**Transfer learning for covid-19 diagnosis**

Covid-19 virus resulted in the worldwide pandemic because of its infectivity and millions of people have died because of this. This virus can be diagnosed in several ways. For example, reverse transcription-polymerase chain reaction) tests, antibody tests, and CT scans. The CT scan may involve the application of artificial intelligence. I am going to use the frame of tensor flow for this implementation. I also use the convolutional neural networks built from scratch to classify the CT images and use transfer learning using resnet-18 to classify COVID-19 positive and negative lung CT scans. Scratch is defining a neural network by doing the convolution, max pool, and full connection. After that, modify the code which defines the network to get feature maps of a layer.

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from the related task that has already been learned. The reason to use transfer learning is that sometimes the dataset is too small, and we need to use the base model trained on a larger set of data.

The objective of this project is to develop a 2d CNN for Covid-19 diagnosis using CT images of lungs. the left image is the CT image of healthy people, and the right image is the CT image of the covid-19 patient. The indicator of COVID-19 positive is the ground class opacity at both lobes.

I may then use artificial intelligence to find this indicator. 图片包含 杯子, 动物, 桌子, 瓶子

描述已自动生成 盘子上放了料理

中度可信度描述已自动生成

Data:

* 1252 covid positive CT images
* 1230 covid negative CT images
* A test CSV file with column names of filename and label
* A train CSV file with column names of filename and label
* A Validation CSV file with column names of filename and label.

Preprocess:

* read & label the image
* Resize
* Split data
* Convert image to PyTorch tensors (Load RNG, add channel, to tensor, scale intensity)

Model development:

transfer learning

* Use the resnet-18 as pretrained model
* Download the model weights
* Freeze weights
* Create a sequence of trainable layers to replace the classification layer
* Tune the learning rate

Scratch

* Define a basic model with three convolutional layers
* Do the max pooling after each layer
* Draw the accuracy versus epoch diagram
* Add data augmentation
* Repeat the first 2 steps until overfitting fixed
* Generate report of accuracy

After finishing the transfer learning and the scratch, I will compare the F1 score and the recall of these two approaches. The transfer learning is supposed to have a higher F1 score because the number of samples is quite small in this case. And the improvement from the base model should be more accurate if the patterns of the data are similar.