The Tversky loss function and its modifications for medical image segmentation

Hidy Gábor
Project supervisor: dr. Lukács András

Eötvös Loránd Tudományegyetem
Institute of Mathematics

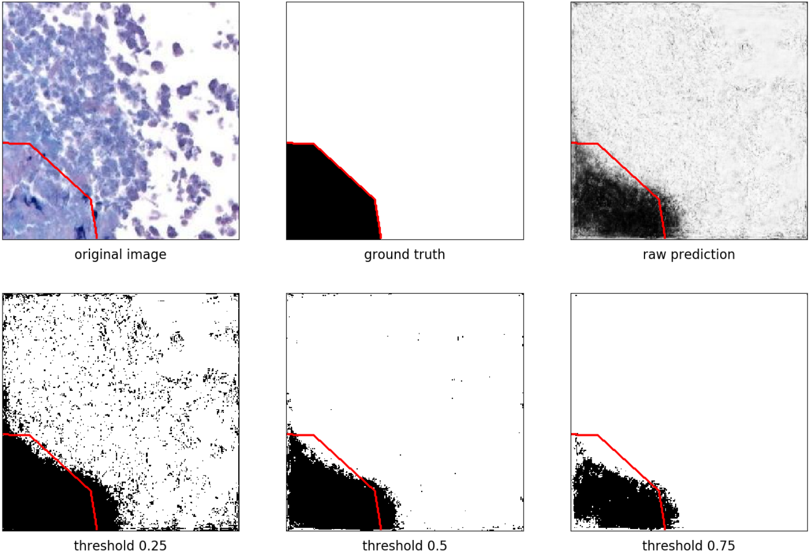
May 19, 2022

Semantic segmentation:

input: $\mathbf{x} \in [0,1]^{D \times H \times W}$ image,
output: $\hat{\mathbf{y}} \in [0,1]^{C \times H \times W}$ prediction
target: $\mathbf{y} \in \{0,1\}^{C \times H \times W}$ mask

$C = 1$: binary case

$y_i = 1$: positive pixel
$y_i = 0$: negative pixel


original image
ground truth
raw prediction
threshold 0.25
threshold 0.5
threshold 0.75

Tversky index

$$T_{\alpha,\beta}(\mathbf{y},\hat{\mathbf{y}}) = \frac{\mathrm{TP}}{\mathrm{TP} + \alpha\mathrm{FP} + \beta\mathrm{FN}}$$

$(\alpha, \beta > 0,\ \mathbf{y}, \hat{\mathbf{y}} \in \{0,1\}^M.)$

- doesn't count true negative $\Rightarrow$ good for imbalanced datasets
- $\alpha = \frac{1}{2} = \beta$: Dice index
- $\alpha = 1 = \beta$: Jaccard index (IoU score)

Tversky loss

$$1 - \mathcal{T}_{\alpha,\beta}(\mathbf{y}, \hat{\mathbf{y}})$$

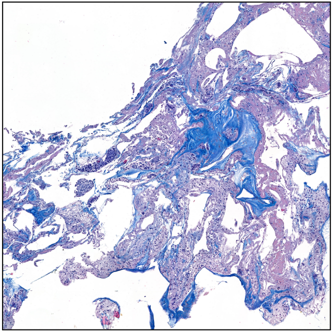
where

$$\mathcal{T}_{\alpha,\beta}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\mathbf{y}\hat{\mathbf{y}} + \delta}{\mathbf{y}\hat{\mathbf{y}} + \alpha(\mathbf{1} - \mathbf{y})\hat{\mathbf{y}} + \beta\mathbf{y}(\mathbf{1} - \hat{\mathbf{y}}) + \delta}$$

$(\alpha, \beta > 0,\ \mathbf{y}, \hat{\mathbf{y}} \in [0, 1]^M.)$

- equal to the Tversky index if $\mathbf{y}, \hat{\mathbf{y}} \in \{0, 1\}^M$
- if $\mathbf{y} \in \{0, 1\}^M$, then $\hat{\mathbf{y}} = \mathbf{y}$ is the unique minimum
- differentiable
- two approaches to calculate the loss ove a batch
  - imagewise loss: calculate for each image then average
  - batchwise loss: calculate over the whole batch

- histopathology slides from the Országos Korányi Pulmonológiai Intézet
- cancerous regions annotated by expert
- processed data: 2800 images, 360 of them positive

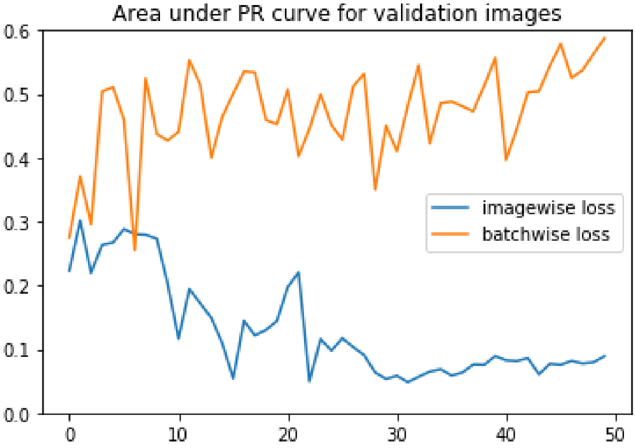


Goal: compare imagewise and batchwise Tversky

Details:

- model: U-Net
- loss: Dice loss
- optimizer: Adam
- learning rate: 10^{-6}
- two different tasks
  - segmentation of the whole dataset
  - segmentation of only the positive images

Results on the whole dataset



Results on the positive images

batch size	8		16		32		64	
loss domain	img	batch	img	batch	img	batch	img	batch
avg. prec. ↑	0.746	0.809	0.694	0.771	0.703	0.783	0.717	0.778
bal. acc. ↑	0.873	0.878	0.870	0.877	0.877	0.888	0.860	0.885
Dice idx. ↑	0.635	0.690	0.606	0.665	0.641	0.705	0.593	0.667
Jaccard idx. ↑	0.493	0.549	0.465	0.529	0.476	0.548	0.424	0.502
HD95 ↓	0.164	0.151	0.171	0.149	0.270	0.184	0.377	0.313

Label smoothing

Instead of the target $\mathbf{y} \in \{0, 1\}^M$, use $\tilde{\mathbf{y}} = (1 - \varepsilon)\mathbf{y} + \frac{\varepsilon}{C}\mathbf{1}$. $(0 < \varepsilon < \frac{1}{2}.)$

- combats vanishing gradients
- used in state-of-the-art networks (with crossentropy loss)
  - the minimum of crossentropy is still at $\hat{y} = \tilde{y}$

Claim 1

The maximum of $\mathcal{T}_{\alpha,\beta}(\tilde{\mathbf{y}}, \cdot)$ is assumed at $\hat{\mathbf{y}} = \mathbf{y}$.

$\Rightarrow$ label smoothing doesn't work with Tversky loss.

Claim 2

When using crossentropy loss, label smoothing is equivalent to adding $\varepsilon D\left(\frac{1}{C}\mathbf{1}\middle\|\hat{\mathbf{y}}\right)$, where D is the Kullback–Leibler divergence.

- this works with all losses
- $\varepsilon D\left(\frac{1}{C}\mathbf{1}\middle\|\hat{\mathbf{y}}\right)$ can be replaced with $\varepsilon D\left(\hat{\mathbf{y}}\middle\|\frac{1}{C}\mathbf{1}\right)$

Label noise penalty regularisation

$$\tilde{\mathcal{L}}(\mathbf{y},\hat{\mathbf{y}}) = \mathcal{L}(\mathbf{y},\hat{\mathbf{y}}) - \varepsilon\sum_{i=1}^{C}\log\hat{y}_i$$

Confidence penalty regularisation

$$\tilde{\mathcal{L}}(\mathbf{y},\hat{\mathbf{y}}) = \mathcal{L}(\mathbf{y},\hat{\mathbf{y}}) - \varepsilon H(\hat{\mathbf{y}})$$

[1] S. S. M. Salehi, D. Erdogmus, A. Gholipur. Tversky loss function for image segmentation using 3D fully convolutional deep networks. *Machine Learning in Medical Imaging*, September 2017, pp. 379–387.

[2] C. Szegedy *et al*. Rethinking the Inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

[3] G. Pereyra *et al*. Regularizing neural networks by penalizing confident output distributions. arXiv preprint, Jan 2017. arXiv:1701.06548v1 [cs.NE]

[4] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, November 2015, pp. 234–241.