Going Deeper with Convolutions

1 Citation

Szegedy, Christian, et al. "Going deeper with convolutions." Cvpr, 2015.

https://www.cv-foundation.org/openaccess/content$_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf$

2 Abstract

We win ILSVRC 2014 classification and detection with GoogleNet, which is deep and wide but not too computationally intensive. We introduce the Inception module, which reduces the computational burden, employs the Hebbian principle (neurons that fire together also wire together), and handles multiple scales.

3 Introduction

We set the state of the art on ImageNet classification + detection with a model that does ≤ 1.5 billion multiply-adds.

4 Related Work

LeNet was first to combine convolutional layers, contrast normalization, max-pooling, and fully connected layers. Modern deep networks are deeper and wider, using Dropout to prevent overfitting.

Inspired by neuroscience, previous work used Gabor filters to handle multiple scales. Our Inception module does that.

Network-in-Network used 1×1 filters. We use these filters for dimensionality reduction to reduce computational burden.

R-CNN, the state of the art object detector, uses low-level cues like color and texture to propose regions and then classifies them with a CNN. We improve both of these by proposing multiple boxes and using an ensemble to classify.

5 Motivation and High Level Considerations

To improve a network, we can increase its depth. However, this can lead to overfitting and it can eat up computation resources (linear increase in number of filters yields quadratic increase in compute usage). You can solve both of these problems by having sparse connections between layers. This also mimics biology, where neurons that fire together also wire together (Hebbian principle).

The challenge, however, is that modern GPU/CPU hardware is designed for dense computations. The way people handle this problem is by clustering a sparse matrix into dense submatrices. The Inception module was built to see if this concept can be applied to deep learning.

6 Architectural Details

How do we model local sparsity well? Let's try using 1×1, 3×3, and 5×5 filters and concatenate the filter outputs together. We also combine it with a pooling output. Finally, to limit the number of computations, we do 1×1 before hand. Thus, we have four paths from layer to the concatenation: 1×1 convolution, 1×1 conv followed by 3×3 conv, 1×1 conv followed by 5×5 conv, and 3×3 max pooling followed by 1×1 conv - this is called an Inception module. To build an Inception network, use some convolutional layers and then stack a bunch of Inception modules mixed with max pooling every now and then.

7 GoogLeNet

We try a bunch of architectures, but basically they are just two conv layers, bunch of inception layers (with occasional max pooling), average pooling, dropout at 40%, linear layer (to fine-tune, just retrain this), and softmax.

We tried using auxiliary classification layers in the middle of the network, computing loss there, and combining all the losses together. We thought this would help the gradient flow more easily if loss was computed at different depths, but it turned out not to make much of a difference.

8 Training Methodology

You could train it in a week using a few high end GPUs. We use SGD with 0.9 momentum and cut learning rate by 4% every 8 epochs. We took crops that were between 8% to 100% of the image size and we applied distortions to the images too - this data augmentation helped fight overfitting.

9 ILSVRC 2014 Classification Challenge Setup and Results

ILSVRC classification is 1000-way image classification on a 1.3M image dataset where you are scored by top-5 error rate (we also compute top-1 error rate).

We trained 7 models.

At test-time, we pick 4 image scales and take the 4 corners and center. We predict for each and average over scale, crop, and model.

10 ILSVRC 2014 Detection Challenge Setup and Results

ILSRVC detection is 200-category multi-object + bounding box prediction. A detection is correct if category is correct and bounding box overlaps ground truth with intersection-over-union ¿ 50%. False positives are penalized. Final metric is mean-average-precision.

We use R-CNN, but use Inception network for region classifier. To improve region proposal, we combine selective search with multi-box proposals. We increase super-pixel size by 2x to halve proposals. Ultimately, we get 60% of the proposals but increase coverage by 1%. Each region is classified by our GoogLeNet ensemble of 6 models. We didn't use bounding box regression (an area for improvement).

11 Conclusion

The Inception module was effective at modeling local sparse connections while limiting the number multiply-adds.