Deep Learning Lab Course

Gabriel Leivas Oliveira, Tonmoy Saikia

Due: December 4th, 2018

The goal of this exercise is to understand the main concepts and components of a decoder module of Fully Convolutional Networks (FCNs) for semantic segmentation. At the end send us a small report (1 or 2 pages) with the below described plot and fill the table with the obtained results for each configuration.

1 Implementing a decoder module for FCNs

First, we will implement a simple decoder module to upsample the final features from the provided encoder. Our FCN encoder consists of four convolution layers with stride 2 which will produce a final dimension at the end of the encoder $16x$ smaller than the input image. The first task will be to implement a module to upsample the features back to the original resolution with a single module. For that a transposed convolution with stride similar to the upsampling rate must be used. This experiment will be called configuration 1.

In order to implement this and all the other following configuration use the file **nets_definition.py** provided.

2 Hierarchical refinement and impact of skip connections

After you implement a single stage decoder to upsample the data back to image resolution you should look at multi-stage ways for upsample your data. Figure 1 shows the refinement block you should implement. The refinement cosists of the upsampling module you implemented previously, which is composed by transposed convolution and an activation function. After that you must check if the upsampled features and skip connection features are the same resolution, if not crop the largest one and concatenate both before sending to a convolution layer.

Once the refinement block is implemented it should be tested on three different configurations, respectively with one, two and three refinement blocks. All results of the four configurations must be reported in terms of their testing values and the maximum IoU obtained at each configuration. Consult **Test_Net.py**

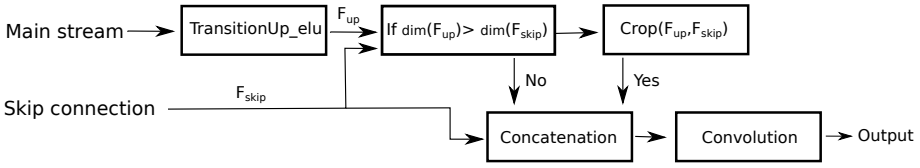

Figure 1: The refinement module to be implemented.

to generate the values to be plotted and to obtain the maximum Intersection Over Union(IoU) obtained value.

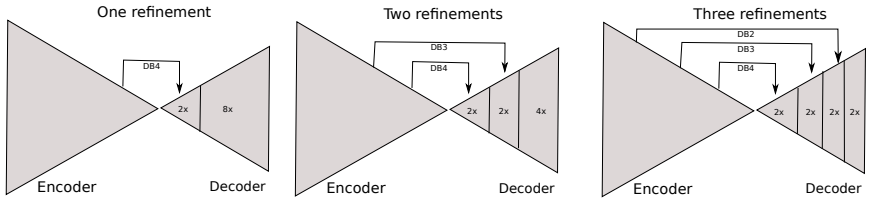

Figure 2: Configurations 2,3 and 4 to be implemented. Each configuration include a new refiment module with upsampling rate of $2\times$ and its corresponding skip-connection.

3 Dataset

The used dataset is the CamVid semantic segmentation dataset. The dataset is constituted by 468 training images and 233 testing images. The dataset is divided in 11 classes: Sky(0), Building(1), Pole(2), Road(3), Sidewalk(4), Tree(5), Sign(6), Fence(7), Car(8), Pedestrian(9), Cyclist(10).

Data must me downloaded following the readme file provide at the data/ folder.

4 Training and Testing

We must fill the gaps at **nets_definition.py** for the four different configurations. See **readme.me** for the proper use of the training and testing code, respectively **Train_Net.py** and **Test_Net.py**.