Generative adversarial networks

Jan J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio

Département d'informatique et de recherche opérationnelle Université de Montréal

Presentation was made by Dmytro Semenchenko

Overview

1. Discriminative and generative approaches

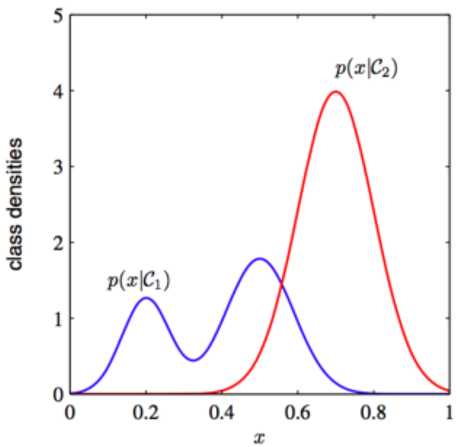
2. Generative modelling: Generally
    1. Main idea
    2. Generative models: Related work

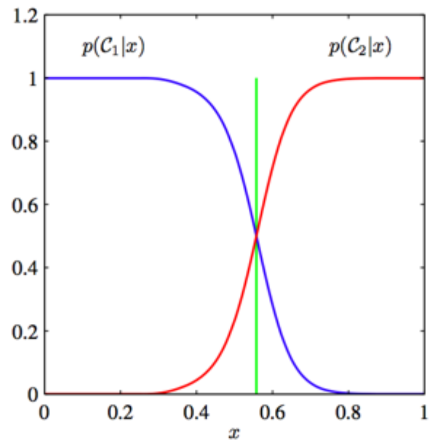
3. Generative adversarial networks
    1. Main idea
    2. Algorithm
    3. Examples of working
    4. Advantages & Disadvantages
    5. Comparison with other works
    6. Where used

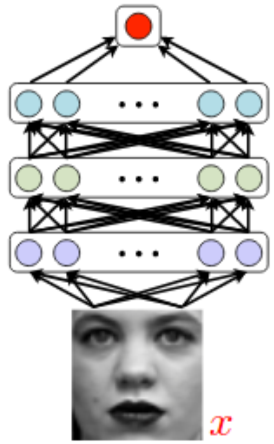
Discriminative vs Generative

There are two main approaches in statistical classification which called *discriminative* and *generative*.

For tasks such as classification and regression discriminative models can yield superior performance (in part because they have fewer variables to compute). On the other hand, generative models allow to generate samples from the joint distribution of observed and target variables. In addition, most discriminative models are inherently supervised and cannot easily support unsupervised learning.

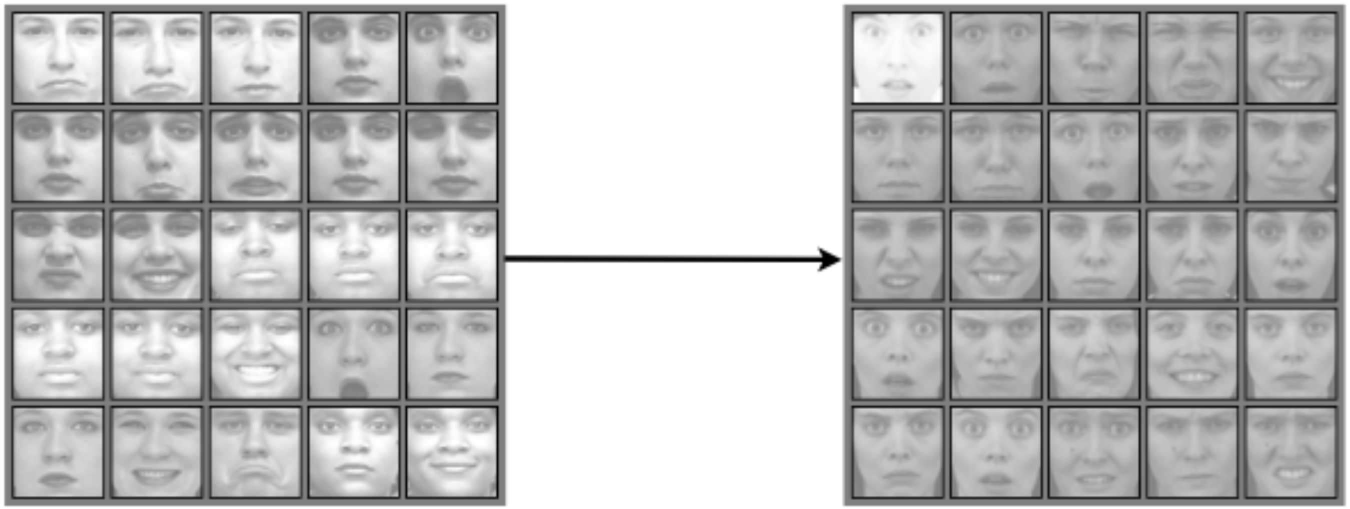






Generative modelling

- Have training examples $x \sim p_{data}(x)$
- Want a model that can draw samples: $x \sim p_{model}(x)$
- Where $p_{data}(x) \approx p_{model}(x)$

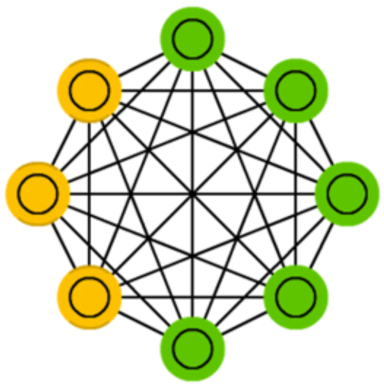


$x \sim p_{data}(x)$ $x \sim p_{model}(x)$

Generative models

(flagship undirected graphical model)

Boltzmann machine (BM)



Intractable likelihood functions ⇒ numerous approximations to the likelihood gradient;

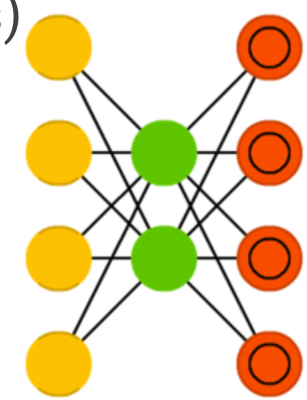
Badly parameterized for learning high quality samples: peaked distributions ⇒ slow mixing

Exact backpropagation rather than the numerous approximations required for Boltzmann machines;

But the same constraints on mixing as undirected graphical models

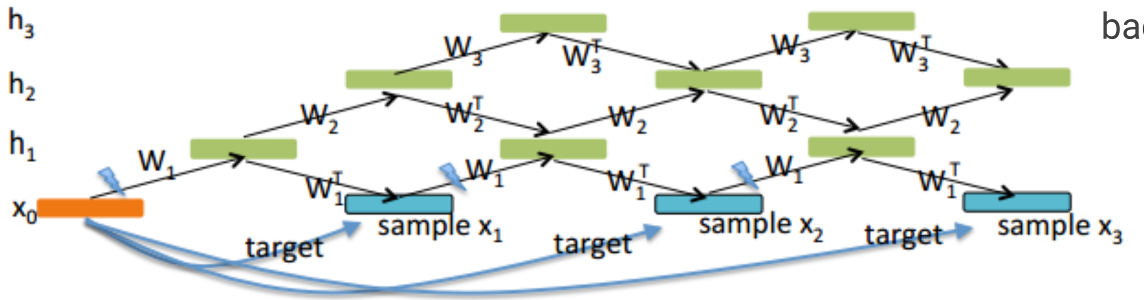
(directed graphical model; also 2014)

Variational autoencoders (VAEs)



Use a reparametrization that allows them to train very efficiently with gradient backpropagation.

Generative stochastic networks (GSN)



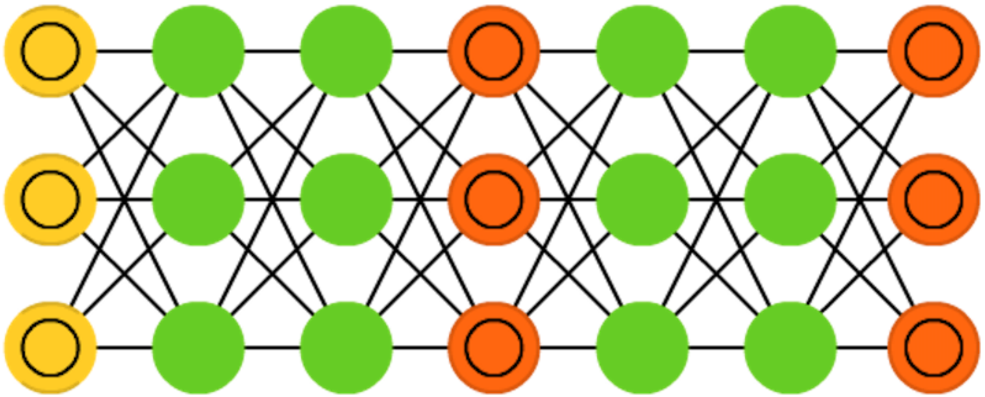
Generative adversarial networks (GANs): Idea

- Do not write a formula for *p(x)*, just learn to sample directly (GSN: Do not write a formula for *p(x)*, just learn to sample incrementally)

- No Markov Chain (GSN: Markov Chain is required)

- No variational bound

Generative adversarial networks (GANs)



Main idea: minimax two-player game of two neural networks.

- We train two models: generative model *G* and discriminative model *D*.
- *G* generates some input sample for *D*.
- *D* is trying to distinguish samples from the training data and samples generated by *G*.
- The goal of *G* is to maximize the probability of *D* making a mistake.

G
input: some noise *z*;
output: input sample for *D*

goal: maximize *D(G(z))*
⇨ *log(D(G(z))*

D
input: sample from training data *x* or *G(z)*
output: the probability that a sample came from the training data rather than *G*
goal: maximize *D(x)(1 - D(G(z)))*
⇨ *log(D(x)) + log(1 - D(G(z)))*

GANs. More formally.

$p_{\boldsymbol{z}}(\boldsymbol{z})$ — input noise variables

p_g — generator's data distribution

p_{data} — real data distribution

$G(\boldsymbol{z};\theta_g)$ — a differential function represented by multilayer perceptron

$D(\boldsymbol{x};\theta_d)$ — another multilayer perceptron. *D(x)* represent the probability that *x* from the training data rather than p_g

Value function (two-player minimax game):

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

Algorithm

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

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

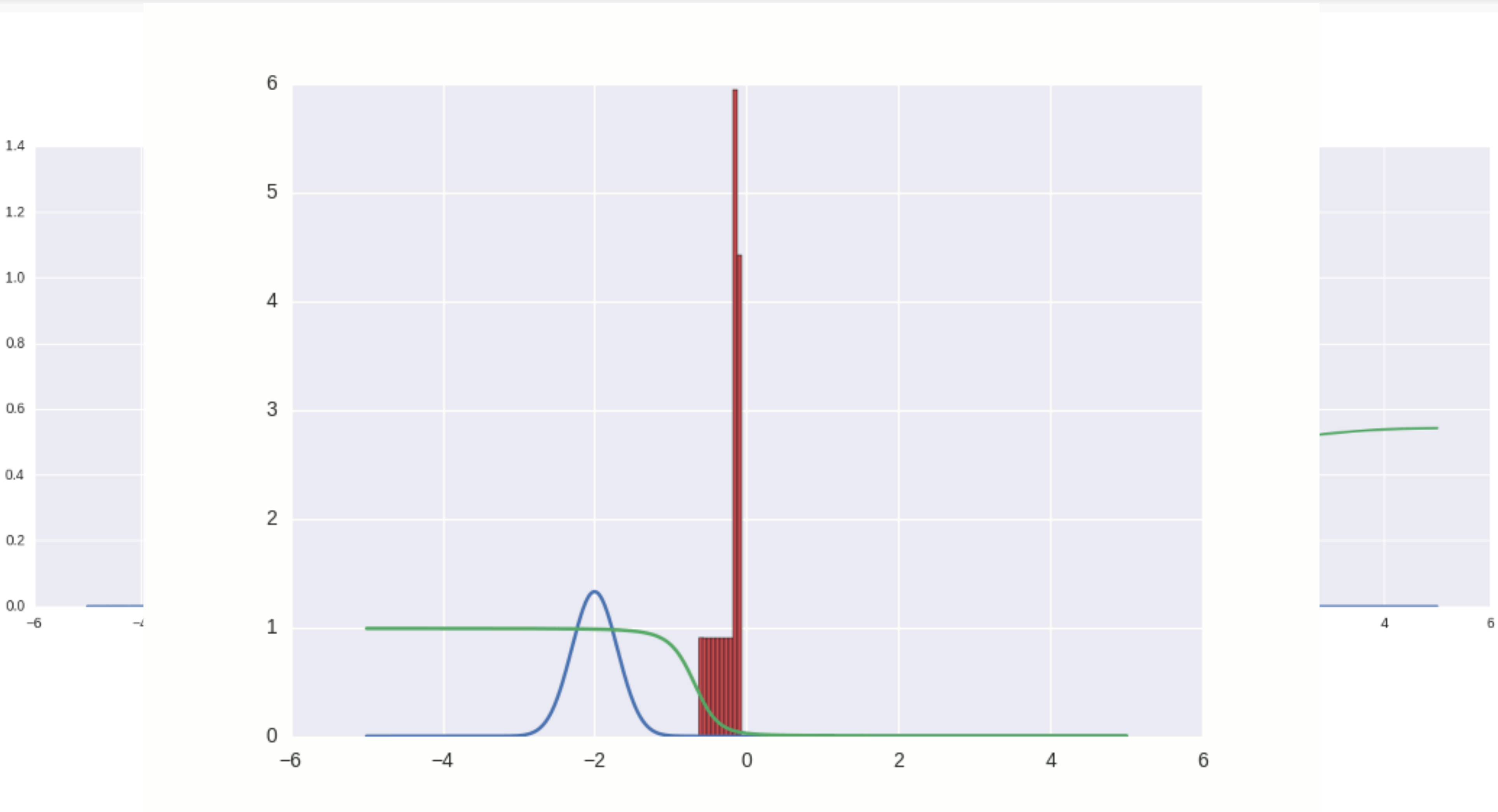
end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

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

end for

Illustration



Advantages and disadvantages

+

- No markov chains are needed
- No inference is needed during learning
- A wide variety of functions can be incorporated into the model
- Can represent very sharp distributions (Markov chains based methods require some blurry distribution)
- Can gain some statistical advantage

—

- no explicit representation of p_g
- Synchronization between *D* and *G*

Comparison

	Deep directed graphical models	Deep undirected graphical models	Generative autoencoders	Adversarial models
Training	Inference needed during training.	Inference needed during training. MCMC needed to approximate partition function gradient.	Enforced tradeoff between mixing and power of reconstruction generation	Synchronizing the discriminator with the generator.
Inference	Learned approximate inference	Variational inference	MCMC-based inference	Learned approximate inference
Sampling	No difficulties	Requires Markov chain	Requires Markov chain	No difficulties
Evaluating *p(x)*	Intractable, may be approximated with AIS	Intractable, may be approximated with AIS	Not explicitly represented, may be approximated with Parzen density estimation	Not explicitly represented, may be approximated with Parzen density estimation
Model design	Models need to be designed to work with the desired inference scheme	Careful design needed to ensure multiple properties	Any differentiable function is theoretically permitted	Any differentiable function is theoretically permitted

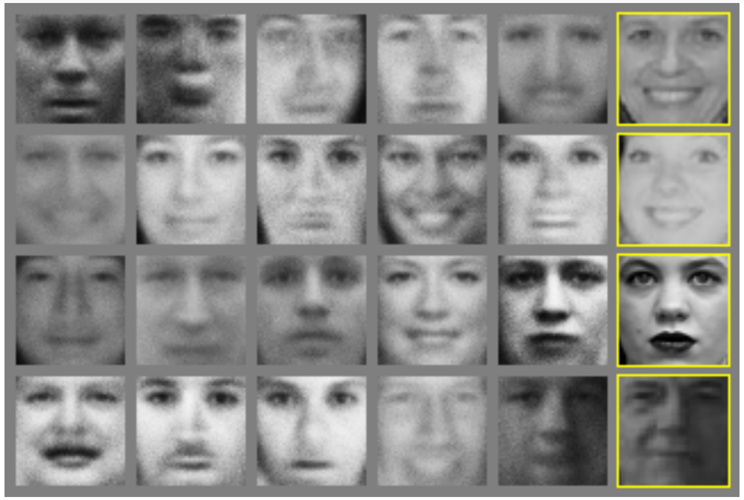
Comparison. Experiments

Model	MNIST (database of handwritten digits)	TFD (Toronto Face Database)
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [5]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

Parzen window-based log-likelihood estimates.
The reported numbers on MNIST are the mean log-likelihood of samples on test set, with the standard error of the mean computed across examples.
On TFD, was computed the standard error across folds of the dataset, with a different σ chosen using the validation set of each fold. On TFD, σ was cross validated on each fold and mean log-likelihood on each fold were computed.

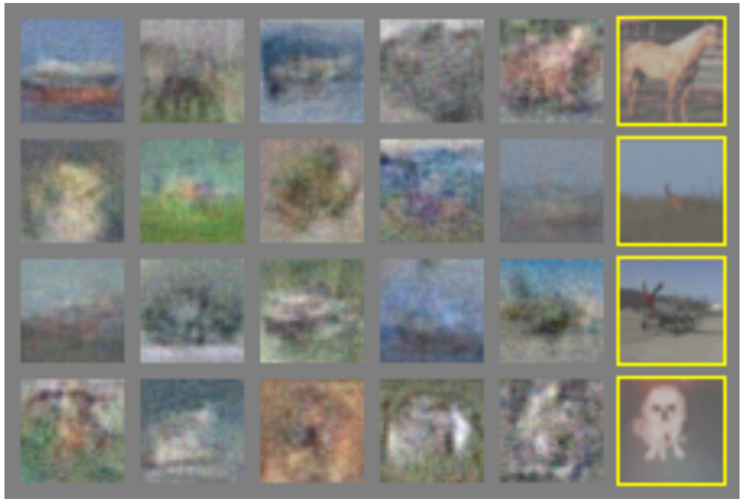
Experiments. Illustration

MNIST



TDF

CIFAR-10

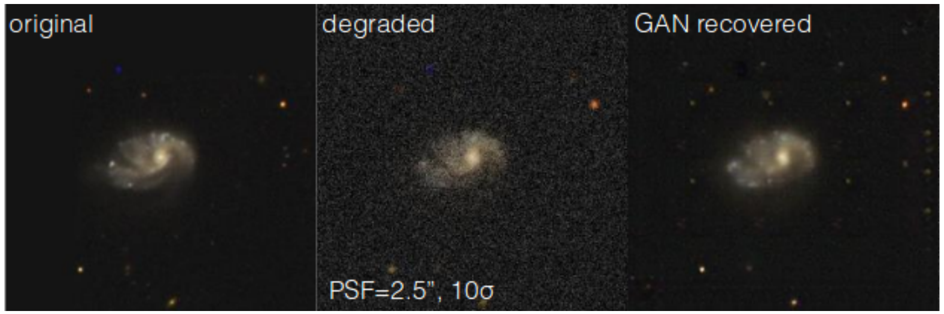


What tasks are GANs used for?

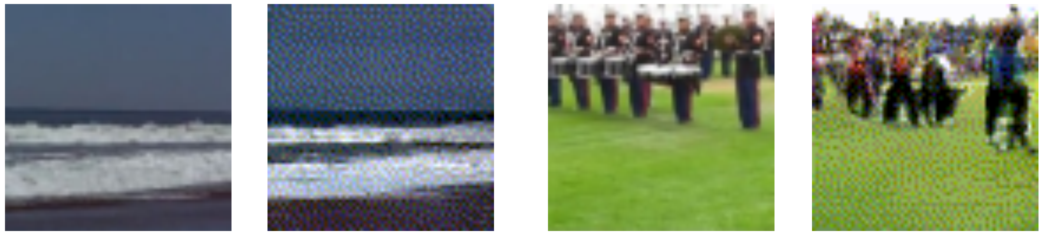
Reconstruction 3D models of objects from images

Fake videos

Improvement of astronomical images



Generating videos with scene dynamics



References

- Goodfellow ,I. ;Pouget-Abadie ,J. ; Mirza , M. ; Xu , B. ; Warde-Farley ,D. ; Ozair , S. ; Courville , A. ; Bengio , Y. :Generative Adversarial Nets. Version: 2014. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf . In: Ghahramani , Z. (Hrsg.) ;Welling , M. (Hrsg.) ; Cortes , C. (Hrsg.) ; Lawrence , N. D. (Hrsg.) ;Weinberger , K. Q. (Hrsg.): Advances in Neural Information Processing Systems 27 . Curran Associates, Inc., 2672-2680
- Ng, A. Y.; Jordan, M. I.: On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In: *Advances in neural information processing systems , 2002, S. 841848*
- Schawinski, Kevin, Zhang, Ce, Zhang, Hantian, Fowler, Lucas & Santhanam, Gokula Krishnan (2017). Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Monthly Notices of the Royal Astronomical Society: Letters, 467*, L110-L114
- Asimov Institute. 2016. *The Neural Network Zoo*. [ONLINE] Available at: http://www.asimovinstitute.org/neural-network-zoo/. [Accessed 15 November 2018].
- Artem Hurshudov. 2016. *Counterfeiters against bankers: pitting adversarial networks in Theano*. [ONLINE] Available at: https://habr.com/post/275429/. [Accessed 15 November 2018].
- Carl Vondrick, Hamed Pirsiavash, Antonio Torralba. 2016. *Generating Videos with Scene Dynamics*. [ONLINE] Available at: http://www.cs.columbia.edu/~vondrick/tinyvideo/. [Accessed 15 November 2018].