Reflective Report

Sung Joon Park

July 24th 2020

Professor Jung Hyun Kim

DXP Lab

Reflective Report

Machine learning is such an interesting course of study that attracts many but, at the same time, yields frustration by its unfamiliarity. I also had a great interest but never had a chance to study for it. However, with great opportunity to be part of the DXP Lab of Korea University and to get a guidance from Professor Jung Hyun Kim, I was able to start a project regarding the field of AI/ML. At first, there was some concerns which topic I should instigate, since AI/ML was an extensive study where I had no profound knowledge of. During the consultation to decide the concept of the project with Professor Kim, he told me the dilemma of his tennis club. The tennis court used by his club members was located outdoors, and therefore, they always had to check the weather whenever they wanted to play tennis. However, the weather report alone made it difficult to discern the condition of the field, so it was inevitable for someone to go out and check it all the time regardless of the report. Taking a hint from his story, we reached to a conclusion that the unnecessary behavior of examining the court every time can be automated, and that became the cornerstone of my project.

The fundamental plan to automate the action was simple – create two datasets of 'CAN' and 'CANNOT,' build a structure that learns from datasets, determine the status of the court, and derive the result. To construct the datasets and collect data, first step was to install an IP camera that has a clear vision of the whole field. However, there was no power outlet to connect a camera near the court, so I had to set it up on the balcony of Professor Kim's house, which had a view of the whole court. I was able to access to camera view through the internet and collect images of both available and unavailable days to play tennis.

The important part was a structure to learn and classify image. With a lack of thorough study of AI/ML, I educated myself using Sung Kim's slides and videos which provided lectures about basic concept of deep learning from Linear Regression to RNN (Recurrent Neural Networks) and practical labs for application using Tensorflow. I focused on CNN

(Convolutional Neural Networks) to build a framework with two layers: convolution layer that breaks down an image into features and analyzes them independently and fully-connected layer (FC layer) that receives the result from a convolution layer to make final classification determination. First, I created two placeholders, X placeholder as to receive an image as a float array and Y as to label it either 'CAN' or 'CANNOT.' I reshaped imported image array into applicable size and shape for convolution layers. Then, I constructed two convolution layers, where the input is scanned by certain numbers of filter with a given size, replaced the negative values to zero from the scanned input using ReLU (Rectified Linear Unit) to improve effectiveness, and lastly, calculated the largest value in each patch of each feature using maximum pooling. The output from the final pooling was flattened before connecting to the fully-connected layer. Under the FC layer, flattened input went through its own backpropagation processes, deciding the most accurate weight value and determining the best-matching classification label.

Since the structure was built, the final phase was to bring the collected images and test if the final model was resulting with an acceptable accuracy. I divided each of two datasets of ‘CAN’ and ‘CANNOT’ into another two directories – one to train the model and the other, to test for accuracy. First, I imported images from ‘train’ directory and connected into the placeholder from the structure I created earlier so that the model will learn the features of images and educate itself whether they are available or not to play tennis. Additionally, I designed different structure that takes the images from ‘test’ directory, bring through the model that finished learning from images of ‘training’ directory, and calculate how precisely the images are classified. Moreover, I had added predicting framework that randomly selects an image from test directory and shows model’s calculated label along with actual label (either ‘CAN’ or ‘CANNOT’) for comparison. The model presented moderate prediction, an average of 73% accuracy.

Evaluating the accuracy of my model and my whole project, 73% efficiency seemed insufficient and was not fascinating enough to implement it on the tennis court right away. The deficiency of the model has been assumed for various reasons: First of all, it could have resulted due to the lack of data images. The IP camera used was only accessible through its own website that had runtime of 5 minutes. I had to collect captures one by one in every hour or two, and the action could not be automated. Therefore, it was somewhat impossible to obtain plentiful data that could have allowed the model to train more precisely. Also, during the time of the project, the fine weather caused difficulty to collect 'CANNOT' images, since the rainy day only lasted less than 5 days. Another presumed obstacle could be explained by the capture itself. The absence of power outlet on the court led to placing it on the 12th floor of an apartment building. Because of the distance from the camera and the scene, if it was tightly zoomed only to feature the field itself, the image cracked. Therefore, the captures had to include surroundings such as parking lots, roads and trees, which could have worked as drawbacks that disturbed the training process for the model to just focus on the target. One last reason was from compute speed. It was difficult to test different filter sizes or add another convolution layer because the architecture required high degree of memory allocation .More convolution layers could have been built and tested to enhance preciseness, if the memory was allowed.

Through the project, I have learned and felt both the necessity and the downside of machine learning solution. The inconvenience aspect was a complexity to form a dataset. The dataset had to be worked closely and precisely and also required great volume size to make accurate result. The needs for greater stability and accuracy was examined as well. Overfitting was another frustration, which shows decent fit with provided training data set, but brings unacceptable result with test data or in real usage. Data storage and compute speed could be other obstacles. However, there were more favorable aspects – its incredible potential was a model's independent adaptation. When fed with acceptable data, it learned from previous computations and produced reliable results. It used all features to find complex, fraudulent patterns, even with huge volume of data. As a person who wants to have a job and work in the pertaining field of study, the project was a valuable experience to learn about machine learning and practice it with such real-life example, even though it had some unsatisfactory accuracy and unfortunate hindrances. I look forward to see and be part of the impacts that the machine learning is about to bring to our lives.