Machine Learning Engineer Nanodegree

Capstone Proposal

Alejandro Trujillo

May 7th, 2018

Proposal

Domain Background

During my previous job, at a medical imaging company, I was exposed to the features and potential of computer assisted analysis and diagnosis in the medical field. The suite of products that I helped built gathered DICOM images from medical imaging devices such as CT scanners and MRI machines. These images can be used to identify patterns that match a specific disease. A machine learning algorithm could help physicians diagnose their patients by classifying imaging patterns to diseases.

I believe that there is a lot of untapped potential for growth and improvement in the medical diagnosis field by leveraging machine learning methods.

Eventually, the company was acquired for 1 billion dollars placing strategic value in the data that was acquired.

For my project, I will analyse patterns in Optical Coherence Tomography (OCT) images encoded in jpeg format.

Optical coherence tomography (OCT) is a non-invasive imaging test. OCT uses light waves to take cross-section pictures of your retina.

With OCT, ophthalmologists can see each of the retina’s distinctive layers. This allows them to map and measure their thickness. These measurements help with diagnosis. They also provide treatment guidance for glaucoma and diseases of the retina. These retinal diseases include age-related macular degeneration (AMD) and diabetic eye disease.

Problem Statement

Approximately 30 million OCT scans are performed each year, and the analysis and interpretation of these images takes up a significant amount of time (Swanson and Fujimoto, 2017).

Well trained ophthalmologists can do an outstanding job at classifying any disease depicted in a OCT scan. However, long hours at work and eye fatigue can hinder their diagnosis. Even one misdiagnosed case can hinder the relationship with their patients and their reputation. Here I present an approach to aid ophthalmologists in their diagnosis and further enhance their relationship with their patients.

My approach uses as inputs 84,495 labeled images. Each image has one of 4 possible labels NORMAL,CNV,DME or DRUSEN. Each label represents the diagnosis assigned to the image. This diagnosis is based on the tiered grading system described in the datasets and inputs section.

These inputs are then transformed into 4D tensors. These tensors are used by a deep learning algorithm to learn the patterns within the images.

Finally, once the algorithm has been trained, it's used to provide an estimate of how likely each of the 4 diagnosis is depicted on the image. This estimate can be used by ophthalmologists to further strengthen or change their diagnosis.

In other words, as described by Tom M. Mitchell (1997):

- **Task (T)**: Classify an OCT scan with the probability of having each of the 4 diagnosis.
- **Experience (E)**: A set of training and validation images where each is labeled with its correct diagnosis.
- **Performance (P)**: Classification accuracy, the number of diagnosis predicted correctly out of all diagnosis considered as a percentage.

Computer assisted diagnosis can enhance the identification of retinal diseases quicker and more consistently. This allows a physician to consult a Machine Learning application about any given case before giving a diagnosis. Enhancing a medical diagnosis by an automated machine learning algorithm will provide a greater peace of mind to the physicians and patients due to the accuracy and consistency of a machine learning algorithm.

Datasets and Inputs

The dataset is organized into 3 folders:

1. Train
2. Test
3. Validation

It contains subfolders for each of the 4 categories (diagnosis):

1. NORMAL
2. CNV
3. DME
4. DRUSEN

There are 84,495 X-Ray images in total. Each image is encoded in the JPEG format and each image will be resized to 224×224 pixels. Each image will later be converted into a 4D tensor with dimensions (nb_samples 1, rows 224, columns 224, channels 1).

Optical coherence tomography (OCT) images (Spectralis OCT, Heidelberg Engineering, Germany) were selected from retrospective cohorts of adult patients from the Shiley Eye Institute of the University of California San Diego, the California Retinal Research Foundation, Medical Center Ophthalmology Associates, the Shanghai First People's Hospital, and Beijing Tongren Eye Center between July 1, 2013 and March 1, 2017.

Before training, each image went through a tiered grading system consisting of multiple layers of trained graders of increasing expertise for verification and correction of image labels.

Each image imported into the database started with a label matching the most recent diagnosis of the patient.

The first tier of graders consisted of undergraduate and medical students who had taken and passed an OCT interpretation course review. This first tier of graders conducted initial quality control and excluded OCT images containing severe artifacts or significant image resolution reductions.

The second tier of graders consisted of four ophthalmologists who independently graded each image that had passed the first tier. The presence or absence of choroidal neovascularization (active or in the form of subretinal fibrosis), macular edema, drusen, and other pathologies visible on the OCT scan were recorded.

Finally, a third tier of two senior independent retinal specialists, each with over 20 years of clinical retina experience, verified the true labels for each image.

Solution Statement

My solution will leverage the power of deep learning and convolutional neural networks (CNN) to find patterns within the images of OCT scans. The optimized solution utilizes a transfer learning model with the goal of achieving at least 70% accuracy.

The pretrain models are listed below and were taken from the Dog Breed classification project:

- VGG-19 bottleneck features
- ResNet-50 bottleneck features
- Inception bottleneck features
- Xception bottleneck features

The technologies to build the benchmark and the optimized model are Keras to build the CNN architecture, using a TensorFlow engine in the background, and Python 3.

CNNs are very powerful in analysing images and finding the patterns within. Finding the patterns is possible by utilizing filters and layers in the CNN architecture. Each layer has the purpose of identifying more specific patterns than its predecessor.

At the last layer I plan to use a dense layer with 4 nodes to get the final probabilities for each diagnosis (NORMAL,CNV,DME,DRUSEN) in our dataset.

I will use a softmax activation function, because this is a multi-class classification problem. This ensures that the network outputs an estimate for the probability that each potential diagnosis is depicted in the image.

Benchmark Model

For the benchmark model, I plan to create a basic CNN architecture from scratch. This model’s goal is to achieve more than 1% accuracy. The benchmark architecture will be as follows.

1. Convolutional 2D layer with kernel size 2 and ReLU activation function.

2. Max pooling 2D layer with pool size 2.
3. Convolutional 2D layer with kernel size 2 and ReLU activation function.
4. Max pooling 2D layer with pool size 2.
5. Convolutional 2D layer with kernel size 2 and ReLU activation function.
6. Max pooling 2D layer with pool size 2.
7. Dropout layer with 0.3 probability.
8. Flatten
9. Dense with 4 nodes and softmax activation function.

This benchmark model is used to compare the results of the optimized transfer learning model.

Evaluation Metrics

The dataset is divided in training, validation and testing to prevent overfitting.

Additionally the following metrics are used to quantify the performance of the benchmark and optimized algorithms.

- **Training time:** Measures the time that the algorithm takes to learn the patterns within the training set.
- **Accuracy:** Indicates the degree to which the resulted output matches the correct label.
- **Optimizer:** RMSprop divides the learning rate by an exponentially decaying average of squared gradients.

Project Design

The goal is to create an optimized model that reduces training time and achieves over 70% accuracy using transfer learning and convolutional neural networks (CNN).

First I plan to load the dataset and get the total number of training, testing and validation images.

Then I plan to preprocess the data. First, I plan to convert the RGB images into grayscale images to reduce the training time. Then when using TensorFlow as backend, Keras CNNs require a 4D array (which is often referred to as a 4D tensor) as input, with shape (nb_samples, rows, columns, channels). To create the 4D tensor the data is preprocessed by loading the image and converting it into a 3D tensor then we convert the 3D tensor to 4D tensor and return a 4D tensor.

After preprocessing the data my next step will be to build the benchmark model and measure its accuracy. The goal of the benchmark model is to achieve more than 1% accuracy.

Then, I plan to build an optimized model using transfer learning by leveraging the pretrain models listed below (taken from the Dog Breed classification project):

- VGG-19 bottleneck features
- ResNet-50 bottleneck features
- Inception bottleneck features
- Xception bottleneck features

Each model's correctness will be measured using accuracy. At the end the model that achieves the best accuracy will be chosen.

The optimized model will be compiled using:

- Categorical cross-entropy for its loss function.
- RMSProp optimizer.
- Accuracy metric.

The optimized model will follow the same architecture as the benchmark model using the architecture described below:

1. Convolutional 2D layer with kernel size 2 and ReLU activation function.
2. Max pooling 2D layer with pool size 2.
3. Convolutional 2D layer with kernel size 2 and ReLU activation function.
4. Max pooling 2D layer with pool size 2.
5. Convolutional 2D layer with kernel size 2 and ReLU activation function.
6. Max pooling 2D layer with pool size 2.
7. Dropout layer with 0.3 probability.
8. Flatten
9. Dense with 4 nodes and softmax activation function.

The convolutional layers are only locally connected which should reduce training time compared to a fully connected layer.

The following measures are taken to prevent overfitting:

- Max pooling layers after each convolutional layer to reduce the dimensionality of the image.

- Dropout layer with a probability that any node can be taken off the network.

Finally, a dense layer with a softmax activation function is used to get the probability of any of the 4 diagnosis is depicted in the image.

References:

- *Turbert, D. (2018). What Is Optical Coherence Tomography?. [online] American Academy of Ophthalmology. Available at: https://www.aao.org/eye-health/treatments/what-is-optical-coherence-tomography [Accessed 9 May 2018].*

- *Kaggle.com. (2018). Retinal OCT Images (optical coherence tomography) | Kaggle. [online] Available at: https://www.kaggle.com/paultimothymooney/kermany2018/data [Accessed 9 May 2018].*

- *Sebastian Ruder. (2018). An overview of gradient descent optimization algorithms. [online] Available at: http://ruder.io/optimizing-gradient-descent/ [Accessed 12 May 2018].*

- *Mitchell, T. (1997). Machine Learning. New York: McGraw-Hill.*

- *https://data.mendeley.com/datasets/rscbjbr9sj/2*