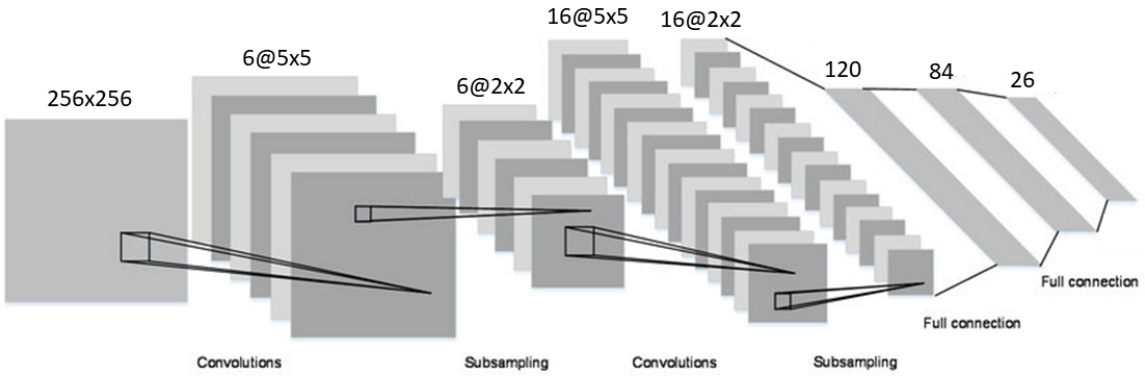
Übungsblatt: 10

Bearbeitung am 16. Juli

Aufgabe 1: Klassifikation

Gegeben sei das folgende Netz zur Klassifizierung von Großbuchstaben in Bildern der Größe 256x256 (Graustufen), wobei die Conv-Layer Valid-Padding haben und die Aktivierungsfunktion ReLU sei. Dabei können mehrere verschiedene Buchstaben auf einem Bild vorkommen. Jede Art von Buchstabe kann aber nur höchstens einmal auf einem Bild vorkommen.

Die Zahlen über den Schichten geben jeweils die Anzahl der Feature Maps und die Filtergröße bzw. die Anzahl an Knoten an.



(a) Geben Sie den Pseudo-Code und die Ausgabedimension aller Layer für die Implementierung des Netzwerks an.

(b) Geben Sie eine passende Lossfunktion (mit zugehöriger Formel für ein Trainingsbeispiel) für das Training an.

(c) Welche Formen der Regularisierung könnte man einbauen und welche Operationen der Datenaugmentierung sind hier sinnvoll? Warum/Warum nicht?

(d) Wie müssen das Netz und die Lossfunktion angepasst werden, damit man nur genau einen Buchstaben pro Bild klassifiziert?

Lösung:

(a)
```
(Input              256x256x1)
Conv2D 6, 5x5       252x252x6
ReLU                252x252x6
MaxPool2D 2x2       126x126x6
Conv2D 16, 5x5      122x122x16
ReLU                61x61x16
MaxPool2D 2x2       61x61x16
Flatten             59.536
```

FC 120	120
ReLU	120
FC 84	84
ReLU	84
FC 26	26
Sigmoid	26

(b) Binäre Cross Entropy (negativer Log Likelihood). Formel (für ein Trainingsbeispiel):

$$\sum_{c=1}^{26} -y_c \log P(Y_c = 1|X) - (1 - y_c) \log P(Y_c = 0|X)$$

(c)
- Dropout nach den FC Layern (z.B. Rate = 0.5)
- Dropout nach Conv-Layern mit kleiner Rate (z.B. 0.1)
- Regularisierungsterme für Weigth Decay
- Rauschen auf Input/Gewichte/Ausgabe
- Early Stopping
- (Batch Normalization)
- Datenaugmentierung

Für Datenaugmentierung sinnvoll wäre:

- (Kleine) Rotationen
- Brightness Shift
- Zoom/Crop (klein)
- (Kleine) Translationen (in beide Richtungen)
- Bedingte Spiegelungen bei symmetrischen Buchstaben (z.B. A, I, H etc.)

NICHT geeignet:

- Große Rotationen und Spiegelungen, bei denen das Label verändert wird
- Skalierungen, da wir eine feste Auflösung brauchen
- großes Cropping/Translationen, bei denen Teile der Buchstaben entfernt werden

(d) Für nur einen Buchstaben pro Bild: Sigmoid in der letzten Schicht zu Softmax und Categorical statt Binärer Cross Entropy verwenden.

Aufgabe 2: Sequenzklassifizierung

Im Folgenden soll das Netz aus Aufgabe 1 zur Sequenzerkennung von Buchstaben

erweitert werden. Dafür nehmen wir vereinfacht an, dass nur die Buchstaben von A bis J vorkommen können. Es sollen Sequenzen von 0 bis 5 Buchstaben erkannt werden können.

(a) Gegeben sei die folgende Ausgabe des Netzes y_k^t. Dekodieren Sie die Sequenz mittels des CTC-Greedy-Decoders.

y_k^t	0	1	2	3	4
-	0.9	0.1	0.2	0.3	0.15
A	0	0.8	0.1	0	0
B	0	0	0.7	0.4	0
C	0	0	0	0	0
D	0	0	0	0	0.1
E	0	0	0	0.3	0.15
F	0	0	0	0	0.1
G	0.1	0.1	0	0	0.15
H	0	0	0	0	0.15
I	0	0	0	0	0.2
J	0	0	0	0	0

Lösung: Argmax an jeder Stelle ergibt **-ABBI** und wird damit zu **ABI**

(b) Die Ground-Truth für die Ausgabe ist eigentlich AI. Berechnen Sie die Forward- und Backward-Variablen α und β. Berechnen Sie damit die Gesamtwahrscheinlichkeit von AI.

Lösung:

α	0	1	2	3	4
-	0.9	0.09	0.018	0.0054	0.00081
A	0	0.72	0.081	0	0
-	0	0	0.144	0.0675	0.010125
I	0	0	0	0	0.0135
-	0	0	0	0	0

β	0	1	2	3	4
-	0.0135	0.0006	0	0	0
A	0	0.0144	0.006	0	0
-	0.00108	0.0012	0.012	0.06	0
I	0	0	0	0	0.2
-	0.00081	0.0009	0.009	0.045	0.15

P	0.0135	0.0135	0.0135	0.0135	0.0135

Aufgabe 3: Bildsegmentierung und Objekterkennung

(a) Wie unterscheiden sich Bildsegmentierung und Objekterkennung, d.h. wie wird klassifiziert, was sind die Ausgaben etc.?

Lösung: Bei der Bildsegmentierung wird i.d.R. jeder Pixel des Bildes einzeln klassifiziert, d.h. die Ausgabe ist eine Maske der gleichen Dimension wie das Bild. Bei manchen Architekturen hat die Maske auch eine etwas kleinere Auflösung und muss hochskaliert werden. Die Klassen sind meistens exklusiv, d.h. die Regionen überschneiden sich nicht. Jedem Pixel wird dabei ein Klassenvektor zugeordnet, dessen Länge der Anzahl der Klassen entspricht.

Bei der Objekterkennung werden Bounding Boxen berechnet, die Regionen des Bildes rechteckig eingrenzen und diese Region klassifizieren. Die Klassen sind nie exklusiv, d.h. die Boxen können sich überschneiden. Es können auch mehrere Instanzen der gleichen Klasse auftreten. Die genaue Anzahl der Boxen ist abhängig von der Architektur bzw. von den Convolutions, aber jeder Box werden 4 Koordinaten und n Klassen zugeordnet.

(b) Gegeben sei das bekannte FCN U-Net (Abbildung 1) mit der Variation, dass die Conv-Layer Padding = Same haben. Die Up-Convolutions sollen dabei die gleiche Filtergröße wie die normalen Convolutions haben. Geben Sie den Pseudocode (ohne Dimensionen) zu dieser Architektur an. Die Aktivierungsfunktion kann hier als Attribut der Conv-Layer aufgefasst werden. Als Ausgabe sollen 10 verschiedene, exklusive Klassen möglich sein.

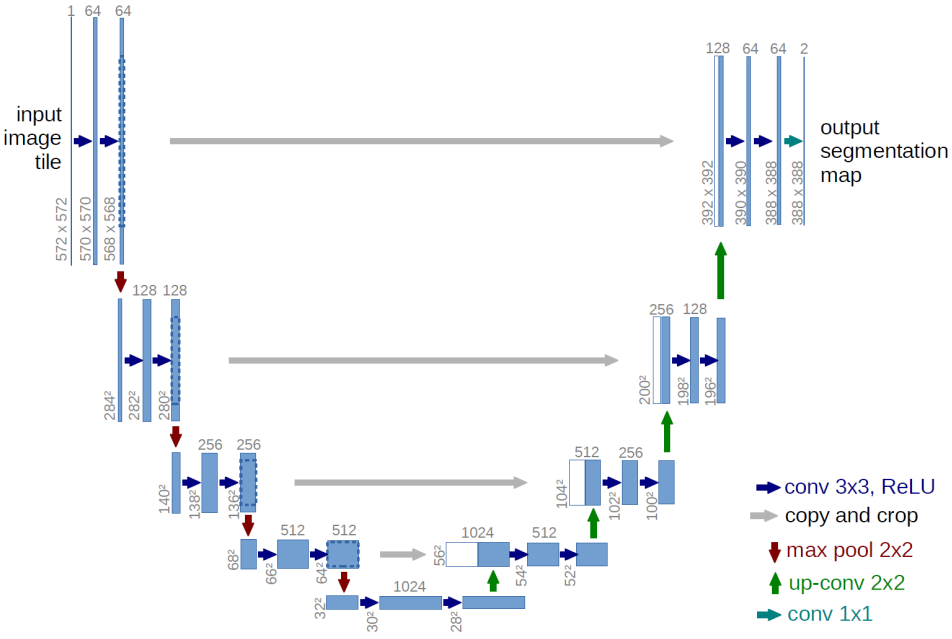


Abbildung 1: U-Net

Lösung:

Alle Convs/Deconvs mit padding = same

Parameter Deconv: Anzahl Filter, Filtergröße, Strides

L0 = Conv 64, 3x3, ReLU
L1 = Conv 64, 3x3, ReLU

```
L2 = Max Pool, 2x2
L3 = Conv 128, 3x3, ReLU
L4 = Conv 128, 3x3, ReLU
L5 = Max Pool, 2x2
L6 = Conv 256, 3x3, ReLU
L7 = Conv 256, 3x3, ReLU
L8 = Max Pool, 2x2
L9 = Conv 512, 3x3, ReLU
L10 = Conv 512, 3x3, ReLU
L11 = Max Pool, 2x2
L12 = Conv 1024, 3x3, ReLU
L13 = Conv 1024, 3x3, ReLU
L14 = Deconv 512, 3x3, 2x2
L15 = Concat L14, L10
L16 = Conv 512, 3x3, ReLU
L17 = Conv 512, 3x3, ReLU
L18 = Deconv 256, 3x3, 2x2
L19 = Concat L18, L7
L20 = Conv 256, 3x3, ReLU
L21 = Conv 256, 3x3, ReLU
L22 = Deconv 128, 3x3, 2x2
L23 = Concat L22, L4
L24 = Conv 128, 3x3, ReLU
L25 = Conv 128, 3x3, ReLU
L26 = Deconv 64, 3x3, 2x2
L27 = Concat L26, L1
L28 = Conv 64, 3x3, ReLU
L29 = Conv 64, 3x3, ReLU
L30 = Conv 10, 1x1
L31 = Softmax
```

(c) Listen Sie die (Haupt-)Ansätze der Klassifikation in der Objekterkennung auf und nennen Sie jeweils eine Beispielarchitektur.

Lösung: Hauptansätze:

- Two Stage Detektoren: Erst Bestimmung vieler Regions of Intereset ("Objectness Score"), danach Klassifikation dieser RoIs (und zusätzlich Anpassung der Koordinaten). Beispiel dafür ist Faster R-CNN.
- One/Single Stage/Shot Detektoren: Regression und Klassifikation in einem Schritt. Beispiele sind YOLO, Single Shot Detector und ankerfreie Methoden.
- Ankerfreie Methoden lassen sich als eigener Ansatz betrachten, auch wenn sie ebenfalls als Single Stage Methode definiert werden können.

Beispiele sind CornerNet und CenterNet.

(d) Welche Arten von Segmentierung gibt es? Wodurch zeichnen sie sich aus?

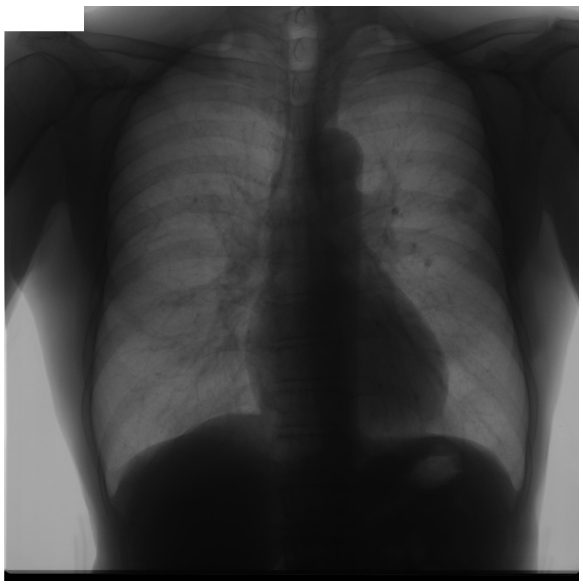
Lösung: Es gibt drei verschiedene Arten:

- Semantic Segmentation: Klassifiziert das gesamte Bild in Regionen, aber nur nach Label, also Person, Gebäude, Boden, Tier, Baum, etc. und unterscheidet dabei nicht zwischen verschiedenen Personen, Autos, Tieren usw.
- Instance Segmentation: Klassifiziert nur zählbare Objekte im Bild, wie Personen, Autos, Verkehrsschilder etc. und ignoriert dabei großfläche Bereiche wie Gebäude, Straßen usw. Dafür wird zwischen einzelnen Personen unterschieden, jede erhält ein eigenes Label.
- Panoptic Segmentation: Kombiniert Semantic und Instance Segmentation, d.h. sowohl unzählbare Regionen als auch zählbare Objekte werden klassifiziert und die zählbaren Objekte sind untereinander unterscheidbar.

Aufgabe 4: Medizinische Anwendung

Betrachten Sie die folgende Anwendung in der Medizin: Sie möchten die Diagnose von Ärzten untersützten, indem Sie aus Röntgen-Scans das (gesamte) Lungenvolumen bzw. als ersten Schritt die Lungenfläche des Patienten bestimmen.

Beschreiben Sie, welche Art von Neuronalem Netz Sie einsetzen könnten und welche Art von Daten die Mediziner zum Training zur Verfügung stellen müssten.



Lösung: Segmentierungsaufgabe, z. B. U-Net. Zum Training müssten die Lungenflügel markiert sein, z. B. als binäre Maske oder als Polygonzug (anschließend umwandeln). Eine Möglichkeit wäre auch, die Lungenflügel als getrennte Klassen zu markieren, um so unabhängige Berechnungen durchführen zu können.