Assignment 2

GSN Team

April 2020

1 Data

In Assignment 2, we solve the problem of figure segmentation. Data might be downloaded from here. It consists of two datasets - training data with 2K images and validation data, which consist of 500 images. Each image is an RGB image with a 128x128 resolution. Each mask has a form of a `numpy.array` of shape `(128, 128, 1)`. In each of these masks, the value of `0` implies a lack of mask for a given pixel and value of `255` - otherwise.

2 Task

One should implement the `U-NET` architecture for figure segmentation. By `U-NET` architecture we mean the *U-shaped* downsample-upsample architecture with skip connections. We leave the details of skip-connections (e.g., sum vs. concatenation), and students decide the exact design of each block.

2.1 U-NET architecture 1.5 pts

Implement a `U-Net` architecture network. It should contain downsample-upsample blocks with skip-connections. Students should decide about the exact details of blocks (e.g., number of convolutional layers), a way of downsampling (e.g., using strided convolution or pooling), and upsampling (e.g., transposed convolution, upsampling, subpixel upsampling). The student should be able to present a visual schema of the final architecture. Additionally - an *intersection over union* metric should be computed during training (however, there is no need to use it as a loss function in any form). The task is considered to be done if pixel accuracy exceeds 90%.

2.2 Data augmentation - 1.5pts

Data augmentation is crucial when the provided dataset is small. Because of that, we expect students to implement the following data augmentation procedures to extend data available for model training:

- horizontal symmetries,

- rotations by 90/270 degrees,

And use them during model training. Moreover - a final prediction should be obtained using the so-called self-augmentation procedure. In this procedure an input image is transformed using the transformation presented above (and additional *identity* transformation ;)). After obtaining the augmented inputs - they are fed into a model. Then the final prediction is obtained by averaging these outputs after the application of an inverse transform. We consider a task to be done when at least 4 different transformations are used for a final prediction.

2.3 Model results analysis - 1.0 pt

It is important to understand the downsides of the developed model. Because of that, we expect the student to analyze model downsides. This analysis should include:

- an analysis of cases with the highest loss on a train and validation sets,
- an analysis of the relationship between pixel prediction entropy and misclassification.

The student should provide a short, visual description of potential hard cases and the correlation between pixel-prediction entropy and misclassification error.

3 Extra Exercises

One can get additional points by:

- implement at least 8 kinds of model augmentations (max 0.5 pts).
- the author of a model with the best accuracy / IOU on additional test set provided during results presentation gets 1 additional point (1 person per lab group).
- the author of the fastest model (model execution time measured in Colab with GPU backend) with test accuracy of at least 90% gets 1 additional point (again 1 person per lab group). Note that the student could have a second, fastest model, different than the one for the best IOU task.

Models without IOU implemented does not take part in the contest.

4 Deadline

You should submit your solution by email by 23:59 on 19.05.2020 (Tuesday) to your lab teacher with email title "Assignment 2 - Deep neural networks". Your code will be inspected during the lab session following the deadline. Note that even if you are one minute late after the deadline, your solution will not be inspected. We have no mercy whatsoever so you better not count on that.