Machine Learning Engineer Nanodegree

Capstone Proposal

Samuel A. Rodriguez L.

July 19th , 2019

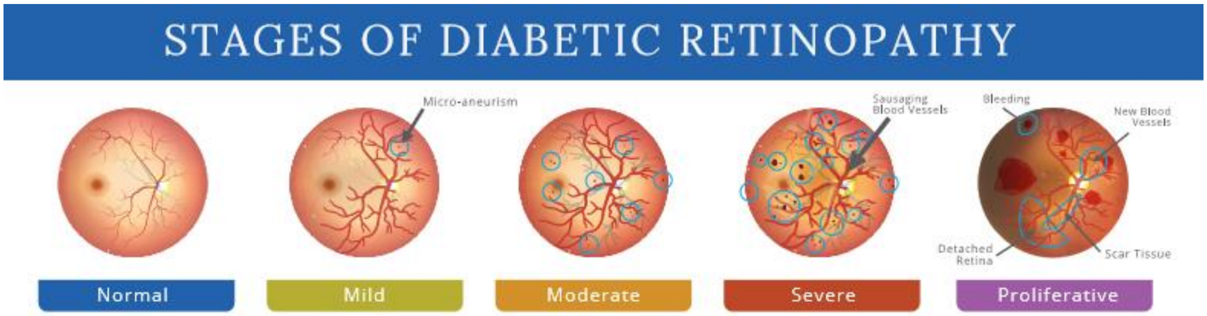
Diabetic Retinopathy Detection (Kaggle[1] Competition)

Domain Background

Millions of people suffer from *diabetic retinopathy*[1], the leading cause of blindness among working aged adults. Generally, this condition is treatable, but it has to be caught early enough. In developing countries like India, the ophthalmologist-patient ratio is at a dismal *1:10,000*[2]. Therefore, an automated tool for grading the severity of diabetic retinopathy would be very useful for accelerating detection and treatment, especially in populations living in rural areas.

Recently, there have been a number of attempts to utilize deep learning to diagnose DR and automatically grade diabetic retinopathy. This includes a previous *competition*[3] and *work*[4] by Google. Even one deep-learning based system is FDA *approved*[5].

Clearly, this dataset and deep learning problem is quite important.



Problem Statement

This is a multi-class classification problem. The goal is to create an algorithm that is able to specify the severity of the disease given retina images taken using *fundus photography*[3] as input and producing as output 1 of 5 possible categories of severity going from 0 to 4 (0 means not having the condition).

Datasets and Inputs

The data set consists of a large set of retina images taken using fundus photography under a variety of imaging conditions.

A clinician has rated each image for the severity of diabetic retinopathy on a scale of 0 to 4:

0 - No DR

1 - Mild

2 - Moderate

3 - Severe

4 - Proliferative DR

Images may contain artifacts, be out of focus, underexposed, or overexposed. The images were gathered from multiple clinics using a variety of cameras over an extended period.

Solution Statement

Given the characteristics of the problem applying a convolutional neural network as image recognition technique seems to be the most likely approach to find a satisfactory solution. Two of the most successful architectures in recent years have been Microsoft's *ResNet*[4] and Google's *Inception*[5] both of which seem to be reasonable options to implement. Furthermore, to take advantage of pre-trained models I'll use transfer learning and then fine tune all the weights.

Benchmark Model

For this project I'll use as baseline a ResNet50 model pre-trained on ImageNet + Global Average Pooling layer + Dense Layer with 2048 nodes:

```
def create_model(input_shape, n_out):
    input_tensor = Input(shape=input_shape)
    base_model = applications.ResNet50(weights=None,
                        include_top=False,
                        input_tensor=input_tensor)
    base_model.load_weights('../input/resnet50/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5')

    x = GlobalAveragePooling2D()(base_model.output)
    x = Dropout(0.5)(x)
    x = Dense(2048, activation='relu')(x)
    x = Dropout(0.5)(x)
    final_output = Dense(n_out, activation='softmax', name='final_output')(x)
    model = Model(input_tensor, final_output)

    return model
```

Evaluation Metrics

The metric will be the quadratic weighted kappa, which measures the agreement between two ratings. This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters). In the event that there is less agreement between the raters than expected by chance, this metric may go below 0. The quadratic weighted kappa is calculated between the scores assigned by the human rater and the predicted scores.

Images have five possible ratings, 0,1,2,3,4. Each image is characterized by a tuple (e,e), which corresponds to its scores by Rater A (human) and Rater B (predicted). The quadratic weighted kappa is calculated as follows. First, an N x N histogram matrix O is constructed, such that O corresponds to the number of images that received a rating i by A and a rating j by B. An N-by-N matrix of weights, w, is calculated based on the difference between raters' scores:

An N-by-N histogram matrix of expected ratings, E, is calculated, assuming that there is no correlation between rating scores. This is calculated as the outer product between each rater's histogram vector of ratings, normalized such that E and O have the same sum.

Project Design

One the first steps when working with data science projects is doing exploratory data analysis. In this case, I will start by looking at a sample of the images in the dataset to get an idea of the type of images I am working with and determine if some type of preprocessing might be beneficial, like resizing or scaling. Then statistical analysis might be done to understand the distribution of the labels and know if we are dealing with an imbalanced dataset (one category is considerably more common than others) or to look at the 'mean image' to get some insight into the underlying structure of the dataset.

After Understanding the data set better, I will proceed to define the model architecture. As mentioned before ResNet and Inception are among the best performers, so I will use transfer learning with them. I will start by testing the performance of the basic versions of both models 'out of the box' and also doing some small modifications to each in order to get a sense of which one might work better for this problem.

Once I have decided the model, I will test different versions (e.g. ResNet50, ResNet101, Inception V1-4) and pick the one with best results (It might be assumed that newer versions will always perform better than older ones but this is not necessarily always the case). The Next step would be start playing with more complex architectures on top of the base model, which will involve including, but not limited to, trying different combinations of Dropout, Pooling and Batch Normalization Layers with different layer sizes and activation functions. Lastly, hyper-parameter optimization will be my focus, this will vary importantly depending on the architecture and other aspects decided but surely trying different batch sizes, learning rates, number of epochs and other parameters will be necessary.

After achieving a satisfactory performance (or perishing in the attempt), I will submit the results and the final score will be calculated against the private dataset provided by Kaggle.

References

1. https://nei.nih.gov/health/diabetic/retinopathy
2. https://www.sankaranethralaya.org/a-step-towards-combating-blindness-in-rural-areas.html
3. https://kaggle.com/c/diabetic-retinopathy-detection
4. https://ai.googleblog.com/2016/11/deep-learning-for-detection-of-diabetic.html
5. https://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm604357.html
6. https://en.wikipedia.org/wiki/Fundus_photography
7. https://arxiv.org/abs/1512.03385
8. https://arxiv.org/abs/1409.4842