Technical Report: Implementing Partial Cross-Entropy Loss for Remote Sensing Image Segmentation

1. Introduction

In remote sensing image segmentation, pixel-level labeling is essential for accurate classification of different segments within an image. However, in many practical scenarios, only a subset of the pixels may be labeled, posing challenges for traditional segmentation methods. This report details the implementation of a custom loss function, Partial Cross-Entropy Loss (pfCE), designed to handle sparsely labeled masks. It also explores the impact of varying the number of labeled pixels and the use of spatial proximity linking to improve model performance.

2. Methodology

2.1 Partial Cross-Entropy Loss

The Partial Cross-Entropy Loss is a custom loss function designed to operate on images where only certain pixels are labeled. The loss is computed as a weighted average of the Focal Loss, focusing on the labeled pixels.
Here, the Focal Loss helps to address class imbalance by giving more importance to hard-to-classify pixels. The mask ensures that the loss is computed only over the labeled pixels, making it suitable for scenarios with sparse annotations.

2.2 Dataset and Preprocessing

The ISPRS Potsdam dataset, available on Kaggle, was used for this study. This dataset contains 2400 labeled images.
Each image's mask was processed to simulate sparse labeling and to apply the `link_pixel()` function for spatial proximity linking. This processing took approximately 1 second per pixel, leading to a 40-45 minute wait when loading the dataset using Google’s TPU in the fourth experiment.

To avoid completely removing any class from an image, careful attention was given to the random deletion process. The number of labeled pixels was varied across experiments to assess the impact on model performance.

3. Experiments

3.1 Experiment 1: Baseline with 700 Labeled Pixels per Class

- **Purpose:**To establish a baseline performance using a minimal number of labeled pixels per class.
- **Hypothesis:** Training with only 700 labeled pixels per class will result in poor model performance due to insufficient training data.
- **Experimental Process:**
    - A mask was generated by randomly selecting 700 labeled pixels per class in each image.
    - The model was trained using the Partial Cross-Entropy Loss function for 10 epochs.
    - No additional regularization techniques were applied.

- **Results:**

The model performed poorly, confirming the hypothesis. The exact results were not recorded due to an oversight, but the performance was significantly below expectations.

3.2 Experiment 2: Increased Labeling with 2100 Pixels per Class (2100 without tuning.ipynb)

- **Purpose:**To investigate whether increasing the number of labeled pixels per class improves model performance.
- **Hypothesis:** Increasing the number of labeled pixels to 2100 per class will improve performance but may lead to overfitting.
- **Experimental Process:**
    - The mask was modified to include 2100 labeled pixels per class.
    - The model was trained using the same setup as Experiment 1 for 10 epochs.
    - **File Reference:** `2100 without tuning.ipynb`

• **Results:**

◦ The model showed improved initial performance, but overfitting became evident as training progressed.
◦ **Training Loss (Epoch 10):** 0.0545
◦ **Validation Loss (Epoch 10):** 0.2350
The hypothesis was partially confirmed: while more labeled pixels improved the model's ability to learn, overfitting due to the larger dataset was observed.

3.3 Experiment 3: Addressing Overfitting with RegularizationTechniques (2100 with tuning.ipynb)

- **Purpose:**To test the effectiveness of regularization techniques in preventing overfitting when training with 2100 labeled pixels per class.
- **Hypothesis:** Applying regularization techniques such as IoU, weight decay, learning rate scheduling, and gradient clipping will reduce overfitting and improve generalization.
- **Experimental Process:**

◦ The same dataset with 2100 labeled pixels per class was used.
◦ The following regularization techniques were implemented:
  IoU metric: Used as an additional evaluation metric during training.
  Weight decay: Applied to penalize large weights.
  Learning rate scheduler: Adjusted the learning rate dynamically during training.
  Gradient clipping: Used to prevent exploding gradients.
◦ The model was trained for 10 epochs with these adjustments.
◦ **File Reference:** `2100 with tuning.ipynb`

• **Results:**

◦ The model's generalization improved significantly, with reduced overfitting.
◦ **Validation Loss (Epoch 10):** 0.0614
◦ **Validation IoU (Epoch 10):** 0.7490
◦ **Test IoU:** 0.7344
Better overall Performance.

3.4 Experiment 4: Enhancing Label Coverage with KDTree-Based Proximity Linking (500 with fitb.ipynb)

- **Purpose:**To explore the impact of linking spatially close pixels on model performance, especially when starting with a low number of labeled pixels.
- **Hypothesis:** Using KDTree-based proximity linking to increase the labeled area will enhance model performance without increasing the computational burden.
- **Experimental Process:**

◦ The number of labeled pixels was reduced to 500 per class.
◦ A KDTree-based proximity linking method was implemented:
  A function was created to link pixels within a specified distance (D), expanding the labeled area.
  KDTree allowed for efficient querying of all pairs of points within distance (D), reducing the computational complexity from (O(n^2)) to (O(n \log n)).

◦ The processing time for each mask led to a 40-45 minute wait when loading the dataset.
◦ **File Reference:** `500 with fitb.ipynb`

• **Results:**

◦ The model showed substantial improvement in performance, with a significant increase in IoU.
◦ **Validation IoU (Epoch 10):** 0.7890
◦ **Test IoU:** 0.8344
The hypothesis was confirmed: spatial proximity linking effectively improved the model's learning by increasing the labeled area, leading to better segmentation results.

4. Conclusion

The Partial Cross-Entropy Loss function is well-suited for scenarios with sparse pixel-level annotations. Through a series of experiments, it was demonstrated that the number of labeled pixels and the use of spatial proximity linking significantly impact model performance. The combination of these techniques led to a robust segmentation model capable of handling sparsely labeled remote sensing images.

Future work could explore further refinements in the proximity linking method and the application of this approach to different datasets and segmentation tasks.