Analyzing xView Dataset:
Individual Project Report

Machine Learning II DATS 6203 - 11
Group 2: Jiarong Che, Diana Holcomb, Jiajun Wu

Introduction

In this project we built a deep convolutional neural network (CNN) based on PyTorch to detect objects in high-resolution satellite imagery. We also compared the difference between our model with the given baseline models (vanilla, multi-resolution, and multi-resolution augmented) that come with the xView dataset. xView is the largest publicly available sets of overhead imagery to date, with the over one million objects across 60 classes in over 1,400 square km of imagery.

This topic is of interest to the team from a disaster relief perspective. We're also interested in using our new deep learning and PyTorch skills that we learned in class to analyze this dataset. Finally, the dataset is relatively recent, so there is a lot of unexplored territory and opportunity. These are the main motivation of our project.

Individual Work

I tested the baseline models, and changed the code for CNN model with PyTorch, after my teammates finished the data preprocessing. My own code is 66% of the code, if your code in GitHub counts as my own code as you said in class. Diana and I tried to figure out how to fix that the multi-target cannot fit into criterion loss function, but we failed.

Result

The xView team evaluated their pretrained Tensorflow models with a method known as mAP, or mean average precision.

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$

where Q is the number of class in the set, and AveP(q) is the average precision (AP) for a given class, q.

MultiRes performed the best, with a mAP of 0.2590. Vanilla performed the worst with a mAP of 0.1456. MultiRes_Aug has a mAP of .1549.

	Vanilla	Multires	Aug
Aircraft Hangar	0.1698	0.5270	0.3247
Barge	0.1829	0.3738	0.2210
Building	0.4718	0.5534	0.4451
Bus	0.2949	0.3773	0.2609
Cargo Truck	0.0493	0.0972	0.0445
Cargo/container car	0.3659	0.4737	0.1676
Cement mixer	0.0863	0.1441	0.1220
Construction site	0.0172	0.1711	0.0032
Container crane	0.0663	0.2879	0.1648
Container ship	0.2269	0.4660	0.3400
Crane Truck	0.0946	0.0838	0.0894
Damaged/demolished building	0.0269	0.0785	0.0366
Dump truck	0.1468	0.2275	0.0858
Engineering vehicle	0.0020	0.1234	0.0357
Excavator	0.3535	0.4691	0.2064
Facility	0.0777	0.3750	0.1201
Ferry	0.0532	0.3771	0.2197
Fishing vessel	0.1768	0.1839	0.0968
Fixed-wing aircraft	0.0888	0.1218	0.1042
Flat car	0.0000	0.0000	0.0000
Front loader/Bulldozer	0.1644	0.3220	0.1959
Ground grader	0.1590	0.1910	0.0289
Haul truck	0.3542	0.2109	0.6875
Helicopter	0.3788	0.5800	0.2965
Helipad	0.2459	0.4500	0.1889
Hut/Tent	0.0004	0.0006	0.0000
Locomotive	0.0760	0.1929	0.1124
Maritime vessel	0.1947	0.4040	0.2884
Mobile crane	0.0248	0.1375	0.0945
Motorboat	0.0811	0.2488	0.1110
Oil Tanker	0.0958	0.3677	0.1193
Passenger Vehicle	0.4765	0.5569	0.2980
Passenger car	0.0305	0.0471	0.0000
Passenger/cargo plane	0.6508	0.6691	0.6104

Best-detected classes per model:
Vanilla: passenger/cargo plane, small aircraft, building, passenger car, cargo/container car
MultiRes: passenger/cargo plane, helicopter, shipping container lot, passenger car, building
MultiRes_Aug: haul truck, passenger/cargo plane, small aircraft, building, and tugboat

Conclusions

Processing xView data in PyTorch proved to be much more challenging than our FashionMNIST assignment. PyTorch doesn't easily accommodate variable length in the target arrays. We had hoped to do a comparison of Tensorflow and PyTorch over a large overhead imagery dataset, but got too bogged down in the engineering aspects.
Our current challenge is properly calculating loss from a dataset with a variable length of target arrays.

References

Baseline models are from https://github.com/DIUx-xView/baseline/releases
CNN model is from https://github.com/amir-jafari/Deep-Learning/blob/master/Pytorch_/6-Conv_Mnist/Conv_Mnist_gpu.py