MIDDLE EAST TECHNICAL UNIVERSITY

DI504
Foundations of Deep Learning

Advanced Architectures

This week:

- This week we are going to talk about some advanced architecture designs in deep learning.
- We will start with two succeders of AlexNet and VGGNet
  - ResNet and GoogleNet
- These networks are important to understand, because unlike Alexnet or VGGNet, they are directed-acyclic-graphs, not serial networks.
- And they are still being widely used as universal feature extractors.

Residual Networks

- *Problem*: When deeper networks starts converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated and then degrades rapidly. (a.k.a vanishing gradients)
- This figure is from the original Resnet paper [He et al., 2015].
- By experimentation, [He et al., 2015] have shown that for a plain/vanilla CNN, as the network got deeper, (after some point) the training loss did not improve.

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

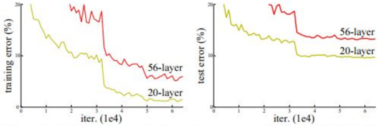


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

METU

Residual Networks

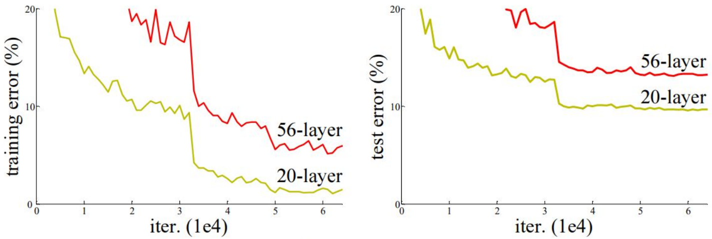


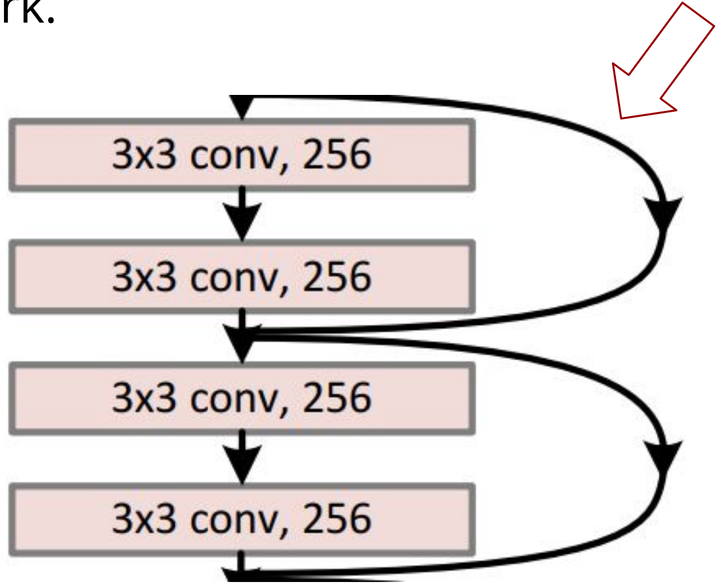
Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- *Problem*: When deeper networks starts converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated and then degrades rapidly. (because of a problem called vanishing gradients)
- This figure is from the original Resnet paper [He et al., 2015].
- By experimentation, [He et al., 2015] have shown that for a plain/vanilla CNN, as the network got deeper, (after some point) the training loss did not improve.

METU

Residual Networks

- They performed a simple set, where they introduced a type of skip connection to a serial (VGG like) network.



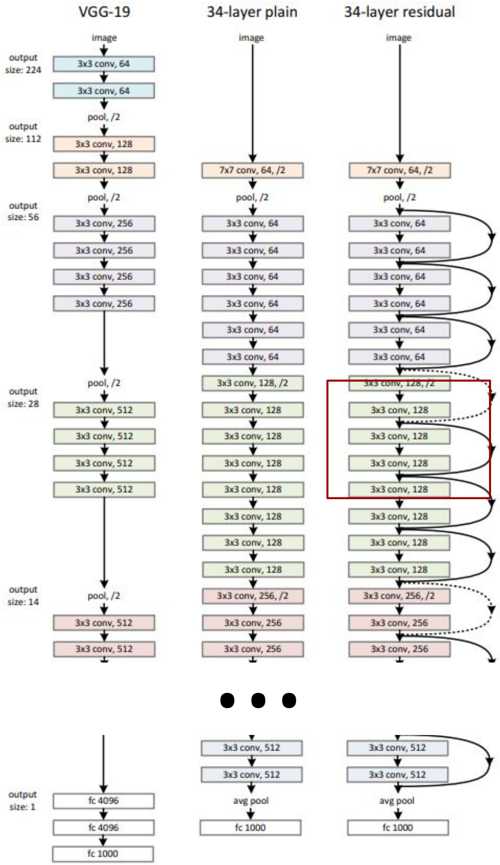


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

Residual Networks

- They performed a simple set, where they introduced a type of skip connection to a serial (VGG like) network.

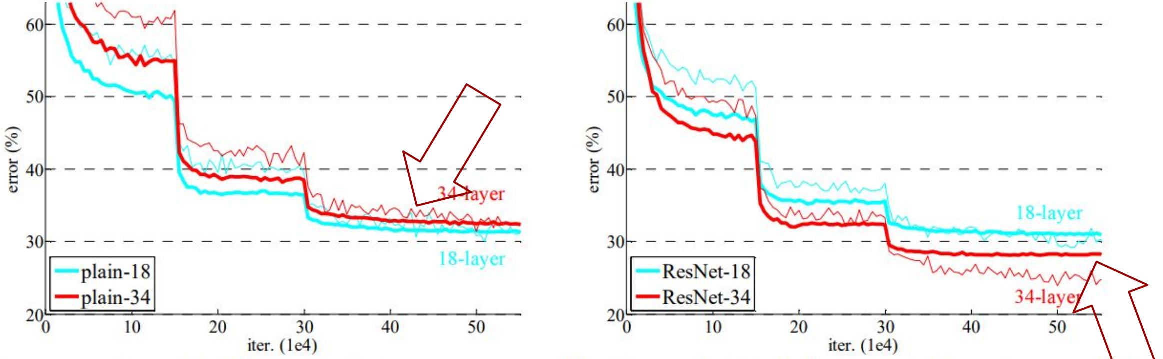


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

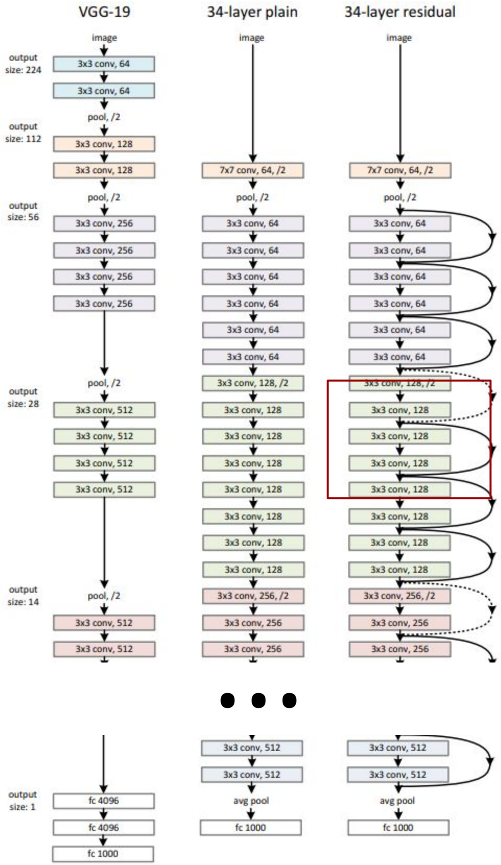
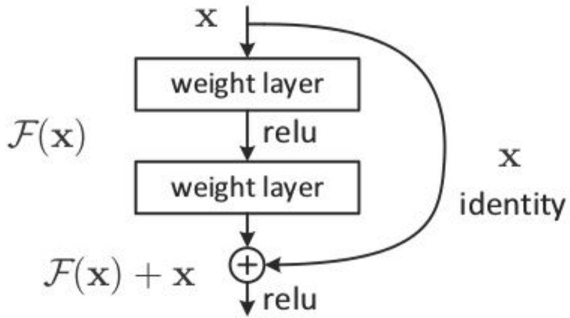


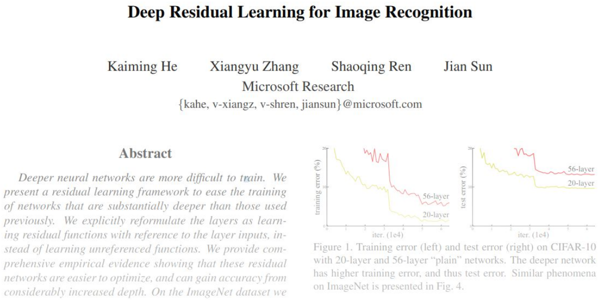
Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

Residual Networks

- So, [He et al., 2015] observed that, as the network got deeper, training was more difficult.
- In their paper, they show that the phenomenon is not largely related to overfitting, but vanishing gradient. (please read the paper, it is an AI miltestone)
- The theory behind ResNet idea was that
  - «Each layer had to learn the representation from scratch» (which is correct)
  - As the networks gets deeper, it get harder to optimize this.
  - So why not let some layers, «not learn», if they do not want to. Or if they do not **need** to.



Residual Networks


Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

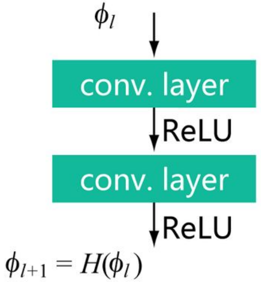
Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- So let the layer (or block) pass the input as is, instead of ruining it.
- The idea is to add an «identity mapping» from the input to the output, via an empty skip connection.

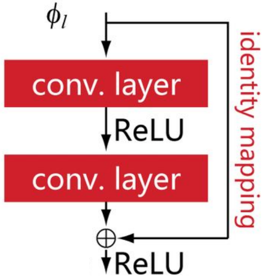
Plain Convolutional Block



$$y = F(x)$$

vs

Residual Convolutional Block



$$y = F(x) + x$$

METU

Residual Networks

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

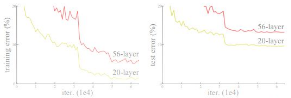
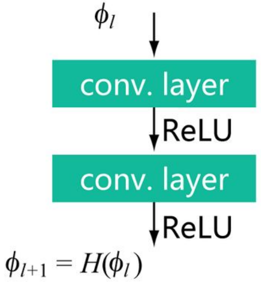


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- The author's **hypothesis** was that
  it is easy to optimize the residual mapping function F(x)+x than to optimize the original, unreferenced mapping F(x).
- Why?

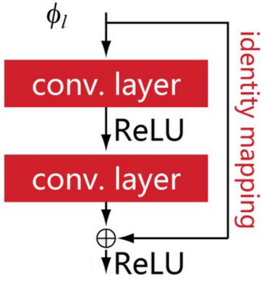
Plain Convolutional Block



$y = F(x)$

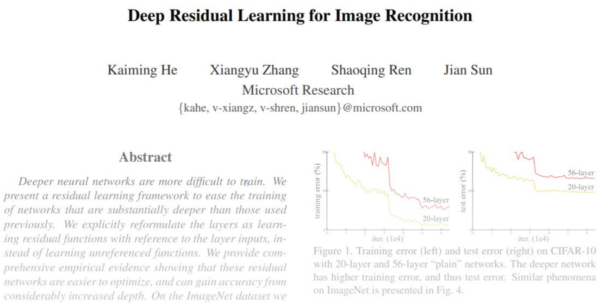
vs

Residual Convolutional Block



$y = F(x) + x$

Residual Networks


Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

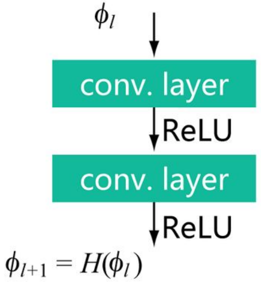
Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- The real magic takes place in the backward pass.
- The gradients never vanish, because there is always an «open highway»

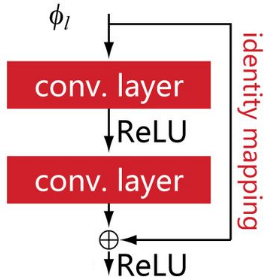
Plain Convolutional Block



$$y = F(x)$$

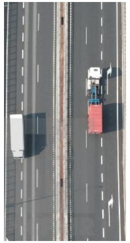
vs

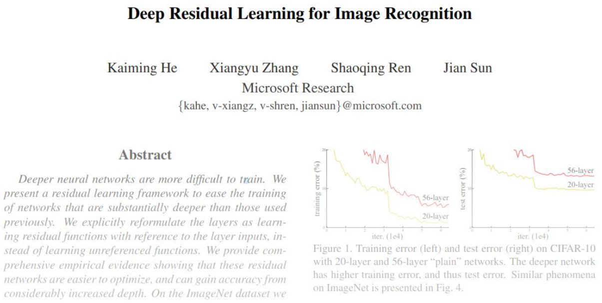
Residual Convolutional Block



$$y = F(x) + x$$

gradients flow...





Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Residual Networks

- They also come up with an efficient version
- A Bottleneck Residual Block is a variant of the residual block that utilises 1x1 convolutions to create a bottleneck.
- The use of a bottleneck reduces the number of parameters and matrix multiplications.
- The idea is to make residual blocks as thin as possible to increase depth and have less parameters.

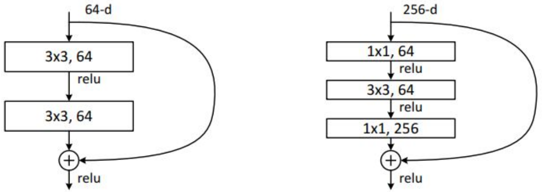
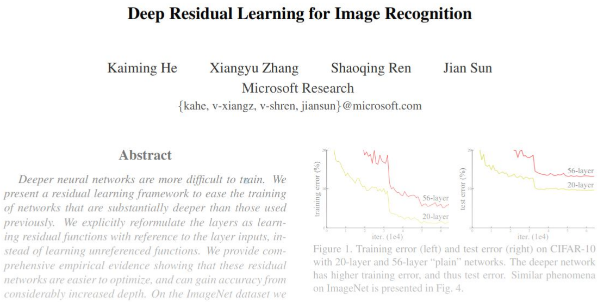


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

METU

Residual Networks


Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- They also come up with an efficient version
- A Bottleneck Residual Block is a variant of the residual block that utilises 1x1 convolutions to create a bottleneck
- The use of a ... ers and matrix multi
- The idea is t ...) increase dep

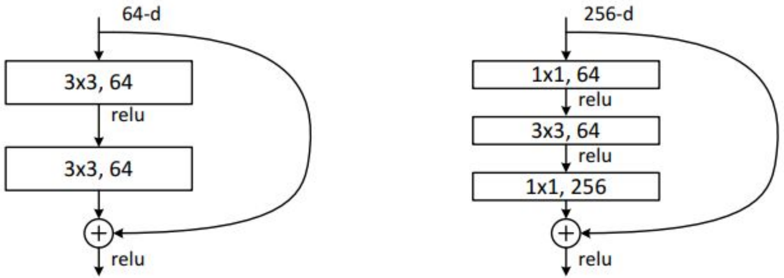
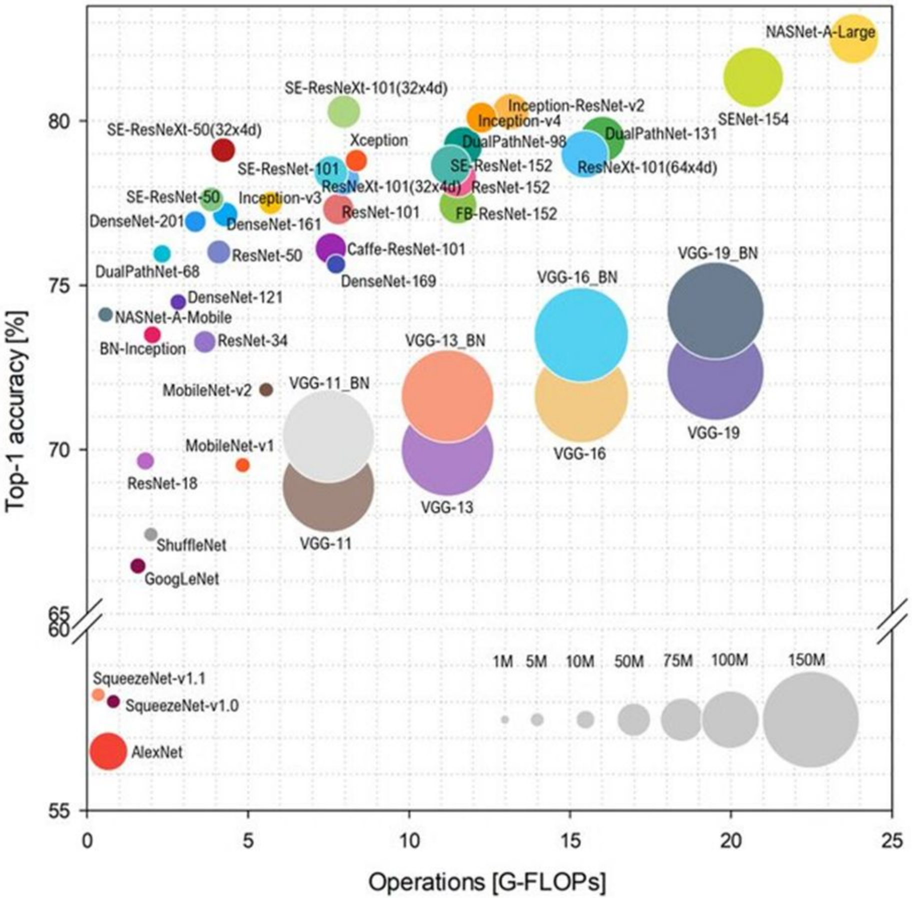


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

Classification Networks

- There are many classification studies that succeeded AlexNet and VGG.
- All trying to optimize computation, memory and accuracy.



METU

Inception Modules

- Inception Module’s constant evolution lead to the creation of several versions of the network. The popular versions are as follows:
  - Inception v1.
  - Inception v2 and Inception v3.
  - Inception v4 and Inception-ResNet.
- Each version is an iterative improvement over the previous one. Understanding the upgrades can help us to build custom classifiers that are optimized both in speed and accuracy.
- Also, depending on your data, a lower version may actually work better.

Going deeper with convolutions

Christian Szegedy Google Inc.
Wei Liu University of North Carolina, Chapel Hill
Yangqing Jia Google Inc.
Pierre Sermanet Google Inc.
Scott Reed University of Michigan
Dragomir Anguelov Google Inc.
Dumitru Erhan Google Inc.
Vincent Vanhoucke Google Inc.
Andrew Rabinovich Google Inc.

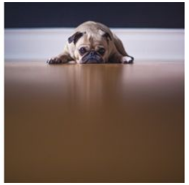
Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

Inception Modules

- Salient parts in the image can have extremely large variation in size. For instance, an image with a dog can be either of the following, as shown below. The area occupied by the dog is different in each image.





- Because of this huge variation in the location of the information, choosing the right kernel size for the convolution operation becomes tough.
- A larger kernel is preferred for information that is distributed more globally, and a smaller kernel is preferred for information that is distributed more locally.

Going deeper with convolutions

Christian Szegedy Google Inc.
Wei Liu University of North Carolina, Chapel Hill
Yangqing Jia Google Inc.
Pierre Sermanet Google Inc.
Scott Reed University of Michigan
Dragomir Anguelov Google Inc.
Dumitru Erhan Google Inc.
Vincent Vanhoucke Google Inc.
Andrew Rabinovich Google Inc.

Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

Inception Modules

- Salient parts in the image can have extremely large variation in size. For instance, an image with a dog can be either of the following, as shown below. The area occupied by the dog is different in each image.

- Because of this huge variation in the location of the information, choosing the right kernel size for the convolution operation becomes tough.
- A larger kernel is preferred for information that is distributed more globally, and a smaller kernel is preferred for information that is distributed more locally.

Going deeper with convolutions

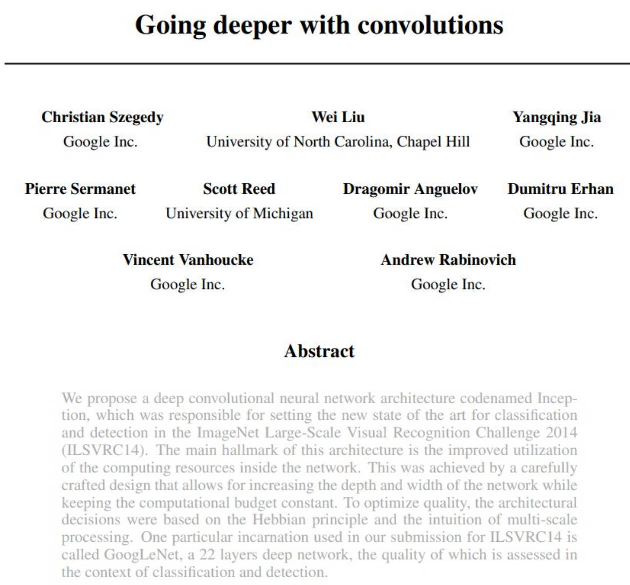
Christian Szegedy Google Inc.
Wei Liu University of North Carolina, Chapel Hill
Yangqing Jia Google Inc.
Pierre Sermanet Google Inc.
Scott Reed University of Michigan
Dragomir Anguelov Google Inc.
Dumitru Erhan Google Inc.
Vincent Vanhoucke Google Inc.
Andrew Rabinovich Google Inc.

Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

METU

Inception Modules


Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

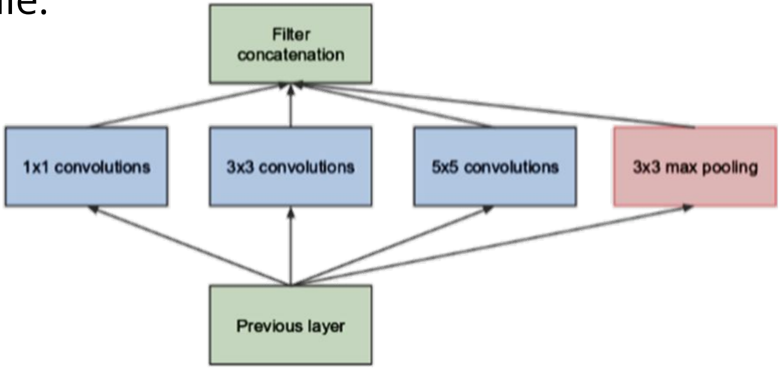
Vincent Vanhoucke
Google Inc.

Andrew Rabinovich
Google Inc.

Abstract

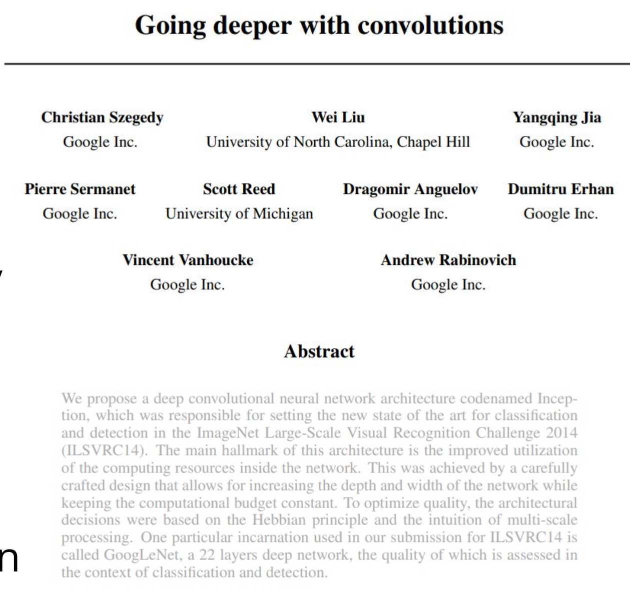
We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

- Why not have filters with multiple sizes operate on the same level?
- The network essentially would get a bit “wider” rather than “deeper”.
- The below image is the “naive” inception module.
  It performs convolution on an input, with 3 different sizes of filters (1x1, 3x3, 5x5).
- Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module.



Inception Modules

- As stated before, deep neural networks are computationally expensive.
- To make it cheaper, the authors limit the number of input channels by adding an extra 1x1 convolution before the 3x3 and 5x5 convolutions.
- Though adding an extra operation may seem counterintuitive, 1x1 convolutions are far more cheaper than 5x5 convolutions, and the reduced number of input channels also help.

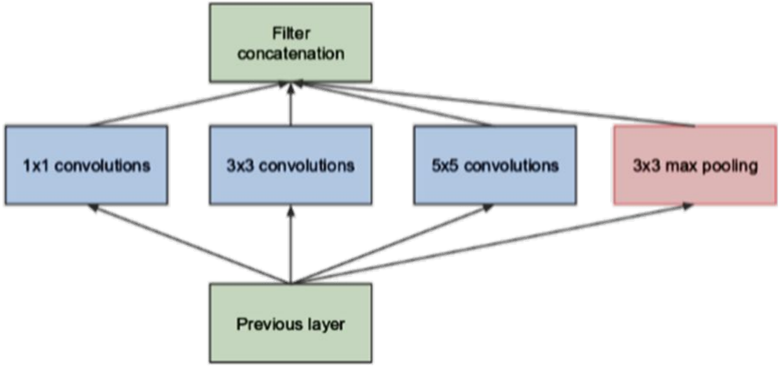


Going deeper with convolutions

Christian Szegedy, Google Inc.
Wei Liu, University of North Carolina, Chapel Hill
Yangqing Jia, Google Inc.
Pierre Sermanet, Google Inc.
Scott Reed, University of Michigan
Dragomir Anguelov, Google Inc.
Dumitru Erhan, Google Inc.
Vincent Vanhoucke, Google Inc.
Andrew Rabinovich, Google Inc.

Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.



Inception Modules (Naive vs v.1)

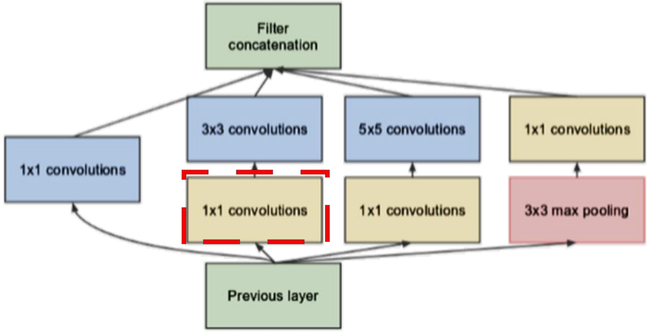
- For an the input is M x N x D1, let's assume that
  - the filter is f x g x L provides
  - O x P x L output
  - with O x P x D1 x f x g x L multiplications
- With a M x N x D2 conv layer in between (where D2 < D1), you will do less operations:
  - O x P x D2 x f x g x L multiplications
- Most of the time the input depth (D1) is mostly correlated and redundant.

Going deeper with convolutions

Christian Szegedy, Google Inc.
Wei Liu, University of North Carolina, Chapel Hill
Yangqing Jia, Google Inc.
Pierre Sermanet, Google Inc.
Scott Reed, University of Michigan
Dragomir Anguelov, Google Inc.
Dumitru Erhan, Google Inc.
Vincent Vanhoucke, Google Inc.
Andrew Rabinovich, Google Inc.

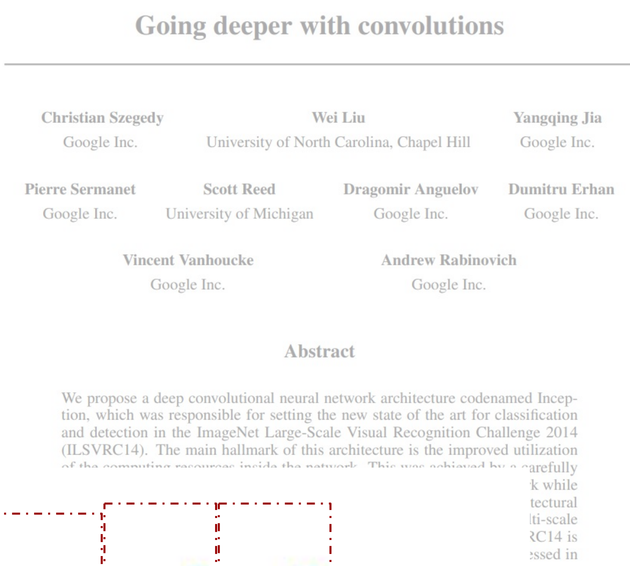
Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.



Inception Modules (Naive vs v.1)

- The GoogleNet (i.e. Inception Net)
  - A collection of inception modules.

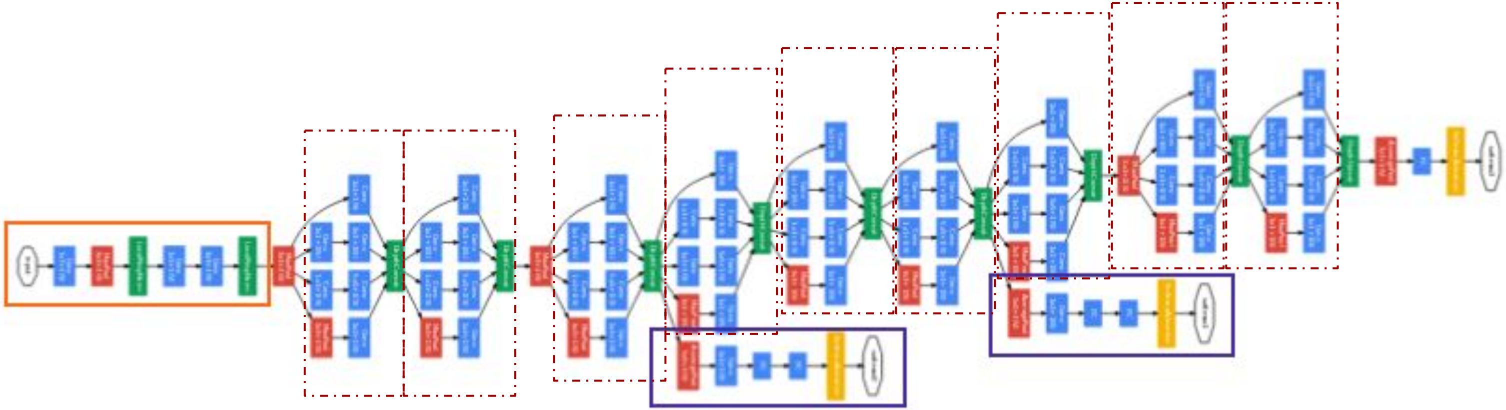


Going deeper with convolutions

Christian Szegedy, Google Inc.
Wei Liu, University of North Carolina, Chapel Hill
Yangqing Jia, Google Inc.
Pierre Sermanet, Google Inc.
Scott Reed, University of Michigan
Dragomir Anguelov, Google Inc.
Dumitru Erhan, Google Inc.
Vincent Vanhoucke, Google Inc.
Andrew Rabinovich, Google Inc.

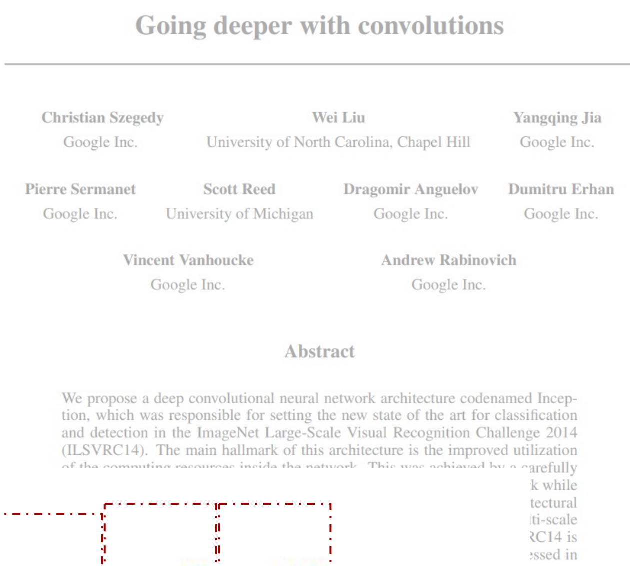
Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization ... arefully ... k while ... tectural ... lti-scale ... RC14 is ... ssed in



Inception Modules (Naive vs v.1)

- The GoogleNet (i.e. Inception Net)
  - A collection of inception modules.



Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

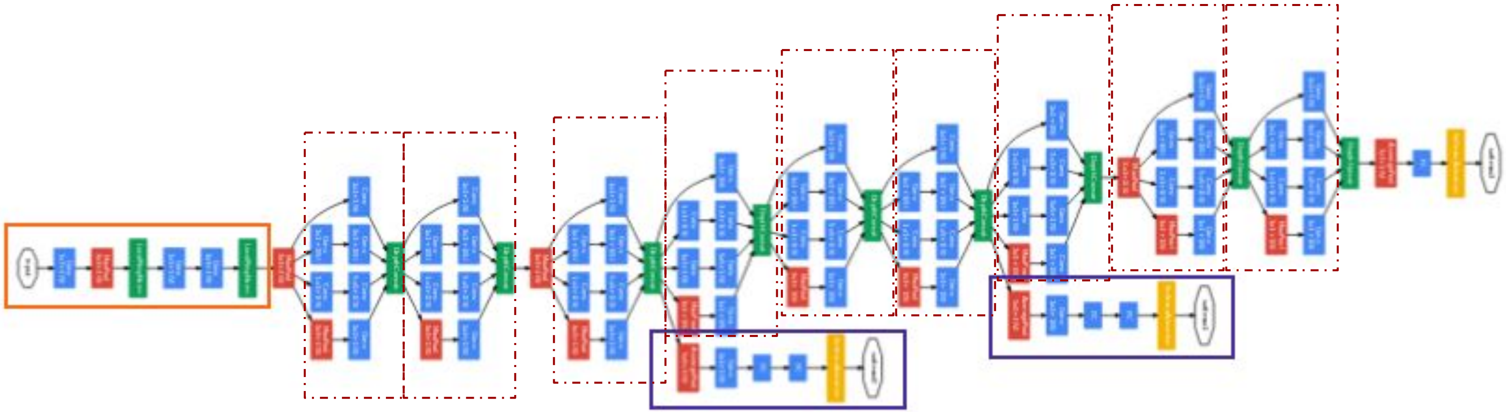
Dumitru Erhan
Google Inc.

Vincent Vanhoucke
Google Inc.

Andrew Rabinovich
Google Inc.

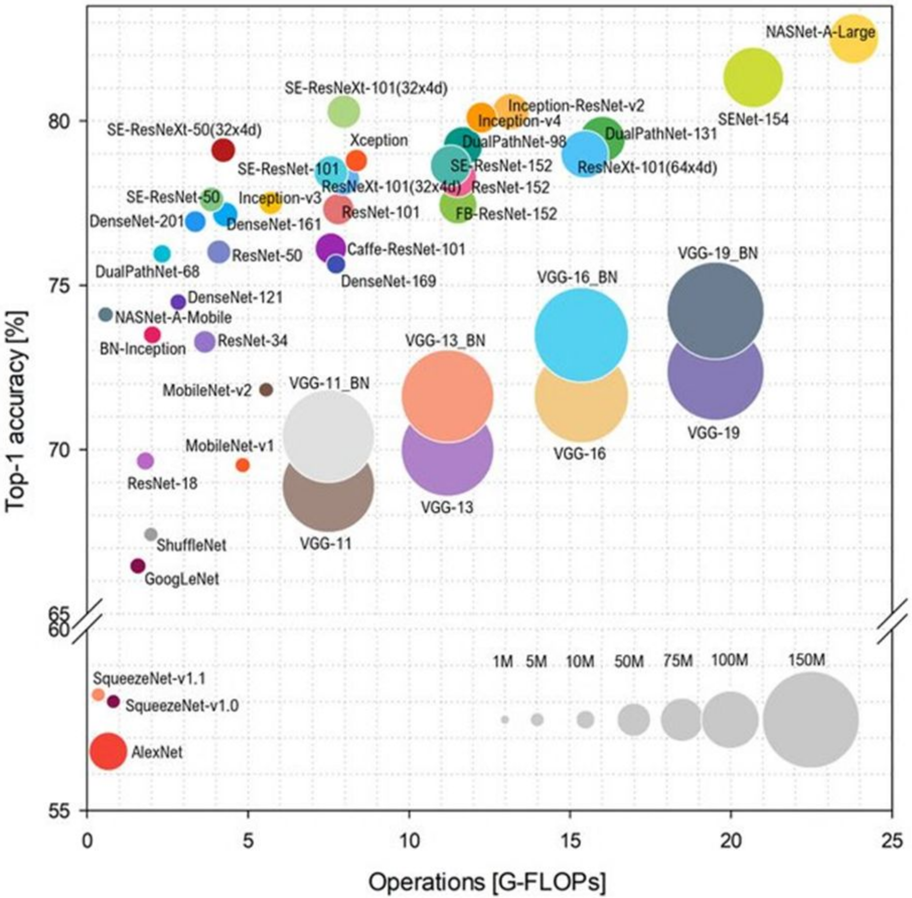
Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization

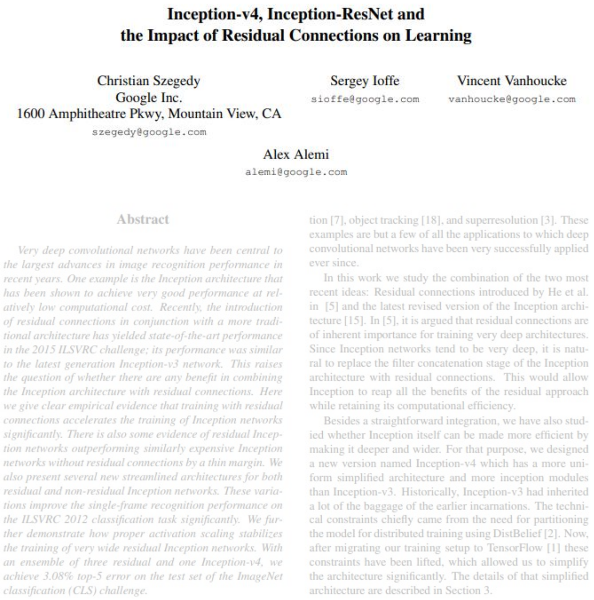


Classification Networks

- There are many classification studies that succeeded AlexNet and VGG.
- All trying to optimize computation, memory and accuracy.



Inception ResNet



Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

Christian Szegedy
Google Inc.
1600 Amphitheatre Pkwy, Mountain View, CA
szegedy@google.com

Sergey Ioffe
sioffe@google.com

Vincent Vanhoucke
vanhoucke@google.com

Alex Alemi
alemi@google.com

Abstract

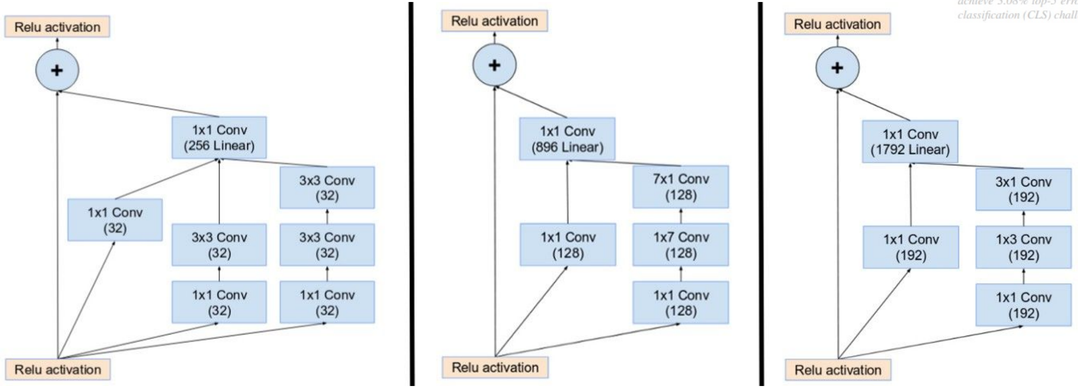
Very deep convolutional networks have been central to the largest advances in image recognition performance in recent years. One example is the Inception architecture that has been shown to achieve very good performance at relatively low computational cost. Recently, the introduction of residual connections 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. There is also some evidence of residual Inception networks outperforming similarly expensive Inception networks without residual connections by a thin margin. We also present several new streamlined architectures for both residual and non-residual Inception networks. These variations improve the single-frame recognition performance on the ILSVRC 2012 classification task significantly. We further demonstrate how proper activation scaling stabilizes the training of very wide residual Inception networks. With an ensemble of three residual and one Inception-v4, we achieve 3.08% top-5 error on the test set of the ImageNet classification (CLS) challenge.

tion [7], object tracking [18], and superresolution [3]. These examples are but a few of all the applications to which deep convolutional networks have been very successfully applied ever since.

In this work we study the combination of the two most recent ideas: Residual connections introduced by He et al. in [5] and the latest revised version of the Inception architecture [15]. In [5], it is argued that residual connections are of inherent importance for training very deep architectures. 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 to reap all the benefits of the residual approach while retaining its 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 than Inception-v3. Historically, Inception-v3 had inherited a lot of the baggage of the earlier incarnations. The technical constraints chiefly came from the need for partitioning the model for distributed training using DistBelief [2]. Now, after migrating our training setup to TensorFlow [1] these constraints have been lifted, which allowed us to simplify the architecture significantly. The details of that simplified architecture are described in Section 3.

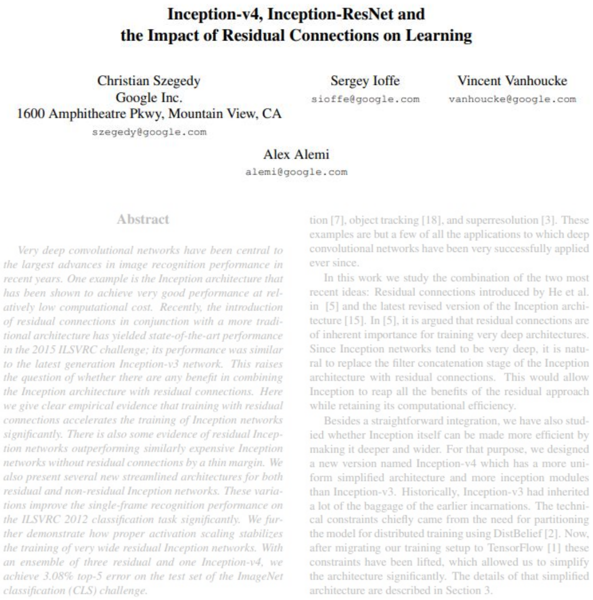
- First Google guys adopted Residual connections to their architecture, hence the Inception ResNet
- The Premise: Introduce residual connections that add the output of the convolution operation of the inception module, to the input.



METU

Inception ResNet

- The original paper didn't use BatchNorm after summation to train the model on a single GPU (To fit the entire model on a single GPU).
- It was found that Inception-ResNet models were able to achieve higher accuracies at a lower epoch.
-



Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

Christian Szegedy
Google Inc.
1600 Amphitheatre Pkwy, Mountain View, CA
szegedy@google.com

Sergey Ioffe
sioffe@google.com

Vincent Vanhoucke
vanhoucke@google.com

Alex Alemi
alemi@google.com

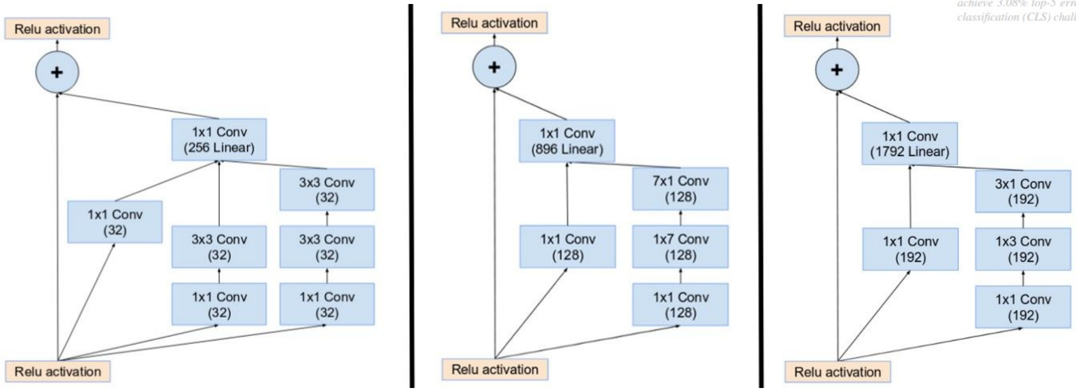
Abstract

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 very 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. There is also some evidence of residual Inception networks outperforming similarly expensive Inception networks without residual connections by a thin margin. We also present several new streamlined architectures for both residual and non-residual Inception networks. These variations improve the single-frame recognition performance on the ILSVRC 2012 classification task significantly. We further demonstrate how proper activation scaling stabilizes the training of very wide residual Inception networks. With an ensemble of three residual and one Inception-v4, we achieve 3.08% top-5 error on the test set of the ImageNet classification (CLS) challenge.

tion [7], object tracking [18], and superresolution [3]. These examples are but a few of all the applications to which deep convolutional networks have been very successfully applied ever since.

In this work we study the combination of the two most recent ideas: Residual connections introduced by He et al. in [5] and the latest revised version of the Inception architecture [15]. In [5], it is argued that residual connections are of inherent importance for training very deep architectures. 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 to reap all the benefits of the residual approach while retaining its 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 than Inception-v3. Historically, Inception-v3 had inherited a lot of the baggage of the earlier incarnations. The technical constraints chiefly came from the need for partitioning the model for distributed training using DistBelief [2]. Now, after migrating our training setup to TensorFlow [1] these constraints have been lifted, which allowed us to simplify the architecture significantly. The details of that simplified architecture are described in Section 3.



Inception ResNet

- The original paper didn't use BatchNorm after summation to train the model on a single GPU (To fit the entire model on a single GPU).
- It was found that Inception-ResNet models were able to achieve higher accuracies at a lower epoch.
-

Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

Christian Szegedy
Google Inc.
1600 Amphitheatre Pkwy, Mountain View, CA
szegedy@google.com

Sergey Ioffe
sioffe@google.com

Vincent Vanhoucke
vanhoucke@google.com

Alex Alemi
alemi@google.com

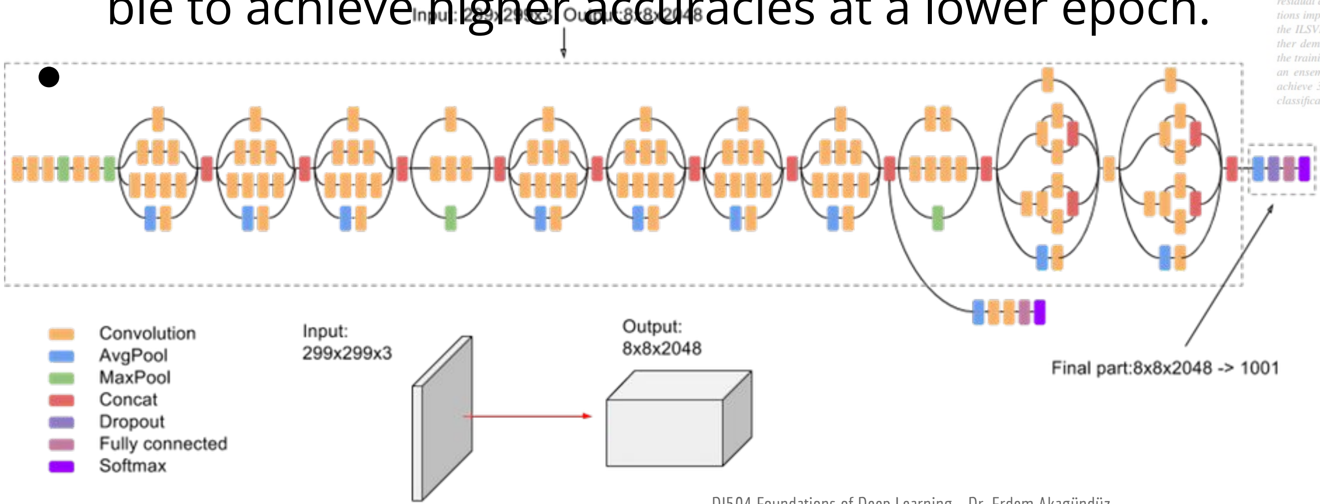
Abstract

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 very 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. There is also some evidence of residual Inception networks outperforming similarly expensive Inception networks without residual connections by a thin margin. We also present several new streamlined architectures for both residual and non-residual Inception networks. These variations improve the single-frame recognition performance on the ILSVRC 2012 classification task significantly. We further demonstrate how proper activation scaling stabilizes the training of very wide residual Inception networks. With an ensemble of three residual and one Inception-v4, we achieve 3.08% top-5 error on the test set of the ImageNet classification (CLS) challenge.

tion [7], object tracking [18], and superresolution [3]. These examples are but a few of all the applications to which deep convolutional networks have been very successfully applied ever since.

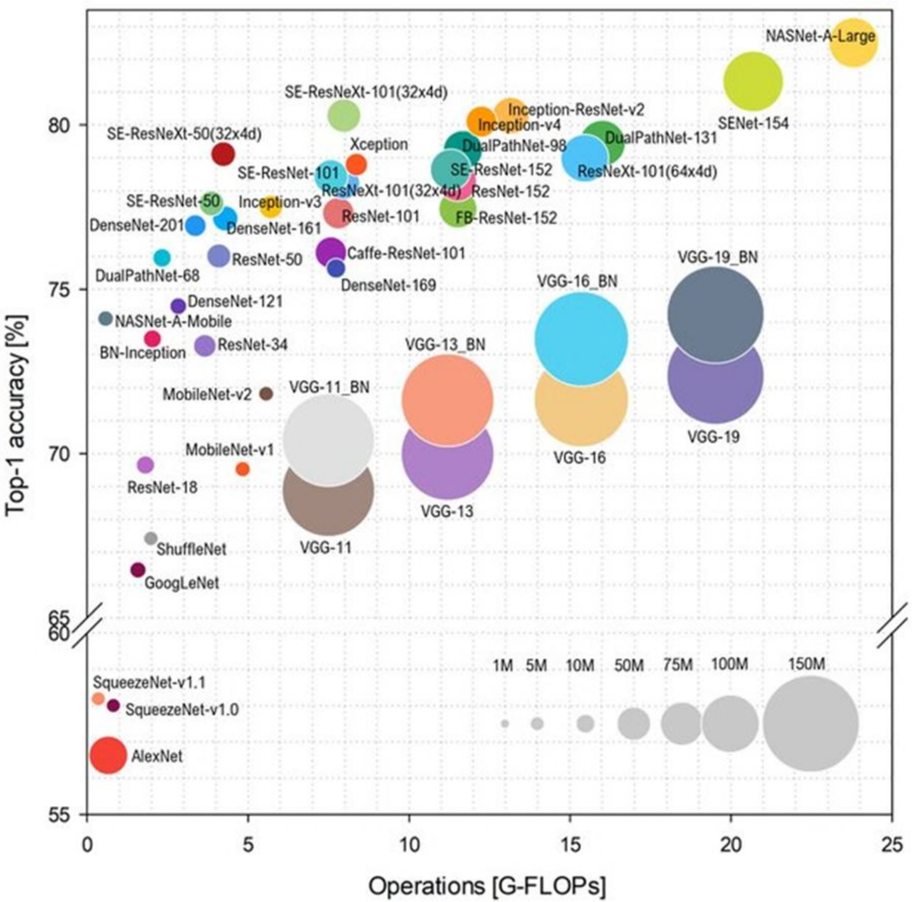
In this work we study the combination of the two most recent ideas: Residual connections introduced by He et al. in [5] and the latest revised version of the Inception architecture [15]. In [5], it is argued that residual connections are of inherent importance for training very deep architectures. Since 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 to reap all the benefits of the residual approach while retaining its 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 than Inception-v3. Historically, Inception-v3 had inherited a lot of the baggage of the earlier incarnations. The technical constraints chiefly came from the need for partitioning the model for distributed training using DistBelief [2]. Now, after migrating our training setup to TensorFlow [1] these constraints have been lifted, which allowed us to simplify the architecture significantly. The details of that simplified architecture are described in Section 3.

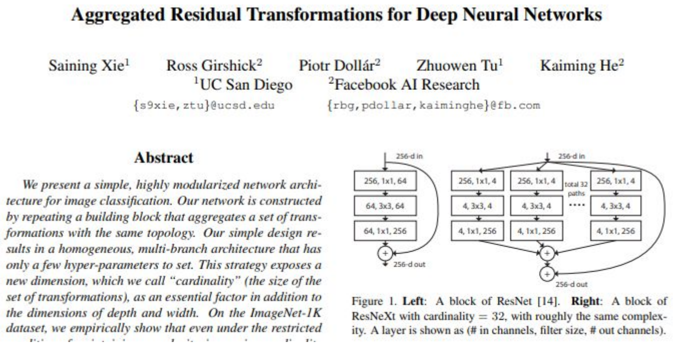


Classification Networks

- There are many classification studies that succeeded AlexNet and VGG.
- All trying to optimize computation, memory and accuracy.



ResNeXt



Aggregated Residual Transformations for Deep Neural Networks

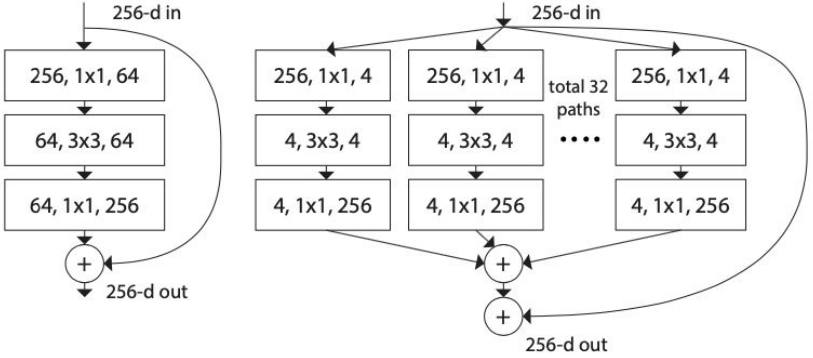
Saining Xie[1] Ross Girshick[2] Piotr Dollár[2] Zhuowen Tu[1] Kaiming He[2]
[1]UC San Diego [2]Facebook AI Research
{s9xie,ztu}@ucsd.edu {rbg,pdollar,kaiminghe}@fb.com

Abstract

We present a simple, highly modularized network architecture for image classification. Our network is constructed by repeating a building block that aggregates a set of transformations with the same topology. Our simple design results in a homogeneous, multi-branch architecture that has only a few hyper-parameters to set. This strategy exposes a new dimension, which we call "cardinality" (the size of the set of transformations), as an essential factor in addition to the dimensions of depth and width. On the ImageNet-1K dataset, we empirically show that even under the restricted

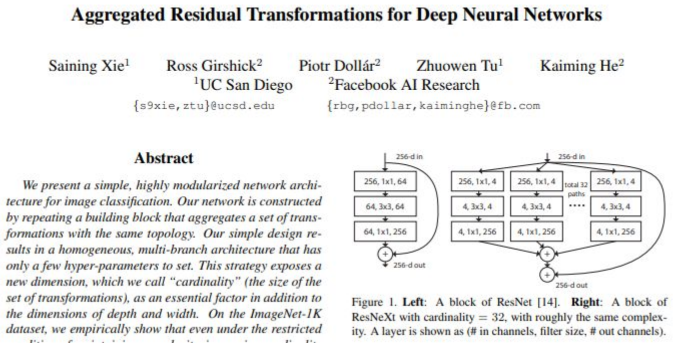
Figure 1. **Left**: A block of ResNet [14]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

- In 2017 same group (including He)
comes with an advanced version: the "ResNeXt"
- This paper introduces the concept of 'cardinality,' an additional dimension to depth and width of a CNN, and shows that aggregating residual blocks with the same topology and hyper parameters is more effective in gaining accuracy than going deeper or wider.



ResNeXt

- In 2017 same group (including He)
  comes with an advanced version: the “ResNeXt”



Aggregated Residual Transformations for Deep Neural Networks

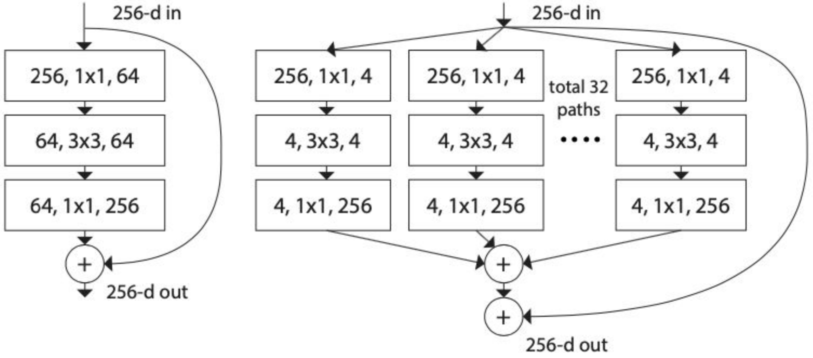
Saining Xie[1] Ross Girshick[2] Piotr Dollár[2] Zhuowen Tu[1] Kaiming He[2]
[1]UC San Diego [2]Facebook AI Research
{s9xie,ztu}@ucsd.edu {rbg,pdollar,kaiminghe}@fb.com

Abstract

We present a simple, highly modularized network architecture for image classification. Our network is constructed by repeating a building block that aggregates a set of transformations with the same topology. Our simple design results in a homogeneous, multi-branch architecture that has only a few hyper-parameters to set. This strategy exposes a new dimension, which we call "cardinality" (the size of the set of transformations), as an essential factor in addition to the dimensions of depth and width. On the ImageNet-1K dataset, we empirically show that even under the restricted

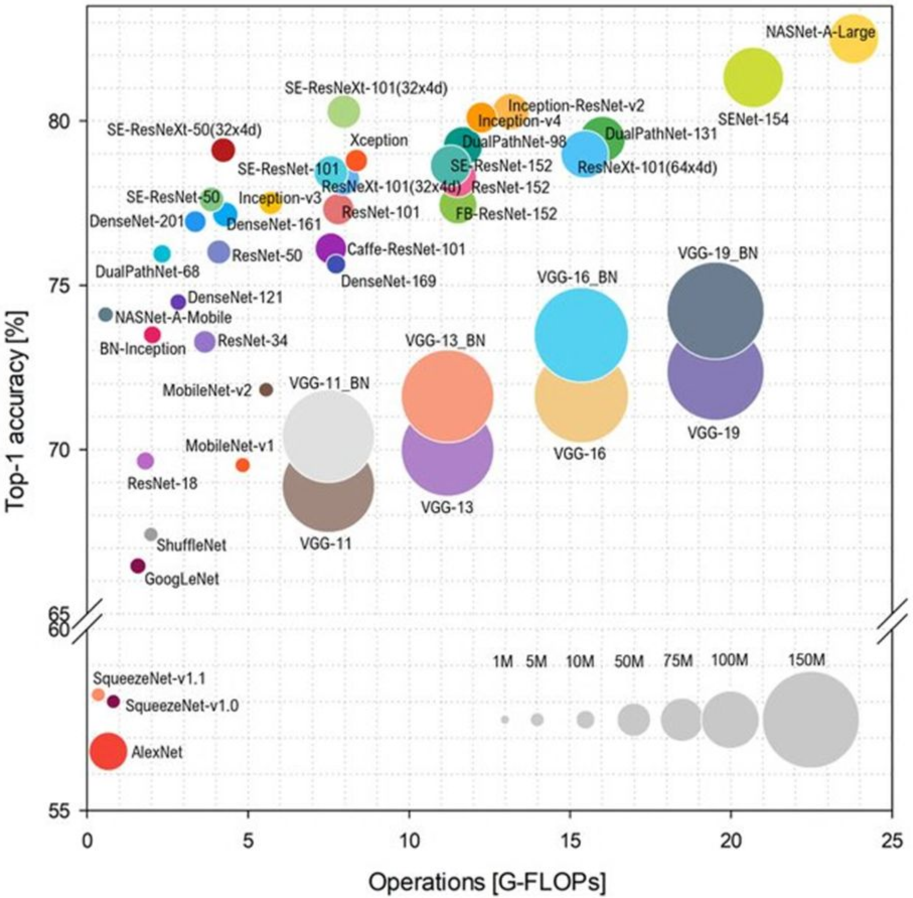
Figure 1. **Left**: A block of ResNet [14]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

- The idea of ResNeXt is inherited from an earlier paper, the one that we mentioned in the previous slides: GoogleNet, or also known as the InceptionNet.



Classification Networks

- There are many classification studies that succeeded AlexNet and VGG.
- All trying to optimize computation, memory and accuracy.



METU

Additional Reading & References

- https://medium.com/@waya.ai/deep-residual-learning-9610bb62c355
- https://paperswithcode.com/method/bottleneck-residual-block
- https://kjo3.medium.com/aggregated-residual-transformation-for-deep-neural-networks-e4c37694cf10
- https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202