Team Leabra

ADAPTING MODEL ARCHITECTURES FOR TUMOR SEGMENTATION IN BRAIN MRI WITH LIMITED GPU RAM/VRAM

Motivation & background

- In medical imaging, features of interest, like tumors, need to be detected in MRI scans.
- Traditionally, human radiologists perform this work, but radiologists are not perfect.
- Neural networks can be used as a second pair of eyes for radiologists.

Motivation & background

- Medical image segmentation is identifying the pixels in an image that contain a feature of interest.
- A popular convolutional neural network architecture for performing medical image segmentation is U-net.
- U-net can be trained much faster with GPU acceleration

Motivation & background

- Training U-net with GPU acceleration consumes a lot of GPU RAM/VRAM.
- WIth a limited setup, in terms of GPUs, it can affect training approaches: needing very small batch sizes, etc. to avoid running out of VRAM

Key Aims

- We want to see how reducing the size of U-net affects performance on tumor segmentation in brain MRI.
- We will reduce the size of U-net by decreasing the number of filters in each convolutional layer.
- Data from our experiments will serve as guidance for how U-net can be adapted when training with limited VRAM.
- We want to answer the question: Can U-net learn tumor segmentation in brain MRI, when less filters are used, to enable training in a setup with limited GPU resources?

Strategy

- Run 4 experiments to see how U-net with different amounts of filters performs on tumor segmentation in brain MRI.
- Each experiment consists of 5 runs, where a new model is initialized for each run to generate an average sense of performance with different numbers of filters.

- Experiment 1 : DS 1
  - U-net with all the filters
- Experiment 2 : DS 2
  - U-net with half the filters
- Experiment 3 : DS 4
  - U-net with a quarter of the filters
- Experiment 4 : DS 8
  - U-net with an eighth of the filters

Team Contributions

- Brian: paper
- David: algorithms reducing data size, solved training memory issue
- Christine: outlier detection, paper
- Joshua LaFever : outlier detection, dice loss, paper
- Joshua Ortner : data pipeline code, data/preprocessing in paper
- Lucas: training/testing code, preprocessing, paper

Demo

Questions