Image Analysis Project: Image Classification
REU – KSoC 2024

Please read instructions on the project's Canvas page before start working on your solution.

Project Objective

The objective of this project is to develop, evaluate, and test a medical image classifier using deep neural networks using both PyTorch and network architectures available from MONAI.

Key aspects of this project include

- **Handling medical images (MedMNIST)** [4]: You will experiment with MedMNIST, v2, a large-scale MNIST-like collection of standardized biomedical images, including 12 datasets for 2D and 6 datasets for 3D. All images are pre-processed into 28x28 (2D) or 28x28x28 (3D) with the corresponding classification labels.
- **Implementing deep network architectures for image classification**: You will gain hands-on experience with the nuances of neural network architectures and how they can be optimized for image recognition tasks. Additionally, you will explore and experiment with advanced architectures provided by MONAI.
- **Evaluating model performance**: You will build a model and critically evaluate its performance. This involves understanding evaluation metrics for image classification, such as accuracy and area under the receiver operating characteristic (ROC) curve[1].

Part 1: 2D image classification using PyTorch (50% of total points)

In this part, you will be working with a 2D MedMNIST dataset to construct and utilize a convolutional neural network (CNN). Your goal will be to train, validate, and test an image classifier using this network. Look at the Python script (classification-part1.py) for more information. Your aim is to complete the code and report test set results.

Exercise structure

1. **Dataset preparation and pre-processing**:
   - Define and perform data pre-processing such as normalization, resizing, and augmentation.
   - Use the same split of the dataset into training, validation, and test sets provided by MedMNIST
2. **Model building**
   - Implement a CNN model for image classification using a given architecture design.
3. **Training the model:**
   - Compile the model with an appropriate optimizer, loss function, and metrics.
   - Train the model on the training dataset while validating on the validation set.
   - Implement callbacks like model checkpoint and early stopping for efficient training.
4. **Evaluation on test set:**
   - Evaluate the trained model on the test dataset.
   - Analyze the model's performance using metrics such as accuracy and loss.

[1] An ROC curve is a graph showing the performance of a classification model at all classification thresholds, plotting the True Positive Rate and False Positive Rate.

Part 2: 2D image classification using MONAI pretrained backbones (50% of total points)

In this part, you will use with the MedMNIST dataset to explore and analyze different image classification network architectures available in MONAI, a framework specifically tailored for medical image analysis. Your goal is to evaluate and compare the effectiveness of three different network architectures from distinct architectural families in MONAI. Look at the Python script (classification-part2.py) for more information.

Exercise Structure

1. **Dataset preparation and pre-processing**:
   - Use torch transforms as in Part #1 to define and apply data pre-processing steps such as normalization and resizing to the MedMNIST dataset.
   - Use the same split of the dataset into training, validation, and test sets provided by MedMNIST.
2. **Model building and exploration**:
   - Use different architectural families, namely CNN-based (ResNet and DenseNet) and transformer-based (ViT) models, for image classification from MONAI's collection.
   - Ensure each model is appropriately adapted to handle the input data from the MedMNIST dataset.
3. **Training the models**:
   - Configure each model with a suitable optimizer and loss function. Utilize relevant metrics to monitor the training process.
   - Train each model on the MedMNIST training dataset, employing training and validation workflows that you have developed for Part #1.
4. **Evaluation and comparison**:
   - Evaluate each trained model on the test dataset. Focus on comparing their performance based on metrics such as accuracy, loss, and area under curve.
   - Analyze how different architectures influence the model's performance and suitability for medical image classification tasks.

Your task here is to leverage different architectural families for image classification, utilizing models from MONAI's extensive collection. Specifically, you are to use CNN-based models, such as ResNet[2] and DenseNet[3], as well as a transformer-based model, ViT (Vision Transformer)[1].

1. **ResNet (Residual Network)**[2]: ResNet introduces the concept of residual blocks that have residual connections. These connections allow the network to skip one or more layers, effectively addressing the vanishing gradient problem and enabling the training of extremely deep neural networks. ResNet variants, such as ResNet-50, ResNet-101, and ResNet-152, differ in their depth, denoted by the number in their names.
2. **DenseNet (Densely Connected Convolutional Network)**[3]: In a DenseNet, each layer is connected to every other layer in a feed-forward fashion. This dense connectivity pattern ensures maximum information flow between layers in the network. Unlike ResNet, DenseNet concatenates outputs from previous layers instead of summing them, to preserve and reuse features from earlier layers. DenseNet variants, such as DenseNet-121, DenseNet-169, DenseNet-201, signifies the total number of layers in the network that have trainable weights.
3. **ViT (Vision Transformer)**[1]: ViT treats an image as a sequence of patches and applies self-attention mechanisms to capture global dependencies within the image. This approach allows ViT to focus on relevant parts of the image, adapting to a wide range of tasks without being constrained by the locality bias of CNNs.

For each network, assign a distinct variable name, namely, *model_densenet_ex3* for the DenseNet model, *mode_resnet_ex3* for the ResNet model, and *model_vit_ex3* for the ViT model. This approach will allow you to compare and contrast the performance of traditional CNN architectures with the more recent transformer-based approach in image classification tasks.

You may use the default hyperparameters for each model. You should ensure that each model is appropriately adapted to handle the input data from the MedMNIST dataset. This entails adjusting spatial_dims, n_input_channels, and num_classes for resnet101, spatial_dims, in_channels, and out_channels for DenseNet121, and spatial_dims, in_channels, img_size, patch_size, and num_classes for ViT. The classification flag for ViT should be set to True to have a match between the dimensions of the output from your model (predictions) and the target labels.

Note: It's advisable to transfer each network to the designated computing device right before starting its training. Additionally, consider utilizing the CHPC (or other cloud resources) for training these models, as it can significantly enhance the training efficiency.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[4] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.