INCEPTION-V4, INCEPTION-RESNET AND THE IMPACT OF RESIDUAL CONNECTIONS ON LEARNING.

Christian Szegedy & Sergey Ioffe & Vincent Vanhoucke
{szegedy,sioffe,vanhoucke}@google.com

ABSTRACT

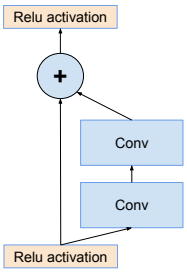
Very deep convolutional networks have been central to the largest advances in image recognition performance in recent years. One example is the Inception architecture that has been shown to achieve good performance at relatively low computational cost. Recently, the introduction of residual connections in conjunction with a more traditional architecture has yielded state-of-the-art performance in the 2015 ILSVRC challenge; its performance was similar to the latest generation Inception-v3 network. This raises the question of whether there are any benefit in combining the Inception architecture with residual connections. Here we give clear empirical evidence that training with residual connections accelerates the training of Inception networks significantly, however, when fully trained, the final quality of the non-residual Inception variants seem to be close to those of residual versions. We present several new streamlined architectures for both residual and non-residual Inception networks. With an ensemble of three residual and one pure Inception-v4, we achieve 3.08% top-5 error on the test set of the ImageNet classification (CLS) challenge.

1 INTRODUCTION

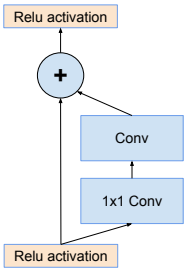
In this work we study the combination of the two most recent ideas: Residual connections introduced in He et al. (2015) and the latest revised version of the Inception architecture Szegedy et al. (2015). In He et al. (2015), it is argued that residual connections are of inherent importance for training very deep architectures. However, Since Inception networks tend to be very deep, it is natural to replace the filter concatenation stage of the Inception architecture with residual connections. This would allow Inception networks to reap all the benefits of the residual approach while retaining their computational efficiency. Besides a straightforward integration, we have also studied whether Inception itself can be made more efficient by making it deeper and wider. For that purpose, we designed a new version named Inception-v4 which has a more uniform simplified architecture and more inception modules. In this report, we will compare the two pure Inception variants, Inception-v3 and v4, with similarly expensive hybrid Inception-ResNet versions and tested their performance on the ImageNet classification challenge Russakovsky et al. (2014) dataset.

2 MODEL

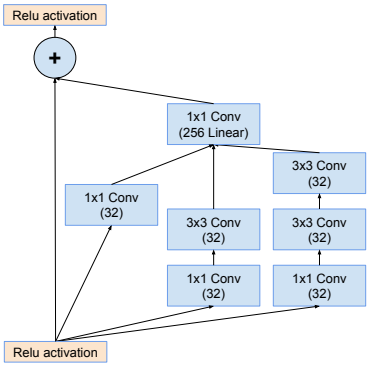
Residual connection were introduced by He et al. in He et al. (2015) in which they give convincing theoretical and practical evidence for the advantages of utilizing additive merging of signals both for image recognition, and especially for object detection. See figures 1a and 1b. The authors argue that residual connections are inherently necessary for training very deep convolutional models. Our findings do not seem to support this hypothesis. In the experimental section we demonstrate that it is not too difficult to train competitive very deep networks without utilizing residual connections. However the use of residual connections seems improve the training speed greatly, which is alone a great argument for their use. Here, we propose a hybrid Inception-ResNet architecture for computer vision built from hybrid like in Figure 1c. Note that although this increases the number of layers, the overall computation performed by each layer might be reduced. Our overall proposed network architecture is based in the Inception-v3 network introduced in Szegedy et al. (2015). The



(a) Residual module introduced by He et al. (2015)

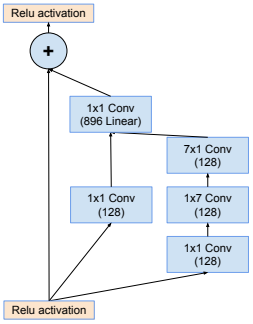


(b) Optimized version of residual modules to shield computation, like in Lin et al. (2013).

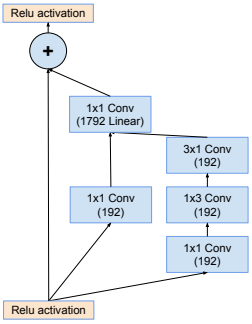


(c) Inception-ResNet example module for the 35×35 grid.

Figure 1: Residual network module variants. The first two variants were introduced in He et al. (2015).



(a) Inception-ResNet example module for the 17×17 grid



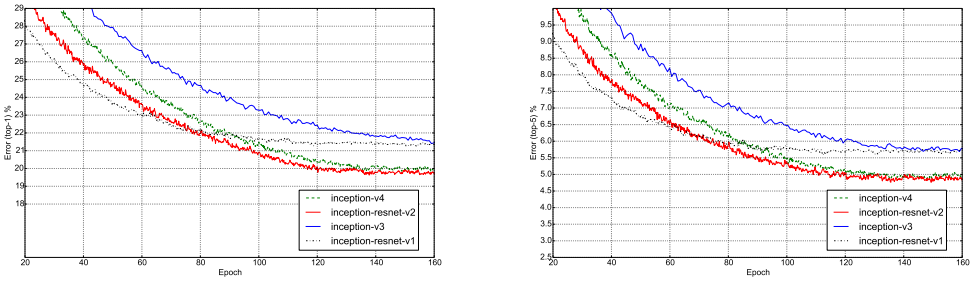
(b) Inception-ResNet example module for the 8×8 grid

Figure 2: Further example Inception module variants optimized for certain grid sizes.

main difference is that we replace each inception module by modules like in 1c. Also, in addition, we increase the number of modules. An extra change that seemed to improve the stability of the training of residual variants is to scale the output of the residuals by a relatively small numbers, 0.1 in our experiments. The intuitive motivation for that is that the activation vectors should approximate a path in a high-dimensional space as the number of modules increases. In addition to the ResNet variants we tried a new costlier version of Inception-v3, code named Inception-v4 which does not employ residual connections but utilizes more wider Inception modules than Inception-v3. We have trained our networks with stochastic gradient utilizing the TensorFlow Abadi et al. (2015) distributed machine learning system using 20 replicas running each on a NVidia Kepler GPU using RMSProp Tieleman & Hinton and a learning rate of 0.045, decayed every by 6% in every two epochs.

3 Results

We have tested the above variants on the ImageNet classification dataset Russakovsky et al. (2014). Firs we compare the training behavior of Inception-v3 Szegedy et al. (2015) with Inception-ResNet-v1 utilizing the hybrid residual Inception modules While the residual variant seems to train much faster, it levels off at an almost identical error rate as the traditional Inception variant as can be seen in Figure 3 This graph also shows the training behavior of the newline introduced simplified, but more expensive Inception-v4 network with a similarly costly hybrid Inception-ResNet-v2 variant which has more filters per layer than Inception-ResNet-v1. We can see that both larger networks yielded very similar results while the residual variant trained faster and reached a slightly better result.



(a) Top-1 error

(b) Top-5 error

Figure 3: Error evolution of two wider Inception variants. The residual model has trained faster and reached very similar final accuracy and top-5 recall as the traditional variant with similar cost.

Network	Models	**Top-1 Error**	**Top-5 Error**
ResNet-151 He et al. (2015)	6	–	3.6%
Inception-v3 Szegedy et al. (2015)	4	17.3%	3.6%
Inception-v4(+Residual)	4	16.5%	3.1%

Table 1: Ensemble results

However the final quality seems to be much more correlated with the model size than with the use of residual connections. Finally we took our four best models, including one traditional Inception-v4 and three Inception-ResNet-v2 style models that differed less then 0.2% in their single crop top-1 accuracy. The ensembled results can be seen in Table Ensemble results with 144 crops/dense evaluation. We report them on the test of ILSVRC 2012. For Inception-v4(+Residual), the ensemble consists of one pure Inception-v4 and three Inception-ResNet-v2 models and were evaluated both on the validation and on the test-set. The test-set performance (as measured on the test server) was 3.08% top-5 error verifying that we didn't over-fit on the validation set. The other two results were reported during the ILSVRC 2015 Competition.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

Tijmen Tieleman and Geoffrey Hinton. Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 4, 2012. Accessed: 2015-11-05.