Pages / … / Current Research Topics

Deep Residual Networks

Created by Brenner, Michael, last modified on 07.February 2017

Author: Michael Brenner

Introduction

In recent years Deep Convolutional Neural Networks (CNN) demonstrated a high performance on image classification tasks. Experiments showed that the number of layers (depth) in a CNN is correlated to the performance in image recognition tasks. This led to the idea that deeper networks should perform better. Creating deep networks is not as simple as adding layers. One problem is the vanishing/exploding gradients, which hamper the convergence. This obstacle can be overcome by normalized initialization and intermediate normalization layers, so that networks start converging for stochastic gradient descend (SGD) using the backpropagation algorithm. Another problem is the degradation, if the depth of a network increases, the accuracy gets saturated and then degrades rapidly. (Figure 1) A way to counter the degradation problem is using residual learning. (1)

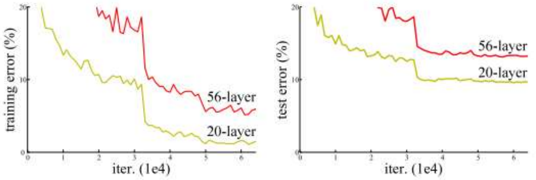


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Deep Residual Learning

Residual Learning

It is possible to fit an desired underlying mapping $H(x)$by a few stacked nonlinear layers, so they can also fit an another underlying mapping $F(x) = H(x) - x$. As a result, it is possible to reformulate it to $H(x) = F(x) + x$, which consists of the Residual Function $F(x)$ and input x. The connection of the input to the output is called a skipt connection or identity mapping. The general idea is that if multiple nonlinear layers can approximate the complicated function $H(x)$, then it is possible for them to approximate the residual function $F(x)$. Therefore the stacked layers are not used to fit $H(x)$, instead these layers approximate the residual function $F(x)$. Both forms should be able to fit the underlying mapping.

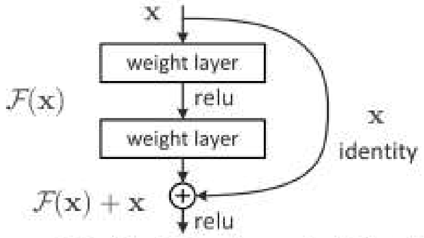

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2. Residual learning: a building block.

One reason for the degradation problem could be the difficulties in approximating identity mappings by nonlinear layers. The reformulation used identity mapping as a reference and let the residual function represent the perturbations. The identity mapping can be generated by the solver through driving the weights of the residual function to zero if need be. (1)

Implementation

Residual learning is implented to every few stacked layers. Figure 2 shows an example of 2 layers. As an example, formulation (1) can be defined as:

$$F(x) = W_2\sigma(W_1x) + x \tag{1}$$

Where W_1 and W_2 are the weights for the convolutinoal layers and σ is the activation function, in this case a RELU function. The operation $F + x$ is realized by a shortcut connection and element-wise addition. The addition is followed by an activation function σ.

The resulting formulation for a residual block is:

$$y(x) = \sigma(W_2\sigma(W_1x) + x) \ . \tag{2}$$

After each convolution (weight) layer a batch normalization method (BN) is adopted. The training of the network is achiebed by stochastic gradient descent (SGD) with a mini-batch size of 256. The learning rate starts from 0.1 and is divided by 10 when the error plateaus. The weight decay rate is 0.0001 and has a value of 0.9. (1)

Network Architecures

Plain Networks

The plain networks are adopted from the VGG nets (Figure 3(left)). The convolutional layers have mostly 3x3 filters and the design follows two rules:
1. For the same output feature map size, the layers have the same number of filters, and

2. if the feature map size is halved, the number of filters is doubled in order to preserve the time complexity per layer.

The downsampling operation is performed by the convolutional layers that have a stride of 2, hence no pooling layers. The network ends with a global average pooling layer and a 1000-way fully connected layer with softmax function.

Figure 3 (middle) shows a plain model with 34 layers. (1)

Residual Network

To convert the plain model to the residual version, shortcut connections are added, as demonstrated in the figure 3 (right). The solid line shortcuts are identity mapping. When the dimensions increases there are 2 options (dotted line shortcut):

1. The shortcut still performs identity mapping with zero padding to increasing the dimensions or

2. the shortcut is used to match dimensions utilizing 1x1 convolution.

In both options, when the shortcut go across feature maps of different sizes, they used a stride of 2. Generally the second option is used.(1)

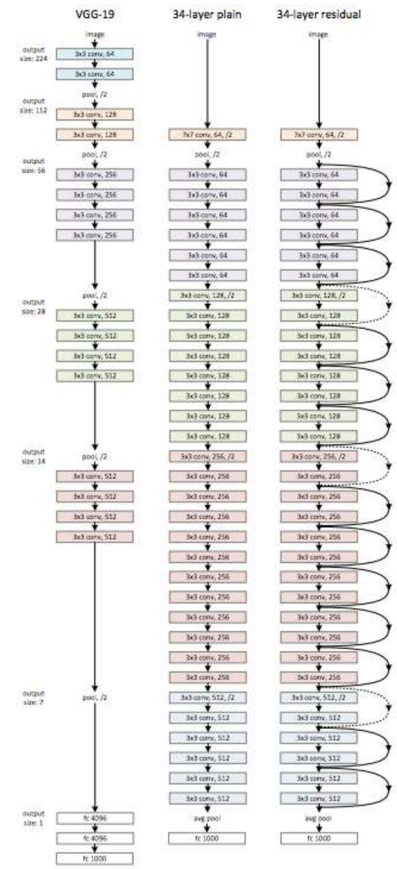


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model as a reference. **Middle**: a plain network with 34 layers. **Right**: a residual network with 34 layers. The dotted shortcuts increases dimensions

Results

The residual and plain networks are compared on the ImageNet 2012 classification dataset that consist of 1000 classes. All Networks are trained on 1.28 million training images and evaluated on the 50k validation images. The final result was obtained on the 100k test images.

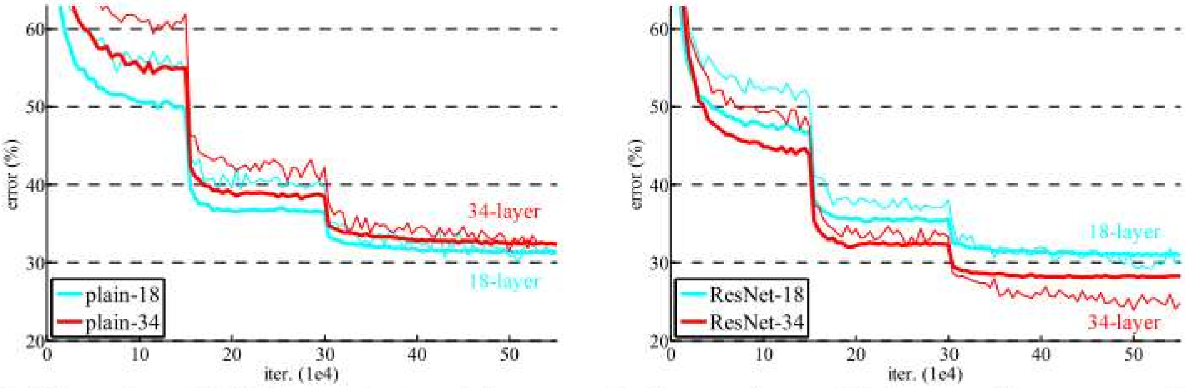


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

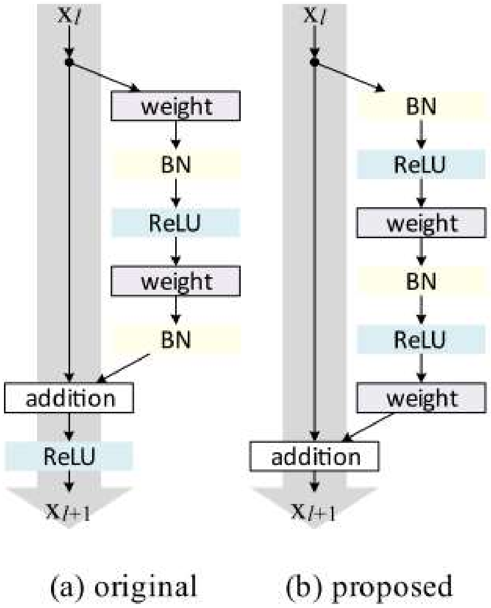
The evaluation of the plain models showed that the 34 layer network has a higher training error than the 18 layer model. (Figure 4 left) The reason behind this result is the degradation problem.

The residual models show a contradictiory result. The deeper ResNet with 34 layers has a smaller training error than the 18 layer (Figure 4 right). This result proves that the degradation problem can be addressed with residual learning and that an increased network depth results in a gain accuracy.(1)

Optimization

The original design of the residual block (Figure 1) can be represented in a more detailed way (Figure 5 a). A proposed optimization is shown in Figure 5 b. The proposed design consist of a direct path for the propagating information through the residual block as a result, through the entire network. This allows the signal to propagate from one block to any other block, during both forward and backward passes. The complexity of the training also becomes simpler with the new block design.

The original Residual block is described as $y(x) = \sigma(x + F(x))$. The new design change the representation to $y(x) = x + F(x)$, because it is important to have a "clean" path from one block to the next one. The removal of the ReLU function on the main path and the different design of the Residual function $F(x)$ is related (Figure 6).



(a) original (b) proposed

Figure 5. (a) original Residual Unit design; (b) proposed Residual Unit. The grey arrows indicate the easiest paths for the information to propagate.

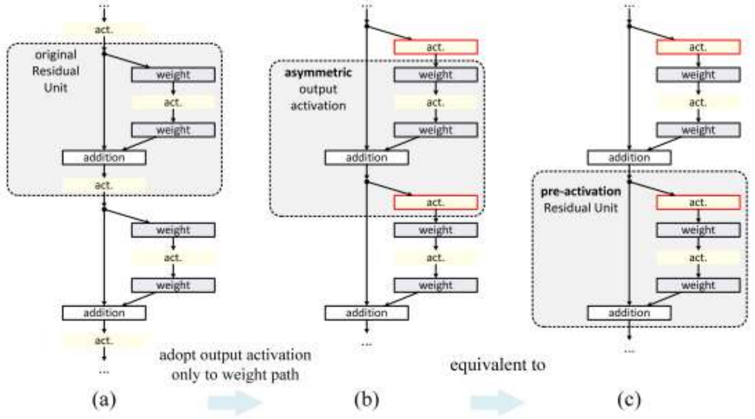
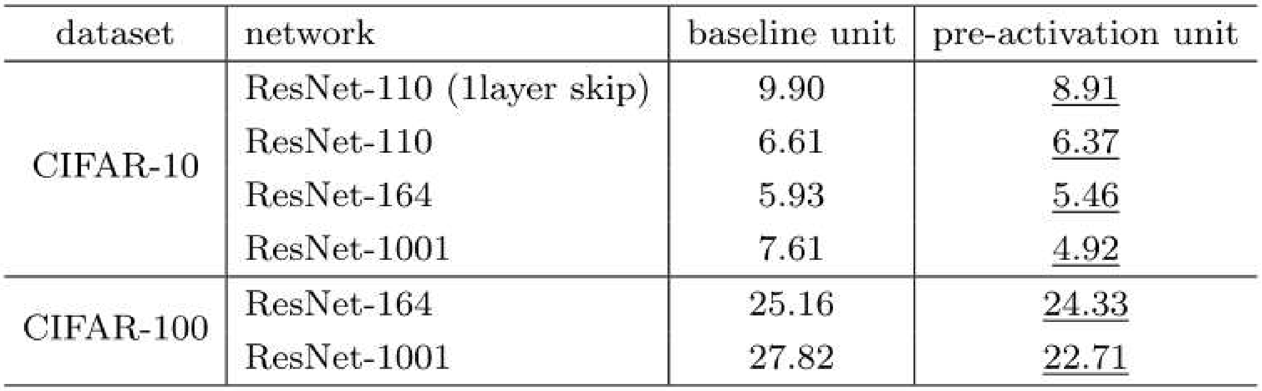


Figure 6. Using asymmetric after-addition activation is equivalent to constructing a *pre-activation* Residual Unit.

Figure 6 shows that the activation function from the main path is moved to the the residual function of the next block. This means that the activation functions (ReLU) are now a "pre-activation" function of the weight layers. Experiments showed that the ReLU-only pre-activation performs similiarly to the original design. By adding BN to the pre-activation, the result can be improved by a healty margin. This "pre-activation" model shows consistently better results then original counterpart (Table 1) and the computational complexity is linear to the depth of the Network. (2)



dataset	network	baseline unit	pre-activation unit
CIFAR-10	ResNet-110 (1layer skip)	9.90	8.91
	ResNet-110	6.61	6.37
	ResNet-164	5.93	5.46
	ResNet-1001	7.61	4.92
CIFAR-100	ResNet-164	25.16	24.33
	ResNet-1001	27.82	22.71

Table 1: Classification error (%) on the CIFAR-10/100 test set using the original Residual units and the pre-activation Residual units.

Literature

1. **He, Kaiming, et al. "Deep residual learning for image recognition."** ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*****. 2016.**

2. **He, Kaiming, et al. "Identity mappings in deep residual networks."** ***European Conference on Computer Vision*****. Springer International Publishing, 2016.**

Weblinks

Links to related/additional Content in the Web

No labels

1 Comment

Aykin, Can

General problems and suggestions:

- Avoid using subjective pronouns such as "we", "you" etc., use passive versions of the sentences.
- Do not use short forms such as "isn't", don't or "cant't", use long forms such as "is not", "do not" and "can not" instead.
- Add image sources to your bibliography/weblinks.
- You might want to add the used databases to your bibliography as well (CIFAR 10/100 etc.)
- Linking to the other wiki pages for things like "RELU Function" or "nonlinearity layer" is generally a good idea.
- You might want to add anchors to the images and when you refer to them in your text you can add a link to them.

Consistency problems and suggestions:

- The general page suffers from a grammatical inconsistency where you kept switching between past and present tense. I would recommend using present tense since this is a scientific wiki page but regardless of the choice, I would recommend keeping it consistent through the article.
- Some of your images are numbered and described, where others do not even have any annotation under them.
- You refer to one of the unannotated images as "Figure above" which is also inconsistent with the rest of your referrals.
- Your capitalization is inconsistent throughout the article. (for example either say "Residual Function" or "residual function" but do not alternate between them, same applies to "Figure x" and "figure x").
- Your citation enumeration style is inconsistent with the rest of the wiki (we decided to use (1) style for enumerations).
- Your bibliography style is inconsistent with the rest of the wiki.

Visual problems and suggestions:

- Remember to check and eliminate double spaces and underspacing.
- There are some minor paragraph spacing problems (for example the double spacing between first and second paragraphs in section Deep Residual Learning).
- I would recommend using numbered lists for options in Network Architectures section. It should fix the improper line spacing in that section.
- I do not think you need to write the author and date since they are already available under page title.
- The non-mathematical counts such as "2 options" should preferably be in written form ("two options"). (same applies to "option 2" which would be "second option" but these two are not the only ones)
- For all mathematical numbering I would recommend using math inlines (for example "1x1 convolutions" written as "1×1 convolutions")
- For better readability balancing the length of paragraphs might be a good idea (i.e., splitting long paragraphs and merging short ones)

Corrections (note that some of the corrections have more than one changes):

- In the last years
- In recent years / In the last few years
- demonstrated to have a high performance
- demonstrated a high performance
- on image classification task
- on image classification tasks
- related to the result
- correlated to the performance
- number of layers (depth)
- number of layers (depth) in a CNN
- image recognition
- image recognition tasks
- This led to the idea, that
- This led to the idea that
- isn’t as easy as
- is not as simple as
- This obstacle can be solved by
- This obstacle can be overcome by
- with backpropagation
- using the backpropagation algorithm
- Another problem is the degradation,
- Another problem is the degradation;
- and then degrades rapidly. (Figure 1)
- prior to degrading rapidly (Figure 1).
- is by using residual learning.

- is using residual learning.
- Let us consider that we can fit an desired underlying mapping
- It is possible to fit a desired underlying mapping
- layers. So they can also fit an another underlying mapping
- layers, so they can also fit another underlying mapping
- Then you can reformulate it to
- As a result, it is possible to reformulate it to
- which consist of
- which consists of
- of the Residual Function F(x) and x which represent the input
- of the Residual Function F(x) and input x
- input, we call this skip connection, identity mapping
- input x. The input x in this representation is called a skip connection or identity mapping.
- The general idea is, that multiple nonlinear layers can approximate the complicate function H(x), then you can also say they can approximate the residual function F(x).
- The general idea is that if multiple nonlinearity layers can approximate the complicated function H(x), then it is possible for them to approximate the residual function F(x).
- So we don’t use stacked layers to fit H(x), we let these layers approximate the residual function.
- Therefore the stacked layers are not used to fit H(x), instead these layers are utilized to approximate the residual function.
- difficulties in approximating identity mappings by nonlinear layers.
- the difficulties in approximating identity mappings by nonlinearity layers.
- By the reformulation the identity mapping is optimal and is used as reference.
- The reformulation provides an optimal identity mapping which is used as reference.
- So it is easier to find
- As a result it is easier to find
- with the residual function
- within the residual function
- and if it is asked for the identity mapping than the solver drive the weights of the residual function to zero.
- and the identity mapping can be generated by the solver through driving the weights of the residual function to zero if need be.
- The formulation (1) can be defined in for this example to
- As an example, formulation (1) can be defined as
- W1 and W2 are the weights
- Where W1 and W2 are the weights
- convolutinoal layers
- convolutional layers
- in this case RELU function.
- in this case a RELU function.
- After the addition follows an activation function σ.
- The addition is followed by an activation function σ.
- So the formulation for an residual block is
- The resulting formulation for a residual block is
- layer we adopt batch normalization (BN).
- layer a batch normalization method (BN) is adopted.
- For the training of the network we use stochastic gradient descent (SGD) with a mini-batch size of 256.
- The training of the network is achieved by stochastic gradient descent (SGD) method with a mini-batch size of 256.
- The weight decay is 0.0001 and a momentum of 0.9.
- The weight decay rate is 0.0001 and has a momentum of 0.9. (or value)
- Residual networks are evaluated and compared to plain Networks.
- (This sentence is grammatically correct but seems extremely out of place when starting a new section, I would recommend removing it or converting it to a small introductory paragraph to provide context)
- adopted of the VGG nets
- adopted from the VGG nets
- mostly 3x3 Filter
- mostly 3x3 filters
- doubled so as to preserve
- doubled in order to preserve
- Downsampling is performed by
- The downsampling operation is performed by
- 2, so no pooling.
- 2, hence no pooling layers.
- fully connected layer with softmax.
- fully connected layer with softmax function.
- To turn the plain model

- To convert the plain model
- version, they added shortcut connections. As you can see in the figure 3 (right).
- version, shortcut connections are added, as demonstrated in the figure 3 (right).
- identity mapping, with extra zero entries padded for increasing dimensions or
- identity mapping with zero padding to increase the dimensions or
- dimensions, by 1x1 convolution.
- dimensions utilizing 1x1 convolution.
- when the shortcut go across feature maps of different sizes, they used a stride of 2.
- when the shortcutf go across the feature maps of different sizes, a stride of 2 is chosen.
- Normally option 2 is used.
- Generally the second option is used.
- networks were evaluated
- networks are evaluated
- trained on the 1.28 million training images
- trained on 1.28 million training images
- evaluated on the 50k validation images.
- evaluated on 50 thousand validation images. (also; I would have used "validated" instead of "evaluated" since evaluate is used in the previous sentence in a different meaning, another possibility is to change the first sentence to "compared on".)
- was obtained on the 100k test images.
- is obtained on 100 thousand test images.
- that the 34 layer has
- that the 34 layer network has
- a higher training error, then the 18 layer model. (Figure 4 left)
- a higher training error than the 18 layer model (Figure 4 left).
- The reason for this is
- The reason behind this result is
- The residual models showed a reversed result
- The residual models show a contradictory result
- This proofed that the degradation problem
- This result proves that the degradation problem
- that an increased network depth gains accuracy.
- that an increased network depth results in a gain in accuracy.
- Optimizations (section title)
- Optimization Methods (or "Optimization" - no plural)
- A proposed optimization is showed in Figure 6 b.
- A proposed optimization is shown in Figure 6 b.
- The proposed design is
- The proposed design consists of
- the residual block and so through the whole network.
- the residual block as a result, through the entire network.
- to any other block, in both forward and backward passes.
- to any other block, during both forward and backward passes.
- also becomes easier
- also becomes simpler (or if that is what you mean; "computationally simpler" or "computationally less expensive")
- The training also becomes easier with the new block design.
- (it might be a good idea to remind that it is simpler in comparison to the original design in Figure 1)
- change these to
- change this representation to
- The removal of the ReLU function on the main path and the different design of the Residual function F(x)F(x) is related.
- (I did not understand what you mean by this sentence. Related to what, each other?)
- In the Figure above you can see that
- Figure x shows that
- got moved to the the residual function
- is moved to the the residual function
- This means the activation functions (ReLU) is
- This means that the activation functions (ReLU) are
- a "pre-activation" of the weight layers.
- a "pre-activation" function of the weight layers.
- This means the activation functions (ReLU) is now a "pre-activation" of the weight layers.
- This means that the activation functions (ReLU) now serve as a "pre-activation" function for the weight layers. (alternative to the last two corrections)
- performs similiar
- performs similarly

- By adding BN to the pre-activation, the result improved by a healty margin.
- By adding BN to the pre-activation, the result can be improved by a healthy margin.
- These "pre-activation" model
- This "pre-activation" model

Final comments:

- It is important to note that some of the corrections / suggestions are subjective and for you to decide
- Most of the issues I pointed out under the general topics are omitted in the correction list but I am pretty sure I still missed some stuff. Hopefully the next review will catch any mistakes I skipped.
- Generally a really good effort explaining one of the cutting edge methods to improve CNNs. Anyone who knows the basic part should not have difficulty understanding the page.
- I intended using inline comments but it turns into a mailspam (at least for the reviewer) so decided to use the same method as Sebastian.