Capstone Project

Machine Learning Engineer
Nanodegree
Meskhidze Konstantin
October 6, 2018

Definition

For the final project I've decided to join to one of competitions on Kaggle.com.

Kaggle.com launched a competition RSNA Pneumonia Detection Challenge. In this competition kagglers need to build a model to detect a visual signal pneumonia disease in a medical images of a patient's chest and locate it. Radiological Society of North America (RSNA®) and National Institutes of Health Clinical Center provide this competition with datasets. The society hopes this competition would help the diagnosis of pneumonia because the disease kills 15% of patients under 5 years old internationally.

Data Description

There are the following files in a dataset:

- **stage_1_train_images.zip** and **stage_1_test_images.zip** – training and test images.
- **stage_1_train.csv –** training labels.
- **stage_1_sample_submission.csv** - provides the IDs for the test set, as well as a sample of what your submission should look like.
- **stage_1_detailed_class_info.csv** - contains detailed information about the positive and negative classes in the training set, and may be used to build more nuanced models.

Data fields

- **patientId** _- A patientId. Each patientId corresponds to a unique image.
- **x**_ - the upper-left x coordinate of the bounding box.
- **y**_ - the upper-left y coordinate of the bounding box.

- **width_** - the width of the bounding box.
- **height_** - the height of the bounding box.
- **Target_** - the binary Target, indicating whether this sample has evidence of pneumonia.

All files can be downloaded from https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data

Solution statement

The accuracy will be evaluated using a ROC curve.

https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

In the project I will be following the next process:

1. Data overview.
    - Example image visualizing.
    - Labels and data analysis.
    - Drawing boxes in images with the disease.
2. Constructing a simple CNN with segmentation using Keras.
    - Resample images (reduce image dimension)
    - Choosing network architecture.
    - Train and test the CNN.
    - ROC curve accuracy
3. Using transfer learning to get better accuracy.
    - Trying different CNN architectures like : VGG16, RESNET52, Xception, Mask R-CNN, AlexNet ect.
4. Comparing the results and load the best on Kaggle.
5. Making a conclusion.