مبانی بینایی کامپیوتر

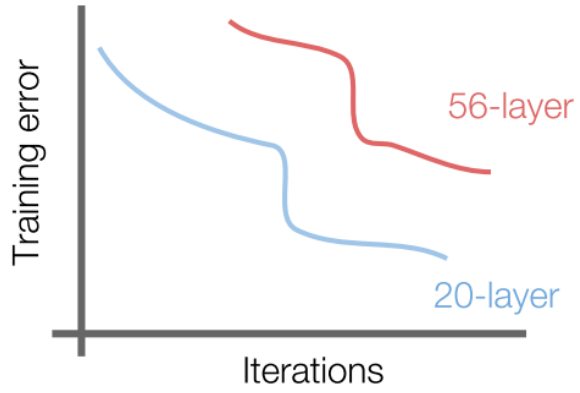
مدرس: محمدرضا محمدی

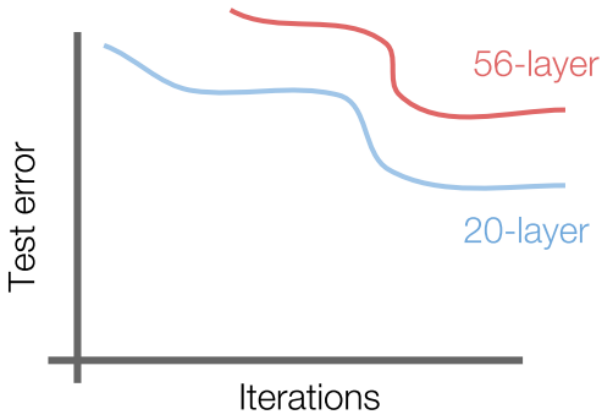
۱۴۰۱

معماری‌های CNN

CNN Architectures

ResNet

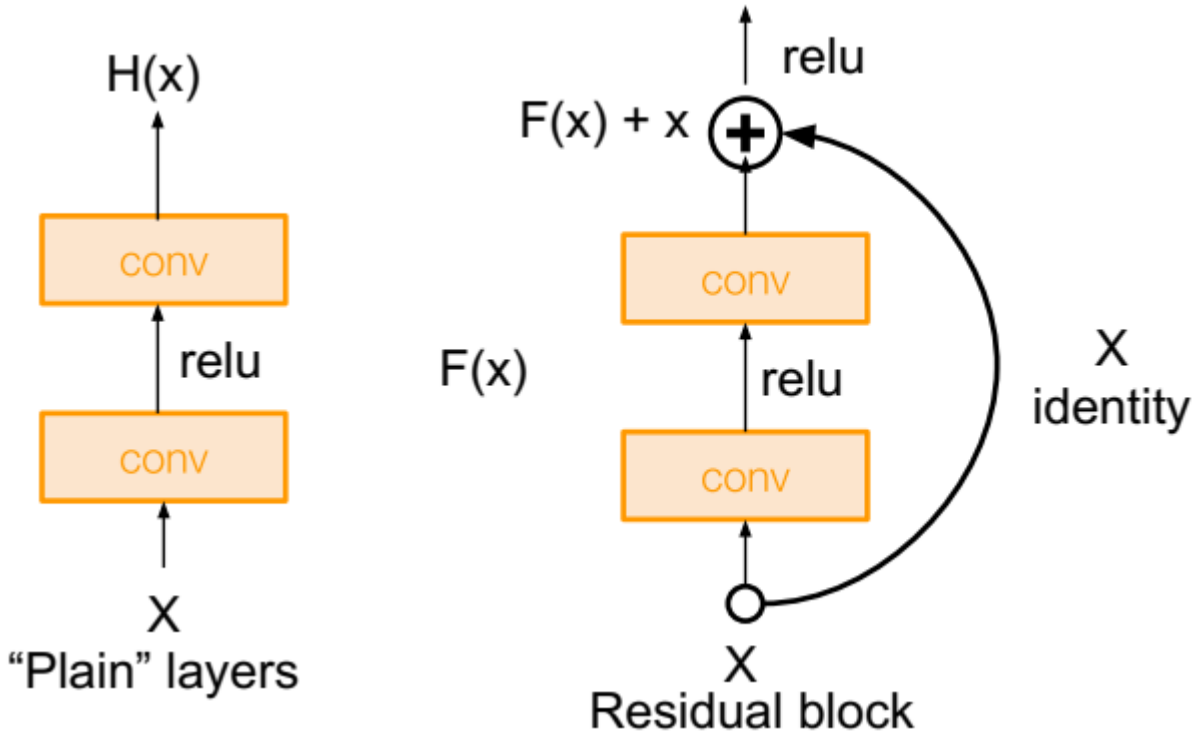


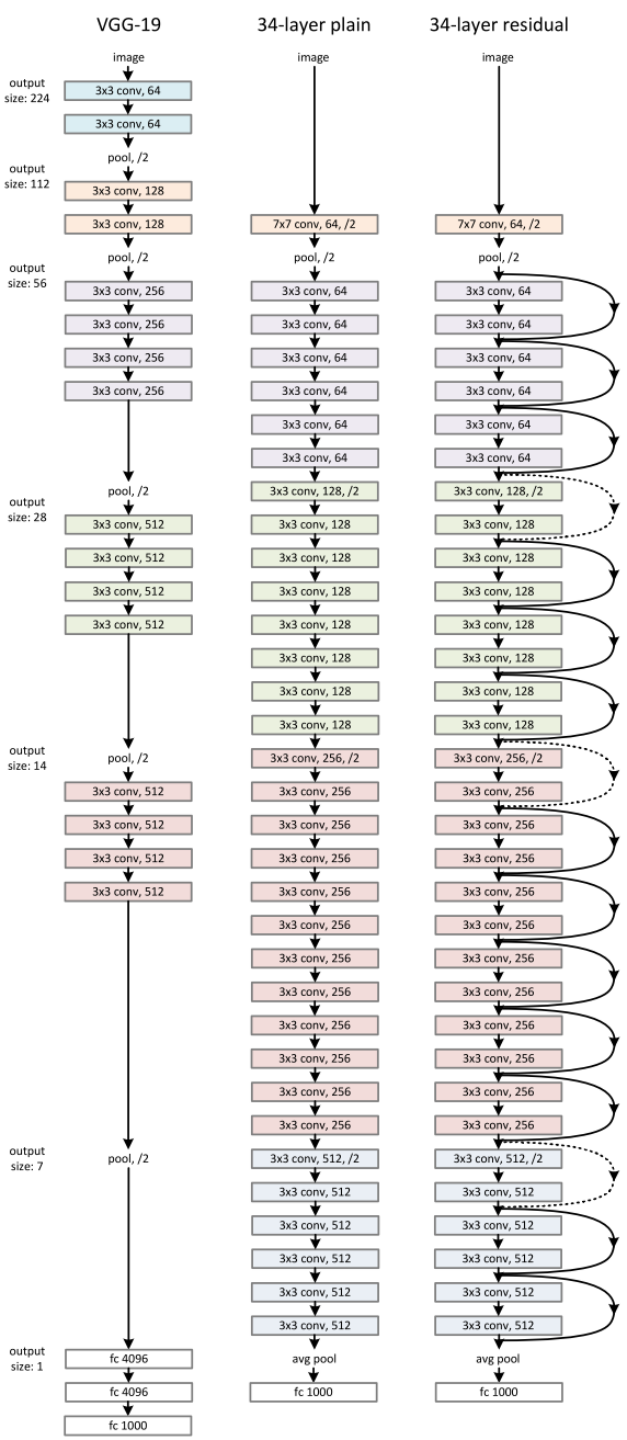


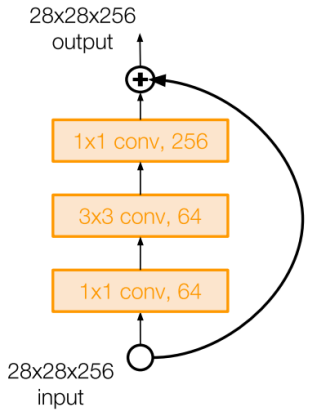
- اگر تعداد لایه‌های کانولوشنی ساده را بسیار زیاد کنیم چه اتفاقی می‌افتد؟
- چرا شبکه عمیق‌تر هم در آموزش و هم در آزمون عملکرد ضعیف‌تری دارد؟
  - البته مشکل از overfitting نیست!
- فرضیه: مشکل در مسئله بهینه‌سازی است
  - بهینه‌سازی مدل‌های عمیق‌تر دشوارتر است
- عملکرد مدل‌های عمیق‌تر باید حداقل به خوبی مدل‌های با عمق کمتر باشد
  - می‌توان وزن‌های مدل کم‌عمق را به لایه‌های نخست شبکه عمیق کپی کرد و لایه‌های اضافی را به گونه‌ای تنظیم کرد که نگاشت همانی را انجام دهند
- ایده ResNet آن است که لایه‌های شبکه بجای آموختن نگاشت مطلوب، باقی‌مانده آن را یاد بگیرند

ResNet

- لایه‌ها باید $F(x) = H(x) - x$ را بجای $H(x)$ بیاموزند







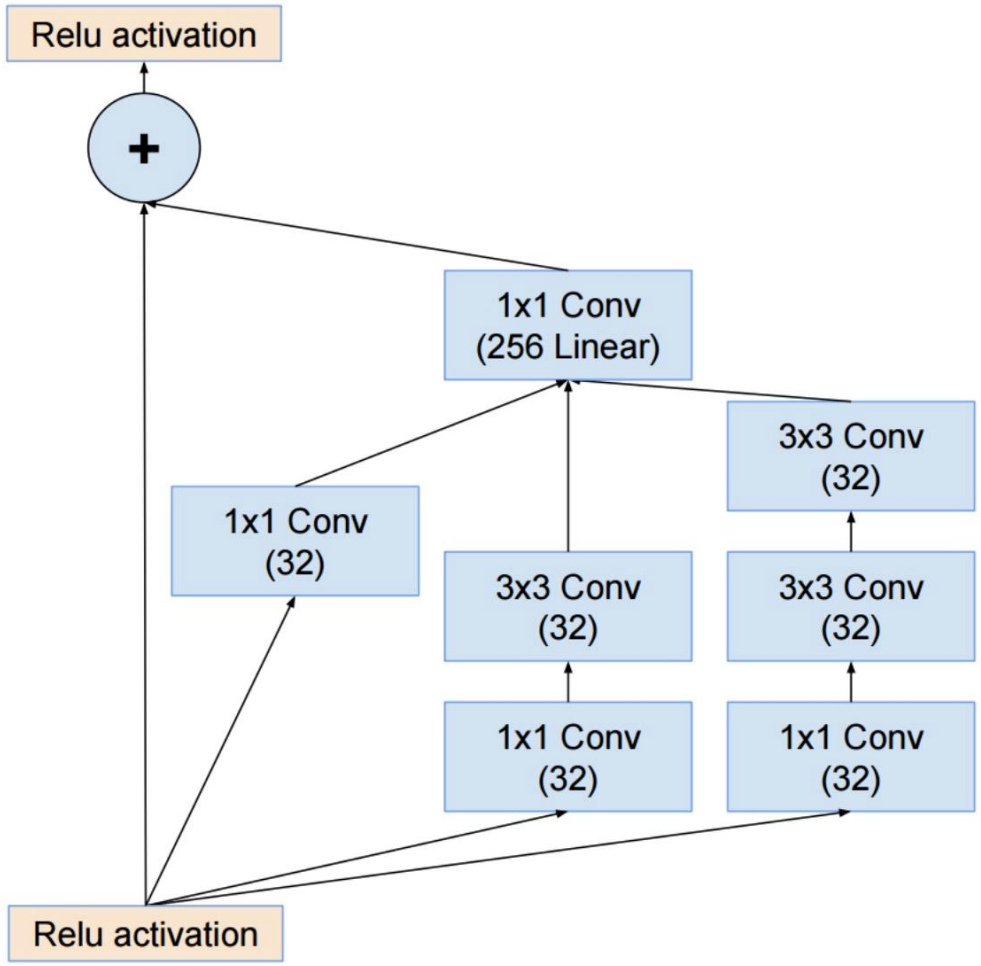
ResNet

- از تعداد زیادی بلوک باقی‌مانده تشکیل شده است
- هر بلوک باقی‌مانده دارای ۲ لایه کانولوشنی ۳×۳ است
- به طور دوره‌ای، تعداد فیلترها ۲ برابر شده و رزولوشن مکانی نصف می‌شود
- در ابتدا دارای یک لایه کانولوشنی است
- پس از آخرین بلوک باقی‌مانده، ابعاد داده‌ها با استفاده از Average Pooling کاهش می‌یابد و یک لایه FC برای دسته‌بندی استفاده می‌شود
- برای مسئله ImageNet عمق‌های مختلف شبکه شامل ۳۴، ۵۰، ۱۰۱ و ۱۵۲ استفاده شده‌اند
- در شبکه‌های عمیق‌تر، از لایه کانولوشنی ۱×۱ برای بهبود بهره‌وری استفاده شده است

ResNet

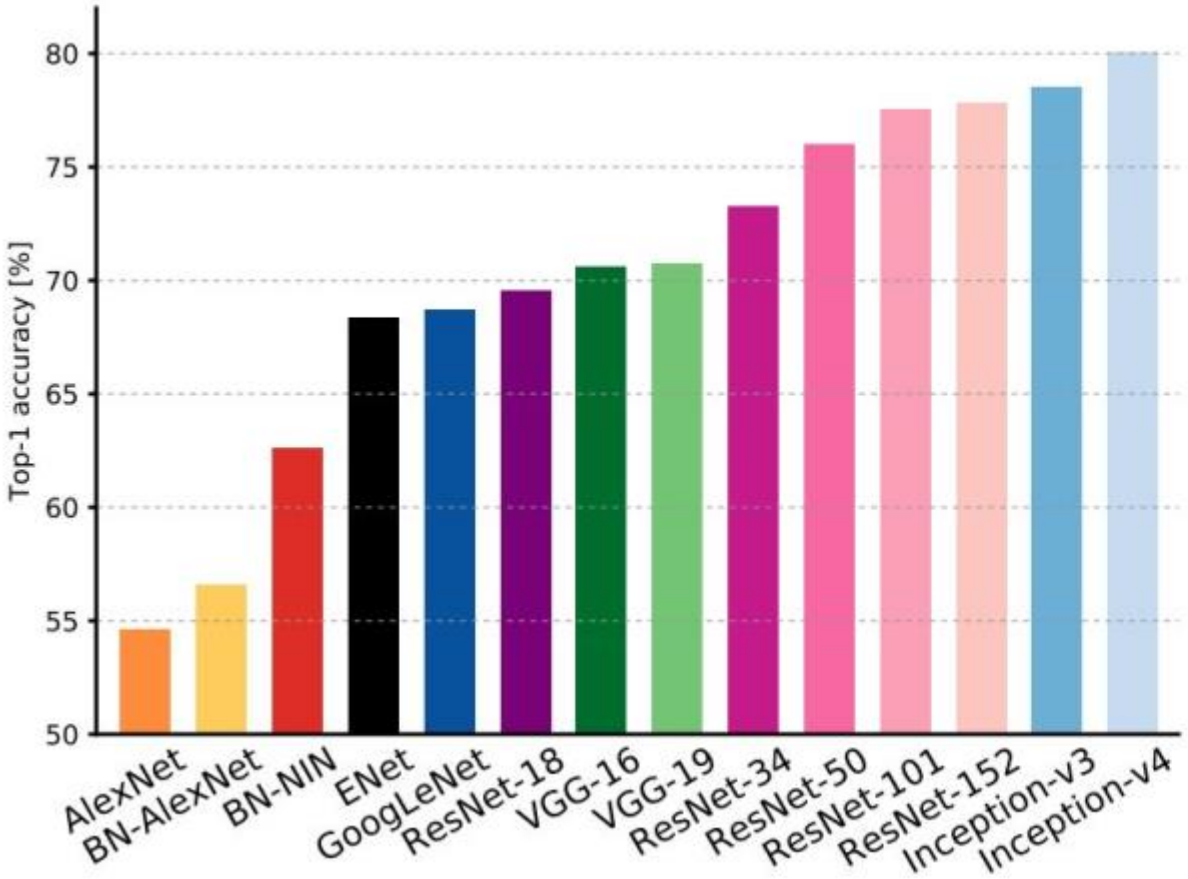
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	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

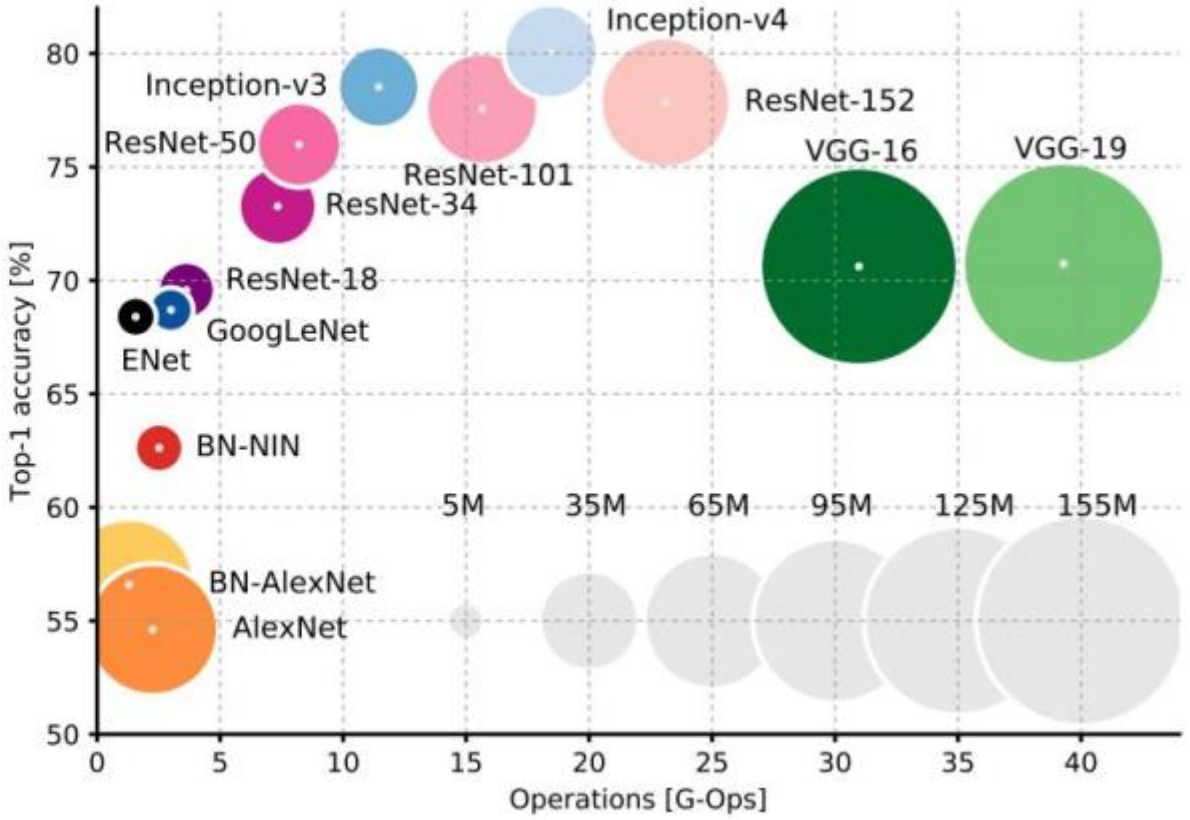
Inception V4



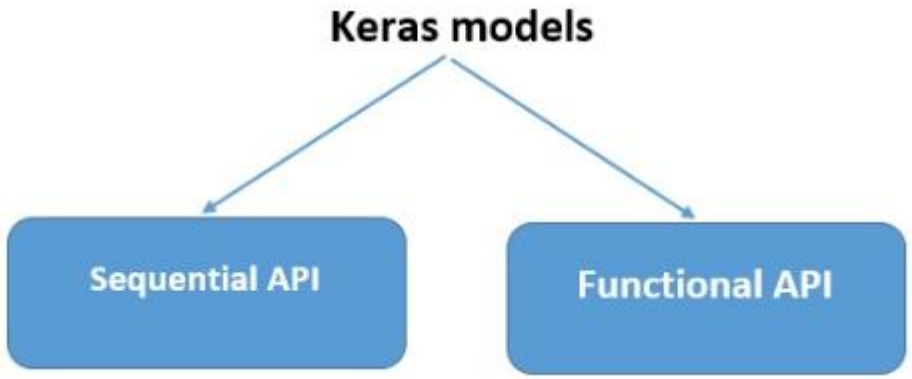
- ورودی همانی به ماژول Inception افزوده شده است

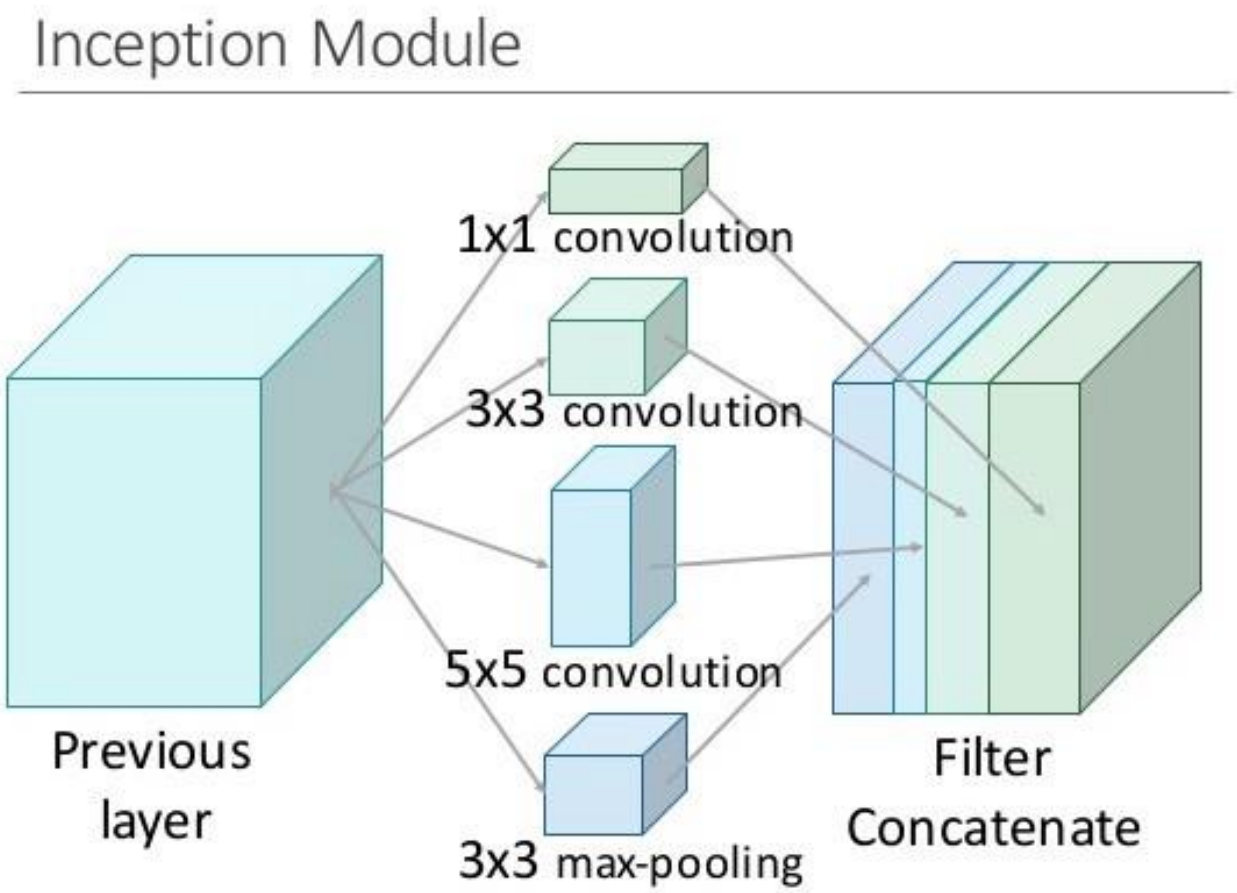
مقایسه


Top-1 accuracy [%]
80
75
70
65
60
55
50
AlexNet
BN-AlexNet
BN-NIN
ENet
GoogLeNet
ResNet-18
VGG-16
VGG-19
ResNet-34
ResNet-50
ResNet-101
ResNet-152
Inception-v3
Inception-v4


Top-1 accuracy [%]
80
75
70
65
60
55
50
0
5
10
15
20
25
30
35
40
Operations [G-Ops]
Inception-v4
Inception-v3
ResNet-152
ResNet-50
ResNet-101
VGG-16
VGG-19
ResNet-34
ResNet-18
GoogLeNet
ENet
BN-NIN
5M
35M
65M
95M
125M
155M
BN-AlexNet
AlexNet

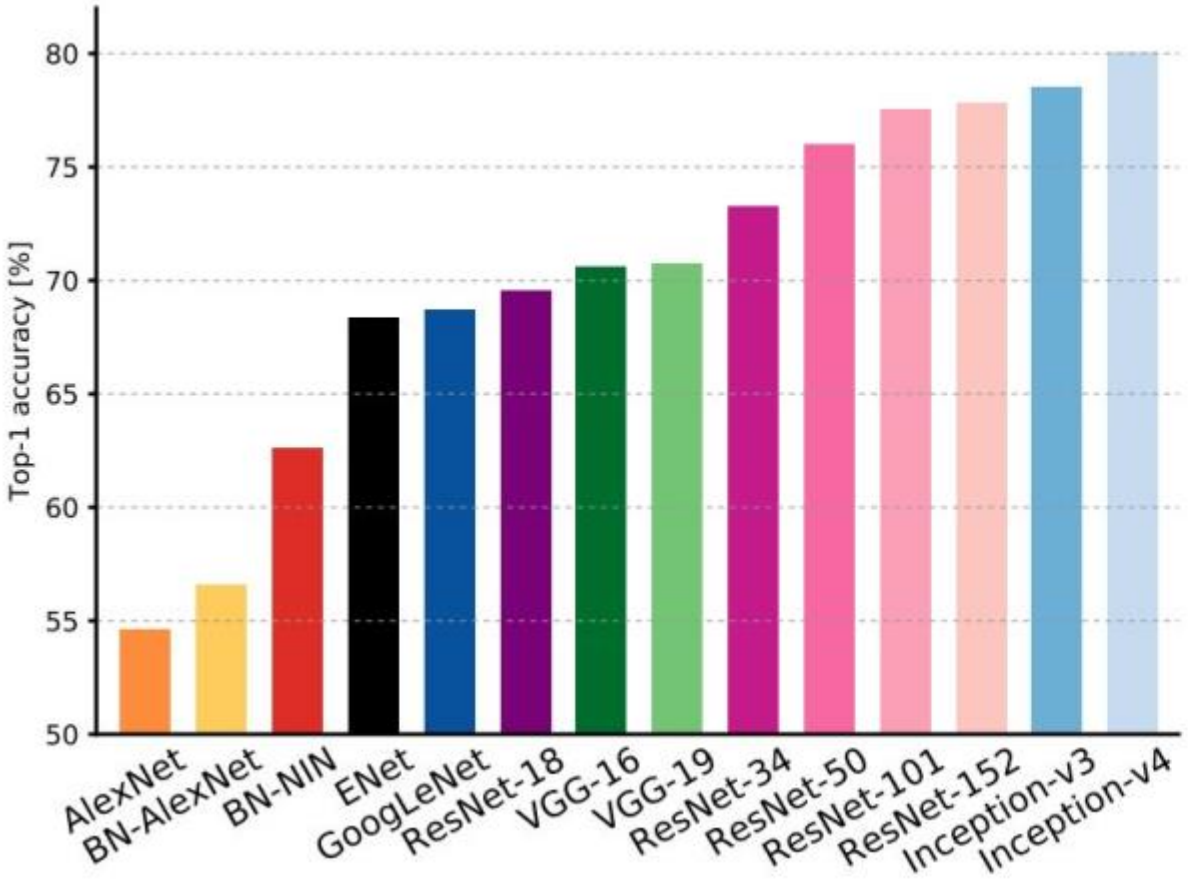
مدل‌های Functional

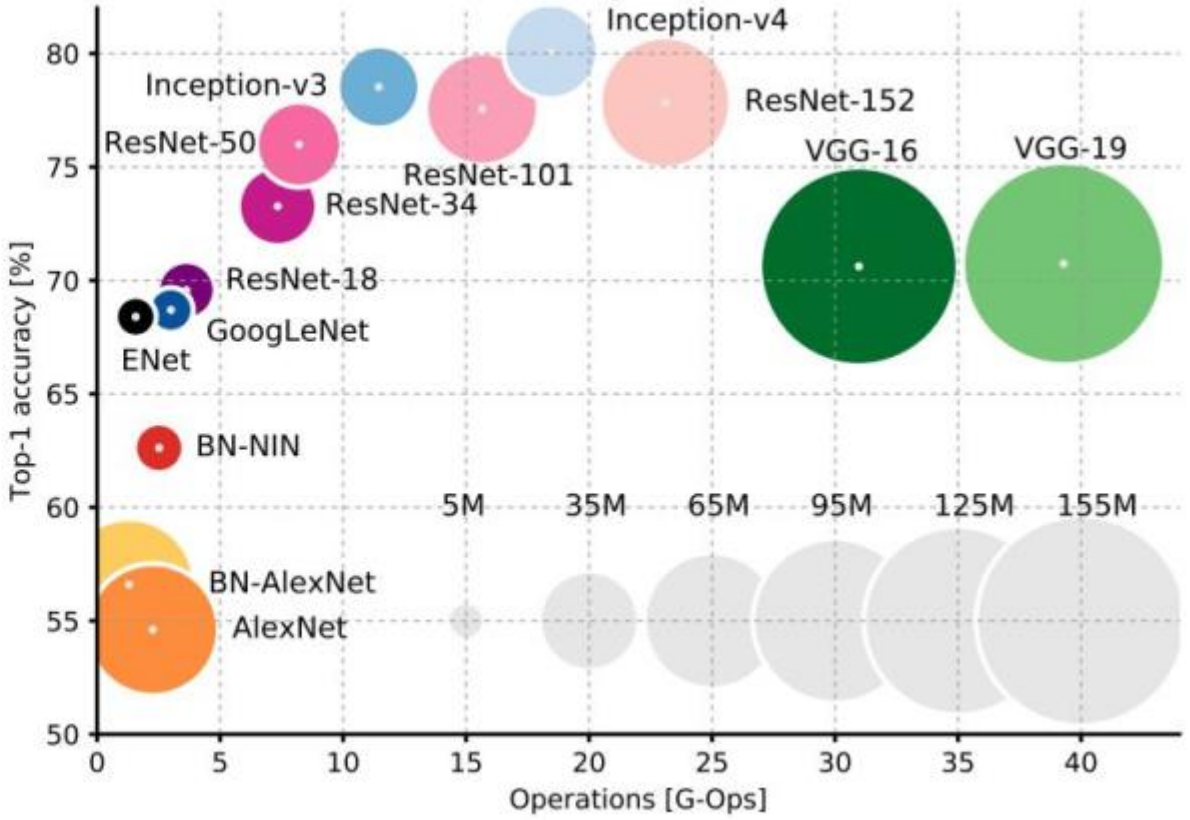




https://colab.research.google.com/drive/1DZ-kJ-HFWD2RWK12s2ak4TlNG6mEMszB?usp=sharing

مقایسه


Top-1 accuracy [%]
80
75
70
65
60
55
50
AlexNet
BN-AlexNet
BN-NIN
ENet
GoogLeNet
ResNet-18
VGG-16
VGG-19
ResNet-34
ResNet-50
ResNet-101
ResNet-152
Inception-v3
Inception-v4


Top-1 accuracy [%]
80
75
70
65
60
55
50
0
5
10
15
20
25
30
35
40
Operations [G-Ops]
Inception-v4
Inception-v3
ResNet-152
ResNet-50
ResNet-101
VGG-16
VGG-19
ResNet-34
ResNet-18
GoogLeNet
ENet
BN-NIN
5M
35M
65M
95M
125M
155M
BN-AlexNet
AlexNet

Available models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-

Cars Dataset

Overview

The *Cars* dataset contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of *Make, Model, Year*, e.g. 2012 Tesla Model S or 2012 BMW M3 coupe.

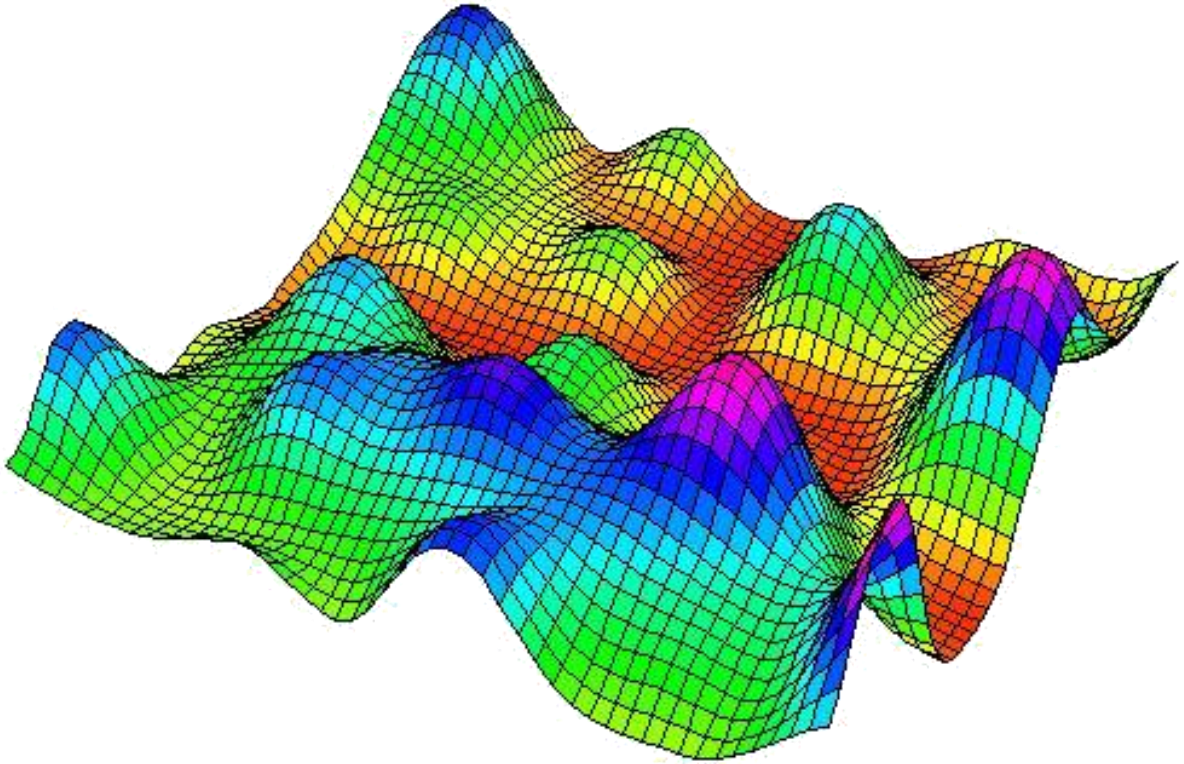
https://colab.research.google.com/drive/1Iw2IBKc-Y3m4ALwKLvkI6JiBXZqQ8eMu?usp=sharing

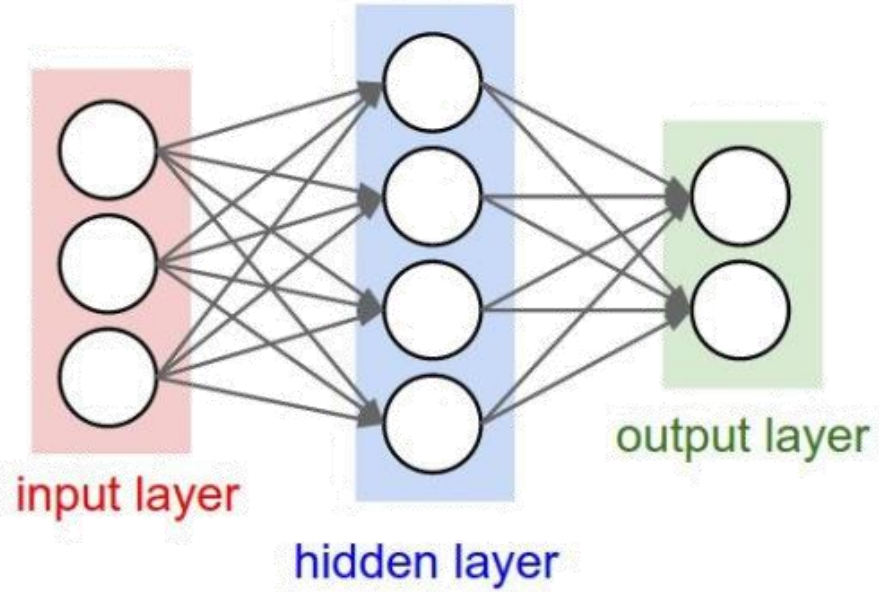
مقداردهی اولیه وزن‌ها

Weight Initialization

مقداردهی اولیه

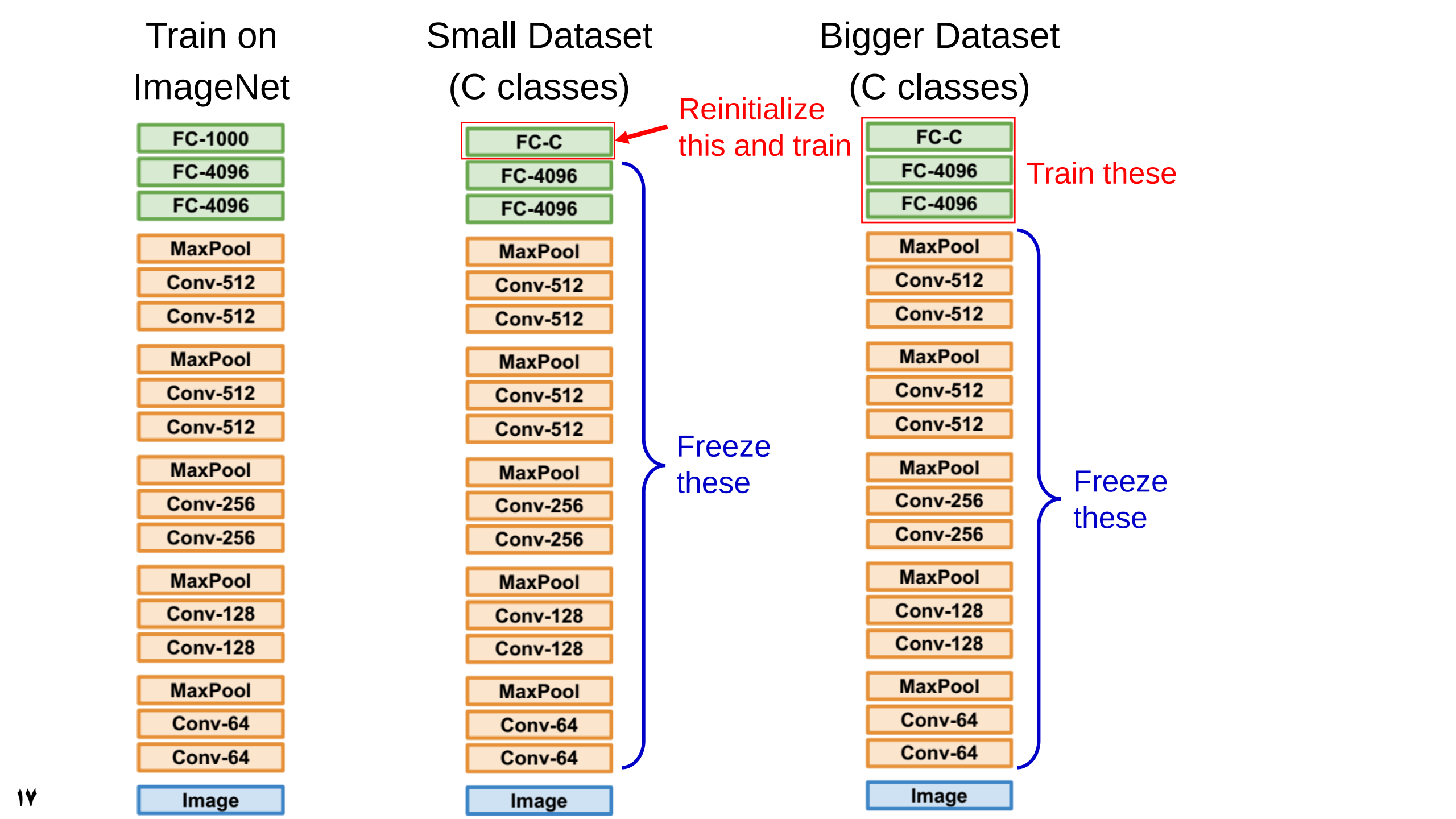
- در روش‌های بهینه‌سازی مبتنی بر تکرار، نقطه شروع بهینه‌سازی بسیار مهم است
- اگر در ابتدای کار تمام وزن‌های شبکه مقدار صفر داشته باشند چه اتفاقی می‌افتد؟

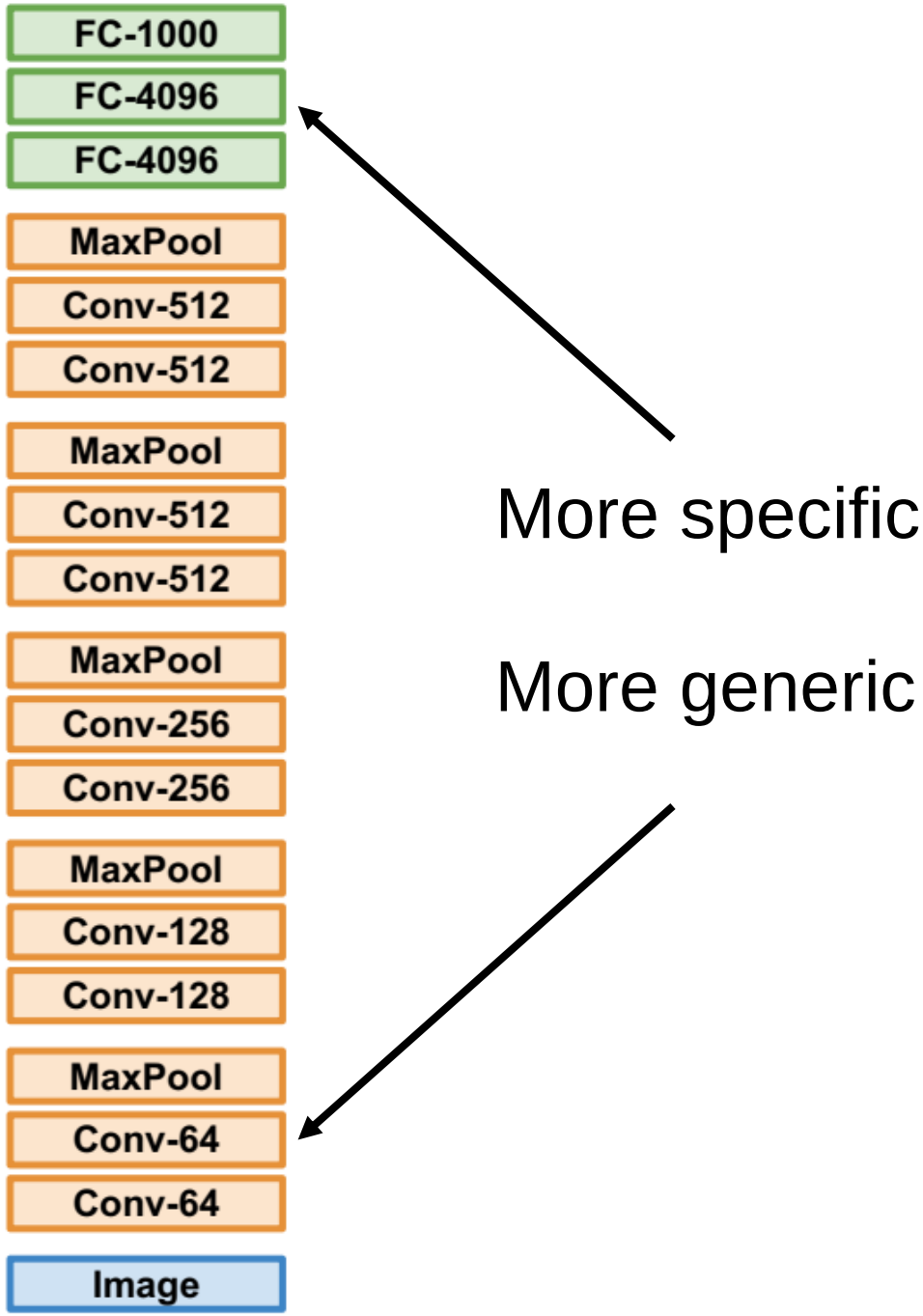




مقداردهی اولیه

- روش مقداردهی اولیه Xavier یکی از معروفترین روش‌های وزن‌دهای اولیه است
- مقداردهی اولیه مناسب هنوز یک زمینه تحقیقاتی فعال است
- به خصوص برای مجموعه داده‌های کوچک، مقداردهی اولیه بسیار حائز اهمیت است
- برای آموزش یک شبکه CNN با میلیون‌ها پارامتر، حجم زیادی از داده‌های آموزشی لازم است
- با استفاده از تقویت داده و دیگر روش‌های تنظیم پارامترهای شبکه می‌توان تا حدی کمبود داده را جبران کرد
- یکی از بهترین روش‌ها برای مقداردهی اولیه پارامترهای یک شبکه استفاده از شبکه‌های pretrained است
- انتقال یادگیری روش بسیار موثری است تا دانش بدست آمده توسط یک شبکه به شبکه جدید منتقل شود


Train on ImageNet
FC-1000
FC-4096
FC-4096
MaxPool
Conv-512
Conv-512
MaxPool
Conv-512
Conv-512
MaxPool
Conv-256
Conv-256
MaxPool
Conv-128
Conv-128
MaxPool
Conv-64
Conv-64
Image
Small Dataset (C classes)
FC-C
FC-4096
FC-4096
MaxPool
Conv-512
Conv-512
MaxPool
Conv-512
Conv-512
MaxPool
Conv-256
Conv-256
MaxPool
Conv-128
Conv-128
MaxPool
Conv-64
Conv-64
Image
Reinitialize this and train
Freeze these
Bigger Dataset (C classes)
FC-C
FC-4096
FC-4096
Train these
MaxPool
Conv-512
Conv-512
MaxPool
Conv-512
Conv-512
MaxPool
Conv-256
Conv-256
MaxPool
Conv-128
Conv-128
MaxPool
Conv-64
Conv-64
Image
Freeze these


FC-1000
FC-4096
FC-4096
MaxPool
Conv-512
Conv-512
MaxPool
Conv-512
Conv-512
MaxPool
Conv-256
Conv-256
MaxPool
Conv-128
Conv-128
MaxPool
Conv-64
Conv-64
Image
More specific
More generic

	مجموعه داده بسیار مشابه	مجموعه داده بسیار متفاوت
مجموعه داده خیلی کم	؟	؟
مجموعه داده زیاد	؟	؟

	مجموعه داده بسیار مشابه	مجموعه داده بسیار متفاوت
مجموعه داده خیلی کم	از یک دسته‌بند خطی در آخرین لایه استفاده شود	؟
مجموعه داده زیاد	؟	؟

	مجموعه داده بسیار مشابه	مجموعه داده بسیار متفاوت
مجموعه داده خیلی کم	از یک دسته‌بند خطی در آخرین لایه استفاده شود	؟
مجموعه داده زیاد	تعدادی از لایه‌های انتهایی تنظیم دقیق شوند	؟

	مجموعه داده بسیار مشابه	مجموعه داده بسیار متفاوت
مجموعه داده خیلی کم	از یک دسته‌بند خطی در آخرین لایه استفاده شود	مشکل است! می‌توان دسته‌بند خطی را در گام‌های مختلف امتحان کرد
مجموعه داده زیاد	تعدادی از لایه‌های انتهایی تنظیم دقیق شوند	؟

	مجموعه داده بسیار مشابه	مجموعه داده بسیار متفاوت
مجموعه داده خیلی کم	از یک دسته‌بند خطی در آخرین لایه استفاده شود	مشکل است! می‌توان دسته‌بند خطی را در گام‌های مختلف امتحان کرد
مجموعه داده زیاد	تعدادی از لایه‌های انتهایی تنظیم دقیق شوند	تعداد زیادی از لایه‌های انتهایی تنظیم دقیق شوند

انتقال یادگیری

- در صورتیکه مجموعه داده‌های شما به اندازه کافی بزرگ نیست و مسئله پیچیده است (شبکه دارای پارامترهای زیادی است):
  - یک مجموعه داده بسیار بزرگ که به مجموعه داده مورد نظر مشابه است انتخاب و شبکه کانولوشنی با آن آموزش ببیند
  - انتقال یادگیری به مجموعه داده مورد نظر انجام شود
- خوشبختانه مدل‌های پیش آموخته زیادی در دسترس هستند

PyTorch: https://github.com/pytorch/vision

TensorFlow: https://github.com/tensorflow/models

Caffe: https://github.com/BVLC/caffe/wiki/Model-Zoo

MatConvNet: http://www.vlfeat.org/matconvnet/pretrained/

Keras: https://github.com/fchollet/deep-learning-models/releases/