Machine Learning Engineer Nanodegree

Capstone Proposal

Beáta Hargitay
October 15th, 2018

Domain Background

Nowadays the usage of Clinical Decision Support Systems (CDSS) are spreading like a virus and being used more and more in healthcare institutes. But what is a CDSS? It is a system that helps in clinical decision-making and diagnosing for health professional with grounded data analysis based on other patients with similar symptoms. Using CDSS not only can help to improve the quality of healthcare but can also drastically reduce costs. These systems' history takes back until the early 1970s and now is major topic in Artificial Intelligence.

My personal motivation for investigating this domain is that I think CDSS is more than interesting and has a lot of undiscovered subdomains that would be great for the entire humanity to have. Maybe one day, Clinical Decision Support Systems are going to be advanced enough to detect cancer in an early stage, saving millions of lives.

Problem Statement

My project's aim is to build a base for a CDSS (a computer program for example) for radiology assistants to help diagnose Pneumonia easier, quicker and more accurately from chest X-Ray images. My solution is going to be based on a deep learning with the usage of transfer learning.

Datasets and Inputs

The dataset used for this project were downloaded from Kaggle. It contains 5 863 JPG files organized into 3 folders (train, test, val for training, testing and validation respectively) which all have 2 subfolders for each image category (Pneumonia/Normal).

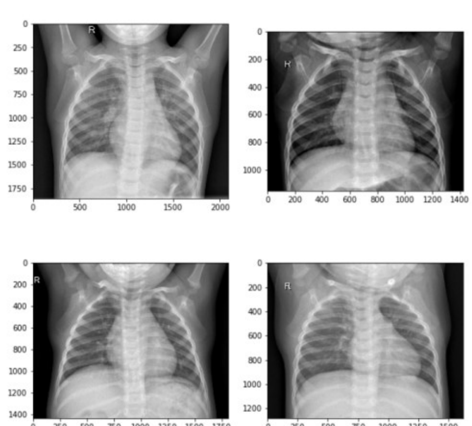

Figure 1. Examples on X-ray images with no pneumonia

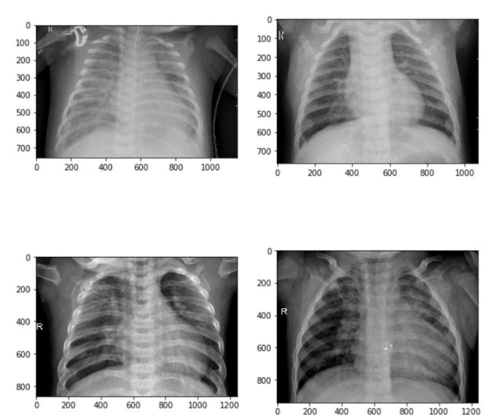

Figure 2. Examples of X-ray images with Pneumonia

“Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children’s Medical Center. Guangzhou. All chest X-ray imaging was performed as part of patients’ routine clinical care.

For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert.” (Mooney, 2018)

Solution Statement

My solution for a medical diagnostic tool that could recognize Pneumonia from chest X-ray images is going be a Deep Learning Model built with Tensorflow and Keras hand in hand. Since the dataset I am going to use has a really small number of images for training (only 5 216) it would make no sense to create my own model from scratch. This is why I am going to use exclusively transfer learning for solving this problem.

My plan includes using the following architectures:

- InceptionResNetV2
- Xception
- InceptionV3

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
NASNetLarge	343 MB	0.825	0.960	88.949.818	-
InceptionResNetV2	215 MB	0.803	0.953	55.873.736	572
Xception	88 MB	0.790	0.945	22.910.480	126
InceptionV3	92 MB	0.779	0.937	23.851.784	159
DenseNet201	80 MB	0.773	0.936	20.242.984	201

Figure 3. Performance of the top 5 implemented models (based on their accuracy on their Top-1 Accuracy[1].) in Keras in descending orders Source: https://keras.io/applications/

I chose these particular models because they have the best Top-1 and Top-5 accuracy on the ImageNet’s validation dataset among the architectures that are not using large images to train (NASNetLarge uses 331 x 331 pictures).

[1] *The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset.*

Benchmark Model

As a benchmark model I am going to use the top-1 accuracy for the InceptionResNetV2 on the ImageNet's validation dataset, which is 80%. This means that my goal is to reach at least 80% of accuracy on the validation set of the dataset I am using for detecting Pneumonia.

Apart from that, I should mention that the goodness of the model's performance is depending more on the accuracy on the test set. This is a whole other set that in this case is much bigger than the validation set and also is a completely new set for the model to score. In my capstone project I am also going to show the performance of the model on the test set.

Evaluation Metrics

The most important metric I am going to use to evaluate my model is going to be accuracy. This value can be a little misleading as it is showing how good can the model predict the different labels.

$$Accuracy = \frac{(True\ Positive + True\ Negative)}{Total\ Number\ of\ predicitons} * 100$$

At this problem domain, it would be more important to know how good can the model predict if somebody has Pneumonia. This is why I am going to check the precision as well.

$$Precison = \frac{True\ positive}{True\ positive + False\ positive}$$

As I mentioned above, I am going to evaluate both the validation and the test set with these metrics.

Project Design

The theoretical workflow that I would like to follow is presented below on *figure 4*.

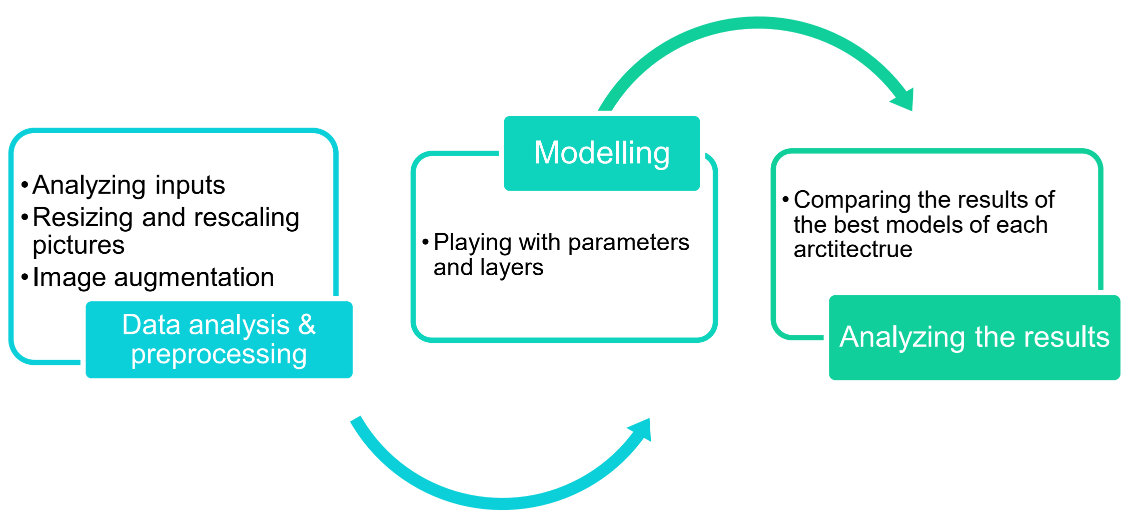

Figure 4. Theoretical Workflow of the development of my model for the Capstone project

Data Analysis & Preprocessing

My intention is start with a little data analysis. I would like to examine the structure of the dataset, see what kind of images it contains, how large and clear are those data in there. After that I would step to data preprocessing, which includes the following steps in this order: resizing, rescaling and augmenting the images.

First of all, to train our model to identify something, we have to use a given size (fixed width and height) for the images that we use as input, i.e. we have to resize them. Secondly, rescaling our images is also important because standardizing the values of the image pixels (from 1 to 255 to 0 to 1) helps our network to perform better. At last, data augmentation is also a really important step in CNN projects for two major reasons: first, if you have a small amount of data you can make a bigger one with this technique and second, it helps to avoid overfitting.

Modeling

The next step would be to create some benchmark models from each architecture and then playing a little with all the parameters and layers possible until I find the model per each architecture, which gives the best accuracy. My goal would be to attain at least 80% accuracy on the validation set.

Analyzing the results

When I am finished I would like to make some analysis to compare the result of the different architectures with each other to find the best model. The best model is going to be the one that has the best accuracy on the test set.

Bibliography

Mooney, P. (2018, March 24). *Chest X-Ray Images (Pneumonia)*. Retrieved October 10, 2018, from Kaggle: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia/data

Wikipedia. (n.d.). *Clinical Decision Support System*. Retrieved 10 13, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Clinical_decision_support_system