

Assignment 10: Build Custom Training Function for Semantic Segmentation

40 Points

Deliverable: Notebook (.ipynb file) with all required code to complete the stated tasks. Answer all questions in Markdown cells.

Overview: The goal of this exercise is to build a custom function to instantiate, train, and evaluate a model.

Tasks

T1: Create a custom function to train a model. The function should meet the following criteria: (Up to 30 Points)

1. The function must accept DataSets for the training and validation set.
2. User should be able to define the number of input channels and the number of classes being predicted. Design the model for a multiclass problem.
3. It should internally instantiate training and validation DataLoaders with a user defined mini-batch size. The function should allow the user to choose to shuffle the training data and whether or not to drop the last mini-batch from both DataLoaders.
4. User should be able to select between cross entropy or Dice loss (can use Dice loss implemented by Segmentation Models).
5. Choose between using a UNet or DeepLabv3+ architecture as implemented with Segmentation Models.
6. Choose from a set of available backbone encoders including: ResNet-18, ResNet-34, ResNet-50, MobileNet-v2, or DenseNet-121.
7. Select from the following optimizers: stochastic gradient descent (SGD), RMSProp, or AdamW.
8. Set learning rates for each encoder block, the encoder, and the classification head separately.
9. Choose to use pre-trained weights from ImageNet within the encoder.
10. Freeze or unfreeze the encoder.
11. Implement a one-cycle learning rate policy in which the user can define the maximum learning rate.
12. The function should write the following to disk:
 - a. The best model (as defined by the epoch that provided the highest F1-score for the validation data) as a .pt file.
 - b. A CSV log file that provides the following information for each epoch: Epoch Number, Training Loss, Training Overall Accuracy, Training Class Aggregated Macro-Averaged F1-score, Validation Loss, Validation Overall Accuracy, and Validation Class Aggregated Macro-Averaged F1-score.
 - c. A CSV file that summarizes the user's settings including:
 - i. Number of training epochs
 - ii. Mini-batch size

- iii. Number of input channels
- iv. Number of classes differentiated
 - v. Total number of training samples
- vi. Base architecture used
- vii. Encoder backbone used
- viii. Whether or not the encoder was frozen
 - ix. Loss used
 - x. Optimizer used
- xi. Base learning rates for the different components of the model
- xii. Whether or not a one cycle learning rate policy was used
- xiii. If a one cycle learning rate policy was used, the maximum learning rate.

T2: Test the function by training a model for a few epochs on a semantic segmentation dataset. For simplicity, you can choose a semantic segmentation dataset made available by torchvision. (Up to 10 Points)