Xception Architecture for Image Classification

Yuliya Trofimova

***Abstract*—This article is about evolution of convolutional neural networks. We are going to consider Inception and Xception architectures and describe general principles of working. Also we will compare their size, operating time and accuracy.**

Introduction

Convolutional Neural Network (CNN) models solve various computer vision problems such as classifications[1], prediction[2], regression[3], search query retrieval, sentence modeling etc. But we will consider problem of classification. There is The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale. In 2012, there is an epoch-making event - ILSVRC was won deep convolutional network with an error of prediction is 15%, while next winner had 26%. It was AlexNet CNN's by Alex Krizhevsky from Hinton University. It had 5 convolutional and 3 fully-connected layer and also 60 million weights. After that Google started to improve this CNN and make more pragmatic[4]. The main ideas are:

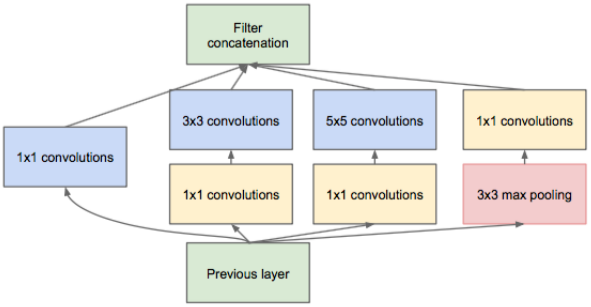
1) Reduce convolutions and increase layer.

2) Sharply reduce the number of dimensions using convolution.

3) At each layer, we will run several 1x1 convolution kernels of different sizes to get features of different scale. If the scale is too large for the current layer, it is recognized on the next.

4) Do not do hidden FC layers at all, because there are so many parameters in them. Instead, at the last level, make a global average pool and attach it to the output layer directly.

Generally, CNN consists of convolution layer. It attempts to analysis in cross-channel correlations (it is about width, height and channel) with use of 1x1 convolutions it maps data into 3 or 4 separate spaces with smaller dimensions, learning correlations between channels, and then performs regular 1x1 convolutions to learn spatial correlations. One Inception module looks like this, but GoogLeNet consists of 9 such units.

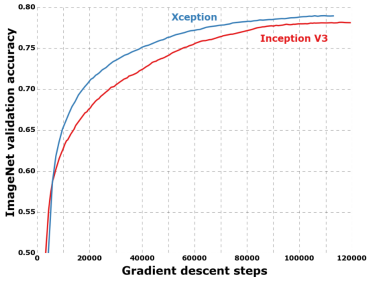


This CNN affords 6.67% error, but works 10 times faster.

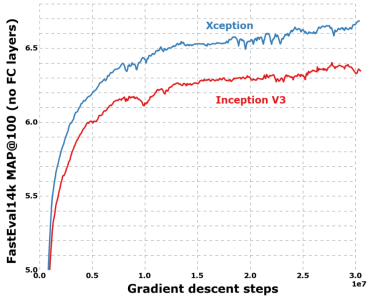
Main part

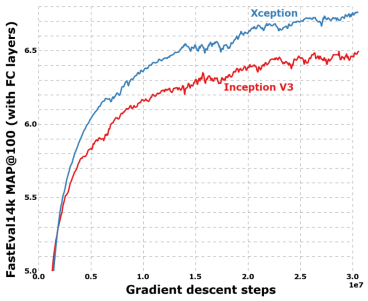
Xception is extreme version of an Inception module with using depthwise separable convolution instead of simple convolutions. It is also include part "pointwise convolution", when at first use a 1x1 convolution to map cross-channel correlations. Next step is depthwise spatial convolution use 3x3 convolutions performed on each channel severally. It can be seen as Inception module without non-linearities which maps data to significantly bigger number of dimensions. The hypothesis of this architecture is decoupling map of cross-channels correlations and spatial correlations. The Xception architecture has 36 convolutional layers forming the feature extraction base of the network, and are structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules. It is a linear stack of depthwise separable convolution layers with residual connections. in the implementation use a high-level library such as Keras[5] or TensorFlowSlim[6].

For comparison Xception and Inception V3 was used JFT and ImageNet dataset. Next graph shows slight enhancement in evaluation for ImageNet (4.3% relative improvement on the FastEval14k MAP@100 metric).



The Xception architecture shows a much larger improvement on the JFT dataset compared to the ImageNet dataset. Presumably, Inception V3 was developed with a focus on ImageNet and may be overfit to this dataset. On the other hand, neither architecture was tuned for JFT. Probably, setting parameters can give a better result.

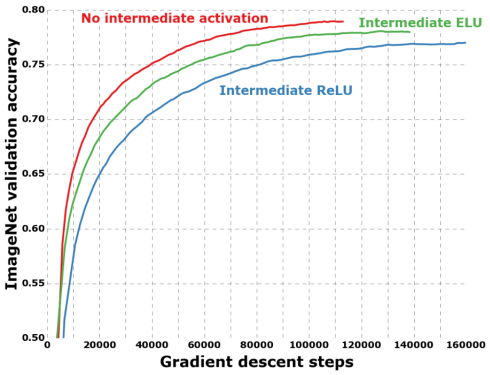




Although Xception give better accuracy, it works slower for the same size.

	Parameter count	Steps/second
Inception V3	23,626,728	31
Xception	22,855,952	28

Due to the fact that both architectures have the same number of parameters, the improvement seen on ImageNet and JFT does not come from added capacity but rather from a more efficient use of the model parameters. There is experimentally established that to include non-linear did not give improving result. The graph shows inclusion of either ReLU or ELU as intermedi- ate non-linearity and follow-up testing on ImageNet.



There is no reason to believe that depthwise separable convolutions are optimal. It may be that intermediate points on the spectrum, lying between regular Inception modules and depthwise separable convolutions, hold further advantages. This question is left for future investigation.

Conclusion

We showed how convolutions and depthwise separable convolutions lie at both extremes of a discrete spectrum, with Inception modules being an intermediate point in between. This observation has led to us to propose replacing Inception modules with depthwise separable convolutions in neural computer vision architectures. We presented a novel architecture based on this idea, named Xception, which has a similar parameter count as Inception V3. Compared to Inception V3, Xception shows small gains in classification performance on the ImageNet dataset and large gains on the JFT dataset. We expect depthwise separable convolutions to become a cornerstone of convolutional neural network architecture design in the future, since they offer similar properties as Inception modules, yet are as easy to use as regular convolution layers.

References

[1] Kim, Yoon Convolutional Neural Networks for Sentence Classification
[2] Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning. "Proceedings of the 25th international conference on Machine learning. ACM, 2008.
[3] YangJing Long (2009). "Human age estimation by metric learning for regression problems"
[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich (2014). "Going Deeper with Convolutions"
[5] F. Chollet. Keras. https://github.com/fchollet/keras, 2015.
[6] N. Silberman and S. Guadarrama. Tf-slim, 2016
[7] Franois Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, 2017