
بسم الله الرحمن الرحيم

مبانی بینایی کامپیوتر

مدرس: محمدرضا محمدی

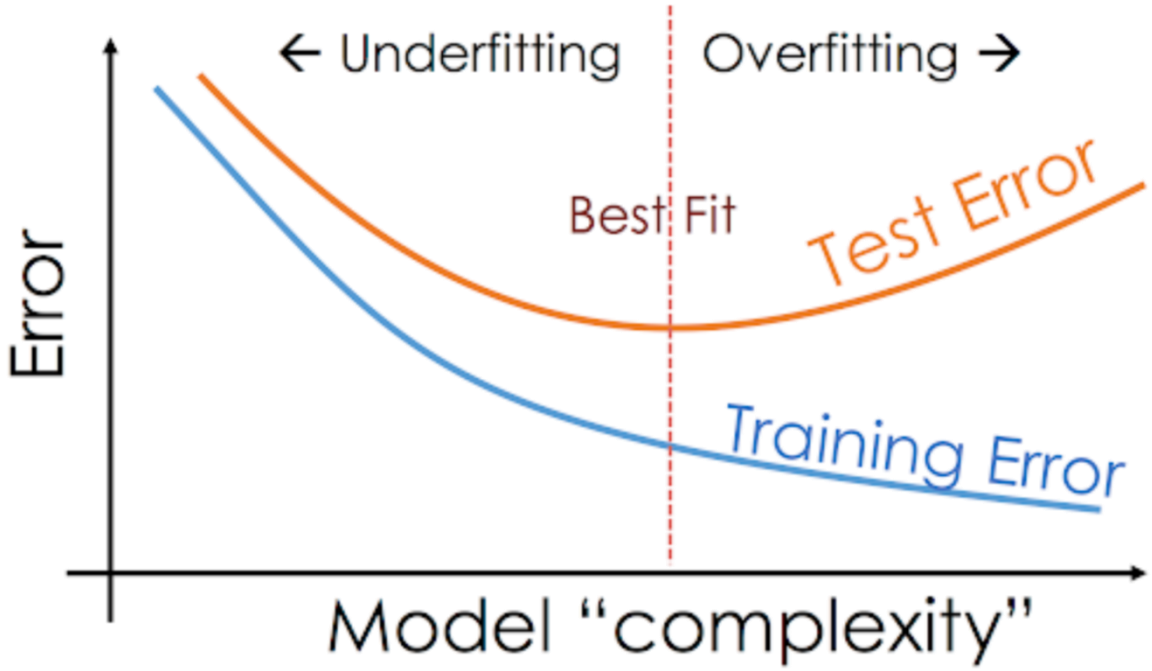
۱۳۹۹

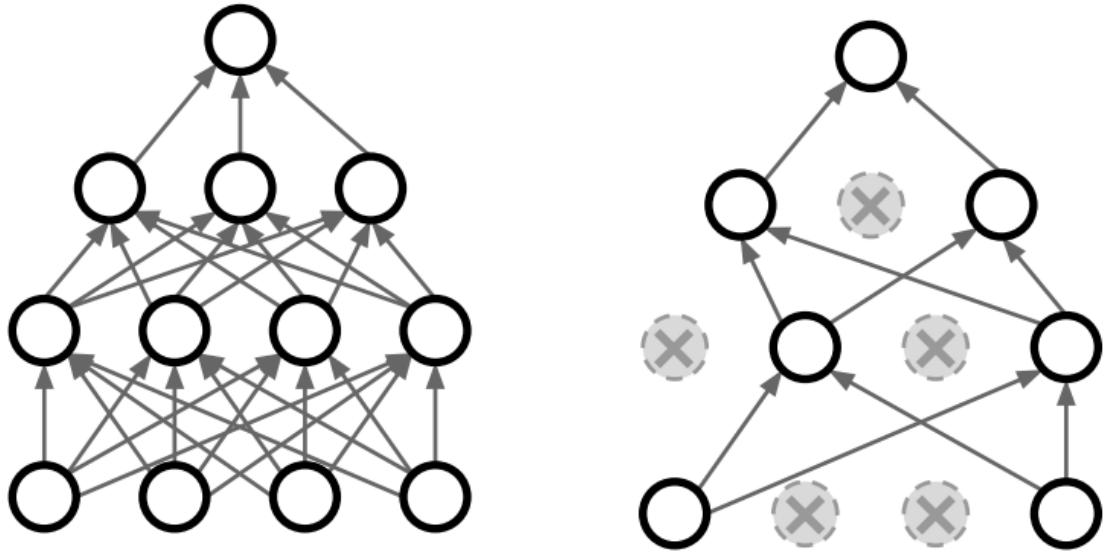
داده‌افزایی

Data Augmentation

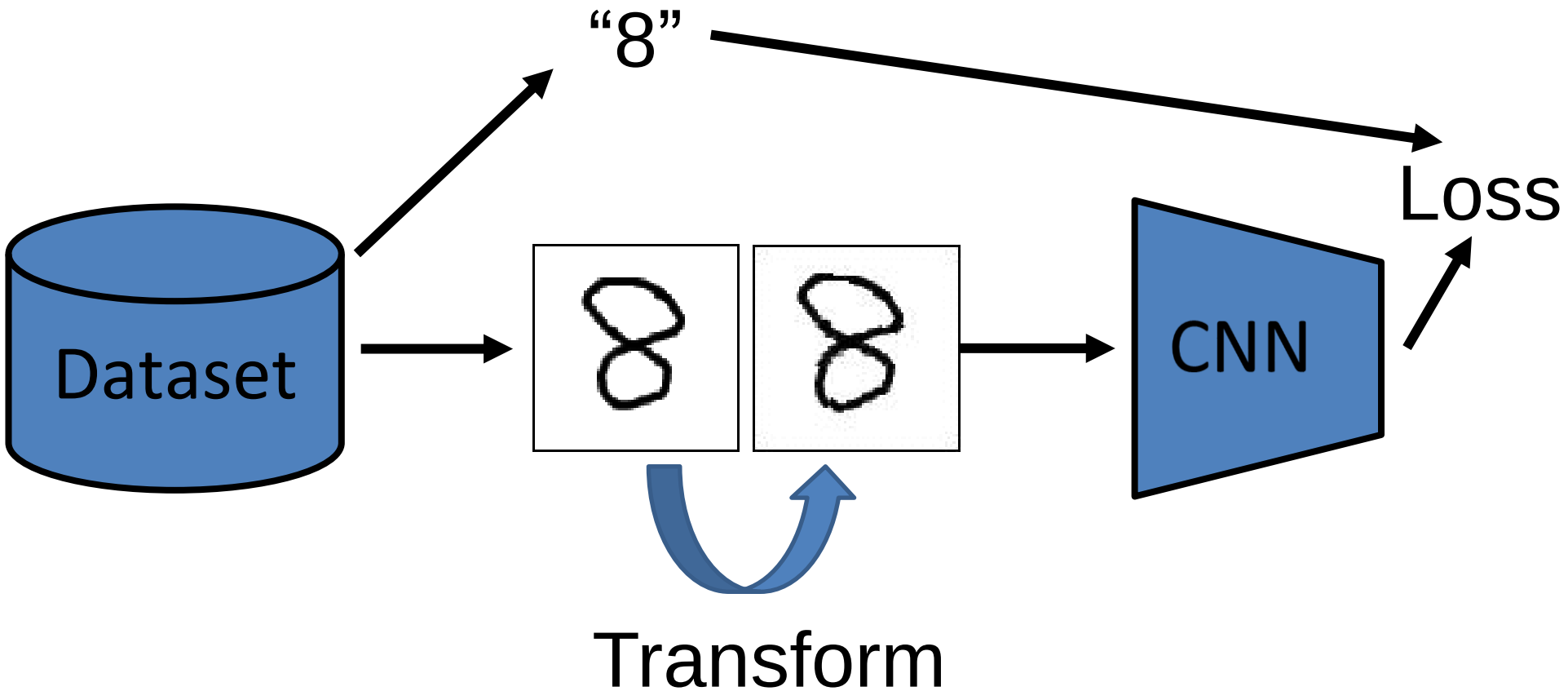
Overfitting vs Underfitting

- برای جلوگیری از overfitting و استفاده از مزایای شبکه‌های دارای توانایی یادگیری بیشتر، می‌توان مسئله‌ای که قرار است توسط شبکه حل شود را پیچیده‌تر کرد
- استفاده از dropout نمونه‌ای از این موارد است





داده‌افزایی



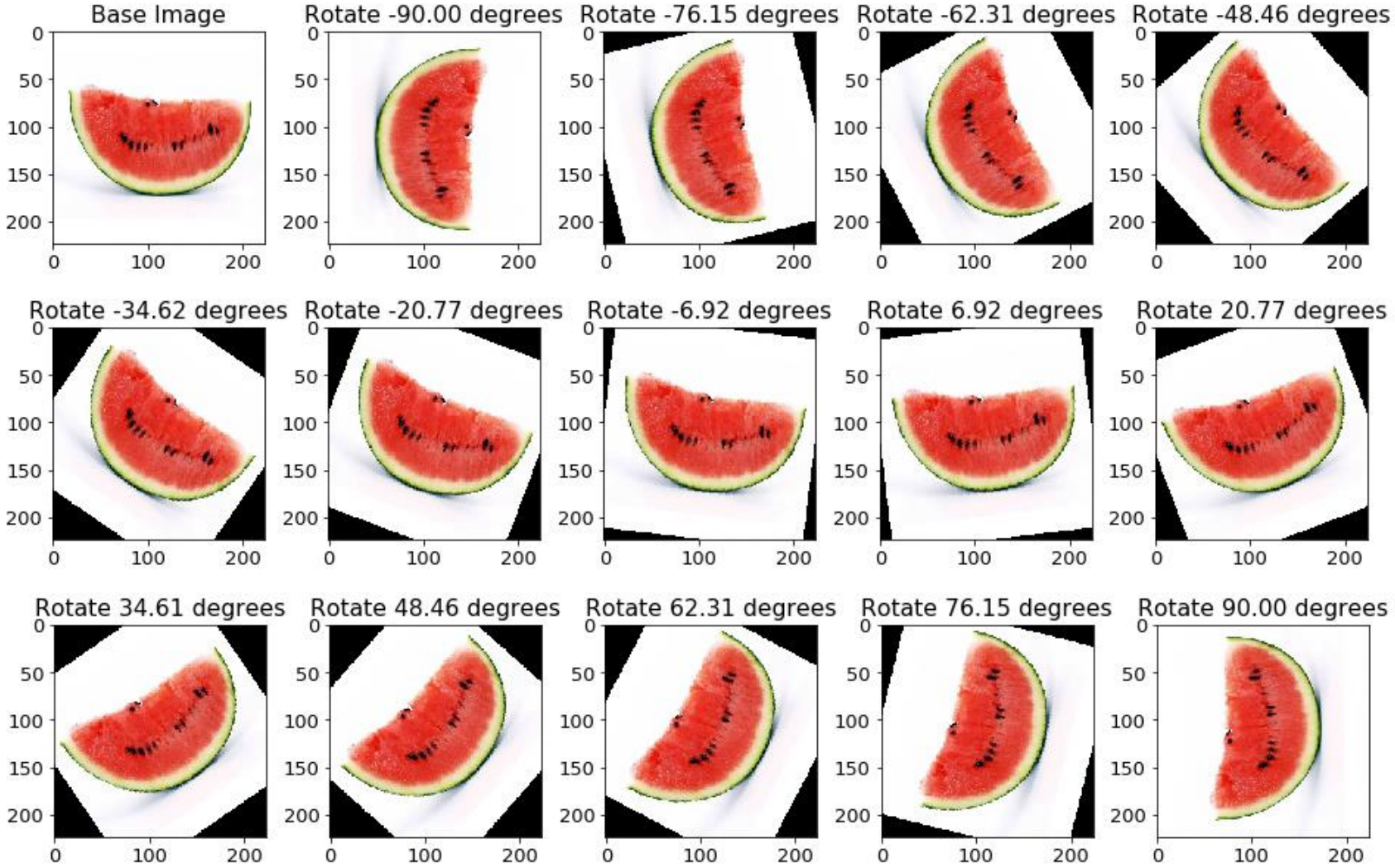
داده‌افزایی: Flip



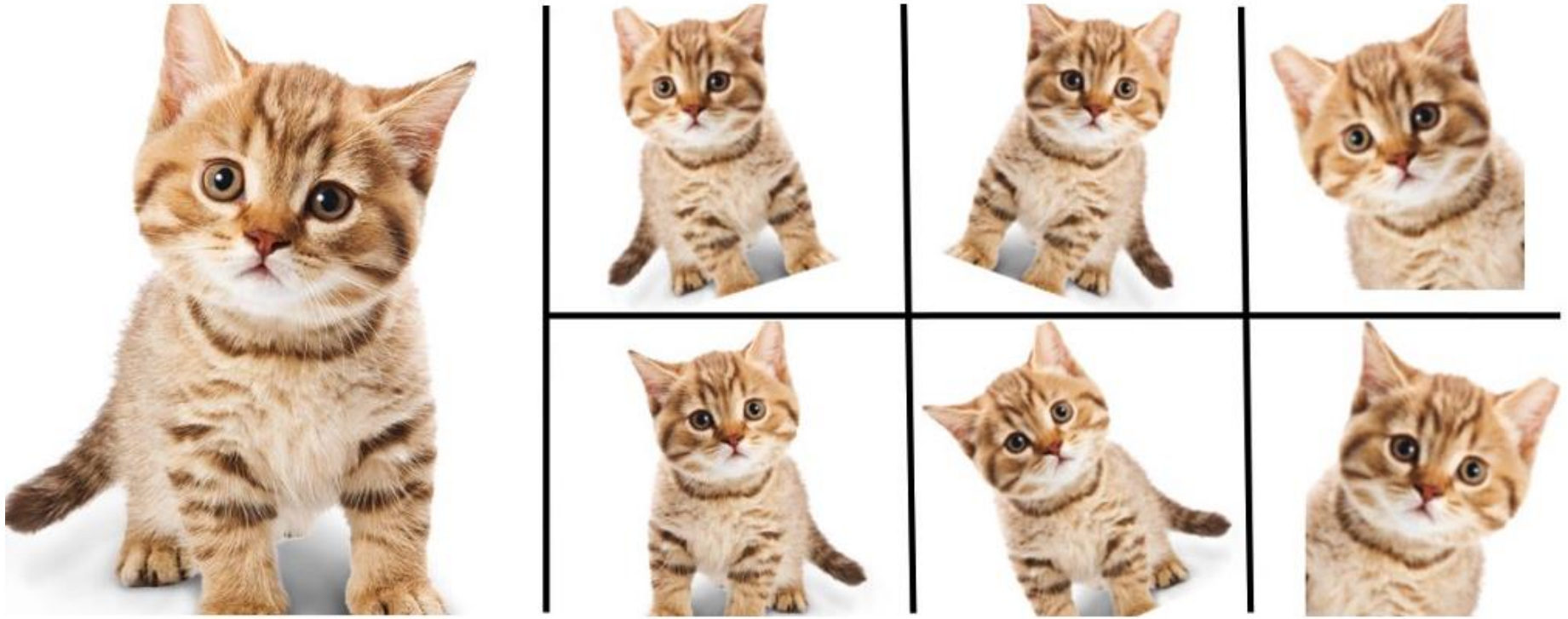

ɓ ɑ

ɐ ɡ

داده‌افزایی: چرخش

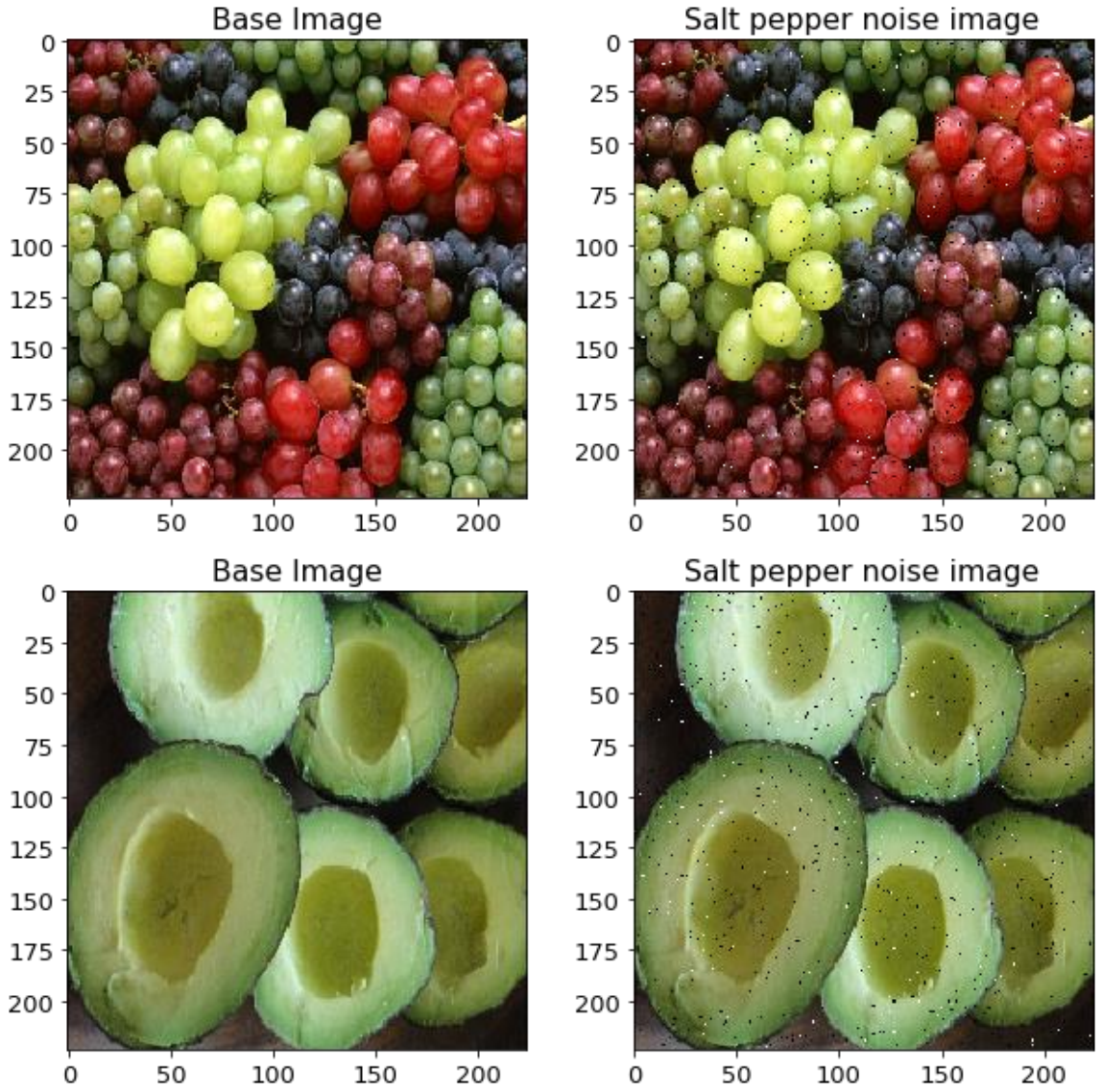


داده‌افزایی: تبدیلات هندسی

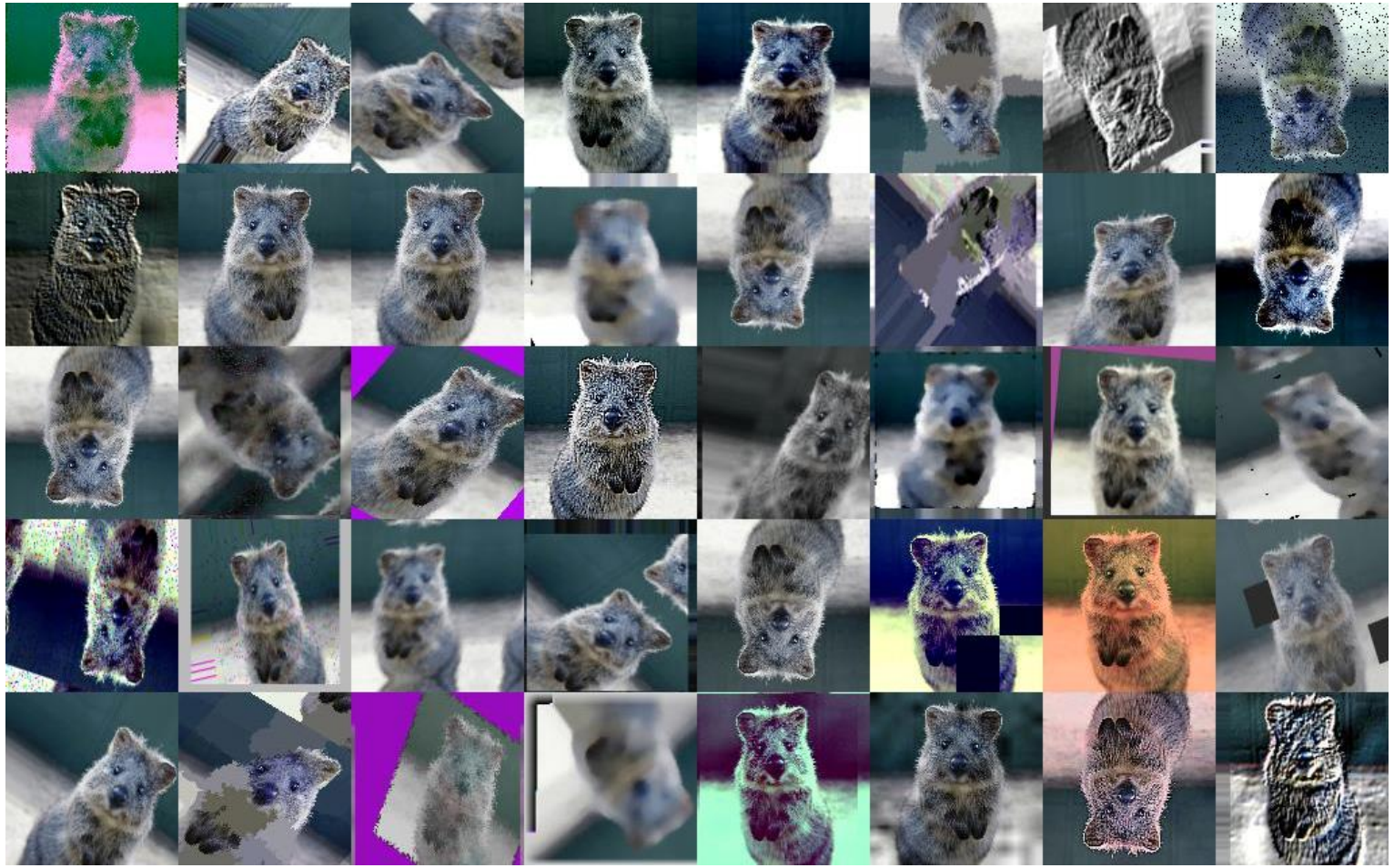


داده‌افزایی: تغییر رنگ

داده‌افزایی: افزودن نویز



داده‌افزایی

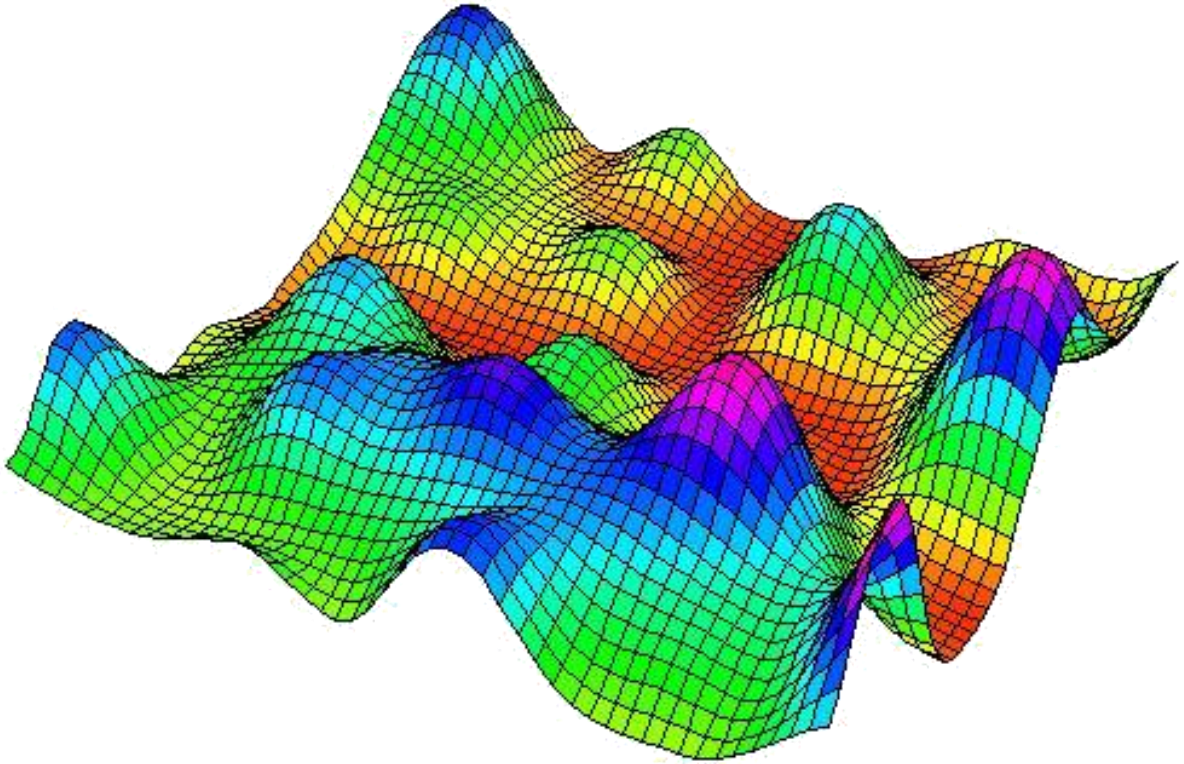


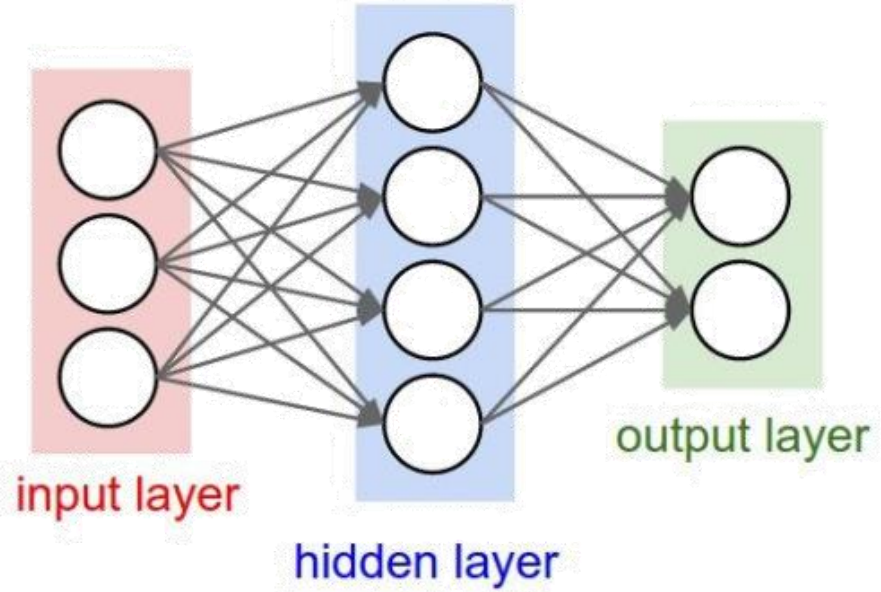
مقداردهی اولیه وزن‌ها

Weight Initialization

مقداردهی اولیه

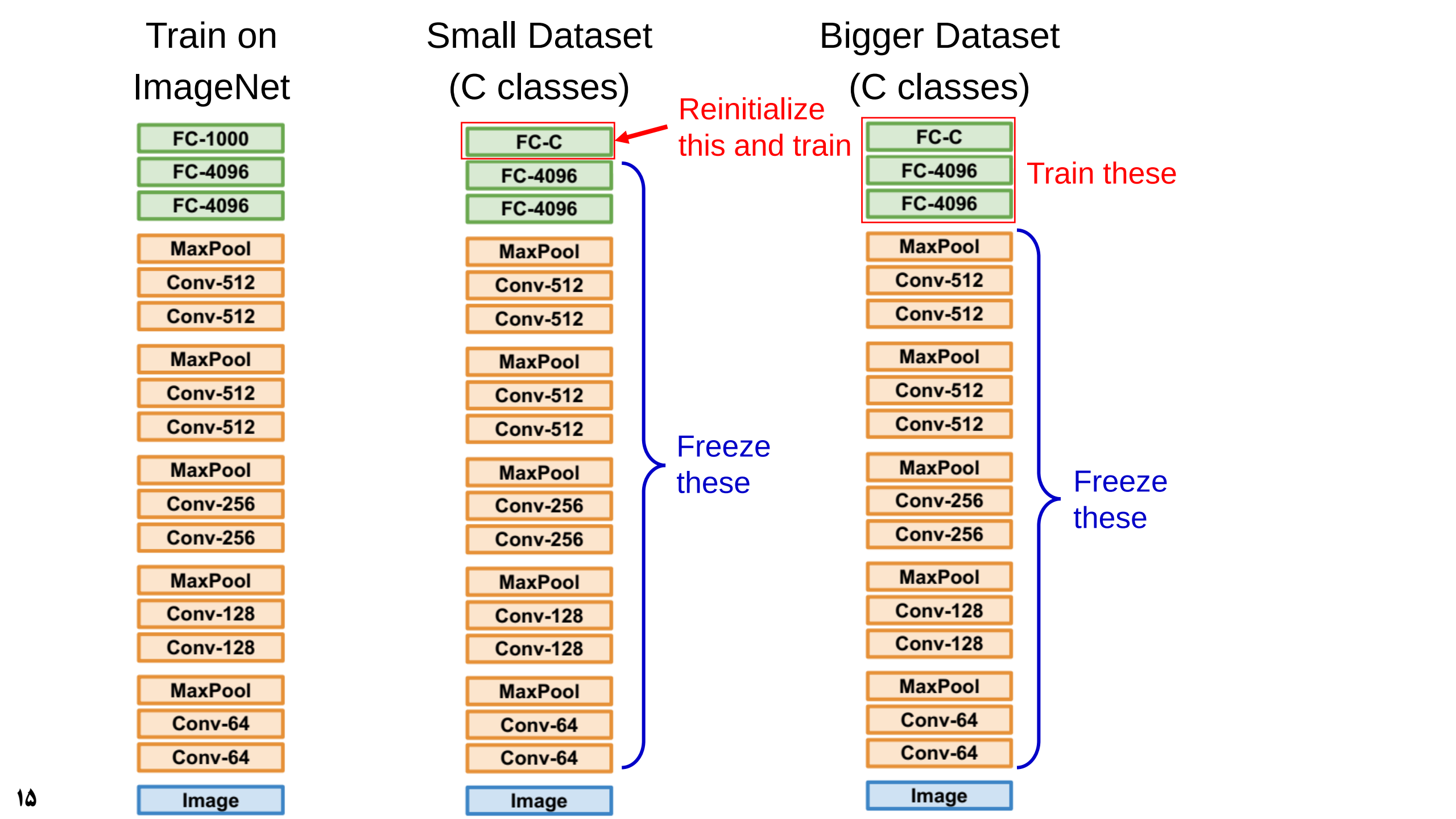
- در روش‌های بهینه‌سازی مبتنی بر تکرار، نقطه شروع بهینه‌سازی بسیار مهم است
- اگر در ابتدای کار تمام وزن‌های شبکه مقدار صفر داشته باشند چه اتفاقی می‌افتد؟
- مقدار یکسان؟

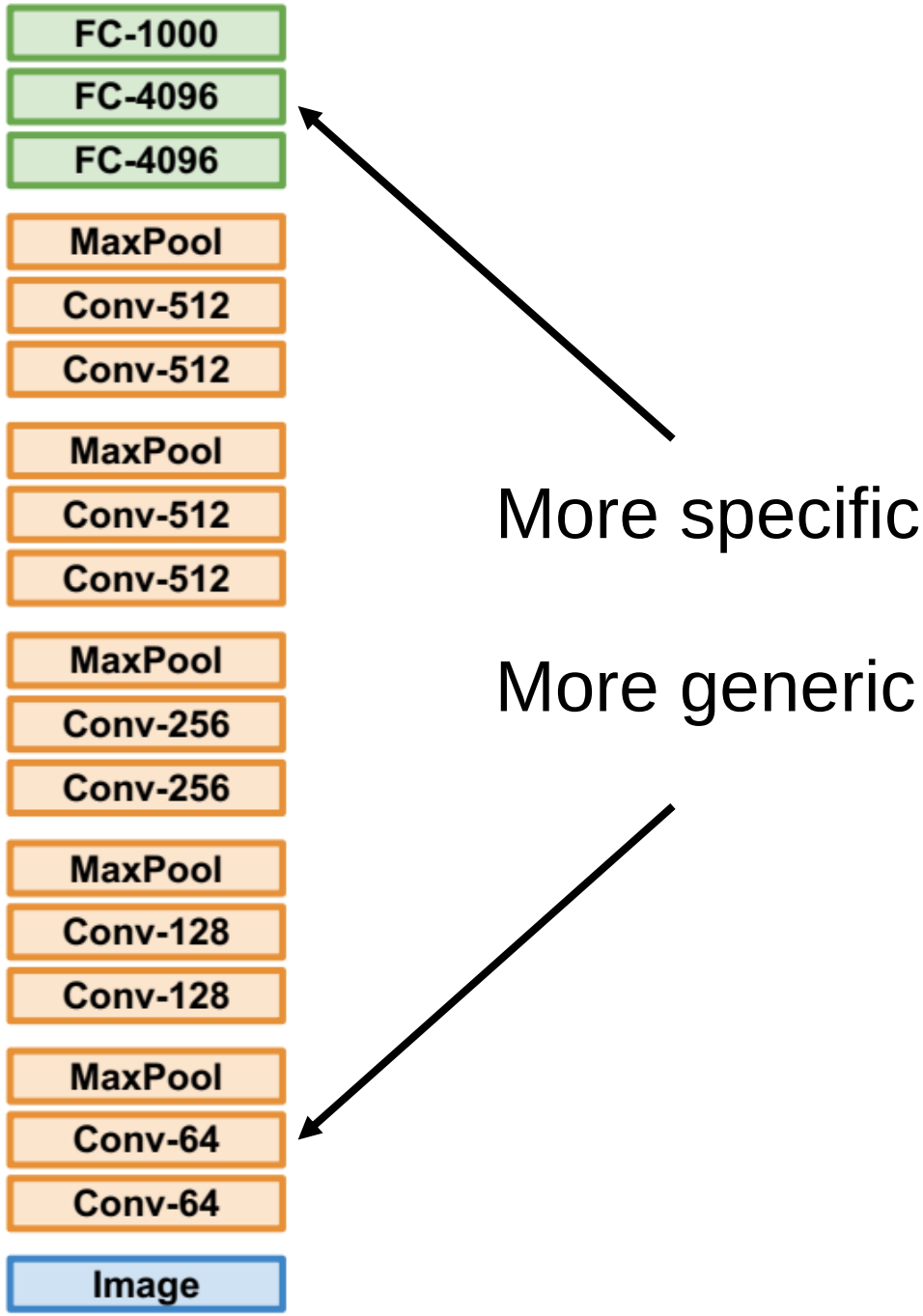




مقداردهی اولیه

- 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)
Reinitialize this and train
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
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/

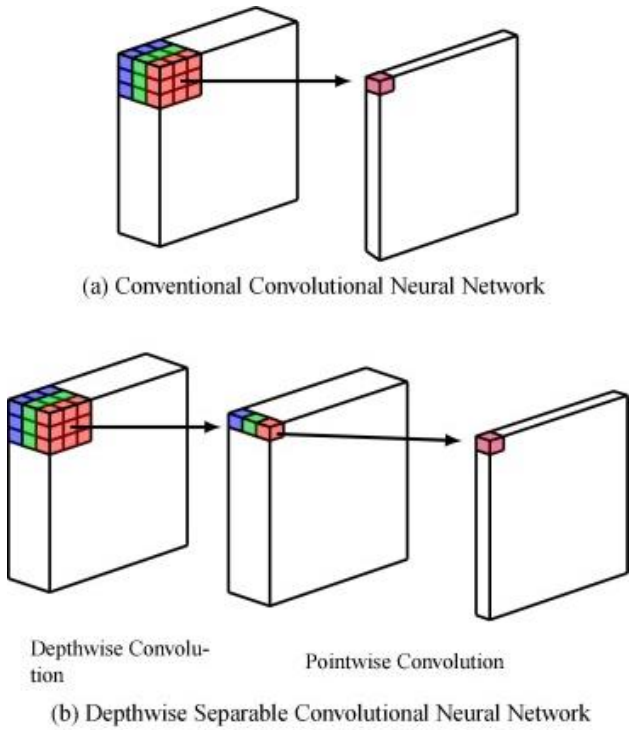
Keras: https://github.com/fchollet/deep-learning-models/releases/

برخی شبکه‌های دیگر

Some Other Networks

کانولوشن با عمق جداپذیر (Depthwise Separable)

- در کانولوشن معمولی، هر فیلتر دارای یک ابعاد مکانی است و عمق آن برابر با عمق خروجی لایه قبل است
- در بسیاری از کاربردهای پردازش تصویر مناسب است تا یک فیلتر مکانی بر روی تمام ویژگی‌های لایه قبل اعمال شوند (مانند Sobel)
- سپس می‌توان با استفاده از فیلترهای ۱×۱ آنها را ترکیب کرد
- برای داشتن ۶۴ فیلتر ۳×۳ با عمق ۳۲، هر روش چند پارامتر دارد؟



$$64 \times (3 \times 3 \times 32 + 1) = 18496$$

$$M \times (3 \times 3 + 1) + 64 \times (32 \times M + 1) = 2058M + 64$$

Chollet, François. "Xception: Deep learning with depthwise separable convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2017).

لایه کانولوشن با عمق جداپذیر در Keras

```
tf.keras.layers.SeparableConv2D(
    filters,
    kernel_size,
    strides=(1, 1),
    padding="valid",
    data_format=None,
    dilation_rate=(1, 1),
    depth_multiplier=1,
    activation=None,
    use_bias=True,
    depthwise_initializer="glorot_uniform",
    pointwise_initializer="glorot_uniform",
    bias_initializer="zeros",
    depthwise_regularizer=None,
    pointwise_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    depthwise_constraint=None,
    pointwise_constraint=None,
    bias_constraint=None,
    **kwargs
)
```

filters: Integer, the dimensionality of the output space (i.e. the number of output filters in the convolution).

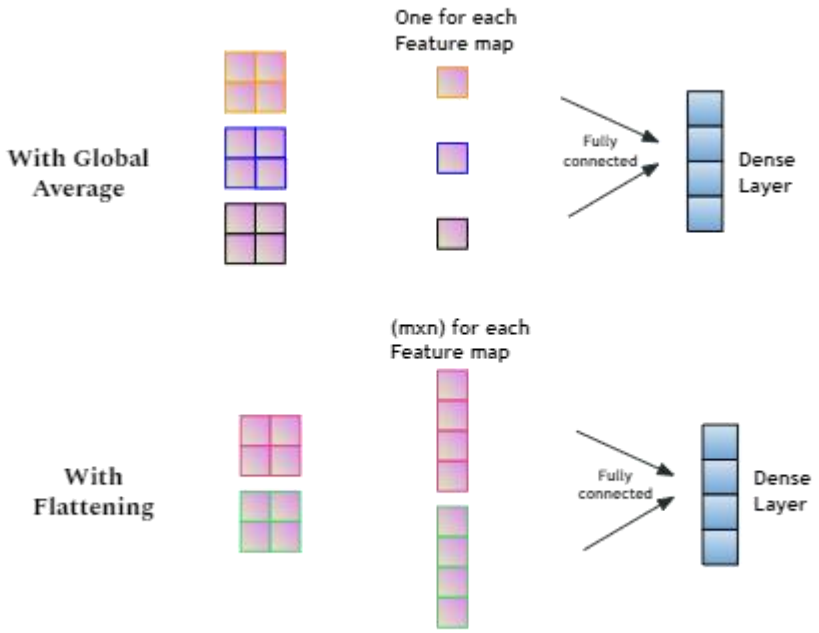
kernel_size: An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window

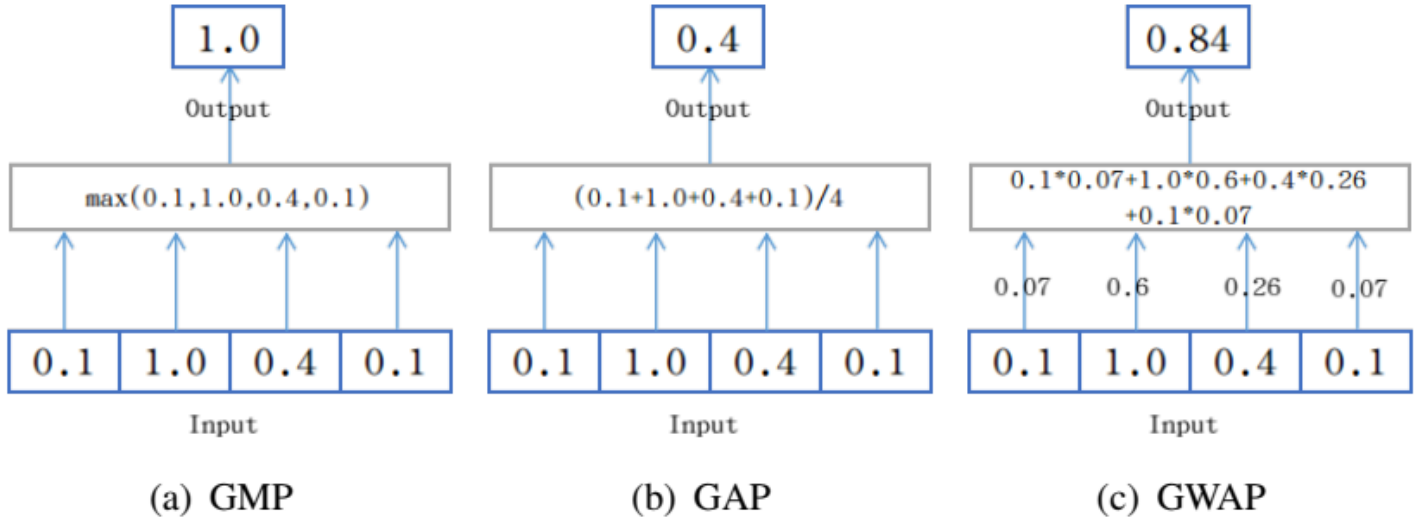
depth_multiplier: The number of depthwise convolution output channels for each input channel. The total number of depthwise convolution output channels will be equal to filters_in * depth_multiplier.

https://keras.io/api/layers/convolution_layers/separable_convolution2d/

ادغام میانگین وزن‌دار سراسری (GWAP)

- در بسیاری از شبکه‌های اخیر پس از لایه‌های کانولوشنی، از لایه GAP استفاده می‌شود
- اگرچه GAP باعث می‌شود تعداد ویژگی‌ها برای لایه بعد به مراتب کاهش بیابد، اما اطلاعات مکانی را به کلّی از بین می‌برد
- می‌توان میانگین‌گیری را به صورت وزن‌دار انجام داد

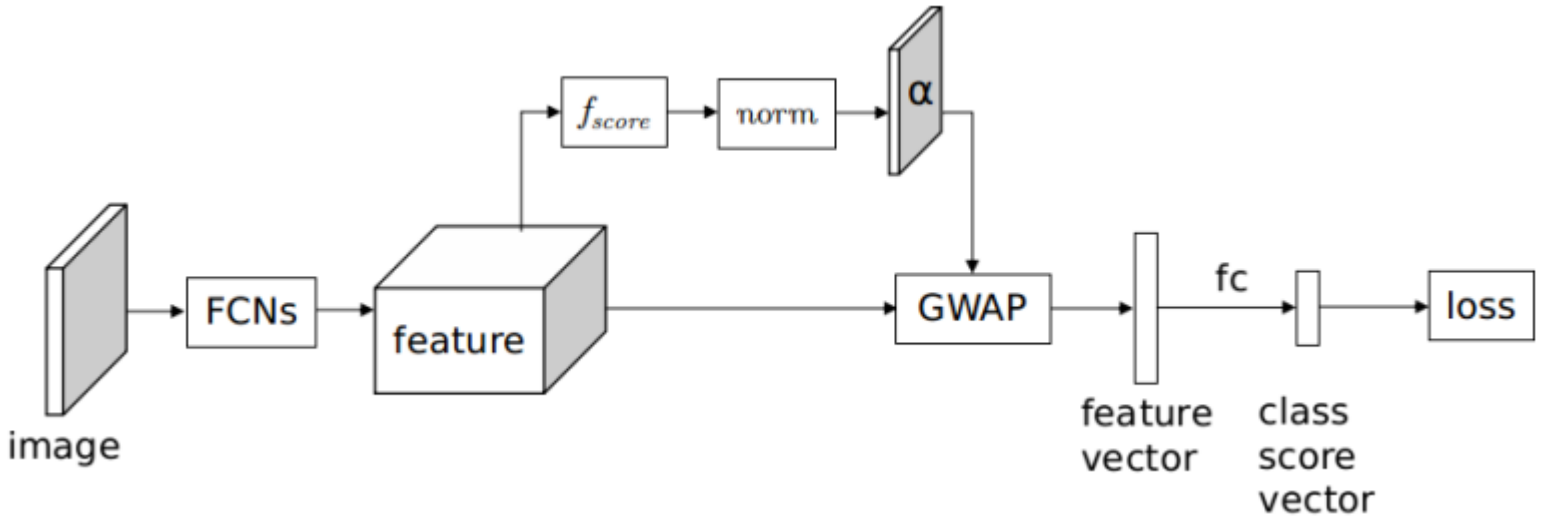




Qiu, Suo. "Global weighted average pooling bridges pixel-level localization and image-level classification." *arXiv preprint arXiv:1809.08264* (2018).

ادغام میانگین وزن‌دار سراسری (GWAP)

- در بسیاری از شبکه‌های اخیر پس از لایه‌های کانولوشنی، از لایه GAP استفاده می‌شود
- اگرچه GAP باعث می‌شود تعداد ویژگی‌ها برای لایه بعد به مراتب کاهش بیابد، اما اطلاعات مکانی را به کلّی از بین می‌برد
- می‌توان میانگین‌گیری را به صورت وزن‌دار انجام داد



Qiu, Suo. "Global weighted average pooling bridges pixel-level localization and image-level classification." *arXiv preprint arXiv:1809.08264* (2018).

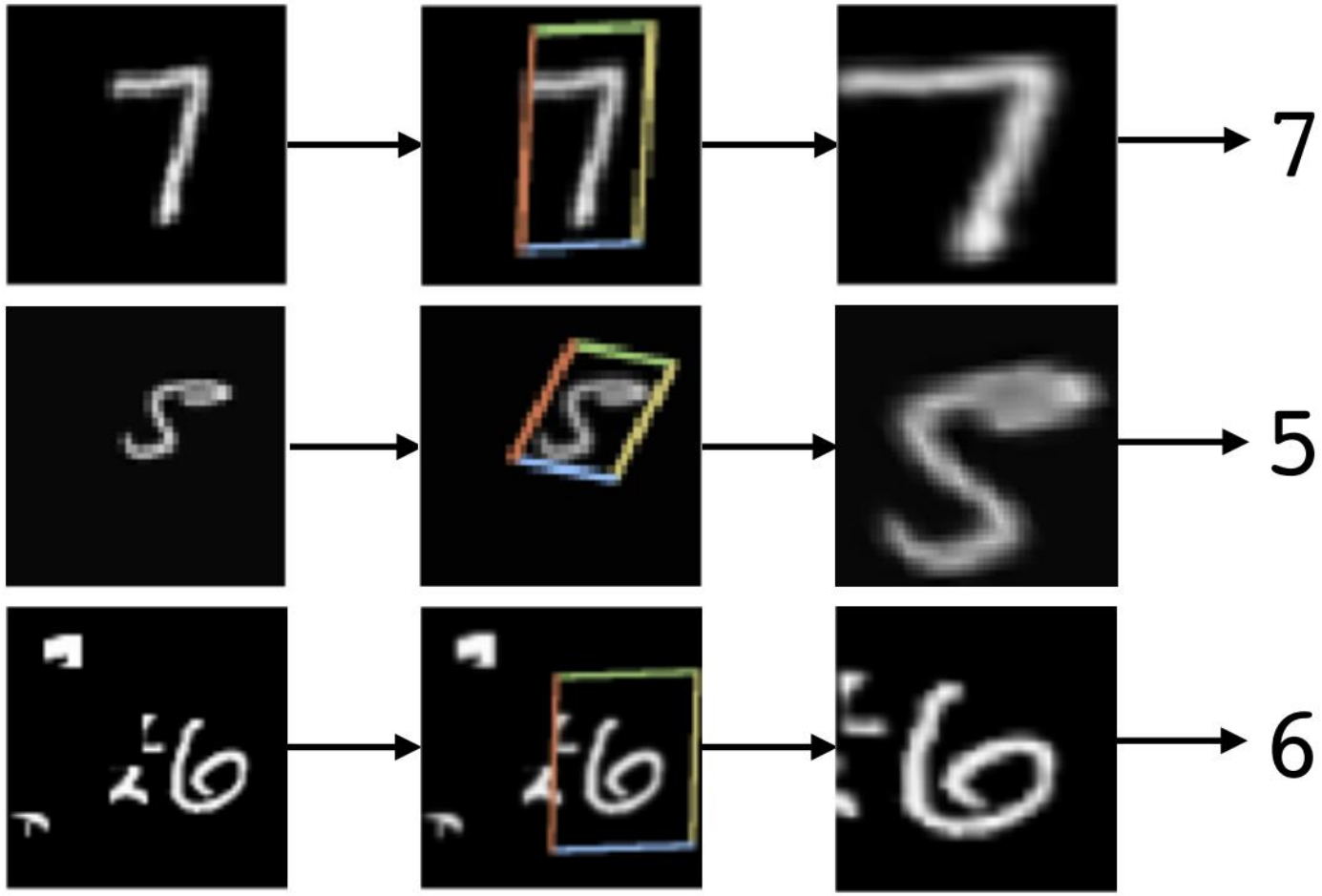
Spatial Transformer

- اگر شیئ مورد نظر به درستی در مرکز تصویر قرار نگرفته باشد، وظیفه شبکه برای دسته‌بندی آن پیچیده خواهد بود
- شبکه‌های کانولوشنی بسیار عمیق با داده‌های آموزشی زیاد و/یا با داده‌افزایی هندسی بسیار زیاد ممکن است بتوانند مدل مستقل از مکان را با هزینه زیاد آموزش ببینند
- اگر بتوانیم مکان شیئ را به خوبی تخمین بزنیم و دور شیئ را به خوبی برش بزنیم، مسئله دسته‌بندی برای شبکه بسیار ساده‌تر خواهد شد
- توسعه الگوریتمی که بتواند مکان شیئ را به درستی مشخص کنید کار ساده‌ای نیست
- چطور می‌توان نحوه برش زدن مناسب را آموخت؟
- می‌توانیم یک لایه طراحی کنیم که پارامترهای برش را تخمین بزند

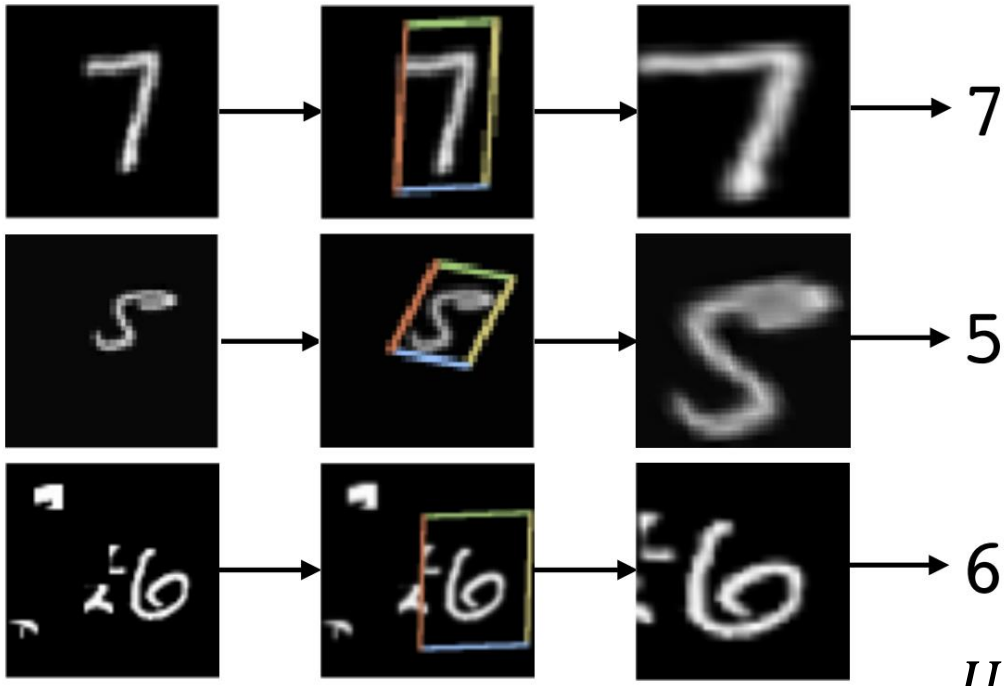
Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." *Advances in neural information processing systems*. (2015).

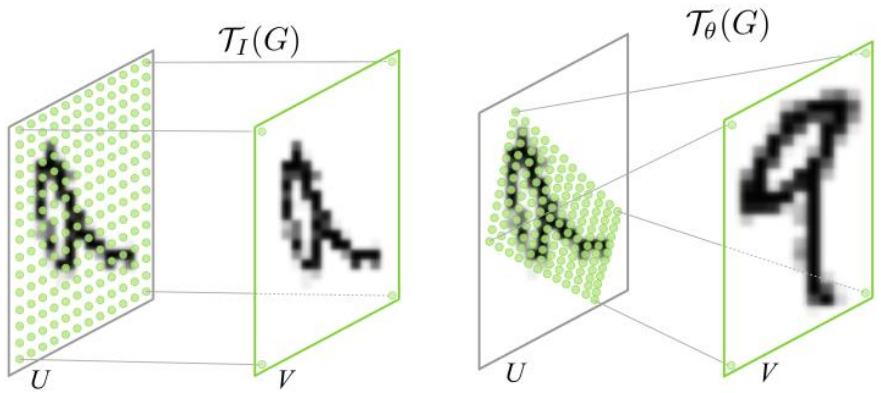
Spatial Transformer

- بعد از آنکه تبدیل مکانی به خوبی انجام شد، می‌توان یک شبکه ساده برای دسته‌بندی قرار داد



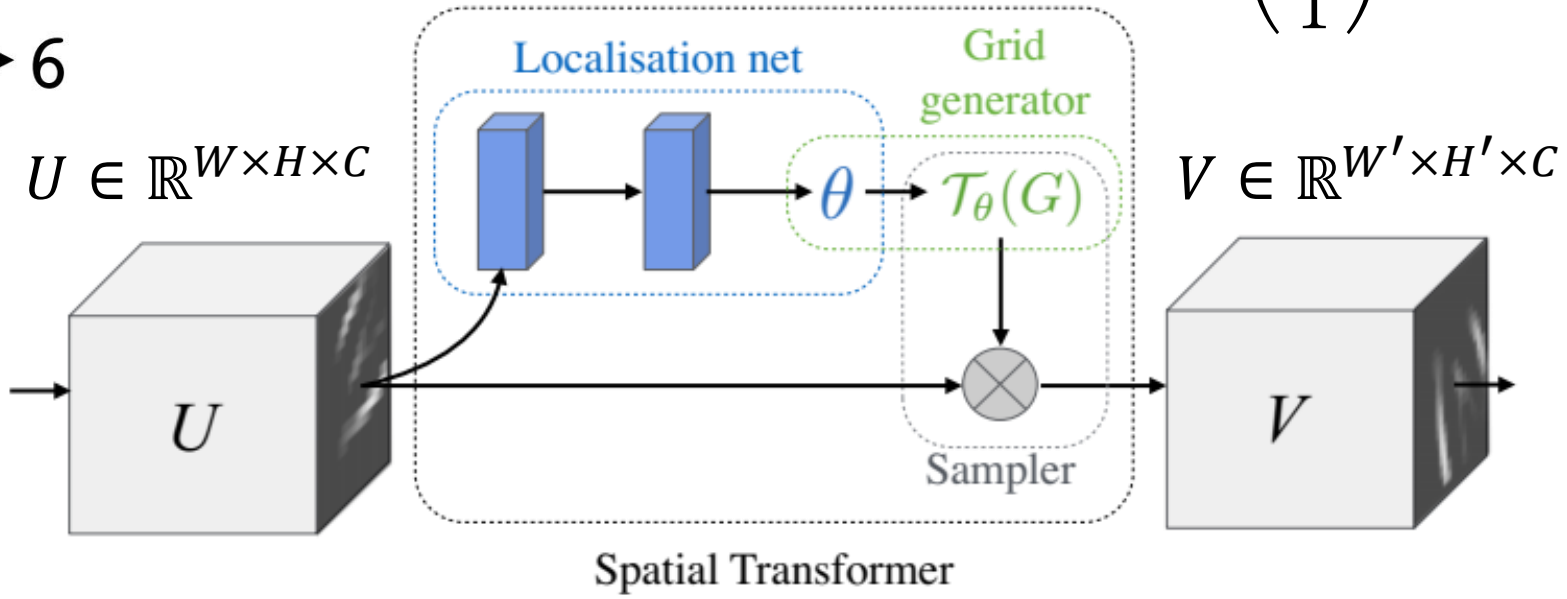
Spatial Transformer





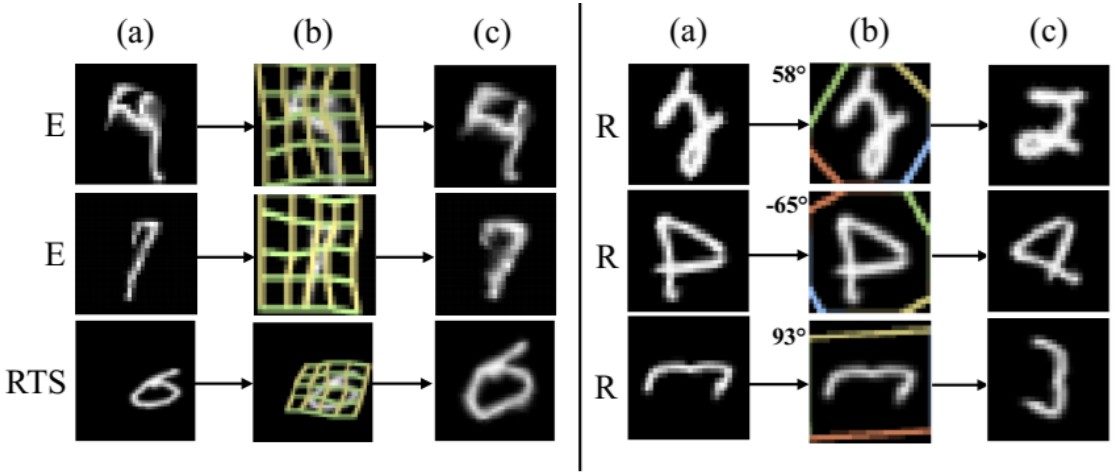
- برای داشتن قابلیت بهینه‌سازی لازم است تا هر لایه‌ای که تعریف می‌کنیم نسبت به پارامترهای آن مشتق‌پذیر باشد

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

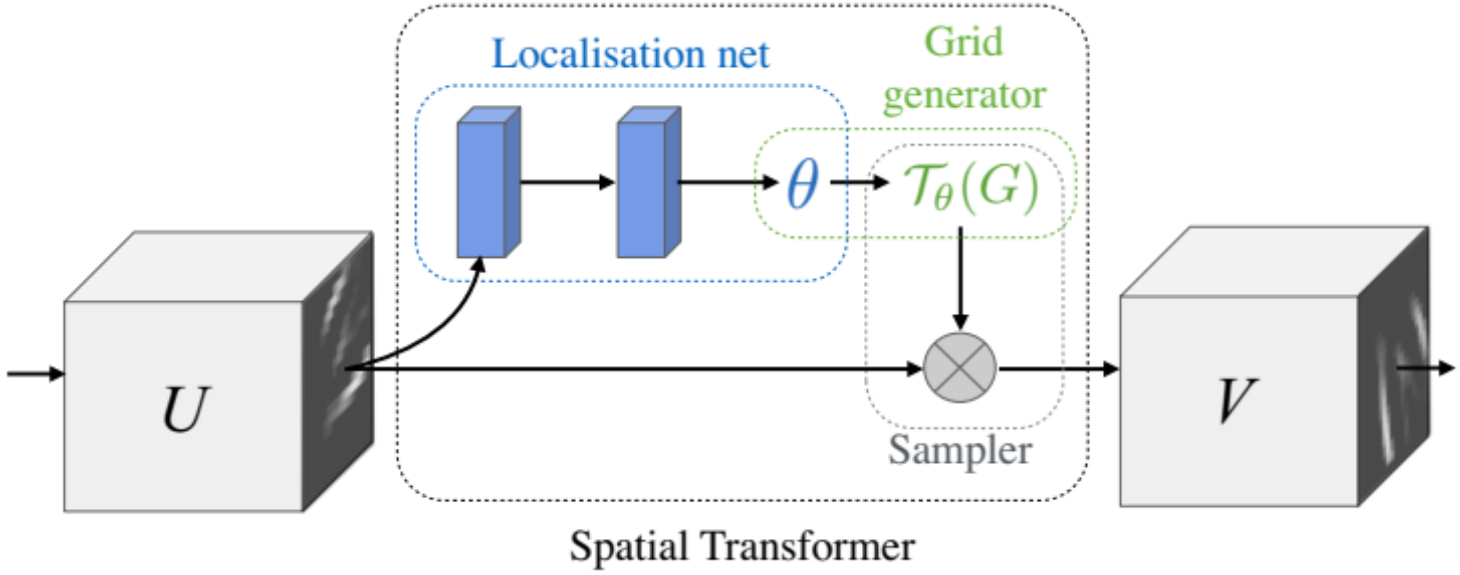


Spatial Transformer

Model		MNIST Distortion R	RTS	P	E
FCN		2.1	5.2	3.1	3.2
CNN		1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5	2.7
	Proj	1.3	0.9	1.4	2.6
	TPS	1.1	0.8	1.4	2.4
ST-CNN	Aff	0.7	0.5	0.8	1.2
	Proj	0.8	0.6	0.8	1.3
	TPS	0.7	0.5	0.8	1.1



R: rotated
RTS: rotated, translated, scaled
P: projective
E: elastic
Aff: affine transformation
Proj: projective transformation
TPS: 16-point thin plate spline transformation



Spatial Transformer

- بخش مربوط به تبدیل مکانی نیز مشابه با لایه‌های دیگر آموزش می‌بیند و تابع ضرر جداگانه‌ای برای این بخش تعریف نمی‌شود
- با بهینه‌سازی پارامترهای شبکه برای دسته‌بندی، پارامترهای مربوط به بخش تبدیل مکانی نیز آموزش می‌بینند