One step forward every day

Those who know are the beginning of action, those who act are the completion of knowledge. A gentleman works on the basics, and the way will emerge when the basics are established.

ResNet detailed explanation and analysis

2020-02-25 20:03 shine-lee 86852 3 Edit Favorite Report

Table of contents

Blog: Blog Garden | CSDN | blog

What problem does Resnet solve?

ResNets aims to solve the "degradation" problem of deep neural networks.

What is "degeneration"?

We know that gradually adding layers to a shallow network will improve the performance of the model on the training set and test set, because the model is more complex and has stronger expressive power, and can better fit the underlying mapping relationship. **"Degradation" refers to the situation where the performance drops rapidly after adding more layers to the network.**

The performance degradation on the training set can rule out overfitting, and the introduction of the BN layer also basically solves the gradient vanishing and gradient exploding problems of the plain net. If it is not caused by overfitting and gradient vanishing, what is the reason?

Logically, if we add more layers to the network, the solution space of the shallow network is included in the solution space of the deep network. The solution space of the deep network is at least as good as the solution of the shallow network, because we can get the same performance as the shallow network by simply turning the added layers into identity mappings and copying the weights of other layers to the shallow network. **Why can't we find a better solution when it clearly exists? Why do we find a worse solution?**

Obviously, this is an optimization problem, which reflects that the optimization difficulty of models with similar structures is different, and the increase in difficulty is not linear. The deeper the model, the more difficult it is to optimize.

There are two solutions. **One is to adjust the solution method, such as better initialization, better gradient descent algorithm, etc.; the other is to adjust the model structure to make the model easier to optimize – changing the model structu** ł **ɔe of the error surface.**

announcement

Nickname: shine-lee

Age: 10 years and 8

Followers: 423

Following: 3

+Add Follow

search

Points and rankinç

★ Points – 191805

🏆 Rank – 6300

Essay Category (76

- backbone network
- coding(7)
- Data Mining (1)
- Python 100 Questi
- Windows(2)
- Compiler(1)
- Operating System
- Traditional Compu Vision(8)
- Computer Skills(8
- Machine Learning
- Libraries & Frameworks(8)
- on(
- ion(
- Deep Learning Ba

👍 51 👎 2

The author of ResNet started with the latter and explored a better model structure. The stacked layers are called a block. For a block, the function that can be fitted is$F(x)$, if the desired potential mapping is$H(x)$, **rather than letting$F(x)$Instead of learning the potential mapping directly, it is better to learn the residual$H(x)-x$,Right now$F(x) := H(x)-x$, so the original forward path becomes$F(x)+x$,use$F(x)+x$To fit$H(x)$The author** believes that this may be easier to optimize because **compared to letting$F(x)$Learn to be an identity mapping, let$F(x)$It is easier to learn to be 0 – the latter can be easily achieved by L2 regularization** .$F(x) \rightarrow 0$You can get an identity mapp with no loss in performance.

> Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a **residual mapping** . Formally, denoting the desired underlying mapping as$H(x)$, we let the stacked nonlinear layers fit another mapping of$F(x) := H(x)-x$. The original mapping is recast into$F(x)+x$. We **hypothesize** that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, **if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.**
>
> —— from Deep Residual Learning for Image Recognition

The following problem becomes$F(x)+x$How to design it.

Design of Residual Block

$F(x)+x$The block formed is called **Residual Block** , as shown in the **figure** below. Multiple similar Residual Blocks are connected in series to form ResNet.

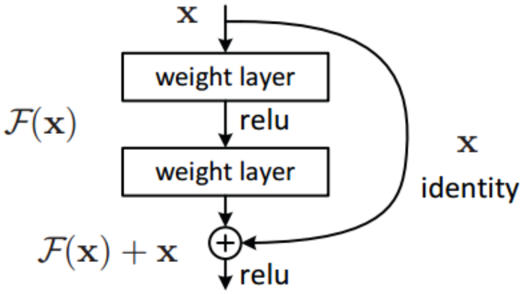

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2. Residual learning: a building block.

A residual block has 2 paths$F(x)$and$x$,$F(x)$The path fitting residual may be called the residual path.x The path is the identity mapping, which is called a "shortcut" .**$\oplus$For element-wise addition, the number of$F(x)$and$x$The size of the two should be the same** . So the question that follows is,

- How to design the residual path?
- How to design shortcut path?
- How to connect Residual Blocks?

In the original paper, the residual path can be roughly divided into two types. One has a **bottleneck** structure, which is the middle right of the figure below.1×1The convolutional layer is used to reduce the dimension first and then increase the dimension. It is mainly for **the practical consideration of reducing the computational complexity** . It is called " **bottleneck block** ". The other type has no bottleneck structure, as shown in the left figure below, and is called " **basic block** ". The basic block consists of two3×3The convolutional layer is composed of the bottleneck block.1×1

Data Structure an
Algorithm(3)
More

Essay Archives (76)

August 2020(1)
July 2020(4)
May 2020(1)
April 2020(1)
February 2020(5)
January 2020(1)
December 2019(4)
November 2019(4)
October 2019(4)
September 2019(3
August 2019(1)
May 2019(3)
April 2019(4)
February 2019(1)
January 2019(3)
More

Reading Rankings

1. Convolutional N
etwork: Convolutiona
ation, Function and I
8719)
2. Understanding
e transformation anc
nsformation matrix (
3. ResNet detailec
ation and analysis (8
4. VSCode Python
pment environment c
ation (66432)
5. Group Convolut
pthwise Convolution
obal Depthwise Conv
(62425)

rar
der
radients, partial deri

👍 51 👎 2

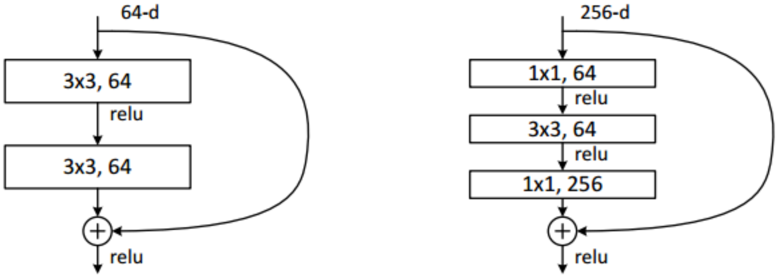
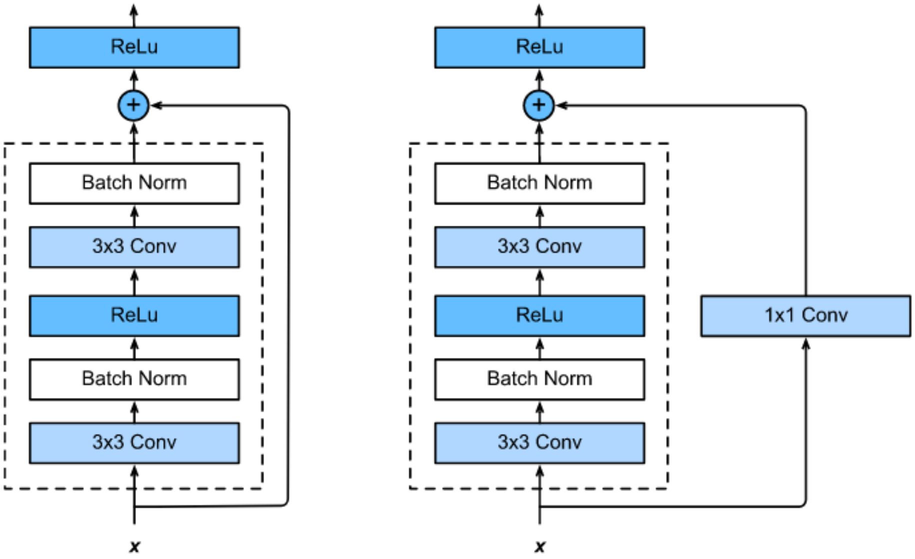


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

The shortcut path can be roughly divided into two types, depending on whether the residual path changes the number and size of feature maps.x**The other one needs to be** outputted intact. 1×1 **Convolution is used to increase the dimension or/and downsample** . Its main function is **to convert the output$F(x)$The output of the path maintains the same shape** , and the improvement of network performance is not obvious. The two structures are shown in the figure below.



As for the connection between Residual Blocks, in the original paper,$F(x)+x$go through$ReLU$Then directly use it as the input of the next block$x$.

for$F(x)$The connection between paths, shortcut paths, and blocks is further studied in the paper Identity Mappings in Deep Residual Networks , which is discussed in detail later in the article.

ResNet network structure

ResNet is a series of multiple Residual Blocks. Let's take a look at the comparison between ResNet-34 and 34-layer plain net and VGG, as well as the different ResNets obtained by stacking different numbers of Residual Blocks.

directional derivative
al vectors, etc. (63)

2. Understanding
e transformation anc
nsformation matrix (

3. Detailed explan
nd analysis of ResNe

4. Convolutional N
etwork: Convolutiona
ation, Function and I
2)

5. Why do we nee
e normalization/stan
tion? (41)

Latest Comments

1. Re: Understand
affine transformatior
its transformation m

Then multiply it by
uniform scaling to g
similarity transform
Is the matrix writter
wrong? s only need
be multiplied to the
diagonal, right?

--Ukn

2. Re: Summary c
pixel numerical extre
value detection algoi

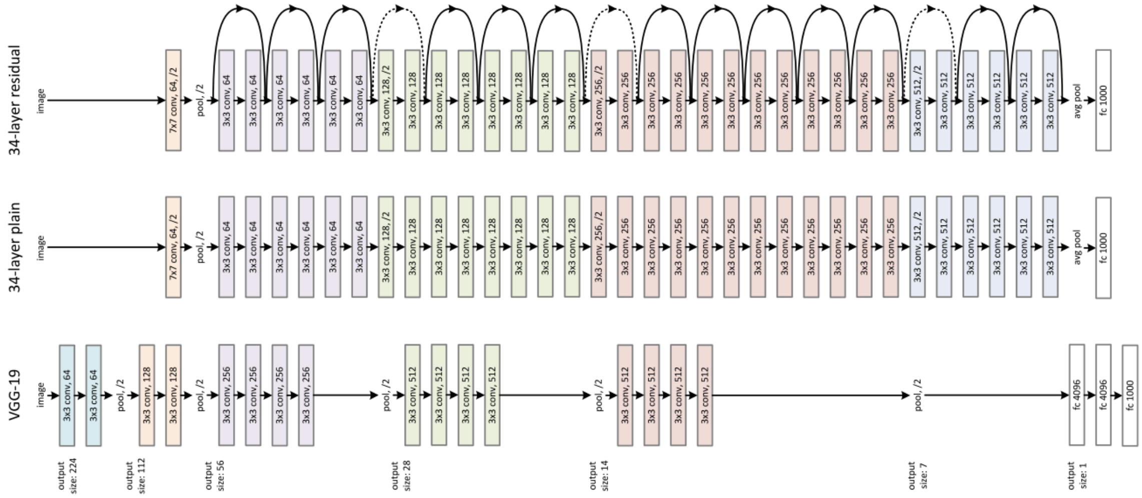
My supervisor aske
to read this paper.
Thanks to the blogg
helped me a lot.

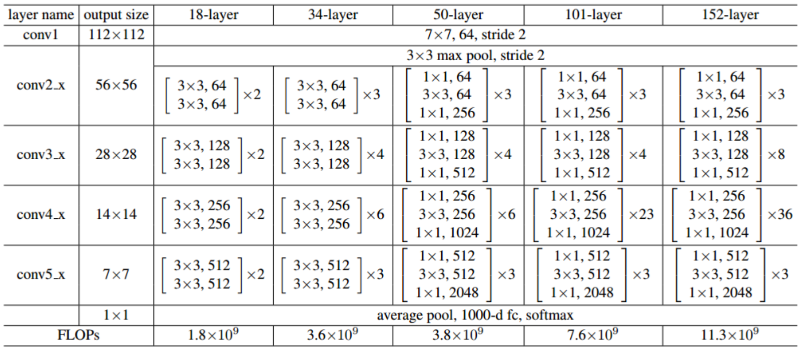
-

3. Re: Thoroughly
understand the mear
calculation of recept

51 2

be
something wrong w





layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1. Architectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked. Downsampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

The design of ResNet has the following characteristics:

- Compared with plain net, ResNet has many more "bypasses", i.e. shortcut paths, and the layers circled at the beginning and end constitute a Residual Block;
- In ResNet, all Residual Blocks have no pooling layer, and **downsampling is achieved through the stride of conv** ;
- In the conv3_1, conv4_1 and conv5_1 Residual Blocks, the sampling is downsampled by 1 times, and the number of feature maps is increased by 1 times, as shown in the blocks marked by dotted lines in the figure;
- **The final features are obtained through Average Pooling** instead of through the fully connected layer;
- Each convolutional layer is followed by a BatchNorm layer, which is not marked in the figure for simplicity;

The ResNet structure is very easy to modify and expand. By adjusting the number of channels in a block and the number of stacked blocks, the width and depth of the network can be easily adjusted to obtain networks with different expressive capabilities without worrying too much about the "degeneration" of the network. As long as there is enough training data and the network is gradually deepened, better performance can be achieved.

The following is a comparison of network performance.

calculation of the receptive field cent we always perform convolution with k= s=2, p=1, (k−1)/2−p always 0, and the receptive field cent obtained according formula is always tl same, but this is obviously wrong. Fc example: with the p (4, 4) of the first la the center, convolu

--The sett

4. Re: Why do we to do feature normalization/standa

It's really well writt specially registered account to like it, b turns out that "new registered users ca like it after 1 day"

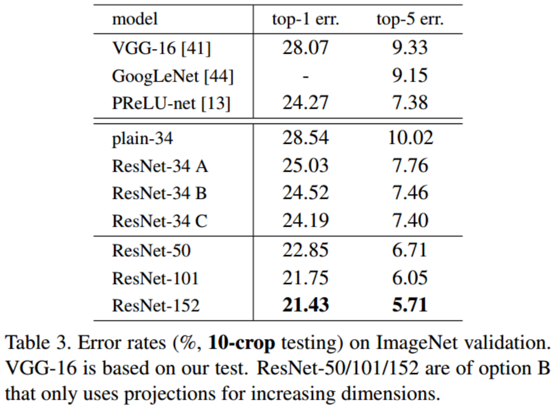
--

5. Re: Batch Normalization

@blackcat_cop Of c it didn't work...

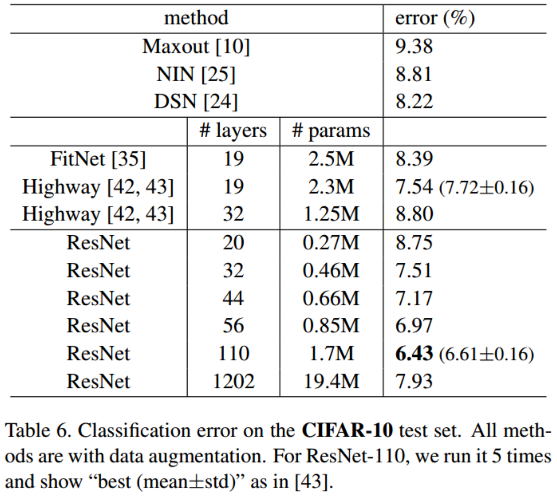
--Xia

51 2



model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. Error rates (%, **10-crop** testing) on ImageNet validation. VGG-16 is based on our test. ResNet-50/101/152 are of option B that only uses projections for increasing dimensions.

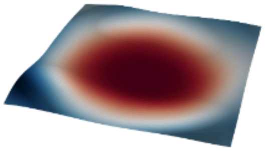
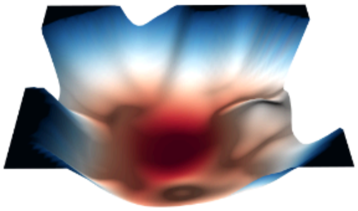
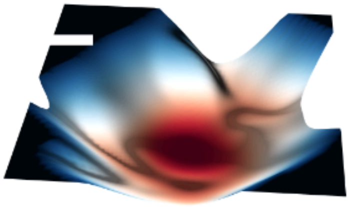
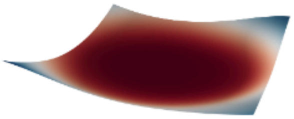
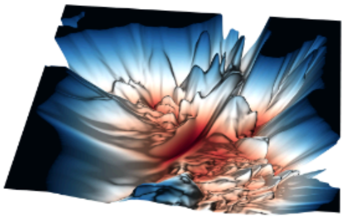
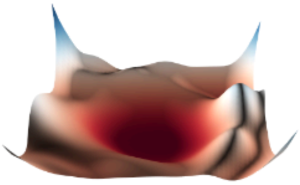


method			error (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [35]	19	2.5M	8.39
Highway [42, 43]	19	2.3M	7.54 (7.72±0.16)
Highway [42, 43]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

Table 6. Classification error on the **CIFAR-10** test set. All methods are with data augmentation. For ResNet-110, we run it 5 times and show "best (mean±std)" as in [43].

Error surface comparison

The above experiment shows that increasing the depth of ResNet to more than 1000 layers does not cause "degradation", which shows the effectiveness of Residual Block. The motivation of ResNet is **that fitting the residual is easier to optimize than directly fitting the potential mapping** . The following is an intuitive feeling of the role of the shortcut path by drawing the error surface. The picture is taken from Loss Visualization .

VGG	ResNet-20 (no short)	ResNet-20 (short)
		
DenseNet-121	ResNet-56 (no short)	ResNet-56 (short)
		

It can be found that:

- The error surface of the shallow plain net of ResNet-20 (no short) is not very complex and optimization is difficult, but the complexity increases greatly after increasing to 56 layers. **For the plain net, the error surface rapidly "deteriorates" with the increase of depth** ;
- After the shortcut is introduced, **the error surfa** **other, the gradient becomes more predictable, and it is obviously easier to c**

Analysis and Improvement of Residual Block

The paper Identity Mappings in Deep Residual Networks further studies ResNet. Through theoretical analysis of ResNet back propagation and adjustment of the structure of Residual Block, a new structure is obtained as follows

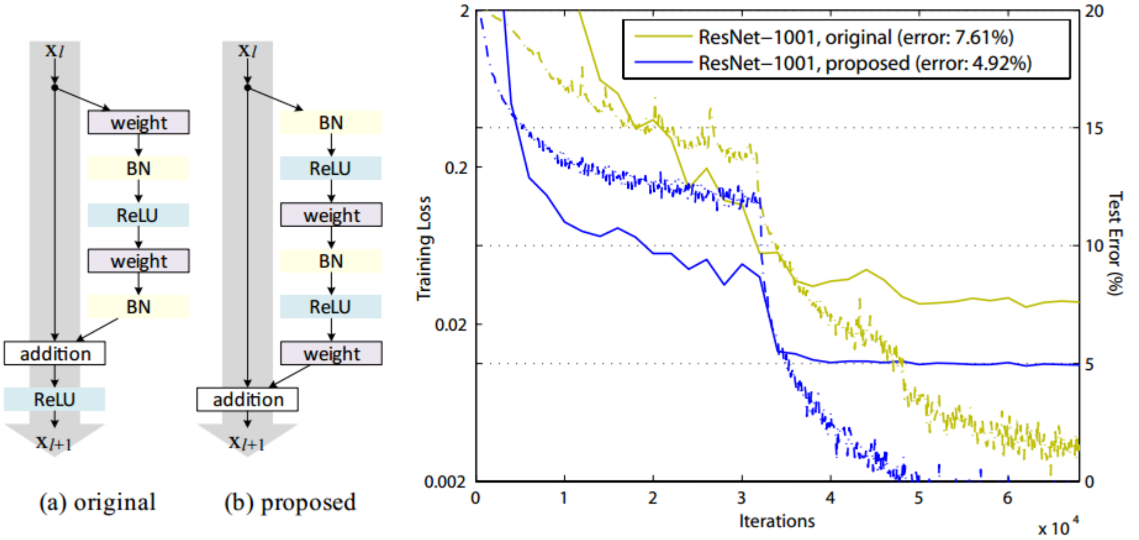


Figure 1. Left: (a) original Residual Unit in [1]; (b) proposed Residual Unit. The grey arrows indicate the easiest paths for the information to propagate, corresponding to the additive term "$\mathbf{x}_l$" in Eqn.(4) (forward propagation) and the additive term "1" in Eqn.(5) (backward propagation). **Right**: training curves on CIFAR-10 of **1001-layer** ResNets. Solid lines denote test error (y-axis on the right), and dashed lines denote training loss (y-axis on the left). The proposed unit makes ResNet-1001 easier to train.

Note that the perspective here is different from the previous one. Here, the shortcut path is regarded as the main path, and the residual path is regarded as the bypass.

The newly proposed Residual Block structure has stronger generalization ability and can better avoid "degradation". After stacking more than 1,000 layers, the performance is still getting better. The specific changes are:

- **It is very important to keep the shortcut path "clean" so that information can be smoothly transmitted in forward and backward propagation.** For this reason, if it is not necessary, do not introduce1×1Convolution and other operations, while moving the ReLU on the gray path in the above figure to$F(x)$On the path.
- On the residual path, **BN and ReLU are uniformly placed before weight as pre-activation** , achieving the effects of "Ease of optimization" and "Reducing overfitting".

The following is a detailed explanation.

make$h(x_l)$is the transformation on the shortcut path,fThis is the transformation after addition. In the original Residual Block$f = ReLU$, **when$h$and$f$When they are all identical mappings, we can get any two layers$x_L$andx_lThe relationship between the two, at this time the information can be$x_l$andx_L Lossless direct transmission, as shown in the following forward propagationx_lAnd in back propagation1.**

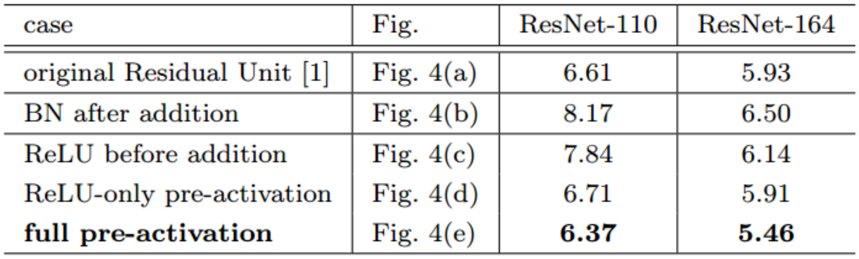
$$
\begin{gathered}
\mathbf{y}_l = h\left(\mathbf{x}_l\right) + \mathcal{F}\left(\mathbf{x}_l, \mathcal{W}_l\right) \\
\mathbf{x}_{l+1} = f\left(\mathbf{y}_l\right) \\
\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}\left(\mathbf{x}_l, \mathcal{W}_l\right) \\
\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}\left(\mathbf{x}_i, \mathcal{W}_i\right) \\
\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} \mathcal{F}\left(\mathbf{x}_i, \mathcal{W}_i\right)\right)
\end{gathered}
$$

This in back propagation1It has a good property **that the back propagation between any two layers is**1, which **can effectively avoid gradient vanishing and gradient exploding.** $h$and$f$If it is not an identity mapping, this item will become complicated. If it is set to a scale factor greater than or less than 1, the gradient may explode or vanish after back propagation. The more layers there are, the more obvious it is. This is why ResNet performs better than highway network. **It should be noted that the BN layer solves the gradient vanishing and exploding of plain net. The 1 here can avoid the gradient vanishing and exploding on the short cut path.**

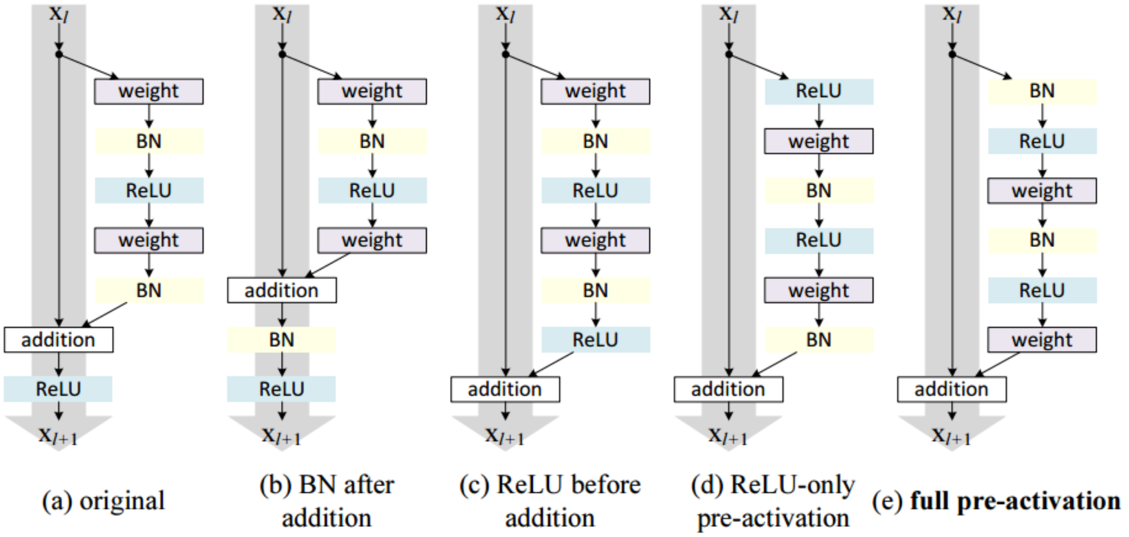
The shortcut path changes the back propagation from a multiplication form to an addition form , so that the final loss of the network can reach each block without loss during the back propagation, which also means that the weight update of each block is partially directly affected by the final loss. Looking at the formula of forward propagation above, we can see a certain **ensemble** form. Although information can be directly transmitted between any two layers, this direct transmission is actually implicit. For a certain block, it can only see the result of addition, but does not know whether each addend in the addition is a majority. **From the perspective of information path, it is not yet complete - thus, DenseNet was born** .

To improve the residual path, the author conducted different comparative experiments and finally obtained **a full pre-activation structure that puts BN and ReLU before the weight** .

Table 2. Classification error (%) on the CIFAR-10 test set using different activation functions.



case	Fig.	ResNet-110	ResNet-164
original Residual Unit [1]	Fig. 4(a)	6.61	5.93
BN after addition	Fig. 4(b)	8.17	6.50
ReLU before addition	Fig. 4(c)	7.84	6.14
ReLU-only pre-activation	Fig. 4(d)	6.71	5.91
full pre-activation	Fig. 4(e)	**6.37**	**5.46**



(a) original

(b) BN after addition

(c) ReLU before addition

(d) ReLU-only pre-activation

(e) **full pre-activation**

Figure 4. Various usages of activation in Table 2. All these units consist of the same components — only the orders are different.

summary

The motivation of ResNet is to solve the "degradation" problem. The design of residual blocks makes it easy to learn the identity mapping. Even if too many blocks are stacked, ResNet can make redundant blocks learn the identity mapping without performance degradation. Therefore, the "actual depth" of the network is determined during the training process, that is, ResNet has a certain **depth adaptation** capability.

Deep adaptation can explain why there is no "degradation", but why can it be better?

By visualizing the error surface, we can see the smoothing effect of the shortcut, but this is only a result. What is the root cause behind it?

Perhaps further research is needed to fully understand ResNet, but there are many different perspectives on it.

- A Proposal on Machine Learning via Dynamical Systems from the perspective of differential equations
- From the perspective of ensemble, Residual Networks Behave Like Ensembles of Relatively Shallow Networks
- From the perspective of information/gradient pathways, Identity Mappings in Deep Residual Networks
- Analogy to Taylor expansion, analogy to wavelet...

By trying to explain from different perspectives, we can gain a deeper and more comprehensive understanding of ResNet. Due to space limitations, this article will not expand on this.

PS: Actually, the author has not yet sorted out a (lea (escape

refer to

- paper: Deep Residual Learning for Image Recognition
- paper: Identity Mappings in Deep Residual Networks
- Loss Visualization
- blog: ResNet, torchvision, bottlenecks, and layers not as they seem
- code: pytorch-resnet
- Residual Networks (ResNet)
- code: resnet-1k-layers/resnet-pre-act.lua

Category: backbone network

♡ Follow me ☆ Save this article WeChat Share

« Previous: Understanding the memory layout and design philosophy of ndarray in numpy
» Next: Easy to understand DenseNet

Members' power lights up the hope of the garden Refresh the page to return to the top

Log in to view or post comments, log in now or visit the blog homepage

少年白马醉春风·AI江湖创作
Create
阿里云X优酷
AI江湖创作大赛
AI生图赢万元现金大奖、耳机、键盘及周边等千份好礼

Editor's recommendation:

· Use old things - transform the set-top box into a Linux development machine!
· This is the hardest part of DDD modeling (actually very simple)
· In order to implement DDD, I "PUA" everyone in this way
· The mysterious Arco style appears, and Webpack is used to solve unexpected reference problems
· An example of how to use the experimental feature Interceptor of c#12

Reading ranking:

· This is the most difficult part of DDD modeling (a
· Reuse of old things - transforming the set-top box into a Linux development machine!

👍 51 👎 2

- · Sweeping the chaos of Hongmeng pop-up windows, SmartDialog was born
- · In order to implement DDD for Javaers, we have to write open source components
- · Detailed tutorial from entry to actual combat of .NET 8 operation of SQLite in seven days (selection, development, release, deployment)

👍 51 👎 2