chaimeleon_open_challenges_rcjh - Prostate Risk Classification

Algorithm: ProstateCombinedResnet18PretrainedModel

Run inside of `./library/process_prostate.py` this algorithm is a deep learning algorithm combining a residual block inspired metadata fed set of layers with a Resnet18 based image feature extractor to predict 'low' and 'high' risk values separate classes. Several metadata to image combinations were experimented with but the one in this particular model converts the metadata into 3 float values which are used to 'enhance' the 3 layers fed into the image feature extractor.

An important consideration when preparing data for training models within this challenge was the number of layers necessary to maintain adequate information from the prostate scans. Some sort of slice level normalization was required due to the raw scans have different third dimensional layer counts and slice counts between 1 and 24 were evaluated for a set of from scratch models built. Based on these results, and the ease of integration with Resnet18 layers, three layers were chosen to be used for development of the 'release' models.

Training of this algorithm used binary cross entropy loss with bias towards high risk cases to attempt to balance performance across the classes adn made use of Adam optimization. Models were trained with various Resnet18 layers frozen - the best performing being fully unfrozen.

Resnet18 was chosen as the basis pretrained model due to its relatively small size and the difficulties faced with fully from scratch models generalizing well. With the limited amount of data provided for training augmentation was extensively made use of with random affine transformations applied to the image and random scaling +/- 15% applied to the numerical metadata values.

Evaluation was performed on augmented validation sets as test sets rather than truly 'unseen' images due to the limited number of cases provided for this phase of the challenge. It seemed more feasible to generate a 'test' set and reserve a greater number of images for training - though alternatives were not investigated and no elaborate logic was used to split cases into datasets that could've benefited training and evaluation both.

Team rcjh:

- Jordan Hendriksen

Methods - Classification Phase

Data Preparation

Both cancer types make use of a common parent class `ChaimeleonData`. This class encapsulates several methods used for reading and preparing data for use in training and evaluation.

- `process_data_directory`: Responsible for loading the raw data into the dataset. It uses the add_raw_case_data method to add the raw data to the dataset. This method looks for case folders in the specified data directory and adds the raw data for each case folder.

  - `add_raw_case_data`: Loads the image and metadata files for a given case and adds them to the dataset class. If the case_image_archive parameter is specified, the code loads the image data from the archive. If the release flag is set to True, it loads the data from the directory regardless of archive existence.
  - `add_raw_metadata` and `add_raw_ground_truth` methods simply open their respective json files and store the data within the file as dictionaries.
  - `load_image_file`: loads the image files (`.nii.gz` or `.mha`) into numpy arrays and applies a set of preprocessing (to be expanded upon below in the 'Prostate Risk Specific' and 'Lung Overall Survival Specific' sections below) to the image before adding them to the dataset class.
- `get_dataset_splits`: randomly splits the keys for the different cases within the dataset into train, validation, and test sets. A random seed is set - and can be defined at initialization of the dataset - while the percentage of cases targeted for each split is defined when the method is called.

After a dataset is initialized the first time it's images are stored as a compressed numpy array archive in 'npzc' format. During subsequent initializations if the release flag is not set to True the code loads the data from the archive if the archive still exists. This drastically cuts down on IO and computation if no parameters affecting the images loaded are being experimented with.

Prostate Risk Specific:

- `load_image_file` preprocessing for prostate data involves two main preprocessing steps:
  - aggregating the 3rd dimension slices into *N* 'average' slices (defined at initiat ion of the dataset). For example, with 24 starting slices and a target of 3 'average' slices the first 8 slices will be averaged, the middle 8 will be average, and the last 8 will be averaged, and these slices will be concatenated into one array for use in training/evaluation
  - scaling the values of the 'average' slices between 0 and 1
- Dataset preparation is completed in the `ProstateCancerDataset` class by doing the following for cases in the current metadata split:
  - normalizing metadata (one hot encoding categorical metadata) and loading both encoded categorical and numerical metadata into a vector. For this cancer there are no categorical metadata values provided and so the final vector is structured as [age, PSA]
  - normalizing ground truth values into a two class classification format (as opposed to binary classification that this could be viewed as)
  - Image and metadata augmentation strategies are defined
    - only during train and 'test' - which for this phase was an augmented version of the validation set

Models and Training

Prostate Risk

- Training:

  - BCE loss with bias balance loss towards 'high risk' cases
  - Tracked a variety of statistics and saved models based on both accuracy (number of validation samples with the correct classification) and score (weighted average of a few metrics surrounding classification and confidence)

    - Tracked variables required in order to recreate or *very nearly* recreate a model by using the same training parameters

- Models:

    - Various from scratch image models
    - Various from scratch metadata models
    - Various from scratch combined image metadata models
        - experimented with a variety of average image slices from a singular slice to 24 slices
    - Image based model from Resnet18 base
    - Combined image and metadata models from Resnet18 base (all with 3 average slices)
        - `ProstateCombinedResnet18PretrainedModel` (no resnet layers frozen) - two versions submitted to validation set with one version submitted to classification test set
        - `ProstateCombinedResnet18PretrainedModel_V2_1_Grid` (resnet layers 2 and 3 frozen) - one versions submitted to validation set

Evaluation

The actual evaluations were run inside a set of jupyter notebooks - `/notebooks/evaluate_prostate_models_V2.ipynb` and `/notebooks/evaluate_lung_models.ipynb`. Evaluations were performed by generating scores for each model on three datasets - the validation set used during training and two test sets.

In this phase due to a small number of cases in each cancers dataset the test sets were generated by applying various strengths of translation, rotation, skew, and scaling augmentations to the validation set to create images that the model 'had not' directly seen before. There are flaws in this system but it's the one that was chosen for the classification phase.

After generating scores for each model on each of the three datasets the scores were ranked and an average rank was calculated. Models were ordered by average rank and their individual scores against the different sets were compared to one another to try to determine which model might perform the best on the Grand Challenge datasets. I.e. a model with the best average rank because it did best on the validation set but faired more poorly on the test sets compared to the second or third average ranked model might not be the best choice as it seems to generalize more poorly.