



ResNet (Shortcut Connection과 Identity Mapping) [설명/요약/정리]

2019.12.31 11:19 딥러닝/Concepts

글 작성자: 만렙개발자

Best Python IDE

Debug Python code inside Docker containers

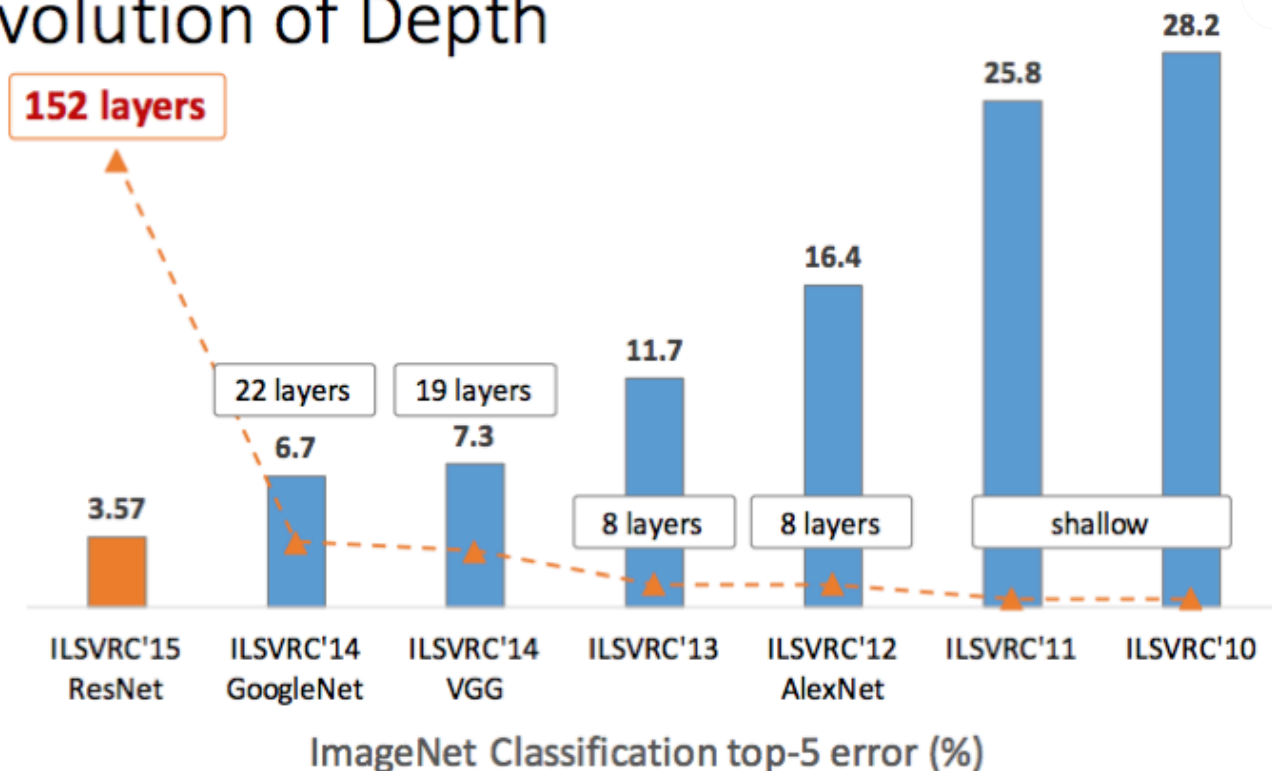
JetBrains

Download

등장 배경

레이어가 많아져, 인공신경망이 더 깊어질 수록 gradient vanishing/exploding 문제가 커집니다. 이전의 다른 방법들로 이 문제를 해결했다고 생각했지만, 수십개의 레이어를 가진 모델에서는 레이어를 더 추가한다고 해서 성능이 이전보다 더 좋아지지 않았습니다. 그래서 깊은 레이어까지 잘 학습이 되도록하는 방법을 Kaming he가 고안했고, 그것이 바로 shortcut(Skip) connection입니다. keras기준에서 쉽게 얘기하면, 단순히 더하는 add입니다. 레이어간의 연결이 순서대로 연속적인 것만 있는 것이 아니라, 중간을 뛰어넘어 전달하는(더하는) shortcut이 추가된 것입니다. 연산은 매우 간단하고, 개념도 매우 간단하지만 이것이 그래디언트를 직접적으로 잘 전달하여 gradient vanishing/exploding 문제를 해결하는 큰 효과를 불러일으키게 됩니다. ResNet을 공개한 첫 논문에서는 121개의 레이어를 가진 모델까지도 성능 향상을 보였습니다.

Revolution of Depth



레이어 수의 증가폭이 ResNet을 통해 엄청나게 변화한 걸 볼 수 있습니다. (ResNet v2에서는 심지어 1000개 까지도...)

Shortcut(Skip) Connection

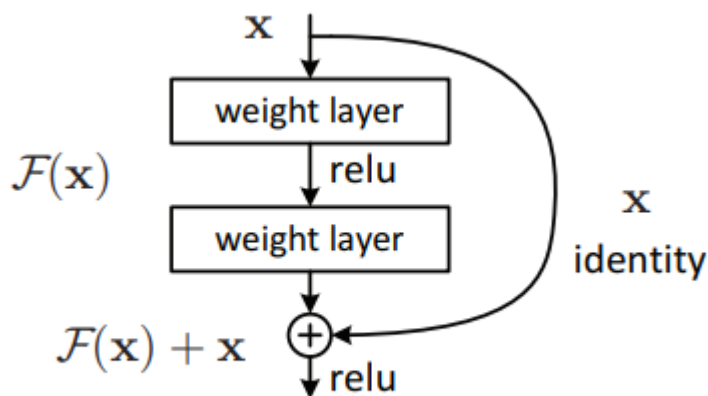
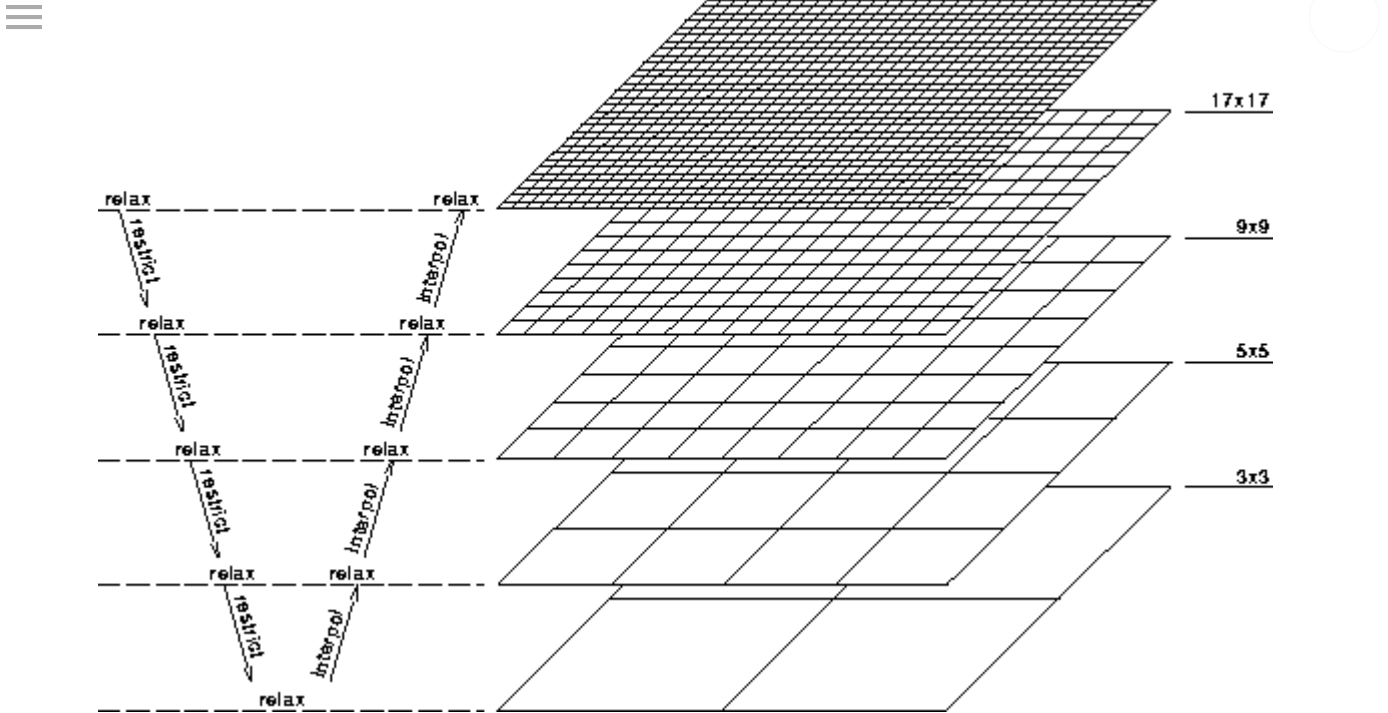
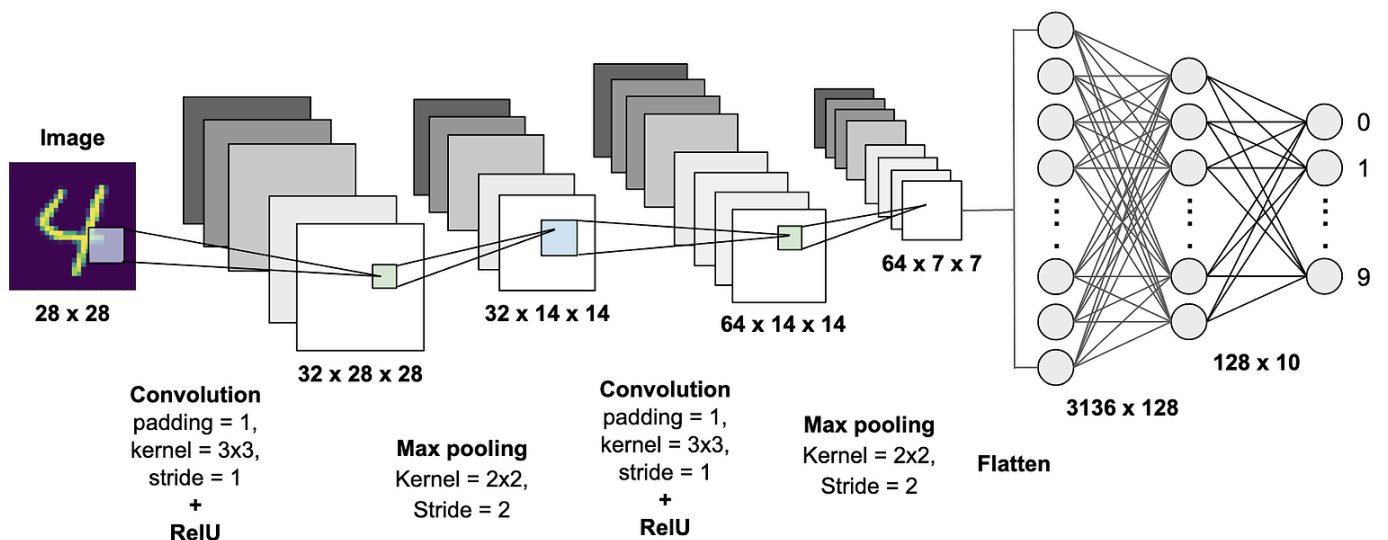


Figure 2. Residual learning: a building block.

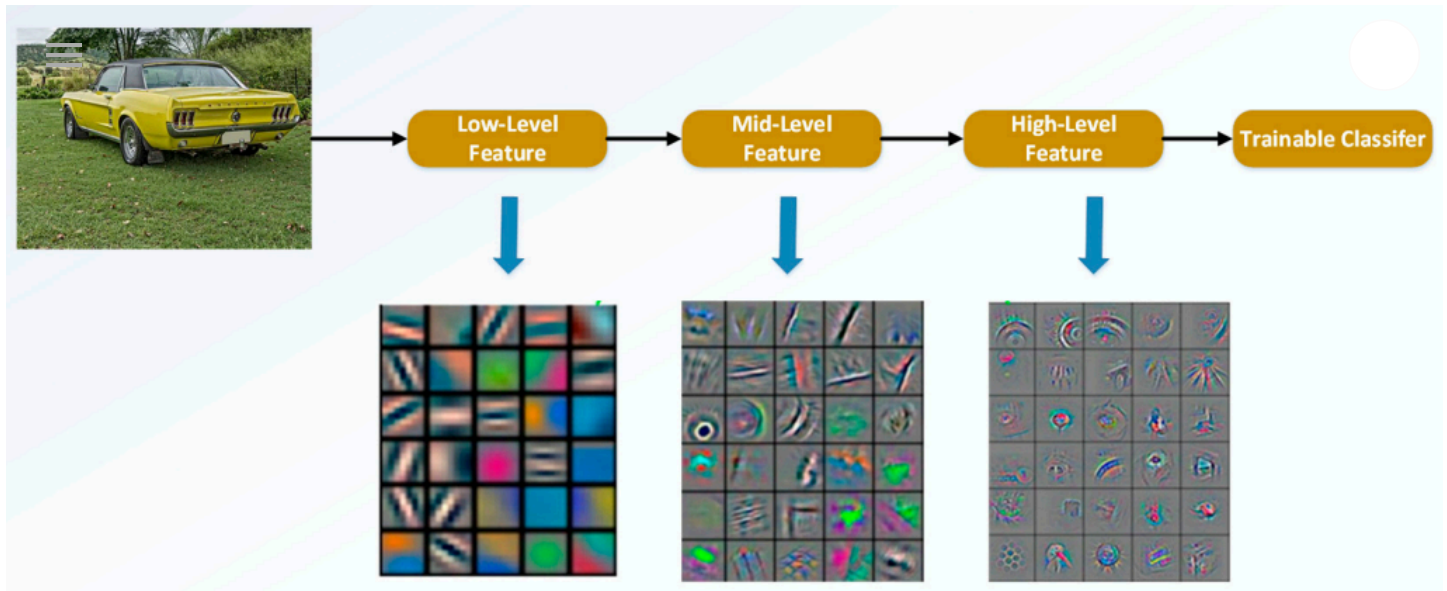
low-level vision과 computer graphics 분야에서 Partial Differential Equations(PDEs)를 풀기 위해 Multigrid 방법들이 많이 사용됩니다. 이것은 다양한 스케일(크기)에서의 하위문제로 시스템을 재정의하는 것입니다. 쉽게 설명하면, 코카콜라 캔을 학습한다고 하였을 때, 코카콜라 캔이 가까이에서 찍힌 것은 크게 나타나고, 멀리서 찍힌 것은 작게 나타납니다. 이러한 스케일의 변화에 대응하기 위한 컴퓨터 비전 기술은 pyramid와 같은 형태의 multigrid를 만들어서 해결합니다.



CNN 모델의 구조에서도 아래와 같이 이러한 형태가 나타납니다.



그런데 다른점은 무엇이라면, 이전에는 multigrid를 만들어서 (이미지 크기를 다양하게 하거나, 필터 크기를 다양하게 해서) 각각을 계산하는 방식을 사용하였습니다. 기존의 CNN 구조에서는 이전의 것이 다음으로 전달되어 영향을 미치게 됩니다. 입력부분에 가까운 하위 레이어에서는 매우 단순한 구조나 노이즈한 패턴이 보이는 low-level feature가 학습이 되고, 출력부분에 가까운 상위 레이어에서는 구조적인 부분이 학습되는 high-level feature가 학습됩니다.



그런데 앞선 부분의 feature가 뒤쪽까지 영향이 직접적으로 전달되는 것이 아니라, 중간을 거쳐 전달되기 때문에 학습의 과정에서 크게크게 변합니다. 그런데 shortcut connection을 추가해주게 되면 (수식적으로) 이전으로부터 얼마나 변하는지 나머지(residual)만 계산하는 문제로 바뀌게 됩니다. 즉, 현재 레이어의 출력값과 이전 스케일의 레이어 출력값을 더해 입력을 받기 때문에, 그 차이를 볼 수 있게 되는 것이죠. 따라서 학습하는 과정에서 그 '조금'을 하면 되는 것이고, 더 빠르게 학습한다는 장점이 생깁니다!

Identity Mapping

이전에 얇은 모델과 깊은 모델을 비교했을 때, 깊은 모델이 더 안좋아진다고 했었습니다. 그런데 Kaiming He는 그래서 안된다고 했습니다. 그런 문제점을 identity mapping이라는 것을 통해 꼬집었습니다. 얇은 모델에서 단순히 아무것도 하지 않는 layer인(convolution을 통과하지 않고 값을 전달하는) identity mapping을 쌓으면(덧셈 연산으로) 얇은 모델 그대로의 성능을 나타낼 것이라는 자명한 사실에 하나의 가정을 더합니다. "쌓여있는 레이어가 underlying mapping을 fit하는 것보다 residual mapping을 fit하는 것이 쉽다." 그리고 shortcut connection이 이 역할을 정확하게 할 수 있다고 말합니다. 즉, 이전에 학습된 모델(레이어들)의 출력과 추가된 레이어의 출력의 차이값인 나머지(residual)만 학습하면 되기에 연산이 간단해지고, error값 크기의 측면에서 학습이 더 쉽다는 것입니다. identity mapping은 입력값을 그대로 전달한다는 의미에서 identity입니다. 위의 shortcut connection에서 identity로 표현한 것처럼, shortcut connection과 identity mapping은 다른 것이 아니라, 의미적으로 identity는 값을 그대로 보낸다는 것입니다. :)

Bottleneck Architecture

이전에 1x1 convolution에 대해서 이야기하면서 bottleneck architecture를 소개했었습니다. 이 bottleneck 구조는 ResNet-v2에 적용되어있으며, 깊은 네트워크 학습을 빠르게하기 위해서 차원을 축소하여 연산량을 줄이는데 사용됩니다. 자세한 설명은 아래의 포스트를 참고하시길 바랍니다. ResNet-v2에서는 심지어 1001개의 레이어를 쌓아 성능을 보입니다. 추가적으로 Batch Normalization의 위치와 pre-activation function 등 다양한 것에 대한 실험 결과가 있습니다.

[Deep Learning] - 1x1 convolution의 의미 = 차원 축소 그리고 bottleneck



1x1 convolution의 의미 = 차원 축소 그...

1x1 convolution 이면 사실 convolution의 의미는 사라지는 것 같다.

영상처리의 개념에서의 filter는 1x1 ↙

lv99.tistory.com

| CIFAR-10 | error (%) | CIFAR-100 | error (%) |
|---|-------------------------|----------------------------|---------------------------|
| NIN [15] | 8.81 | NIN [15] | 35.68 |
| DSN [16] | 8.22 | DSN [16] | 34.57 |
| FitNet [17] | 8.39 | FitNet [17] | 35.04 |
| Highway [7] | 7.72 | Highway [7] | 32.39 |
| All-CNN [14] | 7.25 | All-CNN [14] | 33.71 |
| ELU [12] | 6.55 | ELU [12] | 24.28 |
| FitResNet, LSUV [18] | 5.84 | FitNet, LSUV [18] | 27.66 |
| ResNet-110 [1] (1.7M) | 6.61 | ResNet-164 [1] (1.7M) | 25.16 |
| ResNet-1202 [1] (19.4M) | 7.93 | ResNet-1001 [1] (10.2M) | 27.82 |
| ResNet-164 [ours] (1.7M) | 5.46 | ResNet-164 [ours] (1.7M) | 24.33 |
| ResNet-1001 [ours] (10.2M) | 4.92 (4.89±0.14) | ResNet-1001 [ours] (10.2M) | 22.71 (22.68±0.22) |
| ResNet-1001 [ours] (10.2M) [†] | 4.62 (4.69±0.20) | | |

ResNet in Keras

```

1 keras.applications.resnet.ResNet50(include_top=True, weights='imagenet',
  input_tensor=None, input_shape=None, pooling=None, classes=1000)
2 keras.applications.resnet.ResNet101(include_top=True, weights='imagenet',
  input_tensor=None, input_shape=None, pooling=None, classes=1000)
3 keras.applications.resnet.ResNet152(include_top=True, weights='imagenet',
  input_tensor=None, input_shape=None, pooling=None, classes=1000)
4 keras.applications.resnet_v2.ResNet50V2(include_top=True, weights='imagenet',
  input_tensor=None, input_shape=None, pooling=None, classes=1000)
5 keras.applications.resnet_v2.ResNet101V2(include_top=True, weights='imagenet',
  input_tensor=None, input_shape=None, pooling=None, classes=1000)
6 keras.applications.resnet_v2.ResNet152V2(include_top=True, weights='imagenet',
  input_tensor=None, input_shape=None, pooling=None, classes=1000)
7 keras.applications.inception_resnet_v2.InceptionResNetV2(include_top=True,
  weights='imagenet', input_tensor=None, input_shape=None, pooling=None, classes=1000)

```

pre-trained된 ResNet들이 공개되어있고, 쉽게 가져다가 사용할 수 있습니다. resnet_v2도 구분되어있으며, inception resnet도 있습니다.

결론

결론적으로 shortcut connection은 덧셈 연산량이 증가할 뿐, 학습 파라미터의 수에는 큰 영향이 없으며, 더 많은 수의 레이어를 갖는 깊은 모델도 잘 학습하고, 심지어 더 쉽고 빠르게 학습이 가능하다는 것입니다. 간단한 '덧셈'으로 이러한 결과를 가져온다는 것이 제게는 충격이 아닐 수가 없었습니다. 현재도 많은 모델들의 대표적인 backbone으로 사용되고 있으며, 발전된 형태의 inception-resnet이나 resnext등도 많이 공개되어있습니다. 현재 residual connection은 깊은 네트워크를 위해서는 필수적인 요소가 되었습니다.

ref: Deep Residual Learning for Image Recognition

ref: Identity Mappings in Deep Residual Networks

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

ref: MNIST Handwritten Digits Classification using a Convolutional Neural Network (CNN)

ref: Multi-Object Detection in Traffic Scenes Based on Improved SSD



...

☆ 구독하기



'📚 딥러닝 > Concepts' 카테고리의 다른 글



- 메트릭 러닝과 트리플렛 로스 (Metric Learning an... → Transformer: All you need is Attention (설명/... →
- 가중치 초기화의 모든 것 (All about weight initial... → AutoEncoder (self-supervised learning) [설명/... →
- 1x1 convolution의 의미 = 차원 축소, 그리고 bott... →

🔍 cnn

Connection

identity

mapping

residual

Resnet

shortcut

Skip

댓글

공유하기

다른 글