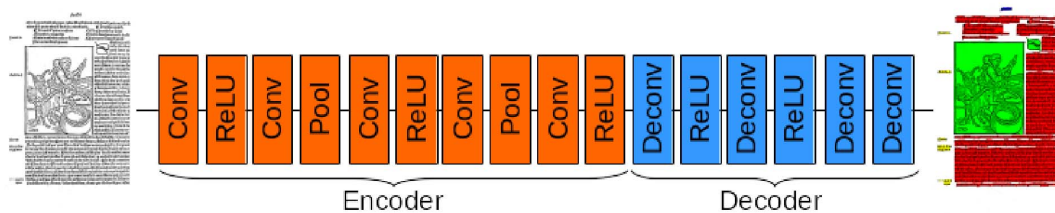


Übungsblatt: 7

Bearbeitung am 18. Juni

Aufgabe 1: FCN

In einem Paper ist folgende FCN-Struktur angegeben:



Alle CNN-Layer haben hierbei einen Stride von 1, eine Kernel-Size von 3x3 und nutzen *same* Padding, d. h. ein Padding von 1. Die Filteranzahl im Encoder ist 20, 40, 60, 80, 100, und im Decoder 80, 60, 40, 5. Alle Pooling-Layer sind normale 2x2-Max-Pooling-Layer. Das erste Deconv-Layer ist äquivalent zu einem normalen Conv-Layer, da es einen Stride 1x1 mit „Same“ Padding verwendet. Die nächsten zwei Deconv-Layer werden zum Upsampling verwendet (Stride 2x2), um das Bild wieder auf Originalgröße zu bringen. Die ersten drei Deconv-Layer haben eine Kernel-Size von 3, das letzte Layer nur von 1x1. Die finale Ausgabe wird in ein Softmax geschickt um die Wahrscheinlichkeitsverteilung zu bestimmen. Dieses FCN soll genutzt werden, um eine Seite großflächig in Hintergrund, Text, Bild, Überschriften und Marginalien zu segmentieren.

(a) Warum hat die letzte FCN-Schicht 5 Kanäle?

Lösung: 5 verschiedene Klassen

(b) Hat das Netzwerk-Skip-Connections? Ist das so sinnvoll?

Lösung: Dieses FCN-Netzwerk hat keine Skip-Connections. Ob das sinnvoll ist kann in beide Richtungen beantwortet werden:

- Ja: Beim Layout sind nur großflächige Regionen, aber keine einzelnen Buchstaben von Interesse.
- Nein: Zur Abgrenzung von Marginalien, Bildern und Text sind eventuell feine Auflösungen erforderlich.

(c) Geben Sie den Pseudo-Code für das Netzwerk an.

Lösung:

```
Conv 20, 3x3, padding=same
ReLU
```

```

Conv 40, 3x3, padding=same
Pool 2x2
Conv 60, 3x3, padding=same
ReLU
Conv 80, 3x3, padding=same
Pool
Conv 100, 3x3, padding=same
ReLU
Deconv 80, 3x3, padding=same (stride=1x1)
ReLU
Deconv 60, 3x3, 2x2, padding=same
ReLU
Deconv 40, 3x3, 2x2, padding=same
Deconv 5, 1x1
Softmax
    
```

- (d) Geben Sie für jede Schicht des Netzwerks die Eingabe- und Ausgabedimension sowohl die Anzahl der trainierbaren Gewichte an.

Lösung:

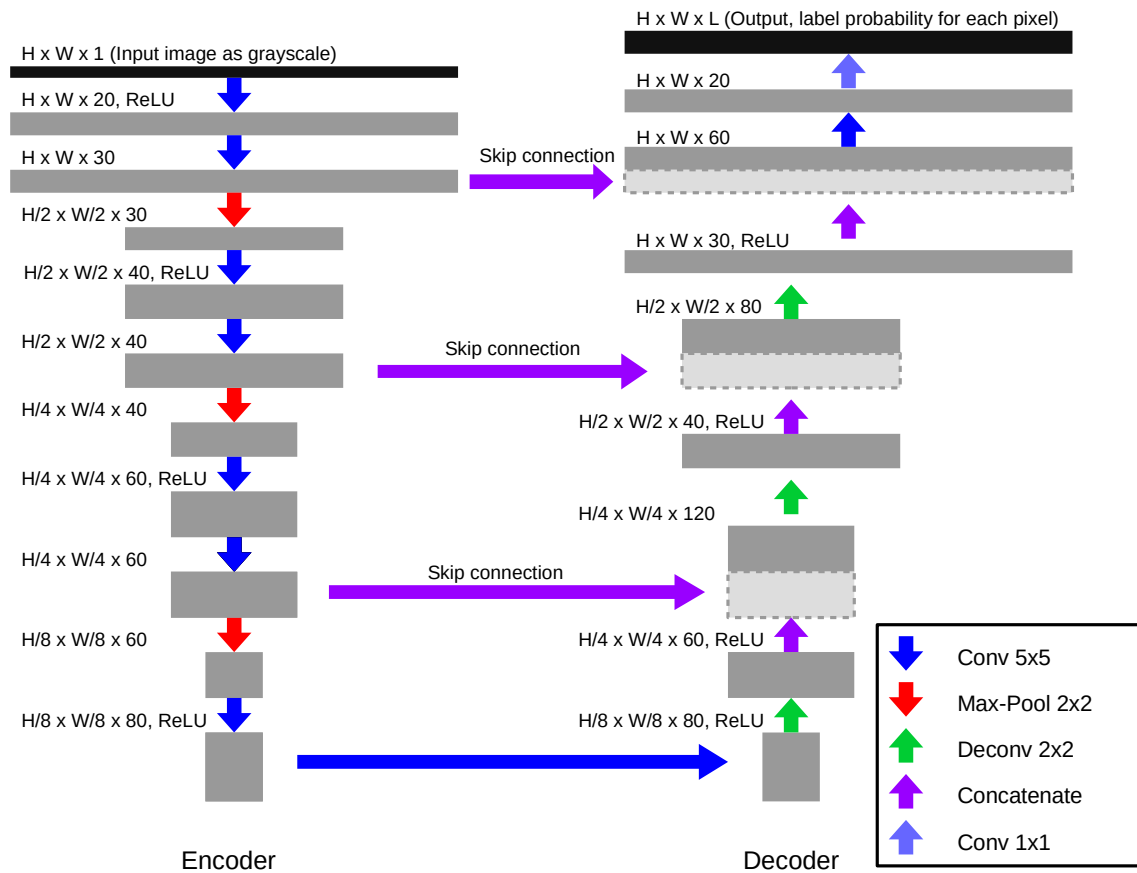
Bias nicht vergessen!

Layer	Input	Output	N_{train}
Conv	$h \times w \times 1$	$h \times w \times 20$	$1 \cdot 3 \cdot 3 \cdot 20 + 20$
ReLU	$h \times w \times 20$	$h \times w \times 20$	0
Conv	$h \times w \times 20$	$h \times w \times 40$	$20 \cdot 3 \cdot 3 \cdot 40 + 40$
Pool	$h \times w \times 40$	$h/2 \times w/2 \times 40$	0
Conv	$h/2 \times w/2 \times 40$	$h \times w \times 60$	$40 \cdot 3 \cdot 3 \cdot 60 + 60$
ReLU	$h/2 \times w/2 \times 60$	$h/2 \times w/2 \times 60$	0
Conv	$h/2 \times w/2 \times 60$	$h/2 \times w/2 \times 80$	$60 \cdot 3 \cdot 3 \cdot 80 + 80$
Pool	$h/2 \times w/2 \times 80$	$h/4 \times w/4 \times 80$	0
Conv	$h/4 \times w/4 \times 80$	$h/4 \times w/4 \times 100$	$80 \cdot 3 \cdot 3 \cdot 100 + 100$
ReLU	$h/4 \times w/4 \times 100$	$h/4 \times w/4 \times 100$	0
Deconv	$h/4 \times w/4 \times 100$	$h/4 \times w/4 \times 80$	$100 \cdot 3 \cdot 3 \cdot 80 + 80$
ReLU	$h/4 \times w/4 \times 80$	$h/4 \times w/4 \times 80$	0
Deconv	$h/4 \times w/4 \times 80$	$h/2 \times w/2 \times 60$	$80 \cdot 3 \cdot 3 \cdot 60 + 60$
ReLU	$h/2 \times w/2 \times 60$	$h/2 \times w/2 \times 60$	0
Deconv	$h/2 \times w/2 \times 60$	$h \times w \times 40$	$60 \cdot 3 \cdot 3 \cdot 40 + 40$
Deconv	$h \times w \times 40$	$h \times w \times 5$	$40 \cdot 1 \cdot 1 \cdot 5 + 5$
Softmax	$h \times w \times 5$	$h \times w \times 5$	0

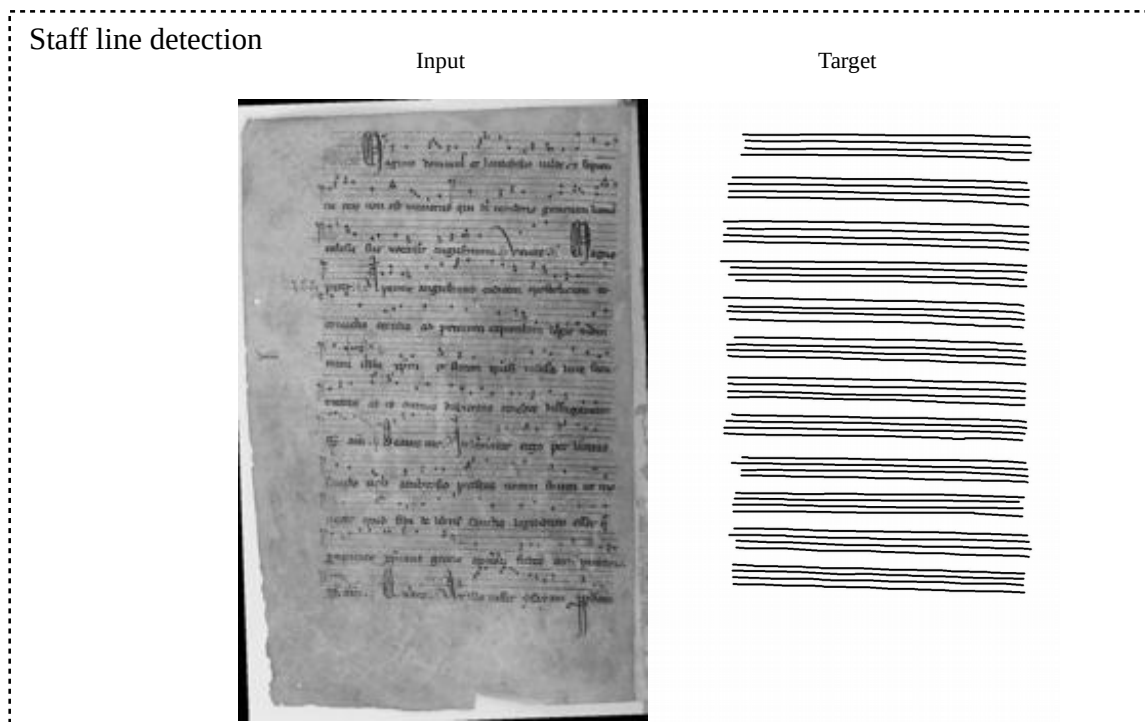
- (e) Implementieren Sie die Netzstruktur und trainieren Sie das Netzwerk auf den vorgegebenen Daten.

Aufgabe 2: FCN

In einem anderen Paper ist folgende FCN-Struktur angegeben:



Die finale Ausgabe (Logits) wird in ein Softmax geschickt um die Wahrscheinlichkeitsverteilung zu bestimmen. Dieses FCN soll genutzt werden, um auf einer Seite Notenlinien zu erkennen.



(a) Wie viele Kanäle hat die letzte Schicht?

Lösung: Entweder 2 Kanäle (Vordergrund/Hintergrund), dann mit Softmax-Aktivierungsfunktion und Cross-Entropy-Loss, oder 1 Kanal, dann mit Sigmoid und Logistic-Regression-Loss (Logistische Regression)

(b) Hat das Netzwerk-Skip-Connections? Ist das so sinnvoll?

Lösung: Ja! Ist sinnvoll, bzw erforderlich, da Notenlinien nur sehr schmale/dünne Objekte sind.

(c) Geben Sie den Pseudo-Code für das Netzwerk an.

Lösung:

Alle Conv/Deconv layer mit padding=same

L0 = Conv 20, 5x5

L1 = ReLU

L2 = Conv 30, 5x5

L3 = Pool

L4 = Conv 40, 5x5

L5 = ReLU

L6 = Conv 40, 5x5

L7 = Pool

L8 = Conv 60, 5x5

L9 = ReLU

```

L10 = Conv 60, 5x5
L11 = Pool
L12 = Conv 80, 5x5
L13 = ReLU
L14 = Deconv 60, 5x5, 2x2
L15 = ReLU
L16 = Concat L15, L10
L17 = Deconv 40, 5x5, 2x2
L18 = ReLU
L19 = Concat L18, L6
L20 = Deconv 30, 5x5, 2x2
L21 = ReLU
L22 = Concat L21, L2
L23 = Conv 20, 5x5
L24 = Conv 2, 1x1
L25 = Softmax
    
```

- (d) Geben Sie für jede Schicht des Netzwerks die Eingabe- und Ausgabedimension sowohl die Anzahl der trainierbaren Gewichte an.

Lösung: Analog zu Aufgabe 1, Concat hat natürlich auch keine Parameter, aber aufpassen, dass die Dimension vom Output der Concat-Layer die Summe der Kanäle ist (in der Abbildung ist das angegeben). Dies wiederum ist dann die Dimension für die Filter.

- (e) Implementieren Sie die Netzstruktur und trainieren Sie das Netzwerk auf den vorgegebenen Daten.