GAN
（Generative Adversarial Nets）

生成对抗网络

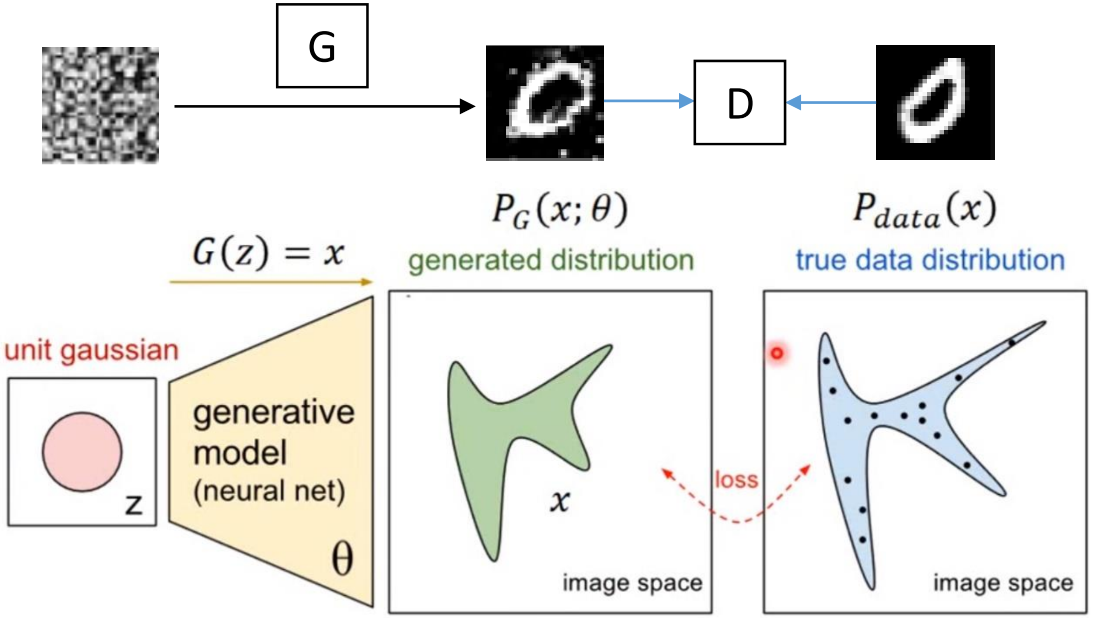
G(X)：生成模型 $\rightarrow$ 输入X来自p_z，输出是p_g； 使 $p_g = p_{data}$

D(X)：判别模型 $\rightarrow$ 输入X来自p_{data}和p_g，输出(0，1)；判断输入X是否来自p_{data}

p_{data}：真实数据分布

p_z：噪声分布

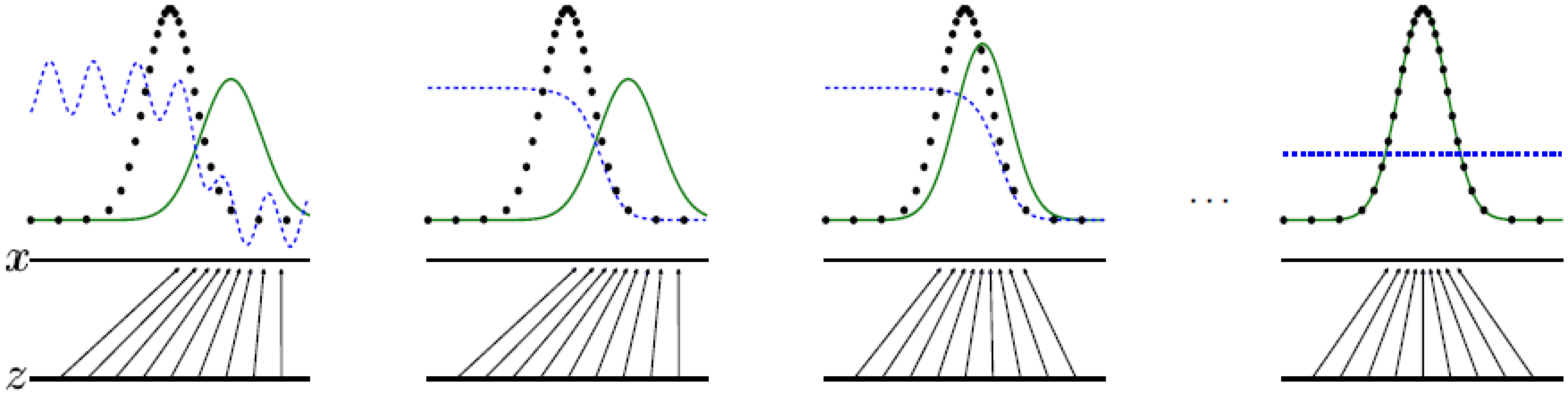
p_g：生成数据分布



目标函数

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

$p_g = p_{data}$，目标函数最优



(a) (b) (c) (d)

黑：p_{data}
蓝：D
绿：p_g

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

- 首先固定G，求解最优的D

$$\begin{aligned} V(G,D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned} \tag{3}$$

For any $(a,b) \in \mathbb{R}^2 \setminus \{0,0\}$, the function $y \to a\log(y) + b\log(1-y)$ achieves its maximum in $[0,1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{\text{data}}) \cup Supp(p_g)$, concluding the proof. □

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

将$D_G^*(x)$代入目标函数

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

$$
\begin{aligned}
&= -2log2 + \int_x P_{data}(x)log\frac{P_{data}(x)}{(P_{data}(x) + P_G(x))/2}dx \\
&\quad + \int_x P_G(x)log\frac{P_G(x)}{(P_{data}(x) + P_G(x))/2}dx \\
&= -2\log 2 + \mathrm{KL}\left(\mathrm{P_{data}}(\mathrm{x})||\frac{\mathrm{P_{data}}(\mathrm{x}) + \mathrm{P_G}(\mathrm{x})}{2}\right) \\
&\quad + \mathrm{KL}\left(\mathrm{P_G}(\mathrm{x})||\frac{\mathrm{P_{data}}(\mathrm{x}) + \mathrm{P_G}(\mathrm{x})}{2}\right) \\
&= -2log2 + 2JSD(P_{data}(x)||P_G(x))
\end{aligned}
$$

Jensen-Shannon divergence

KL divergence:

$$D(P||Q) = \sum(P(x)log(P(x)/Q(x)))$$

JS divergence:表示了两个分布之间的差异

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}D(P \parallel M) + \frac{1}{2}D(Q \parallel M)$$

$$M = \frac{1}{2}(P + Q)$$

$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

因为JSD非负，当且仅当$p_g = p_{data}$时为零。

因此，仅当$p_g = p_{data}$时，目标函数最优，$D_G^*(\mathrm{x}) = 0.5$，$C^* = -\log(4)$。

训练

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  • Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.

  • Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.

  • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right].$$

  end for

  • Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.

  • Update the generator by descending its stochastic gradient:

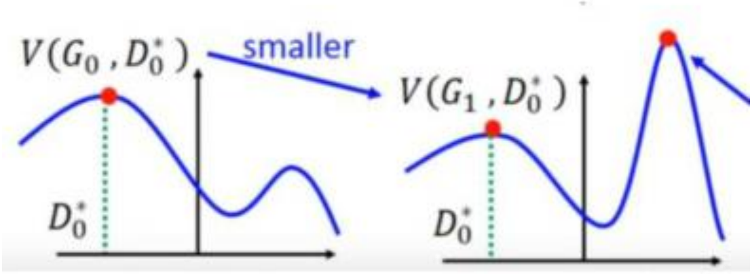
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$

end for

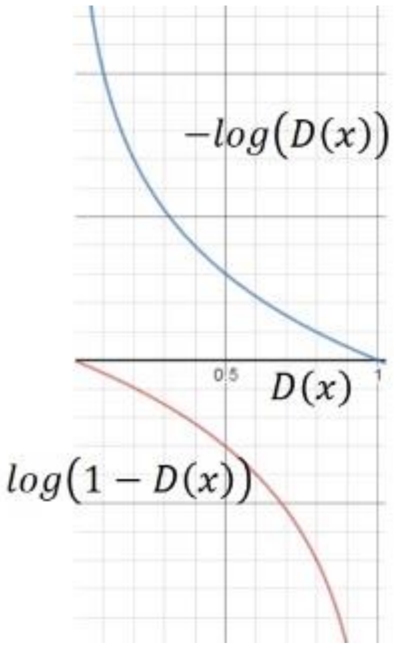
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

works. The disadvantages are primarily that there is no explicit representation of $p_g(x)$, and that D must be synchronized well with G during training (in particular, G must not be trained too much without updating D, in order to avoid "the Helvetica scenario" in which G collapses too many values of z to the same value of x to have enough diversity to model p_{data}), much as the negative chains of a Boltzmann machine must be kept up to date between learning steps. The advantages are that Markov

但是这里有个问题就是，你可能在 D_0^* 的位置取到了 $\max_D V(G_0, D_0) = V(G_0, D_0^*)$，然后更新 G_0 为 G_1，可能 $V(G_1, D_0^*) < V(G_0, D_0^*)$ 了，但是并不保证会出现一个新的点 D_1^* 使得 $V(G_1, D_1^*) > V(G_0, D_0^*)$，这样更新G就没达到它原来应该要的效果，如下图所示：

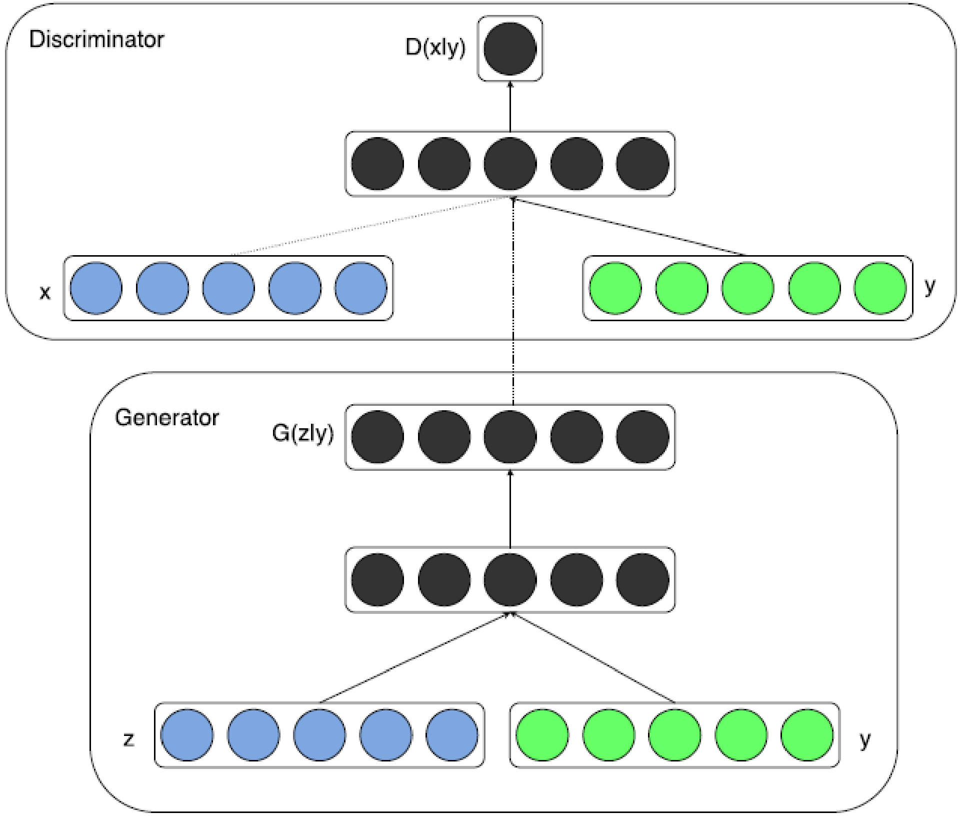


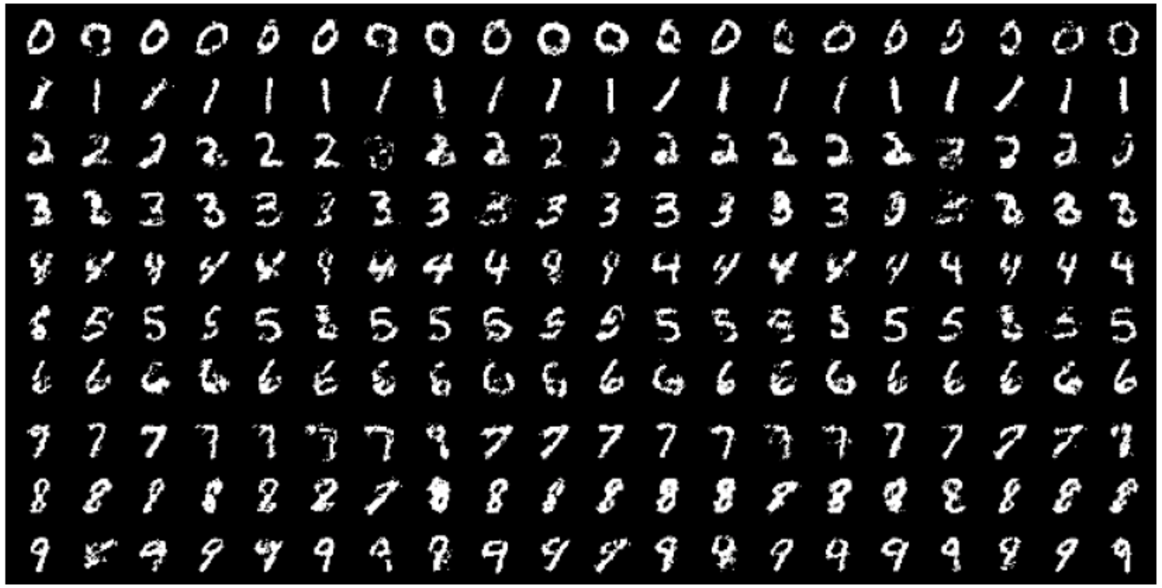
避免上述情况的方法就是更新G的时候，不要更新G太多。



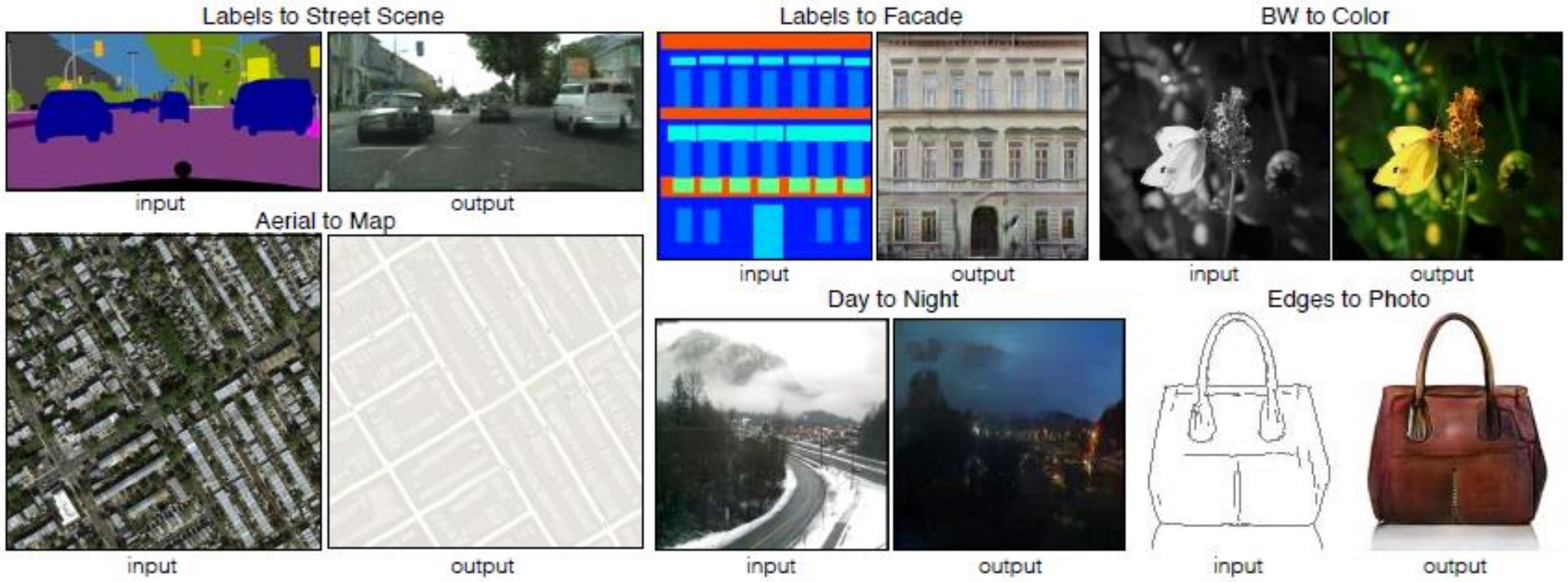
CGAN
(Conditional Generative Adversarial Nets)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))].$$





pix2pix



目标函数：

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) =& \mathbb{E}_{x,y}[\log D(x, y)]+ \\ & \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))],\end{aligned}$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1].$$

U-Net

G

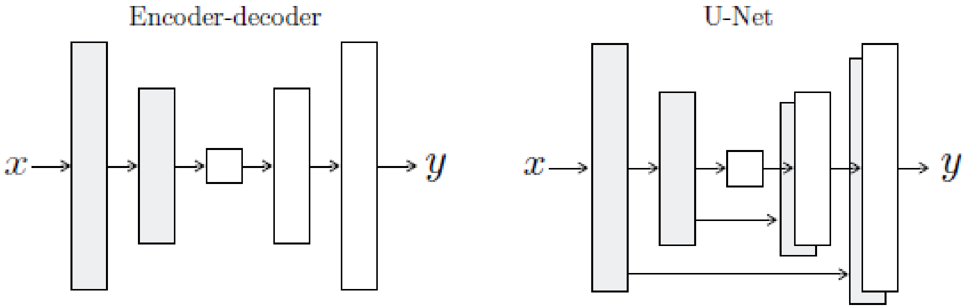
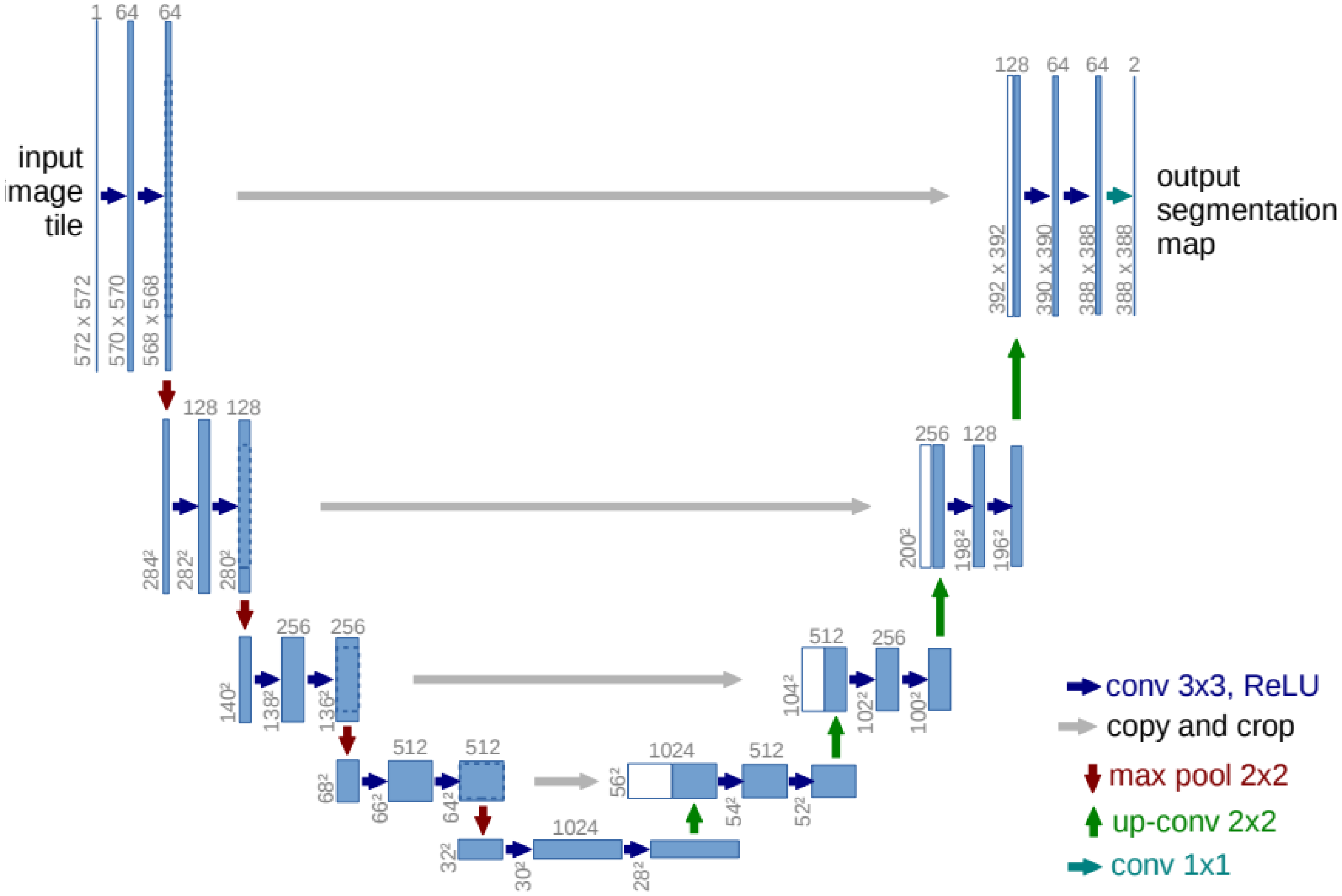


Figure 3: Two choices for the architecture of the generator. The "U-Net" [49] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.



D Markovian discriminator (PatchGAN)

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

PatchGAN的思想是，GAN只负责处理高频成分，那么判别器就没必要以一整张图作为输入，只需要对NxN的一个图像patch去进行判别就可以了。

这也是为什么叫Markovian discriminator，因为在patch以外的部分认为和本patch互相独立。


Input
Ground truth
L1
cGAN
L1 + cGAN