Comparison of Loss Functions for Building Detection in Remote Sensing Images Using U-Net

Yosodipuro Nicholaus Danispadmanaba (48216644)
Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
yosodipuro-nicholaus@g.ecc.u-tokyo.ac.jp
Codalab username: nicholausdy

Abstract

Semantic segmentation is the task of separating groups of pixels within an image from another groups of pixels based on the semantic classes represented by those pixels. Based on what kind of semantic classes that need to be derived from an image, semantic segmentation can be further divided into more specific tasks. One such task is building detection, which is a form of binary semantic segmentation task, in remote sensing images obtained from Earth-orbiting sensors, e.g., satellites. Building detection is often crucial for land and population density research purposes. The current state-of-the-art approach in solving this kind of task is by using neural networks (NN). Since NN is a learning-based method, loss function is required to calculate prediction errors, i.e., loss value, when compared to the ground truth during the learning process. Loss function directly affects the performance of the model because loss value computed by the loss function is used as an input for the weight optimizer during the backpropagation process. Therefore, the choice of loss function greatly matters in building a general neural network including a neural network for building detection. In this report, we compare the performance of six loss functions that are specific to the task of binary semantic segmentation. The neural network architecture used in the experiment is U-Net, which is often used in semantic segmentation task. By comparing the Intersection over Union (IoU) scores on a remote sensing images dataset, we find that Dice-Sorensen Coefficient Loss (DSC Loss) function has the best performance out of the six loss functions with an IoU score of 0.5088.

1. Introduction

Semantic segmentation is the task of separating groups of pixels within an image from another groups of pixels based on the semantic classes represented by those pixels [2]. This task can be divided into more specific tasks depending on what kind of semantic classes that need to be derived from an image, e.g., tree detection, water detection, etc. Afterwards, based on the number of classes, semantic segmentation can be further divided into two types, which consist of binary semantic segmentation and multi-class semantic segmentation. In this report, we will only cover building detection task for remote sensing images gathered by satellites. Building detection itself is considered a binary semantic segmentation task because it only detects two classes of objects: buildings and non-buildings.

Currently, neural networks (NN) are extensively used for semantic segmentation task due to their capability to learn and detect hidden and complex image features that correspond with specific semantic classes. The learning capability of NN rests on the backpropagation process that enables NN to update their network weights using optimizers, e.g., Stochastic Gradient Descent (SGD), Adam, etc., in order to minimize error/residuals, i.e., loss value, between prediction and ground truth during training. Optimizer in turn requires a loss function to obtain the aforementioned loss value as an input for subsequent computation. Thus, the role of loss function is to compute error/residuals between prediction and ground truth during the training phase [7].

There are various loss functions that can be specifically used for binary semantic segmentation task. Thus, the goal of this report is to compare the performance of different loss functions in solving building detection task on a remote sensing images dataset. This report will cover six of such loss functions, which consist of: Binary Cross Entropy with Logits Loss (BCE Logits), Dice-Sorensen Coefficient Loss (DSC), BCE-Dice Loss, Intersection over Union (IoU) Loss, Focal Loss, and Tversky Loss. During experiments, each of those loss functions is plugged into a NN architecture commonly used for segmentation task, which is U-Net [8]. Intersection over Union (IoU) is then used as the performance metric to measure and compare the prediction

accuracy of U-Net models fitted with different loss functions.

2. Loss Functions for Binary Semantic Segmentation

2.1. Binary Cross Entropy with Logits Loss

Binary Cross Entropy with Logits Loss (BCE Logits) aims to integrate sigmoid function (σ) with binary cross entropy (BCE) loss function [6]. The role of sigmoid function (σ) is to map any real numbers as inputs into values between 0 and 1 as outputs. Meanwhile, the role of BCE is to measure the uncertainty between the target and predicted values distribution. The mathematical formulation for BCE Logits ($l(x, y)$) is given by Eq. (1),

$$\begin{aligned} l(x,y) &= BCE(\sigma(x), y), \\ BCE(x,y) &= -\frac{1}{N}\sum_{n=1}^{N} w_n[\, y_n \log x_n \\ &\quad + (1-y_n)\log(1-x_n)]\,, \\ \sigma(x) &= \frac{1}{1+e^{-x}} \end{aligned} \tag{1}$$

where x is the prediction result, y is the target output, N is the number of classes to be detected ($N = 1$ in the case of binary classification), and w_n is the assigned weight for a particular class.

2.2. Dice-Sorensen Coefficient Loss

Dice-Sorensen Coefficient (DSC) is used to measure the similarity between two samples [1]. In the context of semantic segmentation, this coefficient can be used to measure the similarity between the prediction result and the target output. With slight modification, DSC can be used as a loss function as given by Eq. (2),

$$\begin{aligned} l(X,Y) &= 1 - DSC(X,Y), \\ DSC(X,Y) &= \frac{2|X \cap Y|}{|X| + |Y|} \end{aligned} \tag{2}$$

where X is the prediction set, Y is the target set, and $|X|$ and $|Y|$ are the number of cardinalities of the two sets, i.e., the number of elements in each respective set.

2.3. BCE-Dice Loss

BCE-Dice Loss combines BCE and DSC loss functions by utilizing simple addition. The mathematical formulation is given by Eq. (3),

$$l(x,y) = BCE(x,y) + \ Eq.\ (2) \tag{3}$$

where x is the prediction result and y is the target output.

2.4. Intersection over Union Loss

Intersection over Union (IoU) or Jaccard Index is used to measure the similarity between two sets by calculating the ratio of overlap [3]. With slight modification, IoU can be used as a loss function as given by Eq. (4),

$$\begin{aligned} l(X,Y) &= 1 - J(X,Y), \\ J(X,Y) &= \frac{|X \cap Y|}{|X \cup Y|} \end{aligned} \tag{4}$$

where X and Y is the prediction and target set respectively.

2.5. Focal Loss

Focal Loss is used to deal with extremely imbalanced datasets. This loss function is formulated by adding a modulating factor to the cross entropy loss [5]. The mathematical formulation is given by Eq. (5),

$$l(x,y) = \alpha(1 - e^{-BCE(x,y)})^{\gamma} BCE(x,y) \tag{5}$$

where x is the prediction result, y is the target output, α is the balancing factor, and γ is the focusing parameter that directly affects the modulating factor.

2.6. Tversky Loss

Tversky Loss is also used to deal with imbalanced datasets. It is derived from the aforementioned DSC. This loss function works by adding weight values to different types of errors, i.e., false positives and false negatives. Therefore, different types of errors can be penalized differently according to the characteristics of the dataset used [9]. The mathematical formulation is given by Eq. (6),

$$\begin{aligned} l &= 1 - T, \\ T &= \frac{TP}{TP + \alpha FP + \beta FN} \end{aligned} \tag{6}$$

where TP is true positive predictions, FP is false positive predictions, and FN is false negative predictions. Meanwhile, α and β are the assigned weights for FP and FN respectively.

3. Experiments

Each of the loss functions listed on Sec. 2 is plugged into U-Net models for comparison purpose. The architecture of the U-Net models used in the experiments is composed of four downsampling layers that represent the depth of the network and Adam optimizer [4] for updating network weights. The model receives a $256x256$ pixels RGB image as its input. Afterwards, the model outputs a $256x256$ single-channel image as its prediction result.

The training and validation dataset contain 333 and 37 $uint8$ RGB PNG images respectively along with their corresponding labels. The size of both the training and validation images is $512x512$ pixels. Meanwhile, both the training and validation labels are single-channel $512x512$ pixels images with the building segments already annotated. Before being fed into the U-Net model, both of the training and validation images are augmented first. The training images are augmented by flipping them horizontally and vertically, rotating them for 90 degrees, transposing them, resizing them, and adding masks. Meanwhile, the validation images are augmented simply by resizing them and adding masks.

During training, the dataset is divided into multiple batches, each consisting of 8 images, to save memory. Weight updates are then done for each of those training batches. The training is done for 250 epochs, while the learning rate is set at 0.0001 to ensure smooth gradient descent. Separate training is done for each models fitted with different loss functions to compare the result later on.

During testing, an unlabeled dataset containing 100 RGB PNG images is fed into all of the trained models. The resulting segmented images outputted by each of the models fitted with different loss functions are then evaluated using IoU metric. The evaluation results for each of those loss functions are then compared with each other as can be seen on Tab. 1. Convergence graphs are also collected at the end of the evaluation as can be seen on Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5, and Fig. 6. For the convergence graphs, the horizontal axes show the number of epochs, while the vertical axes show the prediction accuracy. The implementation code of the whole training and testing pipeline can be seen in the following link: https://colab.research.google.com/drive/1K8rcbtt3EQRqj5J4-TyQaJXHPklc8ftH?usp=sharing.

For each DSC Loss, BCE-Dice Loss, and IoU Loss, smoothing constants are added to the equations to avoid division by zero. For Focal Loss, α and γ values are set at 0.8 and 2 respectively. Meanhile, for Tversky Loss, α and β values are set at 0.6 and 0.4 respectively. Implementation of the loss functions closely follows the following link: https://www.kaggle.com/bigironsphere/loss-function-library-keras-pytorch#Jaccard/Intersection-over-Union-(IoU)-Loss. Based on Tab. 1, comparative evaluation of the loss functions using IoU metric for building detection task clearly shows that DSC Loss has the best performance out of all six functions with the score of 0.5088.

4. Discussion

Aside from comparing the performance of each loss functions, it is also important to inspect and compare the convergence graphs of each of those loss functions. By inspecting the convergence graphs, we can grasp a better understanding of how the accuracy / error values converge into the global maxima / minima of the respective functions, especially in regards to the fluctuation and speed of convergence.

By looking into each of the convergence graphs, we can see that DSC Loss(Fig. 2), IoU Loss(Fig. 4), and Tversky

Table 1. Comparative evaluation of loss functions using IoU

Methods	IoU
BCE Logits	0.4818
DSC Loss	**0.5088**
BCE-Dice Loss	0.5074
IoU Loss	0.4827
Focal Loss	0.4588
Tversky Loss	0.5039
Ideal value	1.0000

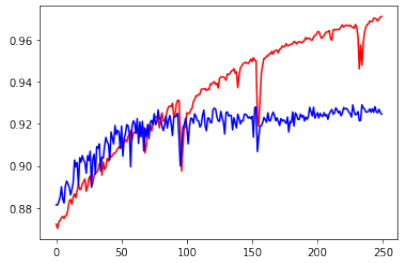

Figure 1. Convergence graph of U-Net with BCE Logits

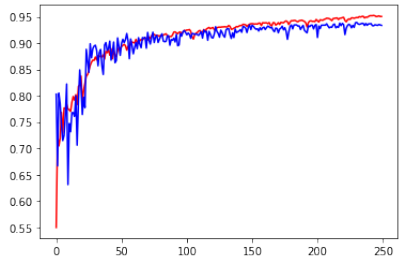

Figure 2. Convergence graph of U-Net with DSC Loss

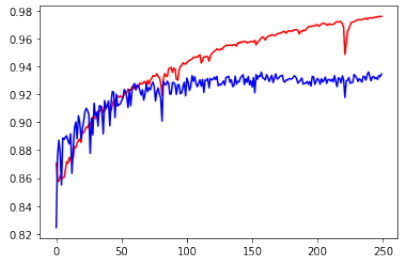

Figure 3. Convergence graph of U-Net with BCE-Dice Loss

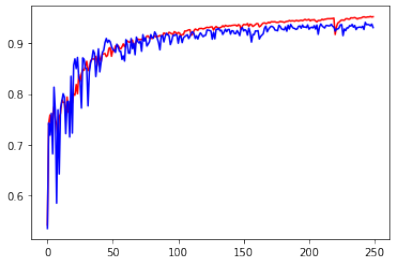

Figure 4. Convergence graph of U-Net with IoU Loss

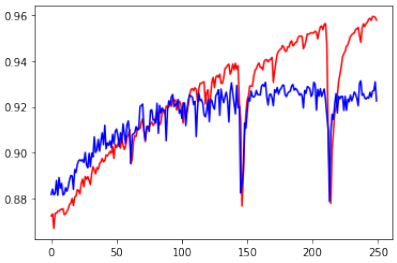

Figure 5. Convergence graph of U-Net with Focal Loss

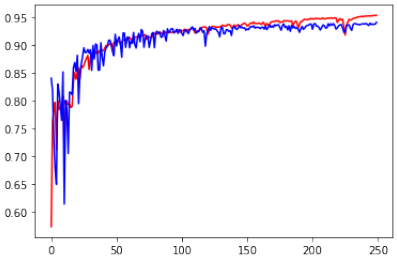

Figure 6. Convergence graph of U-Net with Tversky Loss

Loss(Fig. 6), had quite high prediction accuracy fluctuations compared to other loss functions, mainly on the earlier epochs. This means that those listed loss functions are significantly less stable on the earlier epochs compared to the other loss functions, although they gradually become more stable as the training progresses. Also, an interesting observation can be made on the convergence graph of Focal Loss(Fig. 5). High fluctuations suddenly happened on the middle and later epochs instead of the earlier epochs. This means that Focal Loss can become less stable with too many training epochs.

As for the speed of convergence, we can see from each convergence graph that each loss function has roughly similar speed of convergence. Regardless of the fluctuations, each of those loss functions tends to converge to the global maxima after 50 training epochs. This means that each loss function is equally capable of reaching the global maxima without needing too many training epochs.

5. Conclusion and Future Work

In this report, we compare the performance of six loss functions on the task of building detection in remote sensing images. Those six loss functions consist of BCE Logits, DSC Loss, BCE-Dice Loss, Focal Loss, and Tversky Loss. All of those functions were plugged into U-Net and then trained with the appropriate remote sensing dataset to obtain models capable of detecting buildings. Comparative evaluation shows that DSC Loss obtained the best performance out of the six functions with an IoU score of 0.5088.

For future work, we recommend further comparative evaluation of the six loss functions on different segmentation task and NN architectures. The aforementioned different segmentation task can mean either detecting different class of objects or even multi-class classification task. Also, it is known that there are other NN architectures for semantic segmentation beside U-Net. Therefore, comparative evaluation can be done on other NN architectures in order to ensure that the loss functions behave similarly when plugged into other NN architectures as well.

References

[1] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 2

[2] Dazhou Guo, Yanting Pei, Kang Zheng, Hongkai Yu, Yuhang Lu, and Song Wang. Degraded image semantic segmentation with dense-gram networks. *IEEE Transactions on Image Processing*, 29:782–795, 2020. 1

[3] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912. 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 2

[5] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 2

[6] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2013. 2

[7] Sebastian Raschka and Vahid Mirjalili. *Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt, 2019. 1

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1

[9] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. *Machine Learning in Medical Imaging Lecture Notes in Computer Science*, page 379–387, 2017. 2