

Transporte Ótimo para Redes Neurais

Autor: Davi Sales Barreira

1. Introdução

2. Teoria de Transporte Ótimo

- 2.1 Monge & Kantorovich
- 2.2 Distância de Wasserstein
- 2.3 Variações da Distância de Wasserstein

3. OT & Redes Neurais

- 3.1 Introduzindo Generative Adversarial Networks
- 3.2 Problemas em GANs
- 3.3 Wasserstein GAN
- 3.4 Variações das Wasserstein GAN
- 3.5 Outras Aplicações

Transporte Ótimo (OT) é uma área da matemática que estuda o problema de transportar “massa” em uma configuração para outra enquanto se miniza o custo de transporte.

Apesar de parecer um problema bastante específico, a ideia de se transportar objetos de maneira ótima é bastante ubíqua e possui diversas utilidades.

OT tem aparecido em diversas aplicações recentes de Machine Learning, como: **transfer learning, clustering, redução de dimensionalidade, modelos generativos**, entre outros.

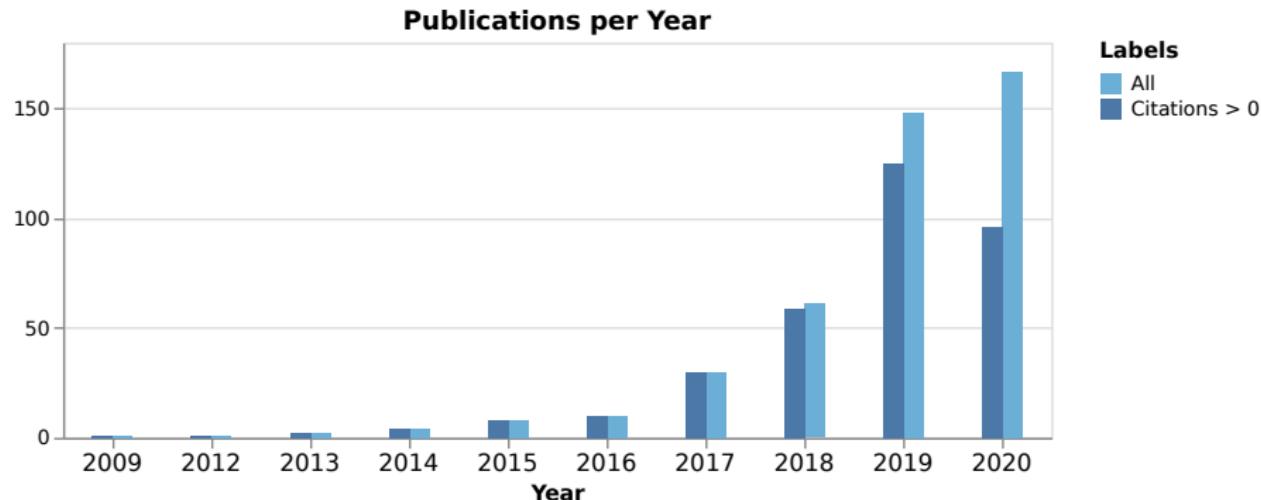


Figure 1: Gráficos com a evolução do número de publicações relacionadas a Transporte Ótimo com Machine Learning [10].

A solução de um problema de Transporte Ótimo sempre resulta em dois subprodutos, o **plano (mapa)** ótimo de transporte e o **custo mínimo** para realizar o transporte.

A maioria das aplicações em ML utiliza o custo mínimo para definir uma métrica de distância (e.g. Wasserstein). Porém, existem aplicações como Transfer Learning que utilizam os mapas ótimos [10].

Nesta apresentação vamos focar na aplicação mais celebrada de OT em redes neurais, as chamadas **Wasserstein Generative Neural Networks** [1].

Problema de Monge - Qual a maneira ótima de transportar massa de uma configuração para outra?

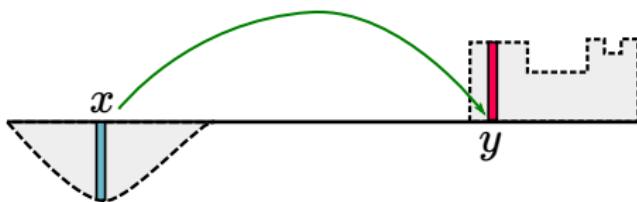


Figure 2: Massa não pode ser separada.

Kantorovich Problem - Relaxação do problema original de Monge.

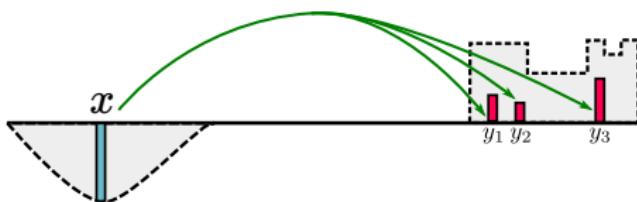


Figure 3: Massa pode ser separada.

Definition (Problema de Monge)

Dadas duas medidas de probabilidade $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ e uma função de custo $c : X \times Y \rightarrow [0, +\infty]$, resolva:

$$(MP) \quad \inf \left\{ \int_X c(x, T(x)) d\mu \quad : \quad T_{\#}\mu = \nu \right\} \quad (1)$$

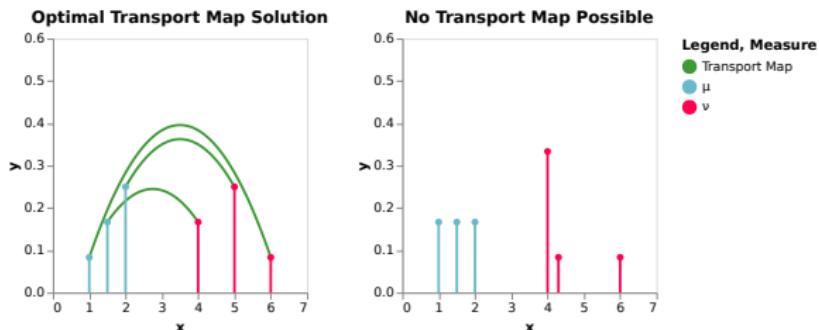


Figure 4: Exemplo de dois problemas de Transporte Ótimo.

Definition (Acoplamento (*Coupling*))

Sejam (X, μ) e (Y, ν) espaços de probabilidade. Para $\gamma \in \mathcal{P}(X \times Y)$, dizemos que γ é um acoplamento de (μ, ν) se $(\pi_X)_\# \gamma = \mu$ e $(\pi_Y)_\# \gamma = \nu$. Chamamos $\Pi(\mu, \nu)$ do conjunto de **Planos de Transporte**:

$$\Pi(\mu, \nu) := \{\gamma \in \mathcal{P}(X \times Y) : (\pi_X)_\# \gamma = \mu \text{ and } (\pi_Y)_\# \gamma = \nu\} \quad (2)$$

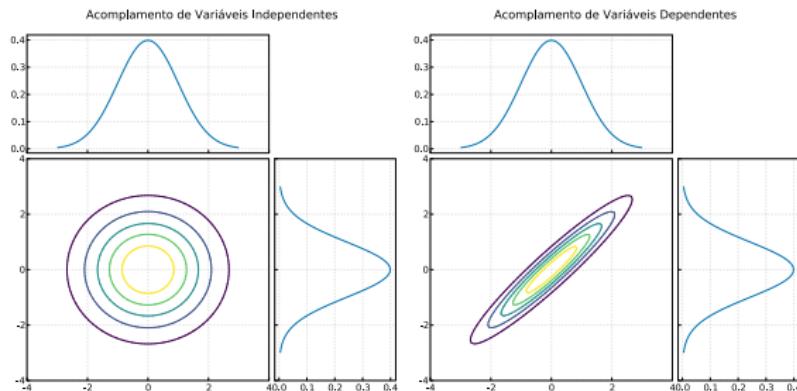


Figure 5: Exemplos de acoplamento.

Definition (Problema de Kantorovich)

Dadas duas medidas de probabilidade $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ e a função de custo $c : X \times Y \rightarrow [0, +\infty]$, resolva:

$$(KP) \quad \inf \left\{ \int_{X \times Y} c(x, y) d\gamma : \gamma \in \Pi(\mu, \nu) \right\} \quad (3)$$

O Problema de Kantorovich tem uma formulação dual, que para certas condições de regularidade possui a mesma solução ótima que o problema primal (dualidade forte).

Definition (Problema Dual)

Dadas $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ e custo $c : X \times Y \rightarrow \mathbb{R}_+$. O Problema Dual é

$$(DP) \quad \sup \left\{ \int_X \phi \, d\mu + \int_Y \psi \, d\nu \mid \phi \in C_b(X), \psi \in C_b(Y), \phi \oplus \psi \leq c \right\} \quad (4)$$

Funções ϕ, ψ são chamadas de **Potenciais de Kantorovich**. Essa formulação é utilizada nas Wasserstein GANs.

Definition (Distância de Wasserstein)

Seja (X, d) um espaço métrico polonês, com $c : X \times X \rightarrow \mathbb{R}$ tal que $c(x, y) = d(x, y)^p$, e $p \in [1, +\infty)$. Para $\mu, \nu \in \mathcal{P}_p(X)$, a distância de Wasserstein é dada por:

$$W_p(\mu, \nu) := \left(\inf_{\gamma \in \Pi(\mu, \nu)} \int_{X \times X} d(x, y)^p \, d\gamma \right)^{1/p} \quad (5)$$

$\mathcal{P}_p(X)$ é o espaço de medidas de probabilidade com p -ésimo momento.

A distância de Wasserstein preserva a geometria do espaço no qual está definida a medida.

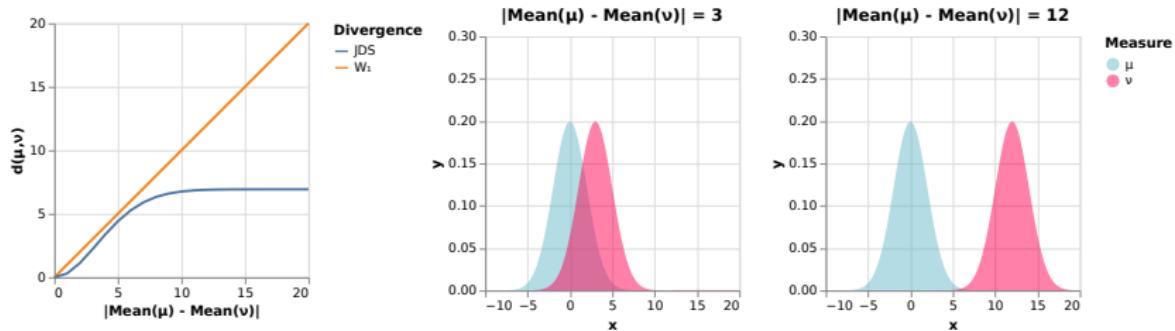


Figure 6: Comparação entre a distância de Wasserstein e Jensen-Shannon.

Dadas duas medidas de probabilidade μ e ν , como então computamos a distância de Wasserstein entre elas?

Sabemos resolver para alguns casos!

1. Distribuições em 1D (continuas e discretas);
2. Distribuições Gaussianas em n dimensões;
3. Distribuições discretas finitas em n dimensões.

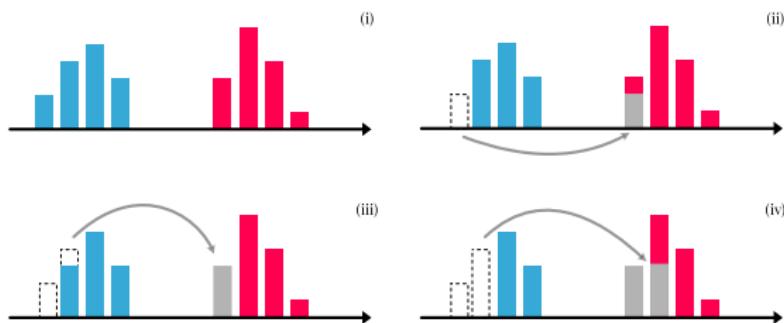


Figure 7: Algoritmo para caso 1D com medida finita discreta.

Calcular a distância de Wasserstein em espaços de alta dimensão (e.g. imagens) é bastante custoso, pois precisamos resolver um problema de Programação Linear.

Assim, variações da distância de Wasserstein foram desenvolvidas. A variação mais famosa é a chamada distância de **Wasserstein Entrópica**.

$$\overline{OT}_{c,\varepsilon}(\mu, \nu) := \inf_{\gamma \in \Pi(\mu, \nu)} \int_{X \times Y} c(x, y) d\gamma + \varepsilon \text{KL}(\gamma | \mu \otimes \nu)$$

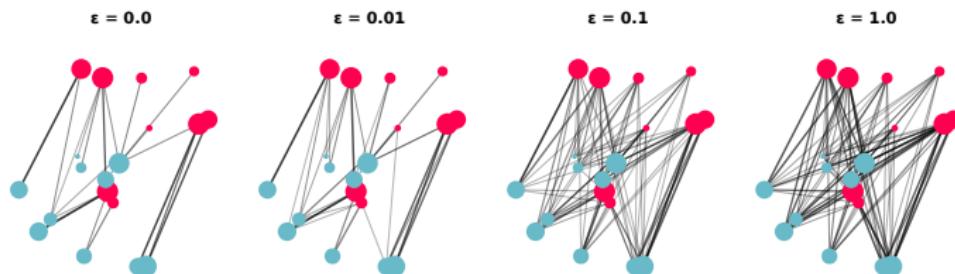


Figure 8: Exemplo de solução de OT com regularização entrópica [10].

Outra variação também bastante utilizada em Machine Learning é a chamada **Sliced Wasserstein**.

$$SW_p(\mu, \nu) = \left(\int_{\mathbb{S}^{d-1}} [W_p(P_\theta \# \mu, P_\theta \# \nu)]^p d\theta \right)^{\frac{1}{p}}$$

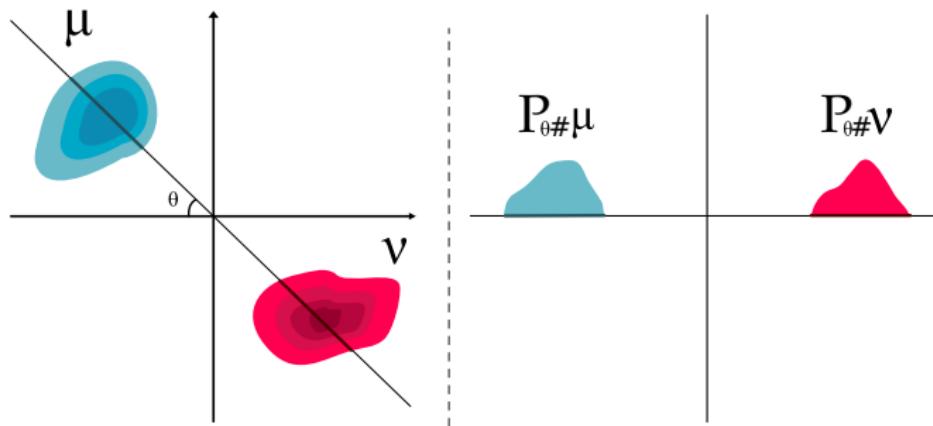


Figure 9: Desenho esquemático da Sliced Wasserstein [10].

Generative Adversarial Networks (GAN) foram originalmente introduzidas por Goodfellow et al. [6]. Essas redes são utilizadas com o objetivo de gerar dados sintéticos realísticos a partir de dados reais.



Figure 10: Faces geradas por GANs [8].

A ideia geral por trás das GANs é utilizar duas redes neurais competindo uma com a outra, sendo uma rede responsável por gerar amostras parecidas com os dados reais (*gerador*) , enquanto a outra busca identificar quando o dado é real ou sintético (*descriminador*).

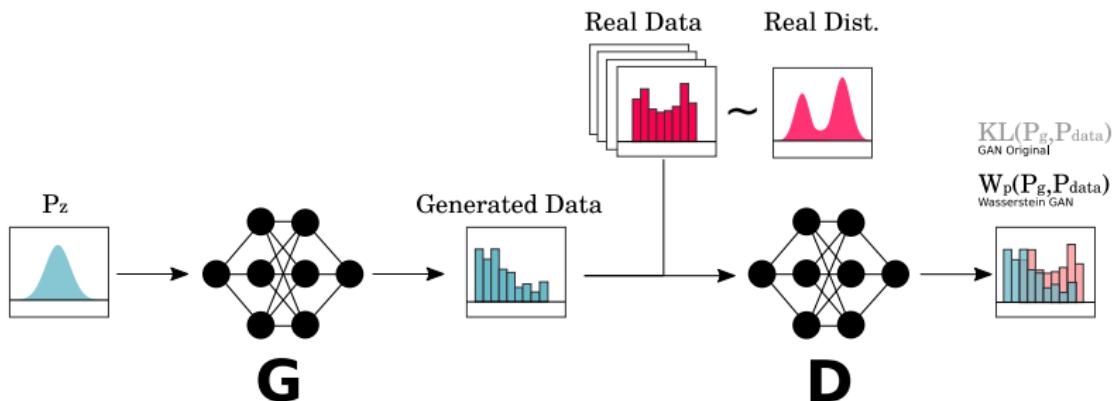


Figure 11: Generative Adversarial Network [10].

A GAN original busca resolver o seguinte problema de otimização mini-max:

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

Treinamos o discriminador D buscando maximizar a capacidade de discernir dados de P_{data} de P_g . Em um segundo momento, treinamos G buscando minimizar a distância entre P_g e P_{data} .

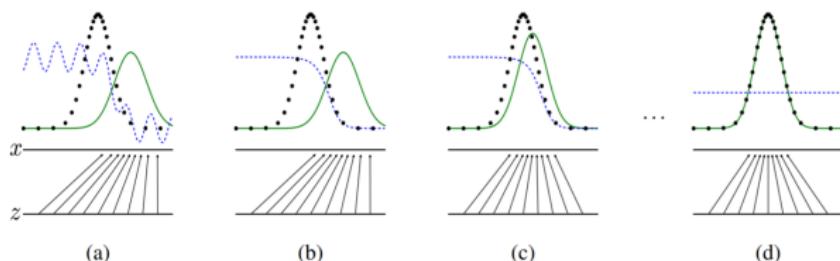


Figure 12: De (a) até (d), o desenho ilustra a evolução do algoritmo ao ser treinado. A linha azul representa a distribuição do discriminador, a linha verde representa a P_g , e os pontos pretos representam P_{data} [6].

GANs tradicionais possuem falhas comuns que tem sido foco de pesquisas.

- **Dissipação de gradientes:** Na prática, ao treinar o desclassificador até seu estado ótimo ($\frac{p_{data}}{p_{data} + p_g}$), os gradientes tendem a se dissipar no treinamento do gerador. Além disso, quando as distribuições não compartilham o suporte, o gradiente se torna nulo.
- **“Mode Collapse”:** Normalmente, você deseja que sua GAN produza uma ampla variedade de saídas. No entanto, se um gerador produz uma saída especialmente plausível, ele pode aprender a produzir apenas essa saída.
- **Falha de Convergência:** GANs podem apresentar instabilidade na atualização dos gradientes do gerador, levando a falhas de convergência.

Arjovsky et al. [1] propuseram a Wasserstein GAN, onde se usa $W_1(P_g, P_{data})$ invés de $KL(P_g, P_{data})$ (ou variações).

Usando formulação dual da distância de Wasserstein, o problema de otimização da WGAN se torna:

$$\min_G W_1(P_{\text{data}}, P_g) = \min_G \max_{D \in Lip_1} \mathbb{E}_{P_{\text{data}}}[D(x)] - \mathbb{E}_{P_g}[D(x)]$$

Mas como se computa o gradiente de $W_1(\cdot, \cdot)$?

Arjovsky et al. [1] mostrou que o gradiente pode então ser calculado por

$$\nabla_{\theta_g} W_1(P_{\text{data}}, P_g) = -\mathbb{E}_{P_{\text{data}}} [\nabla_{\theta_g} D(G(\mathbf{z}))].$$

onde θ_g são os parâmetros do gerador.

Para forçar que $D \in Lip_1$, os autores propuseram *clippar* os parâmetros de D entre $[-a, a]$, $a \in \mathbb{R}_+$.

WGANS não são perfeitas!

1. Dificuldade de treinar;
2. Hiperparâmetro de clipping para tunar, além de tornar D esparsa e instável;

Várias modificações sugeridas na literatura, por exemplo:

1. Gulrajani et al. [7] - Trocar clipping por penalização no gradiente;
2. Genevay et al. [5] - Utilizar Entropic Wasserstein;
3. Deshpande et al. [3] - Utilizar Sliced-Wasserstein.

Além de GANs, Transporte Ótimo tem sido usado em diversas outras aplicações.

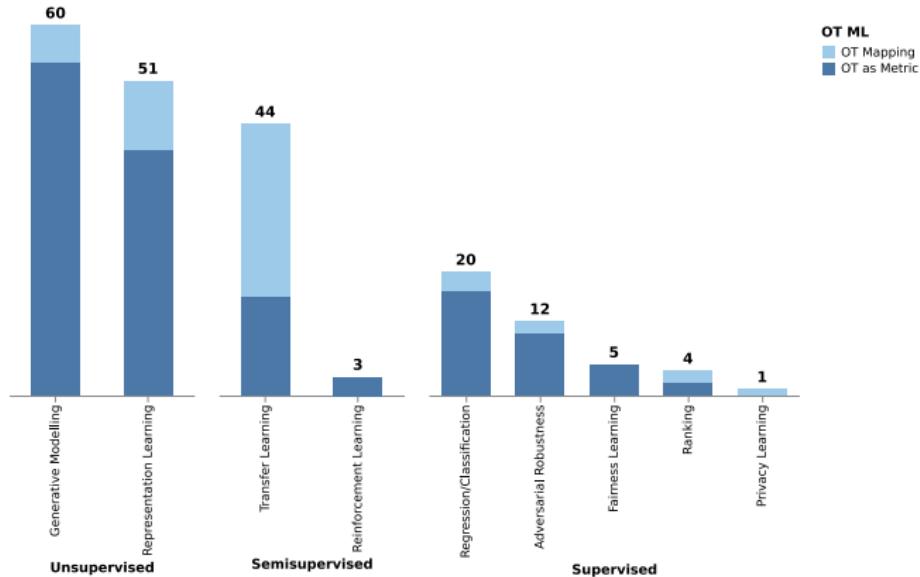


Figure 13: Distribuição por categoria dos 200 papers mais citados de OT com ML[10].

Modelo DeepJDOT de Transfer Learning em Redes Neurais [2].

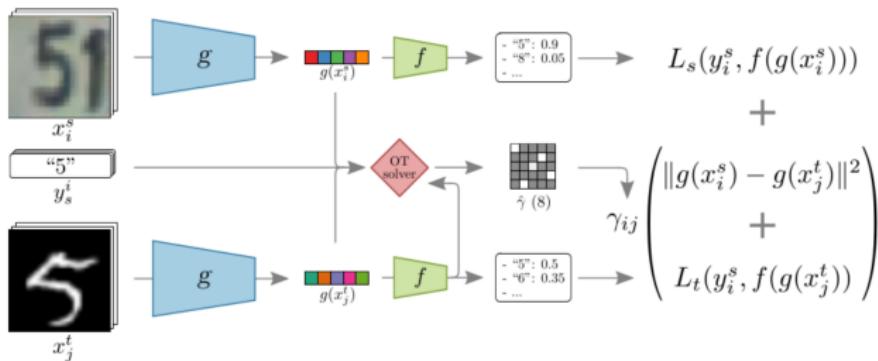


Figure 14: Imagem de Damodaran et al. [2].

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- [2] Bharath Bhushan Damodaran, Benjamin Kellenberger, R émi Flamary, Devis Tuia, and Nicolas Courty. Deepdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 447–463, 2018.
- [3] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3483–3491, 2018.
- [4] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss. *arXiv preprint arXiv:1506.05439*, 2015.
- [5] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.
- [6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>.
- [7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.
- [8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018.
- [9] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- [10] Davi Sales Barreira. *Optimal Transport for Machine Learning: Theory and Applications*. PhD thesis, 2021.
- [11] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkhäuser, NY*, 55(58-63):94, 2015.