Residual Networks Behave Like Ensembles of Relatively Shallow Networks

Andreas Veit  Michael Wilber  Serge Belongie

05/2016 ARXIV, Advances in Neural Information Processing Systems 29 (NIPS 2016)
Overview

• Introduction
• Background
  • Previous investigations on neural networks
  • Deep Residual Networks (ResNets): 10 to 100 layers
  • Importance of Identity Mapping: 100 to 1000 layers
• Key takeaway 1
  • Existing systems are feed-forward, with only one path.
  • ResNets contain many paths instead, shown by the «unraveled view».
• Key takeaway 2
  • Path lengths are binomially distributed.
  • |Gradient| decreases exponentially with increasing path length.
  • Only short paths contribute gradient during training.
• Q & A
Introduction
Sequential vision pipelines influence our thinking.
Existing systems are feed-forward, with only one path
Existing systems are feed-forward, with only one path.

What happens when we delete a step?
Existing systems are feed-forward, with only one path
Existing systems are feed-forward, with only one path.
Existing systems are feed-forward, with only one path

Any alternatives?
Existing systems are feed-forward, with only one path

Any alternatives?
ResNets!

Slide adapted from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
What is the reason behind ResNets’ increased performance?

Hypothesis by He et al. 2016†:
«via a simple but essential concept – going deeper.»

Veit et al. 2016:
A complementary explanation...

†He et al. 2016, "Identity Mappings in Deep Residual Networks"
Background
Previous investigations: What do we know about neural networks?

• Shown by Bengio et al. 1994 and Hochreiter 1991:
  • Length of paths affect magnitude of the gradient during backpropagation.

• Lesion studies on AlexNet by Yosinski et al. 2014:
  • Early layers little co-adaptation: General, applicable to many datasets and tasks
  • Later layers have more co-adaptation: Specific
generality -> specificity

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Deep Residual Networks (ResNets)

• «Deep Residual Learning for Image Recognition». CVPR 2016 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun

• A Simple and clean framework of training very deep nets

• State-of-the-art performance for
  • Image classification
  • Object detection
  • Semantic segmentation
  • and more...

Slide adapted from ICML 2016 Tutorial by Kaiming He
Deep Residual Networks (ResNets)

• «Deep Residual Learning for Image Recognition». CVPR 2016 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun

• A Simple and clean framework of training very deep nets

• State-of-the-art performance for
  • Image classification
  • Object detection
  • Semantic segmentation
  • and more...

Slide adapted from ICML 2016 Tutorial by Kaiming He
ResNets for «training very deep nets»

Revolution of Depth

ImageNet Classification top-5 error (%)

ILSVRC'15 ResNet: 3.57
ILSVRC'14 GoogleNet: 6.7
ILSVRC'14 VGG: 7.3
ILSVRC'13: 11.7
ILSVRC'12 AlexNet: 16.4
ILSVRC'11 shallow: 25.8
ILSVRC'10: 28.2

152 layers
ResNets for achieving «state-of-the-art performance»

ResNets @ ILSVRC & COCO 2015 Competitions

• **1st places in all five main tracks**
  • ImageNet Classification: “Ultra-deep” 152-layer nets
  • ImageNet Detection: 16% better than 2nd
  • ImageNet Localization: 27% better than 2nd
  • COCO Detection: 11% better than 2nd
  • COCO Segmentation: 12% better than 2nd

*improvements are relative numbers*
Deep Residual Learning

Plain Net

\[ x \]

weight layer

\[ \text{relu} \]

weight layer

\[ \text{relu} \]

\[ H(x) \]

H(x) is any desired mapping, hope the 2 weight layers fit H(x)

Slide from ICML 2016 Tutorial by Kaiming He

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Deep Residual Learning

Residual Net

\[ H(x) = F(x) + x \]

- \( H(x) \) is any desired mapping,
- hope the 2 weight layers fit \( H(x) \)
- hope the 2 weight layers fit \( F(x) \)
- let \( H(x) = F(x) + x \)

- \( F(x) \) is a residual mapping w.r.t. identity

Slide from ICML 2016 Tutorial by Kaiming He

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
An issue on learning deep models

- **Optimization ability**
  - Feasibility of finding an optimum
  - Not all models are equally easy to optimize

...(other issues)

How do ResNets address this issue?

- **Optimization ability**
  - Enable very smooth forward/backward prop
  - Greatly ease optimizing **deeper** models

Slide adapted from ICML 2016 Tutorial by Kaiming He
On identity mappings for **optimization**

- shortcut mapping: $h = \text{identity}$
- after-add mapping: $f = \text{ReLU}$

$$x_{l+1} = f(h(x_l) + F(x_l))$$
On identity mappings for **optimization**

- shortcut mapping: \( h = \text{identity} \)
- after-add mapping: \( f = \text{ReLU} \)
- What if \( f = \text{identity} \)?

\[
x_{l+1} = f(h(x_l) + F(x_l))
\]
On identity mappings for **optimization**

- shortcut mapping: $h = \text{identity}$
- after-add mapping: $f = \text{ReLU}$
- What if $f = \text{identity}$?

$$x_{l+1} = f(h(x_l) + F(x_l))$$
Very smooth backward propagation

\[
\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L} (1 + \frac{\partial}{\partial x_l} \sum_{i=1} F(x_i))
\]

- Any \(\frac{\partial E}{\partial x_L}\) is **directly** back-prop to any \(\frac{\partial E}{\partial x_l}\), plus residual.
- Any \(\frac{\partial E}{\partial x_l}\) is **additive**; unlikely to vanish
  - in contrast to **multiplicative**: \(\frac{\partial E}{\partial x_l} = \prod_{i=l}^{L-1} W_i \frac{\partial E}{\partial x_L}\)


Slide from ICML 2016 Tutorial by Kaiming He

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Key takeaways
Key takeaways

Residual networks contain many paths.

Previous networks have a single path.

Only short paths contribute gradient during training.

Vanishing gradient suppresses gradient from long paths.
Key takeaways

- Residual networks contain many paths.
- Previous networks have a single path.
- Only short paths contribute gradient during training.
- Vanishing gradient suppresses gradient from long paths.

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
For example, what happens when we delete layers at test time?

Which layers to drop
For example, what happens when we delete layers at test time?

![Graph showing error vs. random chance for VGG with a notable increase in error marked as Catastrophe.](Slide from NIPS 2016 Spotlight Video)

Which layers to drop
For example, what happens when we delete layers at test time?

VGG (Catastrophe)

Which layers to drop

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
For example, what happens when we delete layers at test time?

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
For example, what happens when we delete layers at test time?

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Why does this happen? The «unraveled view»

VGG

ResNet

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Why does this happen? The «unraveled view»

(a) Conventional 3-block residual network

Unraveled view of (a)
Why does this happen? The «unraveled view»

(a) Conventional 3-block residual network

Unraveled view of (a)

Building block
Skip connection
Residual module

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Why does this happen? The «unraveled view»

The unraveled view is equivalent and showcases the many paths in ResNet.
Deletion of one layer

VGG
All paths are affected

ResNet
Only half of the paths are affected
Performance varies smoothly when deleting several layers.

Error when deleting layers

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Performance varies smoothly when **re-ordering** layers.

Error when permuting layers

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Conclusion 1:

- Residual Networks consist of many paths.
- Although trained jointly, they do not strongly depend on each other: Ensemble-like behavior.
Key takeaways

Residual networks contain many paths.

Previous networks have a single path.

Only short paths contribute gradient during training.

Vanishing gradient suppresses gradient from long paths.
Distribution of path length

There are very few short paths...

Paths length follows a binomial distribution.
Distribution of path length

There are very few short paths...

And very few long paths...
Distribution of path length

There are very few short paths...

And very few long paths...

Most paths are medium length!
Distribution of path length

There are very few short paths...

And very few long paths...

Most paths are medium length!

Paths length follows a binomial distribution.

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Vanishing gradient

The gradient magnitude decreases exponentially with increasing path length.

Slide from NIPS 2016 Spotlight Video

Andreas Veit, Michael Wilber & Serge Belonie. NIPS 2016.
Combining the path length distribution and the vanishing gradients, one can observe that most of the gradient comes from relatively short paths.
Conclusion 2:

- Most paths through a ResNet are relatively short.
- During training, gradients only flow through short paths.
Q & A
References


