

# Orvosi képelemzés és szegmentálás neuronhálókkal

Csányi Dávid

Témavezető: Maga Balázs

2021.12.07.

## 1. Bevezetés

Az orvosi képelemző módszereknek(CT, MRI, ultrahang, röntgen, stb) köszönhetően világszerte napi szinten hatalmas mennyiségű adat jön létre, melyek diagnosztikai célú elemzése nagyszámú speciálisan képzett szakembert kíván. Ez a folyamat számos esetben meglehetősen időigényes, így igény mutatkozik arra, hogy automatizált rendszerek segítségével tudjunk hozzá támogatást nyújtani. A mély konvolúciós neuronháló 2010-es évek elején elért sikere óta a deep learning vált a legnépszerűbb módszerré a gépi látási feladatok megoldására. Az elmúlt néhány évben számos megoldási módot alkalmaztak sikeresen orvosi képszegmentálásra is. Ezen projekt célja az egyik ismert architektúra, az U-Net használata agydaganatok kijelölésére MRI felvételeken.

## 2. Az adathalmaz

A Brain Tumor Segmentation Challenge(BRATS) egy minden évben megrendezett verseny, mely multimoduláris MRI felvételekből álló adathalmazt biztosít a state-of-the-art módszerek összehasonlítására. A 2020-as tanító adathalmaz([4], [5], [2], [3]) 369 betegről tartalmaz felvételeket, minden pácienshez négyféle képelemző eljárással(t1, t1ce, t2, flair) készült kép és egy kézzel annotált maszk tartozik. A neuroradiológusok három különböző daganattípust jelöltek be. A képek nifti fájlformátumban adódtak és  $155 \times 240 \times 240$  voxelből(volume pixel) állnak.

### 2.1. Előfeldolgozás

A közzétett adathalmazon már bizonyos preprocesszási lépéseket elvégeztek: a felvételeket azonos felbontásra( $1mm^3$ ) interpolálták és levágták a koponyán kívüli részeket. További lépésként a különböző daganattípusokat azonos címkével láttam el, hogy egy egyszerűbb szegmentálási feladatot kapjak. Ezután a képeket minden dimenzió mentén 32-vel osztható voxelszámra egészítettem ki(padding), ezzel biztosítva, hogy a pooling operációk 2-es stride-al mindig páros számú voxelen dolgozzanak. A képeket a matplotlib és ipynbwidgets könyvtárak segítségével vizualizáltam. Az adathalmazt egy 269 elemű tanító és egy 100 elemű validációs részre osztottam.

### 2.2. Augmentációk

Az augmentációkat a PyTorch dataloader on-the-fly módon, minden egyes bemeneti képre egymástól függetlenül, véletlenszerűen hajtotta végre. Az  $x$  és  $y$  **tengely felcserélése** és az  $x$ ,  $y$ ,  $z$  tengelyekre való **tükrözés** 0.5 valószínűséggel történt meg. Ezen kívül minden képet megszo-roztam egy  $[0.8, 1.2]$  intervallumbeli véletlen számmal. Továbbá a koponyára minden dimenzió mentén egy  $[-8, 8]$  intervallumbeli egész értékkel való **eltolást** alkalmaztam.

## 3. Az architektúra

### 3.1. U-Net

Orvosi képszegmentálásra gyakran használt az **U-Net** architektúra ([6]). A szokásos konvolúciós és pooling rétegek sorozata adja az összehúzó ágat, amely célja a magasszintű információk kinyerése. Ezután a szétterjedő ágban konvolúciók és felskálázások sorozatos alkalmazásával növeljük a felbontását, míg visszakapjuk az eredeti képméretet. A pontos lokalizációt az oldalirányú kapcsolatok segítik, melyek az összehúzó ág nagyfelbontású képeit konkatenálják a felskálázott képpel a szétterjedő ágban. Az eredeti U-Net architektúra három dimenziós változatát használtam azazal a különbséggel, hogy a csatornák számát mindenhol a felére csökkentettem az elfogadható memóraigény érdekében. Továbbá a háló legelején egy average pooling, a végén egy kétszeres up-sampling réteget alkalmaztam. Így csak feleakkora felbontású képekkel dolgozott a háló, cserébe a tanítás gyorsabb lett és a batchek méretét egyről nyolcra növelhettem. (A 25GB-os videokártya memória limit alatt maradv.)

### 3.2. Költségfüggvények

A szegmentációk kiértékelésére gyakran használják a **Dice Score**-t és az abból kapott költséget.

$$DSC = \frac{2TP}{2TP + FP + FN}, DL = 1 - DSC$$

Ezen megközelítés hátránya, hogy egyformán súlyozza a true positive és false negative predikciókat. Ez általában magas PPV-t, de alacsony sensitivity-t eredményez. Ezt a problémát az általánosabb **Tversky Index** segítségével lehet kezelni.  $TI = \frac{2TP}{2TP + \alpha FP + \beta FN}$   $TL = 1 - TI$

Egy másik probléma, hogy a könnyű háttérvoxeleken és a nehezen megjósolható fontos részeken egyforma hangsúly van, ezt próbálja kezelni a **Focal Tversky Loss**.  $FTL = (1 - TI)^{\frac{1}{\gamma}}$  [1]

## 4. Eredmények

Az első tanítást augmentációk nélkül indítottam próbaként, az 500. epoch után a train dsc 0.66, a validációs dsc 0.42 volt. A második tanítást validációkkal 1257 epoch-ig futtattam, ami 42 óráig tartott. A train dsc 0.79 lett, a validációs dsc pedig 0.67-ig emelkedett. Mindkét esetben a Focal Tversky költségfüggvényt használtam.

## 5. Összefoglalás

Az U-Net modellt használva a kitűzött feladatot sikerült eredményesen megoldani. A tanítás kiemelkedően sok számítási kapacitást és időt igényelt. A feladatban további lehetőségek rejlenek, érdemes lenne más modelleket (például Attention U-Net) kipróbálni. A három különböző daganattípus elkülönítése egy összetettebb feladat, amely az elért eredményeket figyelembe véve jól megoldhatónak ígérkezik.

## Hivatkozások

- [1] Nabila Abraham és Naimul Mefraz Khan. “A novel Focal Tversky loss function with improved attention U-Net for lesion segmentation”. (2018). URL: <https://arxiv.org/pdf/1810.07842.pdf>.

- [2] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J.S. Kirby és tsai. “Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features”. (2017).
- [3] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi és tsai. “Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge”. (2019). URL: <https://arxiv.org/abs/1811.02629>.
- [4] *Kaggle BraTS2020 Dataset*. URL: <https://www.kaggle.com/awsaf49/brats20-dataset-training-validation>.
- [5] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby és tsai. “The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)”. (2015).
- [6] Olaf Ronneberger, Philipp Fischer és Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. (2015). URL: <https://arxiv.org/pdf/1505.04597.pdf>.