Generative Adversarial Nets

Schuiki Johannes, Kempnter Stefan

January 31, 2020

Disclaimer

The main aspects and ideas of this presentation are based on the paper "Generative Adversarial Networks" from Ian Goodfellow et al. (2014) [2].

Contents

Introduction

This person does not exist!

Figure: Result of StyleGAN. Src.:https://www.thispersondoesnotexist.com/ [1]

Prerequisites

The Perceptron

$$f(\mathbf{x};\mathbf{w}) = \phi\left(\sum_{i=1}^{n} x_i w_i + b\right) = y$$

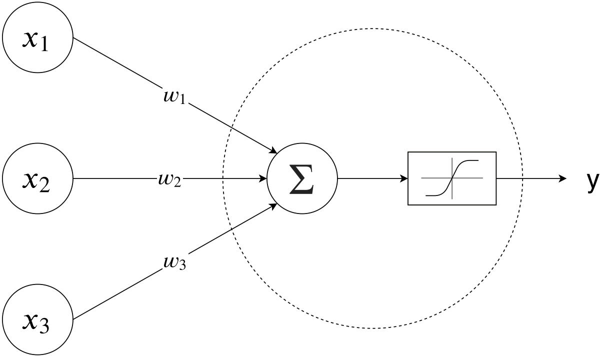

Figure: A perceptron with three inputs x_1 to x_3 and an activation function ϕ

Neural Networks / Multilayer Perceptron

$$f(\mathbf{x}; \mathbf{w}) = ? = y$$

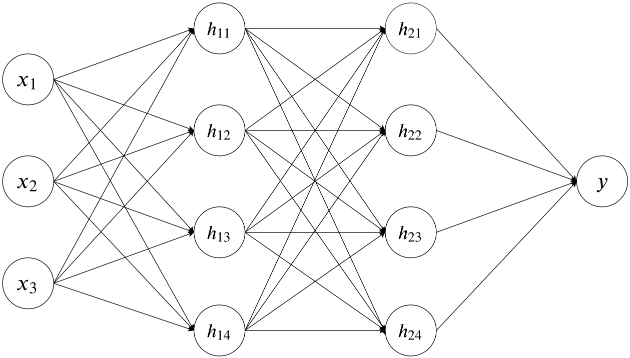

Figure: A multi-layer-perceptron with three inputs x_1 to x_3 and an activation function ϕ and hidden layers h

Forward Pass

$$h_{1i} = \phi\left(\sum_j x_j w_{1ji}\right)$$

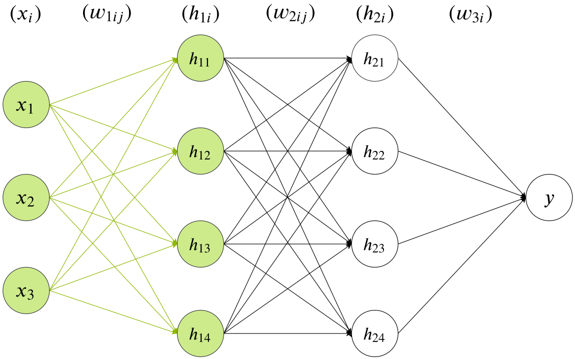

Figure: A multi-layer-perceptron with three inputs x_1 to x_3 and an activation function ϕ and hidden layers h

Forward Pass

$$h_{2i} = \phi\left(\sum_j h_{1j} w_{2ji}\right)$$

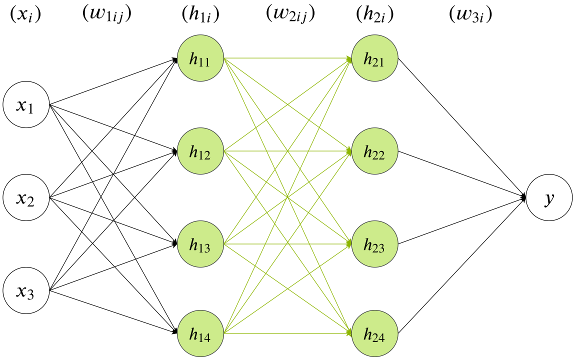

Figure: A multi-layer-perceptron with three inputs x_1 to x_3 and an activation function ϕ and hidden layers h

Forward Pass

$$y = \phi\left(\sum_j h_{2j} w_{3j}\right)$$

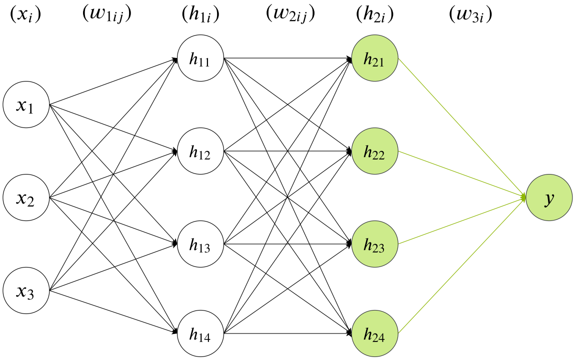

Figure: A multi-layer-perceptron with three inputs x_1 to x_3 and an activation function ϕ and hidden layers h

Forward Pass

$$y = f(\mathbf{x};\mathbf{w}) = \phi\left(\sum_i \phi\left(\sum_j \phi\left(\sum_k x_k w_{1kj}\right) w_{2ji}\right) w_{3i}\right)$$

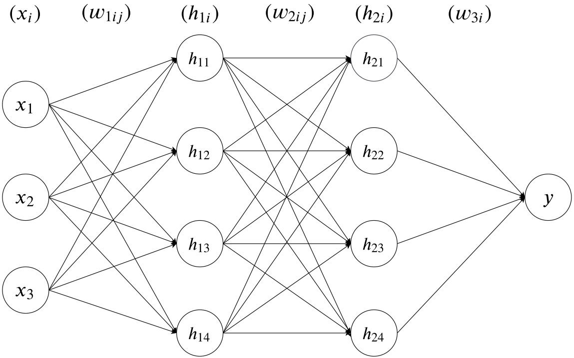

Figure: A multi-layer-perceptron with three inputs x_1 to x_3 and an activation function ϕ and hidden layers h

Activation Function ϕ

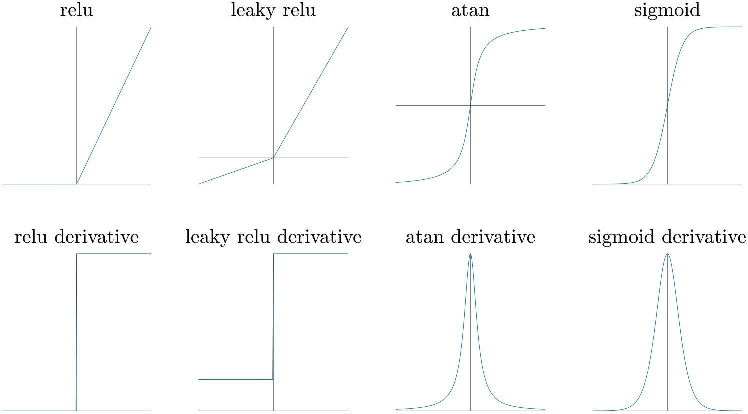

Figure: Different activation functions and their derivatives.

Forward Pass

$$f(\mathbf{x};\mathbf{w}) = \phi\left(\sum_i \phi\left(\sum_j \phi\left(\sum_k x_k w_{1kj}\right) w_{2ji}\right) w_{3i}\right)$$

f is differentiable!

Objective function (Cost/Loss)

We use a loss function (in this case Sum of Squared Differences) as a quality metric of the net.

$$L(\mathbf{x};\mathbf{w}) = \sum_{n=1}^{N} (f(\mathbf{x};\mathbf{w}) - target_n)^2$$

Gradient Descent

$$\mathbf{w}^{(\mathbf{i}+1)} = \mathbf{w}^{(\mathbf{i})} - \eta \frac{\partial L(\mathbf{w})}{\partial \mathbf{w}}$$

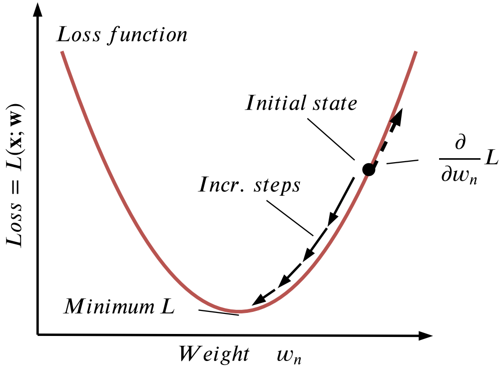

Figure: Simplified illustration of stochastic gradient descent with one weight.

Back propagation

All we need: Chain Rule

$$y = f(u) \qquad u = g(x)$$

$$\frac{dy}{dx} = \frac{dy}{du} * \frac{du}{dx}$$

Back propagation

$$\frac{\partial loss}{\partial w_{1ij}}$$

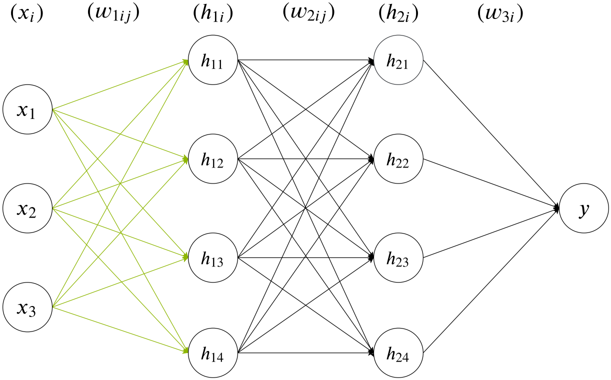

Figure: A multi-layer-perceptron with three inputs x_1 to x_3 and an activation function ϕ and hidden layers h

Back propagation

We remember:

$$L(\mathbf{x};\mathbf{w}) = \sum_{n=1}^{N} (f(\mathbf{x};\mathbf{w}) - target_n)^2$$

$$f(\mathbf{x};\mathbf{w}) = \phi(\sum_i \underbrace{\phi(\sum_j \underbrace{\phi(\sum_k x_k w_{1kj})}_{h1} w_{2ji})}_{h2} w_{3i})$$

We calculate:

$$\frac{\partial L(\mathbf{x};\mathbf{w})}{\partial w_{1ij}} = \frac{\partial L(\mathbf{x};\mathbf{w})}{\partial f(\mathbf{x};\mathbf{w})} \dots$$

Back propagation

We remember:

$$L(\mathbf{x};\mathbf{w}) = \sum_{n=1}^{N}(f(\mathbf{x};\mathbf{w}) - target_n)^2$$

$$f(\mathbf{x};\mathbf{w}) = \phi(\sum_i \underbrace{\phi(\sum_j \underbrace{\phi(\sum_k x_k w_{1kj})}_{h1} w_{2ji})}_{h2} w_{3i})$$

We calculate:

$$\frac{\partial L(\mathbf{x};\mathbf{w})}{\partial w_{1ij}} = \frac{\partial L(\mathbf{x};\mathbf{w})}{\partial f(\mathbf{x};\mathbf{w})} \frac{\partial f(\mathbf{x};\mathbf{w})}{\partial h2} \dots$$

Back propagation

We remember:

$$L(\mathbf{x};\mathbf{w}) = \sum_{n=1}^{N}(f(\mathbf{x};\mathbf{w}) - target_n)^2$$

$$f(\mathbf{x};\mathbf{w}) = \phi(\sum_i \underbrace{\phi(\sum_j \underbrace{\phi(\sum_k x_k w_{1kj})}_{h1} w_{2ji})}_{h2} w_{3i})$$

We calculate:

$$\frac{\partial L(\mathbf{x};\mathbf{w})}{\partial w_{1ij}} = \frac{\partial L(\mathbf{x};\mathbf{w})}{\partial f(\mathbf{x};\mathbf{w})} \frac{\partial f(\mathbf{x};\mathbf{w})}{\partial h2} \frac{\partial h2}{\partial h1} \dots$$

Back propagation

We remember:

$$L(\mathbf{x};\mathbf{w}) = \sum_{n=1}^{N} (f(\mathbf{x};\mathbf{w}) - target_n)^2$$

$$f(\mathbf{x};\mathbf{w}) = \phi(\sum_i \underbrace{\phi(\sum_j \underbrace{\phi(\sum_k x_k w_{1kj})}_{h1} w_{2ji})}_{h2} w_{3i})$$

We calculate:

$$\frac{\partial L(\mathbf{x};\mathbf{w})}{\partial w_{1ij}} = \frac{\partial L(\mathbf{x};\mathbf{w})}{\partial f(\mathbf{x};\mathbf{w})} \frac{\partial f(\mathbf{x};\mathbf{w})}{\partial h2} \frac{\partial h2}{\partial h1} \frac{\partial h1}{\partial w_{1ij}}$$

Generative Adversarial Nets

Generative Models

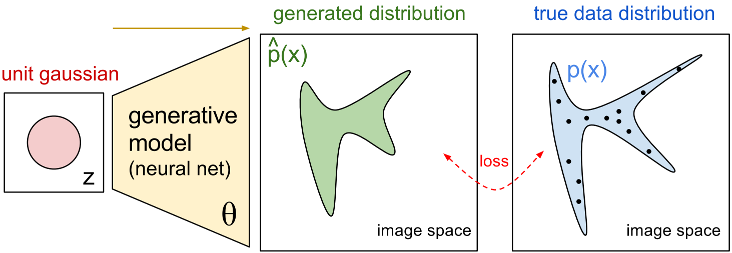

Figure: The generative model approximates the true data distribution by mapping z p_z to the data-space. Src.:https://openai.com/blog/generative-models/

The Generative Adversarial Framework

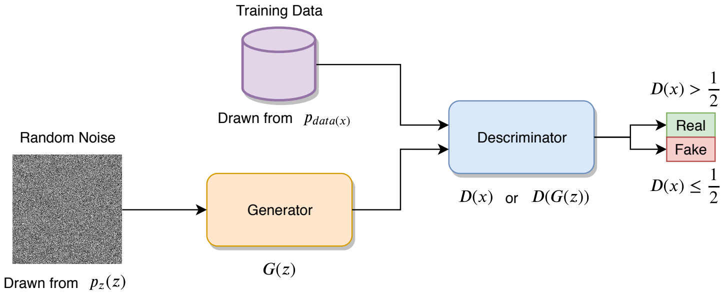

Figure: The components of the generative adversarial framework as discribed in [2].

Generative Adversarial Nets

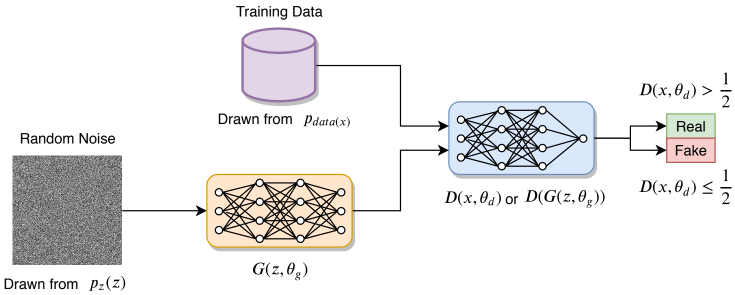

Figure: The components of generative adversarial nets as discribed in [2].

The objective function

A min max game!

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

The objective function

Discriminator perspective

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

The objective function

Generator perspective

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

for *number of training iterations* **do**

for *k steps* **do**

- Sample minibatch of m samples $\{z^{(1)}, ..., z^{(m)}\}$ from p_z.
- Sample minibatch of m samples $\{x^{(1)}, ..., x^{(m)}\}$ from p_{data}.
- Update the discriminator by **ascending** its stochastic gradient:

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

end

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

$$\nabla\theta_g \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)})))$$

end

for *number of training iterations* **do**

for *k steps* **do**

- Sample minibatch of m samples $\{z^{(1)}, ..., z^{(m)}\}$ from p_z.
- Sample minibatch of m samples $\{x^{(1)}, ..., x^{(m)}\}$ from p_{data}.
- Update the discriminator by **ascending** its stochastic gradient:

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

end

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

$$\nabla\theta_g \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)})))$$

end

Visualization of the Training Process

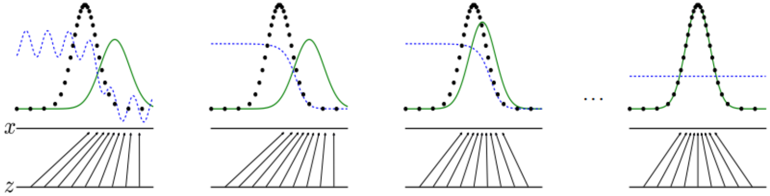

Figure: A GAN trying to approximate the normal distribution. Black represents the normal distribution. Green the distribution generated by the Generator and blue the discriminative distribution. Src.: [2]

Theoretical Results

Nice! But can you prove it?
Yes No Maybe?!

Global Optimality

- All proofs consider the non parametric setting. (Infinite capability of the nets.)
- At any step the proofs consider an optimal discriminator.
- Using the optimal discriminator they reformulate the criterion and show that global minimum of the criterion is given for $p_g = p_{data}$.

Convergence of the Algorithm

- The reformulated "virtual" training criterion uses the Jenson-Shannon divergence.
- The proofs argue that this divergence is convex for a fixed p_{data} and variable p_g.
- Since it is already proved that the global minimum is at $p_g = p_{data}$ the algorithm will converge towards it.

The reality

In practice, adversarial nets represent a limited family of p_g distributions via the function $G(z;\theta_g)$, and we optimize θ_g rather than p_g itself, so the proofs don't apply.[2, p.7]

Advantages

- Backpropagation can be used. No other methods and/or approximations due to intractable gradients required.
- Statistical: The generator never sees images from the original data distribution. Only the gradients. $\rightarrow$ No "memorization" possible
- GANs can model sharp and degenerate distributions while other models need the distrubution to be "smooth" to a certain extent.

Disadvantages

- There is no explicit representation of the distribution p_g
- The training of D and G has to be balanced
- Immense amounts of data are required for training.

Latent Space Arithmetic

- The space where you draw a random input vector for your generator is called 'Latent Space'
- Some directions in this space have semantic meaning

Figure: man with glasses - man + woman = woman with glasses. Screenshot from [3].

Change of Style

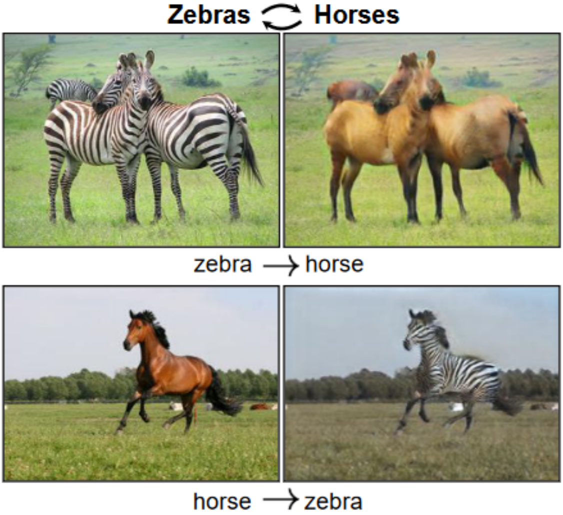

Figure: cycleGAN. Screenshot from [4].

HD Image generation from semantic map

Figure: Screenshot from [5].

Text to Image

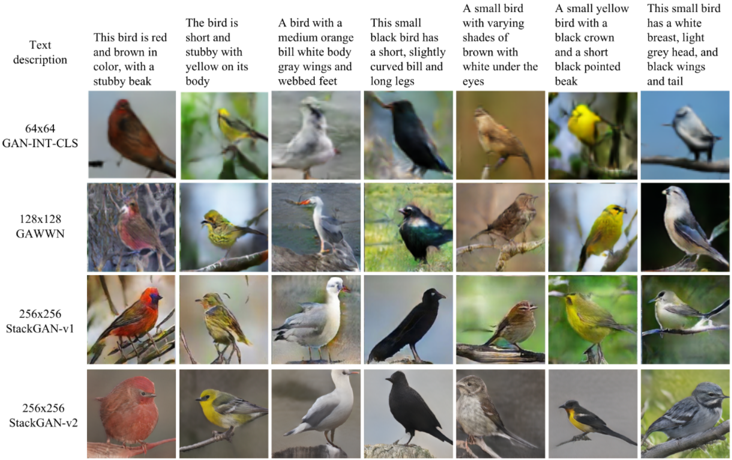

Figure: Screenshot from [6].

Ethical Issues: Deepfake

Figure: Screenshot from Youtube [7].

Thank You!

Resources I

T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *ArXiv*, vol. abs/1912.04958, 2019.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," *ArXiv*, vol. abs/1406.2661, 2014.

P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, "Optimizing the latent space of generative networks," 07 2017.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

Resources II

T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, 10 2017.

"https://www.youtube.com/watch?v=vwrhrbb-1ig."