4 Years of Generative Adversarial Networks (GANs)

Lu Lu

Crunch Seminar

Chinese New Year 2018

Overview

1. What is “adversarial”?
2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN
3. Paper review: Daskalakis, Training GANs with optimism, 2017
4. Boundary equilibrium GAN (BEGAN)
5. Adversarial examples

Do Not Be Afraid of GANs.

Neural network AI is simple.

By Brandon Wirtz (CEO and Founder at Recognant), Feb 15, 2018

- ... 99% of these things are completely stupid...
- **So you built a neural network from scratch And it runs on a phone**
  Great. So you converted 11 lines of python that would fit on a t-shirt... You have mastered what a cross compiler can do in 3 seconds.

What about GANs? 3%.

Overview

1. What is "adversarial"?
2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN
3. Paper review: Daskalakis, Training GANs with optimism, 2017
4. Boundary equilibrium GAN (BEGAN)
5. Adversarial examples

What Is Adversarial?

Evolve with competition

Deer-leopard minimax game

$$\min_{\text{leopard}} \max_{\text{deer}} V(\text{deer}, \text{leopard}) = \text{distance between deer and leopard}$$

What Doesn't Kill You Makes You Stronger!

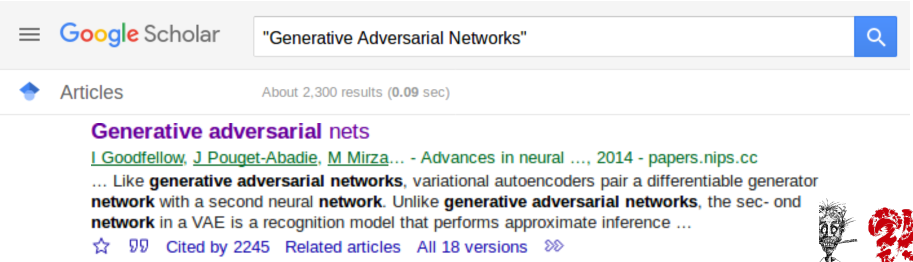
What Is Adversarial?

- Generative adversarial networks (GANs)

Overview

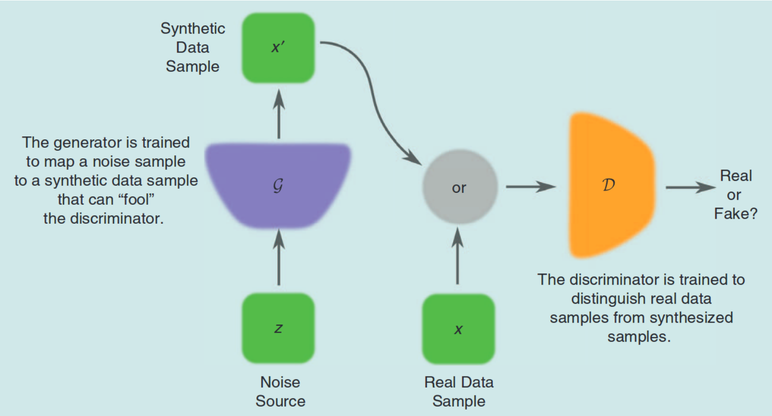
1. What is "adversarial"?
2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN
3. Paper review: Daskalakis, Training GANs with optimism, 2017
4. Boundary equilibrium GAN (BEGAN)
5. Adversarial examples

Since 2014,



Vanilla GAN

- Generator G: capture the data distribution (make realistic images)
- Discriminator D: estimate the probability that a sample came from the training data rather than G (tell real and fake images apart)

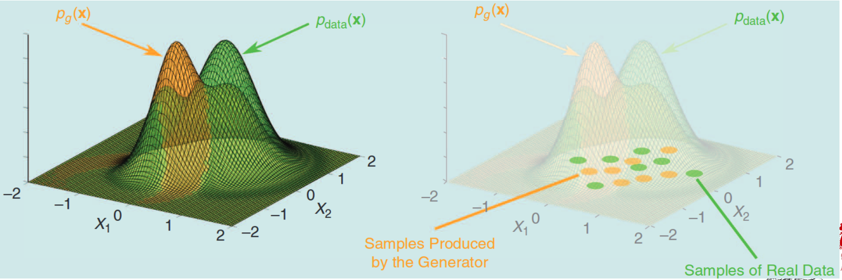


Vanilla GAN

- $p_{\mathbf{z}}(\mathbf{z})$: input noise
- $p_{\text{data}}(\mathbf{x})$: real data's distribution
- $p_g(\mathbf{x})$: generator's distribution of $G(\mathbf{z})$

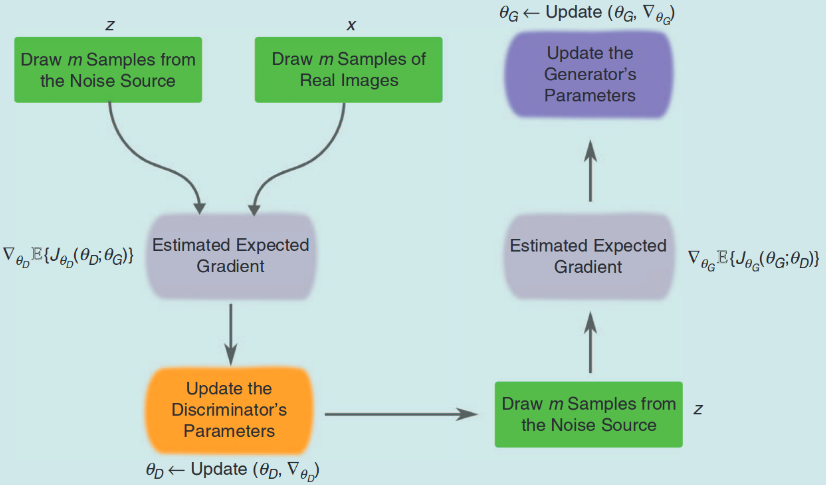
Two-player minimax game

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



Vanilla GAN

Main loop of GAN training



Vanilla GAN

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ 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 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ 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

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Vanilla GAN

Difficult to train [Salimans, 2016, Arjovsky, 2017a]

- Vanishing gradients
- Instability
- Model collapse
- ...

Improvements

17 tips to make GANs work (`https://github.com/soumith/ganhacks`)

- Use a spherical **z**
- Batch normalization [Ioffe, 2015]
- ...

GAN variants (>100)

- Deep convolutional GAN (DCGAN) [Radford, 2015]
- Conditional GAN [Mirza, 2014]
- Adversarially learned inference (ALI) [Dumoulin, 2016]
- Adversarial autoencoder (AAE) [Makhzani, 2015]
- Energy-based GAN (EBGAN) [Zhao, 2016]
- Wasserstein GAN (WGAN) [Arjovsky, 2017b]
- Boundary equilibrium GAN (BEGAN) [Berthelot, 2017]
- Bayesian GAN [Saatchi, 2017]
- ...

Overview

1. What is “adversarial”?
2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN
3. Paper review: Daskalakis, Training GANs with optimism, 2017
4. Boundary equilibrium GAN (BEGAN)
5. Adversarial examples

WGAN

Recall GAN

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

Earth-Mover (EM) distance or Wasserstein-1 [Monge, 1781]

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$

By Kantorovich-Rubinstein duality [Villani, 2008],

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{||f||_L \le 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

Discriminator f_w, generator g_θ

$$\min_\theta \max_{w\in\mathcal{W}} \mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))]$$

WGAN

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

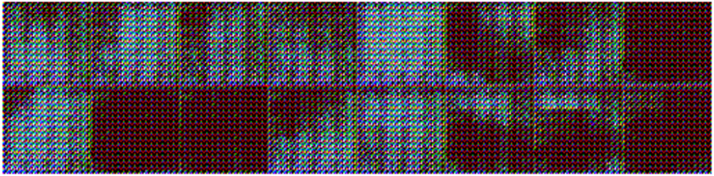
Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

WGAN

- Improved stability of learning
- Get rid of mode collapse
- Meaningful learning curves

GAN without tricks during training



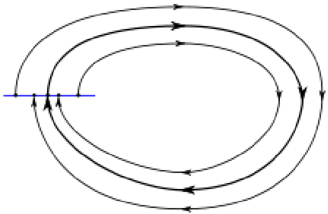
WGAN samples

Overview

1. What is "adversarial"?
2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN
3. Paper review: Daskalakis, Training GANs with optimism, 2017
4. Boundary equilibrium GAN (BEGAN)
5. Adversarial examples

Problem of WGAN

Limit cycling behavior in training (W)GAN



[https://en.wikipedia.org/wiki/Limit_cycle]

Gradient Descent (GD) vs. Optimistic Mirror Descent (OMD)

GD

$$w_{t+1} = w_t + \eta \cdot \nabla_{w,t}$$

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta,t}$$

OMD [Rakhlin, 2013]: $M_{\cdot,t+1}$ is a predictor of $\nabla_{\cdot,t}$

$$w_{t+1} = w_t + \eta \cdot (\nabla_{w,t} + M_{w,t+1} - M_{w,t})$$

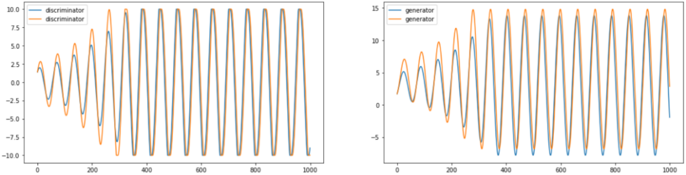
$$\theta_{t+1} = \theta_t - \eta \cdot (\nabla_{\theta,t} + M_{\theta,t+1} - M_{\theta,t})$$

In this paper, choose $M_{\cdot,t+1} = \nabla_{\cdot,t}$

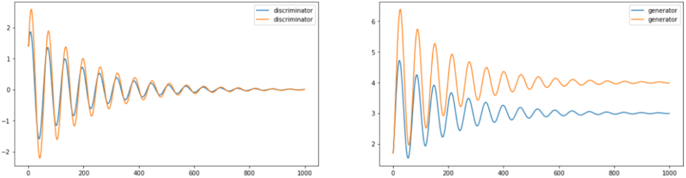
$$w_{t+1} = w_t + \eta \cdot (2\nabla_{w,t} - \nabla_{w,t-1}) = w_t + 2\eta \cdot \nabla_{w,t} - \eta \cdot \nabla_{w,t-1}$$

$$\theta_{t+1} = \theta_t - \eta \cdot (2\nabla_{\theta,t} - \nabla_{\theta,t-1}) = \theta_t - 2\eta \cdot \nabla_{\theta,t} + \eta \cdot \nabla_{\theta,t-1}$$

Gradient Descent (GD) vs. Optimistic Mirror Descent (OMD)



(a) GD dynamics.



(b) OMD dynamics.

OMD dynamics converge in terms of the last iterate.

Optimistic ADAM

ADAM (adaptive moment estimation) [Kingma, 2014] (6475 citations)

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Optimistic ADAM

ADAM:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

where $\hat{m}_t$ is first moment, $\hat{v}_t$ is second moment

---

Algorithm 1 *Optimistic ADAM*, proposed algorithm for training WGANs on images.

Parameters: stepsize η, exponential decay rates for moment estimates $\beta_1, \beta_2 \in [0, 1)$, stochastic loss as a function of weights $\ell_t(\theta)$, initial parameters θ_0

for each iteration $t \in \{1, \ldots, T\}$ **do**

  Compute stochastic gradient: $\nabla_{\theta,t} = \nabla_\theta \ell_t(\theta)$

  Update biased estimate of first moment: $m_t = \beta_1 m_{t-1} + (1 - \beta_1) \cdot \nabla_{\theta,t}$

  Update biased estimate of second moment: $v_t = \beta_2 v_{t-1} + (1 - \beta_2) \cdot \nabla^2_{\theta,t}$

  Compute bias corrected first moment: $\hat{m}_t = m_t / (1 - \beta_1^t)$

  Compute bias corrected second moment: $\hat{v}_t = v_t / (1 - \beta_2^t)$

  Perform *optimistic gradient step*: $\theta_t = \theta_{t-1} - 2\eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \eta \frac{\hat{m}_{t-1}}{\sqrt{\hat{v}_{t-1}} + \epsilon}$

Return θ_T

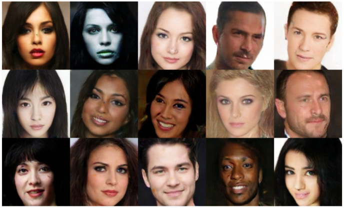
Optimistic ADAM

Overview

1. What is "adversarial"?
2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN
3. Paper review: Daskalakis, Training GANs with optimism, 2017
4. Boundary equilibrium GAN (BEGAN)
5. Adversarial examples

Boundary equilibrium GAN (BEGAN)

Samples



Interpolations of real images

Overview

1 What is "adversarial"?

2 Generative adversarial networks (GANs)
- Vanilla GAN
- WGAN

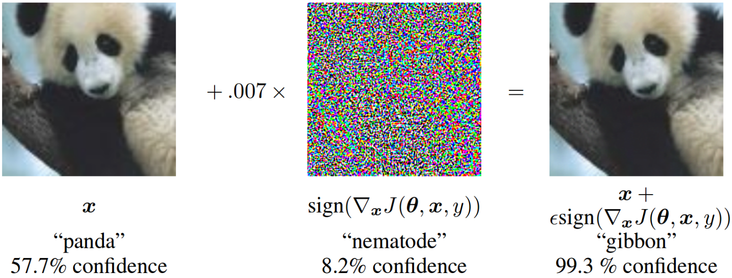
3 Paper review: Daskalakis, Training GANs with optimism, 2017

4 Boundary equilibrium GAN (BEGAN)

5 Adversarial examples

Adversarial Examples

- Examples that are similar to examples in the true distribution, but that fool a classifier [Szegedy, 2013]
- A demonstration of adversarial example [Goodfellow, 2014a]



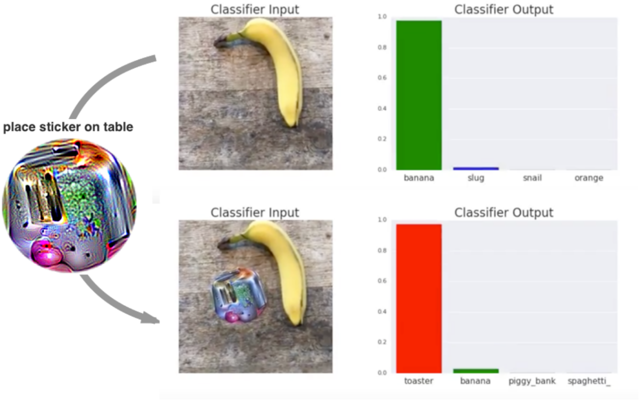
Paper review: Ilyas, The Robust Manifold Defense: Adversarial Training using Generative Models, 2017.

Adversarial Examples

Art?

Adversarial Examples

- Why small changes?
- Universal, robust, targeted adversarial image patches in the real world [Brown, 2017]
- https://youtu.be/i1sp4X57TL4



References I

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013).
Intriguing properties of neural networks.
arXiv preprint arXiv:1312.6199.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014).
Explaining and harnessing adversarial examples.
arXiv preprint arXiv:1412.6572.

Brown, T. B., Man, D., Roy, A., Abadi, M., & Gilmer, J. (2017).
Adversarial patch.
arXiv preprint arXiv:1712.09665.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014).
Generative adversarial nets.
Advances in neural information processing systems, 2672–2680.

References II

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016).
Improved techniques for training gans.
Advances in Neural Information Processing Systems, 2234–2242.

Arjovsky, M., & Bottou, L. (2017).
Towards principled methods for training generative adversarial networks.
arXiv preprint arXiv:1701.04862.

Ioffe, S., & Szegedy, C. (2015).
Batch normalization: Accelerating deep network training by reducing internal covariate shift.
International conference on machine learning, 448–456.

Denton, E. L., Chintala, S., & Fergus, R. (2015).
Deep generative image models using a laplacian pyramid of adversarial networks.
Advances in neural information processing systems, 1486–1494.

References III

Radford, A., Metz, L., & Chintala, S. (2015).
Unsupervised representation learning with deep convolutional generative adversarial networks.
arXiv preprint arXiv:1511.06434.

Mirza, M., & Osindero, S. (2014).
Conditional generative adversarial nets.
arXiv preprint arXiv:1411.1784.

Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. (2016).
Adversarially learned inference.
arXiv preprint arXiv:1606.00704.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015).
Adversarial autoencoders.
arXiv preprint arXiv:1511.05644.

References IV

Mescheder, L., Nowozin, S., & Geiger, A. (2017).
Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks.
arXiv preprint arXiv:1701.04722.

Zhao, J., Mathieu, M., & LeCun, Y. (2016).
Energy-based generative adversarial network.
arXiv preprint arXiv:1609.03126.

Arjovsky, M., Chintala, S., & Bottou, L. (2017).
Wasserstein gan.
arXiv preprint arXiv:1701.07875.

Berthelot, D., Schumm, T., & Metz, L. (2017).
Began: Boundary equilibrium generative adversarial networks.
arXiv preprint arXiv:1703.10717.

Saatchi, Y., & Wilson, A. G. (2017).
Bayesian GAN.
Advances in Neural Information Processing Systems, 3625–3634.

References V

avec les Memoires de Mathematique et de Physique. 1781.
Memoire sur la theorie des deblais et des remblais.
Histoire de l'Academie Royale des Science, Annee 1781.

Villani, C. (2008).
Optimal transport: old and new.
Springer Science & Business Media.

Rakhlin, S., & Sridharan, K. (2013).
Optimization, learning, and games with predictable sequences.
Advances in Neural Information Processing Systems, 3066–3074.

Kingma, D. P., & Ba, J. (2014).
Adam: A method for stochastic optimization.
arXiv preprint arXiv:1412.6980.

Thank you!