Homework Assignment 1:
Brain Tumor Segmentation with U-Net

Due time: *Friday, February 14, 11:59 PM*

Overview

In this homework, you will:

1. Implement a custom U-Net architecture with **4 downsamplers** and **4 upsamplers** to reinforce your understanding of how a U-Net is constructed.
2. Use the **MSD (Medical Segmentation Decathlon) Brain Tumor** dataset.
3. Perform **5-Fold Cross Validation** on the training set:
   - In each fold, train on 80% of the images (training split) and evaluate on the remaining 20% (validation split).
4. **Train your U-Net for 30 epochs** on each fold.
5. Plot the **training loss** and **Dice Score Coefficient** (DSC) for each epoch **within each fold**.
6. Combine the results from all folds to:
   - Plot the **mean** and **standard deviation** of the **training loss** across folds.
   - Plot the **mean** and **standard deviation** of the **Dice Score** across folds.
7. **Visualize** the segmentation performance for **5 randomly selected images** from the validation set in each fold. Your figure for each selected case must show:
   - **Column 1**: The input image slice (one channel if multi-channel).
   - **Column 2**: The ground truth segmentation label.
   - **Column 3**: The model's predicted segmentation.
   - **Display the IoU (Intersection-over-Union)** metric *above* the predicted segmentation.
8. **Answer conceptual questions** about cross-validation, loss functions for segmentation, and common metrics (Cross-Entropy Loss, Dice Coefficient, IoU).

Part A: Conceptual Questions

Please provide short written answers to each:

1. **What is k-fold cross-validation?**
   Define what cross-validation is, what "folds" represent, and why it's used in machine learning.
2. **What is the purpose of cross-validation?**
   Highlight how cross-validation helps in more robust evaluation of your model performance and how it might reduce variance in model performance estimates.
3. **What is an appropriate loss function for segmentation models?**
   - Define **Cross-Entropy Loss** mathematically and in words.
4. **Define the Dice Coefficient Score**
   - Provide the formula and interpret how it is suitable for medical image segmentation tasks.
5. **Define IoU (Intersection over Union)**
   - Provide its formula.
   - Describe how IoU relates to Dice and in which scenarios IoU might be preferable (or not).

Part B: Implementation

1. U-Net Construction

- Implement a **U-Net** with a **4 downsampling blocks and 4 upsampling blocks**.

2. Data Setup

- Use the **MSD Brain Tumor** dataset that was used in the class notebook.
- **Optional**: If you do not have local access, you can replicate the same environment in Google Colab or any environment with PyTorch installed.

3. 5-Fold Cross Validation

1. **Create 5 folds** from your dataset.
    - Randomly shuffle your list of image–label pairs.
    - Split them into **5 roughly equal subsets**.
    - For each fold k:
        - Use the k-th subset (20%) for **validation**.
        - Use the remaining 4 subsets (80%) for **training**.
2. **Within each fold**:
    - Train your U-Net model using training split.
    - Validate using the validation split.
    - **Train for 30 epochs**.
    - Record **training loss** and **Dice score** after each epoch.

4. Training and Logging

- **Loss Function**: Use **Cross-Entropy Loss** or a combination of CE + Dice Loss.
- **Optimizer**: Choose an optimizer.
- For each epoch, record:
    - The **training loss** (averaged across all mini-batches).
    - The **training Dice score** (averaged across mini-batches).
    - The **validation loss** (averaged across the validation subset).
    - The **validation Dice score** (averaged across the validation subset).

5. Plots & Statistical Analysis

1. **Per-Fold Plots**:
   - Plot the training loss vs. epochs for each fold.
   - Plot the Dice score vs. epochs for each fold.
   - Label the fold number clearly.
2. **Aggregate Plots**:
   - Combine the final results from all 5 folds to produce:
     - A single line plot for **training loss** (epoch on the x-axis, loss on the y-axis) that shows the **mean** across folds plus the **standard deviation** (e.g., shaded region or error bars).
     - A single line plot for **Dice score** that similarly shows the **mean** ± **standard deviation**.
   - Clearly label your axes and use a legend or error bars/shading to represent the standard deviation.

6. Segmentation Visualization

For **each fold**:

1. Select **5 random images** from the fold's validation split.
2. For each selected case, produce a **3-column figure**:
   - **Column 1**: Input image (one slice if multi-slice).
   - **Column 2**: Ground-truth segmentation (label).
   - **Column 3**: Model's predicted segmentation **with the IoU metric** displayed above or next to the predicted mask.
     - Compute IoU on the **segmentation mask** at the slice level.
     - Show the numeric value of IoU (e.g., as text on the figure).

Submission Guidelines

Please include your **Code Notebook**, **Written Answers document (PDF)**, and **Figures (if they are not embedded in the code notebook)** in a single zip file named as **[Lastname]_[Firstname]_HW_1**.

1. **Code Notebook**:
   - **[Lastname]_[Firstname]_HW_1.ipynb** containing:
     1. U-Net model implementation.
     2. Data splitting and 5-fold cross-validation loop.
     3. Training procedure (including your choice of hyperparameters).
     4. Loss and Dice plots for each fold and the aggregated mean ± std plots.
     5. Visualization of results with IoU displayed.
2. **Short Report / Written Answers**:
   - Answer the conceptual questions (Part A).
   - Summarize your main results (best Dice, average Dice across folds, best IoU, etc.).
3. **Figures**:
   - Attach or embed final figures in your notebook or provide separate files.
4. **Optional**:
   - Share any insights or difficulties you encountered (data size, GPU memory, overfitting, or unexpected behaviors).

Grading Rubric

1. **Correct U-Net Implementation (20%)**
   - Architecture matches the requested downsamplers/upsamplers.
   - Code compiles and runs without errors.
2. **5-Fold Cross Validation Setup (20%)**
   - Properly divides dataset into 5 folds.
   - Each fold trains on 80% / validates on 20%.
   - Clear reporting of per-fold performance.
3. **Plots (20%)**
   - Training loss and Dice score vs. epoch for each fold.
   - Mean ± standard deviation plots across all folds.
   - Axes labeled, curves are readable.
4. **Segmentation Visualization (20%)**
   - 5 random validation samples *per fold*.
   - 3-column figure: Input / Ground Truth / Prediction + IoU.
   - IoU displayed clearly.
5. **Conceptual Answers (20%)**
   - Definitions: cross-validation, Dice, IoU, Cross-Entropy.
   - Clarity on why cross-validation is used and how it benefits medical imaging with limited data.
   - Appropriate depth of explanation.

Total: 100%