Dual Focal Loss to address class imbalance in semantic segmentation

Md Sazzad Hossain*, John M. Betts, Andrew P. Paplinski

Faculty of Information Technology, Monash University, Melbourne, Australia

Abstract

A common problem in pixelwise classification or semantic segmentation is class imbalance, which tends to reduce the classification accuracy of minority-class regions. An effective way to address this is to tune the loss function, particularly when Cross Entropy (CE), is used for classification. Although several CE variants have been reported in previous studies to address this problem, for example, Weighted Cross Entropy (WCE), Dual Cross Entropy (DCE), and Focal Loss (FL), each has their own limitations, such as introducing a vanishing gradient, penalizing negative classes inversely, or a sub-optimal loss weighting between classes, which limits their ability to improve classification accuracy or ease of use. Focal Loss has proven to be effective at loss balancing by intensifying the loss on hard-to-classify classes, however, it tends to produce a vanishing gradient during backpropagation. To address these limitations, a Dual Focal Loss (DFL) function is proposed to improve the classification accuray of the unbalanced classes in a dataset. The proposed loss function modifies the scaling method of FL to be effective against a vanishing gradient. In addition, inspired by DCE, a regularization term has also been added to DFL to put a constraint on the negative class labels that further reduces the vanishing gradient effect and intensities the loss further on hard-to-classify classes. In this way, the proposed loss function offers a better training performance over CE, WCE, FL and

*Corresponding author

Email addresses: sazzad.hossain@monash.edu (Md Sazzad Hossain), john.betts@monash.edu (John M. Betts), andrew.paplinski@monash.edu (Andrew P. Paplinski)

DCE. Experimental results show that DFL provides better accuracy in every test run conducted over a variety of different network models and datasets.

Keywords: Cross entropy loss, deep neural networks, semantic segmentation, class imbalance.

1. Introduction

Image segmentation plays a key role in feature or object identification, and automatic labelling, for a diverse variety of applications, including medical imaging. Deep neural network-based semantic segmentation has recently gained popularity due to its high level of accuracy and efficiency compared to manual segmentation [1, 2]. A limitation of traditional approaches, such as intensity, edge or hand-crafted feature based segmentation is that accuracy tends to degrade in the presence of complex textures, or when image quality is low. Semantic segmentation, overcomes these difficulties by breaking down the image into a set of high-to-low level feature maps using encoder-decoder type deep neural network (DNN) models and adapting these to the characteristics of the data during training [3]. As semantic segmentation refers to pixelwise classification, it can suffer from class imbalance, whereby the network is biased towards classes having the greatest number of examples, which degrades overall performance [4]. To address this, a popular technique is to apply a weighting factor to the loss function so that the probabilistic decision is balanced between the classes, which improves classification accuracy, for example, [5], [6].

Finding the ideal class weight parameters is a challenging problem. A common technique is to set the weight factor based on the inverse of the number of pixels belonging to each class [7], [8], [6], [9]. However, this does not guarantee the most accurate semantic segmentation results, because the spatial distribution and correlation of the pixels has more effect on semantic segmentation than the pixel frequency for individual classes [10]. Another approach is to use trial and error [11], [12], [13]. Although this may yield suitable weightings, it requires manual adjustment of weights over multiple test runs, and does not guarantee an optimal classification. A recent approach to finding class weights is to identify "hard-to-classify" examples and assign greater weights to them. For example, Lin et al, [5] introduced a variant of cross entropy (CE), Focal Loss (FL), by defining the class weight factor as a function of the network's prediction confidence. In this way, difficult to classify examples had greater weights than easy to classify examples, with the resulting classification outperforming conventional weighted cross entropy (WCE).

A limitation of FL is that its gradient becomes significantly smaller than that of original CE when the predicted output from the classification layer prematurely approaches the actual output. This introduces a "vanishing gradient" effect that dramatically slows down the training of the network. An alternative approach by Li et al. [14] for achieving a better classification accuracy on imbalanced datasets is the use of a Dual Cross Entropy (DCE) loss function. This puts a constraint on negative class labels to alleviate the vanishing gradient effect, but has the limitation that the additional regularization term actually decreases the loss when the prediction error increases on a negative class, which undermines the training process.

In this paper, a novel Dual Focal Loss (DFL) function is proposed, whereby the beneficial properties of FL is integrated with that of the DCE. A modified formulation of FL is used to avoid early gradient saturation in addition to increasing the relative loss on hard-to-classify classes, and a modified formulation of DCE is used to increases the loss when the prediction error grows higher on a negative class. The accuracy and consistency of this new method is tested under varying conditions by performing semantic segmentation on four different datasets using two different fully convolutional network (FCN) models: DeepLabV3+ [15] and VGG19 [16]. The datasets used are: the MICCAI MRI dataset for prostate segmentation [17], transrectal ultrasound (TRUS) images for prostate segmentation, the Camvid dataset [18] and the Cityscape dataset [19]. Results show that the proposed loss function improves the semantic segmentation accuracy over some state-of-the-art loss functions, such as CE, WCE, FL, and GDL. The following section presents related research. The proposed methodology is then developed in detail. Model testing and evaluation is then performed and the results summarised.

2. Related Work

Many strategies have been proposed to address the class imbalance problem in classification problems, including semantic segmentation. One of the most common approaches is to balance the classes in the training data set. For example, Havaei et al. [20] proposed a two-phase training technique where the DNN is first trained on a dataset containing an equal number of labels of each class. A second training is performed, keeping the weights fixed, but tuning only the output layer using a more representational sample of the data. A similar strategy was adopted by Matthew [21], using training samples with equally distributed positive and negative examples. A limitation of these data sampling approaches is that they require large data sets in order that the balanced, sampled data is sufficient for training the DNN. Another approach is to replicate the data of minority classes, commonly known as oversampling [22]. Although this method has been shown to be effective in several studies [23], [24], [25], it can cause overfitting [26], [27]. To address this, researchers have tried alternative methods of resampling. For example, Jo and Japkowicz [28] divided the datasets into a number of clusters and then separately oversampled each cluster. Shen et al. [29] selectively chose the training examples to uniformly distribute the classes in each mini-batch. Guo and Viktor [30] produced synthetic datasets of difficult-to-classify examples of both majority and minority classes. One problem of oversampling is that it requires additional preprocessing of data, which is computationally expensive when the volume of data is large. An alternative approach is undersampling majority classes in order that they have a comparable size to the minority classes. Although Drummond et al. [31] show that undersampling is preferable to oversampling in real-world domains, one limitation is that it may eliminate data required by the DNN to learn essential image features, thereby reducing the effectiveness of the model [4].

Other researchers have addressed class imbalance by proposing modifications to the learning algorithm, or training scheme. Thresholding is one such method, whereby the outputs of a classifier are influenced by a threshold parameter. Lawrence et al. [32], set the threshold value using optimization. Richard and Lippmann [33], use a prior probability measure of the classes in the dataset based on class frequency. An alternative approach is to apply separate costs to each class. For example, Kukar et al. [34] impose a cost parameter to the learning rate, so that examples with high cost have a greater effect on weight updating. A similar, popular approach is to apply a class weight parameter to the loss function itself [7], [8], [6], [9]. This approach typically involves classwise magnification of the cross entropy (CE) loss function [14]. Weighted cross entropy [35, 6], the simplest version of this approach, applies a weighting factor to the CE loss to increase the penalty to the minority classes over the majorities in an attempt to balance the loss between imbalanced classes. Li et al. [14] proposed dual cross entropy (DCE) to increase the overall loss when prediction tends to favour a negative class. Lin et al. [5] proposed Focal Loss (FL) where the weighting factor is formulated as a function of output error so that the hard-to-classify examples will be given greater priority over the easy examples.

A key advantage of FL over WCE is that it offers a dynamic weighting factor instead of fixed weighting that adapts with learning accuracy, and thus focuses more on the minority classes, that may suffer inaccuracy due to class imbalance. As a different approach, Sudre et al. [36] propose a non-CE based loss function, Generalized Dice Loss (GDL), that combines the Dice coefficient with a class weighting method. A limitation of GDL is that it tends to perform poorly on small-sized arrays [37], because a few misclassifications in such cases may result in a larger loss, making it difficult to achieve proper convergence. This is why CE variants have gained a great popularity due to their suitability with different output structures. Although these CE variants, e.g. WCE and FL, have shown to alleviate the class imbalance issue to some extent, they do not put any additional constraint on negative classes that can potentially improve the convergence condition. This can be achieved by Dual Cross Entropy (DCE) [14], which uses an additional regularization term to penalize the negative classes directly, by increasing when the prediction error on negative classes decreases, and vice-versa. This improves the accuracy by alleviating the vanishing gradient problem, despite being counter-intuitive with respect to the neural network learning rule, and balancing loss between classes. In the following section, a modified formulation of the scaling factors in FL and DCE is presented. This overcomes the limitations of each method to achieve greater classification accuracy than either method alone could achieve.

3. Methodology

3.1. Weighted cross entropy loss

For the n-th input to the network, belonging to i-th class among c total classes, Cross Entropy (CE) loss, $L_{n\mathrm{CE}}$, is a measure of the deviation between the predicted output $\mathbf{z}_n$, and the expected output $\mathbf{y}_n$, given by

$$L_{n\mathrm{CE}} = -\mathbf{y}_n^\top \cdot \log \mathbf{z}_n = -\sum_i^c y_{i,n} \log(z_{i,n}). \tag{1}$$

Here, $\{\mathbf{z}_n = s(\mathbf{u}_n) : u_{i,n} \in \mathbb{R} \text{ and } z_{i,n} \in [0,1]\}$, where $s(\cdot)$ is a classifier activation function, typically softmax, given by

$$z_{i,n} = s(u_{i,n}) = \frac{e^{u_{i,n}}}{\sum_i^c e^{u_{i,n}}}. \tag{2}$$

When a dataset contains an unbalanced proportion of classes, the classifier tends to focus more on the class that has the greatest number of samples. This biases the classification performance towards the dominant class. Class imbalance is very common in semantic segmentation, where the number of pixels per class varies greatly. To address this, a common practice is to use weighted cross entropy (WCE), $L_{n\text{WCE}}$, as the loss function. This is a variant of standard CE with an additional class weight parameter, w_i, inversely related to the number of pixels in each class i, to balance the influence of each class. Thus

$$L_{n\text{WCE}} = -\sum_{i}^{c} w_i y_{i,n} \log(z_n). \tag{3}$$

3.2. Dual cross entropy

Dual cross entropy [14], $L_{n\text{DCE}}$, adds an additional regularization term to $L_{n\text{CE}}$ as

$$L_{n\text{DCE}} = -\sum_{i}^{c} y_{i,n} \log(z_n) + \beta \sum_{i}^{c} (1 - y_{i,n}) \log(\alpha + z_n). \tag{4}$$

Here, $\beta \geq 0$ and $\alpha > 0$ are chosen manually to control the intensity of the loss. The additional regularization term puts a constraint on the negative classes in order to reduce the vanishing gradient effect. However, the gradient of this regularization term decreases as $z_{i,n} \to 1$ for the negative classes, that is $\{z_{i,n} : y_{i,n} = 0\}$. This prevents $\{z_{i,n} : y_{i,n} = 0\}$ converging towards $\{y_{i,n} : y_{i,n} = 0\}$ as $\{z_{i,n} : y_{i,n} = 1\} \to \{z_{i,n} : y_{i,n} = 0\}$, which is counter-intuitive, but improves the behaviour of CE by mitigating the vanishing gradient effect. Therefore, it is likely that the regularization term can be optimised to improve the behaviour of the CE function further.

3.3. Focal loss

Focal Loss (FL) [5] is a variant of WCE that formulates the weighting factor as a dynamic value by expressing it as a function of the error between $z_{i,n}$ and $\{y_{i,n} : y_{i,n} = 1\}$, giving

$$L_{n\text{FL}} = -\alpha \sum_{i}^{c} (1 - z_{i,n})^{\gamma} y_{i,n} \log(z_n). \tag{5}$$

In this formulation, classes with low prediction accuracy, that is, hard-to-classify examples, result in greater loss than the easy-to-classify examples. Eq. (5) also shows that FL employs two additional scaling coefficients, $\alpha \in [0, 1]$ and $\gamma \geq 0$, to control the intensity of the loss. When $\gamma = 0$, FL becomes the WCE and α acts as the class weight parameter. When $\gamma \geq 1$, the weighting factor $(1-z_{i,n})^{\gamma}$ increases as $z_{i,n} \to 0$ and vice-versa. Therefore, when $z_{i,n}$ is lower, for a particular class, the weighting factor becomes proportionally larger, increasing the loss value for that class. However, a drawback of FL is that as $z_{i,n} \to y_{i,n}$, the gradient becomes prematurely smaller than that of CE, which exacerbates the vanishing gradient problem compared with CE.

3.4. Dual focal loss

We now propose a new loss function, Dual Focal Loss (DFL), by combining the mechanism of FL and DCE, and improving their individual scaling factors. First, to improve the condition of the DCE loss function, we modify the regularization term, $\beta(1-y_{i,n})\log(\alpha+z_n))$ to $\beta(1-y_{i,n})\log(\rho-z_{i,n})$ as shown in Eq. (6), where $\beta \geq 1$ and $\rho \geq 1$. It can be seen that when $z_{i,n} \to 1$ for negative classes, that is $\{z_{i,n} : y_{i,n} = 0\}$, the modified term penalizes the network proportionally unlike DCE. Appendix A shows in detail that this modification imposes a larger gradient than CE when the network tends to impose a false positive on a class. It also shows that the modified regularization term introduces a similar effect to FL, that is, it imposes a greater loss on hard-to-classify classes compared to CE.

$$L_{n\ \text{modified DCE}} = -\sum_{i}^{c}\big(y_{i,n}\log(z_n) + \beta(1-y_{i,n})\log(\rho - z_{i,n})\big), \qquad (6)$$

Second, the dynamic weight factor of FL, $\alpha(1-z_{i,n})^{\gamma}$ is modified as $\alpha(|y_{i,n} - z_{i,n}|)^{\gamma}$, where $\alpha, \gamma \geq 1$. This weight factor is then added to the CE loss term instead of mulitplying (as applied in FL given by Eq. (7)). This method results in a greater loss value and gradient in the original CE loss term, $L_{n\text{CE}}$ compared to FL, since $\{y_{i,n}, z_{i,n} : 0 \leq y_{i,n}, z_{i,n} \leq 1\}$. Moreover, it allows the loss function to explicitly take into account the error feedback from negative classes, making the derivative of the loss function more robust. Appendix B verifies these assumptions through detailed analysis of the behaviour of gradients of FL and our proposed modification to it (Eq. (7)). The analysis shows that the proposed modification to FL preserves the idea of FL by imposing a greater loss on hard-to-classify classes, and also improves the condition of the gradient while doing so.

$$L_{n\text{ modified FL}} = \alpha(|y_{i,n} - z_{i,n}|)^{\gamma} - \sum_{i}^{c} y_{i,n} \log(z_n). \tag{7}$$

Thus, our proposed loss function, $L_{n\text{DFL}}$ is

$$L_{n\text{DFL}} = -\sum_{i}^{c} \big(y_{i,n} \log(z_n) + \beta(1 - y_{i,n}) \log(\rho - z_{i,n}) + \alpha(|y_{i,n} - z_{i,n}|)\big). \tag{8}$$

Note that the value of the loss intensity control parameters in the loss functions, such as α, β, γ and ρ, can influence the performance of the loss functions arbitrarily, as observed in earlier studies [5, 14]. Therefore, in order to have a fair comparison and focus on the formulation of DFL, DCE, and FL, these parameters were set to 1 throughout this study.

Using Eq. (1),(4), (5) and (8), and putting $\alpha, \beta, \gamma, \rho = 1$, we get the derivatives of DCE, FL, and DFL as,

$$\frac{\delta L_{n\text{CE}}}{\delta z_{i,n}} = \frac{-y_{i,n}}{z_{i,n}} \tag{9}$$

$$\frac{\delta L_{n\text{DCE}}}{\delta z_{i,n}} = \frac{-y_{i,n}}{z_{i,n}} + \frac{1 - y_{i,n}}{1 + z_{i,n}} \tag{10}$$

$$\frac{\delta L_{n\text{FL}}}{\delta z_{i,n}} = \frac{-y_{i,n}}{z_{i,n}}(1 - z_{i,n}) + y_{i,n} \log(z_{i,n}) \tag{11}$$

$$\frac{\delta L_{n\text{DFL}}}{\delta z_{i,n}} = \frac{-y_{i,n}}{z_{i,n}} + \frac{1 - y_{i,n}}{1 - z_{i,n}} + \frac{|y_{i,n} - z_{i,n}|}{y_{i,n} - z_{i,n}}. \tag{12}$$

Fig. 1a and 1b show these derivatives with respect to $z_{i,n}$ for $\{y_{i,n} : y_{i,n} = 1\}$ and $\{y_{i,n} : y_{i,n} = 0\}$ respectively. It can be seen that when $\{y_{i,n} : y_{i,n} = 1\}$ FL is marginally greater than CE initially, but decreases quickly compared to CE as $z_{i,n} \to y_{i,n}$, and approaches to zero. This results in a slower learning rate of the network, and introduces the vanishing gradient problem when $z_{i,n} \to y_{i,n}$. By contrast, DFL always has a larger derivative than CE at any given point, and thus improves the condition of FL. When $\{y_{i,n} : y_{i,n} = 1\}$, CE and DCE produces identical derivative; however, when $\{y_{i,n} : y_{i,n} = 0\}$, the derivative of DCE decreases exponentially as $z_{i,n} \to 1$, while the derivative of CE and FL remains zero. This behaviour of DCE is counter-intuitive from the aspect that it penalizes the network for classifying the negative classes as true negative. We improve this behaviour by DFL, which exponentially increases the derivative when $\{z_{i,n} : y_{i,n} = 0\} \rightarrow 1$, as shown in Fig. 1b. Section 3.5 discusses why such change in derivative by DFL provides better convergence condition than FL and DCE.

3.5. Derivative of the loss function vs. learning

During the training phase, the output of the softmax classifier function, $z_{i,n}$, given by Eq. (2), is evaluated by a loss function, $L(z_{i,n}, y_{i,n})$. As shown in Eq. (A.3) in Appendix A, the derivative of L with respect to any softmax input, $u_{i,n}$ is

$$\frac{\delta L}{\delta u_{i,n}} = z_{i,n} \left(\frac{\delta L}{\delta z_{i,n}} - \sum_{k}^{c} \left(z_{k,n} \frac{\delta L}{\delta z_{k,n}} \right) \right). \tag{13}$$

Let $\frac{\delta L}{\delta z_{i,n}} = q_{i,n}$. Then,

$$\frac{\delta \frac{\delta L}{\delta u_{i,n}}}{\delta q_{i,n}} = \frac{\delta}{\delta q_{i,n}} \left(z_{i,n} \left(q_{i,n} - \sum_{k}^{c} (z_{k,n}\, q_{k,n}) \right) \right) = z_{i,n} - z_{i,n}^{2}. \tag{14}$$

Here, $\{z_{i,n} : 0 < z_{i,n} < 1\}$. Therefore, $\frac{\delta \frac{\delta L}{\delta u_{i,n}}}{\delta q_{i,n}} \rightarrow 0$ when $z_{i,n} \rightarrow 0$, or $z_{i,n} \rightarrow 1$. Otherwise, $\frac{\delta \frac{\delta L}{\delta u_{i,n}}}{\delta q_{i,n}} > 0$. This means that when the softmax function is used, $\frac{\delta L}{\delta z_{i,n}}$ proportionally affects $\frac{\delta L}{\delta u_{i,n}}$, and the effect is significant when the value of $z_{i,n}$ is within its mid-range. Therefore, if the loss value is increased, so too is the derivative $\frac{\delta L}{\delta u_{i,n}}$ during training. However, this does not ensure better learning, given by $L(w + \Delta w) < L(w)$, where w is a given weight in the neural network that is subjected to change Δw, and

$$\Delta w = f(\frac{\delta L}{\delta w}). \tag{15}$$

Eq. (15) shows that Δw will gradually decrease as the learning progresses in order to reach $w = \arg\min_w L(w)$. Due to the high degree of nonlinearity between w and L, $L(w)$ would usually contain multiple local minima. It is still an open research problem of how to avoid these local minima to reach the global minimum. Nonetheless, a smaller $\frac{\delta L}{\delta w}$ has a better chance of reaching the minimum, since a larger $\frac{\delta L}{\delta w}$ will result in a greater oscillation of w around the minimum point, reducing the likelihood of convergence. On the other hand, a smaller $\frac{\delta L}{\delta w}$ will introduce the vanishing gradient problem when the neural network contains a large number of hidden layers. Therefore, it is important to balance the vanishing gradient effect and the decaying of $\frac{\delta L}{\delta w}$ throughout the hidden layers. An efficient technique to achieve both effects together is to use residual/skip connections between layers [38], which is why this technique has gained an enormous popularity in recent years [39, 40].

The vanishing gradient, and decay of $\frac{\delta L}{\delta w}$ can also be addressed through loss functions by introducing a variable weighting factor so that $\frac{\delta L}{\delta w}$ is increasingly sensitive to error. This results in a larger gradient in shallow layers during the inital training process, but also achieves a reduced gradient in both the shallow and deep layers when the network is achieving greater accuracy. Although both FL and DCE use the variable weighting factors, as shown in the earlier sections, FL results in a vanishing gradient, and DCE hinders $\frac{\delta L}{\delta w}$ to decrease when the network is achieving greater accuracy, which prevents the deeper layers from optimal convergence. By contrast, DFL alleviates the vanishing gradient problem, and also reduces $\frac{\delta L}{\delta w}$ significantly as the predicted and actual output tend towards convergence. In this way, DFL offers a better control of the gradient throughout the hidden layers and thus improves the learning procedure.

4. Experimentation scheme

4.1. DNN models and datasets

Semantic segmentation tasks require a variant of DNNs known as fully convolutional networks FCNs, in which the fully connected layer of the DNN is replaced by convolution and upsampling layers to produce a pixelwise classification output of an image. In this study, two different models were used: DeepLabV3+ [15], and VGG19 FCN [3, 16] to evaluate the effectiveness of the proposed scheme. For DeepLabV3+, ResNet-18 [41] was used as the backbone architecture. The structural details and description of DeepLabV3+ and VGG19 FCN are given in references [15] and [3] respectively.

Four different image datasets were used: (1) 3D MHD formatted images of the prostate for 80 patients from the MICCAI Grand Challenge [17], (2) 3D DICOM formatted volumetric Transrectal Ultrasound (TRUS) images of the prostate for 5 patients from the Alfred Hospital, Melbourne; (3) the CamVid dataset [18]; and (4) the Cityscape dataset [19]. Fig. 2 shows sample images from each dataset. Since semantic segmentation is performed only on 2D images in this study, the volumetric images of MRI and TRUS datasets were converted into sets of 2D images. Each 2D MRI and TRUS image consisted of only two pixel classes: prostate and background. The prostate regions were manually segmented by expert radiologists to provide a ground truth comparison. The CamVid dataset contains a collection of streetview images of a city taken while driving. These images are comprised of 32 pixel classes such as road, car, pavement, and pedestrian. In this study, some of these classes have been merged together to simplify the training. This reduced the total number of classes to 11, consisting of: sky, cars, buildings, trees, fences, poles, pedestrians, bicyclists, road, pavement, and signposts. The Cityscape dataset contains similar imagery to the CamVid dataset. These images contain 30 pixelwise classes. The number of classes of the Cityscape dataset was reduced to 8 by grouping multiple similar classes to reduce training time. Images from all four datasets were resized to 224 x 224 pixels to match the default input image size of the FCN models. MRI and TRUS images consisted of single-channelled grayscale images, whereas CamVid and Cityscape images were three channel, RGB images. All images were in portable network graphics (PNG) format to avoid compression loss. A summary of the image sets used is given in Table 1. Apart from CE, WCE, FL and DCE, a non-CE based loss function - Generalized Dice Loss (GDL) [36] has also been included due to its popularity for class-imbalanced semantic segmentation tasks [36], [42], [43], [44], [45].

Table 1: Size and distribution of the datasets.

Datasets	Total size	Training size	Validation size	Testing size
MRI prostate	1378	1309	35	34
TRUS prostate	55	47	4	4
CamVid	701	665	18	18
Cityscape	3301	2000	326	975

4.2. Platform and computational resource

MATLAB was used to implement the FCN models, convert the 3D volumetric images into 2D image sets, and train and test the FCN models. The MASSIVE High Performance Computing (HPC) cluster at Monash University was used for all computation. The computing unit was comprised of: 13 processors, 120GB of RAM and an Nvidia Tesla K80 GPU. The training algorithm throughout was Adaptive Moment Estimation (ADAM) [46]. Training parameters were set as follows: initial learning rate = 0.0001, learn rate drop factor = 0.30, and learn rate drop frequency = 10. These parameters were kept constant for all loss functions, FCN models and datasets. However, the number of epochs varied for different cases, because early stopping, [47] was followed using the validation dataset to avoid overfitting.

4.3. Metrics

Two different metrics were used to quantify the segmentation performance:

- Intersection over Union (IoU) – a similarity based metric, and
- Hausdorff distance (HD) – a surface distance based metric.

Intersection over Union is given by,

$$\mathrm{IoU}(X,Y)=\frac{|X\cap Y|}{|X|+|Y|+|X\cap Y|}, \tag{16}$$

where X and Y are the corresponding pixels belonging to "ground truth" and "predicted" region of a particular class, with '| |' indicating the cardinality of the respective sets.

Hausdorff distance is given by,

$$\mathrm{HD}(S_1, S_2) = \max(D(A, B), D(B, A)), \tag{17}$$

where

$$D(A, B) = \max_{a \epsilon S_1}(\min_{b \epsilon S_2}(||a - b||))$$
$$D(B, A) = \max_{b \epsilon S_2}(\min_{a \epsilon S_1}(||b - a||)).$$

Here, a and b are two sets of points belonging to surface S_1 and S_2 respectively. HD is defined as the maximum of the minimum distances between the sets of points of two surfaces. In case of semantic segmentation tasks, S_1 and S_2 are 2D binary image matrices each belonging to a 3D one-hot array version of the corresponding categorical matrix, that is, the semantic segmentation output. Therefore, to measure HD for semantic segmentation, S_1 and S_2 are first converted into distance maps [48, 49] to measure the surface distance. In this study, Euclidean distance [49] was used for the distance map conversion.

4.4. Design of Experiments

Experiments were fully factorial, with two FCN models, four different datasets, and five popular state-of-the-art loss functions, as well as the Dual Focal Loss function. For each combination of FCN model, dataset, and loss function, 10 trials were performed using 10-fold cross-validation. The accuracy of each loss function was measured by mean and standard deviation of IoU and HD taken over the average accuracy of each cross-validation fold.

5. Results and discussion

5.1. Results

5.1.1. Performance with DeepLabV3+

Table 2 shows the performance of each loss function with the DeepLabV3+ FCN model across different datasets. Results show that DFL outperforms other five loss functions for all four datasets in terms of the mean and standard deviation of both IoU and HD. The second best performing loss function is DCE, while the relative accuracy of other loss functions was inconsistent across the different datasets. DFL shows the greatest increase in accuracy for the MRI dataset, and the lowest for the Cityscape dataset. The MRI dataset contains classes with the greatest imbalance, where the pixel frequency ratio of background to prostate region ranges is approximately 15. By contrast, the Cityscape dataset has the fewest imbalanced classes, where the highest pixel frequency ratio between two classes is approximately 5. A similar degree of improvement can be seen for the CamVid dataset, where the pixel frequency ratio between two classes having the greatest imbalance is approximately 6.

Table 2: Performance of the loss functions with DeepLabV3+

(a) Dataset: MRI prostate

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	86.34	1.47	1.77	1.18
WCE	83.02	1.71	1.96	1.14
DCE	88.41	1.64	1.75	1.19
FL	87.69	1.29	1.65	1.27
GDL	82.30	1.23	1.50	1.29
DFL	**91.26**	**1.09**	**1.51**	**1.17**

(b) Dataset: TRUS prostate

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	85.31	1.97	2.53	1.48
WCE	80.07	1.13	3.12	1.61
DCE	89.33	1.20	2.35	1.24
FL	88.47	1.04	2.46	1.22
GDL	87.27	1.01	2.48	1.25
DFL	**90.40**	**0.96**	**2.31**	**1.17**

(c) Dataset: CamVid

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	61.64	0.27	7.72	2.17
WCE	52.71	0.32	8.56	4.04
DCE	63.18	0.58	7.68	2.20
FL	59.93	0.43	7.83	3.08
GDL	62.09	0.34	7.41	3.91
DFL	**64.99**	**0.17**	**7.03**	**2.11**

(d) Dataset: Cityscape

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	57.88	0.42	4.40	1.88
WCE	47.72	0.62	4.76	1.60
DCE	61.54	0.38	4.10	1.67
FL	58.38	0.47	4.29	1.68
GDL	59.56	0.64	4.18	1.77
DFL	**62.39**	**0.3**	**3.99**	**1.45**

5.1.2. Performance with VGG19 FCN

Table 3 shows the performance of the loss functions when VGG19 FCN was applied to the four datasets, and shows that DFL again achieves the greatest accuracy on all four datasets. The next most accurate loss function over all four datasets appears to be DCE. These results, and those presented previously, indicate that the greatest improvement to accuracy is obtained for the MRI dataset having the greatest class imbalance. Our analysis of why this should be so is discussed later.

Table 3: Performance of the loss functions with VGG19 FCN

(a) Dataset: MRI prostate

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	86.54	2.38	4.83	2.45
WCE	84.23	1.14	4.96	2.35
DCE	84.09	2.11	4.99	1.98
FL	87.80	3.34	4.22	2.33
GDL	83.73	6.36	5.13	2.29
DFL	**88.65**	**1.76**	**4.12**	**1.81**

(b) Dataset: TRUS prostate

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	79.05	2.69	2.91	1.33
WCE	83.33	1.46	5.76	3.34
DCE	84.41	1.83	5.7	1.88
FL	83.62	1.68	5.16	1.02
GDL	82.92	2.27	2.52	1.44
DFL	**85.45**	**1.34**	**2.26**	**1.63**

(c) Dataset: CamVid

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	57.45	1.15	9.70	3.38
WCE	57.81	1.64	9.27	3.55
DCE	59.26	2.18	8.81	3.50
FL	57.35	1.41	9.94	3.97
GDL	57.93	2.36	9.38	3.79
DFL	**61.39**	**0.57**	**9.04**	**2.92**

(d) Dataset: Cityscape

Loss	**IoU (%)**		**HD (mm)**	
functions	Mean	Std dev.	Mean	Std dev.
CE	58.97	0.78	5.08	1.46
WCE	58.72	1.62	4.96	1.74
DCE	59.87	1.04	4.90	1.59
FL	58.92	1.08	5.11	1.32
GDL	58.87	1.12	5.16	1.29
DFL	**60.68**	**1.03**	**4.60**	**1.54**

A comparison of Tables 2 and 3 shows that DeepLabV3+ offered better segmentation accuracy than VGG19 FCN. Some qualitative segmentation results have been demonstrated in Fig. 3 to 6 using the given loss functions with DeepLabV3+.

5.2. Discussion

5.2.1. Influence of DFL on class imbalance

Results in Table 2 and 3 indicate that DFL has improved the classification accuracy for all datasets. This improvement is greater for the datasets having a greater class imbalance. This can be seen in Fig. 3 to 6, where DFL resulted in a greater improvement in segmentation quality for the MRI and TRUS datasets compared to the CamVid and Cityscape datasets. As discussed in Section 3, this is because the DFL results in a greater derivative than other loss functions for hard-to-classify classes. In addition, it alleviated the vanishing gradient problem by prohibiting the loss function approaching zero prematurely. This offered a better weight tuning in the shallower layers. The reason additional penalization for hard-to-classify classes improves the learning relates back to the typical class imbalance problem. During backpropagation, all elements belonging to the output feature map contribute to the gradient that trains the synaptic weights. Due to this, the loss derivatives received from a large number of pixels belonging to the majority class will have a greater impact on the updating of weights compared to the loss derivatives received from the small number of pixels of the minority class. This causes the weights to converge more towards the majority class than the minority class. Additional scaling of the loss derivatives belonging to the minority class pixels would balance the learning gradient. However, achieving this balance by assigning a fixed class-wise weight, such as WCE, derived from the prior distribution of the per-class pixels in the dataset, would not ensure an optimal training [50]. This is why WCE had a lower accuracy across the experimental results compared to CE. By contrast, over-penalizing the hard-to-classify classes, in functions like FL and DFL, resulted in a better balancing in the weight update, because they offered a flexible on-demand weighting of the gradient throughout the training as a function of error between actual and predicted output, making the weighting adaptive to the outcome. This in turn resulted in a more accurate classification with imbalanced datasets. Furthermore, Fig. 7 shows that DFL has the best training performance, measured as the reduction in Mean Absolute Error at each iteration, compared to the other loss function for all combinations of datasets and FCN models.

5.2.2. Inter-FCN model accuracy

Tables 2 and 3 shows that the overall accuracy level of DeepLabV3+ is higher than VGG19 FCN model across the given loss functions and datasets. The primary reason is that VGG19 FCN model does not contain any residual connections, unlike DeepLabV3+, and, as as discussed in Section 3.5, VGG19 FCN results in a poor flow of gradients to the shallower weight layers. Although DFL improved the accuracy compared to the other loss functions, residual connections would likely improve the accuracy level significantly. Apart from these, DeepLabV3+ offers a dilated separable convolution operation that minimizes the effect of local noises in the image. Moreover, DeepLabV3+ performs the pooling operation more effectively with Spatial Pyramid Pooling and Image Pooling technique [15].

5.2.3. Vanishing gradient vs. larger relative loss

Tables 2 and 3, and Fig. 7, show that DCE has the best performance, after DFL. This suggests that the vanishing gradient plays a significant role in the training of FCN models, since a key focus of DCE is to mitigate against the reducing gradient. FL appeared to be less accurate in training compared to other loss functions for most of the cases, possibly due to premature saturation of the gradient. However, due to a greater difference of relative loss between hard and easy-to-classify classes, FL dealt better with the class imbalance issue than CE as observed in MRI and TRUS segmentation results. This indicates that mitigating the vanishing gradient problem as well as a high relative loss difference between hard and easy-to-classify classes are both important for a better training of the network.

6. Conclusion

This paper has proposed a novel Dual Focal Loss (DFL) function to address the class imbalance and class weakness problems of semantic segmentation. DFL was primarily motivated by the idea of Focal Loss (FL) and Dual Cross Entropy (DCE), which are two recent variants of the Cross-Entropy (CE) loss function. DFL modifies the formulation of FL and DCE, and exploits the advantages of each in order to achieve a better outcome than either could achieve separately. DFL puts a greater loss value to hard-to-classify classes, but prohibits the early saturation of gradient unlike FL. DFL also introduces an additional constraint on negative class labels, however, with a better formulation so that it generates a better loss feedback compared with DCE. In this way, DFL offers a better gradient flow throughout the network, and also a better inter-class loss balance during backpropagation.

Experiments were conducted on two different network models and four different datasets with 10-fold cross-validation technique to investigate the performance of the loss functions in various conditions. Experimental results show that DFL provides greater accuracy than some of the most popular state-of-the-art loss functions including FL and DCE for different combinations of datasets and network models. The improvement of segmentation accuracy obtained by DFL was greater when the degree of imbalance of the dataset was greater, indictating its effectiveness at addressing class imbalance. Results also show that the vanishing gradient effect and large relative loss between unbalanced classes both play a significant role in classification accuracy.

A limitation of this study is that it ignores the effect of additional loss intensity control parameters. These are usually tuned manually, and can arbitrarily influence the performance of the loss functions and classification accuracy. Hence, it is likely that effective automatic tuning of the control parameters would make the loss function more robust. This is currently under consideration as a potential future study.

Acknowledgments

The TRUS dataset was provided by The Alfred Hospital, Melbourne, with relevant ethics approval of both Monash University and The Alfred Hospital. Computation was performed on the Massive™ HPC at Monash University.

References

[1] J. Ker, L. Wang, J. Rao, T. Lim, Deep learning applications in medical image analysis, Ieee Access 6 (2017) 9375–9389.

[2] L. Lu, Y. Zheng, G. Carneiro, L. Yang, Deep learning and convolutional neural networks for medical image computing, Advances in Computer Vision and Pattern Recognition (2017).

[3] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

[4] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intelligent data analysis 6 (5) (2002) 429–449.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988.

[6] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE international conference on computer vision, pp. 1395–1403.

[7] C. P. Ferdinand, E. F. MEE, Automatic liver and lesions segmentation using cascaded fully convolutional neural networks and 3d conditional random fields, MICCAI, p. PS4–18 (2016).

[8] Z. Tian, L. Liu, B. Fei, Deep convolutional neural network for prostate mr segmentation, in: Medical Imaging 2017: Image-Guided Procedures,

Robotic Interventions, and Modeling, Vol. 10135, International Society for Optics and Photonics, p. 101351L.

[9] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 5551–5560.

[10] S. R. Bulo, G. Neuhold, P. Kontschieder, Loss max-pooling for semantic image segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 7082–7091.

[11] C. F. Baumgartner, L. M. Koch, M. Pollefeys, E. Konukoglu, An exploration of 2d and 3d deep learning techniques for cardiac mr image segmentation, in: International Workshop on Statistical Atlases and Computational Models of the Heart, Springer, pp. 111–119.

[12] Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, O. Ronneberger, 3d u-net: learning dense volumetric segmentation from sparse annotation, in: International conference on medical image computing and computer-assisted intervention, Springer, pp. 424–432.

[13] J. Zhang, X. Shen, T. Zhuo, H. Zhou, Brain tumor segmentation based on refined fully convolutional neural networks with a hierarchical dice loss, arXiv preprint arXiv:1712.09093 (2017).

[14] X. Li, L. Yu, D. Chang, Z. Ma, J. Cao, Dual cross-entropy loss for small-sample fine-grained vehicle classification, IEEE Transactions on Vehicular Technology 68 (5) (2019) 4204–4212.

[15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), pp. 801–818.

[16] S. A. Kamran, A. S. Sabbir, Efficient yet deep convolutional neural networks for semantic segmentation, in: 2018 International Symposium on Advanced Intelligent Informatics (SAIN), IEEE, pp. 123–130.

[17] G. Litjens, R. Toth, W. van de Ven, C. Hoeks, S. Kerkstra, B. van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang, Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge, Medical image analysis 18 (2) (2014) 359–373.

[18] G. J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and recognition using structure from motion point clouds, in: European conference on computer vision, Springer, pp. 44–57.

[19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3213–3223.

[20] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, H. Larochelle, Brain tumor segmentation with deep neural networks, Medical image analysis 35 (2017) 18–31.

[21] M. Lai, Deep learning for medical image segmentation, arXiv preprint arXiv:1505.02000 (2015).

[22] M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Networks 106 (2018) 249–259.

[23] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: Review of methods and applications, Expert Systems with Applications 73 (2017) 220–239.

[24] N. Jaccard, T. W. Rogers, E. J. Morton, L. D. Griffin, Detection of concealed cars in complex cargo x-ray imagery using deep learning, Journal of X-ray Science and Technology 25 (3) (2017) 323–339.

[25] G. Levi, T. Hassner, Age and gender classification using convolutional neural networks, in: Proceedings of the iEEE conference on computer vision and pattern recognition workshops, pp. 34–42.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[27] K.-J. Wang, B. Makond, K.-H. Chen, K.-M. Wang, A hybrid classifier combining smote with pso to estimate 5-year survivability of breast cancer patients, Applied Soft Computing 20 (2014) 15–24.

[28] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM Sigkdd Explorations Newsletter 6 (1) (2004) 40–49.

[29] L. Shen, Z. Lin, Q. Huang, Relay backpropagation for effective learning of deep convolutional neural networks, in: European conference on computer vision, Springer, pp. 467–482.

[30] H. Guo, H. L. Viktor, Learning from imbalanced data sets with boosting and data generation: the databoost-im approach, ACM Sigkdd Explorations Newsletter 6 (1) (2004) 30–39.

[31] C. Drummond, R. C. Holte, C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, in: Workshop on learning from imbalanced datasets II, Vol. 11, Citeseer, pp. 1–8.

[32] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, C. L. Giles, Neural network classification and prior class probabilities, Springer, 1998, pp. 299–313.

[33] M. D. Richard, R. P. Lippmann, Neural network classifiers estimate bayesian a posteriori probabilities, Neural computation 3 (4) (1991) 461–483.

[34] M. Kukar, I. Kononenko, Cost-sensitive learning with neural networks, in: ECAI, pp. 445–449.

[35] S. Iqbal, M. U. Ghani, T. Saba, A. Rehman, Brain tumor segmentation in multi-spectral MRI using convolutional neural networks (CNN), Microscopy research and technique 81 (4) (2018) 419–427.

[36] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, M. J. Cardoso, Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations, Springer, 2017, pp. 240–248.

[37] K. C. Wong, M. Moradi, H. Tang, T. Syeda-Mahmood, 3d segmentation with exponential logarithmic loss for highly unbalanced object sizes, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, pp. 612–619.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[39] H. Yu, Z. Yang, L. Tan, Y. Wang, W. Sun, M. Sun, Y. Tang, Methods and datasets on semantic segmentation: A review, Neurocomputing 304 (2018) 82–103.

[40] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, J. Garcia-Rodriguez, A review on deep learning techniques applied to semantic segmentation, arXiv preprint arXiv:1704.06857 (2017).

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[42] G. Holste, R. Sullivan, M. Bindschadler, N. Nagy, A. Alessio, Multi-class semantic segmentation of pediatric chest radiographs, in: Medical Imaging 2020: Image Processing, Vol. 11313, International Society for Optics and Photonics, 2020, p. 113131E.

[43] C. Shen, H. R. Roth, H. Oda, M. Oda, Y. Hayashi, K. Misawa, K. Mori, On the influence of dice loss function in multi-class organ segmentation

of abdominal ct using 3d fully convolutional networks, arXiv preprint arXiv:1801.05912 (2018).

[44] G. Wang, J. Shapey, W. Li, R. Dorent, A. Demitriadis, S. Bisdas, I. Paddick, R. Bradford, S. Zhang, S. Ourselin, et al., Automatic segmentation of vestibular schwannoma from T2-weighted MRI by deep spatial attention with hardness-weighted loss, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019, pp. 264–272.

[45] S. Kumar, S. Conjeti, A. G. Roy, C. Wachinger, N. Navab, Infinet: fully convolutional networks for infant brain mri segmentation, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE, 2018, pp. 145–148.

[46] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[47] C. M. Bishop, Pattern recognition and machine learning, springer, 2006.

[48] D. W. Paglieroni, Distance transforms: Properties and machine vision applications, CVGIP: Graphical models and image processing 54 (1) (1992) 56–74.

[49] C. R. Maurer, R. Qi, V. Raghavan, A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2) (2003) 265–270.

[50] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.

Appendix A

Analysis of the behaviour of $\frac{\delta L}{\delta u_{i,n}}$, where $L = -\sum_i^C y_{i,n} \log(z_{i,n}) - \sum_i^C (1 - y_{i,n}) \log(1 - z_{i,n})$

1. Derivative of the loss function with respect to any input of softmax function

Given a softmax output, $z_{i,n}$, and its corresponding expected output, $y_{i,n}$, the derivative of the loss function, $L(z_{i,n}, y_{i,n}) = L_{n \text{ modified DCE}}$ (Eq. (6)), with respect to $z_{i,n}$ is

$$\frac{\delta L}{\delta z_{i,n}} = \frac{-y_{i,n}}{z_{i,n}} + \beta \frac{1 - y_{i,n}}{1 - z_{i,n}}. \tag{A.1}$$

From Bishop [47], the derivative of $z_{k,n}$ with respect to $u_{i,n}$, where $\{k : i \in k \text{ and } \mathrm{k} = 1, 2, 3, \ldots, c\}$, is

$$\frac{\delta z_{k,n}}{\delta u_{i,n}} = \begin{cases} z_{i,n}(1 - z_{k,n}), & \text{if } i = k \\ -z_{i,n} z_{k,n}, & \text{otherwise.} \end{cases} \tag{A.2}$$

Applying the chain rule, the derivative of L with respect to $u_{i,n}$ is

$$\frac{\delta L}{\delta u_{i,n}} = \sum_k \frac{\delta L}{\delta z_{k,n}} \cdot \frac{\delta z_{k,n}}{\delta u_{i,n}} = z_{i,n} \left(\frac{\delta L}{\delta z_{i,n}} - \sum_k \left(z_{k,n} \frac{\delta L}{\delta z_{k,n}} \right) \right). \tag{A.3}$$

Putting $\beta = 1$ in Eq. (A.1), Eq. (A.3) becomes

$$\begin{aligned} \frac{\delta L}{\delta u_{i,n}} &= z_{i,n} \left(\frac{-y_{i,n}}{z_{i,n}} + \frac{1 - y_{i,n}}{1 - z_{i,n}} - \sum_k \left(z_{k,n} \frac{-y_{k,n}}{z_{k,n}} + z_{k,n} \frac{1 - y_{k,n}}{1 - z_{k,n}} \right) \right) \\ &= \frac{z_{i,n} - y_{i,n}}{1 - y_{i,n}} - z_{i,n} \sum_k \frac{z_{k,n} - y_{k,n}}{1 - z_{k,n}} \\ &= \frac{z_{i,n} - y_{i,n}}{1 - y_{i,n}} - z_{i,n} \left(\frac{z_{j,n} - 1}{1 - z_{j,n}} + \sum_m \frac{z_{m,n}}{1 - z_{m,n}} \right) \end{aligned} \tag{A.4}$$

where $\{j : y_{j,n} = 1\}$, and $\{m : y_{m,n} = 0\}$.

2. Derivative of the loss function with respect to the positive class

When, $y_{i,n} = 1$, $i = j$. Therefore,

$$\begin{aligned}\frac{\delta L}{\delta u_{j,n}} &= -1 - z_{j,n}\left(-1 + \sum_m \frac{z_{m,n}}{1 - z_{m,n}}\right) \\ &= z_{j,n} - 1 - z_{j,n}\left(\sum_m \frac{z_{m,n}}{1 - z_{m,n}}\right). \qquad \text{(A.5)}\end{aligned}$$

Let, $e = |z_{i,n} - y_{i,n}|$.

Then, for $y_{i,n} = 1$, $|z_{j,n} - y_{j,n}| = -(z_{j,n} - 1)$, because $\{z_{j,n} : 0 < z_{j,n} < 1\}$.

Thus, $e = -z_{j,n} + 1, or, z_{j,n} = 1 - e$.

Hence, Eq. (A.5) becomes,

$$\begin{aligned}\frac{\delta L}{\delta u_{j,n}} &= 1 - e - 1 - (1 - e)\left(\sum_m \frac{z_{m,n}}{1 - z_{m,n}}\right) \\ &= -e - (1 - e)\left(\sum_m \frac{z_{m,n}}{1 - z_{m,n}}\right). \qquad \text{(A.6)}\end{aligned}$$

The above equation indicates the following:

- $e \geq 0$ and $\sum_m \frac{z_{m,n}}{1-z_{m,n}} \geq 0$. Therefore, $\frac{\delta L}{\delta u_{j,n}} \leq 0$.
- e has a direct relationship with$\sum_m \frac{z_{m,n}}{1-z_{m,n}}$. It is because $z_{j,n} = 1 - e$, and $z_{j,n} + \sum_m z_{m,n} = \sum_i z_{i,n} = 1$. It means that $\sum_m \frac{z_{m,n}}{1-z_{m,n}}$ decreases as $z_{j,n}$ increases. Thus, $(1 - e)$ has an inverse relationship with $\sum_m \frac{z_{m,n}}{1-z_{m,n}}$.
- Since $f(z_{m,n}) = \frac{z_{m,n}}{1-z_{m,n}}$ is a reciprocal function, a linear increase of $(1 - e)$ causes an exponential decrease of $\sum_m \frac{z_{m,n}}{1-z_{m,n}}$.

Therefore, in Eq. (A.6), as e increases, $\frac{\delta L}{\delta u_{j,n}}$ becomes exponentially large.

3. Derivative of the loss function with respect to the negative classes

When $y_{i,n} = 0$, $i = m$. Therefore,

$$\frac{\delta L}{\delta u_{m,n}} = \frac{z_{m,n}}{1 - z_{m,n}} - z_{m,n}\left(-1 + \sum_m \frac{z_{m,n}}{1 - z_{m,n}}\right)$$

$$\begin{aligned}
&= \frac{z_{m,n}}{1 - z_{m,n}} + z_{m,n} - z_{m,n}\left(\sum_{m} \frac{z_{m,n}}{1 - z_{m,n}}\right) \\
&= \frac{z_{m,n}}{1 - z_{m,n}} + z_{m,n} - z_{m,n}\left(\frac{z_{m,n}}{1 - z_{m,n}} + \sum_{l} \frac{z_{l,n}}{1 - z_{l,n}}\right), \\
&\quad \text{where } \{l : l \neq m, j \text{ and } y_{l,n} = 0\}. \\
\therefore \frac{\delta L}{\delta u_{m,n}} &= \frac{2z_{m,n} - 2z_{m.n}^{2}}{1 - z_{m,n}} - z_{m,n}\left(\sum_{l} \frac{z_{l,n}}{1 - z_{l,n}}\right) \\
&= z_{m,n}\left(2 - \sum_{l} \frac{z_{l,n}}{1 - z_{l,n}}\right). \qquad \text{(A.7)}
\end{aligned}$$

When $y_{i,n} = 0$, $e = |z_{i,n} - y_{i,n}| = |z_{m,n} - y_{m,n}| = z_{m,n}$. Consequently,

$$\frac{\delta L}{\delta u_{m,n}} = e\left(2 - \sum_{l} \frac{z_{l,n}}{1 - z_{l,n}}\right). \qquad \text{(A.8)}$$

There are plausible values of $z_{l,n}$ for which:

$$\left(2 - \sum_{l} \frac{z_{l,n}}{1 - z_{l,n}}\right) \leq 0. \qquad \text{(A.9)}$$

Therefore, for certain values of $z_{l,n}$, $\frac{\delta L}{\delta u_{m,n}}$ can be negative or zero. Although this behaviour looks problematic for proper training, it discontinues shortly after the training begins. It is because, as shown earlier, $\frac{\delta L}{\delta u_{j,n}} < 1$, and $\frac{\delta L}{\delta u_{j,n}} \to 1$. It means that $\frac{\delta L}{\delta u_{j,n}}$ always try to increase $z_{j,n}$ so that $z_{j,n} \approx 1$. This causes $\sum_{l} \frac{z_{l,n}}{1 - z_{l,n}}$ to decrease gradually as the training goes on, because $z_{j,n} + z_{m,n} + \sum_{l} z_{l,n} = 1$. As a result, even if $\sum_{l} \frac{z_{l,n}}{1 - z_{l,n}}$ was large enough at the beginning of the training that satisfies Eq. (A.9), it keeps decreasing exponentially as $z_{j,n}$ increases and passes the point where Eq. (A.9) is no longer satisfied. Thus, $\frac{\delta L}{\delta u_{m,n}}$ increases/decreases exponentially as $z_{m,n} = e$ increases/decreases as the training progresses.

Appendix B

Evidence that $|y_{i,n} - z_{i,n}| + L_{n\mathbf{CE}}$ produces similar $\frac{\delta L}{\delta u_{i,n}}$ to $L_{n\mathbf{FL}}$

---

1. Derivative of $L_{n\ \textbf{modified FL}}$ with respect to any input of softmax function

From Eq. (1), the standard CE loss is given by,

$$L_{n\mathrm{CE}} = -\sum_{i}^{c} y_{i,n} \log(z_n). \tag{B.1}$$

We propose to modify $L_{n\mathrm{FL}}$ by adding $\alpha \sum_i^c (|y_{i,n} - z_{i,n}|)^\gamma$ to $L_{n\mathrm{CE}}$, giving

$$L_{n\ \text{modified FL}} = -\sum_{i}^{c} y_{i,n} \log(z_n) + \alpha \sum_{i}^{c} (|y_{i,n} - z_{i,n}|)^\gamma. \tag{B.2}$$

Then for any element $z_{i,n}$, putting $\alpha, \gamma = 1$ in Eq. (B.2),

$$\frac{\delta L_{n\ \text{modified FL}}}{\delta z_{i,n}} = \frac{y_{i,n}}{z_{i,n}} - \frac{|y_{i,n} - z_{i,n}|}{y_{i,n} - z_{i,n}}. \tag{B.3}$$

Taking $L(z_{i,n}, y_{i,n}) = L_{n\ \text{modified FL}}$ in Eq. (A.3),

$$\begin{aligned}
\frac{\delta L}{\delta u_{i,n}} &= z_{i,n} \left(\frac{\delta L}{\delta z_{i,n}} - \sum_k \left(z_{k,n} \frac{\delta L}{\delta z_{k,n}} \right) \right) \\
&= z_{i,n} \left(\left(-\frac{y_{i,n}}{z_{i,n}} - \frac{|y_{i,n} - z_{i,n}|}{y_{i,n} - z_{i,n}} \right) - \sum_k z_{k,n} \left(-\frac{y_{k,n}}{z_{k,n}} - \frac{|y_{k,n} - z_{k,n}|}{y_{k,n} - z_{k,n}} \right) \right) \\
&= z_{i,n} \left(\left(-\frac{y_{i,n}}{z_{i,n}} - \frac{|y_{i,n} - z_{i,n}|}{y_{i,n} - z_{i,n}} \right) - A \right),
\end{aligned} \tag{B.4}$$

where $A = \sum_k z_{k,n} \left(-\frac{y_{k,n}}{z_{k,n}} - \frac{|y_{k,n} - z_{k,n}|}{y_{k,n} - z_{k,n}} \right)$.

Here, $\frac{|y_{k,n} - z_{k,n}|}{y_{k,n} - z_{k,n}} = 1$ when $y_{k,n} = 1$, and $\frac{|y_{k,n} - z_{k,n}|}{y_{k,n} - z_{k,n}} = -1$ when $y_{k,n} = 0$. Therefore,

$$A = \left(z_{j,n} \left(-\frac{1}{z_{j,n}} - 1 \right) + \sum_m z_{m,n} \right). \tag{B.5}$$

where $\{j : y_{j,n} = 1\}$ and $\{m : y_{m,n} = 0\}$.

Since $\sum_i^C z_{i,n} = 1$,

$$\sum_m z_{m,n} = 1 - z_{j,n}$$

$$\therefore A = (-1 - z_{j,n} + 1 - z_{j,n}) = -2z_{j,n} \tag{B.6}$$

1.1 Derivative of the loss function with respect to the positive class

When $y_{i,n} = 1$, $i = j$, thus

$$\begin{aligned}\frac{\delta L}{\delta u_{i,n}} &= z_{i,n}\left(\left(-\frac{1}{z_{i,n}} - 1\right) + 2z_{j,n}\right)\\ &= -1 - z_{i,n} + 2z_{i,n}^2\\ &= z_{i,n} - 1 - 2z_{i,n} + 2z_{i,n}^2\\ &= z_{i,n} - 1 + 2z_{i,n}\left(z_{i,n} - 1\right).\end{aligned} \tag{B.7}$$

Using the above equation, when $y_{i,n} = 1$, the graph of absolute error, $|z_{i,n} - y_{i,n}|$, vs. $\frac{\delta L}{\delta u_{i,n}}$ becomes,

1.2 Derivative of the loss function with respect to the negative classes When $y_{i,n} = 0$,

$$\frac{\delta L}{\delta u_{i,n}} = z_{i,n}\left((0+1) + 2z_{i,n}\right) = z_{i,n} + 2z_{i,n}z_{j,n}. \tag{B.8}$$

The above equation contains two variables: $z_{i,n}$ and $z_{j,n}$, where $z_{i,n} + z_{j,n} \leq 1$. Therefore, for $y_{i,n} = 0$, the graph of $|z_{i,n} - y_{i,n}|$, vs. $\frac{\delta L}{\delta u_{i,n}}$ will be a 3D surface plot as shown in Fig. B.2.

2. Derivative of $L_{n\mathbf{FL}}$ with respect to any input of softmax function

The focal loss function is given by,

$$L_{n\mathrm{FL}} = -\sum_{i}^{c} \alpha(1-z_{i,n})^{\gamma} y_{i,n} \log(z_n), \tag{B.9}$$

where α andγ two additional loss intensity parameters such that $0 \leq \alpha \leq 1$ and $\gamma > 0$.

Then, for any element, $z_{i,n}$,

$$\frac{\delta L_{n\mathrm{FL}}}{\delta z_{i,n}} = -\alpha(1-z_{i,n})^{\gamma}\, \frac{y_{i,n}}{z_{i,n}}\, y_{i,n} + \alpha\gamma(1-z_{i,n})^{\gamma-1} y_{i,n} \log(z_{i,n}) = B, \tag{B.10}$$

where $B = -\alpha(1-z_{i,n})^{\gamma}\, \frac{y_{i,n}}{z_{i,n}}\, y_{i,n} + \alpha\gamma(1-z_{i,n})^{\gamma-1} y_{i,n} \log(z_{i,n})$.

Taking $L(z_{i,n}, y_{i,n}) = L_{n\mathrm{FL}}$ in Eq. (A.3),

$$\frac{\delta L}{\delta u_{i,n}} = z_{i,n}\left(B - \sum_{k} (z_{k,n} B)\right). \tag{B.11}$$

When $y_{i,n} = 1$,

$$B = -\alpha(1-z_{i,n})^{\gamma}\, \frac{1}{z_{i,n}}\, y_{i,n} + \alpha\gamma(1-z_{i,n})^{\gamma-1} y_{i,n} \log(z_{i,n}), \tag{B.12}$$

and, when $y_{i,n} = 0$, $B = 0$. Therefore, when $y_{i,n} = 1$, $\frac{\delta L}{\delta u_{i,n}}$ becomes,

$$\begin{aligned} \frac{\delta L}{\delta u_{i,n}} &= \left(-\alpha(1-z_{i,n})^{\gamma} + \alpha\gamma(1-z_{i,n})^{\gamma-1} z_{i,n} \log(z_{i,n})\right) \\ &\quad - z_{i,n}\left(-\alpha(1-z_{j,n})^{\gamma} + \alpha\gamma(1-z_{j,n})^{\gamma-1} z_{j,n} \log(z_{j,n})\right), \end{aligned} \tag{B.13}$$

where $\{j : y_{j,n} = 1\}$.

2.1 Derivative of the loss function with respect to the positive class

When $y_{i,n} = 1$, $i = j$. Therefore,

$$\frac{\delta L}{\delta u_{i,n}} = D - z_{i,n} D = D(1 - z_{i,n}), \tag{B.14}$$

where $D = -\alpha(1-z_{i,n})^{\gamma} + \alpha\gamma(1-z_{i,n})^{\gamma-1} z_{i,n} \log(z_{i,n})$.

Although Lin et al. [5] found that $\alpha = 0.25$, and $\gamma = 2$ provides the best result in their study, $\alpha = 1$, and $\gamma = 1$ will be used in this analysis in order to avoid the influence of these additional parameters and focus primarily on the formulation of FL. So, using $\alpha, \gamma = 1$,

$$D = -(1 - z_{i,n})^2 + z_{i,n} \log(z_{i,n}). \tag{B.15}$$

The graph of the absolute error, $|z_{i,n} - y_{i,n}|$, vs $\frac{\delta L}{\delta u_{i,n}}$ is plotted using Eq. (B.14) and (B.15).

2.2 Derivative of the loss function with respect to the negative classes

When $y_{i,n} = 0$,

$$\begin{aligned} \frac{\delta L}{\delta u_{i,n}} &= 0 - z_{i,n} \left(\alpha(1 - z_{i,n})^{\gamma} + \alpha\gamma(1 - z_{i,n})^{\gamma-1} z_{i,n} \log(z_{i,n}) \right) \\ &= -z_{i,n} D \end{aligned} \tag{B.16}$$

The above equation contains two variables: $z_{i,n}$ and $z_{j,n}$, where $z_{i,n} + \sum_j z_{j,n} = 1$. Therefore, for $y_{i,n} = 0$, the graph of $|z_{i,n} - y_{i,n}|$ vs $\frac{\delta L}{\delta u_{i,n}}$ will be a 3D surface plot as shown in Fig. (B.4), where D is as given in Eq. (B.15).

3 Comparison between the derivatives of $L_{n\ \textbf{modified FL}}$ and $L_{n\textbf{FL}}$

From Fig. (B.1), (B.2), (B.3), and (B.4), it can be seen that, the intensity of both $(\frac{\delta L}{\delta u_{i,n}})_{\text{modified FL}}$ and $(\frac{\delta L}{\delta u_{i,n}})_{\text{FL}}$ keep increasing as $|z_{i,n} - y_{i,n}|$ increases. However, it is also noticeable that both derivatives have a local minimum when $y_{i,n} = 1$, and a local maximum when $y_{i,n} = 0$, near the point, $|z_{i,n} - y_{i,n}| \approx 1$, where their intensity starts to decrease as $|z_{i,n} - y_{i,n}|$ increases. Nevertheless, such local extrema do not prevent the network from convergence, because both derivatives return a non-zero negative value for any $|z_{i,n} - y_{i,n}| > 0$. The non-zero negative value ensures that $\{z_{i,n} : 0 < z_{i,n} < 1\}$ always tends to approach $y_{i,n}$.

A notable difference between $(\frac{\delta L}{\delta u_{i,n}})_{\text{modified FL}}$ and $(\frac{\delta L}{\delta u_{i,n}})_{\text{FL}}$ is that the ratio, $\frac{|(\frac{\delta L}{\delta u_{i,n}})_{\text{modified FL}}|}{|z_{i,n} - y_{i,n}|}$ is always greater than 1, whereas $\frac{|(\frac{\delta L}{\delta u_{i,n}})_{\text{FL}}|}{|z_{i,n} - y_{i,n}|}$ is less than 1 for almost the entire region of $|z_{i,n} - y_{i,n}| \in [0, 1]$. It means that $(\frac{\delta L}{\delta u_{i,n}})_{\text{modified FL}}$ results in faster convergence than $(\frac{\delta L}{\delta u_{i,n}})_{\text{FL}}$ and avoids early saturation of the gradient.