Woche 13 Theorie: Bildverarbeitung

Skript

Erarbeitet von
Joana Grah

Lernziele 1
Inhalt 2
Einstieg 2
Das digitale Bild 2
Faltungen 3
Convolutional Neural Networks 5
Take-Home Message 6
Quellen 7
Weiterführendes Material 8
Disclaimer 8

Lernziele

- Erklären können, wie ein Bild digital im Computer gespeichert ist
- Bildkanten als wichtige Charakterisierung von Bildern nennen können
- Bildableitung als Indikator für Bildkanten nennen und erläutern können
- Das Prinzip der (diskreten) Faltung eines Bilds mit einem Filter anhand eines Beispiels erklären können
- Eigenschaften eines Convolutional Neural Networks, insbesondere die Rolle der Filter, erläutern können

Projekt „KI für alle"
HeiCAD – Heine Center for Artificial Intelligence and Data Science

CC BY

Inhalt

Einstieg

Einblendung Macaca-nigra-„Selfie“

Ein Bild sagt mehr als 1000 Worte. Stimmt schon manchmal, oder? Doch wie extrahieren wir maschinell die vielfältigen Informationen aus einem Bild?

Heute beschäftigen wir uns mit der Bildverarbeitung im Kontext von Deep Learning.

Das digitale Bild

Bevor wir richtig loslegen: eine kurze Wiederholung. Wie sieht ein Computer ein digitales Bild?

Nehmen wir mal dieses hier.

Einblendung Bild Mathematical Bridge, Heranzoomen bis Pixelstruktur sichtbar wird

Zoomen wir immer weiter in das Bild rein, bemerken wir irgendwann die Pixel-Struktur. Ein digitales Bild besteht also aus quadratischen Pixeln, die in Zeilen und Spalten angeordnet sind. Jedes Pixel beinhaltet eine Zahl oder einen Vektor aus Zahlen – entweder einen Graustufenwert oder im Fall eines RGB-Farbbildes eine Kombination aus drei Intensitätswerten für Rot, Grün und Blau. Ein digitales Bild ist also mathematisch gesehen eine Matrix.

Mit dieser Matrix können wir natürlich auch rechnen, z. B. Bilder addieren – was erstmal nicht besonders viel Sinn ergibt – oder sie mit einem Faktor multiplizieren. Was sind sinnvolle Rechenoperationen auf Bildern, die uns relevante Informationen – oder vielmehr interessante Features – liefern, die dann auch möglicherweise von neuronalen Netzwerken aufgegriffen werden?

Einblendung Bild Hund Lucky

Dafür sollten wir uns vielleicht zunächst überlegen, was überhaupt interessante und aussagekräftige Komponenten eines Bildes sind. Schaut euch dieses Bild mal an... wenn ich euch jetzt bitte das aus dem Gedächtnis aufzumalen, was würdet ihr zeichnen? Wahrscheinlich würdet ihr damit beginnen, die Bildkanten aufzumalen, oder? Kanten sind ein wichtiger Bestandteil von Bildern.

Quellen [1, 2, 3]

Doch wie lässt sich das mathematisch beschreiben?

Einblendung Bild Hund Lucky mit Intensitätsfunktion

Schauen wir uns noch einmal dieses Bild an. Wir betrachten jetzt mal eine bestimmte Zeile der Bildmatrix, diese hier. Wir können jetzt die Intensitätswerte des Bildes in dieser Zeile als Funktion betrachten, also auf der x-Achse die Position im Bild und auf der y-Achse die Graustufe oder den Intensitätswert aufführen.
Zum Beispiel erkennen wir in der Intensitätsfunktion helle und dunkle Stellen im Fell wieder. An dieser Stelle im Bild ist das Fell schwarz, also der Wert nahe bei 0, hier haben wir helleres Fell und der Wert nähert sich dem Maximum 255 an.
Insbesondere beim Übergang von Schwarz zu Weiß haben wir eine abrupte Veränderung, eine Bildkante. Wie sehen wir das in der Funktion? An dieser Stelle ändert sich die Intensitätsfunktion sehr schnell, sie hat dort eine hohe Steigung. Und wie erkennen wir so eine hohe Steigung? Klar, wir müssen die Ableitungen checken! An dieser Stelle ist die Ableitung des Bildes besonders hoch.

Doch wie berechnen wir Ableitungen eines digitalen Bildes? Also einer Matrix aus Zahlen? Da die Ableitung ja eine Veränderung angibt, könnten wir uns jeweils zwei benachbarte Pixel anschauen – horizontal oder vertikal – und davon die Differenzen berechnen.

Einblendung Bild Hund Lucky, Ableitungen horizontal und vertikal

So sieht das dann aus und wir sehen deutlich, dass die resultierenden Bilder tatsächlich Kanten hervorheben.

Faltungen

Jetzt zum eigentlichen Thema: Bildverarbeitung im Deep Learning. Es gibt eine bestimmte Art von neuronalen Netzwerken, die besonders geeignet für die Verarbeitung von Bilddaten ist – und vielfach sehr erfolgreich eingesetzt wird: sogenannte „Convolutional Neural Networks“ – oder kurz CNNs. Eine Convolution oder auf Deutsch Faltung ist eine mathematische Operation, die auf zwei Funktionen angewandt wird und eine neue Funktion als Ergebnis liefert. Das klingt ziemlich abstrakt, aber schauen wir uns vielleicht am besten direkt ein Beispiel aus der Bildverarbeitung an, um uns dem Konzept zu nähern.

Einblendungen Bild Hund Lucky (links) und Filter (rechts)

Hier (linke Seite) seht ihr wieder unser Bild und hier (rechte Seite) einen sogenannten Filter, das ist, wenn man ihn genau betrachtet, auch ein Bild, bestehend aus 9 Pixeln. Weil er 3 Zeilen und 3 Spalten hat, nennt man ihn auch 3x3-Filter. Wir sehen, dass er genau wie das Bild hier (linke Seite) numerische Einträge in den Pixeln hat. Diese nennt man auch Gewichte oder Weights (Hint Hint!).

Tatsächlich wird uns die Anwendung des Filters auf das Bild bzw. die Faltung des Bilds mit dem Filter mathematisch an die Berechnung eines Neurons in einem neuronalen Netzwerk erinnern. Auch hier benötigen wir Addition und Multiplikation.
Wir arbeiten uns jetzt von oben links nach unten rechts Pixel für Pixel durch das Bild durch und stellen uns vor, dass wir den Filter jeweils so über das Bild legen, dass das mittlere Pixel des Filters über das aktuelle Pixel im Bild gelegt wird.

Das bedeutet, um ein Pixel im Bild, das wir nach der Faltung erhalten, zu berechnen, benötigen wir Informationen der benachbarten Pixel, in unserem Fall sind das 8 Pixel. Jetzt wird jeder Wert im originalen Bild mit dem entsprechenden Wert im Filter multipliziert. Anschließend werden diese 9 Werte addiert. Das Ergebnis wird nun im entsprechenden Pixel des Ergebnisbilds gespeichert.

Aber was soll die ganze Theorie, ein Bild sagt ja mehr als 1000 Worte? Ein Video bestehend aus ganz vielen Bildern ist manchmal noch aussagekräftiger. Oder wie mein Mathelehrer aus der Oberstufe sagte: „Wir sind visuelle Typen!“

Einblendung Video 1 Filterung

Quelle [4]

Gucken wir uns ein Video von Grant Sanderson – oder besser bekannt als „3Blue1Brown“ – an – an dieser Stelle vielen Dank, dass wir das Material benutzen dürfen. Hier sehen wir den gleichen Filter, den wir auch benutzt haben: einen 3x3-Filter mit jeweils $\frac{1}{9}$ als Eintrag. Er gewichtet also jedes Pixel in der 3x3-Nachbarschaft gleich und berechnet somit einen Durchschnittswert. Welchen Effekt hat das wohl für das ganze Bild?

Abspielen Video 1 Filterung

Wer hat’s gewusst? Das resultierende Bild ist eine verschwommene oder geblurrte Version des Ausgangsbilds. Es ist unschärfer und Kanten werden verwischt.

Einblendung Video 2 Filterung

Quelle [5]

Hier sehen wir ein zweites Beispiel mit einem etwas größeren Filter, nämlich 5x5, d. h. er besteht aus 25 Pixeln und anderen Gewichten.

Abspielen Video 2 Filterung

CC BY

Das Gewicht in der Mitte ist am höchsten und nimmt zum Rand des Filters hin ab. Die Gewichte hier folgen sogar einer mathematischen Verteilung, der Gauß- oder Normalverteilung, daher nennt man die Anwendung des Filters auch „Gaussian Blur“. Die Anwendung so eines Filters auf ein verrauschtes Bild ist übrigens eine sehr simple Methode zur Bildentrauschung.

Als letztes Beispiel möchte ich noch einmal zurück zu den Bildkanten als charakteristische Bestandteile von Bildern kommen. Die Ableitungen in horizontaler und vertikaler Richtung können auch mit Hilfe von Filtern berechnet werden. Habt ihr eine Idee, wie die aussehen würden?

Einblendung Bild Ableitungsfilter

Tatsächlich reichen da im simpelsten Fall 2x1- und 1x2-Filter aus. An dieser Stelle noch einmal der Hinweis, dass wir hier natürlich nur sehr stark vereinfachte und exemplarische Fälle zeigen. In Wirklichkeit gibt es noch ganz viele andere Möglichkeiten, so etwas zu berechnen, und ein ganzes Forschungsfeld der mathematischen Bildverarbeitung.

Convolutional Neural Networks

Aber warum zeige ich euch das alles? Was hat das mit Deep Learning zu tun?

Tatsächlich werden bei Convolutional Neural Networks in jedem Hidden Layer ganz viele Faltungen angewandt. Die Filter oder genauer gesagt deren Gewichte bzw. Weights sind genau wieder die Parameter, die während des Trainings gelernt werden. Im Prinzip werden also pro Layer mehrere Filter gelernt.

Einblendung Bild Filter im ersten Layer eines CNNs

Quelle [6]

Diese kann man gut visualisieren. Hier seht ihr 96 verschiedene 11x11-Filter, diesmal für Farbbilder, die im ersten Hidden Layer eines CNNs gelernt wurden. Kommt euch das bekannt vor? Da sind sehr viele Kantenfilter dabei, die horizontale, vertikale und auch diagonale Kanten hervorheben können.

Typischerweise werden in den ersten Layers eher gröbere und globale Eigenschaften von Bildern wie Kanten hervorgehoben. Je „tiefer“ unsere Daten durch das CNN wandern, desto detaillierter werden die Charakteristiken oder Features, die in den Layers aufgegriffen werden.

Einblendung Bild Features eines CNNs in den einzelnen Layers

Quelle [7]

In diesem Bild sieht man das ganz gut. Wir stellen uns vor, dass unsere typische Darstellung eines neuronalen Netzwerkes, hier eines CNNs, um 90 Grad gedreht ist. Hier geht es um die Klassifikation von Bildern in „Car“, „Person“ und „Animal“. Ganz unten ist der Input-Layer, mit den Farbpixeln als Daten-Input. In den darauffolgenden drei Hidden Layers sieht man sehr gut, dass nach und nach immer detailliertere Features der Daten aufgegriffen werden; anfangs Kanten, dann Ecken und Konturen und im letzten Hidden Layer sogar Teile von Objekten, wie z. B. ein Autoreifen. Basierend darauf kann im Output-Layer die Klassifikation vorgenommen werden.

Einblendung Bild Feature Visualisation GoogLeNet

Quelle [8]

Und zum Schluss noch mehr faszinierende Visualisierungen von Features eines CNNs zur Klassifikation von Bildern. Auch hier wird deutlich, dass in den ersten Layers des CNNs links zunächst Kanten erfasst werden und je weiter rechts, d. h. je tiefer im CNN, desto komplexer werden die erkannten Features.

Was ihr vor allem für die Programmierung noch wissen solltet, wir aber leider in diesem Video nicht weiter ausführen können:

Auch in CNNs werden Aktivierungsfunktionen angewandt, z. B. ReLUs, bei Bildern also in jedem Pixel im Anschluss an die Faltung.

Einblendung Grafik MaxPooling

Es gibt eine ganze Reihe von Funktionen, die die Größe der Bilder in verschiedenen Layers reduzieren, Pooling genannt, da gibt es unterschiedliche Herangehensweisen. Beim sogenannten MaxPooling wird beispielsweise die Größe geviertelt, indem aus jedem 2x2-Pixel-Quadrat ein Pixel gemacht wird, welches den maximalen Wert der 4 Pixel im Quadrat enthält.

In der Praxis müssen wir uns Gedanken um die Pixel am Rand eines Bildes machen. Was passiert eigentlich z. B. mit dem Pixel ganz oben links, wenn ein Filter angewandt wird, aber oben und links keine benachbarten Pixel vorhanden sind? Auch da gibt es mehrere Optionen, z. B. kann man sie gleich null setzen. Überlegt euch mal, welche Auswirkungen das haben könnte.

Take-Home Message

Fassen wir noch einmal zusammen: Im Gegensatz zu den neuronalen Netzwerken, die wir bisher kennengelernt haben, werden bei CNNs in jedem Layer mehrere Filter auf die Daten

CC BY

angewandt. Die Gewichte der Filter werden im Training gelernt. Filter helfen dem Modell dabei, ganz verschiedene Bildeigenschaften zu erkennen, indem sie diese hervorheben.

In der Bildverarbeitung wird also ein Input-Bild im ersten Layer mit den entsprechenden Filtern gefaltet. Dieser Vorgang wird im nächsten Layer wiederholt, d. h. neue Filter werden auf das Ergebnisbild aus dem ersten Layer angewandt usw.

Wir können hier in diesem Kurs nur einen ganz kleinen Einblick in die weite Welt des Deep Learning und der Bildverarbeitung gewähren. Hoffentlich habt ihr nun aber eine bessere Intuition, wie man Bildmerkmale mathematisch quantifizieren kann, wie die Faltung eines Bildes funktioniert und was CNNs eigentlich machen.

Quellen

Quelle [1] Marr, D. (1976). Early processing of visual information. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, *275*(942), 483-519.

Quelle [2] Marr, D., & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, *207*(1167), 187-217.

Quelle [3] Elsayed, G., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., & Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. *Advances in neural information processing systems*, *31*.

Quelle [4] Grant Sanderson [@3blue1brown]. (2020, 9. September). *What is a convolution? What is a "Gaussian Kernel"? (This was the main animation used at the intro of last...* [Tweet]. Twitter.

Quelle [5] 3Blue1Brown. (2022, 18. November). *But what is a convolution?* [Video]. YouTube.

Quelle [6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

Quelle [7] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Quelle [8] Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*, *2*(11), e7.

CC BY

Weiterführendes Material

Bredies, K., & Lorenz, D. (2011). *Mathematische Bildverarbeitung* (Vol. 1). Vieweg+ Teubner.

3Blue1Brown. (2022, 18. November). *But what is a convolution?* [Video].

Many thanks to Grant Sanderson for agreeing for me to use snippets of his video material in this context!

Disclaimer

Transkript zu dem Video „Woche 13 Theorie: Bildverarbeitung“, Joana Grah.
Dieses Transkript wurde im Rahmen des Projekts ai4all des Heine Center for Artificial Intelligence and Data Science (HeiCAD) an der Heinrich-Heine-Universität Düsseldorf unter der Creative Commons Lizenz CC-BY 4.0 veröffentlicht. Ausgenommen von der Lizenz sind die verwendeten Logos, alle in den Quellen ausgewiesenen Fremdmaterialien sowie alle als Quellen gekennzeichneten Elemente.