

Python: Grundlagen neuronaler Netze

Von Jonah Lüdemann

22.05.2020

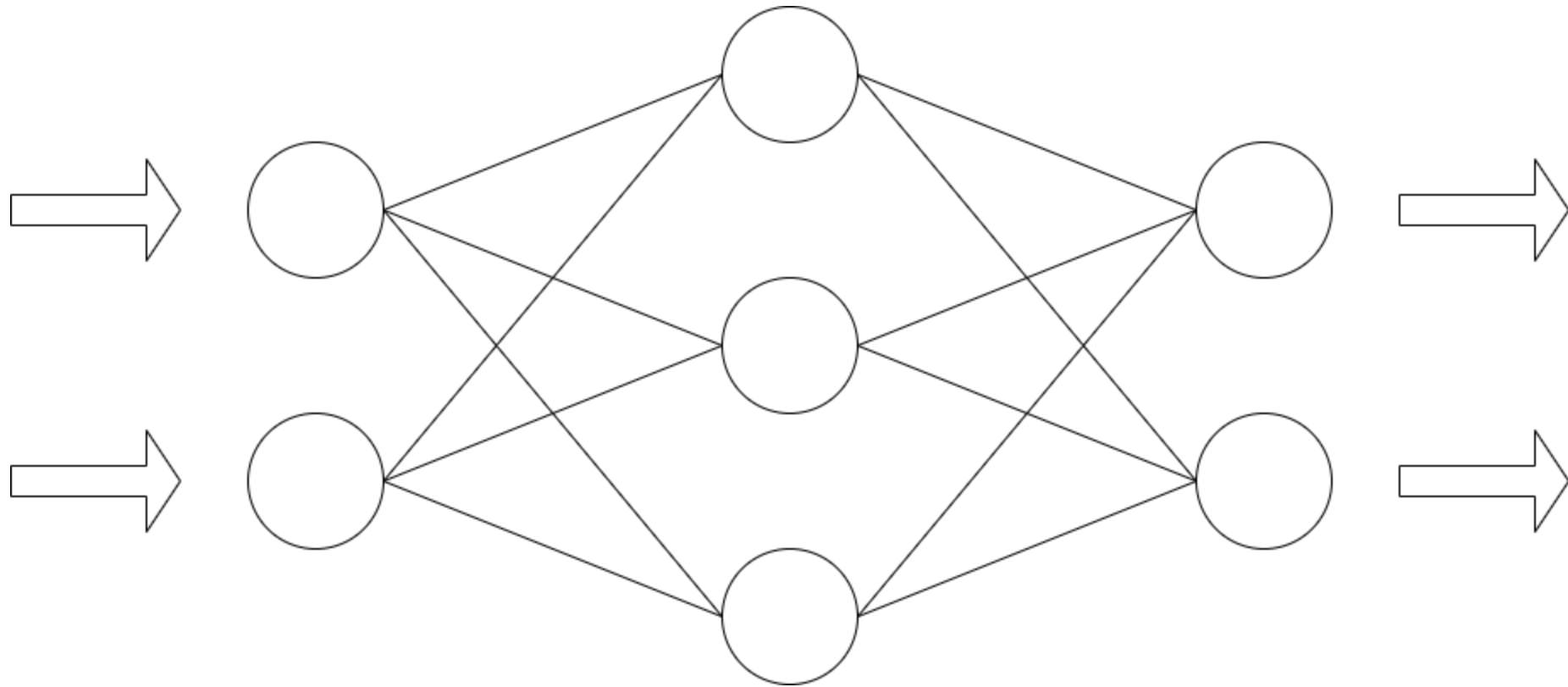
Gliederung

- Was ist ein neuronales Netz
- Aufbau von neuronalen Netzen
- Lernmethoden
- Vor- und Nachteile
- Anwendung
- Frameworks
- Quellen

Was ist ein neuronales Netz

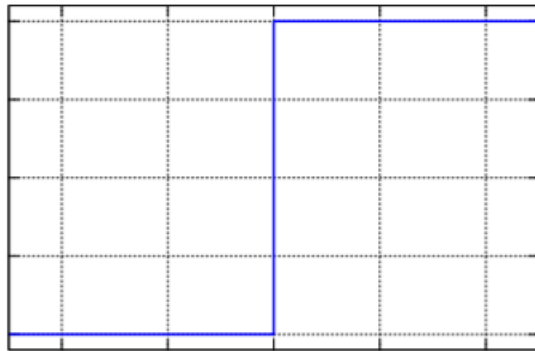
- Abstrahierte Nachbildung eines biologischen Nervensystems
- Eine Verbindung von künstlichen Neuronen
- Es ermöglicht einem Computer selbstständig zu lernen

Aufbau von neuronalen Netzen

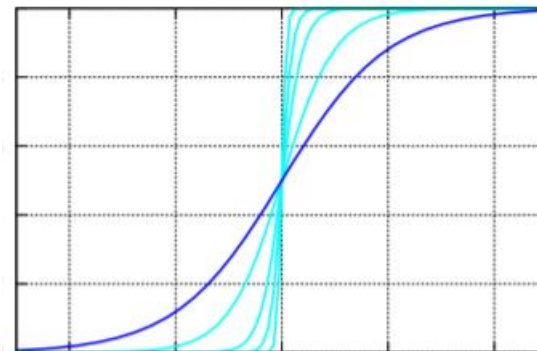


Neuronen

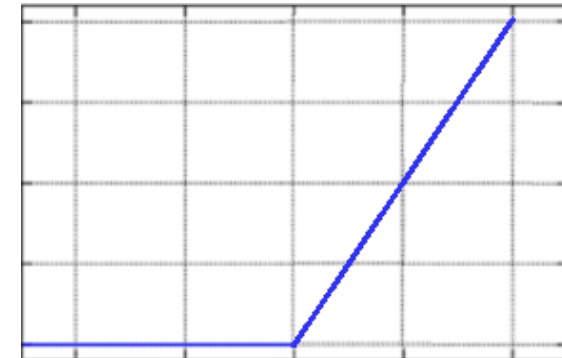
- Propagierungsfunktion
- Aktivierungsfunktion



Heaviside-Funktion $x \rightarrow \begin{cases} 0 & : x < 0 \\ 1 & : x \geq 0 \end{cases}$



Fermifunktion $\frac{1}{1 + e^{\frac{-x}{T}}}$



Rectifier $x \rightarrow \max(0, x)$

- Ausgabefunktion

Verbindungen zwischen Neuronen

- Jede Verbindung hat eine Kantengewichtung zwischen -1 und 1
- Verbindungen leiten die Ausgabe von einem Neuron zum Nächsten
- Der Ausgabewert wird mit der Kantengewichtung multipliziert

Das Bias-Neuron

- Spezielles Neuron, welches immer aktiv ist
- Kann mit jedem beliebigen Neuron des neuronalen Netzes verbunden sein
- Stellt den Schwellenwert zur Aktivierung eines Neurons dar

FeedForward-Netz

- besteht aus drei Schichten
 - Eingabeschicht
 - Beliebig viele versteckte Schichten
 - Ausgabeschicht
- Neuronen sind verbunden mit der nächsten Schicht
- Es ist auch möglich, dass Verbindungen als Short-Cuts Schichten überspringen

Rückgekoppelte Netze

- Neuronen können sich durch ihre Ausgabe selber beeinflussen
- Bei direkter Rückkopplung über eine Verbindung von einem Neuron zu sich selber
- Bei indirekter Rückkopplung über Umwege
- Es gibt nicht unbedingt festgelegte I/O-Neuronen

Vollständig verbundene Netze

- Jedes Neuron des neuronalen Netzes ist mit jedem andern verbunden
- In der Regel gibt es keine direkte Rückkopplung
- Keine definierten Schichten, dementsprechend kann jedes Neuron Input oder Output werden

Lernmethoden

- Neue Verbindungen entwickeln
- Vorhandene Verbindungen löschen
- Verbindungsgewichte ändern
- Schwellenwerte ändern
- Ein oder mehr Neuronenfunktionen ändern
- Neue Neuronen entwickeln
- Vorhandene Neuronen löschen

Supervised Learning

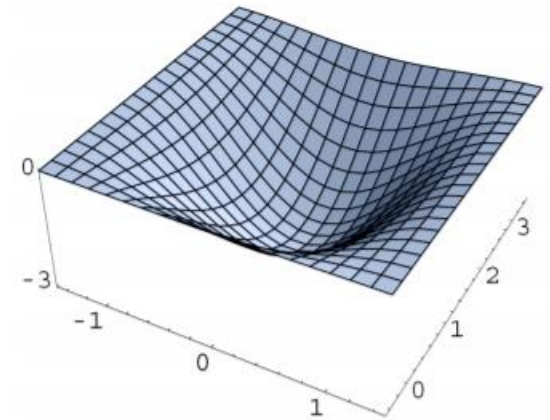
- Zum eingegebenen Lernmaterial erhält das neuronale Netz die erwartete Lösung
- Unterschied zwischen *online* und *offline* lernen
 - online: Änderungen werden nach jedem Beispiel durchgeführt
 - offline: Änderungen werden erst nach einer gewissen Menge an Beispielen durchgeführt

Verlustfunktion

- Ausgabewert a_i eines Neurons und erwartete Ausgabe e_i
- Verlustfunktion = $\sum_i^n (a_i - e_i)^2$
- Die Verlustfunktion ist eine vieldimensionale Funktion
- Je kleiner die Summe, desto genauer entspricht die Ausgabe des Netzes der erwarteten Ausgabe

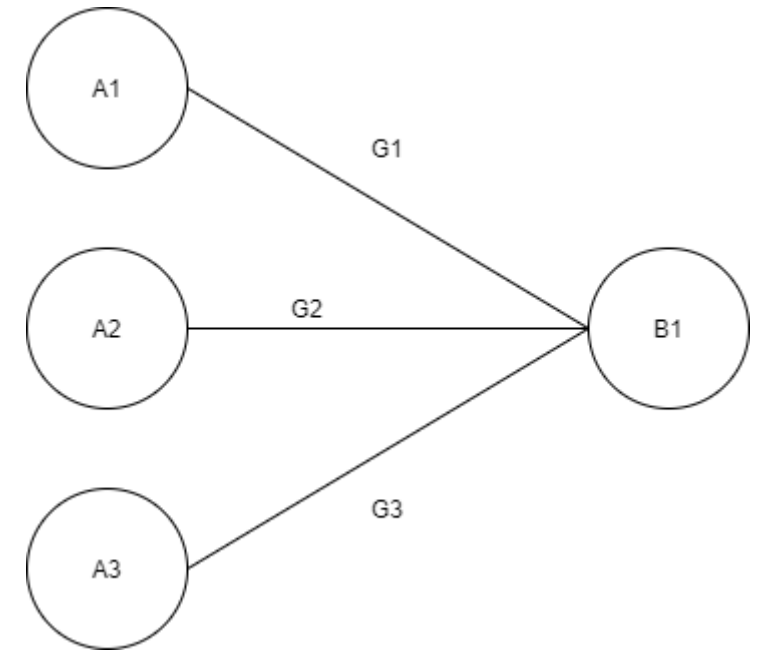
Gradientenabstieg

- Der Gradient ist für jeden Punkt einer Funktion als Vektor g in die Richtung des steilsten Anstiegs definiert
- Der negative Gradient ist folglich der Vektor $-g$ des steilsten Abstiegs
- Der Betrag $|g|$ gibt den Steigungsgrad an
- Beim Gradientenabstieg folgt man dem negativen Gradienten und macht dabei die Schritte proportional zum Steigungsgrad



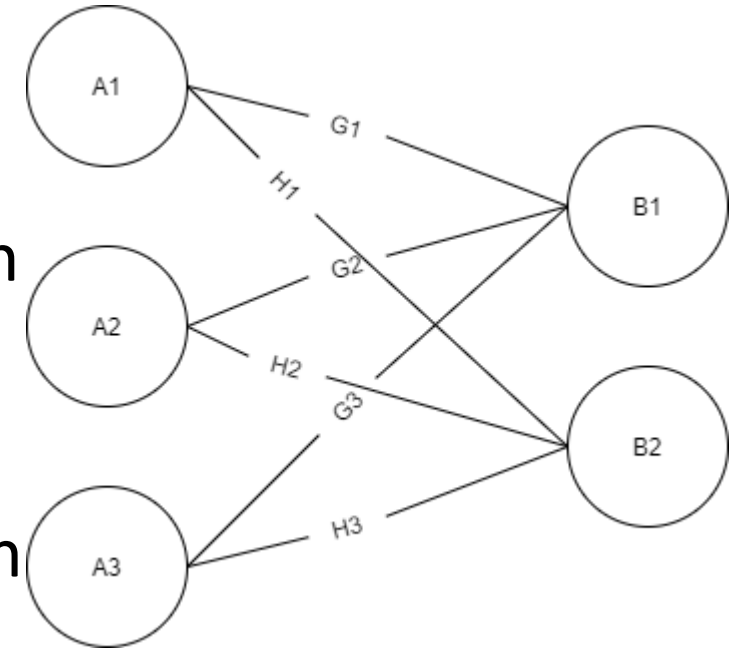
Backpropagation

- $B_1 = f(A_1G_1 + A_2G_2 + A_3G_3)$
wobei f die Aktivierungsfunktion ist
- B_1 ändert sich, wenn entweder das Gewicht G_n oder Ausgabe A_n geändert wird
- Das Gewicht G_n kann direkt geändert werden
- Die Ausgabe A_n kann nicht direkt geändert werden, sondern muss rekursiv wie B_1 eingesetzt werden und so geändert werden

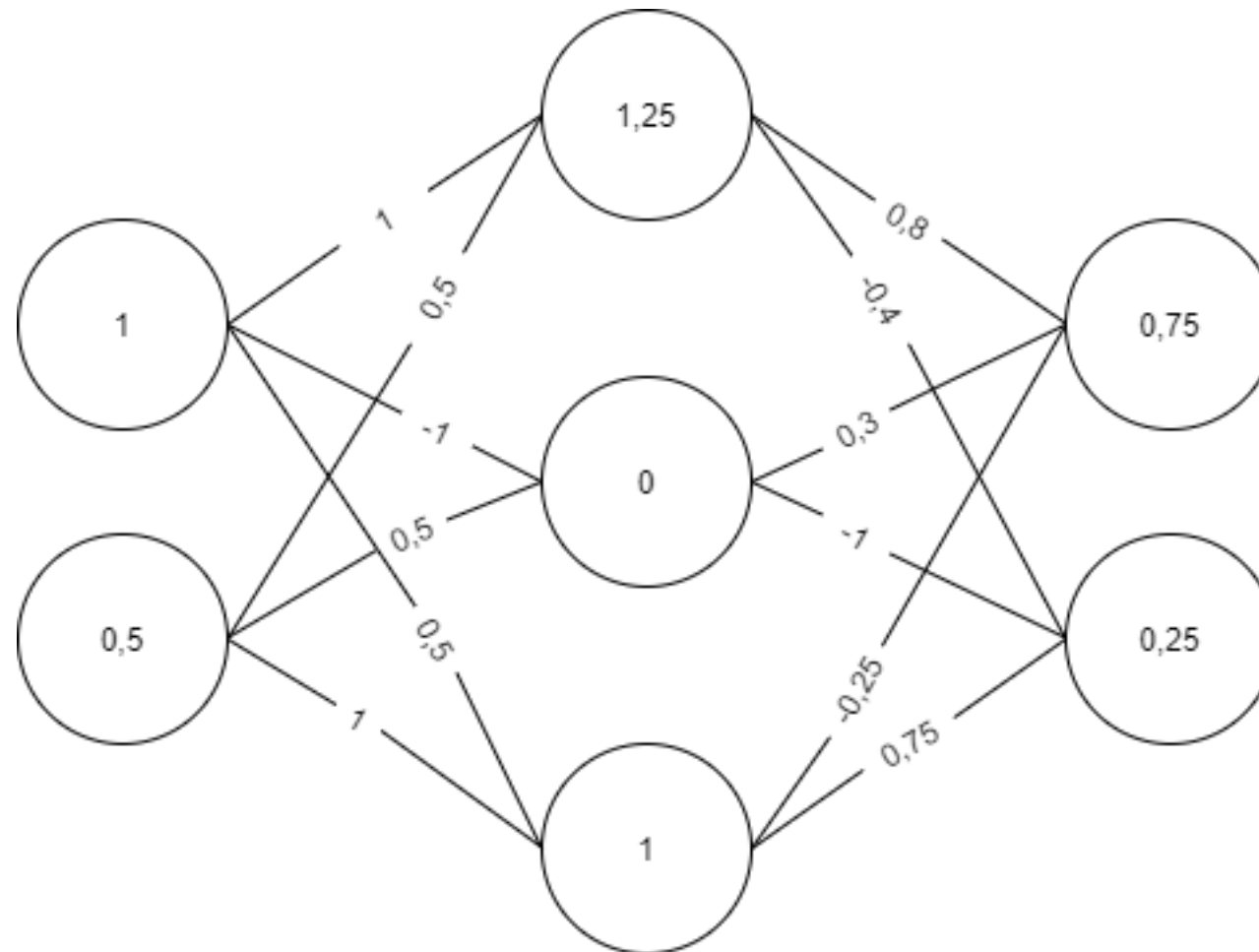


Backpropagation

- Bei mehreren Neuronen der Schicht B wird für die gewünschten Änderungen an einem Neuron A_n der Durchschnitt gebildet
- Der gleiche Prozess wird dann auf die Neuronen der Schicht A mit der davor liegenden Schicht durchgeführt
- Durch die Verlustfunktion wird die Höhe der Änderungen bestimmt



Beispielnetz



Unsupervised Learning

- Das neuronale Netz erhält zum Lernmaterial keine Informationen zur Korrektheit der Ergebnisse
- Das neuronale Netz muss selbstständig Muster erkennen und die gegebenen Daten einordnen

Vorteile

- Ein neuronales Netz kann ohne Vorwissen über bestehende Zusammenhänge arbeiten
- Mit Hilfe von dem richtigen Lernmaterial kann es sehr gut verallgemeinern und so Unbekanntes einordnen
- Es ist tolerant gegenüber unvollständigen oder verzerrten Daten

Nachteile

- Die Vorgänge in einem neuronalen Netz sind meist sehr schwer nachzuvollziehen
- Es ist fast unmöglich den global optimalen Zustand eines neuronalen Netzes zu finden
- Es werden viele Trainingsdaten gebraucht, die aber beim Lernen nicht aus falschen Gründen (z.B. Helligkeit) zugeordnet oder auswendig gelernt werden dürfen (Overfitting)

Anwendung

- Bildverarbeitung zur Gesichts-, Bild- oder Texterkennung
- In der Robotik und Spracherkennung
- Forschung

PyTorch

- Vom Facebook AI Research lab entwickelt
- Basiert auf der Library Torch
- Libraries wie beispielsweise NumPy sind kompatibel

Tensorflow

- Vom Google-Brain-Team entwickelt
- Nachfolger für DistBelief
- Wird in einer Vielzahl von Google-Diensten verwendet

Zusammenfassung

- Ein Neuronales Netz besteht aus meistens in Schichten angeordneten Neuronen und Verbindungen zwischen den Neuronen
- Neuronen besitzen Propagierungs-, Aktivierungs- und Ausgabefunktion
- Lernen findet entweder überwacht oder unüberwacht statt
- Bei überwachtem Lernen werden die Kantengewichte mittels Backpropagation neu berechnet
- Anwendung und Entwicklung finden bei großen IT-Konzernen statt

Quellen

- <https://www.spektrum.de/lexikon/neurowissenschaft/neuronale-netze/8653>
- <http://www.dkriesel.com/media/science/neuronale-netze-de-zeta2-2col-dkrieselcom.pdf>
- <https://sci-hub.st/10.1007/s00101-003-0576-x>
- <https://www.fraunhofer.de/de/presse/presseinformationen/2019/juli/der-blick-in-neuronale-netze.html>
- https://de.wikipedia.org/wiki/K%C3%BCnstliches_neuronales_Netz
- <https://pytorch.org/>
- <https://www.tensorflow.org/>