Deep Learning for Land Use Classification

Brian Chan {bchan17}, Advised by Patrick Cho

Abstract

Recent advancements in satellite technology continue to increase the affordability, accessibility, and accuracy of aerial satellite imaging. Insights from these images, particularly over time, is valuable for efforts ranging from urban planning and economic analysis [3] to military efforts, in both urban and rural as well as developed and developing regions.
We utilize CNN architecture to develop an algorithm that, given a satellite image of a geographic region, classifies it based on the type of land use. Classes include scenes from urban, rural, and natural areas, predicting them to be one of 45 classes.

Dataset

Image samples, clockwise from upper left: airport, sea ice, basketball court, and chaparral classes

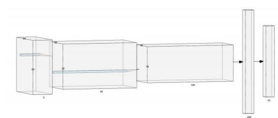
Dataset Features

1. NWPU-RESISC45, a 2017 dataset presented by Gong Cheng, Junwei Han, and Xiaoqiang Lu. It contains 31,500 256x256 RGB images split into 45 classes of 700 images each [2]
2. More diverse in number of classes and samples than other classic land use datasets such as UC Merced Land Use [1], which has only 100 images for 12 classes and Siri-Whu Land Use datasets, which is largely urban China.
3. Contains both urban and rural imaging, small and large features like the "church" class as well as the "harbor" class.
4. Classes include meadow, island, freeway, cloud, etc.

Models

Basic CNN

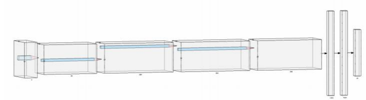
- Multi-layer CNN with ReLU activations and Maxpools.
- Built and trained from scratch
- Tested with different number of layers, filter sizes, and FC nodes
- Final layer is softmax for 45 classes. Utilizes ReLU activations.
- MaxPools not shown. Filter size 3x3.



General CNN architecture. Specifics between models varied

CaffeNet Model

- Derived from CaffeNet architecture, features 5 convolutional layers
- CaffeNet, along with GoogLeNet and AlexNet (from which CaffeNet is derived), are already popular algorithms for use in image classification
- Hyperparameter testing includes editing number of layers, nodes, and filter sizes
- Final layer is a softmax layer for 45 classes



General CaffeNet architecture. Specifics between models varied

Model Equations

- Categorical Cross-entropy: per sample, loss function used to penalize less confident predictions when a sample belongs to a class.

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

$M =$ total number of classes, $y_{o,c} = 1$ if this observation o's true class is c, $p_{o,c} =$ predicted probability of the observation o belonging to c

- RMS-prop: updates weight parameters as a function of the gradient to speed up gradient descent

$$S_{dW} = \mathcal{B} S_{dW}(1-\mathcal{B})dW^2$$
$$S_{db} = \mathcal{B} S_{db}(1-\mathcal{B})db^2$$

$$W = W - \alpha * \sqrt{\frac{dW}{S_{dW} + \epsilon}}$$
$$b = b - \alpha * \sqrt{\frac{db}{S_b + \epsilon}}$$

$\mathcal{B} =$ exponential decay hyperparameter. $SdW, Sdb =$ exponentially weighted average of gradient up to timestep $t-1$. $dW, db =$ gradient at a timestep

Data Processing and Hyperparameter Tuning

Image Preprocessing

1. Image Resizing: All images are resized to 64x64. RGB channels maintained.
2. Data Split: Images are split, per class, into 80:10:10 Training:Validation:Testing. Results in 560:70:70 images per class.
3. Images are also shuffled before distribution in case of underlying patterns in data collection.
4. Keras generator allows for random horizontal flipping of images.

CNN

1. Number of Layers: basic tuning of the number of layers in my CNN model.
2. Learning rate: I experimented between .001 and .002 for the learning rate.

CaffeNet Model

1. Filter Sizes and Strides: slight altering of the filter sizes and strides for early layers in the CaffeNet model.
2. FC Nodes: tuning between 256, 1024, and 2048 nodes in the final FC layers of the network.
3. FC Layers: also experimented with having 1 and 2 FC layers before softmax.

Results

Model	Layers	FC Nodes	LR	Train Acc	Val Acc	Test F1
Basic1	3	512	.001	.4862	.4804	.5593
Basic2	4	512	.001	.4994	.5489	.5342
Basic3	4	1024	.002	.5320	.5345	.5348
Caffe1	Many	256	.001	.5222	.4940	.4948
Caffe2	Many	1024	.001	.6248	.5758	.5846
Caffe3	Many	2048	.001	.6196	.5471	5828

Table 1: Parameters and Metrics of Models

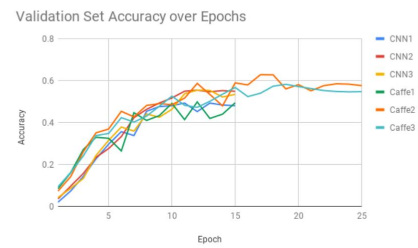


Figure 1. Validation set accuracy over epochs during training. Some models were trained up to 25 epochs.

Discussion

- Overall, both algorithms were not very robust, and they could likely be improved upon by using transfer learning, which will be my next step before the final project report.
- Algorithms with better image segmentation abilities such as ResNet might achieve better results.
- Notably, while using the basic layered CNN, one class, snowberg, had 0 recall.

Future Work

- Final project report: work with pretrained models and transfer learning.
- Investigate other algorithms like GoogLeNet/ResNet.
- Collect/contribute additional classes to the dataset.

References

[1] Marco Castelluccio et al. "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks". In: *CoRR* abs/1508.00092 (2015).

[2] Gong Cheng, Junwei Han, and Xiaoqiang Lu. "Remote Sensing Image Scene Classification: Benchmark and State of the Art". In: *CoRR* abs/1703.00121 (2017).

[3] Neal Jean et al. "Combining satellite imagery and machine learning to predict poverty". In: *Science* 353.6301 (2016), pp. 790–794.