Segmentation of brain tumor regions in MRI scans

Eamonn Tweedy, PhD

The dataset and the tumor segmentation problem

We use the 2020 version of BraTS (Brain Tumor Segmentation) training dataset[1][2][3], consisting of 369 samples. For each:

- MRI images in three volumes: T1-CE, T2, and T2-FLAIR
- ground-truth segmentation: each voxel is one of (0)background, (1)necrotic/non-enhancing tumor core, (2)peritumoral edema, or (3)Gd-enhancing tumor.

Goal: given 3-dimensional MRI data from a brain scan, to identify the following three tumor regions:

- Enhancing tumor (ET) $\leftrightarrow 3$
- Tumor core (TC) $\leftrightarrow 1, 3$
- Whole tumor (WT) $\leftrightarrow 1, 2, 3$

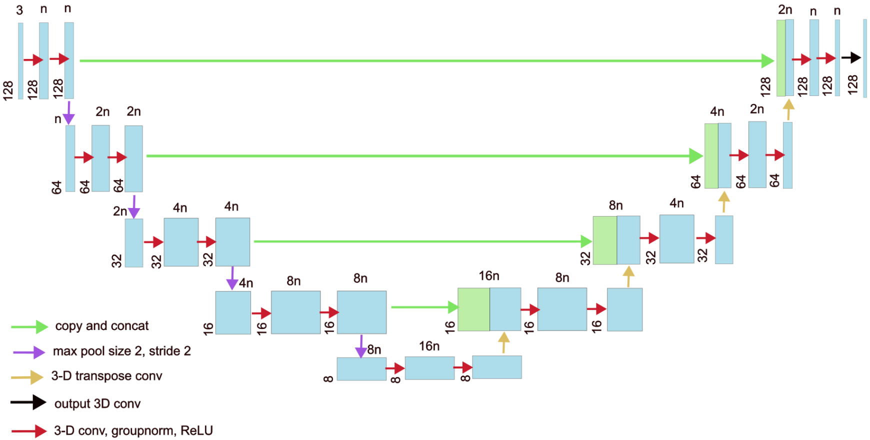
---

[1] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)", IEEE Transactions on Medical Imaging 34(10), 1993-2024 (2015) DOI: 10.1109/TMI.2014.2377694

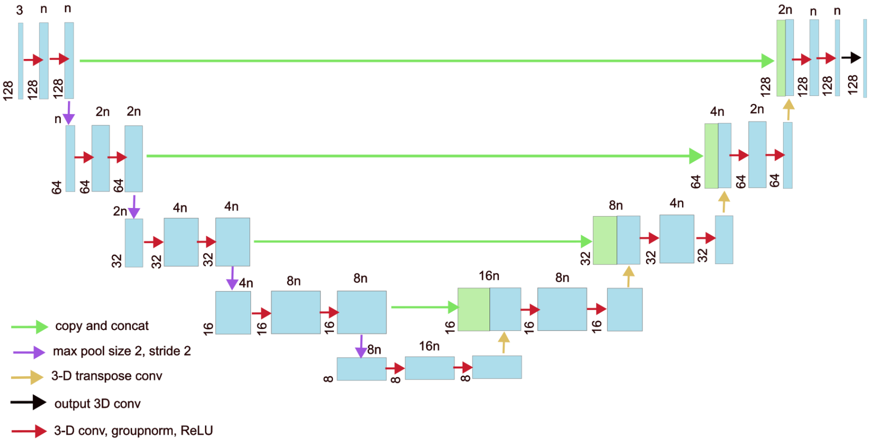
[2] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J.S. Kirby, et al., "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", Nature Scientific Data, 4:170117 (2017) DOI: 10.1038/sdata.2017.117

[3] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, et al., "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge", arXiv preprint arXiv:1811.02629 (2018)

Model class - CNN with "U-Net" architecture



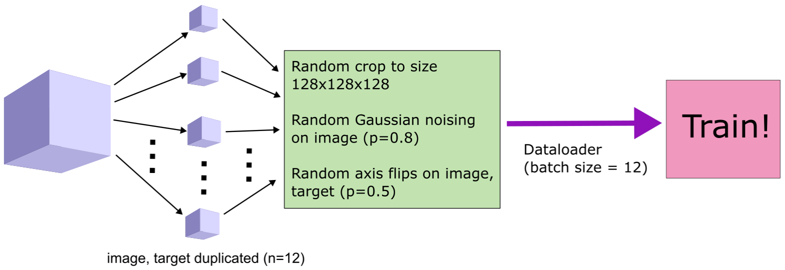
Model class - CNN with “U-Net” architecture



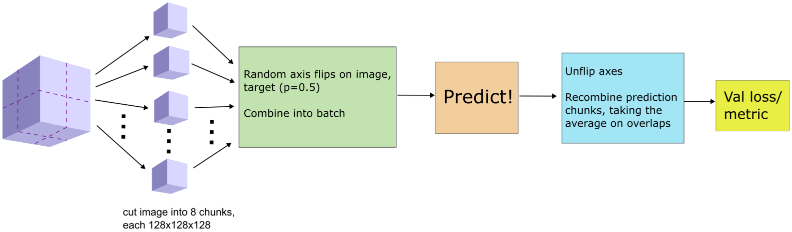
We had the best training and evaluation results with $n = 16$

Data pipelines

Training pipeline with data augmentation:



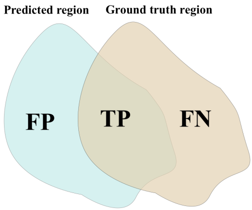
Inference pipeline with TTA:



Dice metric

Predictions $\hat{y}$ evaluated using the Dice metric (a voxel-wise F_1 score):

$$Dice(\hat{y}, y) = \frac{2\sum_n \hat{y}_n y_n}{\sum_n (\hat{y}_n + y_n)} = \frac{2TP}{2TP + FP + FN}$$



- Computed individually for each sample and each class label (ET, TC, WT) and averaged over samples
- If a class if missing from a sample, the score is binary

Training and loss function

As a loss function we use the sum $D + F$ of:

- Dice loss[4], which measures overlap between the predicted probabilities p and the ground truth segmentation y:

$$D(p, y) = 1 - 2\frac{\sum_n p_n y_n + \epsilon}{\sum_n (p_n + y_n) + \epsilon}$$

[4] Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi. *V-Net : Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation.* 2016, Fourth International Conference on 3D Vision (3DV).

[5] Zhu et al. *AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy, Medical Physics 2018*

Training and loss function

As a loss function we use the sum $D + F$ of:

- Dice loss[4], which measures overlap between the predicted probabilities p and the ground truth segmentation y:

$$D(p, y) = 1 - 2\frac{\sum_n p_n y_n + \epsilon}{\sum_n (p_n + y_n) + \epsilon}$$

- Focal loss[5], a variant of binary cross-entropy loss which down-weights the loss from high-confidence correct predictions:

$$F(p, y) = -\sum_n (1 - p_{t,n})^\gamma \ln(p_{t,n}), \qquad p_{t,n} = \begin{cases} p_n & \text{if } y_n = 1 \\ 1 - p_n & \text{if } y_n = 0 \end{cases}$$

we use $\gamma = 2$ as recommended by the authors.

[4] Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi. *V-Net : Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation.* 2016, Fourth International Conference on 3D Vision (3DV).

[5] Zhu et al. *AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy, Medical Physics 2018*

Training and loss function

As a loss function we use the sum $D + F$ of:

- Dice loss[4], which measures overlap between the predicted probabilities p and the ground truth segmentation y:

$$D(p, y) = 1 - 2\frac{\sum_n p_n y_n + \epsilon}{\sum_n (p_n + y_n) + \epsilon}$$

- Focal loss[5], a variant of binary cross-entropy loss which down-weights the loss from high-confidence correct predictions:

$$F(p, y) = -\sum_n (1 - p_{t,n})^\gamma \ln(p_{t,n}), \qquad p_{t,n} = \begin{cases} p_n & \text{if } y_n = 1 \\ 1 - p_n & \text{if } y_n = 0 \end{cases}$$

  we use $\gamma = 2$ as recommended by the authors.

We trained with the Ranger21 optimizer (AdamW with additional features) for 60 epochs with a maximum learning rate of $3e^{-3}$.

[4] Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi. *V-Net : Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation.* 2016, Fourth International Conference on 3D Vision (3DV).

[5] Zhu et al. *AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy, Medical Physics 2018*

Training results

Dice scores on validation set after 20 epochs:

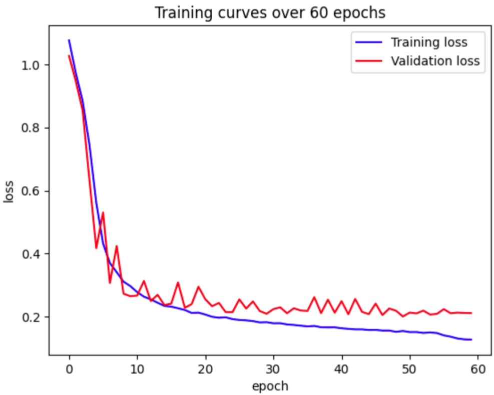
	mean	std dev	25th perc	75th perc
dice_et	0.723416	0.267365	0.674001	0.890223
dice_tc	0.700014	0.329232	0.673844	0.923397
dice_wt	0.835377	0.156345	0.839353	0.914439
dice_avg	0.752936	0.203102	0.685821	0.891766

...after 40 epochs:

	mean	std dev	25th perc	75th perc
dice_et	0.722973	0.285930	0.640575	0.905464
dice_tc	0.849390	0.179324	0.815981	0.944313
dice_wt	0.881843	0.122848	0.876280	0.946589
dice_avg	0.818069	0.153527	0.773626	0.918932

...after 60 epochs:

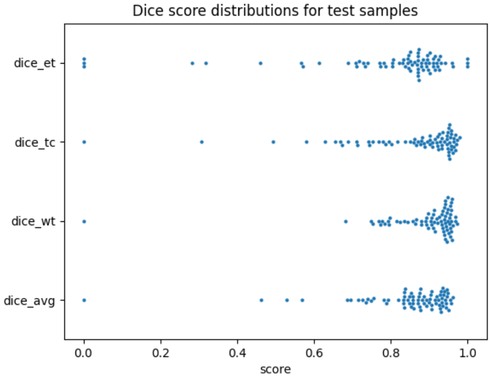
	mean	std dev	25th perc	75th perc
dice_et	0.743370	0.287089	0.739138	0.918957
dice_tc	0.842048	0.205489	0.831299	0.947882
dice_wt	0.881134	0.129708	0.883666	0.944003
dice_avg	0.822184	0.162340	0.790675	0.922468



Evaluation results

Dice scores on holdout test set after 60 epochs (20%, i.e. 74 samples):

	mean	std dev	25th perc	75th perc
dice_et	0.794943	0.210861	0.782632	0.900257
dice_tc	0.853168	0.160083	0.805927	0.949758
dice_wt	0.890323	0.123433	0.879018	0.948179
dice_avg	0.846145	0.139391	0.833853	0.927729



Thanks!

Thanks for your attention!
Questions?

Optimizer used during training

Used the Ranger21[6] optimizer - based on AdamW[7] with several improvements:

- Adaptive gradient clipping to control large gradients
- Gradient centralization and normalization for regularization and smoother training
- Positive-negative momentum and stable weight decay for improved generalization
- Norm loss for weight-space regularization
- "Explore-exploit" learning rate scheduler with linear warm-up (similar to cosine annealing schedule)

[6] L. Wright, N. Demeure. Ranger21: a synergistic deep learning optimizer. 2021. *arXiv preprint arXiv:2106.13731 [cs.LG]*, 2021.

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

Two key questions:

1. Why is the ET score the lowest?
2. For which samples does the model perform most poorly?

ET is the rarest label and is sometimes absent. Some test samples had ET score of zero:

image_332.npy	0.460043	0.673042	0.953435	0.695507	
image_158.npy	0.567380	0.741579	0.748036	0.685665	
image_329.npy	0.000000	0.768515	0.936324	0.568280	Ground truth missing ET, a little ET predicted
image_285.npy	0.000000	0.689380	0.896623	0.528668	
image_176.npy	0.281702	0.306897	0.797664	0.462087	
image_324.npy	0.000000	0.000000	0.000234	0.000078	Ground truth has a little ET, none predicted