

EML Task 14

1. Intro

Meine komplette Abgabe befindet sich diese Woche in diesem Dokument. 14.1.3 und 14.2.3 wurden nicht bearbeitet.

2. Task 14.1.2

| File | nx | ny | nz | dtype | size (GiB) |
|------------------|-----|-----|------|-------|-------------|
| data_train.npz | 782 | 590 | 1006 | fp32 | 1.5986330 |
| labels_train.npz | 782 | 590 | 1006 | int8 | 0.006835938 |
| data_test_1.npz | 782 | 251 | 1006 | fp32 | 0.6816406 |
| data_test_2.npz | 334 | 841 | 1006 | fp32 | 0.9726563 |

Die meisten dieser Daten habe ich direkt von der Website abgelesen. Nur für die labels habe ich folgendes Skript benutzt:

```
from numpy import load
import torch

data = load('seismic/labels_train.npz')
lst = data.files
for item in lst:
    print(item)
    print(data[item].dtype)
    print(torch.as_tensor(data[item]).shape)
```

(Ich habe vorher einen soft link zu “seismic” mit “ln -s /mnt/hd1/data/seismic” erstellt)

3. Task 14.2.2

3.1. Max pooling

Max pooling ist eine Operation welche räumlichen Dimensionen der input feature verringert (im paper wird in jedem Schritt halbiert). Laut https://de.wikipedia.org/wiki/Convolutional_Neural_Network#Max_Pooling wird ein 2x2 Fenster über den 2D input (hier: Bilder) geschoben und in jedem Schritt wird der maximale Wert aus diesem Fenster übernommen, daher auch der Name der Operation.

3.2. Upsampling by bilinear interpolation

Mit dieser Operationen sollen die räumlichen Dimensionen der feature vergrößert werden, zBsp um die Auflösung zu erhöhen. Bilinear heißt, dass wir 2D Räume betrachten und zuerst in eine Dimension (zBsp X) und dann in die andere interpoliert wird (zBsp Y).

3.3. Skip connections

Mit “skip connections” oder “shortcut connections” wird der Ausgang einer Schicht mit einer nicht benachbarten Schicht im Netzwerk zu verbinden, wodurch effektiv eine oder mehrere Schichten übersprungen werden. Im U-Net werden so Schichten des Encoders mit Schichten des Decoders verbunden. Die Idee ist, dass Schichten des Encoders mehr räumliche Informationen enthalten, welche durch

das Downsampling verloren gehen. Durch skip connections sollen diese erhalten und später mit den Schichten des Decoders (welche mehr semantische Informationen enthalten) kombiniert werden.

Im paper standen dazu meiner Meinung nach nicht ausreichend Infos, deshalb habe ich folgende Quelle dazugezogen: <https://theaisummer.com/skip-connections/>

3.4. Batch normalization

Hiermit sollen die inputs jeder Schicht normalisiert werden. Das erreicht man durch Skalierung der Eingaben, sodass sie einen Mittelwert von 0 und eine Standardabweichung von 1 haben, und durch das Lernen von zwei zusätzlichen Parametern, um die normalisierten Werte zu skalieren und zu verschieben.