CIS 680 Homework #3

Oleh Rybkin
University of Pennsylvania

Introduction

The task of the homework was to experiment with different versions of faster R-CNN. I completed the task based on the provided example code.

Task 1

In this task I implemented three different CNN architectures for cifar10 classification. The comparison is in the Tab. 1.

For MobileNet, I used the original model from [2] in that the filter multiplier is 1 and the filter doubling is achieved in the subsequent 1x1 convolutional layer. The alternative would be to use filter multiplier 2.

For ResNet, again following the original paper [1], I increased the number of filters in the first layer of a block. This option gives the network even bigger capacity.

Network	ConvNet	MobileNet	ResNet
Test accuracy	73%	55%	74%
Training time	82s	70s	126s
Multiplications	46 497 792	6 824 960	59 179 008

Table 1: Base networks classification performance

As expected, MobileNet performs much less computation than the baseline ConvNet with the tradeoff of lower accuracy, which is in line with the original motivation behind MobileNet blocks. ResNet performs marginally better and is slower than the baseline, as it contains more layers (in the traditional sense of computational steps). Lack of improvement of ResNet may be due to the small number of training epochs which does not let us to fully explore ResNet capacity. It is also harder to train ResNet block that increase filter dimensions, as our model uses additional 1x1 convolutional layer, which could be otherwise set to identity.

Task 2

In this task I trained a version of region proposal network from [3].

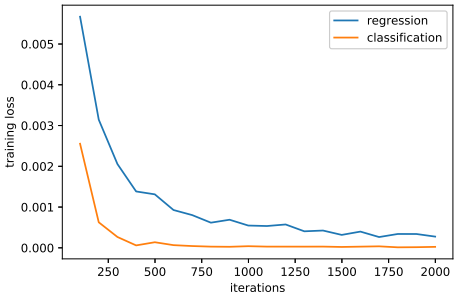

Figure 1: Task 2. Training losses.

The network easily learns to classify objectness with almost absolute test accuracy of 99%. This is expected as the object position is obvious in the images - its boundary has the biggest gradients, object is less blurry than the rest of the image, and it is mainly in the image center[1]

Regression to the bounding box also works almost perfectly as can be seen in the Fig. 2. The testing regression loss is 0.001. The training losses are in the Fig.1.

Task 3

In this task I finished the implementation of the faster R-CNN network. I experimented with different CNN block architectures, comparison between which is in the Tab. 2 and Fig. 5. The difference in performance between networks in this task is similar to the difference in the first task.

Network	ConvNet	MobileNet	ResNet
Test accuracy	53%	35%	55%

Table 2: faster R-CNN networks performance

The drop in performance from the first task should mainly be attributed to the lack of data augmentation in this task. As I do not use image flipping augmentation, the networks overfit strongly. Nevertheless, there is some

[1]Note that while the classifier network is fully translation invariant it still can guess the position in the image using the fact that padded values are always zero.

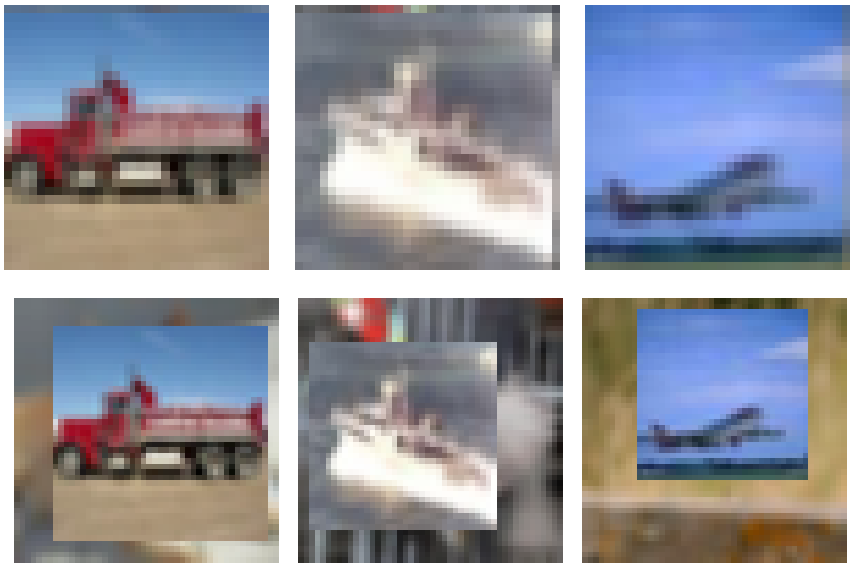

Figure 2: Spatial transformed images. Top: transformed, bottom: original.

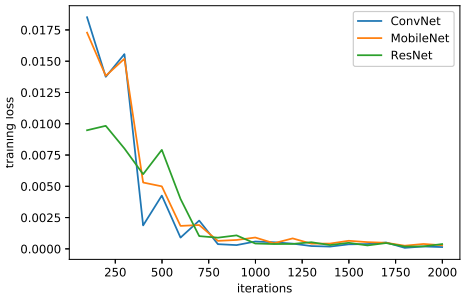

Figure 3: Task 3. Objectness classifier training losses.

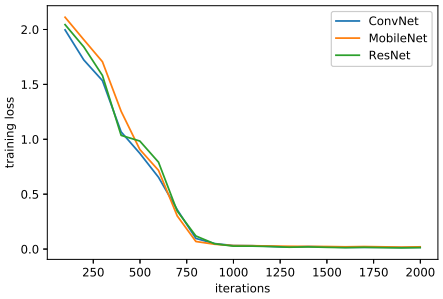

Figure 5: Task 3. Final classifier training losses.

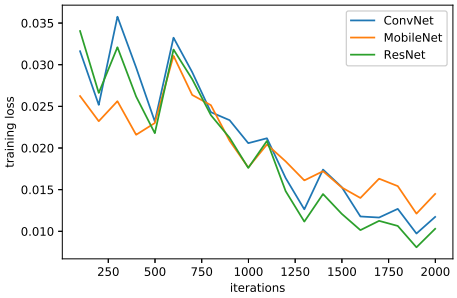

Figure 4: Task 3. Regression training losses.

implicit cropping and padding operation, as the predicted spatial transform is not perfect.

Closing remarks

The homework was implementational in nature and did not require any creative approach. I, however, gained practical and theoretical experience with faster R-CNN and spatial transformer networks.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[3] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.