Generative adversarial networks

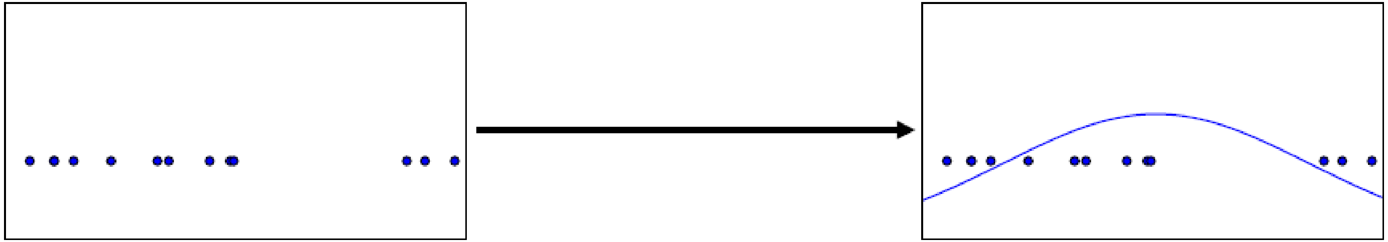
Hakan Bilen

Machine Learning Practical - MLP Lecture 13
23 January 2019
http://www.inf.ed.ac.uk/teaching/courses/mlp/

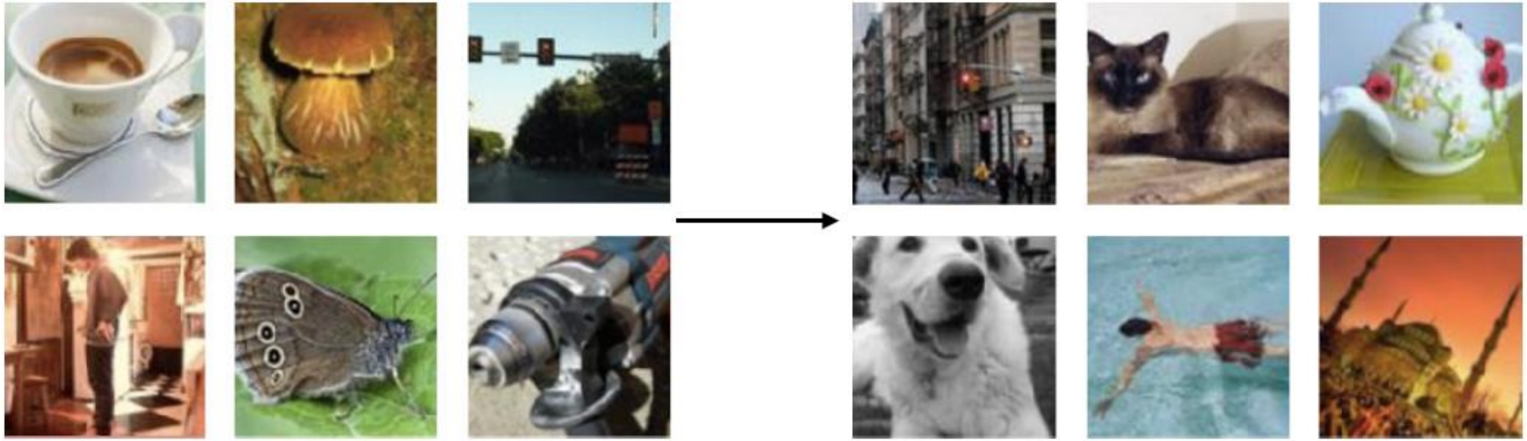
Slide credits: Ian Goodfellow

Generative modeling

- Density estimation



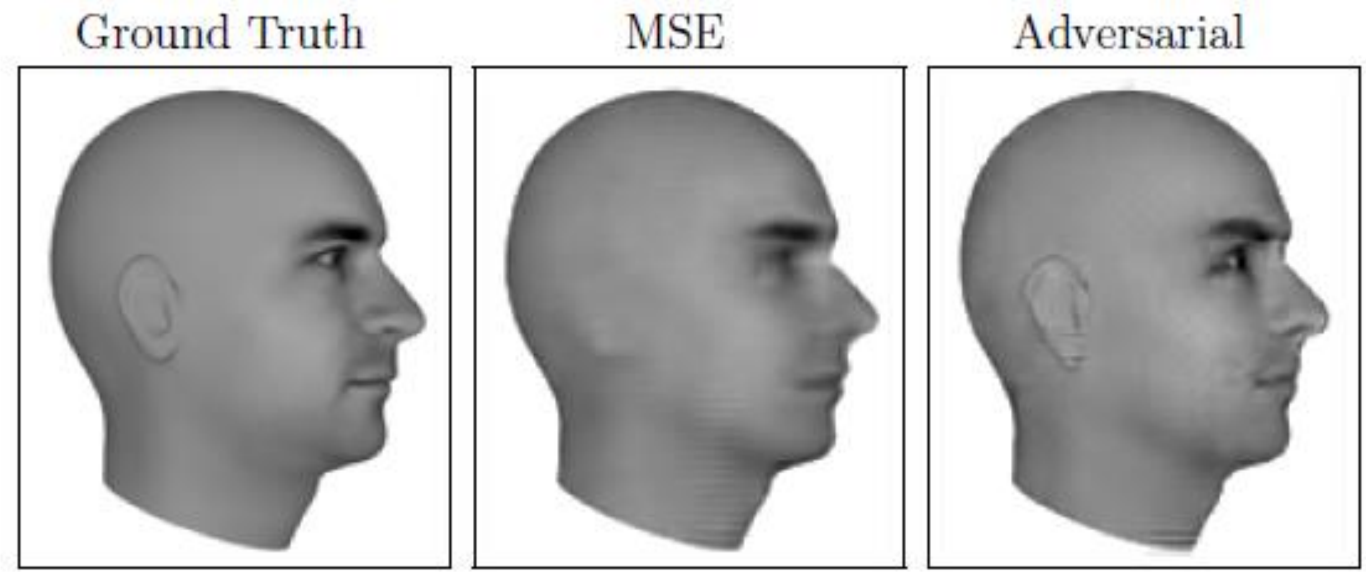
- Sample generation



Why are generative models useful?

- Test of our intellectual ability to use high dimensional probability distributions
- Learn from simulated data and transfer it to real world
- Complete missing data (including multi-modal)
- Realistic generation tasks (image and speech)

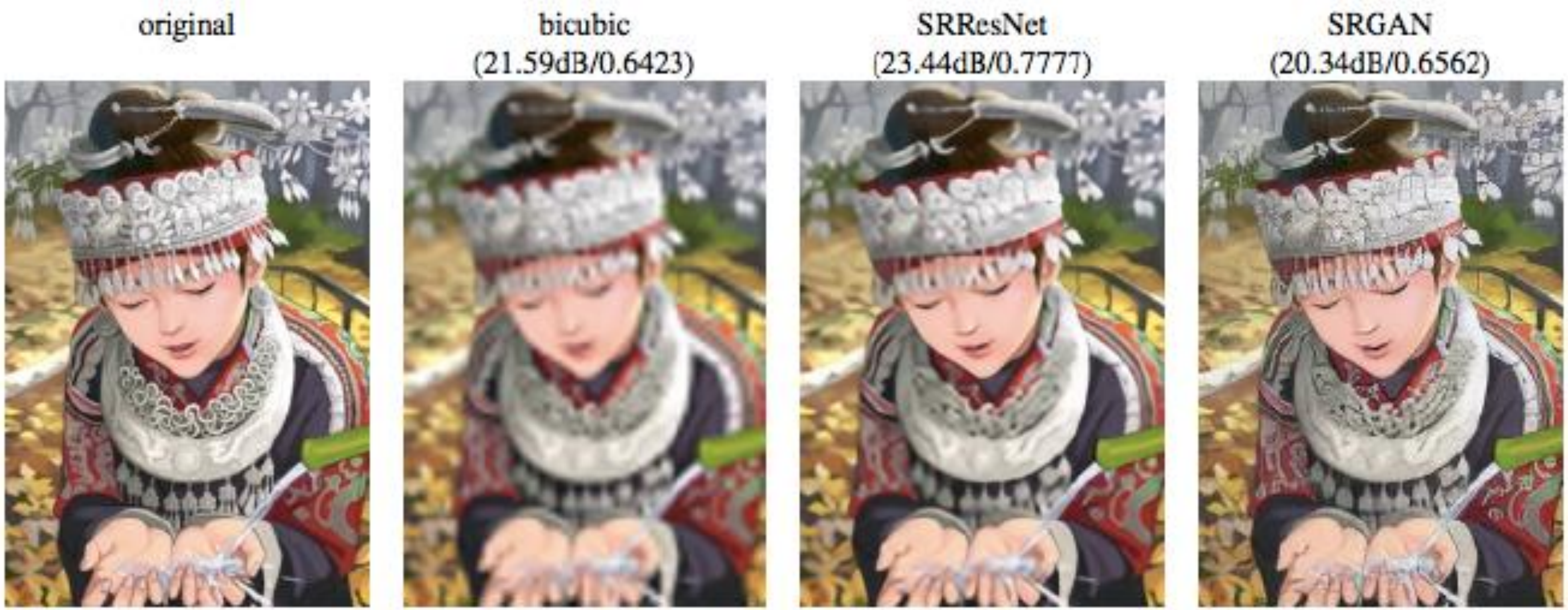
Next video frame prediction



Lotter et al 2016

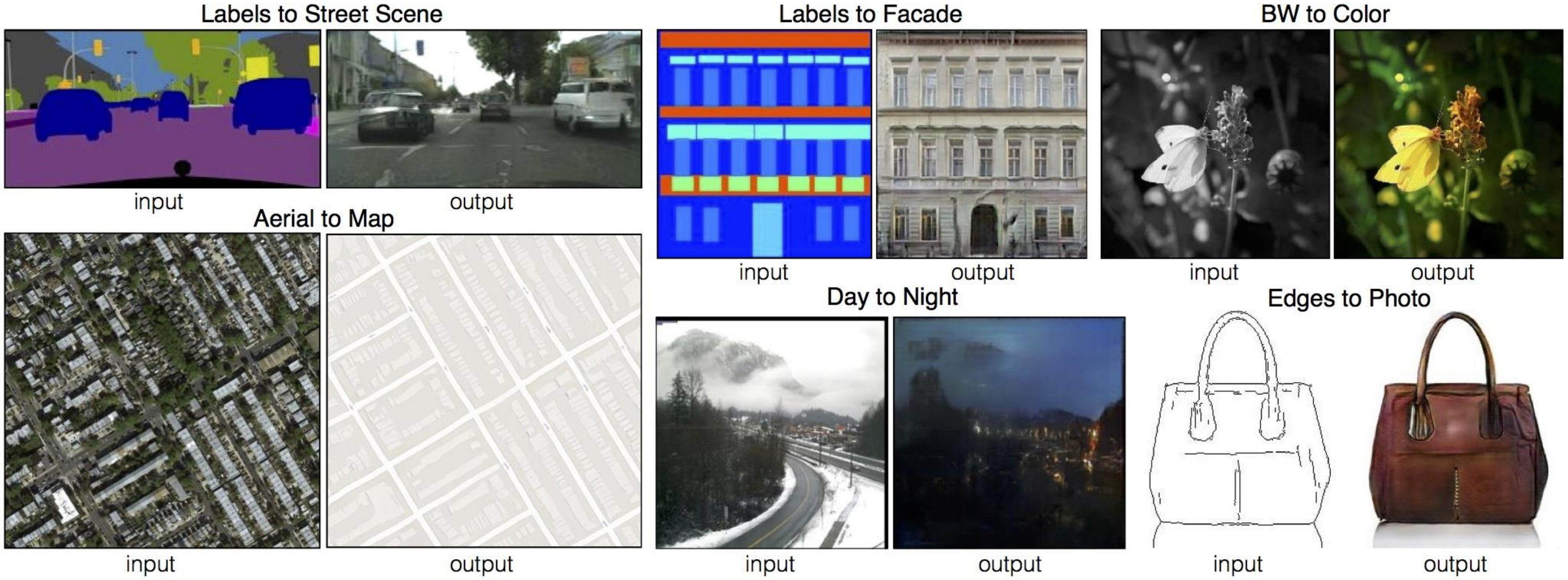
Lotter et al. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning, ICLR'17

Photo-realistic super-resolution



[Ledig et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, CVPR'17]

Image-to-image translation

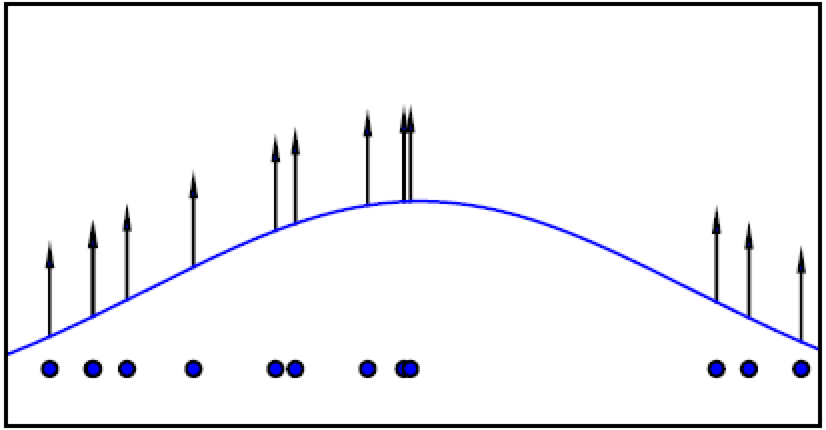


[Isola et al. Image-to-Image Translation with Conditional Adversarial Nets, CVPR'17]

Maximum likelihood

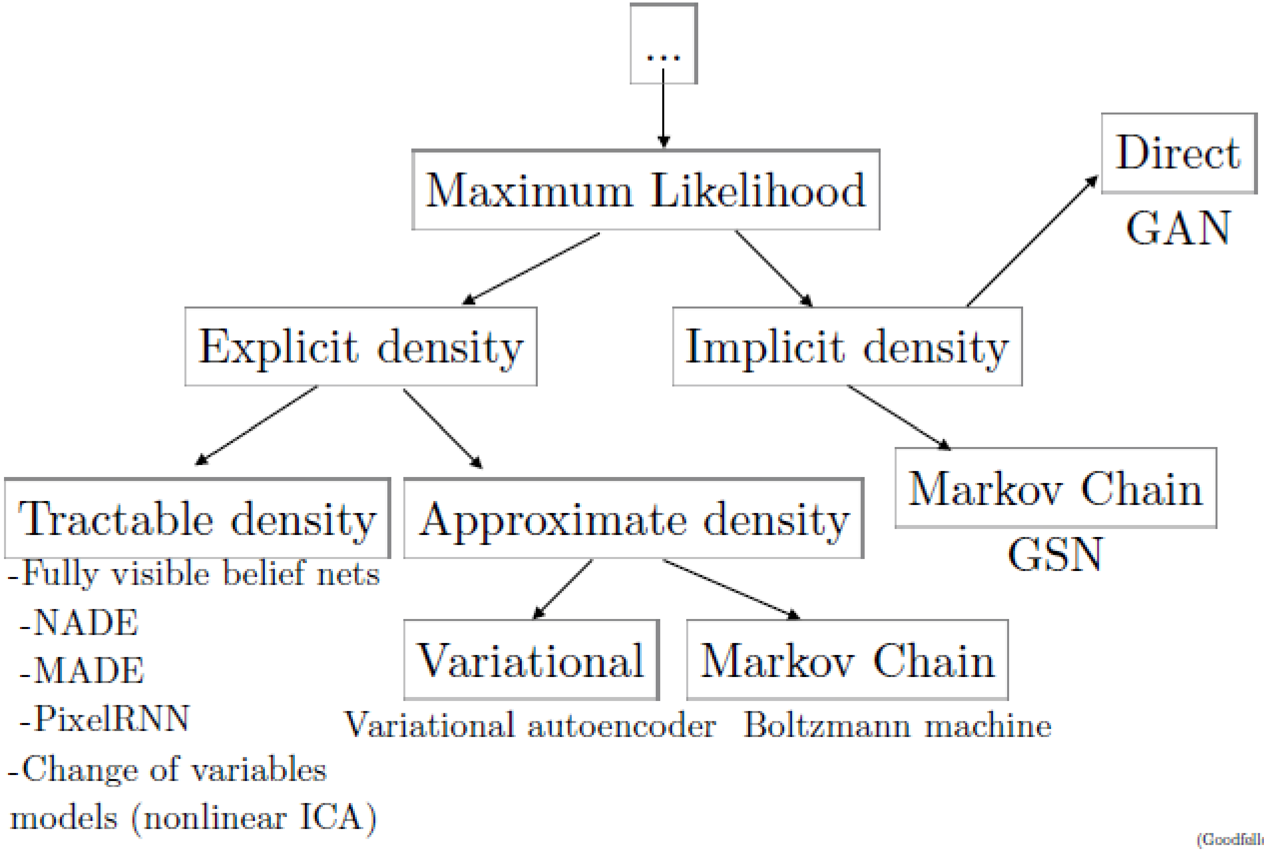
Assume that all generative models maximise the likelihood of the training data

- x: input
- p_{data}: probability density function of input samples
- p_{model}: estimate probability function, parameterised by θ



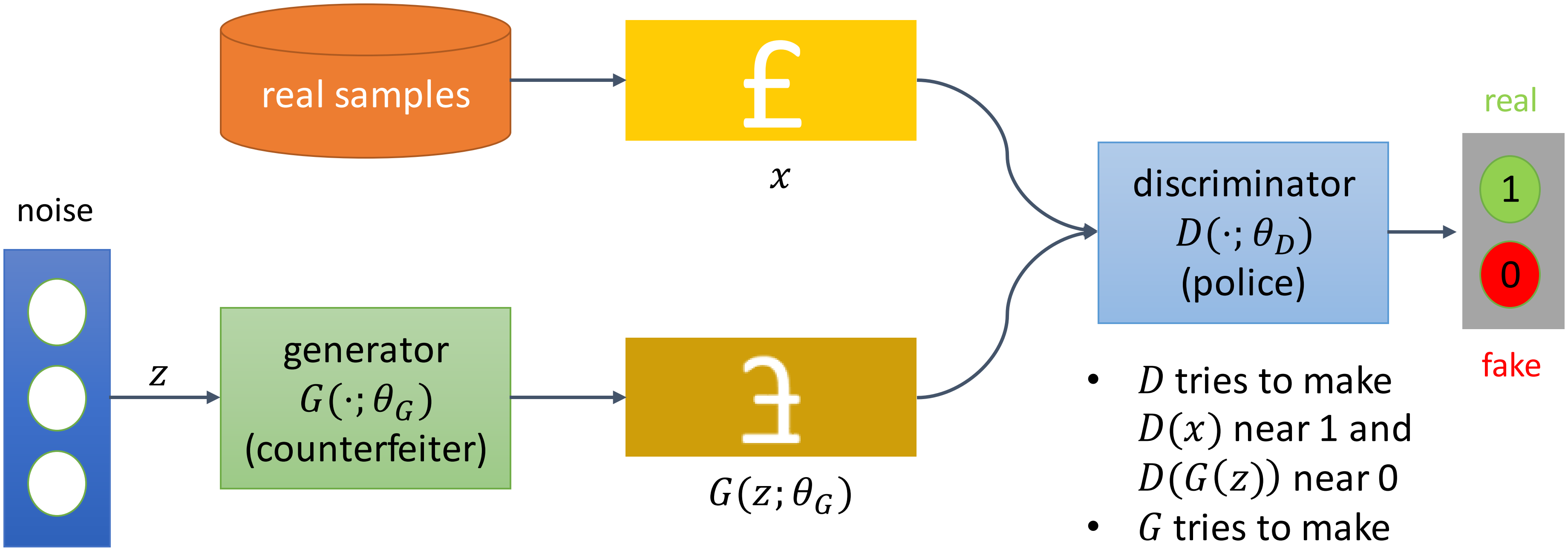
$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(x \mid \theta)$$

Taxonomy of generative methods



(Goodfellow 2016)

Generative adversarial networks



- D tries to make $D(x)$ near 1 and $D(G(z))$ near 0
- G tries to make $D(G(z))$ near 1

Minimax game

$$V(\theta_G, \theta_D) = \frac{1}{2}\mathbb{E}_{x \sim p_{data}} log D(x; \theta_D) + \frac{1}{2}\mathbb{E}_{z \sim p_z} \log\left(1 - D\big(G(z; \theta_G)\big)\right)$$

- $\min_{\theta_G} \max_{\theta_D} \mathrm{V}(\theta_G, \theta_D)$
- D wishes to maximise $\mathrm{V}(\theta_G, \theta_D)$ and controls θ_D
- G wishes to minimise $\mathrm{V}(\theta_G, \theta_D)$ and controls θ_G
- Solution to optimization at local minimum
- Game of two players with a solution at Nash equilibrium

Non-saturating game

$$J^D = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} logD(x;\theta_D) - \frac{1}{2}\mathbb{E}_{z\sim p_z}\log\Big(1 - D\big(G(z;\theta_G)\big)\Big)$$

$$J^G = \frac{1}{2}\mathbb{E}_{z\sim p_z}\log\Big(1 - D\big(G(z;\theta_G)\big)\Big)$$

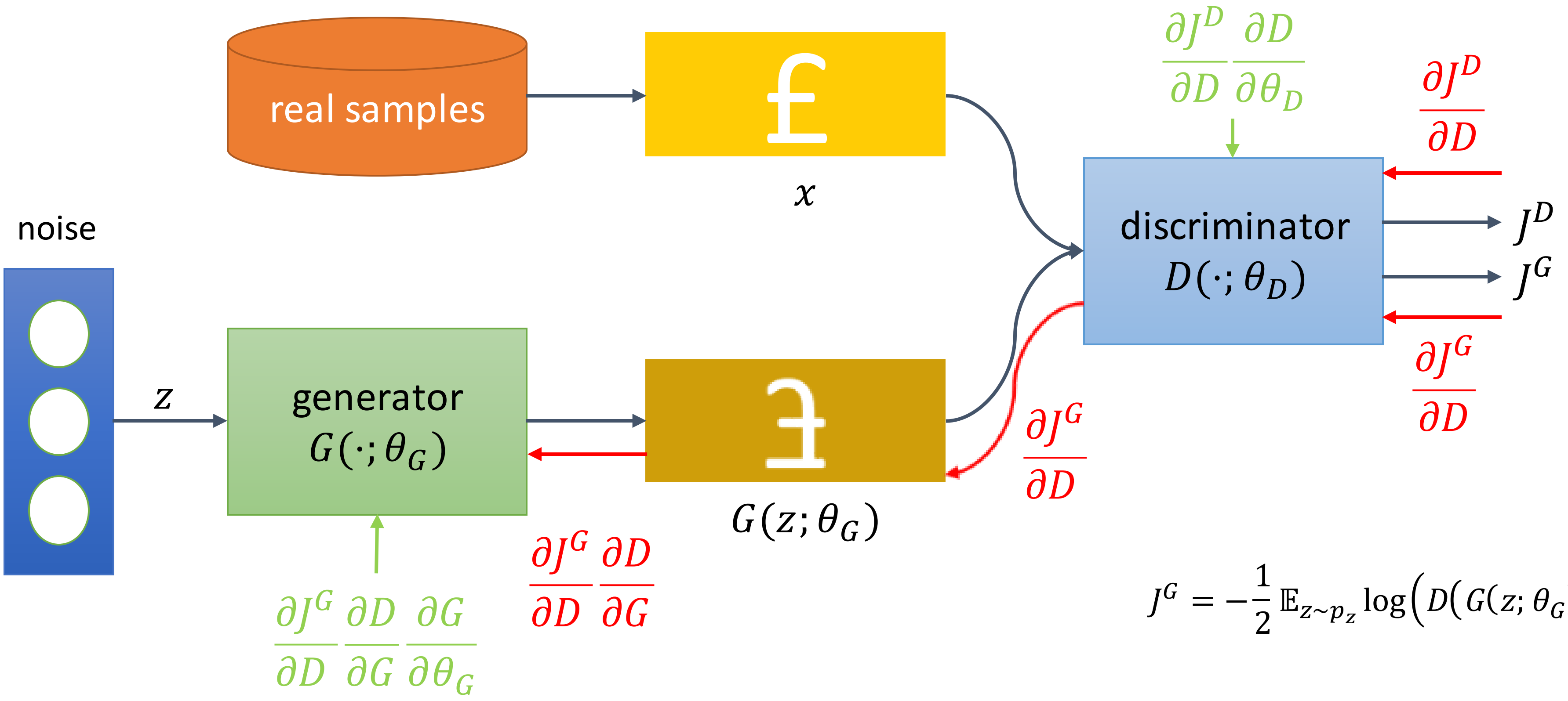
- Problem: when D successfully rejects generator samples, generator's gradient vanishes

- Solution:

$$J^G = -\frac{1}{2}\mathbb{E}_{z\sim p_z}\log\Big(D\big(G(z;\theta_G)\big)\Big)$$

Training GANs

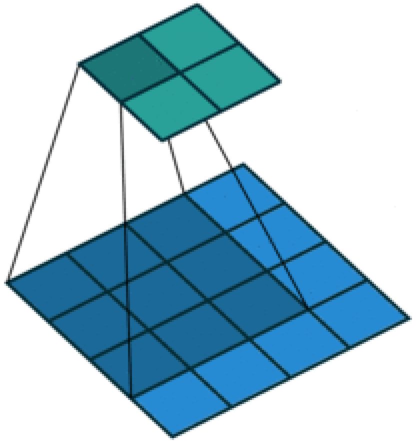
$$J^D = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} log D(x;\theta_D) - \frac{1}{2}\mathbb{E}_{z\sim p_z}\log\Big(1 - D\big(G(z;\theta_G)\big)\Big)$$



$$J^G = -\frac{1}{2}\mathbb{E}_{z\sim p_z}\log\Big(D\big(G(z;\theta_G)\big)\Big)$$

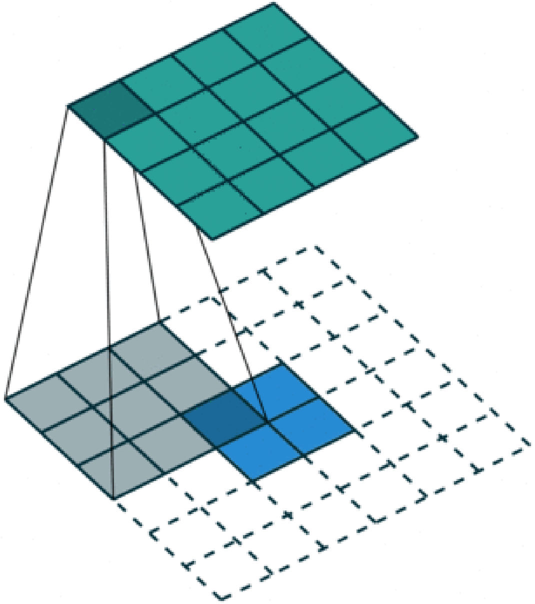
Review: Transposed convolution (deconv)

Convolution



Stride=1

Transpose convolution

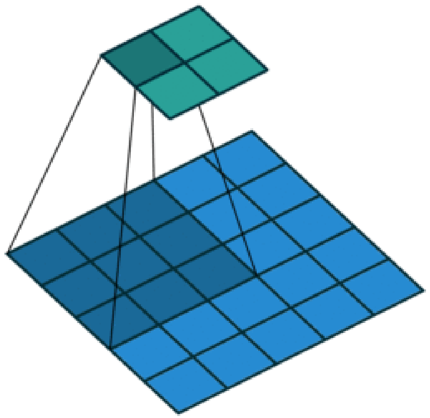


Stride=1

[Dumoulin and Vision (2016), A guide to convolution arithmetic for deep learning, arXiv]

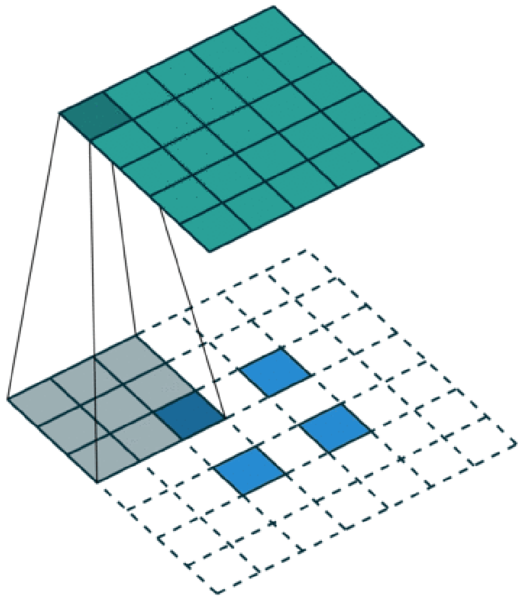
Review: Transposed convolution (deconv)

Convolution



Stride=2

Transpose convolution

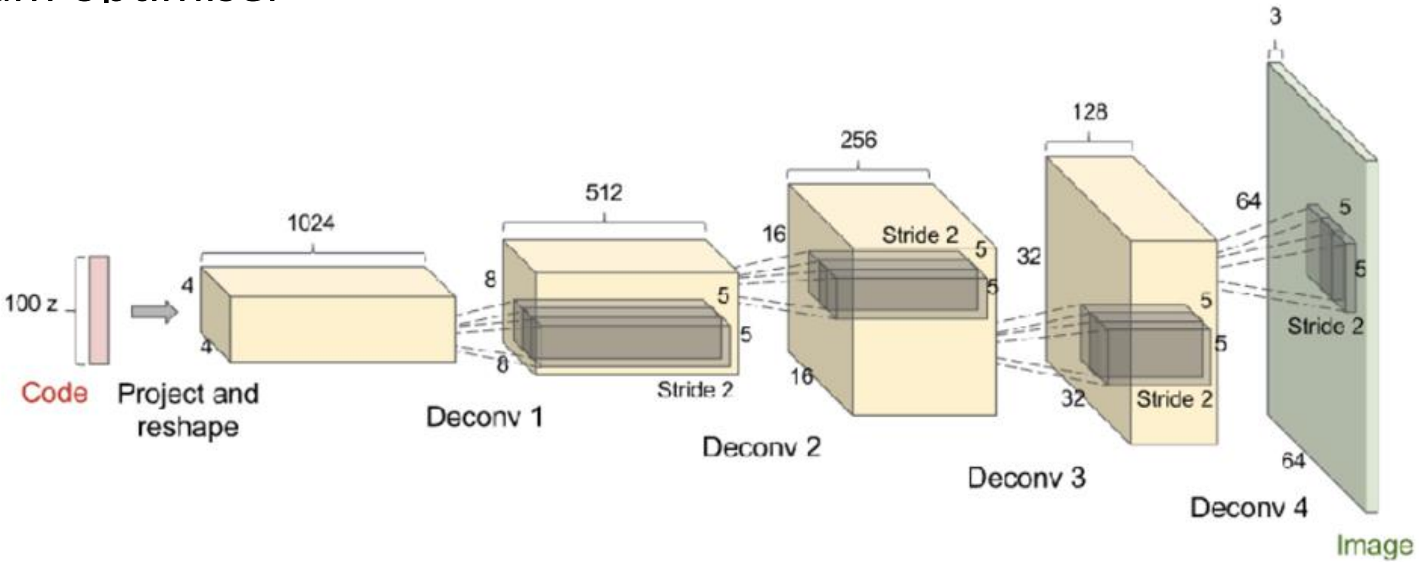


Stride=2

Dumoulin and Vision (2016), A guide to convolution arithmetic for deep learning, arXiv

DCGAN (generator) architecture

- Deconv weights are learned (unlike Zeiler & Fergus, see lecture 12)
- Deconvs are batch normalized
- No fully connected layer
- Adam optimiser



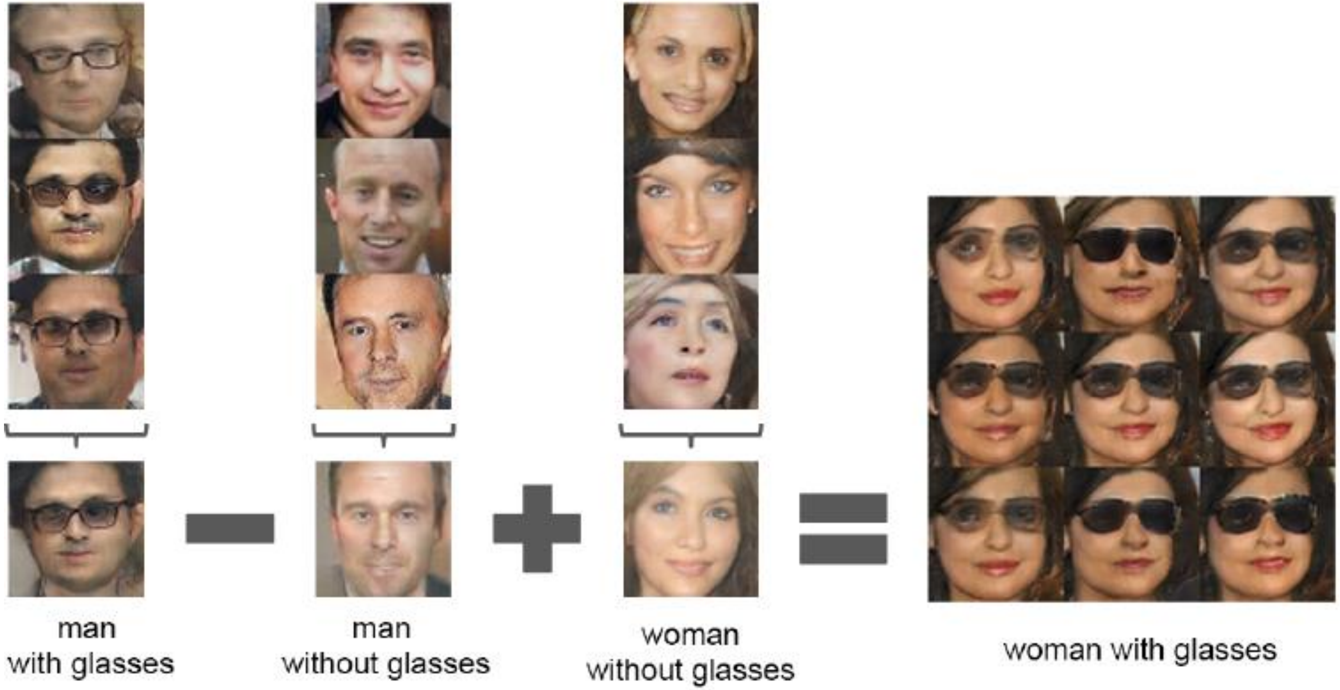
[Radford et al, (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.]

DCGAN for LSUN bedrooms

High quality images on restricted domain

Radford et al (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

Vector arithmetic on face samples



Semantically meaningful latent space

[Radford et al (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.]

Problems with counting and global structure

(Goodfellow 2016)

[Salimans et al, S. (2016). Improved Techniques for Training GANs. NeurIPS.]

Mode collapse

- Another failure mode is for the generator to collapse to a single mode
- If the generator learns to render only one realistic object, this can be enough to fool the discriminator

Salimans et al, S. (2016). Improved Techniques for Training GANs. NeurIPS.

Measuring GAN performance

Ask humans to distinguish between generated data and real data
- Subjective, requires training to spot flaws

Automating the evaluation (Inception score):
1. Image quality: images should contain meaningful objects (few dominant objects)
   - Feed images to Inception Net to obtain conditional label distribution $p(y|x)$
   - For each generated image, entropy over classes $\sum_y p(y|x) log\, p(y|x)$ should be low

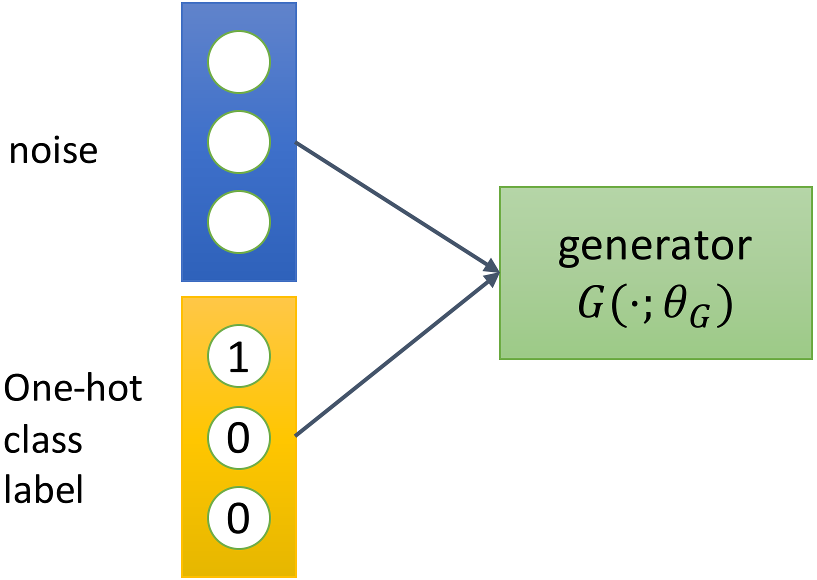
2. Diversity: the model should generate varied images (balanced over classes)
   - Entropy over generated images $\int p(y|x = G(z))dz$ should be high

- Combining two requirements ("how different is the score distribution for a generated image from the overall class balance?")

$$E_x KL(p(y|x)||p(y))$$

Salimans et al, S. (2016). Improved Techniques for Training GANs. NeurIPS.

Class-conditional GANs



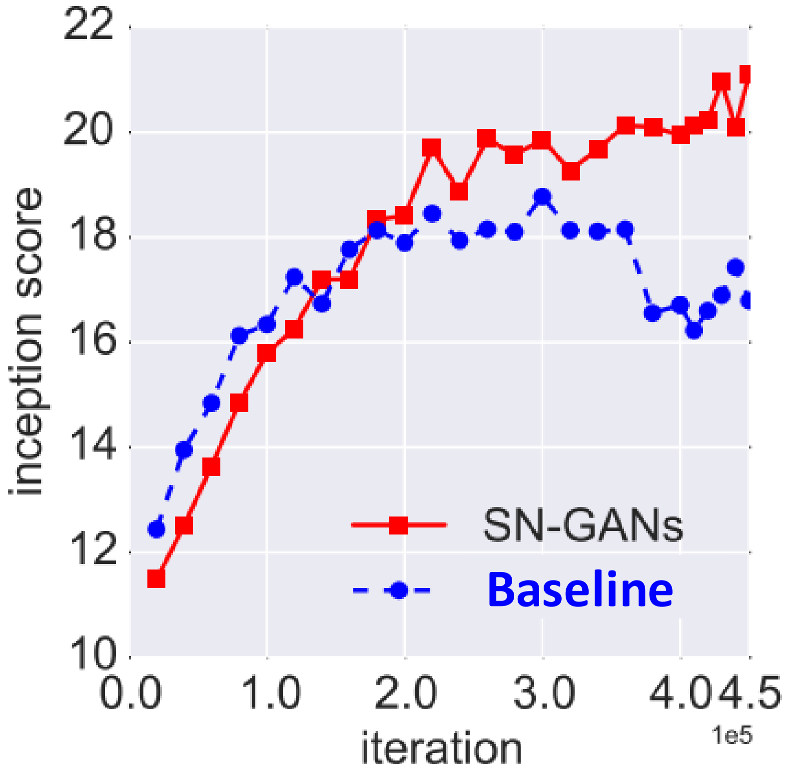
- Add class information in latent code z
- It is not unsupervised anymore

[Mirza and Osindero (2014). Conditional Generative Adversarial Nets. arXiv.]

Spectral normalization

- GAN training can be unstable
- **Culprit**: Gradient updates for G are unbounded
- **Observation:** First layer weights of G are ill-behaved when the training is unstable
- **Solution:** Apply spectral norm

- Spectral norm is the square root of the maximum eigenvalue of $W^T W$
  - $\sigma(W) = max \frac{|Wh|_2}{|h|_2} \rightarrow W/\sigma(W)$



Miyato et al. (2018). Spectral Normalization for GANs. ICLR.

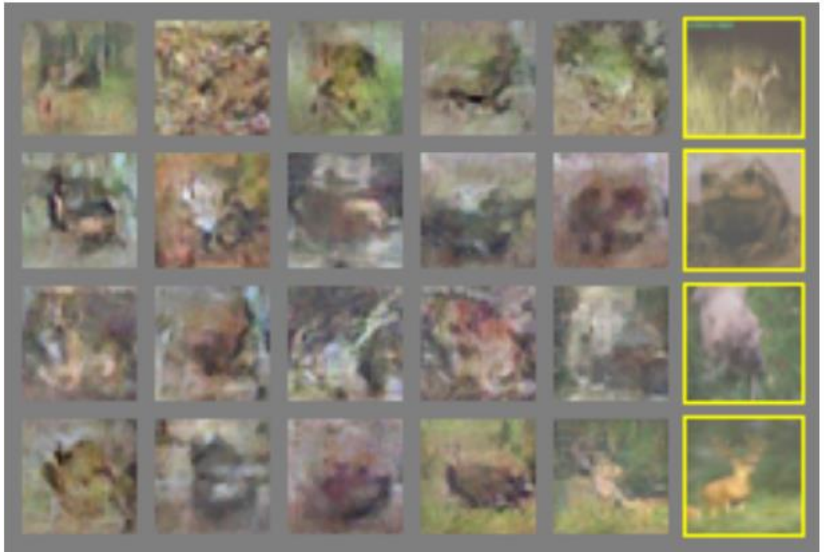
Scaling up GANs

Brock et al show that
- bigger batch sizes
- more number of filter channels (~50% more)
- larger dataset (292 million images with 8.5k images, 512 TPU cores!)

improves state-of-the-art around 35% in Inception Score

Brock et al. (2019). Large Scale GAN Training for High Fidelity Natural Image Synthesis. ICLR.

2014 to 2019



Goodfellow et al. (2014).

Figure 5: Samples generated by our model at 256×256 resolution. Sample sheets are available here.

Figure 6: Additional samples generated by our model at 512×512 resolution.

Brock et al. (2019).

Summary

- Generative models
- Training GANs
- DCGAN Architecture
- (Open) challenges
- Improvements

Reading material

Recommended
- Goodfellow et al, (2014). Generative adversarial nets. NeurIPS.
- Longer version Goodfellow, (2016). NIPS 2016 Tutorial: Generative Adversarial Networks.

Extra
- Radford et al, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.
- Salimans et al, S. (2016). Improved Techniques for Training GANs. NeurIPS.
- Miyato et al. (2018). Spectral Normalization for GANs. ICLR.
- Brock et al. (2019). Large Scale GAN Training for High Fidelity Natural Image Synthesis. ICLR.