Machine Learning Engineer Nanodegree

Capstone Proposal

Praveen Kumar
March 13, 2018

Proposal

Domain Background

Drug discovery and research is very essential for improving quality of life for everyone on the planet. Historically, the number of drugs that can be developed for a billion dollars of R&D spending has been decreasing exponentially over the decades, as described by Eroom's law. For example, in the 1950s, one could develop 30 new drugs for a cost that wouldn't even find a single drug today. One major bottle neck in drug development is identifying cell nuclei in microscopy images. Researchers often test thousands of compounds and their variants on cells to find the one that might be a good drug. Researchers prepare batches of cells and apply a different compound to each batch and take microscopy images. Then they identify which batch responded well by looking to find a batch in which cells became healthier.

The first step in identifying the cell characteristics in the images that the researcher is studying is identifying the nuclei. Since nuclei of cells are at the center of the cell body, a stain that reveals nuclei gives an image of the cell batch in which each cell can be distinguished without much overlap on others. There are hand coded algorithms that do a decent job in identifying cell nuclei when the nuclei are well rounded in shape, regular and not very crowded. Sometime nuclei have very unusual shapes. Sometimes cells can be hard to distinguish in a tissue sample. This can sometimes result in the researcher manually hand label cells by looking at thousands of microscopy images. This is highly time consuming, and time that could have been put to better use.

There have been efforts in developing machine learning models to do this task in recent years. Like the work by Polina Gross and team (*https://arxiv.org/ftp/arxiv/papers/1512/1512.04370.pdf*) to develop software that can characterize and quantify cell nuclei in immunofluorescent tissue images. Works like these have been done by subject expert biologists to target specific type of imaging technique and or cell batch type, preparation, setup, etc. Work done in processing similar (fluorescent) images done by Anton Jackson-Smith at Stanford (*http://cs231n.stanford.edu/reports/2016/pdfs/326_Report.pdf*) shows promise and potential for the approach arrived at by the work to be improved towards a more generalizable model that can handle a wider spectrum of imaging techniques, tissue types, experimental setup, etc.

Problem Statement

The goal of the project create a robust deep learning model that can understand what nuclei look like and identify them in microscopy images regardless of cell type, tissue sample, stain used, scale, illumination, microscopes, resolution, experimental setup, etc. without human (biologist) intervention. The machine learning problem to solve is as follows: A multi loss, regression (image mask generation) + classification (nucleus identification/classification between nucleus and background) problem that takes images (.png) of cell samples and generate image masks representing each nucleus present in the input image.

Datasets and Inputs

I have chosen this project from a Kaggle competition *(https://www.kaggle.com/c/data-science-bowl-2018*). The dataset that will be used in the project is provided on kaggle (*https://www.kaggle.com/c/data-science-bowl-2018/data*) by the organizers of this competition. The dataset is microscopy images of cell batches from actual drug discovery experiments provided by the team at Broad Institute of Harvard and MIT, who are the organizers of this competition.

The dataset has separate training and test examples. The training set will be split and used for fold-cross validation to tune hyper parameters. Training data consists of microscopy images of cell bathes and correspondingly, a set of image masks representing each nucleus in the image. The input images are all 3-channel, RGB images with pixel values ranging 0-255, encoded as .png files. These images vary in width, height from around 200 to above 500 pixels and consist of various cell types, imaging techniques, illumination, resolution, scale/zoom, setup, etc. All in all, the data set contains 25,000 mask-labeled nuclei.

The test data set provided has microscopy images only, without the target masks. Calculation and verification of accuracy/score of the model output on the test dataset will happen on Kaggle's leaderboard (*https://www.kaggle.com/c/data-science-bowl-2018/leaderboard*) once the predictions from the trained model is uploaded to the Leaderboard. Test set ground truth data is not made public to discourage hand labeling so test set accuracy validation has to be done on Kaggle leaderboard.

Solution Statement

The solution to the problem that this project addresses will be as follows: A deep convolutional neural network trained on the dataset of microscopy images and masks that takes a new microscopy image as input and produce image mask for each nucleus present in the input image. Further, for the purpose of competition submission on kaggle, the predicted mask pixels have to be run length encoded in a CSV file as described here: *https://www.kaggle.com/c/data-science-bowl-2018#evaluation* .

Benchmark Model

For the purposes of this project, a purely classical image processing solution will be used to bench mark against. In particular, a classical solution to the problem developed by Gabor Vecsei (*https://www.kaggle.com/gaborvecsei*) will be used. The solution is posted on the Kaggle and can be found at this link: *https://www.kaggle.com/gaborvecsei/basic-pure-computer-vision-segmentation-lb-0-229* .

This model will be used to generate predictions on the same test data that will be used for evaluating the model that will be developed in this project. This will provide an objective ground to make a fair comparison against bench mark.

Evaluation Metrics

Excerpt from https://www.kaggle.com/c/data-science-bowl-2018#evaluation:

This competition is evaluated on the mean average precision at different intersection over union (IoU) thresholds. The IoU of a proposed set of object pixels and a set of true object pixels is calculated as:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

The metric sweeps over a range of IoU thresholds, at each point calculating an average precision value. The threshold values range from 0.5 to 0.95 with a step size of 0.05: (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95). In other words, at a threshold of 0.5, a predicted object is considered a "hit" if its intersection over union with a ground truth object is greater than 0.5. At each threshold value *t*, a precision value is calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP) resulting from comparing the predicted object to all ground truth objects:

$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

A true positive is counted when a single predicted object matches a ground truth object with an IoU above the threshold. A false positive indicates a predicted object had no associated ground truth object. A false negative indicates a ground truth object had no associated predicted object. The average precision of a single image is then calculated as the mean of the above precision values at each IoU threshold:

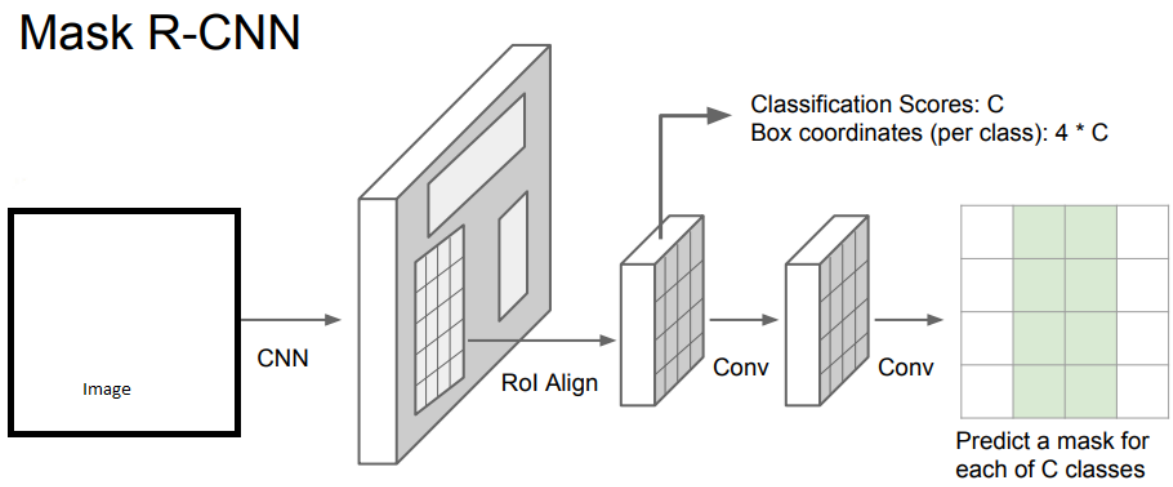
$$\frac{1}{|thresholds|} \sum_{t} \frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

Lastly, the score returned by the competition metric is the mean taken over the individual average precisions of each image in the test dataset.

Project Design

I have been deep learning libraries other than Tensorflow. I have found PyTorch to be exceptionally powerful, flexible and easy to use, especially for dynamic network modification and better access to network state and parameters. So, I have decided to use PyTorch for the capstone project.

At the time of writing this proposal, I am thinking of using of a Mask-RCNN network (*https://arxiv.org/pdf/1703.06870.pdf*) for generating segmentation masks.



Proposed work flow:

- Use lower layers of a pre-trained CNN like RESNET for transfer learning and feature generation
- Construct Mask R-CNN network with conv. nets to predict segmentation masks
- At this moment, my key insight for using the training data set well is to have the network operate on small crops of the input image and scan across the image in a few steps and predict the segmentation masks
- If the network can scan across the input image in crops (fairly big but fixed size crops), it will help a lot in making the solution robust to changes in input image's width and height, which is a requirement for the solution on succeed
- The loss function can be constructed as an intersection over union between predicted masks and ground truth masks.
- Run cross validation to optimally set key hyper-parameters
- Keep a small subset of training data for validating model accuracy during local development
- Train network over entire training data set, run predictions on test dataset, run encode mask images, post predictions on Kaggle leaderboard to find out test set accuracy