KI für Alle 2: Verstehen, Bewerten, Reflektieren

Themenblock Bildklassifikation und Bildsegmentierung:
07_04Model_Segmentierung_Model-Unet

U-Net Architektur

Erarbeitet von
Dr. Ludmila Himmelspach

Lernziele 1
Inhalt 2
Einstieg 2
Die U-Net Architektur 2
Dropout-Regularisierung 3
Dekodierungspfad 4
Trainieren des U-Nets 4
Take-Home Message 5
Quellen 5
Disclaimer 5

Lernziele

- Du kannst erklären, wie das U-Net aufgebaut ist
- Du kannst die Funktionsweise der Dropout-Regularisierung erklären
- Du lernst, wie das U-Net trainiert wird

Projekt ai4all „KI für Alle"
HeiCAD –Heine Center for Artificial Intelligence and Data Science

Inhalt

Einstieg

Die Bildsegmentierungsaufgabe unterscheidet sich von der Klassifikationsaufgabe insofern, als wir ausgehend von einem Bild am Ende keine Klassenbezeichnung vorhersagen, sondern eine Maske, also ein Bild, generieren wollen, in dem nur Regionen markiert werden, die das interessierende Objekt im Ursprungsbild zeigen. Also brauchen wir eine Netzwerkarchitektur, die uns erlaubt, aus einem Bild ein weiteres Bild zu generieren. Diese Eigenschaft besitzt die *U-Net* Architektur, die 2015 im Rahmen eines anderen Wettbewerbs vorgestellt wurde.

Die U-Net Architektur

Bei U-Net handelt es sich um ein Convolutional Neural Network, das aus einem zusammenziehenden auf der linken Seite und einem expandierenden Pfad auf der rechten Seite besteht. Die beiden Pfade werden oft als *Kodierungs-* bzw. *Dekodierungspfade* bezeichnet. Dabei wird im Kodierungspfad der visuelle Inhalt des Bildes erfasst. Im Dekodierungspfad wird die räumliche Information des visuellen Inhalts des Bildes gelernt.

Quelle [1]

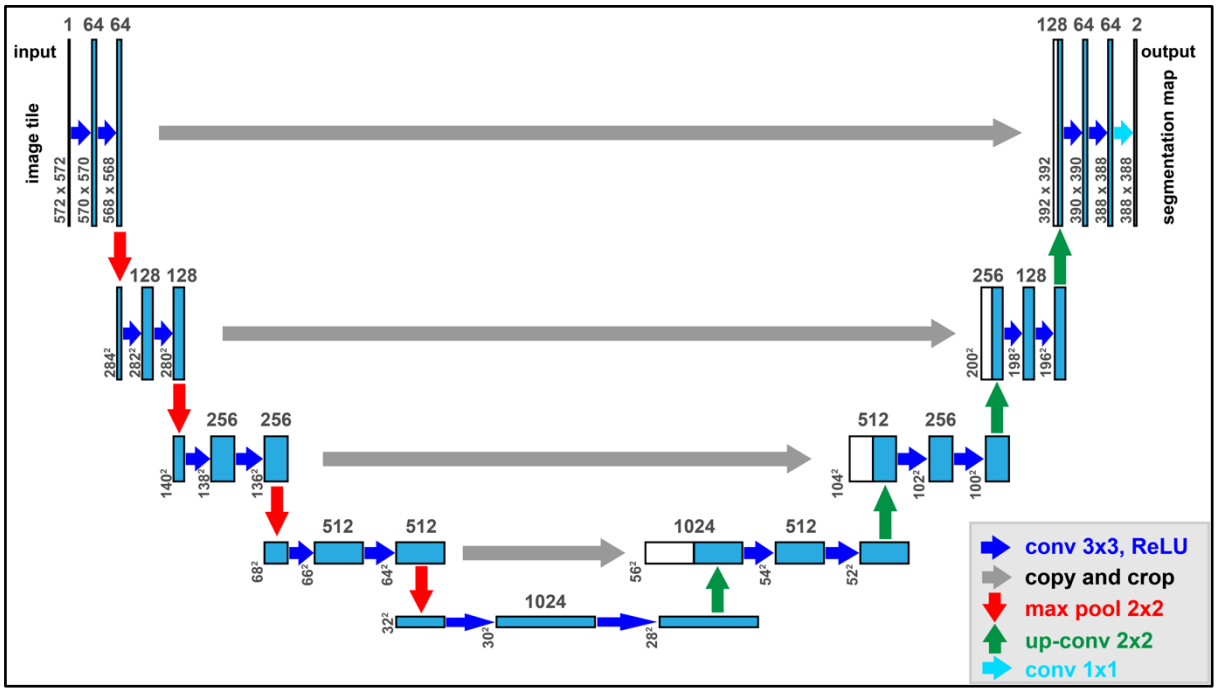

Abbildung 1: Die Architektur des U-Nets.

Der Kodierungspfad stellt dabei ein typisches tiefes Convolutional Neural Network dar, in dem nach jeweils zwei Convolutional Layern mit 3 x 3-großen Filtern ein 2 x 2 Max-Pooling Layer folgt. Dabei wird die Anzahl der Feature Maps nach jedem Max-Pooling-Layer

verdoppelt. Der einzige Unterschied zu uns bekannten CNN-Architekturen stellt die *Dropout-Regularisierung* dar, die jeweils auf dem ersten Convolutional Layer ihre Anwendung in der U-Net-Architektur findet.

Dropout-Regularisierung

Das Problem der tiefen neuronalen Netze besteht darin, dass sie wegen einer Unmenge an Optimierungsparametern zu einer Überanpassung an die Trainingsdaten neigen. Diese Netze lernen Muster, die den Trainingsdaten zu eigen sind und für neue Daten irreführend sein können. Deswegen liefern sie auf Testdaten schlechte Ergebnisse. Um diesem Problem entgegenzuwirken, werden in künstlichen neuronalen Netzen verschiedene *Regularisierungsverfahren* eingesetzt. In Convolutional Neural Networks hat sich die so genannte *Dropout-Regularisierung* als effektivste Methode erwiesen. Bei der Anwendung der Dropout-Regularisierung auf einen Layer werden während des Trainings einige Ausgabewerte dieses Layers zufällig gelöscht bzw. auf null gesetzt. Die *Dropout-Quote*, die den Anteil der durch null zu ersetzende Werte angibt, liegt normalerweise zwischen 0,2 und 0,8.

Quelle [2]

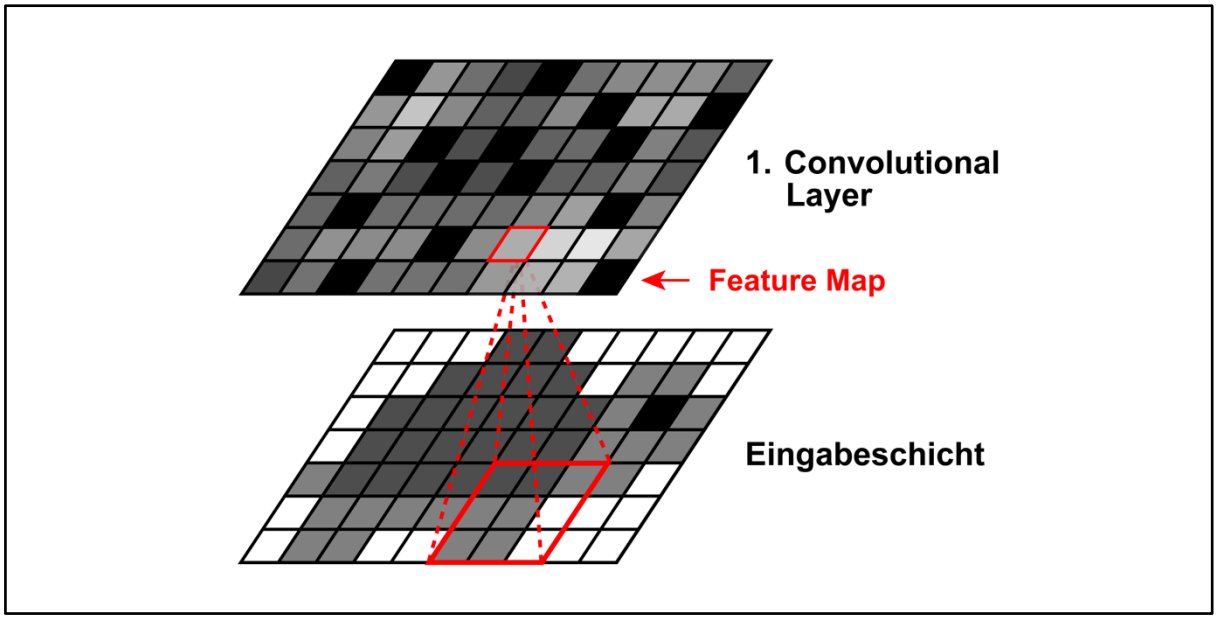

Abbildung 2: Anwendung der Dropout-Regularisierung auf eine Feature-Map mit der Dropout-Quote von 0,2.

Die Feature Maps enthalten die aus den Bildern gelernten Muster. Durch das zufällige Entfernen einiger Werte aus den Feature Maps wird dem Erlernen zu spezieller Muster entgegengewirkt. Bei der Dropout-Regularisierung muss man aber aufpassen, dass nicht zu viele Werte entfernt werden, sonst werden fast gar keine Muster gelernt, was sich auf die Qualität der Ergebnisse negativ auswirken kann.

Dekodierungspfad

Im Dekodierungspfad der U-Net Architektur wird das Ausgabebild, in unserem Fall die Segmentierungsmaske sukzessive aufgebaut. Dafür wird mit Hilfe der sogenannten *transponierten Faltungsoperation* die Dimension der eingegebenen Feature Maps vergrößert.

Die *transponierte Faltungsoperation* wird im Dekodierungspfad als Kontrahent zum Max-Pooling im Kodierungspfad angewendet. Allerdings können die räumlichen Informationen, die beim Max-Pooling verloren gehen, in resultierenden Feature Maps im Dekodierungspfad nicht im vollen Umfang wiederaufgebaut werden. Um diesen Verlust auszugleichen, werden die Feature Maps im Dekodierungspfad mit den entsprechenden Feature Maps aus dem Kodierungspfad verknüpft. Durch die Anwendung der zwei aufeinanderfolgenden Convolutional Layer auf die verketteten Feature Maps soll das Netzwerk lernen, eine präzisere Ausgabe zusammenzustellen.

Du fragst Dich wahrscheinlich, warum die Feature Maps aus dem Kodierungspfad nicht einfach in den Dekodierungspfad kopiert werden. Das wird nicht gemacht, weil wir in der Ausgabe des U-Nets eine Maske und nicht eine Kopie des ursprünglichen Bildes erhalten wollen. Die Feature Maps aus dem Kodierungspfad werden im Dekodierungspfad tatsächlich nur dafür genutzt, um die räumliche Information wiederherzustellen. Und die im Dekodierungspfad erstellten Feature Maps werden dafür genutzt, um Segmentierungsmasken zu erstellen. Auf diese Weise lernt das neuronale Netz einerseits im Kodierungspfad die für die Aufgabenstellung sinnvolle Merkmale aus den Originalbildern zu extrahieren und andererseits diese Merkmale für den Aufbau der Segmentierungsmasken im Dekodierungspfad zu nutzen.

In der Ausgabe jeder *transponierten Faltungsschicht* wird die Anzahl der Feature Maps halbiert, sodass zum Schluss eine einzige Feature Map in der Ausgabe des U-Nets bleibt. Diese entspricht der Segmentierungsmaske.

Trainieren des U-Nets

Das U-Net wird anhand der Trainingsbilder und den dazugehörigen Segmentierungsmasken trainiert. Während des Trainings werden die Optimierungsparameter und das sind vor allem die Gewichte in Filtermasken und Bias-Terme in Convolutional Schichten so angepasst, dass das U-Net für jedes Originalbild eine optimale, für die Aufgabenstellung sinnvolle Segmentierungsmaske erstellt. Die Leistungsfähigkeit des Netzes wird oft dadurch erhöht, dass beim Trainieren des U-Nets eine online oder eine offline Data Augmentation angewendet wird. Dabei werden dem U-Net zusätzliche Bilder zur Verfügung gestellt, die durch die Anwendung verschiedener Transformationen der Trainingsbilder erzeugt werden. Hierbei darf man nicht vergessen, auch die dazugehörigen Masken mit den gleichen Transformationsoperationen zu verändern.

Take-Home Message

In diesem Video hast Du den Aufbau des U-Nets, eines Convolutional Neural Networks für die Bildsegmentierung, kennengelernt. Außerdem haben wir kurz darüber gesprochen, wie U-Nets trainiert werden.

Quellen

Quelle [1] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer International Publishing.

Quelle [2] Chollet, F. (2018). *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG.

Disclaimer