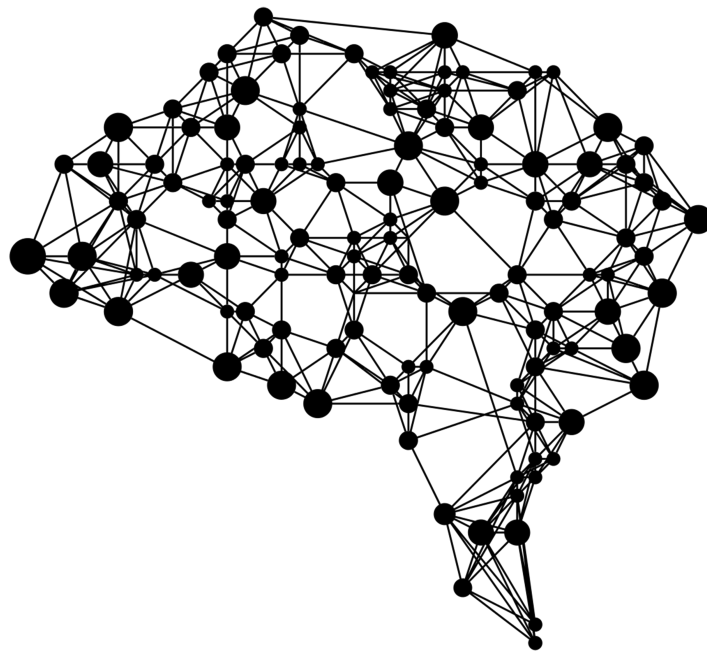


# Machine Learning in Smartphone Apps

ASGSG Informatik, 2020/2021

Phillip Bronzel 18. Dezember 2020



cleanpng 2020

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst, dass ich keine anderen Quellen und Hilfsmittel als die angegebenen benutzt und die Stellen der Arbeit, die anderen Quellen dem Wortlaut oder Sinn nach entnommen sind, in jedem einzelnen Fall unter Angabe von Quellen kenntlich gemacht habe.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
1.1	Wahl des Themas . . . . .	2
1.2	Ziel der Arbeit . . . . .	2
<b>2</b>	<b>Neuronale Netzwerke</b>	<b>3</b>
2.1	Geschichte . . . . .	3
2.2	Aufbau . . . . .	4
2.2.1	Erstellung eines NN anhand eines Beispiels . . . . .	5
2.3	Funktionsweise . . . . .	5
2.3.1	Trainieren - Backpropagation . . . . .	5
	<b>Literaturverzeichnis</b>	<b>6</b>

# 1 Einführung

Im Enterprise und Forschungsbereich spielt Machine Learning schon seit vielen Jahren eine bedeutende Rolle. Doch wie kann es dem Endnutzer, zum Beispiel in mobilen Apps, weiterhelfen?

## 1.1 Wahl des Themas

Seitdem ich im Jahr 2017 meinen ersten richtigen Kontakt mit der Programmierung von Microcontrollern (Arduinos) hatte, habe ich mich stark für die Entwicklung von Software interessiert. Dies war ein guter Einstieg, da man dort schnell und recht einfach Ergebnisse, wie zum Beispiel eine blinkende LED, erzielt.

Auch habe ich mich seitdem immer für die „neuen großen Technologien“ wie Blockchain oder Machine Learning interessiert. Zum Thema Machine Learning habe ich zuvor noch nicht viel gemacht, daher ergriff ich die Chance dieses Jahr meine Facharbeit über dieses Thema zu schreiben.

AI is profound, and we are at a point—and it will get better and better over time—where the GPU is getting so powerful there’s so much capability to do unbelievable things. What all of us have to do is to make sure we are using AI in a way that is for the benefit of humanity, not to the detriment of humanity.<sup>1</sup>

Ich persönlich finde dieses Zitat sehr wichtig; es ist jetzt über 3 Jahre alt und bis heute hat sich enorm viel in diesem Bereich getan. Wir haben nun GPU’s, welche speziell auf mathematische Berechnungen mit Tensoren optimiert sind und so das Trainieren von Neuronalen Netzen um ein Vielfaches beschleunigen.<sup>2</sup>

Des weiteren ist es mir, genauso wie Cook, wichtig, diese mächtige Technologie nicht zu missbrauchen<sup>3</sup>, sondern gute Dinge mit ihr zu schaffen: wie beispielsweise im Bereich der Medizin. In diesem Bereich wurden schon viele beachtliche Anwendungszwecke gefunden, so hat Google’s Tochterfirma DeepMind im Dezember 2020 eine KI<sup>4</sup> präsentiert, welche das Falten von Proteinen akkurat prognostizieren kann; dies war vorher nur sehr langsam und deutlich ungenauer möglich.<sup>5</sup>

## 1.2 Ziel der Arbeit

Mein persönliches Ziel ist es, mehr über den Aufbau von Neuronalen Netzen und die Funktionsweise von Machine Learning zu lernen. Außerdem möchte ich auch ein

---

<sup>1</sup>Byrnes 2017, Tim Cook (CEO von Apple) In einem Interview mit MIT Technology Review

<sup>2</sup>Hebert 2020, NVIDIA Grafikprozessoren mit integrierten Tensor Kernen

<sup>3</sup>Beispiel: Autonome Waffen, wie Drohnen, welche Ziele autonom erfassen können

<sup>4</sup>Künstliche Intelligenz

<sup>5</sup>Rettner 2020

praktisches Ergebniss haben, dafür habe ich im Kapitel !TODO! eine App entwickelt, welche dem Nutzer mehr Informationen über Produkte beim einkaufen liefern soll.

## 2 Neuronale Netzwerke

### 2.1 Geschichte

Im Jahr 1943 wurde die erste Arbeit darüber geschrieben, wie Neuronen im Gehirn funktionieren könnten und die Autoren Warren McCulloch und Walter Pitts experimentierten sogar damit diese mit elektronischen Schaltkreisen nachzubauen.<sup>6</sup>

In den 1950er Jahren haben Forscher von IBM daran gearbeitet ein NN<sup>7</sup> mit einem Computer zu simulieren. Der Versuch scheiterte allerdings.<sup>8</sup>

Immer wieder gab es kleinere Forschungsprojekte, ein sehr großer Durchbruch war aber 1975 die Entwicklung eines „Backpropagation“ Algorithmus durch den Wissenschaftler Paul Werbos. Ähnliche Algorithmen wurden wiederholt und unabhängig entwickelt, aber Werbos’ Algorithmus war der erste mit großer Bedeutung.<sup>9</sup> Das Prinzip des Algorithmus wird auch heute noch verwendet, es ist dieser Algorithmus der dem Neuronalen Netzwerk das selbstständige Lernen ermöglicht.<sup>10</sup>

In 1998 veröffentlichte Yann LeCun und sein Team eine Arbeit über die Anwendung eines „Convolutional Neural Networks“<sup>11</sup> zur Erkennung von geschriebenen Zeichen in einem Dokument.<sup>12</sup> Diese Arbeit gilt als Ursprung des, für beispielsweise Bilderkennungs Software gut geeignete, CNNs und Weiterentwicklungen werden auch heute noch verwendet.

Obwohl ein großes Potenzial erkannt wurde, war es über die nächsten Jahre wieder recht still. Der nächste große Durchbruch passierte in 2012 als Geoffrey Hinton ein Modell entwickelte, was die Fehlerquote in einer öffentlichen Challenge für Bilderkennung beinahe halbierte.<sup>13</sup> Der Grund dafür waren mehrere fundamentale Neuerungen aus dem Bereich Deep Learning; die wahrscheinlich größte Änderung: Starke Parallelisierung des Backpropagation-Prozesses, durch Verschiebung der Last von der CPU auf die GPU. Aufgrund der starken Überlegenheit eines Grafikprozessors in parallelisierten Prozessen, wie die benötigten Tensormultiplikationen durch die deutlich größere Anzahl an (dafür schwächeren) Kernen im Vergleich zu einer herkömmlichen CPU, kann ein NN mehrere hundertmal schneller trainiert werden.

Heute gibt es (vergleichsweise) simple Frameworks, wie das im Jahr 2015 erschienene TensorFlow oder PyTorch aus 2016, welche das erstellen, trainieren und

---

<sup>6</sup>Warren McCulloch 1943

<sup>7</sup>Kurzform für „Neuronales Netzwerk“, wird ab jetzt weiterhin verwendet.

<sup>8</sup>Roberts unbekanntes Jahr, Absatz 3

<sup>9</sup>Werbos 1975

<sup>10</sup>Genaueres in Kapitel 2.3

<sup>11</sup>Ab jetzt als CNN bezeichnet

<sup>12</sup>Yann LeCun und Haffner 1998

<sup>13</sup>Geoffrey E. Hinton 2012

verwenden von NN enorm vereinfachen. Ihr Funktionsumfang wächst durch die große Open-Source Community ständig.

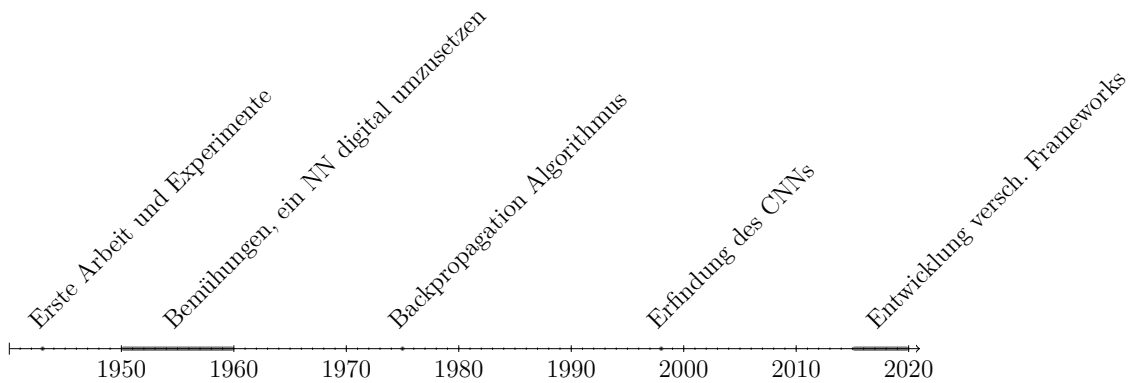


Abbildung 1: Zeitstrahl von 1940 bis 2020 mit den wichtigsten Ereignissen der Entwicklung künstlicher Neuronaler Netzwerke

## 2.2 Aufbau

In Abbildung 2 sieht man den Aufbau eines herkömmlichen künstlichen Neuronalen Netzwerks, so wie es noch vor 40 Jahren verwendet wurde. In der Grafik erkennt man drei Layer mit einer x-beliebigen Anzahl Neuronen, welche untereinander mit jeweils allen Neuronen der vorigen und nächsten Layer verbunden sind. Jede Verbindung hat ein weight zwischen  $-1 \leq w \leq 1$  und auch jedes Neuron hat einen Bias, ebenfalls zwischen  $-1 \leq b \leq 1$ . Die Anzahl der Hidden Layer kann an das Ziel angepasst und ausgewählt werden und auch die Anzahl der einzelnen Neuronen ist erstmal beliebig, als Faustregel für gute Ergebnisse gilt aber:

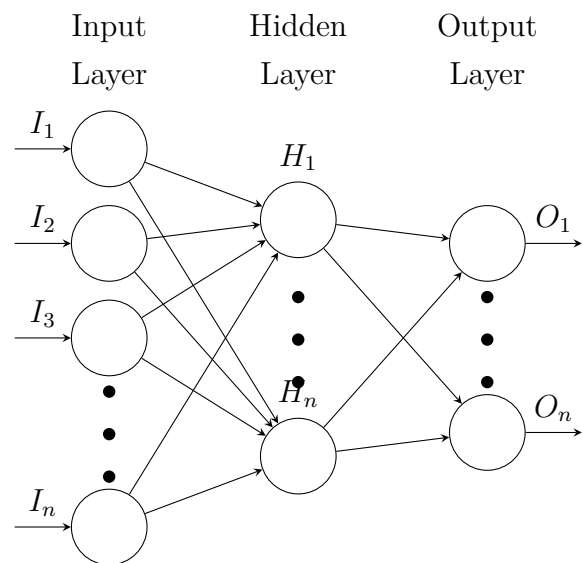


Abbildung 2: Vereinfachter Aufbau eines NN

- Die Anzahl der Neuronen in dem Hidden Layer sollte zwischen der Größe des Input und Output Layers liegen.
- Die Anzahl der Neuronen in dem Hidden Layer sollte etwa  $\frac{2}{3}$  der Größe des Input Layers plus der Größe des Output Layers entsprechen.

- Die Anzahl der Neuronen in einem Hidden Layer sollte weniger als die Hälfte der Größe des Input Layers sein.<sup>14</sup>

### 2.2.1 Erstellung eines NN anhand eines Beispiels

Als Beispiel für ein NN, welches darauf ausgelegt ist, geschriebene Ziffern aus Bildern mit 16x16 Pixeln und nur Graustufen zu erkennen wäre dann: Ein Input Layer mit  $16^2$  Neuronen, jeweils für jeden Pixel, welcher einen Wert zwischen 0 (komplett weiß) und 1 (komplett schwarz) haben kann, eines. Genau 10 Neuronen im Output Layer, für jedes Zahlzeichen eines. Schließlich muss die Anzahl der Hidden Layer und Neuronen festgelegt werden. Ich wähle als Beispiel 2 Layer mit jeweils 16 Neuronen, die Neuronen-Anzahl kann aber auch unterschiedlich sein.

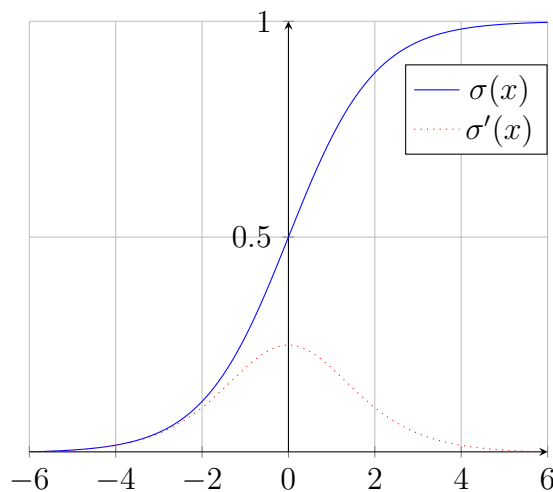
```

1 def fib(n):
2     a, b = 0, 1
3     while a < n:
4         print(a, end=' ')
5         a, b = b, a+b
6     print()
7 fib(1000) # Druckt alle Fibonacci Zahlen bis 1000

```

Python Code um ...

## 2.3 Funktionsweise



### 2.3.1 Trainieren - Backpropagation

<sup>14</sup>Heaton 2017, Alle drei Faustregeln

## Literaturverzeichnis

- Byrnes, Nanette (2017). *Tim Cook: Apple Isn't Falling Behind, It's Just Not Ready to Talk About the Future*. URL: <https://www.technologyreview.com/2017/06/14/4525/tim-cook-apple-isnt-falling-behind-its-just-not-ready-to-talk-about-the-future/> (besucht am 05. 12. 2020).
- cleanpng (2020). *Deep learning maschinelles lernen, Künstliche neuronale Netzwerk der informatik Convolutional neural network - Vernetzung*. URL: <https://de.cleanpng.com/png-s07s8u> (besucht am 17. 11. 2020).
- Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. URL: <https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf> (besucht am 17. 12. 2020).
- Heaton, Jeff (2017). *Heaton Research, The Number of Neurons in the Hidden Layers*. URL: <https://www.heatonresearch.com/2017/06/01/hidden-layers.html> (besucht am 17. 12. 2020).
- Hebert, Chris (2020). *Accelerating WinML and NVIDIA Tensor Cores*. URL: <https://developer.nvidia.com/blog/accelerating-winml-and-nvidia-tensor-cores/> (besucht am 05. 12. 2020).
- Rettner, Rachael (2020). *AI system solves 50-year-old protein folding problem in hours*. URL: <https://www.livescience.com/artificial-intelligence-protein-folding-deepmind.html> (besucht am 05. 12. 2020).
- Roberts, Eric S. (unbekanntes Jahr). *History: The 1940's to the 1970's*. URL: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html> (besucht am 14. 12. 2020).
- Warren McCulloch, Walter Pitts (1943). *A logical calculus of the ideas immanent in nervous activity*.
- Werbos, Paul (1975). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*.
- Yann LeCun Léon Bottou, Yoshua Bengio und Patrick Haffner (1998). *Gradient-Based Learning Applied to Document Recognition*. URL: <http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf> (besucht am 17. 12. 2020).