COMP9444 Neural Networks and Deep Learning

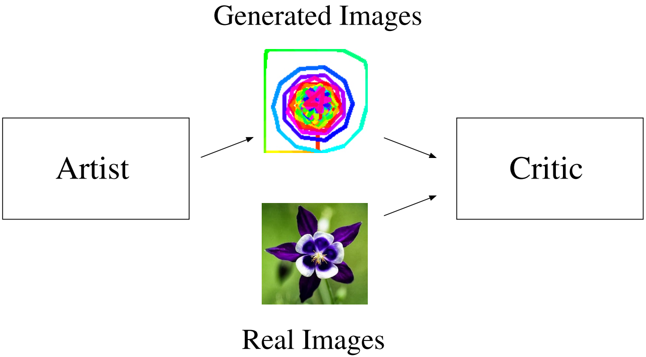
14: Adversarial Training and GANs

Alan Blair, UNSW, 2017-18

Outline

- Artist-Critic Co-Evolution
- Co-Evolution Paradigms
- Blind Watchmaker (GP Artist, Human Critic)
- Evolutonary Art (GP Artist, GP or NN Critic)
- Generative Adversarial Networks (CNN Artist, CNN Critic)

Artist-Critic Co-Evolution

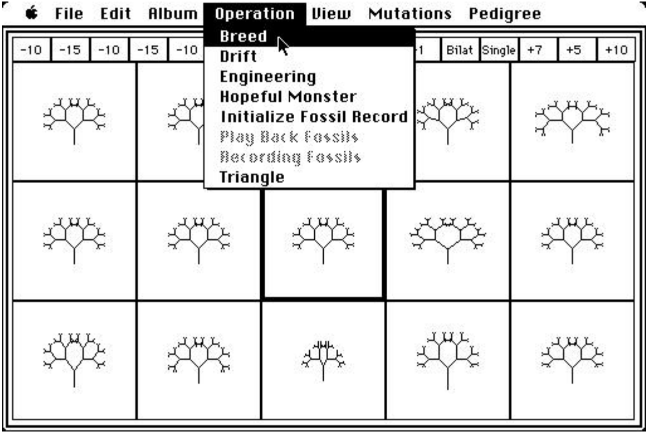


- Critic is rewarded for distinguishing real images from those generated by the artist.
- Artist is rewarded for fooling the critic into thinking that generated images are real.

Co-Evolution Paradigms

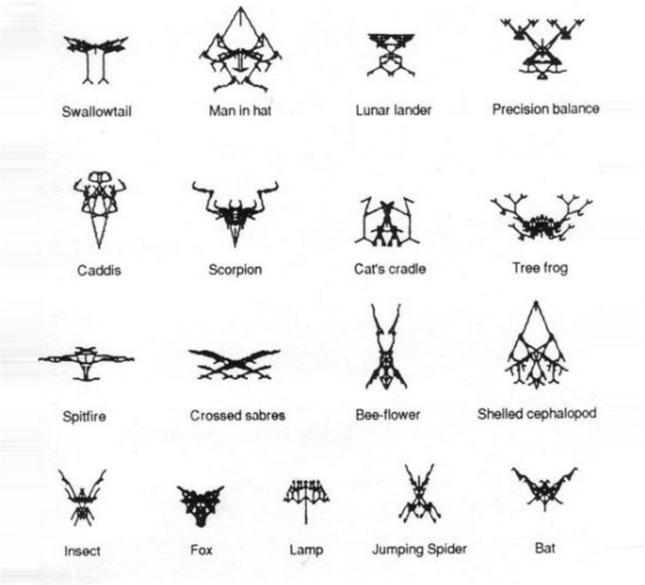
Artist	Critic	Method	Reference
Biomorph	Human	Blind Watchmaker	(Dawkins, 1986)
GP	Human	Blind Watchmaker	(Sims, 1991)
CPPN	Human	PicBreeder	(Secretan, 2011)
CA	Human	EvoEco	(Kowaliw, 2012)
GP	SOM	Artificial Creativity	(Saunders, 2001)
Photo	NN	Computational Aesthetics	(Datta, 2006)
GP	NN	Computational Aesthetics	(Machado, 2008)
Agents	NN	Evolutionary Art	(Greenfield, 2009)
GP	NN	Aesthetic Learning	(Li & Hu, 2010)
HERCL	HERCL	Co-Evolving Line Drawings	(Vickers, 2017)
HERCL	DCNN	HERCL Function/CNN	(Soderlund, 2018)
DCNN	DCNN	Generative Adversarial Nets	(Goodfellow, 2014)
DCNN	DCNN	Plug & Play Generative Nets	(Nguyen, 2016)

Blind Watchmaker (Dawkins, 1986)



- the Human is presented with 15 images
- the chosen individual is used to breed the next generation

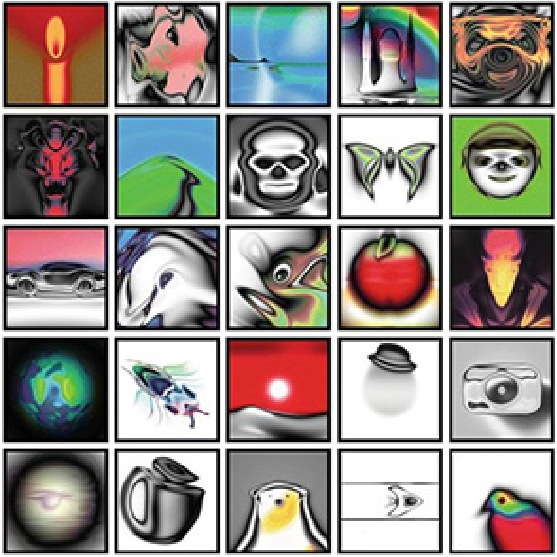
Blind Watchmaker Biomorphs



Blind Watchmaker (Sims, 1991)

- Artist = Genetic Program (GP)
  - used as function to compute R,G,B values for each pixel x, y
- Critic = Human

PicBreeder Examples



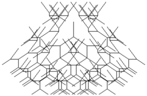
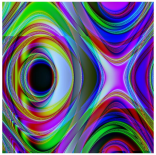
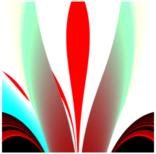
PicBreeder (Secretan, 2011)

- Artist = Convolutional Pattern Producing Neural Network (CPPN)
- Critic = Human
- interactive Web site (`picbreeder.org`) where you can choose existing individual and use it for further breeding

---

- Blind Watchmaker paradigm is cool, but it may require a lot of work from the Human
- Can the Human be replaced by an automated Critic?

Image Generating Paradigms

 Biomorph	 GP	 Picbreeder	 CA
 HERCL(draw)	 HERCL(func)	 HERCL(func)	 GAN

Generative Adversarial Networks

Generator (Artist) G_θ and Discriminator (Critic) D_ψ are both Deep Convolutional Neural Networks.

Generator $G_\theta : z \mapsto x$, with parameters θ, generates an image x from latent variables z (sampled from a Normal distribution).

Discriminator $D_\psi : x \mapsto D_\psi(x) \in (0, 1)$, with parameters ψ, takes an image x and estimates the probability of the image being real.

Generator and Discriminator play a 2-player zero-sum game to compute:

$$\min_\theta \max_\psi \Big(\mathbf{E}_{x \sim p_{\mathrm{data}}} \big[\log D_\psi(x) \big] + \mathbf{E}_{z \sim p_{\mathrm{model}}} \big[\log \big(1 - D_\psi(G_\theta(z)) \big) \big] \Big)$$

Discriminator tries to maximize the bracketed expression, Generator tries to minimize it.

Generative Adversarial Networks

Alternate between:

Gradient ascent on Discriminator:

$$\max_{\psi}\Big(\mathbf{E}_{x\sim p_{\text{data}}}\big[\log D_\psi(x)\big]+\mathbf{E}_{z\sim p_{\text{model}}}\big[\log\big(1-D_\psi(G_\theta(z))\big)\big]\Big)$$

Gradient descent on Generator, using:

$$\min_{\theta}\ \mathbf{E}_{z\sim p_{\text{model}}}\big[\log\big(1-D_\psi(G_\theta(z))\big)\big]$$

Generative Adversarial Networks

Alternate between:

Gradient ascent on Discriminator:

$$\max_{\psi}\Big(\mathbf{E}_{x\sim p_{\rm data}}\big[\log D_\psi(x)\big]+\mathbf{E}_{z\sim p_{\rm model}}\big[\log\big(1-D_\psi(G_\theta(z))\big)\big]\Big)$$

Gradient descent on Generator, using:

$$\cancel{\min_{\theta}\mathbf{E}_{z\sim p_{\rm model}}\big[\log\big(1-D_\psi(G_\theta(z))\big)\big]}$$

This formula puts too much emphasis on images that are correctly classified. Better to do gradient ascent on Generator, using:

$$\max_{\theta}\mathbf{E}_{z\sim p_{\rm model}}\big[\log\big(D_\psi(G_\theta(z))\big)\big]$$

This puts more emphasis on the images that are wrongly classified.

GAN properties:

- one network aims to produces the full range of images x, with different values for the latent variables z
- differentials are backpropagated through the Discriminator network and into the Generator network
- compared to previous approaches, the images produced are much more realistic!

repeat:

for k steps do

sample minibatch of m latent samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from $p(z)$

sample minibatch of m training items $\{x^{(1)}, \ldots, x^{(m)}\}$

update Discriminator by gradient ascent on ψ:

$$\nabla_\psi \frac{1}{m} \sum_{i=1}^{m} \Big[\log D_\psi(x^{(i)}) + \log\big(1 - D_\psi(G_\theta(z^{(i)}))\big)\Big]$$

end for

sample minibatch of m latent samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from $p(z)$

update Generator by gradient ascent on θ:

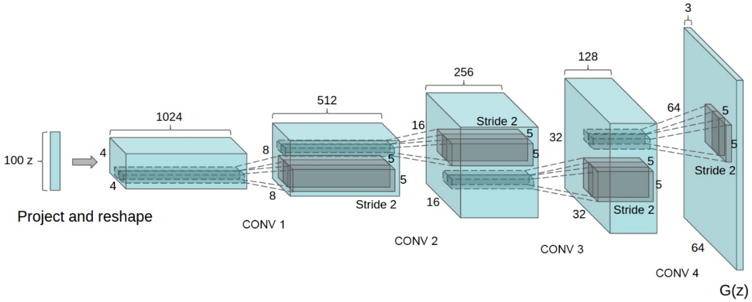
$$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} \log\big(D_\psi(G_\theta(z^{(i)}))\big)$$

end repeat

GAN Convolutional Architectures

- normalize images to between -1 and $+1$
- replace pooling layers with:
  - strided convolutions (Discriminator)
  - fractional-strided convolutions (Generator)
- use BatchNorm in both Generator and Discriminator
- remove fully connected hidden layers for deeper architectures
- use tanh at output layer of Generator,
  ReLU activation in all other layers
- use LeakyReLU activation for all layers of Discriminator

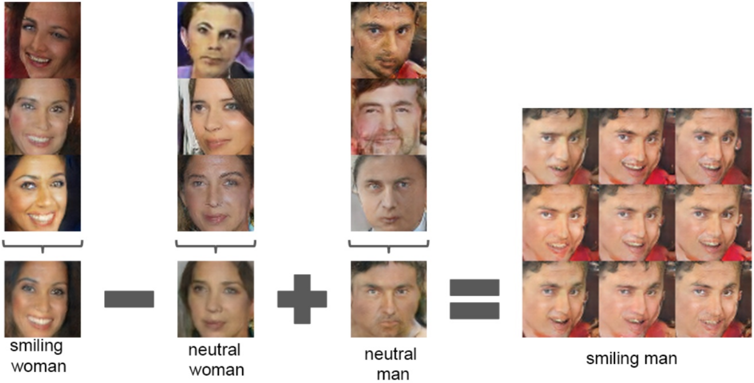
Generator Architecture



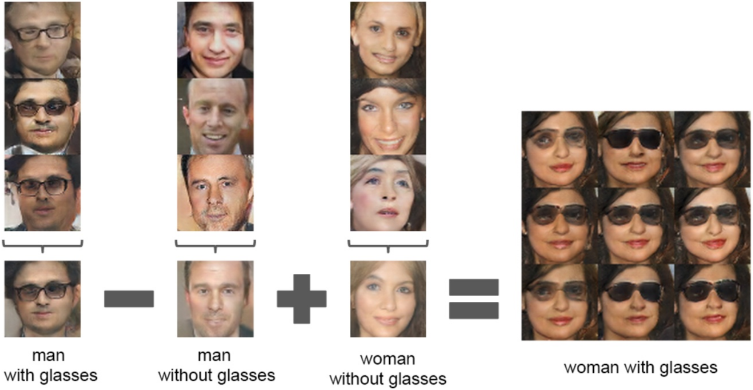
Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (Radford et al., 2016)

GAN Generated Images

GAN Image Vector Arithmetic



GAN Image Vector Arithmetic



Mediocre Stable States

- Like any coevolution, GANs can sometimes oscillate or get stuck in a mediocre stable state.
- **oscillation**: GAN trains for a long time, generating a variety of images, but quality fails to improve (compare IPD)
- **mode collapse**: Generator produces only a small subset of the desired range of images, or converges to a single image (with minor variations)

Methods for avoiding mode collapse:

- Conditioning Augmentation
- Minibatch Features (Fitness Sharing)
- Unrolled GANs

The GAN Zoo

- Contex-Encoder for Image Inpainting
- Texture Synthesis with Patch-based GAN
- Conditional GAN
- Text-to-Image Synthesis
- StackGAN
- Patch-based Discriminator
- S^2-GAN
- Style-GAN
- Plug-and-Play Generative Networks

GAN References

http://dl.ee.cuhk.edu.hk/slides/gan.pdf

cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

http://www.iangoodfellow.com/slides/2016-12-04-NIPS.pdf

https://arxiv.org/abs/1612.00005