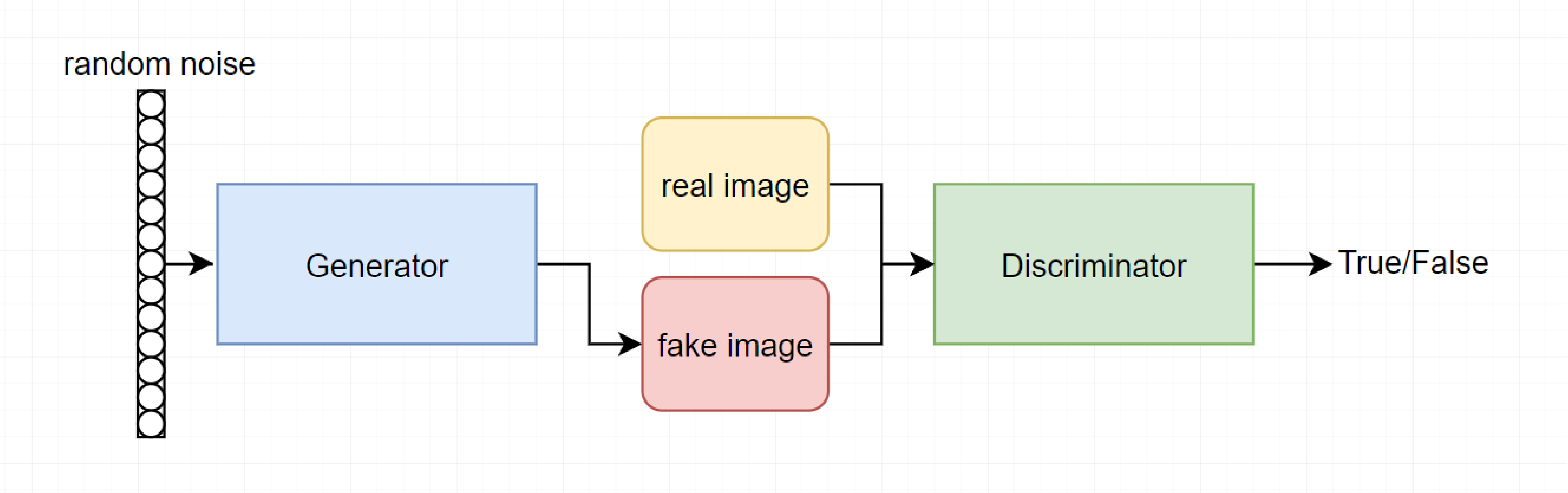
Info-GAN LAB

助教 : 曾敏原

GAN


random noise
Generator
real image
fake image
Discriminator
True/False

DCGAN

- PyTorch DCGAN Tutorial
  - https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

- We split training of GAN into 2 parts
  - Training of discriminator
  - Training of generator

Part 1. – Train the Discriminator

- Maximize $L_D = \log\big(D(x)\big) + \log(1 - D\big(G(z)\big))$
  - passing a batch of real data through D and calculate $\log\big(D(x)\big)$

```
## Train with all-real batch
netD.zero_grad()
# Format batch
real_cpu = data[0].to(device)
b_size = real_cpu.size(0)
label = torch.full((b_size,), real_label, device=device)
# Forward pass real batch through D
output = netD(real_cpu).view(-1)
# Calculate loss on all-real batch
errD_real = criterion(output, label)
# Calculate gradients for D in backward pass
errD_real.backward()
```

Part 1. – Train the Discriminator

- Maximize $L_D = \log(D(x)) + \log(1 - D(G(z)))$
  - construct a batch of fake data using current generator then passing it through D and calculate $\log\left(1 - D(G(z))\right)$

```
## Train with all-fake batch
# Generate batch of latent vectors
noise = torch.randn(b_size, nz, 1, 1, device=device)
# Generate fake image batch with G
fake = netG(noise)
label.fill_(fake_label)
# Classify all fake batch with D
output = netD(fake.detach()).view(-1)
# Calculate D's loss on the all-fake batch
errD_fake = criterion(output, label)
# Calculate the gradients for this batch
errD_fake.backward()
D_G_z1 = output.mean().item()
# Add the gradients from the all-real and all-fake batches
errD = errD_real + errD_fake
```

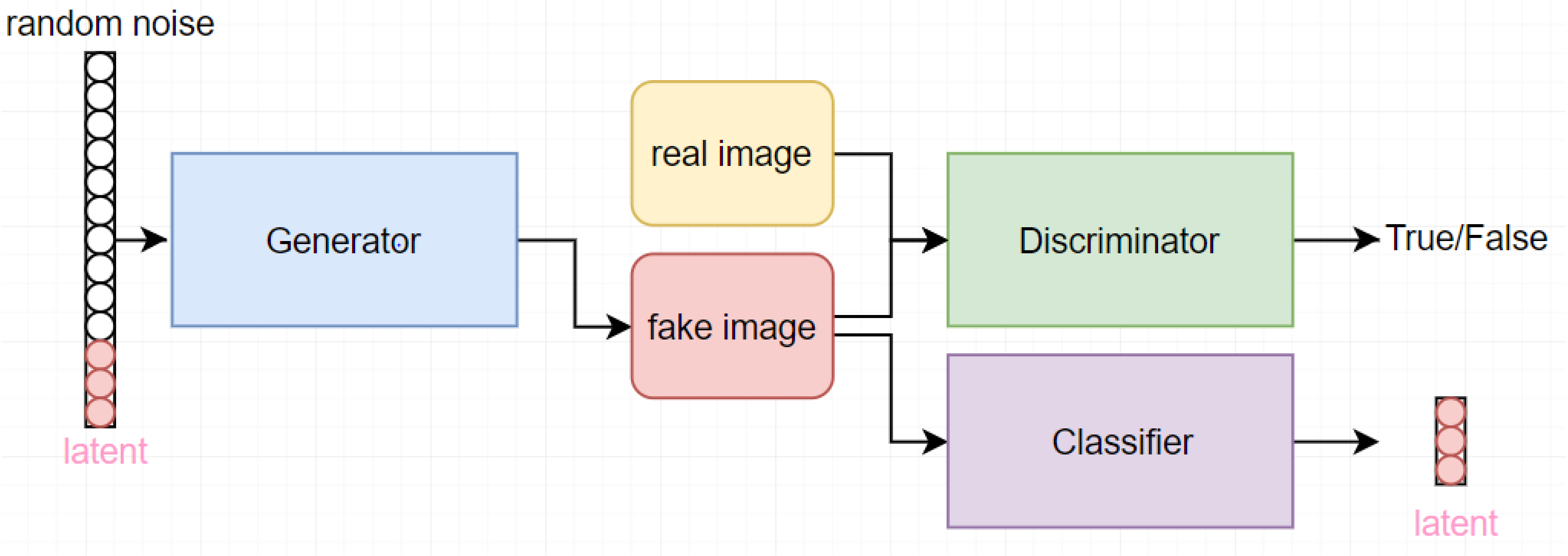
Part 2 - Train the Generator

- Maximize $L_D = \log(D(G(z)))$
    - Classify the output of G using real label, compute gradients of G

```
netG.zero_grad()
label.fill_(real_label)  # fake labels are real for generator cost
# Since we just updated D, perform another forward pass of all-fake batch through D
output = netD(fake).view(-1)
# Calculate G's loss based on this output
errG = criterion(output, label)
# Calculate gradients for G
errG.backward()
```

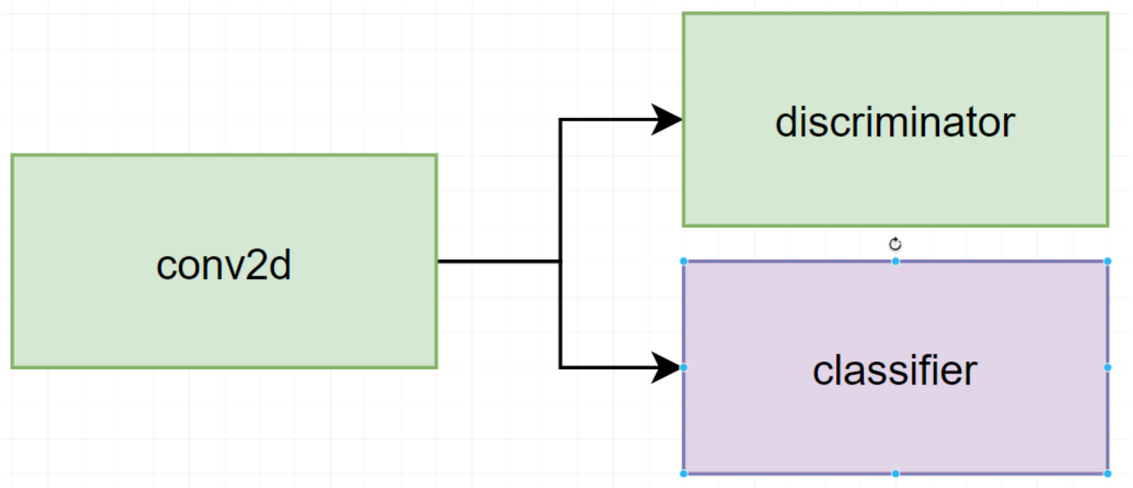
Info-GAN

Paper :
InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
https://arxiv.org/abs/1606.03657



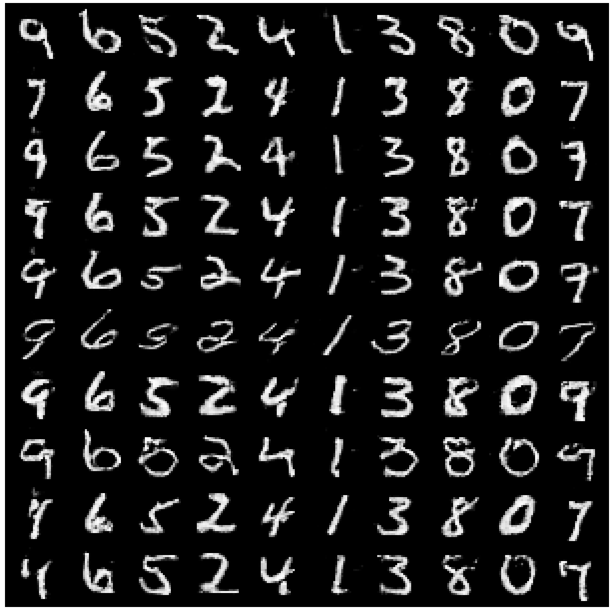
Info-GAN

- With the above example, we only need to add on classification loss then we can have an Info-GAN.
- We shall train classifier and generator at the same time
- Discriminator and Classifier can share the convolution layers

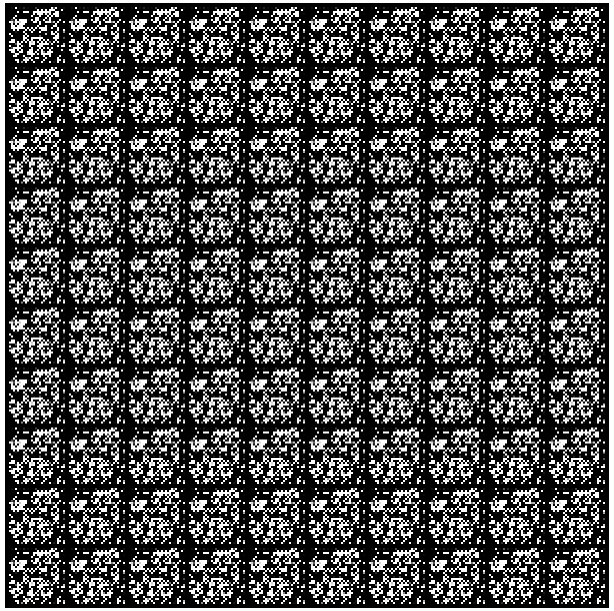


Info-GAN

Success



Fail



Info GAN Prof.

$$I\big(c; G(z,c)\big) = H(c) - H(c|G(z,c))$$

$$\mathrm{H}(Y|X) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)}$$

$$= E_{x\sim G(z,c)}\left[E_{c\sim P(c|X=x)} logP(c|X=x)\right] + \mathrm{H(c)}$$

Adding $E_{x\sim G(z,c)}\left[E_{c\sim P(c|X=x)} logQ(c|X=x)\right]$

$$= E_{x\sim G(z,c)}\left[E_{c\sim P(c|X=x)} logQ(c|X=x) + E_{c\sim P(c|X=x)} log\frac{P(c|X=x)}{Q(c|X=x)}\right] + H(c)$$

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\ln\frac{P(i)}{Q(i)}$$

$$= E_{x\sim G(z,c)}\left[E_{c\sim P(c|X=x)} logQ(c|X=x) + D_{KL}(P(c|X=x)||Q(c|X=x))\right] + H(c)$$

$$\geq E_{x\sim G(z,c)}\left[E_{c\sim P(c|X=x)} logQ(c|X=x)\right] + H(c)$$