COMP448 – Homework #4

Fadime Tokmak | 63985

1. Shape Aware Loss Functions

1.a. Motivation of Shape Aware Loss Functions

Shape Aware Loss Functions are used for improving the performance of semantic segmentation with the help of target object's shape information. Using shape information, it is possible to force the segmented area to be in defined shape and get better accuracy in the pixel classification, especially in the boundary part of the target.

1.b. Example of Shape Aware Loss Function

In the paper [1], they combine two different loss function. The first part of loss function is used for keeping track of the pixel classification errors. It is simply log-loss or cross-entropy loss.

$$\hat{\boldsymbol{W}} = \arg\min_{\boldsymbol{W}} \sum_{n=1}^{N} \left(L_t \left(\{x^{(n)}, y^{(n)}\}; \boldsymbol{W} \right) + L_s \left(\{x^{(n)}, y^{(n)}\}; \boldsymbol{W} \right) \right)$$

$$L_t(\{x, y\}; \boldsymbol{W}) = -\sum_{i \epsilon \Omega_p} \sum_{j=1}^{M} y_i^j \log P(y_i^j = 1 | x_i; \boldsymbol{W}),$$

In the second part, they use a shape aware loss function to penalized inappropriate shape of segmented area. For get a shape aware loss function, for each image, they multiply log loss of each pixel with their created value that is calculated with D function and corresponds to appropriateness of the segmented shape with the ground truth shape.

$$L_s(\{x, y\}; \boldsymbol{W}) = -\sum_{i \epsilon \hat{\Omega}_p} \sum_{j=1}^{M} y_i^j E_i \log P(y_i^j = 1 | x_i; \boldsymbol{W}); \quad E_i = D(\hat{C}, C_{GT}),$$

D simply expresses "distance". To get E value, they compare extracted boundary of segmented area and ground truth boundaries and calculate average distance between these two shapes. In other words, they roughly try to estimate the size of the difference between segmented area and ground truth area.

[1] Al-Arif, S.M., Knapp, K., & Slabaugh, G. (2017). Shape-Aware Deep Convolutional Neural Network for Vertebrae Segmentation. *MSKI@MICCAI*.

2. Wasserstein Loss for GANs

2.a. Motivation of Wasserstein Loss

Since the GAN systems are generally use Minimax Loss Function that is a probability-based metric, it is possible to encounter vanishing gradient problem. Because if the discriminator learns very fast, then the loss function decreases very fast and the updates in the generator part will be very small. This issue requires a continuous and unlimited output rather than a probability-based output metric from discriminator. The second problem is that minimax does not directly express the differentiation power of the discriminator between two classes according to each other, in other words, it is not a metric that difference between output of real and output of fake class. For get a better feature selection and learning it is important to aim to increase the difference between probability distribution of each class, so the motivation of Wasserstein loss is creating a metric that express the output difference between two class.

2.b. Similarities and Differences

The main difference is that discriminator produce a continuous and unlimited a score that express the level of “appropriateness of the reality” of input rather than a probability of belonging a class. Other than that, generator and discriminator do not maximize and minimize same loss. The generator part is trying to maximize the score output of the discriminator for its productions and discriminator is trying to maximize the difference between scores of the real set and scores of the fake set.