Objective:

Training Convolutional Neural Network (CNN) models on Breast Cancer Histology Images

Description of the assignment:

BreAst Cancer Histology (BACH) dataset is collected from four(4) different classes of ['Normal', 'benign', 'inSitu', 'Invasive'] histopathology cases. The dataset can be found from here. The objective of the assignment is to develop a Python package from scratch for training a CNN model on BACH dataset. Here are the steps needed to develop the package:

- BACH dataset is composed of photos and WSI. You only need to download the photos which consists of four different classes mentioned above. Each class consists of 100 photos of size 2048 x 1536 pixels. To harness a proper dataset for training, each photo is needed to be cropped into patches of size 224x224 from the original photo. The overlapping area from each photo should be 50%. Any patch coming from a particular photo intakes the same label for classification. This way you can create of dataset of few thousands of images with size 224x224. Break down your data into 70% train and 30% test subsets.
- CNN model for training: MobileNetV2. You can call the architecture from PyTorch library if your developing your codes in PyTorch. I am assuming you can find the same model in Tensorflow library as well if you are going to develop in tensorflow. Build your model by call the MobileNetV2. Opt-out the classifier and replace with a One(1) fully-connected layer taking the features from last convolutional layer and mapping to Four confidence scores pertaining to different classes
- Load your data in python and perform the following normalization/augmentation for training and testing:
    - Normalization: average RGB values are calculated from train data and deducted from every coming image either for training/testing
    - Augmentation: apply the standard augmentations such as flipping, rotating, translation that is coming in computer vision domain. Note: augmentation is only applied during training, but not testing
- Select your optimization (e.g. Adam) and setup a training library to intake dataset/ as well as the hyper-parameters (e.g. #epochs, batch size, learning rate, scheduler, etc) for training. Setup testing library for testing period
- train the network and then test the performance
- reporting your results: please report the following
    - plot training loss per epoch
    - plot training accuracy per epoch
    - plot testing loss per epoch
    - plot testing accuracy per epoch
    - transfer all train/test dataset via the CNN pipeline and record the feature vectors prior to linear classifier.
    - Apply t-SNE analysis on the feature vectors and demonstrate the distribution of two-dimensional attributes for both train/test with respect to different class representation