Capstone Proposal – Exploring pneumonia xrays

Domain Background

As the covid-19 ravages throughout the world, scientists are racing to understand the characteristics of this new virus. The pandemic is gradually being controlled by the high-income countries and now hitting the small to middle income nations, which according to the world bank could push 71 million people into extreme poverty in 2020 under the baseline scenario and 100 million under the downside scenario. Since tomography is not widely available, x-rays help doctors understand the extent of the damage of the lungs from their patients. An AI platform that suggests the doctor the diagnostic of their patient's lungs could accelerate and improve the speed of attention at hospitals.

In academia, deep learning is starting to be applied more widely as a diagnostic aid (Medical Image Computing and Computer Assisted Intervention — MICCAI 2019). The task of detecting patterns is lengthy and costly for healthcare workers, both in western medicine as well as Chinese complementary medicine. In this last one, signs in the body –such as tongue or heart rate-- are well documented for detecting deficiencies and syndromes in a person. Therefore, I believe that in the future, computer vision and signal processing AI will be a key player in medicine in the whole world as more data is available and models become better.

Problem Statement

The problem is a classification for four classes. This model classifies between normal, and three types of pneumonia: bacteria, virus (non-covid), and covid-19.

Datasets and inputs

The dataset consists in 7 types of x-rays. Stress-Smoking, streptococcus and SARS are discarded because there is not enough data. Also, pneumonia from bacteria has 2.772 images, so the model is unbalanced with respect to the other classes. The information can be downloaded from https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset .

	Label	Label_1_Virus_category	Label_2_Virus_category	Image_Count
0	Normal			1576
1	Pnemonia	Stress-Smoking	ARDS	2
2	Pnemonia	Virus		1493
3	Pnemonia	Virus	COVID-19	58
4	Pnemonia	Virus	SARS	4
5	Pnemonia	bacteria		2772
6	Pnemonia	bacteria	Streptococcus	5

Benchmark model

The benchmark model to compare to is the flower classification from the transfer-learning class from Udacity's deep learning with Pytorch course. This is a VGG model, a CNN whose last layer is trained to classify flowers.

The accuracy for each of the seven types of flowers is the following:

```
Test Accuracy of daisy: 81% (75/92)
Test Accuracy of dandelion: 89% (118/132)
Test Accuracy of roses: 70% (64/91)
Test Accuracy of sunflowers: 64% (65/101)
Test Accuracy of tulips: 65% (81/124)

Test Accuracy (Overall): 74% (403/540)
```

When it is used for the lung images, at least the same overall test accuracy is expected as a benchmark.

Evaluation metrics

The percentage of precision for each class is tracked. That is, how many of the pictures inside a class were classified correctly.

Project design

First, since the covid-19 x-rays are just 58 pictures, the images are augmented to an approximately equal number to the other classes. Stress-Smoking, streptococcus and SARS are discarded because there is not enough data. In the augmentation process, the pictures are cropped to the size appropriate for the algorithm (224x224). They are not mirror-ed (flipped) because the chest-rays are all in the same perspective. However, the pictures are transformed with other methods to increase their number.

The strategy is to use AWS SageMaker to create a CNN model that can use GPUs during training and can be deployed if needed. A third-party application (Cyberduck) is used to upload the files (>1Gb) without interruption to the S3 file storage system.

Then, the code trains the last layers of the pretrained CNN model. It is key to use an already trained model because they are trained for weeks at a time in multiples GPUs. Therefore, the project uses a pretrained VGG16 model as a starting point. The code also takes advantage of the AWS SageMaker SDK for python to instantiate de models and deploying them.