Convolutional Neural Network (CNN) and Generative Adversarial Networks (GAN)

Debashis Sen

Department of Electronics &
Electrical Communication Engineering
Indian Institute of Technology – Kharagpur

dsen@ece.iitkgp.ac.in
http://www.facweb.iitkgp.ac.in/~debashis

Content:

- NN to CNN
- CNN:
  - Operations & Activation
  - Architectures
- Adversarial Attack
- GAN Framework
- Deep Convolutional GAN
- GAN Variants
  - Conditional GAN
  - Patch GAN
  - Cycle GAN
  - InfoGAN
  - Bidirectional GAN
  - RealnessGAN

Neural Network to Convolutional Neural Network

1. Local Connections
2. Weight Sharing

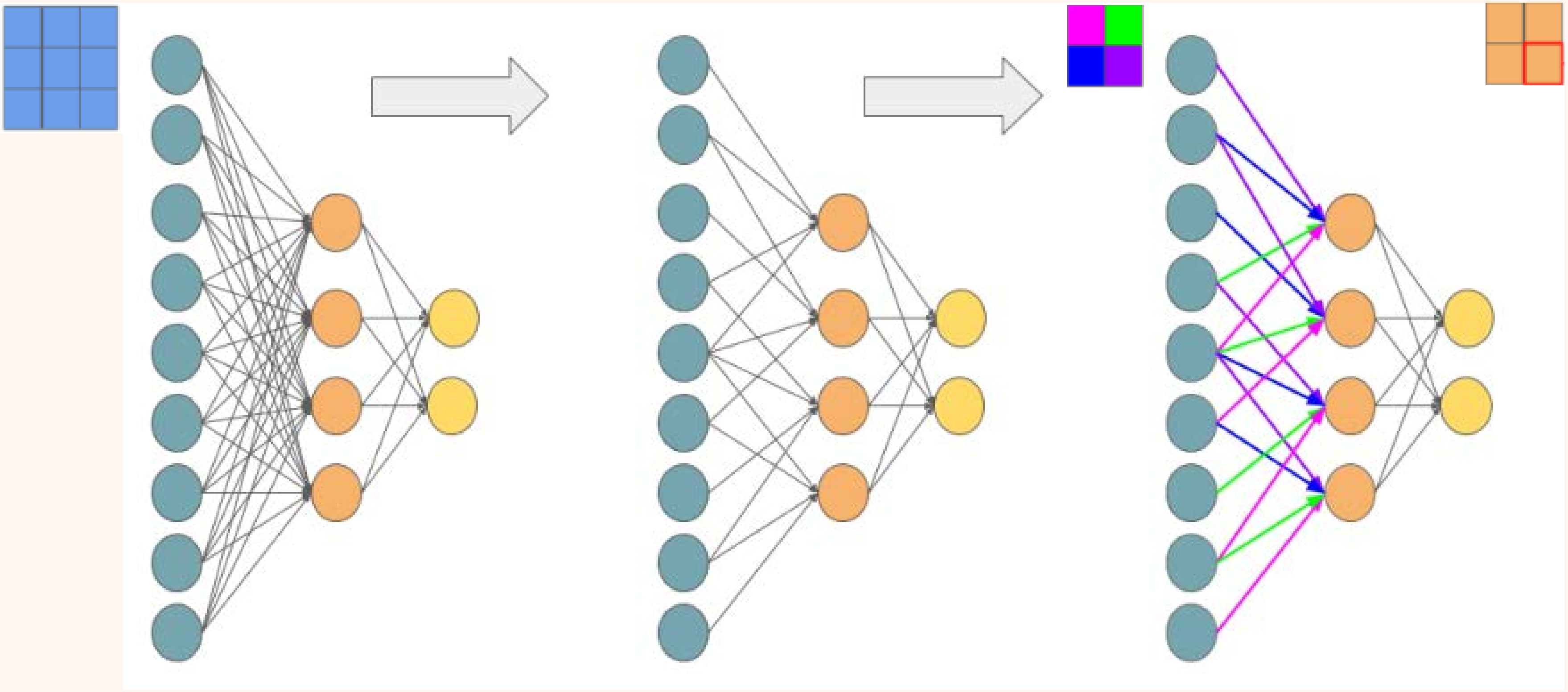
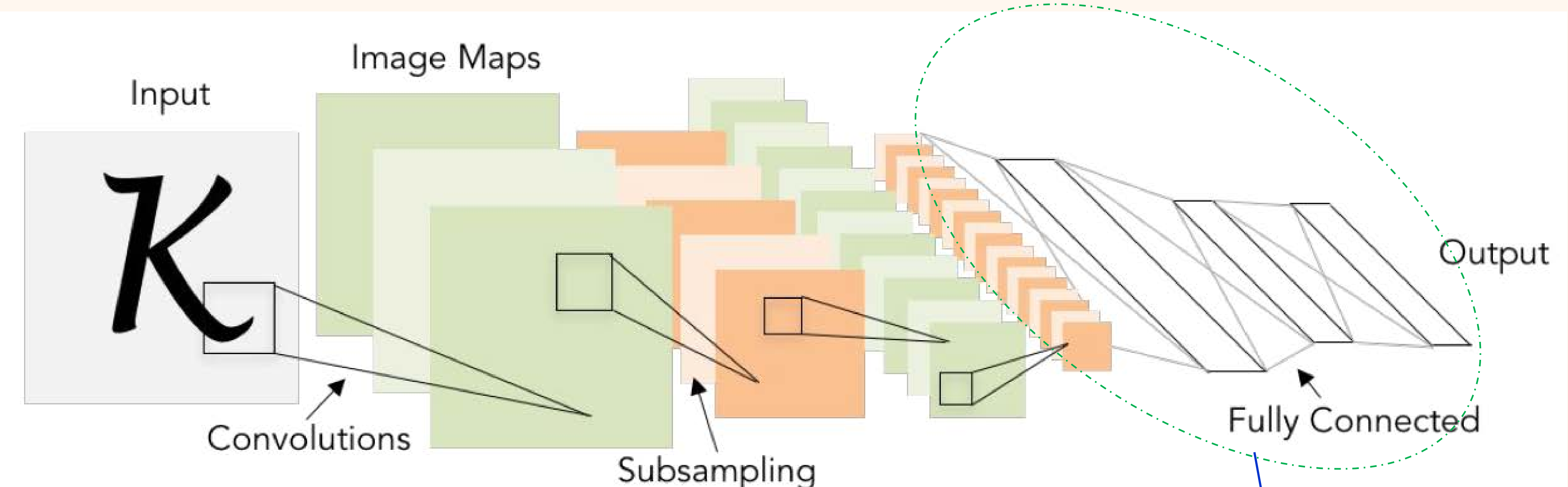


Image source:
https://numpydl.readthedocs.io/en/latest/tutorials/CNN/

Convolutional Neural Network



Learning Features

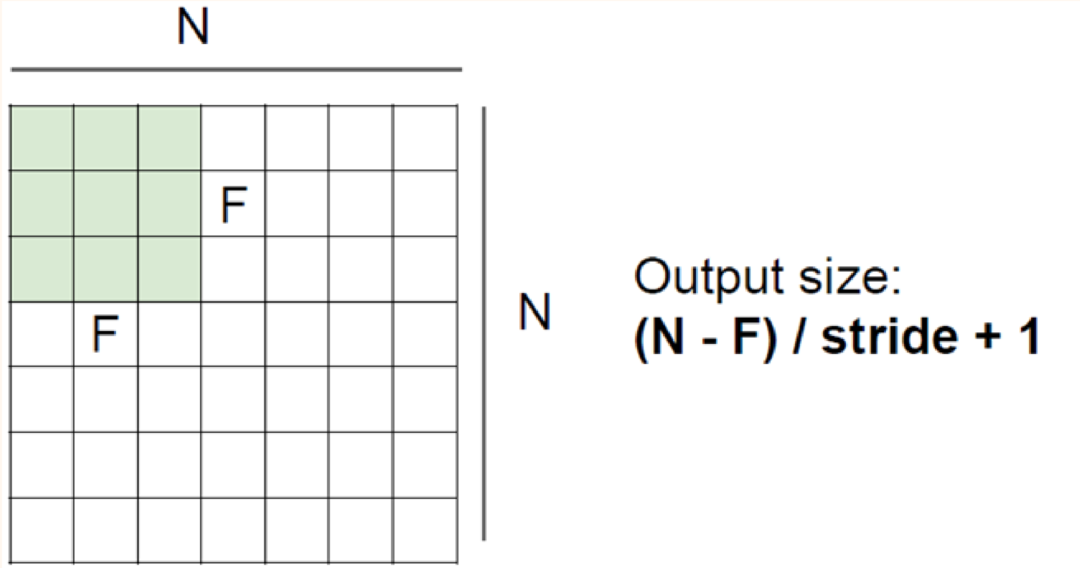
Classification

LeNet-5
1998

Focus on end to end learning!

Image source: http://cs231n.stanford.edu/2017/

Strides:



- Usually after zero padding

Pooling (Max-pooling):

- Makes representations “manageable”
- Introduces 0 parameters
- No zero padding

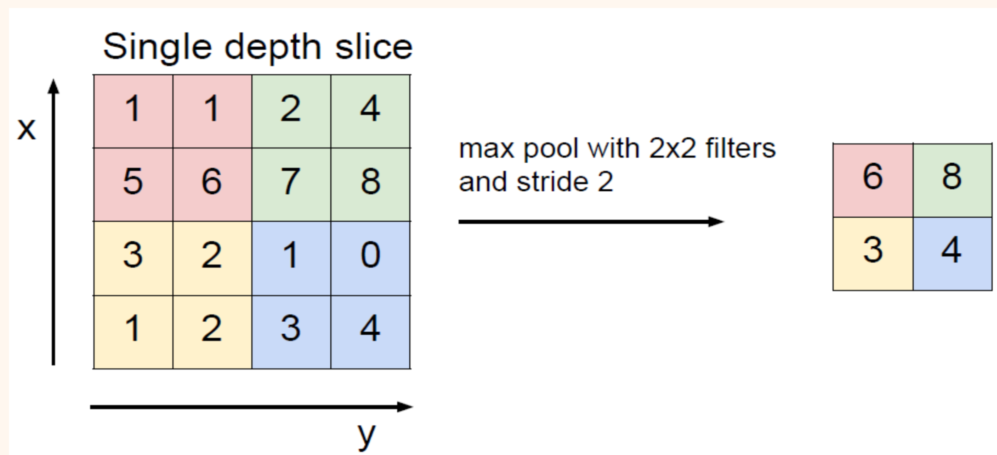
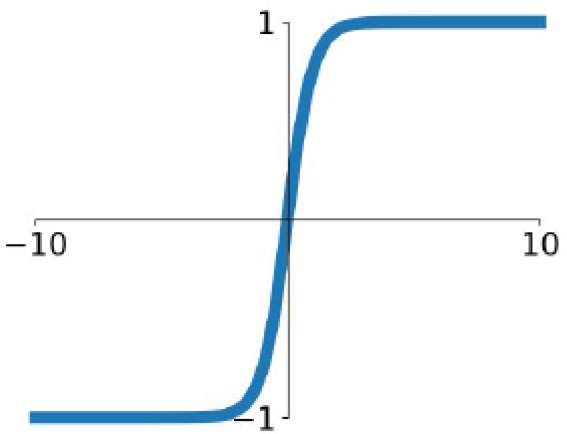
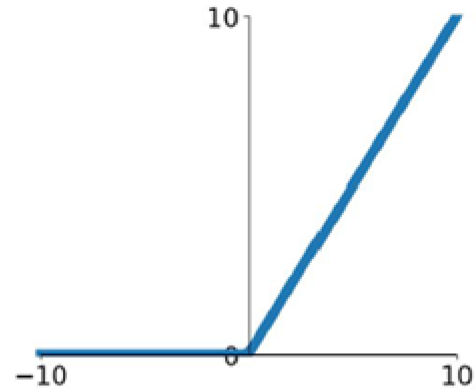


Image source: http://cs231n.stanford.edu/2017/



tanh(x)

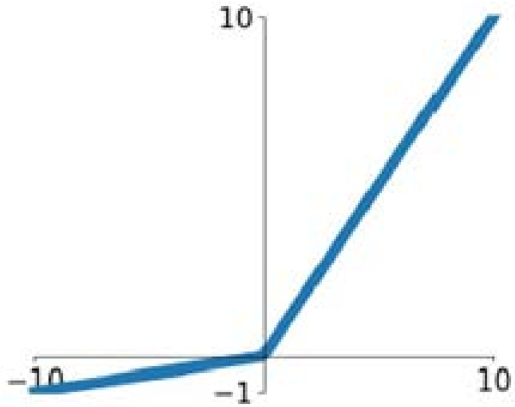


ReLU

(Rectified Linear Unit)

f(x) = max(0,x)

Leaky ReLU



$$f(x) = \max(0.01x, x)$$

Parametric Rectifier (PReLU)

$$f(x) = \max(\alpha x, x)$$

backprop into \alpha
(parameter)

MAXout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

Image source: http://cs231n.stanford.edu/2017/

Some popular CNN architectures

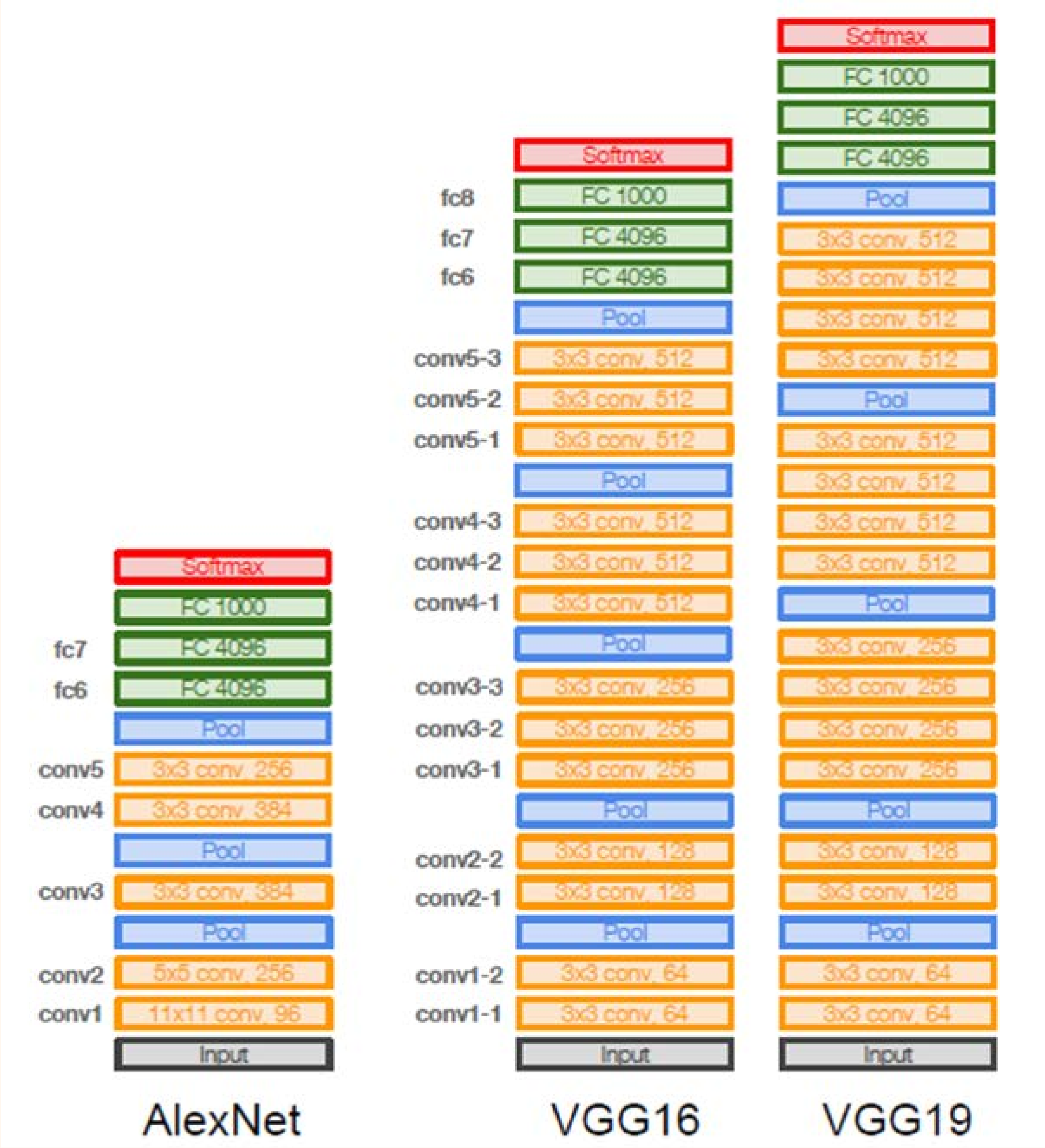
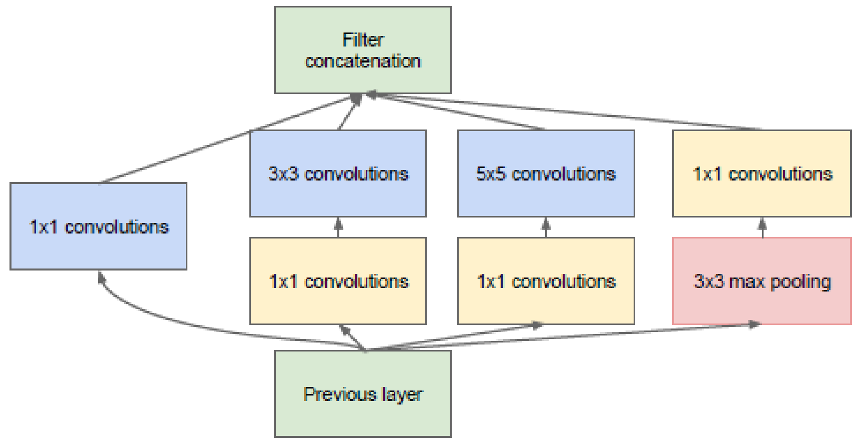


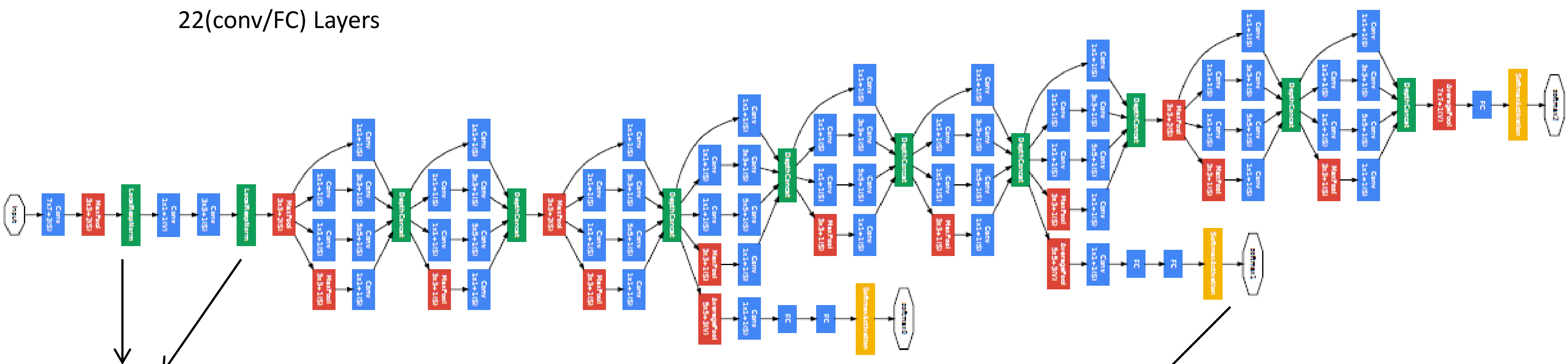
Image source: http://cs231n.stanford.edu/2017/

GoogLeNet



(b) Inception module with dimensionality reduction

22(conv/FC) Layers

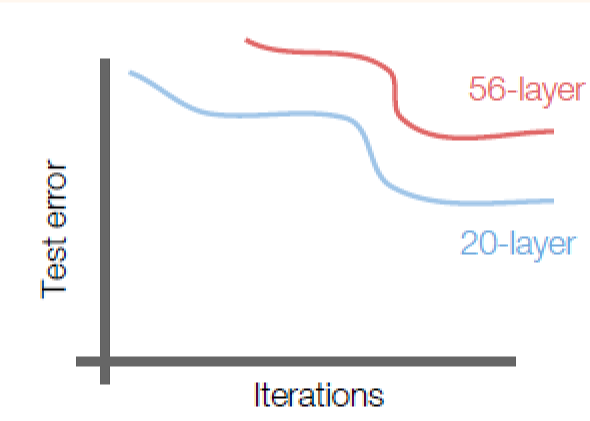


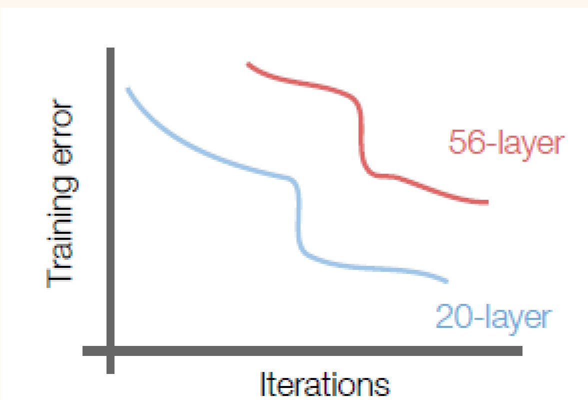
Local Response Normalization

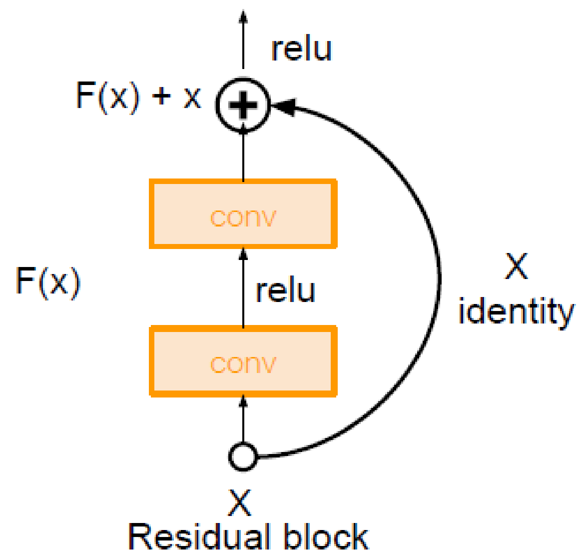
Auxiliary errors

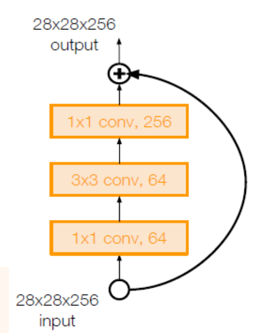
Image source: http://cs231n.stanford.edu/2017/

ResNet









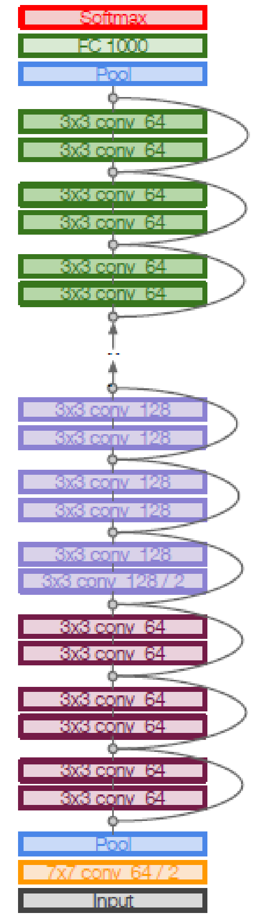


Image source: http://cs231n.stanford.edu/2017/

Densely Connected Convolutional Networks

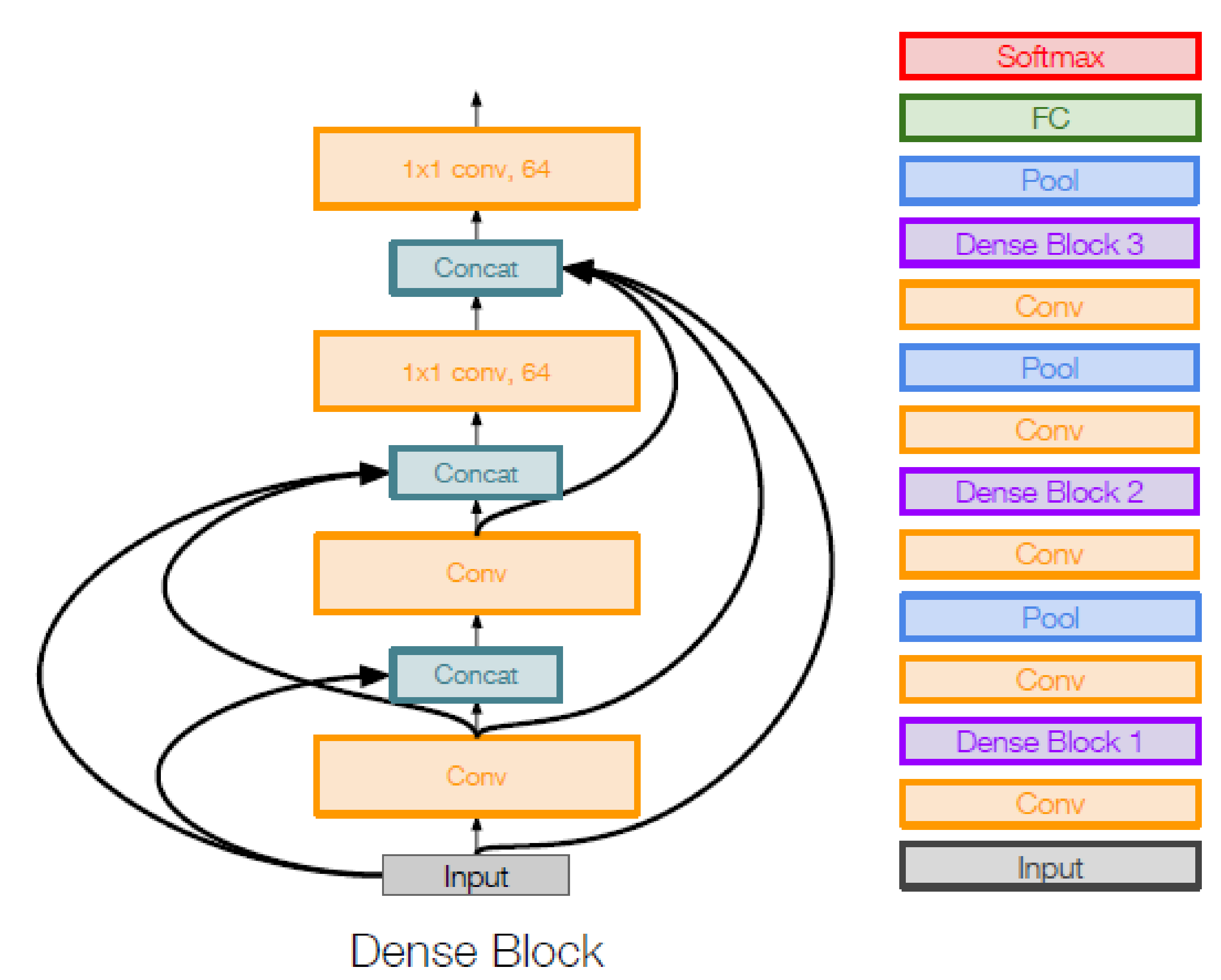


Image source: http://cs231n.stanford.edu/2017/

U-Net:

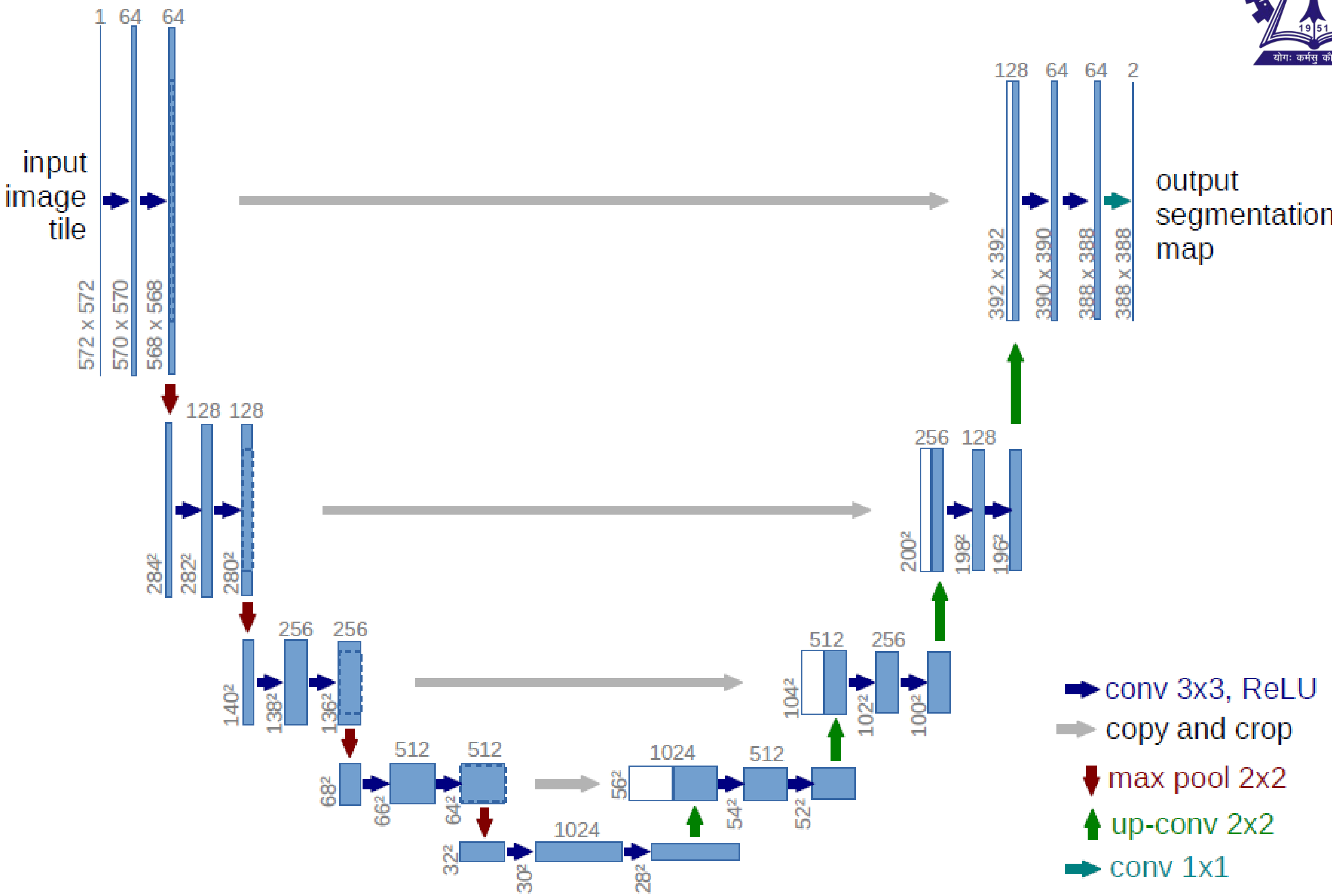


Image source: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

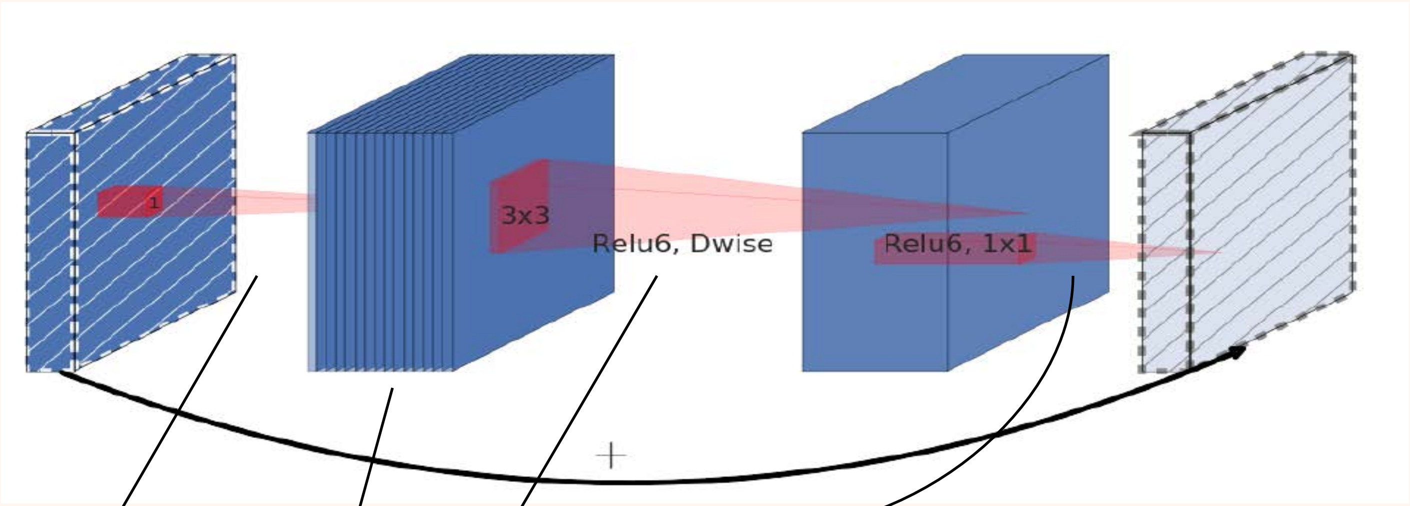
EfficientNet:

EfficientNet-B0 baseline network

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

mobile inverted bottleneck MBConv

Table source: https://arxiv.org/abs/1905.11946



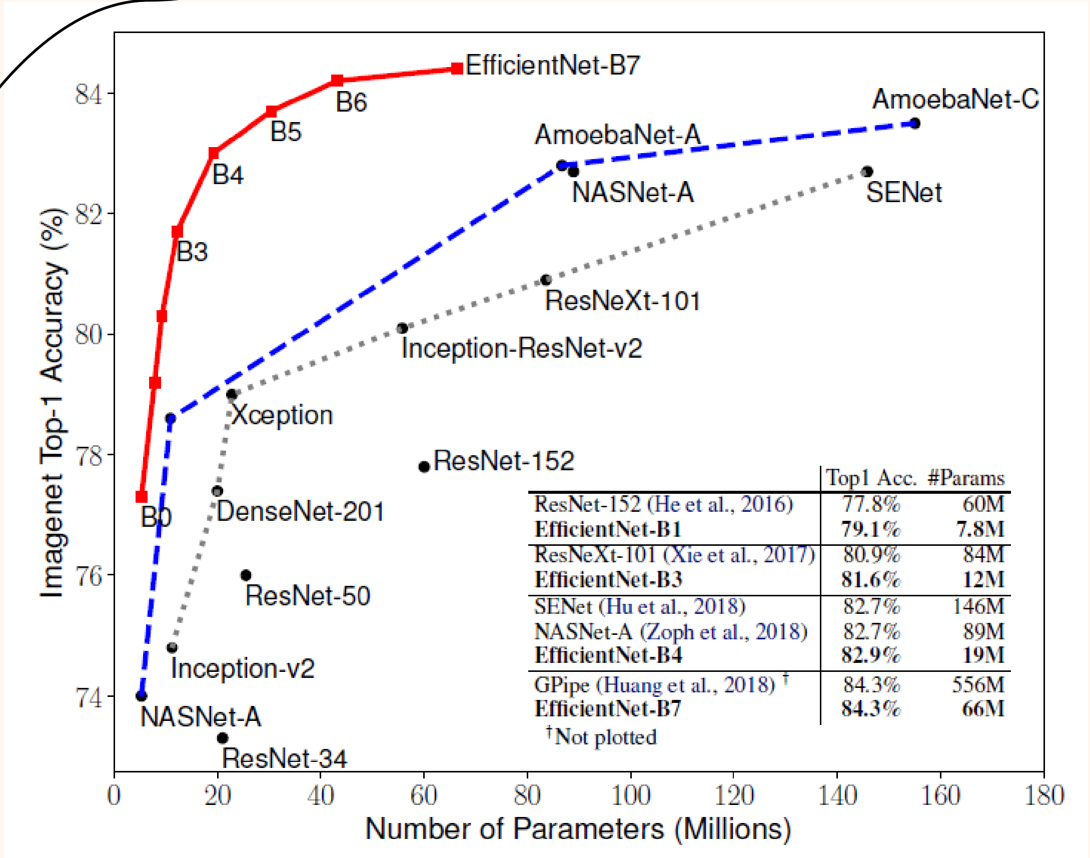
MBconv

Depth wise separable convolution

Expansion

Feature Computation

Compression

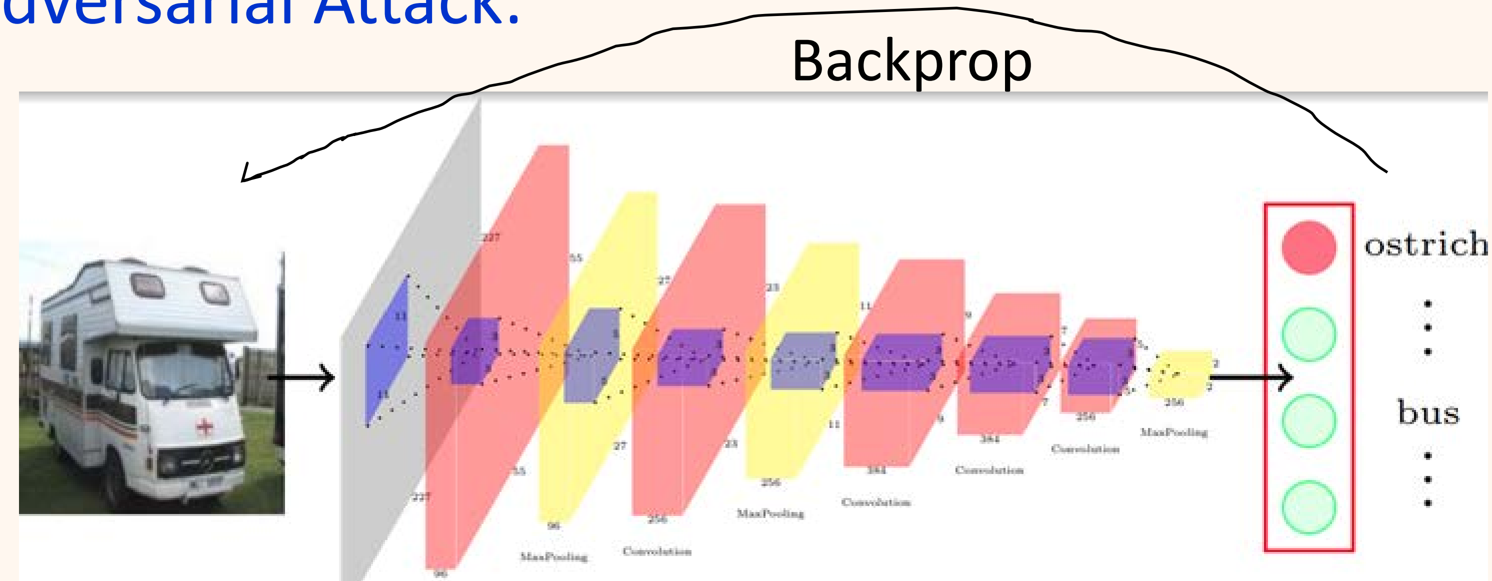


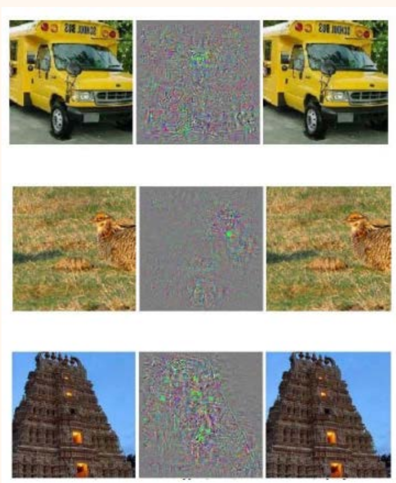
	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

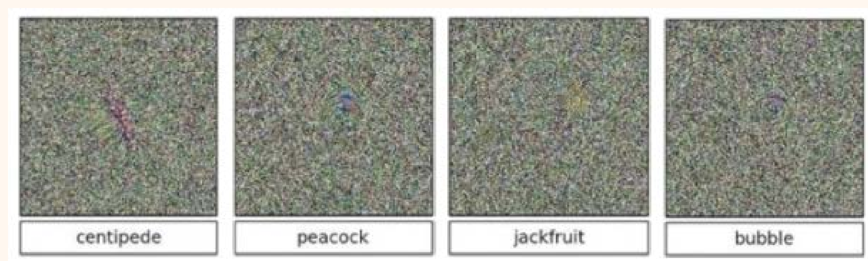
†Not plotted

Image source:
https://arxiv.org/abs/1905.11946,
https://arxiv.org/abs/1801.04381

Adversarial Attack:







Easily fooling NNs!

Reason: Boundaries in very High Dimensional Real Space!

Image Sources:

- http://cs231n.stanford.edu/ [*Top*]
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow and Rob Fergus, Intriguing properties of neural networks. ICLR (Poster) 2014. *[Bottom Left]*
- Anh Mai Nguyen, Jason Yosinski and Jeff Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, CVPR 2015: 427-436. *[Bottom Right]*

GAN:

Two players competing against each other!

Tries to fool the discriminator by generating real-like data

Tries to detect the mischief of the generator differentiating fake from real.

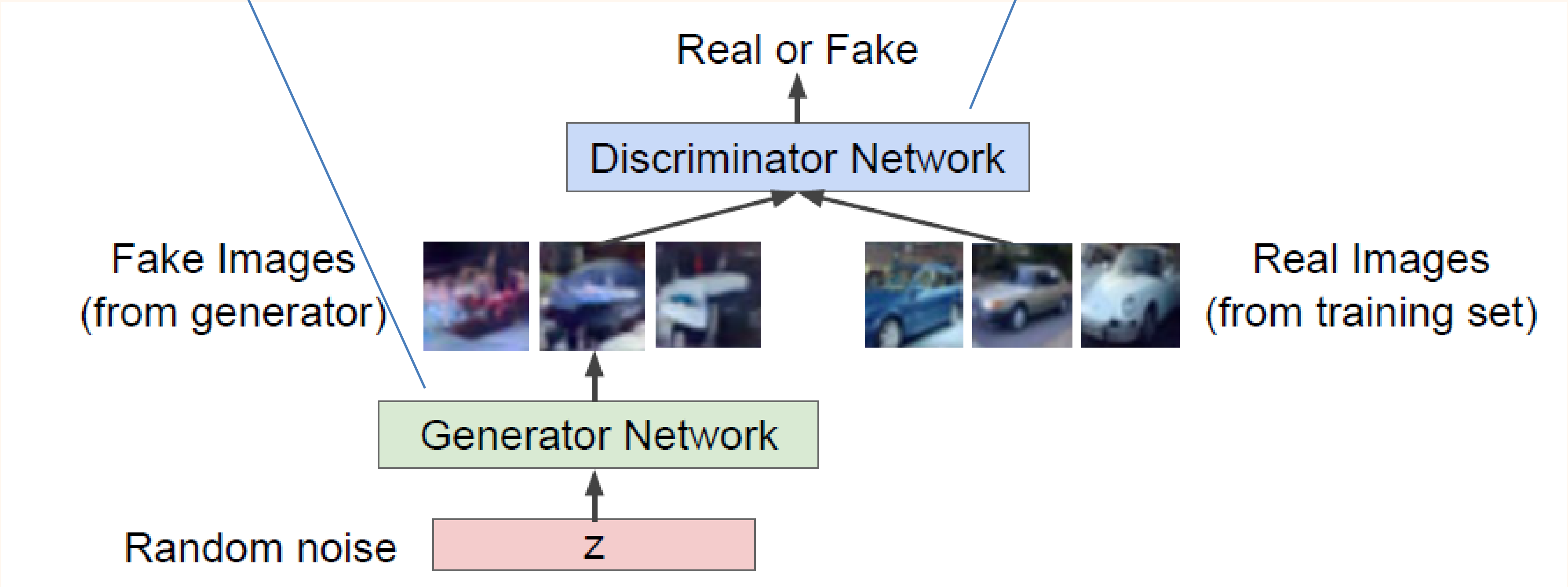


Image Source:
- http://cs231n.stanford.edu/

Minimax optimization:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathrm{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathrm{E}_{z \sim p(z)} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(z)\right)\right)\right]$$

Discriminator output for real data

Discriminator output for fake data

Fake data by generator

$D_{\theta_d}(x) \rightarrow 1$

Maximization by discriminator

$D_{\theta_d}\left(G_{\theta_g}(z)\right) \rightarrow 0$

Minimization by discriminator

$D_{\theta_d}\left(G_{\theta_g}(z)\right) \rightarrow 1$

Maximization by generator

Alternate between:

$$\max_{\theta_d} \left[\mathrm{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathrm{E}_{z \sim p(z)} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(z)\right)\right)\right]$$

$$\max_{\theta_g} \left[\mathrm{E}_{x \sim p(z)} \log D_{\theta_d}\left(G_{\theta_g}(z)\right)\right]$$

GAN article:
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville and Yoshua Bengio, Generative Adversarial Nets, NIPS 2014: 2672-2680.

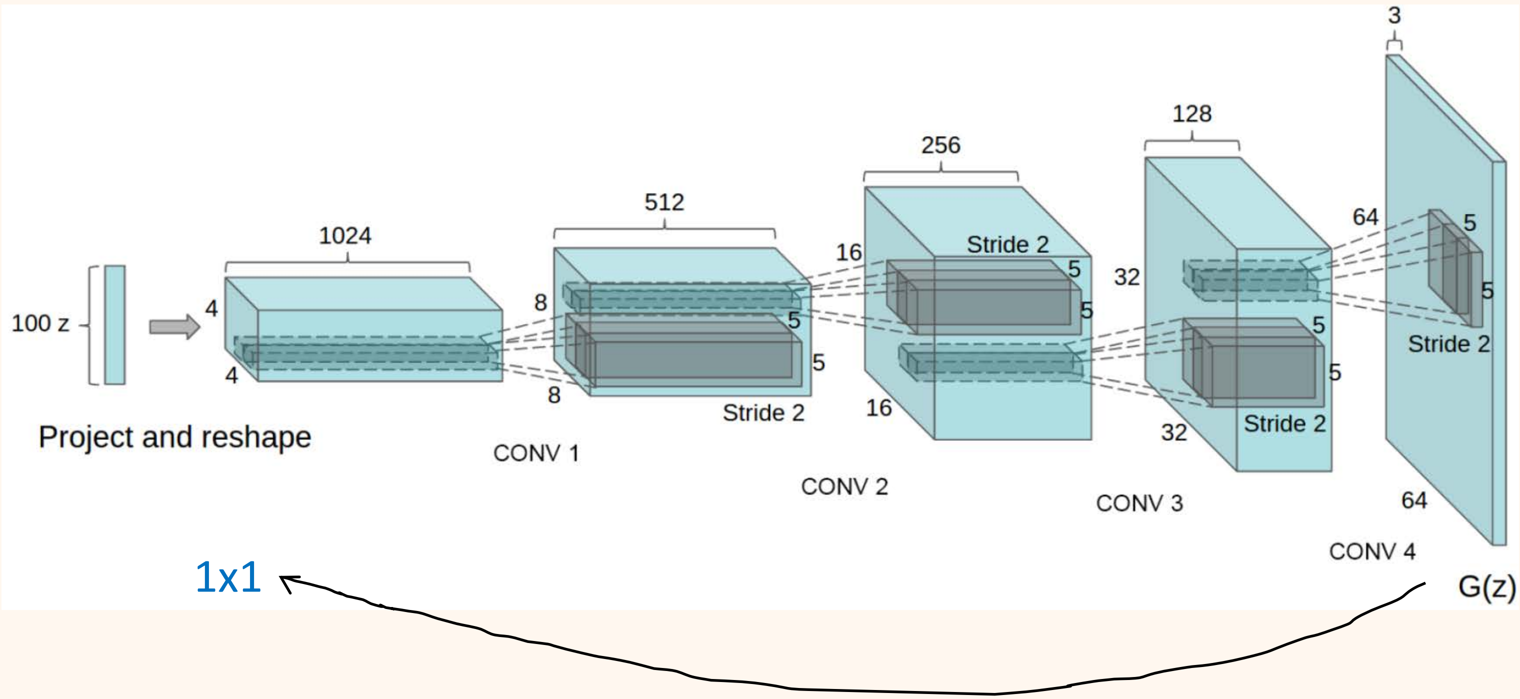
For optimal discriminator:

Generator Objective function: $-\log(4) + 2JSD(p_{data} || p_{fakedata})$

Jensen–Shannon divergence

DC-GAN:

Generator Network:



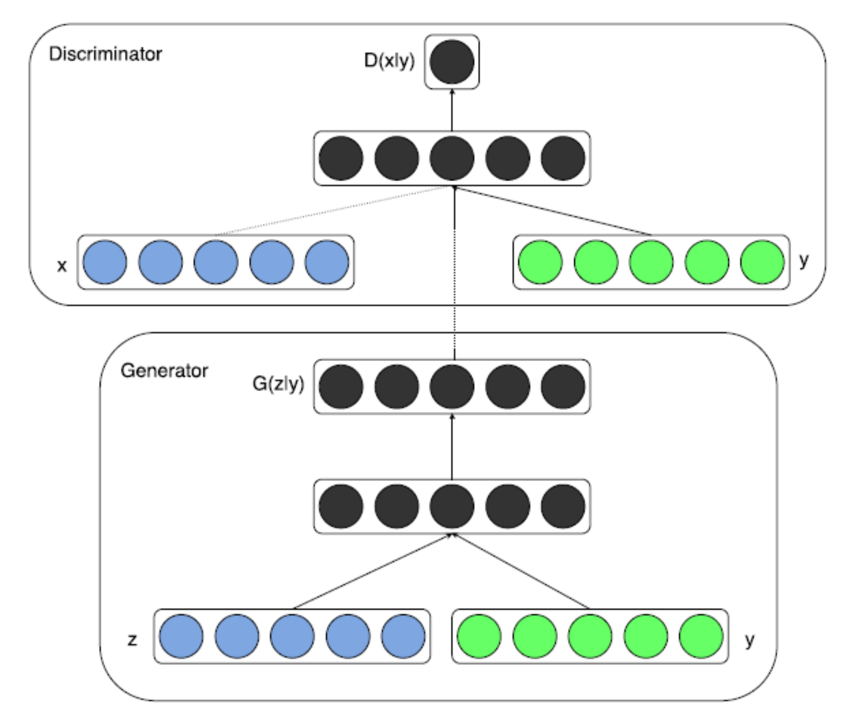
Discriminator Network

Image Generation

DCGAN article & Image source:
Alec Radford, Luke Metz and Soumith Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, ICLR (Poster) 2016.

Conditional GAN:

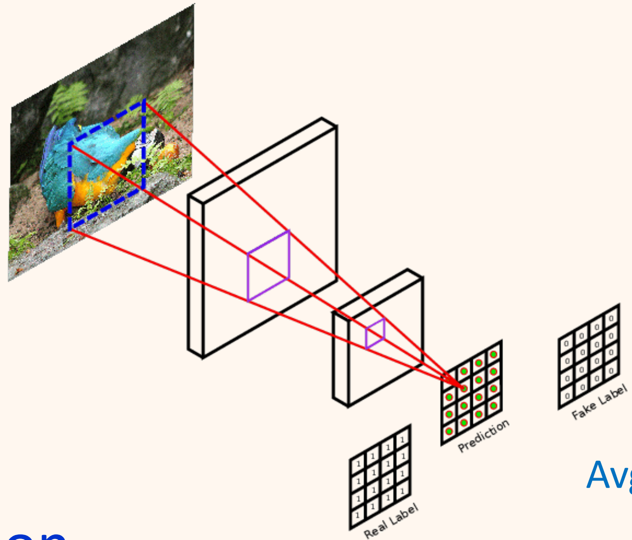
$$\min_{\theta_g} \max_{\theta_d} \left[\mathrm{E}_{(x,y)\sim p_{data}} \log D_{\theta_d}(x|y) + \mathrm{E}_{z\sim p(z),y\sim p_{data}} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(z|y)|y\right)\right)\right]$$



cGAN article & Image source:
Mehdi Mirza and Simon Osindero, Conditional Generative Adversarial Nets. CoRR abs/1411.1784 (2014)

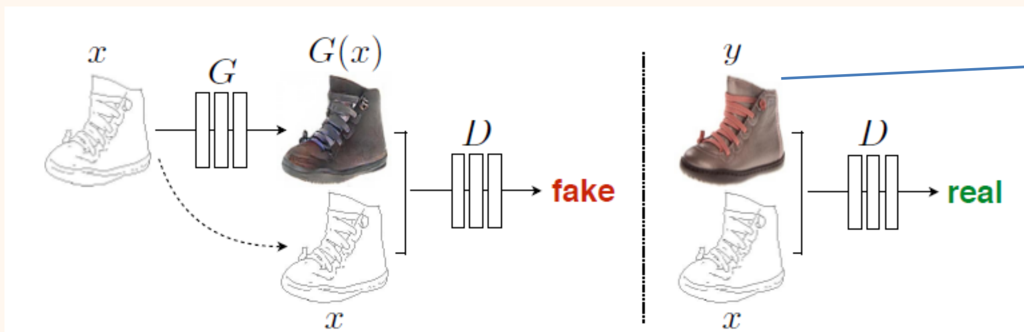
Patch GAN:

Discriminator Network



Avg GAN loss + pixel 2 pixel loss

Image 2 Image Translation



Pair

PatchGAN article & Image Source [Bottom]:
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou and Alexei A. Efros, Image-to-Image Translation with Conditional Adversarial Networks, CVPR 2017: 5967-5976.
Image Source [Top]:
Ugur Demir and Gözde B. Ünal, Patch-Based Image Inpainting with Generative Adversarial Networks, CoRR abs/1803.07422 (2018).

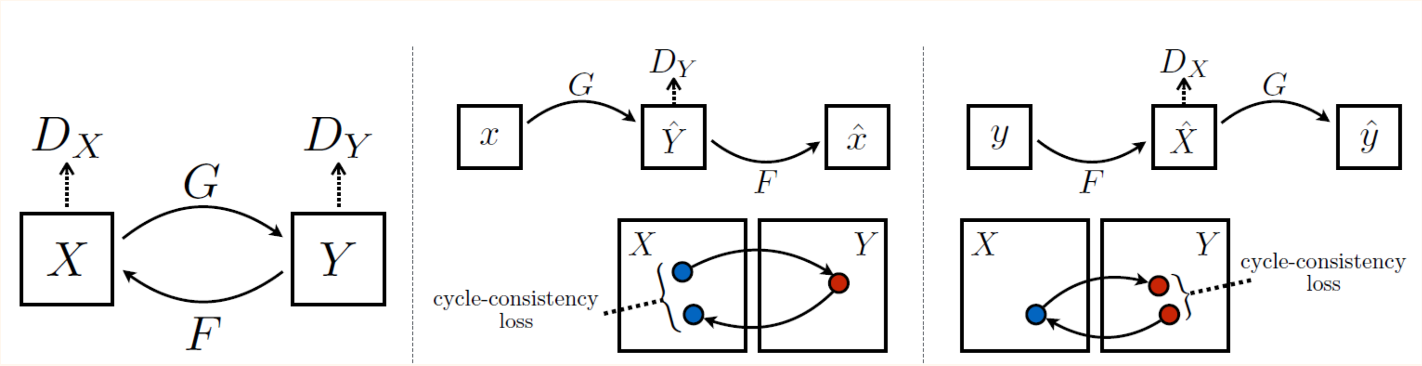
Cycle GAN:

Only unpaired real images are available in both domains

GAN losses:

$$\min_{G} \max_{D_Y} \left[\mathrm{E}_{y \sim p(y)} \log D_Y(y) + \mathrm{E}_{x \sim p(x)} \log\left(1 - D_Y(G(x))\right) \right]$$

$$\min_{F} \max_{D_X} \left[\mathrm{E}_{x \sim p(x)} \log D_X(x) + \mathrm{E}_{y \sim p(y)} \log\left(1 - D_X(F(y))\right) \right]$$



Cyclic consistency loss

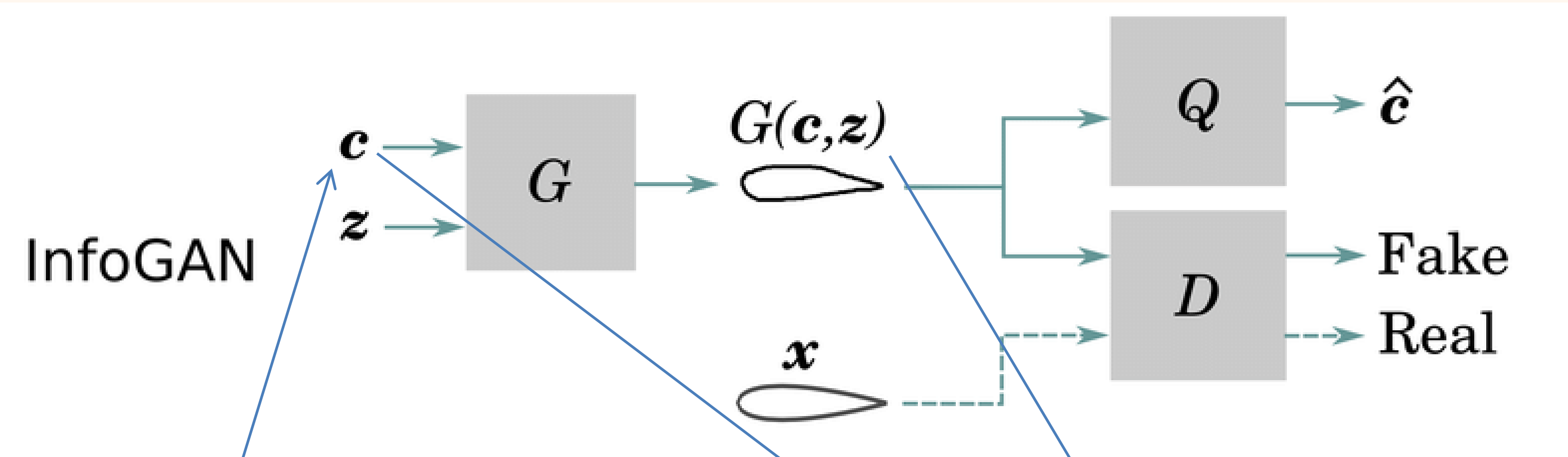
$$\min_{G,F} \left[\mathrm{E}_{x \sim p(x)} \left(\| F(G(x)) - x \| \right) + \mathrm{E}_{y \sim p(y)} \left(\| G(F(y)) - y \| \right) \right]$$

Identity loss: $\min_{G,F} \left[\mathrm{E}_{x \sim p(x)} \left(\| F(x) - x \| \right) + \mathrm{E}_{y \sim p(y)} \left(\| G(y) - y \| \right) \right]$

Weighted sum of the losses!

Cycle GAN article & Image source:
Jun-Yan Zhu, Taesung Park, Phillip Isola and Alexei A. Efros, Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, ICCV 2017: 2242-2251

Info GAN:



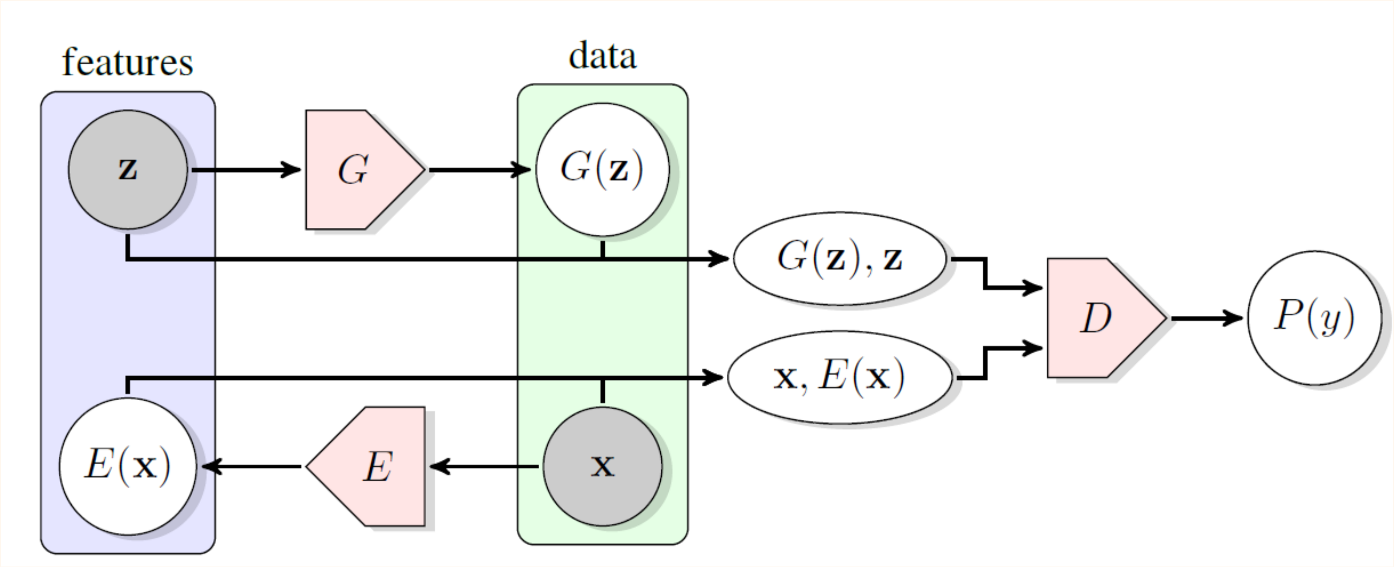
$c \sim p(c/x)$

Mutual Information is Maximized

$$\min_{G,Q} \max_{D} V(D,G,Q) = \Big[\mathrm{E}_{x\sim p_{data}} \log D(x) + \mathrm{E}_{z\sim noise} \log\Big(1 - D\big(G(z)\big)\Big)\Big] - \lambda\big[E_{c\sim p(c), y\sim p(G(z,c))} \log Q(c|y)\big]$$

InfoGAN article:
Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever and Pieter Abbeel, InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, NIPS 2016: 2172-2180.
Image Source:
Wei Chen and Mark Fuge, Synthesizing Designs With Interpart Dependencies Using Hierarchical Generative Adversarial Networks, J. Mech. Des., 2019, 141(11).

BiGAN:



Standard GAN:

$$\min_G \max_D V(D,G) = \left[\mathrm{E}_{x\sim p_{data}} \log D(x) + \mathrm{E}_{z\sim noise} \log\left(1 - D\left(G(z)\right)\right)\right]$$

BiGAN:

$$\min_{G,E} \max_D V(D,E,G) =$$
$$\left[\mathrm{E}_{x\sim p_x} \log D(x, E(x)) + \mathrm{E}_{z\sim P_z} \log\left(1 - D(G(z), z)\right)\right]$$

BiGAN article & Image source:
Jeff Donahue, Philipp Krähenbühl and Trevor Darrell, Adversarial Feature Learning. ICLR (Poster) 2017.

RealnessGAN:

Standard GAN:

$$\min_G \max_D V(D,G) = \left[\mathrm{E}_{x\sim p_{data}} \log D(x) + \mathrm{E}_{z\sim p_z} \log\left(1 - D\left(G(z)\right)\right)\right]$$

During training:

$$\min_G \max_D V(D,G) = \left[\mathrm{E}_{x\sim p_{data}} \log(D(x) - 0) + \mathrm{E}_{x\sim p_g} \log\left(1 - D(x)\right)\right]$$

$$\min_G \max_D V(D,G) = \Big[\mathrm{E}_{x\sim p_{data}} \mathcal{D}_{KL}(\mathcal{A}_1 || D(x)) + \underbrace{\mathrm{E}_{x\sim p_g} \mathcal{D}_{KL}}_{\text{KL-divergence}}(\mathcal{A}_0 || D(x))\Big]$$

Real's distribution

Fake's distribution

KL-divergence

Discrete output distribution: $\dfrac{\exp(\theta_i(x))}{\sum_j \exp\left(\theta_j(x)\right)}$

RealnessGAN article:
Yuanbo Xiangli, Yubin Deng, Bo Dai, Chen Change Loy and Dahua Lin, Real or Not Real, that is the Question, ICLR 2020.

Thank you very much!

Queries?