Machine Learning Ⅱ
Final Project-Airborne Dataset Proposal

04/03/2018

Qianwen Liu
Mark Barna
Thanh Nguyen

What problem did you select and why did you select it?

Since our team members are all interested in exploring image classification, we decided to build a network to classify different landscapes by using DeepSat AirBorne dataset. We think that this network would be useful for developing and updating maps, monitoring the environment, exploring and protecting natural resources, etc. This is a great chance for us not only to practice our analytical skills but also to contribute to the real world development.

What database/dataset will you use? Is it large enough to train a deep network?

We will use DeepSat AirBorne Dataset that is available on Kaggle. Images were extracted from the National Agriculture Imagery Program (NAIP) dataset. The NAIP dataset consists of a total of 330,000 scenes spanning the whole of the Continental United States (CONUS).

Our dataset, which contains a total of 810,000 images, is large enough to train a deep network. Each image in the dataset is 28X28 pixels and contains 4 bands - red, green, blue, near-infrared. There are 6 different categories, representing the six broad land covers, including barren land, trees, grassland, roads, buildings and water bodies, throughout the whole state of California. Our training and test sets have 324,000 and 81,000 entries, respectively.

What deep network will you use? Will it be a standard form of the network, or will you have to customize it?

We will use convolutional neural network because of the advantages it provides for image classification. We want to introduce pooling layers to decrease the size of the input as well as dropout layers to reduce overfitting so we mostly likely will customize our network.

What framework will you use to implement the network? Why?

We will use Pytorch to implement the network after doing some research. Even though Keras is one of the most popular library for deep learning network due to its modularity and use-friendliness, Pytorch outshines Keras with its better debugging abilities as well as community support. Plus, due to the short timeline for this project, we want to use the framework that we are most familiar with.

What reference materials will you use to obtain sufficient background on applying the chosen network to the specific problem that you selected?

For background on implementing convolutional networks, we will read over the materials from the Stanford CS231n course, which presents some tips and suggestions for optimizing the training of convolutional networks for visual processing.

In addition, we will study several key research papers that present specific convolutional networks. Krizhevsky et al.'s paper, "ImageNet Classification with Deep Convolutional Neural Networks," presents a network with five convolutional layers and a softmax output layer. The authors also insert max-pooling layers after three of the convolutional layers. Finally, the paper discusses the use of dropout nodes to reduce overfitting.

Kaiming et al. developed the ResNet architecture, which they present and then further refine in two papers from 2015 and 2016, respectively. They insert bypasses, wherein the network skips over some of the convolutional layers to generate residuals in comparison to those layers. These residuals can then be fed into the layer's input to improve training. In addition, Gross and Wilber authored a blog post in 2016, in which they implement ResNet in Torch and compare it to other architectures.

How will you judge the performance of the network? What metrics will you use?

Because this is a supervised learning network for image classification, the primary metric of performance will be the error classification rate. In addition, generating a confusion matrix of the predictions shows whether the network has biases in the results. We will also track the loss against epochs, which will indicate the stability of the learning rate and best batch size. Calculating the difference between the accuracy of the training set and testing set can signify overfitting. Finally, generating a visualization of the first layer can point to architecture problems if the images appear grainy.

Estimated Project Schedule

March 28 - April 3:

- Project proposal

April 4 - 10:

- Data quality checking, cleaning
- Review background papers and other materials
- Design initial network architecture
- Start training
- Review network results
- Write sections 1 – 3 of paper

April 11 - 17:

- Tweak and improve network
- Finalize network
- Updated network results
- Write sections 4 – 6 of paper
- Draft individual project materials

April 18 - 24:

- Finalize paper
- Finalize individual project materials
- Design presentation slides and practice
- Verify code, paper, presentation and additional materials uploaded and organized on Github