_1, x_2, x_3 und der Ausgabe *output*.

sogenannte *Weights* w_n mit $n \in \{1, \dots, n\}$ eingeführt, welche die Gewichtung der jeweiligen Eingabe ausdrücken. Der Ausgabewert *output* wird mittels der gewichteten Summe $\sum_j w_j x_k$ und einem definierten Grenzwert *threshold* bestimmt.

$$\text{output} := \begin{cases} 0 & \text{falls } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{falls } \sum_j w_j x_j > \text{threshold} \end{cases} \quad (2.1)$$

Werden die *Weights* und der *Threshold* variiert, entstehen unterschiedliche Modelle zur Entscheidungsfindung. Hierbei ist zu beachten, dass eine Minimierung des *Thresholds* den binären Ausgabewert 1 mit einer höheren Wahrscheinlichkeit bedingt.

Der Aufbau des Netzwerks leitet sich aus den unterschiedlichen Modellen der Entscheidungsfindung ab und wird mit Hilfe der Perceptrons abgebildet (siehe Abb. 2.2). Eine Entscheidungsmöglichkeit wird hierbei durch das Perceptron dargestellt. Weiterhin wird eine Spalte von Perceptrons als *Layer* bezeichnet. Der erste Layer fällt Entscheidungen auf Basis der Eingabewerte, indem er diese abwägt. Jedes Perceptron des zweiten Layers hingegen, wägt für die Entscheidungsfindung die Resultate des

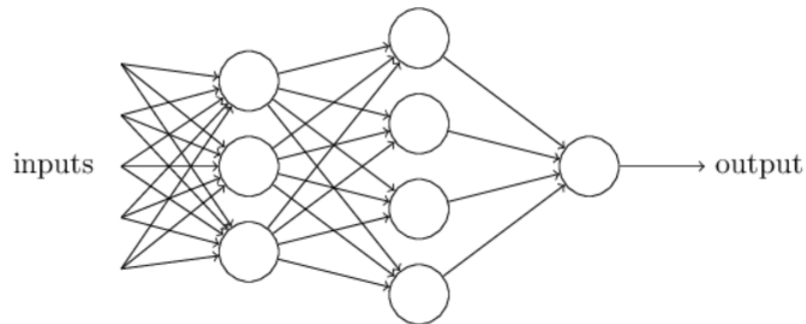


Abbildung 2.2: Unterschiedliche Möglichkeiten der Entscheidungsfindung.

ersten Layers ab. Ein Perceptron auf dem zweiten Layer kann somit eine Entscheidung auf einer abstrakteren und komplexeren Ebene durchführen. Auf diese Weise kann sich ein vielschichtiges Netzwerk von Perceptrons in ein anspruchsvolles Modell zur Entscheidungsfindung entwickeln.

2.4 Sigmoid Neurons

Für die Entwicklung lernender Algorithmen in einem Netzwerk mit Perceptrons, fällt unsere Betrachtung auf das Beispiel der Erkennung von handschriebenen Zahlen. Die Eingabe für das Netzwerk könnten die Raw Pixeldaten der eingescannten Bilder darstellen, welche die handschriebenen Zahlen abbilden. Das Ziel an dieser Stelle ist, dass das Netzwerk anhand der *Weights* und *Biases* lernt eine korrekte Klassifizierung der Zahlen vorzunehmen.