Capstone 2 Proposal: Classification of Satellite Images

Overview

I am developing a deep learning model for classification of different types of land images with different cover types in satellite images.

Client/Business Case

Satellite images can be used for many applications such as, agriculture, city planning, natural resource management, environmental monitoring and disaster response. Since satellite images can cover large areas, the costs to manually categorize land uses within the imagery can be prohibitive. Deep learning has emerged as an important approach for automating land use classification over extensive areas. Applying deep learning to imagery classification can reduce costs when compared to using staff to manually delineate land cover classes.

Data

The Deep Sat (SAT-6) dataset available on Kaggle contains pre-labeled image tiles extracted from the National Agriculture Imagery Program (NAIP) dataset. The full NAIP dataset covers the entire US at 1-meter resolution over 4 bands: red, green, blue and near-IR. The SAT-6 dataset consists of 28x28 pixel non-overlapping tiles extracted from NAIP imagery over different regions of California.
The labels are one-hot encoded vectors for the following 6 land cover classes: barren land, trees, grassland, roads, water and buildings. Both training and test datasets are available, and the datasets have been randomized.

Approach

I will develop the imagery classification model using two methods:

- Develop a Convolutional Neural Network (CNN) from scratch.
- Use a pre-trained CNN (VGG16, Resnet50, etc.) and transfer learning.

The CNNs will be implemented in Python on Google Colab using the Keras interface to Tensorflow and the methods described in Deep Learning with Python (F. Chollet, 2018).

Plan:

- Develop a baseline model structure.
- Build training and validation image data generators in keras.
- Compile the model.

- Run model and plot the training and validation accuracy scores over each epoch. Evaluate score on the hold-out data set
- Manually calibrate the model input parameters (nodes/layers of the CNN, optimizer parameters, etc.)
- Evaluate whether image augmentation improves the model performance.
- In the case of overfitting, apply regularization techniques such as dropout layers, data augmentation.

Deliverables

The deliverables will be a report, Exploratory Data analysis, model development, and performance. I will compare the outcomes of the model built from scratch and the model built using transfer learning. All these deliverables will be uploaded in the GitHub