universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„The Wasserstein Distance and its Application to Generative Adversarial Networks“

verfasst von / submitted by

Alina Franziska Leuchtenberger, B.Sc.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 821
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Mathematics
Betreut von / Supervisor:	Assoz. Prof. Dr. Philipp Grohs

Contents

Contents i

1 Introduction 1

2 The Wasserstein Distance 3
2.1 Optimal Transport 3
2.2 The Wasserstein Distance and its Properties 8
2.3 Learning by the Earth-Mover Distance 19

3 Application of the EM to Generative Adversarial Networks 26
3.1 Generative Adversarial Networks 26
3.1.1 Structure of a GAN 26
3.1.2 Training of a GAN 28
3.1.3 Equilibrium of a trained GAN 31
3.1.4 The Discriminator's Cost 32
3.1.5 The Generator's Cost 34
3.2 Wasserstein Generative Adversarial Nets 39
3.2.1 Motivation for Using the Earth-Mover Distance 39
3.2.2 Approximation of Earth-Mover Distance in a WGAN . 42
3.2.3 Training of a WGAN 43
3.2.4 Performance of a WGAN 44

4 Application of Conditional Wasserstein GAN 46
4.1 Conditional Wasserstein GAN 46
4.2 Example of Conditional Wasserstein GAN 47

A Abstract & Zusammenfassung 50

Bibliography 52

Chapter 1

Introduction

In the last years neural networks had a great revival. Today they are a popular topic in computer science, but also in mathematics. When they appeared in 1940-1960 and later in 1980-1990 neither the computational capacity of the hardware nor the amount of data was sufficient to produce adequate results. [5, pp. 13,18,19] Nowadays these problems were overcome and deep learning networks are widely used in data analysis.

One task deep learning networks can be used for is the generation of samples drawn of a specific distribution. This is useful, for example, when designing several possible environments for reinforcement learning, when performing semi-supervised learning, working with multi-modal output or for quality enhancement of images [7, pp. 3-4].

In 2014 a new type of such a network was found by Ian Goodfellow [6]: The Generative Adversarial Network (GAN). The idea of the GAN was born in a bar, the first implementation done in the same night [20].

Compared to other generating networks GANs have several advantages. The design of their generating part is not very restricted, GANs do not need Markov chains, variational bounds and are asymptotically consistent. Moreover, they are able to perform parallel generation of samples. On the other hand an optimal trained GAN is more difficult to achieve than other generating networks. [7, p. 17]

An interesting variant of a GAN is the Wasserstein GAN which was introduced by Arjovsky et al. in 2016 [1]. It is motivated by the idea to use an approximation of the Wasserstein distance as cost function. This distance between two probability measures has a long history. A variant of it was described by Gini in 1914, it was rediscovered by Kantorovich, Wasserstein, Mallows and Tanaka. [21, p. 118] Thus, there exist a range of mathematical properties of the Wasserstein distance the Wasserstein GANs can take advantage of.

These properties are described in Chapter 2 after stating an overview about the general results of optimal transport. Chapter 3 introduces in its first section the Generative Adversarial Networks in general. In the second section of Chapter 3 the Wasserstein Generative Adversarial Networks are described as an application of the Wasserstein distance to GAN. In the last chapter an application of conditional Wasserstein GAN, a specific variant of Wasserstein GAN, is presented.

Notation A Polish space is defined as a complete, separable metric space with its Borel σ-algebra. The notation $P(\mathcal{X})$ describes the set of all Borel probability measures on a space $\mathcal{X}$.

$p_n \to p$ denotes the weak convergence of p_n to p, meaning for all bounded, continuous functions ϕ $\int \phi dp_n \to \int \phi dp$ holds. The space of all continuous functions $f : \mathcal{X} \to \mathbb{R}$ is denoted by $C(\mathcal{X})$. And for any Lipschitz continuous function f its Lipschitz constant is defined as $L = \|f\|_{Lip}$.

$L^k(\mathcal{X}, p)$ denotes the Lebesgue space of order k for the measure space $(\mathcal{X}, p)$ and is also noted as $L^k(p)$ if $\mathcal{X}$ is apparent from the context. The respective L^k-norm is $\|\cdot\|_{L^k} = \left(\int |f|^k dp\right)^{\frac{1}{k}}$ for every $f \in L^k(\mathcal{X}, p)$. In general $\|\cdot\|$ denotes a norm and $\langle \cdot, \cdot \rangle$ the associated scalar product.

$1_A(x)$ is the indicator function for a measurable set A and δ_x the Dirac measure at x. The identity mapping in every space is denoted by id.

Chapter 2

The Wasserstein Distance

The subject of this chapter is the Wasserstein distance and its diverse properties mainly citing the book 'Optimal Transport, old and new' of Villani. This distance describes the optimal transport cost between two probability distributions. Therefore, the first section considers the foundations of optimal transport and states additionally a few theorems and lemmata which are important for the subsequent. The second section states the definition of the Wasserstein distance and several properties of it. The last section considers learning with a special variant of Wasserstein distance.

2.1 Optimal Transport

To formulate the problem of optimal transport between two probability measures p and q the notation of coupling has to be introduced first.

Definition 2.1 (Couplings and transport plans) *[21, Definition 1.1, p. 22] Let $\mathcal{X}$ and $\mathcal{Y}$ be two measure spaces with respective probability distributions p and q. (X, Y) is a coupling of (p, q) if X is a random variable with distribution p and Y a random variable with distribution q. The distribution of (X, Y) can also be called a coupling of (p, q), but is mostly called transport or transference plan.*

The optimal transport π is then the transport plan which solves the Monge-Kantorovich problem:

Problem 2.2 (Monge-Kantorovich problem) *[21, p. 22] Consider a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$ and $p \in P(\mathcal{X})$, $q \in P(\mathcal{Y})$. Let $\Pi(p, q)$ denote the set of all transference plans between p and q. Then the problem*

$$\inf_{\pi \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\pi} c(x, y)$$

is called the Monge-Kantorovich problem. Its solution $C(p, q)$ is called total cost.

The Monge-Kantorovich problem is solved by the optimal way of transporting one distribution to another. One can interpret this as an economic problem. In this case the distributions are the supply on different locations and the demand on other locations. In this interpretation however, a transport plan between the distributions is a way to transport the products between those locations and the optimal way of doing this is the solution to the problem. Thus, the cost of transporting the products is the interpretation of the solution of the Monge-Kantorovich problem, the total cost. [21, p. 42]

The next theorem, Prokhorov's Theorem, connects tightness and precompactness of measure sets. This connection will be required later on.

Theorem 2.3 (Prokhorov's Theorem) *[2, Theorem 5.1, 5.2] Consider a Polish space $\mathcal{X}$ and $M \subset P(\mathcal{X})$. Then the following statements are equivalent:*

- *M is weakly precompact i.e. any sequence in M has a subsequence which is convergent for the weak topology.*
- *M is tight i.e. $\forall \varepsilon > 0$ it exist a compact set M_ε such that $p(\mathcal{X} \setminus M_\varepsilon) \leq \varepsilon$ $\forall p \in M$.*

The next two lemmata state important properties of transference plans.

Lemma 2.4 (Upper bound of transport cost π) *[21, Lemma 4.3] Consider the Polish spaces $\mathcal{X}$ and $\mathcal{Y}$. Moreover, let $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$ be a lower semicontinuous cost function and $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \cup \{-\infty\}$ an upper semicontinuous cost function with $c \geq h$. If $(\pi_n)_{n\in\mathbb{N}}$, $\pi \in P(\mathcal{X} \times \mathcal{Y})$ and $h \in L^1(\pi_n) \cap L^1(\pi)$ are such that*

$$\int_{\mathcal{X}\times\mathcal{Y}} h d\pi_n \to \int_{\mathcal{X}\times\mathcal{Y}} h d\pi \quad for\ n \to \infty$$

then

$$\int_{\mathcal{X}\times\mathcal{Y}} c d\pi \leq \liminf_{n\to\infty} \int_{\mathcal{X}\times\mathcal{Y}} c d\pi_n.$$

Lemma 2.5 (Transference plans are tight) *[21, Lemma 4.4] Consider the Polish spaces $\mathcal{X}$ and $\mathcal{Y}$. If $\mathcal{P}$ is tight subset of $P(\mathcal{X})$ and $\mathcal{Q}$ is a tight subset of $P(\mathcal{Y})$, then the set of all transference plans between $\mathcal{P}$ and $\mathcal{Q}$, $\Pi(\mathcal{P}, \mathcal{Q})$, is tight in $P(\mathcal{P} \times \mathcal{Q})$.*

The last two Lemmata 2.5 and 2.4 together with Prokhorov's Theorem 2.3 are applied to prove that an optimal coupling exists.

Theorem 2.6 (Existence of optimal coupling) *[21, Theorem 4.1] Consider the Polish spaces $\mathcal{X}$ and $\mathcal{Y}$ with $p \in P(\mathcal{X})$ and $q \in P(\mathcal{Y})$.*

Moreover, let $a : \mathcal{X} \to \mathbb{R} \cup \{-\infty\}$ with $a \in L^1(p)$ and $b : \mathcal{Y} \to \mathbb{R} \cup \{-\infty\}$ with $b \in L^1(q)$ be upper semicontinuous. Furthermore, consider a lower semicontinuous cost function $c : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ with $c(x, y) \geq a(x) + b(y)$ $\forall x, y$. Then there is a coupling of (p, q) such that $\mathbb{E}c(X, Y)$ is minimal among all couplings (X, Y).

Proof [21, Proof of Theorem 4.1] Every probability measure on a Polish space is tight. Thus, since $\mathcal{X}$ and $\mathcal{Y}$ are Polish spaces, p and q are tight in $P(\mathcal{X})$ and $P(\mathcal{Y})$, respectively. From Lemma 2.5 it follows that also the set of transportation plans between p and q is tight in $P(\mathcal{X} \times \mathcal{Y})$. Using Prokhorov's Theorem 2.3 it is clear that $\Pi(p,q)$ is weakly precompact. The map from $\Pi(p,q)$ to its marginals p and q is continuous. Thus, $\Pi(p,q)$ is closed and not only precompact for the weak topology, but also compact.

Consider the probability measures $(\pi_n)_{n\in\mathbb{N}}$ in $\Pi(p,q)$ such that their cost $\int c d\pi_n$ converges to the infimum of the transport cost. Since $\Pi(p,q)$ is compact, at least a subsequence of $(\pi_n)_{n\in\mathbb{N}}$ converges to some $\pi \in \Pi(p,q)$ and without loss of generality it is here assumed that $(\pi_n)_{n\in\mathbb{N}}$ is such that it converges to π.

Furthermore, define the function $h(x,y) \coloneqq a(x) + b(y)$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Then it holds that $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \cup \{-\infty\}$ is as sum of upper semicontinuous functions likewise upper semicontinuous and $c \geq h$. Additionally, $h \in L^1(\pi_n) \cap L^1(\pi)$ and

$$\int_{\mathcal{X}\times\mathcal{Y}} h d\pi_n = \int_{\mathcal{X}} a(x) dp + \int_{\mathcal{Y}} b(y) dq = \int_{\mathcal{X}\times\mathcal{Y}} h d\pi.$$

Consequently, the conditions of Lemma 2.4 are satisfied and thus

$$\int_{\mathcal{X}\times\mathcal{Y}} c d\pi \leq \liminf_{n\to\infty} \int_{\mathcal{X}\times\mathcal{Y}} c d\pi_n$$

holds. Since by assumption $\int c d\pi_n$ converges to the infimum of the transport cost, π is the optimal coupling between the probability measures p and q. □

Note, that the last theorem only proves that an optimal coupling as coupling with the smallest total cost exists. The optimal transport cost is not necessarily finite in this case.

The dual formulation of the Monge-Kantorovich problem is called the dual Kantorovich problem.

Problem 2.7 (Dual Kantorovich problem) *[21, p. 65] Let $\mathcal{X}$ and $\mathcal{Y}$ be Polish spaces and $p \in P(\mathcal{X})$, $q \in P(\mathcal{Y})$. Consider $\phi \in L^1(\mathcal{Y}, q)$ and $\psi \in L^1(\mathcal{X}, p)$. Then*

$$\sup_{\phi(y)-\psi(x)\leq c(x,y)} \int_{\mathcal{Y}} \phi(y) dq(y) - \int_{\mathcal{X}} \psi(x) dp(x)$$

is called the dual Kantorovich problem.

This problem can as well be interpreted in terms of the product transport between supply- and demand-locations. In this example ψ is the price at

which the product must be bought at the supply-locations and ϕ the price at which the product can be sold at the demand-locations. Thus, the solution of the dual problem is the maximal money a company could make by transporting the products between the locations. Obviously, the solution of the dual problem is bound above by the solution of the Monge-Kantorovich problem. The suppliers will transport the products themselves if that is cheaper than the transport by the companies. [21, p. 65]

Moreover, under certain assumptions the solution of the primal and dual problem is exactly the same. These assumptions include the terms of c-convex and c-concave functions which are defined below:

Definition 2.8 (c-convex, c-concave functions) *[21, Definition 5.2, 5.7] Consider the cost function $c : \mathcal{X} \times \mathcal{Y} \to (-\infty, \infty]$ on two sets $\mathcal{X}$ and $\mathcal{Y}$. Moreover, let $\psi : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$. If ψ is not ∞ everywhere and a function $\zeta : \mathcal{Y} \to \mathbb{R} \cup \{\pm\infty\}$ exists such that*

$$\psi(x) = \sup_{y \in \mathcal{Y}} (\zeta(y) - c(x,y)) \qquad \forall x \in \mathcal{X},$$

then ψ is called c-convex. The c-transform of ψ is ψ^c:

$$\psi^c(y) = \inf_{x \in \mathcal{X}} (\psi(x) + c(x,y)) \qquad \forall y \in \mathcal{Y}.$$

The function $\phi : \mathcal{Y} \to \mathbb{R} \cup \{-\infty\}$ is called c-concave if ϕ is not $-\infty$ everywhere and a function $\psi : \mathcal{X} \to \mathbb{R} \cup \{\pm\infty\}$ exists with $\phi = \psi^c$.

Thus, the Kantorovich duality which states the equality of the primal and dual solution of the problem can be described now.

Theorem 2.9 (Kantorovich duality) *[21, Theorem 5.10 (i,iii)] Consider $(\mathcal{X}, p)$ and $(\mathcal{Y}, q)$ as Polish probability spaces and $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$ as lower semicontinuous cost function which satisfies*

$$c(x,y) \geq a(x) + b(y) \qquad \forall (x,y) \in \mathcal{X} \times \mathcal{Y}$$

for the functions $a \in L^1(p)$ and $b \in L^1(q)$ which are real-valued upper semicontinuous. Then it holds that

$$\begin{aligned} \min_{\pi \in \Pi(p,q)} \int_{\mathcal{X} \times \mathcal{Y}} c(x,y) d\pi(x,y) &= \sup_{\psi \in L^1(p)} \left(\int_{\mathcal{Y}} \psi^c(y) dq(y) - \int_{\mathcal{X}} \psi(x) dp(x) \right) \\ &= \sup_{\phi \in L^1(q)} \left(\int_{\mathcal{Y}} \phi(y) dq(y) - \int_{\mathcal{X}} \phi^c(x) dp(x) \right) \end{aligned}$$

where the suprema are taken over ψ c-convex and ϕ c-concave.

If additionally the cost c is real-valued, the total cost $C(p,q) < \infty$ and the cost is pointwise upper-bounded

$$c(x,y) \leq g(x) + h(y) \quad (g,h) \in L^1(p) \times L^1(q)$$

then it holds that

$$\begin{aligned}
&\min_{\pi \in \Pi(p,q)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\pi(x,y) \\
&= \max_{(\psi,\phi)\in L^1(p)\times L^1(q)} \left(\int_{\mathcal{Y}} \phi(y) dq(y) - \int_{\mathcal{X}} \psi(x) dp(x) \right) \\
&= \max_{\psi \in L^1(p)} \left(\int_{\mathcal{Y}} \psi^c(y) dq(y) - \int_{\mathcal{X}} \psi(x) dp(x) \right)
\end{aligned}$$

where the maxima are taken over ψ c-convex and $\phi = \psi^c$.

Therefore, under additional assumptions the supremum of the dual problem is attained. This property will be crucial in the third section of this chapter.

The definition of c-cyclically monotone transference plans comes from the intuition that one can attempt to improve a transference plan by redirecting the transport from one target location to the next one. If the total transport cost of the new transference plan is smaller than the original one, then the attempt was of course successful. If that is not the case, the original transference plan could be a candidate for the optimal transference plan. [21, pp. 63-64]

Definition 2.10 (c-cyclically monotone sets) *[21, Definition 5.1] Consider the function $c : \mathcal{X} \times \mathcal{Y} \to (-\infty, \infty]$. The subset $\Gamma \subset \mathcal{X} \times \mathcal{Y}$ is called c-cyclically monotone if*

$$\sum_{i=1}^{n} c(x_i, y_i) \leq \sum_{i=1}^{n} c(x_i, y_{i+1})$$

holds for any $n \in \mathbb{N}$ and $(x_i, y_i)_{i=1}^n$ in Γ, where $y_{n+1} := y_1$.

A transference plan which is concentrated on such a Γ is also called c-cyclically monotone.

The next theorem states that a weak convergence of probability measures implies the convergence of their optimal transport plans to a c-cyclically monotone transport plan. Moreover, under certain assumptions the resulting transport plan is optimal.

Theorem 2.11 (Optimal transport is stable) *[21, Theorem 5.20] Consider $\mathcal{X}$, $\mathcal{Y}$ as two Polish spaces. Let $(c_n)_{n\in\mathbb{N}}$ be a sequence of continuous cost functions on $\mathcal{X}\times\mathcal{Y}$ which converges uniformly to a real-valued continuous cost function $c \in C(\mathcal{X}\times\mathcal{Y})$ such that $\inf c > -\infty$. Moreover, let $(p_n)_{n\in\mathbb{N}}$ be a sequence of probability measures on $\mathcal{X}$ converging weakly to p and $(q_n)_{n\in\mathbb{N}}$ a sequence of probability measures on $\mathcal{Y}$ converging weakly to q. If π_n denotes the optimal transport between p_n and q_n for each $n \in \mathbb{N}$ and*

$$\int_{\mathcal{X}\times\mathcal{Y}} c_n(x,y)d\pi_n(x,y) < \infty \qquad \forall n \in \mathbb{N},$$

then π_n (or a subsequence) converges weakly to a c-cyclically monotone transport plan $\pi \in \Pi(p,q)$.

In case of

$$\liminf_{n\in\mathbb{N}} \int_{\mathcal{X}\times\mathcal{Y}} c_n(x,y)d\pi_n(x,y) < \infty$$

the total optimal transport cost $C(p,q) < \infty$ and π is an optimal transport.

The results of this section can be used to find several properties of the Wasserstein distance in the next section.

2.2 The Wasserstein Distance and its Properties

The Wasserstein distance between two probability measures is $\frac{1}{k}$ of the optimal total cost as solution of the Monge-Kantorovich problem with cost function d^k. [21, p. 105]

Definition 2.12 (Wasserstein distance) *[21, Definition 6.1] Consider a Polish metric space $(\mathcal{X},d)$, $k \in [1,\infty)$ and p, q as probability measures on $\mathcal{X}$. Then*

$$\begin{aligned} W_k(p,q) &:= \inf_{\pi\in\Pi(p,q)} \left(\int_{\mathcal{X}\times\mathcal{X}} d(x,y)^k d\pi(x,y)\right)^{\frac{1}{k}} \\ &= \inf_{\pi\in\Pi(p,q)} \left(\mathbb{E}_{(x,y)\sim\pi} d(x,y)^k\right)^{\frac{1}{k}} \end{aligned} \tag{2.1}$$

is called the Wasserstein distance of order k between p and q.

Therefore, the Wasserstein distance considers the optimal cost and not explicitly the optimal transport plan. The distance is also called Kantorovich-Rubenstein distance, Earth-Mover distance or Optimal Transport. While in some literature those names are used for all kinds of Wasserstein distances, others use several names only for specific Wasserstein distances. In this thesis for the general case only the term Wasserstein distance is used. The 1-Wasserstein distance with a metric induced by a norm will be called Earth-Mover distance in this thesis. This is done to distinguish between the general

Wasserstein distance and this specific variant which will play a big role in the second part of this thesis.

In general the Wasserstein distance is not necessarily finite. However, in most cases the Wasserstein distance is considered on the Wasserstein space P_k on which it is always finite.

Definition 2.13 (Wasserstein space) *[21, Definition 6.4] Consider the setting of Definition 2.12. Then*

$$P_k := \left\{ p \in P(\mathcal{X}) | \int_{\mathcal{X}} d(x_0, x)^k dp(x) < \infty \right\}$$

for any $x_0 \in \mathcal{X}$ is called Wasserstein space of order k.

Note, that the Wasserstein space depends not on the choice of x_0. [21, Definition 6.4]

With this definition it can be shown that the Wasserstein distance satisfies all conditions of a finite distance.

Theorem 2.14 (W_k is a distance) *[21, pp. 106/107] Consider a Polish metric space $(\mathcal{X}, d)$ and $k \in [1, \infty)$. The Wasserstein distance of order k is a finite distance on $P_k(\mathcal{X})$.*

Proof [21, pp. 106/107] In order to prove that the Wasserstein distance is a finite metric it has to be proven that it is

i. positive definite

ii. symmetric

iii. satisfies the triangle inequality

iv. finite on $P_k(\mathcal{X})$

These properties hold indeed:

i. Since the Wasserstein distance is the infimum of an expected value of a norm, the distance can never be negative. So it remains to show

$$W_k(p, q) = 0 \Leftrightarrow p = q.$$

Since $d(x, y)$ is a metric, it is itself positive definite and thus

$$\begin{aligned} W_k(p, q) = 0 &\Leftrightarrow \exists \pi \in \Pi(p, q) : \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^k d\pi(x, y) = 0 \\ &\Leftrightarrow \exists \pi \in \Pi(p, q) : d(x, y) = 0 \quad if \ \pi(x, y) \neq 0 \\ &\Leftrightarrow id \in \Pi(p, q) \\ &\Leftrightarrow p = q \end{aligned}$$

ii. Using the symmetry of $d(x,y)$ the symmetry of the Wasserstein distance can easily be shown:

$$\begin{aligned} W_k(p,q) &= \left(\inf_{\pi \in \Pi(p,q)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^k d\pi(x,y) \right)^{\frac{1}{k}} \\ &= \left(\inf_{\pi \in \Pi(q,p)} \int_{\mathcal{X}\times\mathcal{X}} d(y,x)^k d\pi(y,x) \right)^{\frac{1}{k}} \\ &= W_k(q,p) \end{aligned}$$

iii. To prove the triangle inequality for the Wasserstein distance the Gluing Lemma is crucial. This Lemma can be found e.g. in [21, pp. 23-24]. Provided $(\mathcal{X}_1, p_1)$, $(\mathcal{X}_2, p_2)$ and $(\mathcal{X}_3, p_3)$ are Polish probability spaces the Gluing Lemma states that for each coupling of (p_1, p_2) and (p_2, p_3) there exist random variables X_1, X_2 and X_3 such that (X_1, X_2) has the same distribution as the coupling of (p_1, p_2) and (X_2, X_3) has the same distribution as the coupling of (p_2, p_3).

Thus, the Gluing Lemma can be applied in the setting of the Wasserstein distance for optimal couplings of (p_1, p_2), (p_2, p_3). Then there exist random variables X_1, X_2 and X_3 on $\mathcal{X}$ such that (X_1, X_2) is an optimal coupling of (p_1, p_2), (X_2, X_3) is an optimal coupling of (p_2, p_3) and (X_1, X_3) is a coupling of (p_1, p_3). Using additionally the fact that $d(x,y)$ fulfills as distance the triangle inequality,

$$W_k(p_1,p_3) \leq (\mathbb{E} d(X_1,X_3)^k)^{\frac{1}{k}} \leq (\mathbb{E}\left(d(X_1,X_2) + d(X_2,X_3)\right)^k)^{\frac{1}{k}}$$

holds.

Moreover, with $\|f+g\|_{L^k} \leq \|f\|_{L^k} + \|g\|_{L^k}$ for $f, g \in L^k$, the Minkowski inequality for L^k, the following inequality results

$$W_k(p_1,p_3) \leq (\mathbb{E}\left(d(X_1,X_2)\right)^k)^{\frac{1}{k}} + (\mathbb{E}\left(d(X_2,X_3)\right)^k)^{\frac{1}{k}}. \tag{2.2}$$

Since (X_1, X_2) and (X_2, X_3) are constructed as optimal couplings, the right side of (2.2) equals $W_k(p_1,p_2) + W_k(p_2,p_3)$. Therefore, the triangle inequality is satisfied for the Wasserstein distance.

iv. If $p, q \in P_k(\mathcal{X})$, then

$$\int_{\mathcal{X}} d(x_0,x)^k dp(x) < \infty \text{ and } \int_{\mathcal{X}} d(y,x_0)^k dq(y) < \infty.$$

Additionally from the triangle inequality and $(a+b)^k \leq 2^{k-1}(a^k + b^k)$ for $a, b \geq 0$ the following inequality holds:

$$d(x,y)^k \leq (d(x,x_0) + d(y,x_0))^k \leq 2^{k-1}\left(d(x,x_0)^k + d(y,x_0)^k\right)$$

With this inequality the desired result follows:

$$\begin{aligned} W_k(p,q)^k &= \inf_{\pi\in\Pi(p,q)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^k d\pi(x,y) \\ &\leq \inf_{\pi\in\Pi(p,q)} \int_{\mathcal{X}\times\mathcal{X}} 2^{k-1}\left(d(x,x_0)^k + d(y,x_0)^k\right) d\pi(x,y) \\ &= 2^{k-1}\int_{\mathcal{X}} d(x,x_0)^k dp(x) + 2^{k-1}\int_{\mathcal{X}} d(y,x_0)^k dq(y) \\ &< \infty \end{aligned}$$

□

The proof of Theorem 2.14 uses the properties of the Wasserstein space only to prove the finiteness. Therefore, the Wasserstein distance is also a distance if it is not defined on the Wasserstein space. However, in this case it is not a finite distance.

A comparison of Wasserstein distances with different orders is provided in the next theorem.

Theorem 2.15 (Order of Wasserstein distances) *[21, Remark 6.6] Consider a Polish metric space $(\mathcal{X}, d)$ and $k_1, k_2 \in [1,\infty)$. In this setting*

$$k_1 \leq k_2 \Rightarrow W_{k_1}(p,q) \leq W_{k_1}(p,q)$$

for any $p, q \in P_{k_1}(\mathcal{X}) \cap P_{k_2}(\mathcal{X})$.

Proof Using Hölder's inequality for the functions $|d|^{k_1}$ and 1 and the Hölder conjugates $\frac{k_2}{k_1}$ and $\frac{k_2}{k_2-k_1}$ the following holds:

$$\begin{aligned} &\int_{\mathcal{X}} ||d(x)|^{k_1}\cdot 1|dp(x) \leq \left(\int_{\mathcal{X}} ||d(x)|^{k_1}|^{\frac{k_2}{k_1}} dp(x)\right)^{\frac{k_1}{k_2}} \left(\int_{\mathcal{X}} |1|^{\frac{k_2}{k_2-k_1}} dp(x)\right)^{\frac{k_2-k_1}{k_2}} \\ \Leftrightarrow &\left(\int_{\mathcal{X}} |d(x)|^{k_1} dp(x)\right)^{\frac{1}{k_1}} \leq \left(\int_{\mathcal{X}} |d(x)|^{k_2} dp(x)\right)^{\frac{1}{k_2}}. \end{aligned}$$

[23, p. 63]

And thus, immediately

$$W_{k_1}(p,q) \leq W_{k_2}(p,q)$$

follows. □

As a consequence the Wasserstein distance of order 1 is the weakest Wasserstein distance. Thus, it can be bounded easier than the other distance. In contrast the Wasserstein distance of order 2 admits better properties in a geometric context. Therefore, these orders are the most convenient. [21, Remark 6.6]

An important property of the Wasserstein distance of each order is the connection of its convergence to the weak convergence in the Wasserstein space. This weak convergence in $P_k(\mathcal{X})$ is characterized in the following definition.

Definition 2.16 (Weak convergence in $P_k(\mathcal{X})$) *[21, Definition 6.8] Consider a Polish metric space $(\mathcal{X}, d)$ and $k \in [1, \infty)$. For a sequence $(p_n)_{n\in\mathbb{N}}$ of probability measures in $P_k(\mathcal{X})$ and $p \in P_k(\mathcal{X})$ the following statements are equivalent:*

i. p_n converges weakly to p and

$$\int_{\mathcal{X}} d(x_0, x)^k dp_n(x) \to \int_{\mathcal{X}} d(x_0, x)^k dp(x).$$

ii. p_n converges weakly to p and

$$\limsup_{n\to\infty} \int_{\mathcal{X}} d(x_0, x)^k dp_n(x) \leq \int_{\mathcal{X}} d(x_0, x)^k dp(x).$$

iii. p_n converges weakly to p and

$$\lim_{R\to\infty} \limsup_{n\to\infty} \int_{d(x_0,x)\geq R} d(x_0, x)^k dp_n(x) = 0.$$

iv. For every continuous ϕ with $|\phi(x)| \leq C(1 + d(x_0, x)^k)$ for $C \in \mathbb{R}$

$$\int_{\mathcal{X}} \phi(x) dp_n(x) \to \int_{\mathcal{X}} \phi(x) dp(x)$$

is satisfied.

If these statements are satisfied for any x_0, p_n is said to converge weakly in $P_k(\mathcal{X})$.

To prove the connection of the convergence in the Wasserstein distance with the weak convergence in $P_k(\mathcal{X})$ the next Lemma is required.

Lemma 2.17 (Tightness of Cauchy sequences in $(P_k(\mathcal{X}), W_k)$) *[21, Lemma 6.14] Consider a Polish space $\mathcal{X}$ and $k \in [1, \infty)$. If $(p_n)_{n\in\mathbb{N}}$ is a Cauchy sequence in the metric space $(P_k(\mathcal{X}), W_k)$, then (p_n) is tight.*

With this knowledge the following theorem can be proven:

Theorem 2.18 (Wasserstein distance metrizes Wasserstein space) *[21, Theorem 6.9] Consider a Polish metric space $(\mathcal{X}, d)$, $k \in [1, \infty)$ and $p \in P_k(\mathcal{X})$. If and only if a sequence $(p_n)_{n\in\mathbb{N}}$ of probability measures in $P_k(\mathcal{X})$ converges weakly in $P_k(\mathcal{X})$ to p then $W_k(p_n, p) \to 0$.*

Proof [21, Proof of Theorem 6.9] "$\Leftarrow$" Let $(p_n)_{n\in\mathbb{N}}$ be a sequence in $P_k(\mathcal{X})$ such that $W_k(p_n, p) \to 0$. Since $(p_n)_{n\in\mathbb{N}}$ is a Cauchy sequence in $P_k(\mathcal{X})$ from Lemma 2.17, it is clear that the sequence is tight. Using Prokohorov's

Theorem 2.3 it follows immediately that there exists a subsequence $(p_{n'})$ which converges weakly to a probability measure $\hat{p}$. Thus, Lemma 2.4 can be applied and results in

$$W_k(\hat{p}, p) \leq \liminf_{n' \to \infty} W_k(p_{n'}, p).$$

Therefore, with $W_k(p_{n'}, p) \to 0$ it follows that $\hat{p} = p$. Since this argument can be applied to every converging subsequence, $(p_n)_{n \in \mathbb{N}}$ converges weakly to p as well.

To prove that $(p_n)_{n \in \mathbb{N}}$ also converges weakly in $P_k(\mathcal{X})$ to p, by Definition 2.16 (ii.) it has additionally to be shown that

$$\limsup_{k \to \infty} \int_{\mathcal{X}} d(x_0, x)^p d\mu_k(x) \leq \int_{\mathcal{X}} d(x_0, x)^p d\mu(x)$$

holds. In order to do so the following statement for $a, b \in \mathbb{R}_+$ will be used:

$$\forall \varepsilon > 0 \, \exists a_\varepsilon > 0 \qquad (a+b)^k \leq (1+\varepsilon)a^k + a_\varepsilon b^k.$$

Using this inequality for the distances between the points $x_0, x, y \in \mathcal{X}$ and the triangle inequality one gets

$$d(x_0, x)^k \leq (d(x_0, y) + d(x, y))^k \leq (1+\varepsilon) d(x_0, y)^k + a_\varepsilon d(x, y)^k. \tag{2.3}$$

Now consider an optimal transference plan $\pi_n \in \Pi(p_n, p)$. Then (2.3) is equivalent to

$$\begin{aligned}
&\int_{\mathcal{X} \times \mathcal{X}} d(x_0, x)^k d\pi_n(x, y) \\
&\leq (1+\varepsilon) \int_{\mathcal{X} \times \mathcal{X}} d(x_0, y)^k d\pi_n(x, y) + a_\varepsilon \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^k d\pi_n(x, y) \\
\Leftrightarrow &\int_{\mathcal{X}} d(x_0, x)^k dp_n(x) \\
&\leq (1+\varepsilon) \int_{\mathcal{X}} d(x_0, y)^k dp(y) + a_\varepsilon \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^k d\pi_n(x, y).
\end{aligned}$$

Since by assumption for $n \to \infty$

$$\int_{\mathcal{X} \times \mathcal{X}} d(x, y)^k d\pi_n(x, y) = W_k(p_n, p)^k \to 0,$$

when choosing $y = x$ the following holds:

$$\limsup_{n \to \infty} \int_{\mathcal{X}} d(x_0, x)^k dp_n(x) \leq (1+\varepsilon) \int_{\mathcal{X}} d(x_0, x)^k dp(x).$$

Letting $\varepsilon \to 0$ the desired statement results:

$$\limsup_{n \to \infty} \int_{\mathcal{X}} d(x_0, x)^k dp_n(x) \leq \int_{\mathcal{X}} d(x_0, x)^k dp(x).$$

"⇒" Consider a sequence of probability measures $(p_n)_{n\in\mathbb{N}}$ in $P_k(\mathcal{X})$ which converges weakly to $p \in P_k(\mathcal{X})$. In addition let $(\pi_n)_{n\in\mathbb{N}}$ be a sequence of optimal transference plans between p_n and p. Since $(p_n)_{n\in\mathbb{N}}$ is by Prokhorov's Theorem 2.3 tight and p is also tight, from Lemma 2.5 follows that $(\pi_n)_{n\in\mathbb{N}}$ is tight as well. Thus, again by Prokhorov's Theorem there exists at least a subsequence of $(\pi_n)_{n\in\mathbb{N}}$ converging weakly to a $\pi \in P(\mathcal{X}\times\mathcal{X})$. From Theorem 2.11 follows that π is as limit of optimal transference plans itself optimal for the limit of the marginals. So π is the optimal coupling between the probability measures p and p and thus the identity coupling. Since this argument holds for every subsequence, the sequence $(\pi_n)_{n\in\mathbb{N}}$ itself converges weakly to π (the identity coupling).

For any $x_0 \in \mathcal{X}$ and $R > 0$

$$\begin{aligned} &R \leq d(x,y) \leq d(x_0,y) + d(x,x_0) \\ &\Rightarrow max(d(x_0,y), d(x,x_0)) \geq \frac{R}{2} \text{ and } max(d(x_0,y), d(x,x_0)) \geq \frac{d(x,y)}{2}. \end{aligned}$$

And thus,

$$\begin{aligned} &max(0, d(x,y)^k - R^k) = d(x,y)^k 1_{[d(x,y)\geq R]} \\ &\leq d(x,y)^k 1_{[d(x_0,y)\geq \frac{R}{2}]\cap[d(x_0,y)\geq \frac{d(x,y)}{2}]} + d(x,y)^k 1_{[d(x,x_0)\geq \frac{R}{2}]\cap[d(x,x_0)\geq \frac{d(x,y)}{2}]} \\ &\leq 2^k d(x_0,y)^k 1_{[d(x_0,y)\geq \frac{R}{2}]} + 2^k d(x,x_0)^k 1_{[d(x,x_0)\geq \frac{R}{2}]} \end{aligned}$$

holds.

This inequality can be applied in the third line of the following while in the first line the definition of π_n as optimal transport is used:

$$\begin{aligned} &W_k(p_n,p)^k = \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^k d\pi_n(x,y) \\ &\leq \int_{\mathcal{X}\times\mathcal{X}} min(d(x,y)^k, R^k) + max(0, d(x,y)^k - R^k) d\pi_n(x,y) \\ &\leq \int_{\mathcal{X}\times\mathcal{X}} min(d(x,y)^k, R^k) + 2^k d(x_0,y)^k 1_{[d(x_0,y)\geq \frac{R}{2}]} \\ &\quad + 2^k d(x,x_0)^k 1_{[d(x,x_0)\geq \frac{R}{2}]} d\pi_n(x,y) \\ &= \int_{\mathcal{X}\times\mathcal{X}} min(d(x,y)^k, R^k) d\pi_n(x,y) + 2^k \int_{\mathcal{X}\times\mathcal{X}} d(x_0,y)^k 1_{[d(x_0,y)\geq \frac{R}{2}]} d\pi_n(x,y) \\ &\quad + 2^k \int_{\mathcal{X}\times\mathcal{X}} d(x,x_0)^k 1_{[d(x,x_0)\geq \frac{R}{2}]} d\pi_n(x,y) \\ &= \int_{\mathcal{X}\times\mathcal{X}} min(d(x,y)^k, R^k) d\pi_n(x,y) + 2^k \int_{\mathcal{X}\times\mathcal{X}} d(x_0,y)^k 1_{[d(x_0,y)\geq \frac{R}{2}]} dp(x,y) \\ &\quad + 2^k \int_{\mathcal{X}\times\mathcal{X}} d(x,x_0)^k 1_{[d(x,x_0)\geq \frac{R}{2}]} dp_n(x,y) \end{aligned} \tag{2.4}$$

Taking the limit of inequality (2.4)

$$\begin{aligned}
&\limsup_{n\to\infty} W_k(p_n, p) \\
\leq &\lim_{n\to\infty} \int_{\mathcal{X}\times\mathcal{X}} min(d(x,y)^k, R^k) d\pi_n(x,y) \\
&+ 2^k \lim_{R\to\infty} \limsup_{n\to\infty} \int_{\mathcal{X}\times\mathcal{X}} d(x_0,y)^k 1_{[d(x_0,y)\geq \frac{R}{2}]} dp(x,y) \\
&+ 2^k \lim_{R\to\infty} \limsup_{n\to\infty} \int_{\mathcal{X}\times\mathcal{X}} d(x,x_0)^k 1_{[d(x,x_0)\geq \frac{R}{2}]} dp_n(x,y)
\end{aligned}$$

results. Since π_n converges weakly to π as $n \to \infty$ and $min(d(x,y)^k, R^k)$ is a bounded and continuous function, the first term becomes 0. As also the second and third term become zero, the convergence $p_n \to p$ in the Wasserstein distance is shown. □

Thus, the Wasserstein distance metrizes the Wasserstein space. This result can be applied to prove the continuity of the Wasserstein distance.

Corollary 2.19 (Continuity of W_k on $P_k(\mathcal{X})$) *[21, Corollary 6.11] Consider a Polish metric space $(\mathcal{X}, d)$ and $k \in [1, \infty)$. Moreover, let $(p_n)_{n\in\mathbb{N}}$ and $(q_n)_{n\in\mathbb{N}}$ be sequences of probability measures and p, q probability measures in $P_k(\mathcal{X})$. If p_n converges weakly to p and q_n to q for $n \to \infty$, then*

$$W_k(p_n, q_n) \to W_k(p, q).$$

Proof Since from Theorem 2.18 it is known that weak convergence in $P_k(\mathcal{X})$ is equivalent to convergence of W_k, it follows immediately that for $(p_n)_{n\in\mathbb{N}}$ converging weakly to p in $P_k(\mathcal{X})$ and $(q_n)_{n\in\mathbb{N}}$ to q the following holds:

$$W_k(p_n, p) \to 0 \text{ and } W_k(q_n, q) \to 0$$

Applying the triangle inequality

$$\begin{aligned}
&W_k(p_n, q_n) \leq W_k(p_n, p) + W_k(p, q) + W_k(q, q_n) \text{ and} \\
&W_k(p, q) \leq W_k(p, p_n) + W_k(p_n, q_n) + W_k(q_n, q)
\end{aligned}$$

follow.

Taking the limit of both inequalities for $n \to 0$

$$W_k(p_n, q_n) \to W_k(p, q).$$

□

Therefore, the Wasserstein distance of order k is indeed continuous on $P_k(\mathcal{X})$.

So it was shown that the Wasserstein distances of each order have several nice properties. However, the Kantorovich-Rubinstein formula applies only to the Wasserstein distance of order 1. It represents the dual formulation of the 1-Wasserstein distance.

Theorem 2.20 (Kantorovich-Rubinstein formula) *[21, Remark 6.5] Consider a Polish metric space $(\mathcal{X}, d)$ and let $p, q \in P_1(\mathcal{X})$. Then the Wasserstein distance of order 1 has the following dual representation:*

$$W_1(p,q) = \sup_{\|f\|_{Lip} \leq 1} \left(\int_{\mathcal{X}} f(x) dp(x) - \int_{\mathcal{X}} f(x) dq(x) \right).$$

Proof [21, Particular Case 5.4, Remark 6.5] Applying Theorem 2.9 to the Wasserstein distance of order 1 the equality

$$\begin{aligned} W_1(p,q) &= \inf_{\pi \in \Pi(p,q)} \int_{\mathcal{X} \times \mathcal{X}} d(x,y) d\pi(x,y) \\ &= \sup_{f\, d-convex} \left(\int_{\mathcal{X}} f(x) dq(x) - \int_{\mathcal{X}} f^d(x) dp(x) \right) \end{aligned} \tag{2.5}$$

with f^d as the d-transform of function f holds.

In order to get the desired result two statements have to be proven. First it must be shown that the d-transform of a d-convex function f is the function itself. Additionally it must hold that a function f is d-convex if and only if it is 1-Lipschitz.

The first statement can be checked straightforward by using the definition of the d-transform of f:

$$f^d(y) = \inf_{x \in \mathcal{X}} (f(x) + d(x,y)) = f(y) + d(y,y) = f(y) \qquad \forall y \in \mathcal{X}$$

Since for a 1-Lipschitz function f obviously holds that

$$f(x) = f(x) - d(x,x) = \sup_{y \in \mathcal{X}} (f(y) - d(x,y)) \qquad \forall x \in \mathcal{X},$$

the function f is d-convex. A d-convex function on the other hand is 1-Lipschitz continuous. Thus, also the second statement holds. □

If p and q are discrete probability distributions, finding the Wasserstein distance of order 1 corresponds to the transportation problem.

Definition 2.21 (Transportation problem) *[9, pp. 307-309] Consider a discrete Polish metric space $(\mathcal{X}, d)$ and discrete probability measures p, q on $\mathcal{X}$ such that*

$$p = \sum_{i=1}^{n} p_i \delta_{x_i},\ q = \sum_{j=1}^{m} q_j \delta_{y_j}$$

for x_i, $y_j \in \mathcal{X}$. Let $D = (d_{ij}) = (d(x_i, y_j)) \in \mathbb{R}^{n \times m}$ be a cost matrix and $\pi = (\pi_{ij}) \in \mathbb{R}^{n \times m}$ be a transportation matrix.

Then $W_1(p,q)$ is the solution of the following problem:

$$\begin{aligned} &\min_{\pi \in \mathbb{R}^{n \times m}} \sum_{j=1}^{m} \sum_{i=1}^{n} d_{ij} \pi_{ij} \\ &\sum_{j=1}^{m} \pi_{ij} = p_i \text{ for } i = 1, ..., n \\ &\sum_{i=1}^{n} \pi_{ij} = q_j \text{ for } j = 1, ..., m \\ &\pi_{ij} \geq 0 \ \forall i, j \end{aligned} \tag{2.6}$$

which is called the transportation problem.

The constraints of the transportation problem ensure π to be from the set of joint probability distributions of p and q: $\Pi(p,q)$.

Since the transportation problem is a linear program, it can be solved by using the simplex algorithms. More specifically, to solve a transportation problem the transportation simplex can be used, a simplex algorithm taking advantage of the special structure of the transportation problem.

Definition 2.22 (Transportation simplex) *[9, pp. 316-329] In the setting of Definition 2.21 the transportation simplex is an algorithm consisting of the following steps:*

1. *find a basic solution which satisfies the constraints*
2. *set $u_i = 0$ for an arbitrary $i \in 1, ..., n$ and for all (i,j) with π_{ij} being a basic variable solve $d_{ij} = u_i + v_j$*
3. *perform optimality test: if $d_{ij} - u_i - v_j \geq 0$ for $i = 1, ..., n$, $j = 1, ..., m$ then the basic solution is optimal and the algorithm can be stopped*
4. *find another basic feasible solution*
   a) *choose π_{ij} with minimal $d_{ij} - u_i - v_j$ as new basic variable*
   b) *identify which basic variables decrease as the new basic variable increases and eliminate the smallest of them*
   c) *assign the value of the eliminated basic variable to a new basic variable*
   d) *update the basic variables according to the constraints as well as u_i for $i = 1, ..., n$ and v_j for $j = 1, ..., m$*
   e) *go to 3.*

Therefore, it is possible to calculate the exact Wasserstein distance of order 1 between discrete probability distributions as long as n and m are small enough. For very large n and m the transportation simplex can become too complex. In those cases the 1-Wasserstein distance can be approximated by solving a coarser discretization of the problem.

Such an approximation can also be used if the Wasserstein distance of order 1 between two continuous probability distributions should be calculated. This is done e.g. when calculating the 1-Wasserstein distance between the colour distributions of images by using their signatures as presented in [16, pp. 7-8]. Here, an image signature is a set $\{s_i = (m_i, w_i)\}$ of clusters which are described by their mean m_j and the percentage of pixels w_j the cluster contains.

Consider two images with the signatures $A = \{(a_1, w_{a_1}), ..., (a_n, w_{a_n})\}$ and $B = \{(b_1, w_{b_1}), ..., (b_m, w_{b_m})\}$. Then the 1-Wasserstein distance between the two signatures A and B is $W_1(p, q)$ where

$$p := \sum_{i=1}^{n} w_{a_i} \delta_{a_i} \text{ and } q := \sum_{j=1}^{m} w_{b_j} \delta_{b_j}.$$

$W_1(p, q)$ in turn is the solution of the transportation problem (2.6) where d_{ij} denotes the ground distance between a_i and b_j one has to define. Using the transportation simplex this problem can be solved exactly if the signature of the images is not too long. [16, pp. 7-8]

There are several other ways to approximate different versions of the Wasserstein distance. As an example Solomon et al. presented in [19] a method to approximate the Wasserstein distance of order 2 with the so called convolutional Wasserstein distance. This convolutional Wasserstein distance is found by the construction of optimal transports via iterative kernel convolutions. Another example is the linear approximation algorithm for a Wasserstein distance of order 1 with a thresholded ground distance presented by Li et al. in [12].

2.3 Learning by the Earth-Mover Distance

As already mentioned in the previous section the term Earth-Mover distance describes in this thesis the Wasserstein distance of order 1 with a metric induced by a norm. Thus, the Earth-Mover distance between probability measures p and q is

$$\begin{aligned} EM(p,q) &:= \inf_{\pi \in \Pi(p,q)} \int_{\mathcal{X}\times\mathcal{X}} \|x-y\| d\pi(x,y) \\ &= \inf_{\pi \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\pi} \|x-y\|. \end{aligned} \tag{2.7}$$

[1, p. 4]

The name Earth-Mover distance is derived from a motivational example of the distance. Assume p as the distribution of an earth pile and q as distribution of holes which have to be filled with earth. Thus, $p(x)$ is the amount of earth in a specific space segment x and $q(x)$ is the amount of earth which must be transported in a specific space segment x. In this case $\pi \in \Pi(p,q)$ can be interpreted as a transport plan of how to move the earth from the earth pile into the holes. Therefore, the Earth-Mover distance would calculate the minimal cost of filling the holes with earth from the earth pile. [16, p. 7]

A distribution p can be learned by a function $g_\theta(z)$ with distribution p_θ by adapting θ in such a way that $EM(p, p_\theta)$ is minimized. To do so, $EM(p, p_\theta)$ should be continuous in θ and differentiable almost everywhere. The next theorem will show that this is indeed true if g_θ is locally Lipschitz and there exist local Lipschitz constants $L(\theta, z)$ such that $\mathbb{E}_{z\sim p_z} L(\theta, z) < \infty$.

Theorem 2.23 (EM of p and p_θ is continuous on θ) *[1, Theorem 1 (i,ii)] Consider a compact metric space $\mathcal{X}$, a space $\mathcal{Z}$ and $d \in \mathbb{N}$. Let $g_\theta : \mathcal{Z} \to \mathcal{X}$ for $\theta \in \mathbb{R}^d$ a function and $Z \sim p_z$ a random variable on $\mathcal{Z}$. Moreover, p_θ denotes the distribution of $g_\theta(Z)$ and p a distribution over $\mathcal{X}$. Then the following statements hold*

i. g_θ is continuous in $\theta \Rightarrow EM(p, p_\theta)$ is continuous in θ

ii. g_θ is locally Lipschitz and local Lipschitz constants $L(\theta, z)$ exist such that $\mathbb{E}_{z\sim p_z} L(\theta, z) < \infty \Rightarrow EM(p, p_\theta)$ is continuous everywhere and differentiable almost everywhere.

Proof [1, Proof of Theorem 1 (i,ii)]

i. Let $\theta, \hat{\theta} \in \mathbb{R}^d$ and π such that $(g_\theta(Z), g_{\hat{\theta}}(Z)) \sim \pi$. Thus, π is a transference plan between p_θ and $p_{\hat{\theta}}$ and it holds that

$$EM(p_\theta, p_{\hat{\theta}}) \leq \mathbb{E}_{(x,y)\sim\pi} \|x-y\| = \mathbb{E}_{z\sim p_z} \|g_\theta(z) - g_{\hat{\theta}}(z)\|. \tag{2.8}$$

If it can be shown that $\|g_\theta(z) - g_{\hat{\theta}}(z)\|$ can be uniformly bounded by a constant and moreover, converges pointwise to 0 then by the bounded convergence theorem its expectation converges to 0.

Since $\mathcal{X}$ is compact, there exists a constant which bounds $\|g_\theta(z) - g_{\hat{\theta}}(z)\|$ uniformly. In addition the function g_θ is by assumption continuous, so for $\theta \to \hat{\theta}$

$$g_\theta(z) \to g_{\hat{\theta}}(z)$$

and

$$\|g_\theta(z) - g_{\hat{\theta}}(z)\| \to 0 \quad \text{pointwise.}$$

Therefore, the conditions of the bounded convergence theorem are fulfilled and for $\theta \to \hat{\theta}$ the convergence

$$\mathbb{E}_{z\sim p_z}\|g_\theta(z) - g_{\hat{\theta}}(z)\| \to 0$$

follows. From (2.8) it is clear that $EM(p_\theta, p_{\hat{\theta}})$ converges for $\theta \to \hat{\theta}$ as well to 0. With the triangle inequality this implies the continuity of $EM(p, p_\theta)$:

$$|EM(p, p_\theta) - EM(p, p_{\hat{\theta}})| \leq EM(p_\theta, p_{\hat{\theta}}) \to 0 \quad \text{for } \theta \to \hat{\theta}$$

ii. From the definition of local Lipschitz continuity there exist for every (θ, z) an open set U which satisfies $(\theta, z) \in U$ and

$$\|g_\theta(z) - g_{\hat{\theta}}(\hat{z})\| \leq L(\theta, z)\left(\|\theta - \hat{\theta}\| + \|z - \hat{z}\|\right) \quad \forall(\hat{\theta}, \hat{z}) \in U$$

When taking the expected value and choosing $\hat{z} = z$ the second condition implies

$$\mathbb{E}_{z\sim p_z}\|g_\theta(z) - g_{\hat{\theta}}(z)\| \leq \mathbb{E}_{z\sim p_z} L(\theta, z)\|\theta - \hat{\theta}\| \quad \forall(\hat{\theta}, z) \in U.$$

By using the definition of the Earth-Mover distance and the triangle inequality it holds that $\forall(\hat{\theta}, z) \in U$

$$\begin{aligned} |EM(p, p_\theta) - EM(p, p_{\hat{\theta}})| &\leq EM(p_\theta, p_{\hat{\theta}}) \\ &\leq \mathbb{E}_{z\sim p_z}\|g_\theta(z) - g_{\hat{\theta}}(z)\| \\ &\leq \mathbb{E}_{z\sim p_z} L(\theta, z)\|\theta - \hat{\theta}\|. \end{aligned}$$

Since $L(\theta) \coloneqq \mathbb{E}_{z\sim p_z} L(\theta, z)$ is by assumption finite and as U is open also $U_\theta \coloneqq \{\theta | (\theta, z) \in U\}$ is open, $EM(p, p_\theta)$ is locally Lipschitz continuous:

$$|EM(p, p_\theta) - EM(p, p_{\hat{\theta}})| \leq L(\theta)\|\theta - \hat{\theta}\| \quad \forall(\hat{\theta}) \in U_\theta.$$

The local Lipschitz continuity implies that $EM(p, p_\theta)$ is continuous everywhere. Since Rademacher's theorem states that a locally Lipschitz continuous function is almost everywhere differentiable, this applies to $EM(p, p_\theta)$.

□

Thus, to learn p with $EM(p, p_\theta)$ only makes sense if g_θ satisfies respective conditions. A class of functions which satisfy these conditions are feedforward neural networks if they are defined as follows:

Definition 2.24 *[1, p. 5] Consider a compact metric space $\mathcal{X}$, a space $\mathcal{Z}$ and $d \in \mathbb{N}$. A function $g_\theta : \mathcal{Z} \to \mathcal{X}$ is called a feedforward neural network if*

$$g_\theta(z) = (h_N \circ A_N \circ h_{N-1} \circ A_{N-1} \circ ... \circ h_1 \circ A_1)(z)$$

holds where $A_1, ..., A_N$ are affine transformations and $h_1, ..., h_{N-1}$ are pointwise non-linear smooth Lipschitz continuous functions. The parameter $\theta \in \mathbb{R}^d$ are defined as the weights $W_1, ..., W_N$ of the affine transformations.

This definition includes neural networks with activation functions as tanh, sigmoid or elu. [1, p. 5] However, neural networks with activation function that are not smooth or Lipschitz continuous such as threshold functions are not considered in the following corollary.

Therefore, the fact that feedforward neural networks satisfy the conditions of Theorem 2.23 can now be proven.

Corollary 2.25 (Learning by EM makes sense) *[1, Corollary 1] Consider the compact metric space $\mathcal{X}$, the space $\mathcal{Z}$ and $d \in \mathbb{N}$. Let $g_\theta : \mathcal{Z} \to \mathcal{X}$ with parameters $\theta \in \mathbb{R}^d$ be a feedforward neural network. Moreover, let Z be a random variable on $\mathcal{Z}$ with distribution p_z such that $\mathbb{E}_{z\sim p_z}\|z\| < \infty$. If p_θ denotes the distribution of $g_\theta(Z)$ and p be a distribution over $\mathcal{X}$, then g_θ is locally Lipschitz continuous and there are local Lipschitz constants $L(\theta, z)$ of g_θ such that $\mathbb{E}_{z\sim p_z} L(\theta, z) < \infty$. Additionally follows that $EM(p, p_\theta)$ is continuous everywhere and differentiable almost everywhere.*

Proof [1, Proof of Corollary 1] To prove this corollary it only has to be shown that the conditions

i. g_θ is locally Lipschitz continuous

ii. there are local Lipschitz constants $L(\theta, z)$ of g_θ with $\mathbb{E}_{z\sim p_z} L(\theta, z) < \infty$

are satisfied for feedforward neural networks g_θ. In this case it follows by 2.23 (ii.) that $EM(p, p_\theta)$ is continuous everywhere and differentiable almost everywhere.

Feedforward neural networks in this context are defined as compositions of affine transformations and pointwise non-linear smooth Lipschitz continuous functions.

Consider g_θ as such a feedforward neural network. Since a composition of locally Lipschitz functions is as well locally Lipschitz, it is known that condition (i.) is satisfied.

As the pointwise non-linearities are additionally assumed to be smooth, g_θ as composition of smooth functions is also smooth.

Thus, for their local Lipschitz constants holds:

$$L(\theta, z) \leq \|\nabla_{\theta,z} g_\theta(z)\| + \varepsilon \quad \forall \varepsilon > 0$$

Therefore,

$$\mathbb{E}_{z\sim p_z}\|\nabla_{\theta,z} g_\theta(z)\| < \infty \Rightarrow \mathbb{E}_{z\sim p_z} L(\theta, z) < \infty$$

holds and if the first is shown both conditions are satisfied.

A feedforward neural network consists of N layers where each layer applies a composition of an affine transformation and a non-linearity on the output of the previous layer respectively the input z. Consider now the matrices W_n for $n = 1, ..., N$ as weight matrices of the affine transformation performed in the n-th layer and D_n for $n = 1, ..., N$ as the Jacobians of the non-linearity in the n-th layer. The composition of all functions performed on the first n layers is denoted by f_n.

Therefore, differentiating the network with respect to z results in

$$\nabla_z g_\theta(z) = \prod_{i=1}^{N} W_i D_i$$

and differentiating with respect to the weights which are the parameters of g_θ results in

$$\nabla_{W_n} g_\theta(z) = \left(\left(\prod_{i=n+1}^{N} W_i D_i \right) D_n \right) f_{n-1}(z).$$

Thus, the inequality

$$\begin{aligned} \|\nabla_{\theta,z} g_\theta(z)\| &\leq \sum_{n=1}^{N} \|\nabla_{W_n} g_\theta(z)\| + \|\nabla_z g_\theta(z)\| \\ &= \sum_{n=1}^{N} \| \left(\prod_{i=n+1}^{N} W_i D_i \right) D_n f_{n-1}(z)\| + \| \prod_{i=1}^{N} W_i D_i\| \end{aligned} \tag{2.9}$$

holds.

Note, that for L being the Lipschitz constant of the non-linearities the inequalities

$$\begin{aligned} &\|D_i\| \leq L \quad \forall i = 1, ..., N \quad \text{and} \\ &\|f_{n-1}(z)\| \leq \|z\| L^{n-1} \prod_{i=1}^{n-1} \|W_i\| \quad \forall n = 1, ..., N \end{aligned}$$

follow.

With these inequalities and (2.9) one gets

$$\begin{aligned}
&\mathbb{E}_{z\sim p_z}\|\nabla_{\theta,z} g_\theta(z)\| \\
&\leq \mathbb{E}_{z\sim p_z}\left(\sum_{n=1}^{N}\| \left(L^{N-n-1}\prod_{i=n+1}^{N} W_i\right) L\| \left(\|z\| L^{n-1}\prod_{i=1}^{n-1} W_i\right) + L^N \prod_{i=1}^{N}\|W_i\|\right) \\
&\leq \mathbb{E}_{z\sim p_z}\left(\|z\| L^N \sum_{n=1}^{N}\left(\prod_{i=n+1}^{N}\|W_i\|\right)\left(\prod_{i=1}^{n-1}\|W_i\|\right) + L^N\prod_{i=1}^{N}\|W_i\|\right) \\
&= \mathbb{E}_{z\sim p_z}\left(\|z\|\right) L^N \sum_{n=1}^{N}\left(\prod_{i=n+1}^{N}\|W_i\|\right)\left(\prod_{i=1}^{n-1}\|W_i\|\right) + L^N\prod_{i=1}^{N}\|W_i\|
\end{aligned}$$

The last term is by the assumption $\mathbb{E}_{z\sim p_z}\|z\| < \infty$ finite and therefore $\mathbb{E}_{z\sim p_z} L(\theta, z) < \infty$ holds. □

Note, that the condition $\mathbb{E}_{z\sim p_z}\|z\| < \infty$ is satisfied for z being e.g. Gaussian or uniformly distributed. [1, Corollary 1] Thus, using a feedforward neural network over such a prior z to learn a distribution p makes sense by minimizing $EM(p, p_\theta)$.

To minimize $EM(p, p_\theta)$ in order to learn p the gradient of $EM(p, p_\theta)$ is used. The next theorem describes the gradient of $EM(p, p_\theta)$ in terms of the solution of an optimization problem.

Theorem 2.26 (Gradient of $EM(p, p_\theta)$) *[1, Theorem 3] Consider the compact metric space $\mathcal{X}$, the space $\mathcal{Z}$ and $d \in \mathbb{N}$. Let Z be a random variable on $\mathcal{Z}$ with distribution p_z. Moreover, let $g_\theta : \mathcal{Z} \to \mathcal{X}$ with parameters $\theta \in \mathbb{R}^d$ be a locally Lipschitz function such that there exist local Lipschitz constants $L(\theta, z)$ with $\mathbb{E}_{z\sim p_z} L(\theta, z) < \infty$. Then, for p_θ denoting the distribution of $g_\theta(Z)$ and p a distribution over $\mathcal{X}$, a function $f^* : \mathcal{X} \to \mathbb{R}$ exists such that*

$$f^* = \underset{\|f\|_{Lip}\leq 1}{\operatorname{argmax}} \mathbb{E}_{x\sim p} f(x) - \mathbb{E}_{x\sim p_\theta} f(x).$$

Additionally,

$$\nabla_\theta EM(p, p_\theta) = -\mathbb{E}_{z\sim p_z} \nabla_\theta f^*(g_\theta(z))$$

holds if both sides of the equation are well-defined.

Proof [1, Proof of Theorem 3] From 2.20 and the assumptions it is clear that

$$\begin{aligned}
EM(p, p_\theta) &= \sup_{f\in\mathcal{F}} \left(\mathbb{E}_{x\sim p}(f(x)) - \mathbb{E}_{x\sim p_\theta}(f(x))\right) \\
&= \sup_{f\in\mathcal{F}} \left(\mathbb{E}_{x\sim p}(f(x)) - \mathbb{E}_{z\sim p_z}(f(g_\theta(z)))\right).
\end{aligned}$$

holds, where $\mathcal{F} := \{f : \mathcal{X} \to \mathbb{R} | \|f\|_{Lip} \leq 1\}$.

By assumption $\mathcal{X}$ is compact and thus in the setting of the Earth-Mover distance also the additional conditions of Theorem 2.9 are satisfied. Therefore, the supremum in the dual formulation is attained and for

$$V(f,\theta) \coloneqq \mathbb{E}_{x\sim p}(f(x)) - \mathbb{E}_{z\sim p_z}(f(g_\theta(z)))$$

the set $X(\theta) = \{f \in \mathcal{F} | EM(p,p_\theta) = V(f,\theta)\}$ is non-empty. From the envelope theorem in [13, Theorem 1] follows

$$\nabla_\theta EM(p,p_\theta) = \nabla_\theta V(f,\theta)$$

for all $f \in X$ if both sides of the equation are well-defined.

Since

$$\nabla_\theta V(f,\theta) = \nabla_\theta \mathbb{E}_{x\sim p}(f(x)) - \mathbb{E}_{z\sim p_z}(f(g_\theta(z))) = -\nabla_\theta \mathbb{E}_{z\sim p_z} f(g_\theta(z))$$

for all $f \in X$, it remains to show that

$$-\nabla_\theta \mathbb{E}_{z\sim p_z} f(g_\theta(z)) = -\mathbb{E}_{z\sim p_z} \nabla_\theta f(g_\theta(z)). \tag{2.10}$$

As $f \in X$ is 1-Lipschitz and g_θ is locally Lipschitz continuous with constants $L(\theta,z)$ the composition $f(g_\theta(z))$ is also locally Lipschitz continuous with constants $L(\theta,z)$. Therefore, by Rademacher's Theorem $f(g_\theta(z))$ is almost everywhere for (θ,z) differentiable, meaning that the measure of the set $A = \{(\theta,z) | f(g_\theta(z)) \text{ is not differentiable}\}$ is zero. Then Fubini's Theorem implies that also the set $A_\theta = \{z | f(g_\theta(z)) \text{ is not differentiable}\}$ has measure zero for almost every θ.

Choosing now a $\hat{\theta}$ such that $A_{\hat{\theta}}$ has indeed measure zero, $\nabla_\theta f(g_\theta(z))|_{\hat{\theta}}$ is well-defined for almost any z. Since additionally

$$\mathbb{E}_{z\sim p_z}(\nabla_\theta f(g_\theta(z))|_{\hat{\theta}}) \leq \mathbb{E}_{z\sim p_z} L(\hat{\theta},z) < \infty$$

holds, the right side of (2.10) is for almost every θ well-defined.

It remains to show that also the left side of (2.10) is well-defined if the right side is. This can be done by proving that

$$\begin{aligned} &\frac{\mathbb{E}_{z\sim p_z} f(g_\theta(z)) - \mathbb{E}_{z\sim p_z}(f(g_{\hat{\theta}}(z))}{\|\theta-\hat{\theta}\|} - \frac{\left\langle (\theta-\hat{\theta}), \mathbb{E}_{z\sim p_z} \nabla_\theta (f(g_\theta(z))|_{\hat{\theta}}) \right\rangle}{\|\theta-\hat{\theta}\|} \\ &= \mathbb{E}_{z\sim p_z}\left(\frac{f(g_\theta(z)) - f(g_{\hat{\theta}}(z)) - \left\langle (\theta-\hat{\theta}), \nabla_\theta f(g_\theta(z))|_{\hat{\theta}} \right\rangle}{\|\theta-\hat{\theta}\|} \right). \end{aligned} \tag{2.11}$$

converges to 0 as $\theta \to \hat{\theta}$ using the Theorem of dominated convergence.

For $\theta \to \hat{\theta}$ $\frac{f(g_\theta(z)) - f(g_{\hat{\theta}}(z)) - \left\langle (\theta - \hat{\theta}), \nabla_\theta f(g_\theta(z))|_{\hat{\theta}} \right\rangle}{\|\theta - \hat{\theta}\|}$ converges to zero p_z-almost everywhere. As $\mathbb{E}_{z \sim p_z} L(\hat{\theta}, z) < \infty$ was assumed and

$$
\begin{aligned}
&\| \frac{f(g_\theta(z)) - f(g_{\hat{\theta}}(z)) - \left\langle (\theta - \hat{\theta}), \nabla_\theta f(g_\theta(z))|_{\hat{\theta}} \right\rangle}{\|\theta - \hat{\theta}\|} \| \\
&\leq \frac{\|f(g_\theta(z)) - f(g_{\hat{\theta}}(z))\|}{\|\theta - \hat{\theta}\|} + \frac{\| \left\langle (\theta - \hat{\theta}), \nabla_\theta f(g_\theta(z))|_{\hat{\theta}} \right\rangle \|}{\|\theta - \hat{\theta}\|} \\
&\leq L(\hat{\theta}, z) + \frac{\|\theta - \hat{\theta}\| \|\nabla_\theta f(g_\theta(z))|_{\hat{\theta}}\|}{\|\theta - \hat{\theta}\|} \\
&\leq 2L(\hat{\theta}, z)
\end{aligned}
$$

holds, also the domination by a integrable function is shown and the Theorem of dominated convergence can be applied. Therefore, as $\theta \to \hat{\theta}$ term (2.11) converges to 0.

And with this convergence equation (2.10) holds if the right side of the equation is well-defined.

□

The results obtained in this section show that the Earth-Mover distance can be used to learn distributions by neural networks. The next chapter presents an example of this application of the Earth-Mover distance.

Chapter 3

Application of the EM to Generative Adversarial Networks

As described in the last chapter it is reasonable that a neural network learns by using the Earth-Mover distance. One kind of neural network which does so is the Wasserstein Generative Adversarial Network (WGAN). This kind of network is the subject of this part.

This chapter is divided into two sections. Firstly the general concept of Generative Adversarial Networks (GANs) is described mainly citing Goodfellows 'NIPS 2016 Tutorial: Generative Adversarial Networks' [7]. The second section considers the Wasserstein Generative Adversarial Networks itself as stated in Arjovsky et al. 'Wasserstein GAN' [1].

3.1 Generative Adversarial Networks

3.1.1 Structure of a GAN

The Generative Adversarial Networks (GANs) were introduced in 2014 by Goodfellow et al. [6]. As the name already mentions these networks are a type of generative model.

Definition 3.1 (Generative models) *[7, p. 2] A model is called generative if it estimates a real distribution p_r by learning from samples of this distribution. The set of samples the generative model learns from is called training set. The output of a generative model can be either the learned estimation itself or samples of this distribution.*

The output of a GAN can be both an explicit distribution or samples of the distribution. Though the most GANs do the second and this thesis will consider this case in the following. [7, p. 2]

The term adversarial in Generative Adversarial Network results from the fact that a GAN consists of two neural networks competing with each other. These neural networks are the Generator and the Discriminator. [7, p. 17]

Definition 3.2 (Generator) *[7, pp. 17-18] Let $l, n \in \mathbb{N}$ with $l \geq n$ and consider two parameter spaces Δ and Γ. The Generator of a GAN is a neural network*

$$G_\gamma : \mathbb{R}^l \to \mathbb{R}^n$$

for $\gamma \in \Gamma$ such that $G \in C(\mathbb{R}^l)$. $\mathbb{R}^l$ is the space of latent variables and $\mathbb{R}^n$ the space of observed variables.

The cost function of the Generator is noted as $J_G : \Delta \times \Gamma \to \mathbb{R}$.

Definition 3.3 (Discriminator) *[7, pp. 17-18] Let $n \in \mathbb{N}$ and consider two parameter spaces Δ and Γ. The Discriminator of a GAN is a neural network*

$$D_\delta : \mathbb{R}^n \to [0, 1]$$

for $\delta \in \Delta$ such that $D \in C(\mathbb{R}^n)$. $\mathbb{R}^n$ is the space of observed variables.

The related cost function for the Discriminator is $J_D : \Delta \times \Gamma \to \mathbb{R}$.

From this two networks one can construct a GAN in the following way.

Definition 3.4 (Generative Adversarial Networks) *[7, pp. 17-19] Consider $l, n, m \in \mathbb{N}$ with $l \geq n$. Let $z \in \mathbb{R}^l$ be a latent variable with the probability distribution p_z, $x \in \mathbb{R}^n$ be an observed variable and $\delta \in \Delta$, $\gamma \in \Gamma$ network parameters. Moreover, consider $(x_i^{train})_{i=1}^m \subset \mathbb{R}^n$ as training set of samples with the distribution p_r.*

A Generative Adversarial Network (GAN) creates samples $x \sim p_g$ from z by using its Generator G_γ. The Discriminator D_δ of the GAN outputs an estimated probability of a sample being of the training data or generated.

Both neural networks are trained on the set $(x_i^{train})_{i=1}^m \subset \mathbb{R}^n$ and latent samples $z \sim p_z$ to minimize their costs $J_G(\delta, \gamma)$ and $J_D(\delta, \gamma)$. These cost functions must satisfy the following two conditions:

$$D_{\delta^*}(x) = \begin{cases} 1 & \text{if } x \in (x_i^{train})_{i=1}^m \\ 0 & \text{if } x = G_\gamma(z) \end{cases} \quad \text{for } \operatorname*{argmin}_\delta J_D(\delta, \gamma) = \delta^*. \tag{3.1}$$

and

$$D_\delta(G_{\gamma^*}(z)) = 1 \text{ for } \operatorname*{argmin}_\gamma J_G(\delta, \gamma) = \gamma^*, z \sim p_z. \tag{3.2}$$

Note, that there are only a few restrictions on the design of the two networks. Beside the networks being differentiable the only restriction is to have $dim(\mathbb{R}^l) \geq dim(\mathbb{R}^n)$. That is, the dimension of the latent variable space

must be greater equal to the dimension of the space of observed variables to ensure p_g having full support on the space of observed variables. [7, pp. 18-20]

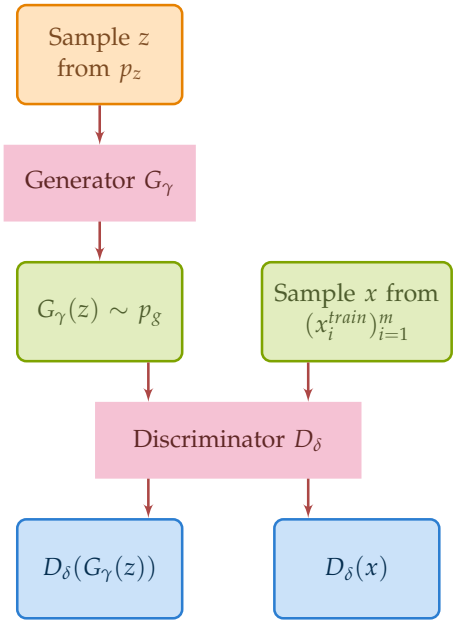

Figure 3.1: Structure of a Generative Adversarial Network

By training a GAN one aims for a Generator which produces samples indistinguishable from the training samples $(x_i^{train})_{i=1}^m$. [7, p. 17]

This leads to three questions:

1. How can the network be trained?
2. Does GAN learning converge to the desired result?
3. What are good choices for the cost functions J_D and J_G?

The general training process of a GAN is introduced in Section 3.1.2. Section 3.1.3 covers the second question and the third question is considered in the Sections 3.1.4 and 3.1.5.

3.1.2 Training of a GAN

Since a GAN is a network consisting of two neural networks, training a GAN means training these neural networks. Both, the Generator and the Discriminator, are trained by using one of the most popular training methods in deep learning: (Minibatch) Stochastic Gradient Descent. [7, p. 20]

For simplicity this method will only be called Stochastic Gradient Descent or SGD as it is done in [5, 8.3.1].

Definition 3.5 (Stochastic Gradient Descent) *[5, p. 294] Let $f \in C(\mathbb{R}^r)$ be a neural network with a loss function $L : C(\mathbb{R}^r) \times \mathbb{R}^r \times \mathbb{R}^s \to \mathbb{R}_+$ and initial parameters θ_0 and let $r, s, k, m \in \mathbb{N}$. Moreover, $(x_i, y_i)_{i=1}^m \subset \mathbb{R}^r \times \mathbb{R}^s$ is the training set with respective solution of f.*

The Stochastic Gradient Descent chooses a minibatch $i_1, ..., i_k$ uniformly at random from $1, .., m$ to calculate

$$g_t = \nabla_{\theta_t} \frac{1}{k} \sum_{j=1}^{k} L(f_{\theta_t}, x_{i_j}, y_{i_j}).$$

It then updates the parameters θ_t by

$$\theta_{t+1} = \theta_t - \varepsilon_t \cdot g_t,$$

where ε_t is the learning rate in step $t \in \mathbb{N}$.

How the learning rate ε_t is designed depends entirely on the SGD variant used. In general one can use any SGD variant for the update of the GAN parameters. [7, p. 20] A range of different variants can be found for example in [17].

Stochastic Gradient Descent is a stochastic approximation of the Gradient Descent Method (see e.g. [5, 4.3]). The expected average gradient of a SGD method equals the average gradient of the Gradient Descent Method as can be seen in Corollary 3.6.

Corollary 3.6 (SGD approximates Gradient Descent) *[5, p. 281] Consider the setting of Definition 3.5. For $i_1, ..., i_k$ chosen uniformly at random from $1, .., m$ the equation*

$$\mathbb{E}_{i_1,...,i_k} \frac{1}{k} \sum_{j=1}^{k} \nabla_{\theta_t} L(f, x_{i_j}, y_{i_j}) = \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta_t} L(f, x_i, y_i)$$

holds.

In contrast to the Gradient Descent method the complexity of a SGD method does not depend on the size of the training set. It does only depend on k, the number of samples chosen of the training set. This property makes SGD useful for large training sets and therefore in neural networks. [5, p. 153]

But how can a SGD method be applied to the Discriminator and the Generator? Since the Discriminator's aim is to distinguish between the generated samples and the training samples $(x_i^{train})_{i=1}^m$, the SGD training data has to

contain both. Therefore, one has to sample two minibatches in each SGD step of the Discriminator. [7, pp. 20-21]

One minibatch contains samples of $(x_i^{train})_{i=1}^m$. The Discriminator's output of these samples has to be compared to the desired result 1. The other minibatch will be filled with samples created by the Generator from $z \sim p_z$. The outputs $D_\delta(G_\gamma(z))$ of the batch of latent variables have to be compared to 0. By comparing the Discriminator's output with the desired output one can calculate a loss for each of the samples. The cost function of the Discriminator J_D is the sum of these losses. In each SGD step its gradient $\nabla_\delta J_D$ is used to update the Discriminator's parameters δ. [7, pp. 20-21]

The training set of the Generator is a minibatch of samples $z \sim p_z$. The loss of these samples is calculated by comparing $D_\delta(G_\gamma(z))$ with the desired value 1. The Generator's cost function J_G is the sum of these losses. Therefore, the gradient of this cost $\nabla_\gamma J_G$ is used to update the parameters of the Generator γ in each SGD step. [7, pp. 20-21]

As described above and shown in Figure 3.2 the training of the Generator involves the Discriminator and vice versa.

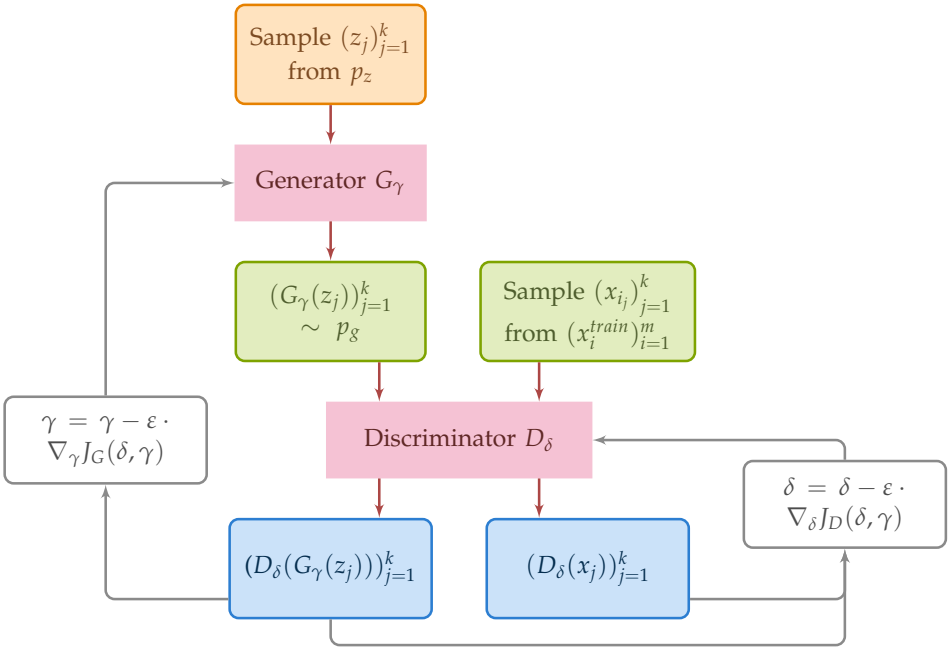

Figure 3.2: Training of a Generative Adversarial Network

It is possible to perform the Gradient Descent for both networks simultaneously, but it is also possible to perform more Gradient Descent steps for one

network than the other. For traditional GAN Goodfellow recommended in late 2016 a simultaneous Gradient Descent with one training step for each player [7, p. 20].

3.1.3 Equilibrium of a trained GAN

The solution of an optimization problem is a minimum. However, in a Generative Adversarial Network there is not only one optimization problem but two, one for each neural network. Additionally the training of each of those two networks involves the parameters of the other network. Consequently it is difficult to separate the two optimization problems from each other in order to find their minimum.

This problem can be avoided when treating the GAN as a game. Instead of the minimum of a typical optimization the solution of a game is a Nash equilibrium. More specifically, since Gradient Descent is a local optimization method, the solution of a GAN trained with SGD is a local Nash equilibrium. [7, p. 18]

Definition 3.7 (Local Nash equilibrium) *[15, Definition 1] Consider the cost functions $J_1 : S_1 \times S_2 \to \mathbb{R}$ and $J_2 : S_1 \times S_2 \to \mathbb{R}$ of two players with respective strategy spaces S_1 and S_2. If $\exists M_1 \subset S_1$, $M_2 \subset S_2$ such that for ${s_1}^* \in M_1$ and ${s_2}^* \in M_2$, it holds that*

$$\begin{aligned} J_1({s_1}^*, {s_2}^*) &\leq J_1(s_1, {s_2}^*) \qquad \forall s_1 \in M_1 \text{ and} \\ J_2({s_1}^*, {s_2}^*) &\leq J_2({s_1}^*, s_2) \qquad \forall s_2 \in M_2, \end{aligned}$$

then $({s_1}^*, {s_2}^*)$ is a local Nash equilibrium.

Considering a GAN as a game is possible by regarding the Generator and the Discriminator as players. Their parameters δ and γ are in this context the strategies of the players. These players aim to minimize their cost functions $J_1(s_1, s_2) := J_G(\delta, \gamma)$ and $J_1(s_1, s_2) := J_D(\delta, \gamma)$. Therefore, the solution of a GAN is a local Nash equilibrium (δ^*, γ^*). [7, p. 18]

Of course the optimal outcome of a GAN training would be a Generator which is producing samples $x \sim p_g = p_r$. In this case G_γ would generate samples of the same distribution as the original data. Thus, one wishes the desired Generator to fulfill the conditions of a local Nash equilibrium. But even if this is the case, it is not sure that the GAN will reach this equilibrium. Although a GAN reaches a local Nash equilibrium if it converges, in general it is not proven that the GAN training converges. [7, p. 34]

Nevertheless, there are convergence proofs for specific GAN variants. One of them is presented in 'GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium' of Heusel et al. [8]. This paper shows that GAN training converges under certain assumptions to a stationary local

Nash equilibrium when using two different learning rates. The main idea is to allow the Discriminator a faster learning than the Generator. This way the Discriminator should learn nearly unaffected by the Generator's adaptions and reaches a local minimum.

3.1.4 The Discriminator's Cost

There are two cost functions in a GAN which highly affect the success and structure of the GAN, the Discriminator's cost J_D and the cost of the Generator J_G. One can interpret the Discriminator as classifier which labels generated data with 0 and original data with 1. Consequently the cost function of the Discriminator is mostly chosen as one of the most popular cost functions: the cross-entropy. [7, p. 21]

This cost function between the probability distribution p of samples x and an estimated probability distribution q of this samples measures the average number of bits required to encode a sample x using q. [14, p. 2]

Definition 3.8 (Cross-entropy) *[14, p. 2] Consider a metric space $\mathcal{X}$ and $P(\mathcal{X})$ as the space of all probability measures defined on $\mathcal{X}$. The cross-entropy between two probability distributions $p, q \in P(\mathcal{X})$ of samples $x \in \mathcal{X}$ is defined as*

$$H(p,q) \coloneqq \mathbb{E}_{x\sim p}(-logq(x)).$$

More specifically, for the Discriminator's cost the cross-entropy for a binary classification with sigmoid output is used. Meaning that in case of the Discriminator the true distribution of the samples is not only one distribution p. Instead, one half of the samples comes from the distribution p_r since it consists of original data. The other half is distributed as p_g since it is generated. Thus, the cost is split up into two partial costs: the cross-entropy between p_r and D_δ and the cross-entropy between p_g and $1 - D_\delta$. [7, p. 21]

Definition 3.9 (Discriminator's cost) *[7, p. 21] Consider the setting of Definition 3.4. The cost function of the Discriminator of a standard GAN is defined as*

$$J_D(\delta,\gamma) \coloneqq -\frac{1}{2}\mathbb{E}_{x\sim p_r} log(D_\delta(x)) - \frac{1}{2}\mathbb{E}_{z\sim p_z} log(1 - D_\delta(G_\gamma(z))). \tag{3.3}$$

With this definition one can find the function $D_\delta(x)$ which minimizes the cost function of the Discriminator and thus what probability the Discriminator with optimal parameter δ^* outputs for every sample x it gets.

Theorem 3.10 (Optimal Discriminator) *[7, pp. 46-47] Consider the setting of Definitions 3.4 and 3.9. Assume that p_r and p_g are non-zero for every x. Then*

$$\underset{D_\delta(x)}{\operatorname{argmin}} J_D(\delta,\gamma) = \frac{p_r(x)}{p_r(x) + p_g(x)}.$$

Proof [7, pp. 46-47] To find the optimal Discriminator for the cost one sets its partial derivative to 0.

$$
\begin{aligned}
& \frac{\partial}{\partial D_\delta(x)} J_D(\delta, \gamma) = 0 \\
\Leftrightarrow \quad & \frac{\partial}{\partial D_\delta(x)} (-\frac{1}{2}\mathbb{E}_{x\sim p_r} log(D_\delta(x)) - \frac{1}{2}\mathbb{E}_{z\sim p_z} log(1 - D_\delta(G(z)))) = 0 \\
\Leftrightarrow \quad & \frac{\partial}{\partial D_\delta(x)} (-\frac{1}{2}\mathbb{E}_{x\sim p_r} log(D_\delta(x)) - \frac{1}{2}\mathbb{E}_{x\sim p_g} log(1 - D_\delta(x))) = 0 \\
\Leftrightarrow \quad & -\frac{1}{2}\frac{p_r(x)}{D_\delta(x)} - \frac{1}{2}\frac{p_g(x)}{1 - D_\delta(x)}(-1) = 0 \\
\Leftrightarrow \quad & \frac{p_r(x)}{D_\delta(x)} = \frac{p_g(x)}{1 - D_\delta(x)} \\
\Leftrightarrow \quad & \frac{1}{D_\delta(x)} = \frac{p_g(x)}{p_r(x)} + 1 \\
\Leftrightarrow \quad & D_\delta(x) = \frac{p_r(x)}{p_g(x) + p_r(x)}
\end{aligned}
$$

Therefore, the optimal Discriminator is $\underset{D_\delta(x)}{\operatorname{argmin}} J_D(\delta, \gamma) = \frac{p_r(x)}{p_g(x)+p_r(x)}$. □

The assumption of p_r and p_g being non-zero in Theorem 3.10 ensures that every sample x can be trained. The behaviour of an untrained sample would be undefined. [7, p. 46]

With this knowledge of an optimal Discriminator one can check if the defined cost function really satisfies the optimality condition (3.1) of the GAN definition 3.4.

Corollary 3.11 (Optimality condition for $D_{\delta*}$) *Consider the setting of Definitions 3.4 and 3.9. The optimal cost of the Discriminator*

$$D_{\delta^*}(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}$$

satisfies the optimality condition (3.1)

$$D_{\delta^*}(x) = \begin{cases} 1 & \text{if } x \in (x_i^{train})_{i=1}^m \\ 0 & \text{if } x = G_\gamma(z) \end{cases}$$

Proof This is caused directly by $x \in (x_i^{train})_{i=1}^m \sim p_r$ and $x = G_\gamma(z) \sim p_g$. □

From Theorem 3.10 is known that $J_D(\delta, \gamma)$ is minimal for parameter δ^* with $D_{\delta^*}(x) = \frac{p_r(x)}{p_r(x)+p_g(x)}$. Thus, the Discriminator in an equilibrium of the GAN is always using this strategy. As discussed in Section 3.1.3 it is not clear

that the GAN will converge. However, since the GAN will converge to a Discriminator with this optimal strategy if it converges, the Discriminator's strategy can be seen as estimate of the optimal strategy $D_{\delta^*}(x) = \frac{p_r(x)}{p_r(x)+p_g(x)}$.

From this ratio-estimate a wide range of divergences can be computed. These values can be used then by the Generator to improve p_g. [7, p. 21]

Keep in mind that the Discriminator is designed as a supervised learning network. As such it does not only have advantages, it also suffers of the disadvantages of this kind of networks as over- and underfitting. Of course over- and underfitting can be avoided in the same way as in the standard supervised learning network by using a lot of training data and a good optimization. [7, p. 21]

3.1.5 The Generator's Cost

While the last section described that the cost function of the Discriminator J_D is usually chosen as the cross-entropy loss of a binary classification, there is a range of choices for the Generator's cost J_G.

Divergences

The Generator aims to minimize the differences between the distribution of the original samples p_r and the distribution of the generated samples p_g. Thus, it appears likely to choose a cost function which measures the divergence between p_r and p_g. Consequently a few probability divergences are introduced before presenting possible choices of cost functions.

A quite intuitive probability divergence is the Total Variation distance. This distance measures the largest difference that two probability distributions have on all Borel subsets.

Definition 3.12 (Total Variation distance) *[1, p. 3] Consider a metric space $\mathcal{X}$ and $P(\mathcal{X})$ as the space of all probability measures defined on $\mathcal{X}$. Denote the set of all Borel subsets of $\mathcal{X}$ with Σ. The Total Variation distance of $p, q \in P(\mathcal{X})$ is defined as*

$$TV(p, q) := \sup_{A \in \Sigma} |p(A) - q(A)|.$$

The Kullback-Leibler divergence between distributions p and q outputs the expected information which is lost when a sample with distribution p is approximated by a model with distribution q. [5, p. 74]

Definition 3.13 (Kullback-Leibler divergence) *[1, p. 3] Consider the metric space $\mathcal{X}$ and $P(\mathcal{X})$ as the space of all probability measures defined on $\mathcal{X}$. Let p, $q \in P(\mathcal{X})$ then the Kullback-Leibler divergence of p and q is*

$$KL(p||q) \coloneqq \mathbb{E}_{x\sim p} log\left(\frac{p(x)}{q(x)}\right).$$

This divergence has several disadvantages. It is not only asymmetric but also becomes infinity if the support of p is not a subset of the support of q. The Jensen-Shannon divergence is based on the Kullback-Leibler divergence. In contrast to the Kullback-Leibler divergence it is symmetric and has only finite values. [14, p. 3]

Definition 3.14 (Jensen-Shannon divergence) *[14, p. 3] Consider a metric space $\mathcal{X}$ and $P(\mathcal{X})$ as the space of all probability measures defined on $\mathcal{X}$. Let p, $q \in P(\mathcal{X})$ then*

$$JS(p,q) \coloneqq \frac{1}{2}KL\left(p||\frac{p+q}{2}\right) + \frac{1}{2}KL\left(q||\frac{p+q}{2}\right)$$

is called the Jensen-Shannon divergence of p and q.

Zero-Sum Game

The simplest choice for the cost of the Generator is to use the negative cost of the Discriminator:

Definition 3.15 (Generator's cost of Zero-Sum Game GAN) *[7, p. 22] Consider the setting of Definitions 3.4 and 3.9. The cost function of the Generator in a GAN of the Zero-Sum Game variant is defined as*

$$J_G \coloneqq -J_D.$$

Considering the Generator as a player which aims to trick the Discriminator motivates this choice of a cost function. In this case the GAN results in a Zero-Sum Game. [7, p. 22]

This kind of game is relatively easy to analyze since it suffices to look on one cost function. While for the Discriminator D_δ the optimal parameters are chosen as

$$\delta^* = \underset{\delta}{\operatorname{argmax}}\, J_G(\delta, \gamma), \tag{3.4}$$

the parameters for the Generator G_γ are chosen as

$$\gamma^* = \underset{\gamma}{\operatorname{argmin}} \max_{\delta} J_G(\delta, \gamma). \tag{3.5}$$

[7, p. 22]

An advantage of the choice of the Generator's cost function as $J_G = -J_D$ is its connection to the Jensen-Shannon divergence which is investigated in Theorem 3.16.

Theorem 3.16 (Connection to the Jensen-Shannon divergence) *[6, Theorem 1] Consider the setting of Definitions 3.4, 3.9 and 3.15. For the optimal Discriminator $D_{\delta^*}(x)$ it holds that*

$$J_G(\delta^*, \gamma) = JS(p_r, p_g) - log(2).$$

Proof [6, Proof of Theorem 1] With Theorem 3.10 the cost function becomes

$$\begin{aligned} J_G(\delta^*, \gamma) &= -J_D(\delta^*, \gamma) \\ &= \frac{1}{2}\mathbb{E}_{x\sim p_r} log(D_{\delta^*}(x)) + \frac{1}{2}\mathbb{E}_{x\sim p_g} log(1 - D_{\delta^*}(x)) \\ &= \frac{1}{2}\mathbb{E}_{x\sim p_r} log\left(\frac{p_r(x)}{p_r(x) + p_g(x)}\right) + \frac{1}{2}\mathbb{E}_{x\sim p_g} log\left(\frac{p_g(x)}{p_r(x) + p_g(x)}\right). \end{aligned}$$

Using Definition 3.13 and 3.14 the cost function can be expressed in terms of the Jensen-Shannon divergence.

$$\begin{aligned} J_G(\delta^*, \gamma) =& \frac{1}{2}\mathbb{E}_{x\sim p_r} log\left(\frac{2p_r(x)}{p_r(x) + p_g(x)}\right) + \frac{1}{2}\mathbb{E}_{x\sim p_g} log\left(\frac{2p_g(x)}{p_r(x) + p_g(x)}\right) \\ & - log(2) \\ =& \frac{1}{2}KL\left(p_r || \frac{p_r + p_g}{2}\right) + \frac{1}{2}KL\left(p_g || \frac{p_r + p_g}{2}\right) - log(2) \\ =& JS(p_r, p_g) - log(2) \end{aligned}$$

□

Therefore, minimizing the Jensen-Shannon divergence between p_r and p_g is equivalent to minimizing $J_G(\delta^*, \gamma)$.

Another advantage of the Generator's cost defined in 3.15 is that a GAN with this cost has a Nash equilibrium with $p_r = p_g$.

Theorem 3.17 (Nash equilibrium of Zero-Sum Game GAN) *[6, Theorem 1] Consider the setting of Definitions 3.4, 3.9 and 3.15. Then (δ^*, γ^*) such that*

$$D_{\delta^*}(x) = \frac{p_r(x)}{p_r(x) + p_g(x)} \text{ and } p_r = p_g$$

is a local Nash equilibrium of a GAN with $J_G = -J_D$.

Proof [6, Proof of Theorem 1] If δ^* and γ^* are such that they minimize $J_D(\delta, \gamma)$ and $J_G(\delta, \gamma)$, respectively, (δ^*, γ^*) is a local Nash equilibrium. From

Theorem 3.10 it is known that $D_{\delta^*} = \frac{p_r(x)}{p_r(x)+p_g(x)}$ is the optimal Discriminator strategy. Thus, it remains to show that for γ^* with $p_g = p_r$ it holds that

$$J_G(\delta^*, \gamma^*) \leq J_G(\delta^*, \gamma) \quad \forall \gamma \in \Gamma.$$

For δ^* the cost function of the Generator becomes

$$\begin{aligned} J_G(\delta^*, \gamma) &= \frac{1}{2}\mathbb{E}_{x\sim p_r} log(D^*_\delta(x)) + \frac{1}{2}\mathbb{E}_{x\sim p_g} log\left(1 - D_\delta * (G_\gamma(z))\right) \\ &= \frac{1}{2}\mathbb{E}_{x\sim p_r} log\left(\frac{p_r(x)}{p_r(x)+p_g(x)}\right) + \frac{1}{2}\mathbb{E}_{x\sim p_g} log\left(\frac{p_g(x)}{p_r(x)+p_g(x)}\right). \end{aligned}$$

For $p_g = p_r$ this resolves to

$$\begin{aligned} J_G(\delta^*, \gamma^*) &= \frac{1}{2}\mathbb{E}_{x\sim p_r} log\left(\frac{p_r(x)}{p_r(x)+p_r(x)}\right) + \frac{1}{2}\mathbb{E}_{x\sim p_r} log\left(\frac{p_r(x)}{p_r(x)+p_r(x)}\right) \\ &= log\left(\frac{1}{2}\right) = -log(2). \end{aligned}$$

From Theorem 3.16 it is known that $J_G(\delta^*, \gamma) = JS(p_r, p_g) - log(2)$. Since the Jensen-Shannon divergence is always non-negative,

$$J_G(\delta^*, \gamma) = JS(p_r, p_g) - log(2) \geq -log(2) = J_G(\delta^*, \gamma^*)$$

follows. □

The downside of the minimax game is that an almost optimal Discriminator causes a vanishing gradient of the Generator's cost as long as the Generator is not optimal. This is demonstrated in the following Corollary which applies since an optimal Discriminator outputs a probability of 0 for every generated sample to be a sample of the training set. [7, p. 22]

Corollary 3.18 (Vanishing Gradient) *Consider the setting of the Definitions 3.4, 3.9 and 3.15. Let δ^* such that*

$$D_{\delta^*}(G_\gamma(z)) = 0 \quad \textit{and} \quad \frac{\partial}{\partial G_\gamma(z)} D_{\delta^*}(G_\gamma(z)) = 0 \quad \forall z \sim p_z.$$

Additionally let $\nabla_\gamma G_\gamma(z) \neq 0$ then

$$\nabla_\gamma J_G(\delta^*, \gamma) = 0.$$

Proof For the optimal Discriminator D_{δ^*} the gradient of the Generator is

$$\begin{aligned}\nabla_\gamma J_G(\delta^*, \gamma) &= \nabla_\gamma \left[\frac{1}{2}\mathbb{E}_{x\sim p_r} log(D_{\delta^*}(x)) + \frac{1}{2}\mathbb{E}_{z\sim p_z} log(1 - D_{\delta^*}(G_\gamma(z)))\right] \\ &= \frac{1}{2}\mathbb{E}_{z\sim p_z} \frac{1}{(1 - D_{\delta^*}(G_\gamma(z)))} \nabla_\gamma(-D_{\delta^*}(G_\gamma(z))) \\ &= -\frac{1}{2}\mathbb{E}_{z\sim p_z} \frac{1}{1} \nabla_\gamma D_{\delta^*}(G_\gamma(z)) \\ &= -\frac{1}{2}\mathbb{E}_{z\sim p_z} \frac{\partial}{\partial G_\gamma(z)} D_{\delta^*}(G_\gamma(z)) \cdot \nabla_\gamma G_\gamma(z).\end{aligned}$$

Then for $\frac{\partial}{\partial G_\gamma(z)} D_{\delta^*}(G_\gamma(z)) = 0$ and $\nabla_\gamma G_\gamma(z) \neq 0$

$$\nabla_\gamma J_G(\delta^*, \gamma) = 0.$$

□

Therefore, it seems not to be advisable to use the cost $J_G(\delta, \gamma) = -J_D(\delta, \gamma)$.

Non-saturating Game

Another approach for the Generator's cost function is to choose the cross-entropy between the distribution of the latent variables z and their output $D_\delta(G_\gamma(z))$. While in the first approach for the Generator's cost one aims for the Discriminator being wrong, one now aims for the Discriminator being tricked by the Generator. [7, p. 23]

In this case the Generator's cost function becomes as follows.

Definition 3.19 (Cost of non-saturating Generator) *[7, p. 22] Consider the setting of Definition 3.4. The cost function of the Generator in a GAN of the non-saturating variant is defined as*

$$J_G := -\frac{1}{2}\mathbb{E}_{z\sim p_z} log(D_\delta(G_\gamma(z))).$$

This cost function does not suffer from the vanishing gradient problem like the cost function of the Zero-Sum Game. In fact this choice of J_G has a large gradient when the Discriminator is almost optimal and thus won't saturate when the Generator performs bad. [7, p. 22]

Although this is an advantage of the non-saturating game over the Zero-Sum Game, the non-saturating game has no theoretical motivation. It is only heuristically motivated. [7, p. 23]

3.2 Wasserstein Generative Adversarial Nets

3.2.1 Motivation for Using the Earth-Mover Distance

Of course one is interested in a cost function of the Generator which is theoretically motivated and does not suffer from the vanishing gradient problem. A strong candidate for such a cost function is the Earth-Mover distance, which was introduced in Chapter 2, since it is relatively weak and thus converges comparatively easy. In the following theorem which is a direct conclusion of [1, Theorem 2] the Earth-Mover distance is compared with several divergences with respect to their strength.

Theorem 3.20 (Weakness of EM) *Consider $\mathcal{X}$ as a compact space with probability space $P(\mathcal{X})$ and $p \in P(\mathcal{X})$. Additionally let $(p_n)_{n\in\mathbb{N}}$ be a sequence of probability distributions in $P(\mathcal{X})$.*

Then $EM(p_n, p) \to 0$ for $n \to \infty$ if at least one of the following statements holds:

- $KL(p_n||p) \to 0$
- $KL(p||p_n) \to 0$
- $JS(p_n, p) \to 0$
- $TV(p_n, p) \to 0$

for $n \to \infty$.

Proof The Theorem's statement is equivalent to the three statements [1, Proof of Theorem 2]:

1. $KL(p_n||p) \to 0$ or $KL(p||p_n) \to 0 \Rightarrow TV(p_n, p) \to 0$
2. $JS(p_n, p) \to 0 \Rightarrow TV(p_n, p) \to 0$
3. $TV(p_n, p) \to 0 \Rightarrow EM(p_n, p) \to 0$.

These three statements can be proven in the following way:

1. For p, $q \in P(\mathcal{X})$ Pinsker's inequality states $TV(p, q) \leq \sqrt{\frac{1}{2}KL(p||q)}$. [3, p. 371] Therefore, it holds that

$$TV(p_n, p) \leq \sqrt{\frac{1}{2}KL(p_n||p)}, \quad TV(p, p_n) \leq \sqrt{\frac{1}{2}KL(p||p_n)}.$$

   And thus $TV(p_n, p) \to 0$ results from $KL(p_n||p) \to 0$ or $KL(p||p_n) \to 0$. [1, Proof of Theorem 2.3.]

2. Since the Total Variation distance satisfies the triangle inequality,

$$TV(p_n, p) \leq TV\left(p_n||\frac{p_n + p}{2}\right) + TV\left(p||\frac{p_n + p}{2}\right)$$

holds. Applying Pinsker's inequality and the definition of the Jensen-Shannon divergence

$$\begin{aligned} TV(p_n, p) &\leq \sqrt{\frac{1}{2} KL\left(p_n || \frac{p_n + p}{2}\right)} + \sqrt{\frac{1}{2} KL\left(p || \frac{p_n + p}{2}\right)} \\ &\leq 2\sqrt{\frac{1}{2} KL\left(p_n || \frac{p_n + p}{2}\right) + \frac{1}{2} KL\left(p || \frac{p_n + p}{2}\right)} \\ &\leq 2\sqrt{JS(p_n, p)}. \end{aligned}$$

follows. Consequently, $TV(p_n, p) \to 0$ if $JS(p_n, p) \to 0$ for $n \to \infty$. [1, Proof of Theorem 2.1.]

3. For every $x, y \in \mathcal{X}$ it is clear that $||x - y|| \leq 1_{x \neq y}\, diam(\mathcal{X})$ where $diam(\mathcal{X}) = sup\{||x - y|| \,|\, x, y \in \mathcal{X}\}$. Therefore, for $p, q \in P(\mathcal{X})$ it also holds that

$$\inf_{\pi \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\pi} ||x - y|| \leq \inf_{\pi \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\pi} 1_{x \neq y}\, diam(\mathcal{X}). \tag{3.6}$$

Since the Total Variation distance can also be characterized as

$$TV(p, q) = \inf_{\pi \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\pi} 1_{x \neq y},$$

inequality (3.6) is equivalent to

$$EM(p, q) \leq TV(p, q)\, diam(\mathcal{X}).$$

[4, Theorem 4]

Thus, if $TV(p_n, p) \to 0$, also $EM(p_n, p) \to 0$ for $n \to \infty$. □

From Theorem 3.20 it is clear that the Earth-Mover distance is at least as weak as the other considered divergences. The following example shows that cases exist where the Earth-Mover distance converges while the other divergences do not. Thus, the Earth-Mover distance is indeed the weakest of the considered divergences.

Example 3.21 *[1, Example 1] Let $Z \sim U(0, 1)$ be a uniformly distributed random variable on the interval $[0, 1]$ and Σ the set of all Borel subsets of $\mathbb{R}^2$.*

In addition consider p_0 as the distribution of $(0, Z) \in \mathbb{R}^2$ and the distribution of $g_\theta(Z) = (\theta, Z) \in \mathbb{R}^2$ as p_θ.

Then for $\theta_t \to 0$ the following statements hold

- *$(p_{\theta_t})_{t \in \mathbb{N}} \to p_0$ with the Earth-Mover distance as*

$$\begin{aligned} EM(p_0, p_\theta) &= \inf_{\pi \in \Pi(p_0, p_\theta)} \mathbb{E}_{(x,y)\sim\pi} ||x - y|| \\ &= \mathbb{E}_{z \sim U(0,1)} ||(0, z) - (\theta, z)|| = |\theta|. \end{aligned}$$

- $(p_{\theta_t})_{t\in\mathbb{N}}$ *does not converge to* p_0 *with the Total Variation distance*

$$\begin{aligned} TV(p_0, p_\theta) &= \sup_{A\in\Sigma} |p_0(A) - p_\theta(A)| \\ &= |p_0(\{(0,z)|z \in [0,1]\}) - p_\theta(\{(0,z)|z \in [0,1]\})| \\ &= \begin{cases} |1-0| = 1 & \text{if } \theta \neq 0 \\ |1-1| = 0 & \text{if } \theta = 0 \end{cases} \end{aligned}$$

- *For* $\theta = 0$

$$KL(p_0||p_\theta) = KL(p_0||p_0) = \mathbb{E}_{x\sim p_0} log\left(\frac{p_0(x)}{p_0(x)}\right) = \mathbb{E}_{x\sim p_0} log(1) = 0$$

results and for $\theta \neq 0$ *the support of* p_θ *is different of the support of* p_0 *such that*

$$KL(p_0||p_\theta) = \mathbb{E}_{x\sim p_0} log\left(\frac{p_0(x)}{p_\theta(x)}\right) = \infty.$$

Therefore, $(p_{\theta_t})_{t\in\mathbb{N}}$ *does not converge to* p_0 *under* $KL(p_0||p_\theta)$*. One can argue analogously for* $KL(p_\theta||p_0)$*.*

- *Under the Jensen-Shannon divergence* $(p_{\theta_t})_{t\in\mathbb{N}} \to p_0$ *does not hold since*

$$\begin{aligned} JS(p_0, p_\theta) &= \frac{1}{2} KL\left(p_0||\frac{p_0+p_\theta}{2}\right) + \frac{1}{2} KL\left(p_\theta||\frac{p_0+p_\theta}{2}\right) \\ &= \frac{1}{2}\mathbb{E}_{x\sim p_0} log\left(\frac{2p_0(x)}{p_0+p_\theta}\right) + \frac{1}{2}\mathbb{E}_{x\sim p_\theta} log\left(\frac{2p_\theta(x)}{p_0+p_\theta}\right) \\ &= \begin{cases} \frac{1}{2} log\left(\frac{2\cdot 1}{1+0}\right) + \frac{1}{2} log\left(\frac{2\cdot 1}{0+1}\right) = log(2) & \text{if } \theta \neq 0 \\ \frac{1}{2} log\left(\frac{2\cdot 1}{1+1}\right) + \frac{1}{2} log\left(\frac{2\cdot 1}{1+1}\right) = 0 & \text{if } \theta = 0 \end{cases} \end{aligned}$$

Another conclusion of the given example is that the statement of Theorem 2.23 would not hold for any of the considered divergences but the Earth-Mover distance. Therefore, using the Earth-Mover distance as the Generator's cost is theoretically justified.

3.2.2 Approximation of Earth-Mover Distance in a WGAN

Thinking back to Chapter 2 it is computationally expensive (in discrete case) if not intractable to calculate a Earth-Mover distance exactly. Same holds for its gradient which is required for the Stochastic Gradient Descent steps if the Earth-Mover distance is chosen as the Generator's cost function.

However, Theorem 2.26 states a way to estimate the gradient of the Earth-Mover distance which can be used in the setting of a GAN.

Corollary 3.22 (Gradient of $EM(p_r, p_g)$) *Consider the setting of Definition 3.4. Let the samples of a Generator be $G_\gamma(z) \sim p_g$ with latent variables $z \sim p_z$. Then there is a function $f^* : \mathbb{R}^n \to \mathbb{R}$ such that*

$$f^* = \underset{||f||_{Lip} \leq 1}{\operatorname{argmax}} \mathbb{E}_{x \sim p_r} f(x) - \mathbb{E}_{z \sim p_z} f(G_\gamma(z)) \tag{3.7}$$

and

$$EM(p_r, p_g) = \mathbb{E}_{x \sim p_r} f^*(x) - \mathbb{E}_{z \sim p_z} f^*(G_\gamma(z)).$$

For this f^* the gradient of the Earth-Mover distance between p_r and p_g is

$$\nabla_\gamma EM(p_r, p_g) = -\mathbb{E}_{z \sim p_z} \nabla_\gamma f^*(G_\gamma(z)).$$

Therefore, it is essential to find the function f^* in order to calculate the desired gradient. Since solving the optimization problem (3.7) exactly is intractable, in a Wasserstein Generative Adversarial Network the solution is roughly approximated by the function with the maximal value among a family of functions $(f_\theta)_{\theta \in \Theta}$:

$$f_{\theta^*} = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathbb{E}_{x \sim p_r} f_\theta(x) - \mathbb{E}_{z \sim p_z} f_\theta(G_\gamma(z)).$$

This is realized by a neural network, called the Critic of the Wasserstein GAN. [1, pp. 6-7]

Definition 3.23 (Critic) *[1, p. 7] The Critic of a Wasserstein GAN is a neural network*

$$f_\theta : \mathbb{R}^n \to \mathbb{R}$$

for $\theta \in \Theta$ with $f_\theta(x)$ is differentiable with respect to x and θ. $\mathbb{R}^n$ is the space of observed variables and Θ compact.

The cost function of the Critic is

$$J_f(\theta, \gamma) := -\mathbb{E}_{x \sim p_r} f_\theta(x) + \mathbb{E}_{z \sim p_z} f_\theta(G_\gamma(z)).$$

The condition of Θ being compact ensures that the Critic f_θ is K-Lipschitz, independent of the actual choice of θ. It holds that

$$K \cdot EM(p_r, p_g) = \sup_{||f||_{Lip} \leq K} \mathbb{E}_{x \sim p_r} f(x) - \mathbb{E}_{z \sim p_z} f(G_\gamma(z)).$$

Consequently the Critic approximates a scaled Earth-Mover distance. [1, pp. 6-7]

The Critic replaces the Discriminator of the standard GAN. Therefore, the Wasserstein GAN consists of the Critic and the Generator with the cost $J_G(\theta, \gamma) = EM(p_r, p_g)$. [1, p. 8]

Definition 3.24 (Wasserstein GAN) *[1, pp. 7-8] Let $l, n, m \in \mathbb{N}$ with $l \geq n$. Consider $z \in \mathbb{R}^l$ as a latent variable with the probability distribution p_z, $x \in \mathbb{R}^n$ as an observed variable and $\theta \in \Theta$, $\gamma \in \Gamma$ as network parameters. Moreover, let $(x_i^{train})_{i=1}^m \subset \mathbb{R}^n$ be a training set of samples with the distribution p_r.*

A Wasserstein Generative Adversarial Network consists of a Critic $f_\theta : \mathbb{R}^n \to \mathbb{R}$ and a Generator $G_\gamma : \mathbb{R}^l \to \mathbb{R}^n$. Both neural networks are trained on the set $(x_i^{train})_{i=1}^m \subset \mathbb{R}^n$ and latent samples $z \sim p_z$ to minimize their costs $J_G(\theta, \gamma)$ and $J_f(\theta, \gamma)$. The cost of the Generator is $J_G(\theta, \gamma) := \mathbb{E}_{x \sim p_r} f_\theta(x) - \mathbb{E}_{z \sim p_z} f_\theta(G_\gamma(z))$.

Thus, a better Critic induces a Generator's cost which is able to approximate $EM(p_r, p_g)$ better. [1, p. 8] This leads to a new interpretation of the two networks. Instead of two separated players which compete with each other the two networks of the WGAN can be considered as a teacher and his student. The better the teacher (Critic) is, the better the student (Generator) will become.

3.2.3 Training of a WGAN

Analogously to the networks of a standard GAN the neural networks of a Wasserstein GAN are trained by step-wise updating their parameters. Again this is done by Stochastic Gradient Descent. The accuracy of the Generator's cost benefits of a good Critic. Consequently the Critic and the Generator are not trained simultaneously in contrast to the standard GAN presented in [7]. Instead the Critic's parameters are updated more often than the Generator's parameters. [1, p. 8]

Additionally one has to ensure that the parameters θ lie indeed in a compact space Θ as it is required in Definition 3.23. This can be done by projecting the calculated parameters onto a compact space Θ. Another way is to clip these parameters. Weight clipping is of course problematic. Up to the clipping parameters it can slowdown the convergence of the parameters or cause vanishing gradients. On the other hand Arjowsky et al. observed good results using clipping and not large differences using projection instead. [1, p. 7]

Thus, the training process can be presented as follows [1, Algorithm 1]:

while γ has not converged **do**

for $t = 0, ..., n_{Critic}$ **do**

Sample $(x_{i_j})_{j=1}^{k}$ from $(x_i^{train})_{i=1}^{m}$

Sample $(z_j)_{j=1}^{k} \sim p_z$

$\theta = \theta - \varepsilon_f \cdot \nabla_\theta [-\frac{1}{k} \sum_{j=1}^{k} f_\theta(x_{i_j}) + \frac{1}{m} \sum_{j=1}^{k} f_\theta(G_\gamma(z_j))]$

Enforce compactness of Θ

end for

Sample $z_{j_{j=1}}^{k} \sim p_z$

$\gamma = \gamma + \varepsilon_G \cdot \nabla_\gamma \frac{1}{k} \sum_{j=1}^{k} f_\theta(G_\gamma(z_j))$

end while

Hereby, ε_f and ε_G are the step length of the Stochastic Gradient Descent of the Critic and Generator, respectively. n_{Critic} is the amount of parameter updates of the Critic for each parameter update of the Generator.

3.2.4 Performance of a WGAN

Arjovsky et al. ran several experiments on the LSUN-Bedrooms dataset where they used a batch size for the training samples x and z of 64, respectively. They chose a constant learning rate of 0.00005 for Critic and Generator and trained the Critic 5 times for each training step of the Generator. To ensure the parameters of the Critic coming from a compact space they clipped each parameter to the interval $[-0.01, 0.01]$. [1, p. 9]

Throughout these experiments they recognized that a reduction of the generator's cost in a WGAN correlates with the visual quality of generated images. Clearly this fact cannot be used to compare different WGAN only by their generator's cost since their costs will differ in their scaling factor. Nevertheless this means that one can trust the learning curves of the cost function and does not need to double check the visual quality of each sample. In contrast both the Generator variants presented in Section 3.1.5 do not have this property. [1, pp. 10-12]

A disadvantage which was observed in [1] is that a WGAN becomes unstable when a momentum based optimizer like ADAM is used on the Critic. This seemed to be caused by the fact that the loss of the Critic is non-stationary. Additionally high learning rates are not advisable, they also induce an unstable training. [1, p. 12]

On the other hand the stability of WGAN is in general better than the stability of the standard GAN training. In standard GAN it can happen that the Generator concentrates on a few modes instead of generating from the full distribution. This mode collapse occurs when the Discriminator performs bad on certain modes and the Generator exploits this. Since it is possible to train the Critic of a WGAN till optimality, this problem cannot occur when it is trained this way. [1, p. 12]

In conclusion it does not only theoretically make sense to use the Earth-Mover distance in the GAN setting. The experiments of Arjovsky et al. also showed that Wasserstein GAN indeed performs well in practice, too.

Chapter 4

Application of Conditional Wasserstein GAN

In the first section of this chapter the conditional Wasserstein GANs are introduced as a variant of Wasserstein GANs. In the second section this variant is applied to the MNIST dataset [11]. The Python code used in the computation is based on a general GAN/WGAN model of Lilian Weng [22] which in turn is an adaption of the code [10].

4.1 Conditional Wasserstein GAN

Many applications of Wasserstein GANs require the use of input data e.g. when the quality of an image should be increased or the next image in a sequence should be found. The Wasserstein GAN introduced in Section 3.2 includes only latent variables z and observed variables x but no variable for the input data. Thus, it can not be applied directly to these kind of problems.

However, a conditional generative model involves not only latent variables z and observed variables x but also conditional variables v. It estimates the conditional probability $p(x|v)$ i.e. the probability of x given a specific input v. [18, pp. 3-4] Therefore, this kind of model can solve the above described problems by choosing v e.g. as the low quality image or the sequence of previous images.

Combining the structure of conditional generative models with Definition 3.24 one gets the conditional Wasserstein GAN. This is a model which extends the Wasserstein GAN design by the use of the conditional variable v in both networks. The Generator is trained to generate samples of the probability $p(x|v)$ which enables the use of the conditional Wasserstein GAN in problems which involve input data.

Definition 4.1 (conditional Wasserstein GAN) *Let $l, n, m, k \in \mathbb{N}$ with $l \geq n$. Consider $z \in \mathbb{R}^l$ as a latent variable with the probability distribution p_z, $x \in \mathbb{R}^n$ as an observed variable and $v \in \mathbb{R}^k$ as conditional variable. Moreover, let $\theta \in \Theta$ and $\gamma \in \Gamma$ be network parameters. $(x_i^{train}, v_i^{train})_{i=1}^m \subset \mathbb{R}^{n \times k}$ is a training set of samples and respective input data with the distribution p_r.*

Then a conditional Wasserstein Generative Adversarial Network consists of a Critic $f_\theta : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$ and a Generator $G_\gamma : \mathbb{R}^l \times \mathbb{R}^k \to \mathbb{R}^n$. Both neural networks are trained on the set $(x_i^{train})_{i=1}^m$, $(v_i^{train})_{i=1}^m$ and latent samples $z \sim p_z$ to minimize their costs $J_G(\theta, \gamma)$ and $J_f(\theta, \gamma)$. The cost of the Generator is

$$J_G(\theta, \gamma) := \mathbb{E}_{(x,v)\sim p_r} f_\theta(x, v) - \mathbb{E}_{v\sim p_r(v), z\sim p_z} f_\theta(G_\gamma(z, v))$$

and the cost of the Critic is

$$J_f(\theta, \gamma) := -\mathbb{E}_{(x,v)\sim p_r} f_\theta(x, v) + \mathbb{E}_{v\sim p_r(v), z\sim p_z} f_\theta(G_\gamma(z, v)).$$

Note, that it is not specified in which way v is inputted in the networks. It can be fed to any layer of the networks, also using it multiple times as input is possible.

4.2 Example of Conditional Wasserstein GAN

In the following an application of a conditional Wasserstein GAN to MNIST [11] (dataset of handwritten digits) is shown. The code regarding this application can be accessed via https://github.com/AlinaLeu/conditional-WGAN. It is based on an implementation of Lilian Weng [22] and thus also on an implementation of Taehoon Kim [10].

The regarded conditional Wasserstein GAN estimates the distribution of the 64×64 greyscale-images of the MNIST dataset given specific columns of the images. The observed variables x are the images themselves and the conditional variables v are chosen as columns of the images. The latent variables z are in this implementation chosen to be uniformly distributed from $[-1, 1]$.

The integration of the conditional variables in the Generator and the Critic is in this implementation done by appending v to z and x, respectively, and then feeding them to the first layer of the networks. This is the only way the conditional variables are involved in the networks.

The results shown in the following are obtained by updating the Critic five times for each training step of the Generator. Moreover, a learning rate of $5 \cdot 10^{-5}$ is used and a batch size of 64. To ensure that the parameters θ lie in a compact space they are clipped to the interval $[-0.01, 0.01]$.

In Figure 4.1 the Generator loss of the first 10000 steps is plotted for different choices of v. Note, that the exact generator loss of different models cannot be compared as they are probably k-Lipschitz for different k. Therefore, only the learning curves can be compared.

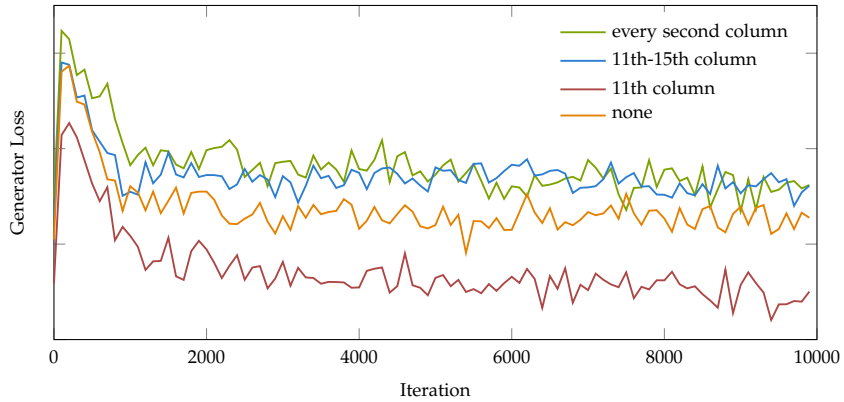

Figure 4.1: Generator loss by iteration steps for different types of conditional variables v

The learning curves of all types of conditional variables have a quite similar shape. Each of the curves shows a good progress in the beginning. From iteration step 1500 on only little progress can be observed while the Generator loss shows strong perturbations.

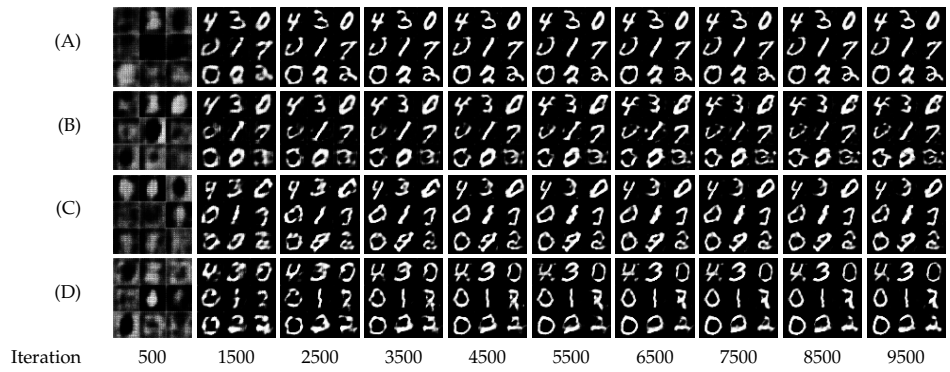

Figure 4.2: Generated image by iteration steps for conditional variable chosen as (A) every second column, (B) the 11th - 15th column, (C) the 11th column, (D) none

Comparing the quality of the images of Figure 4.2 it can be seen that the quality enhancement between step 500 and 1500 is the largest. The quality enhancements between step 1500 and 9500 are comparatively low. Thus,

the enhancement of the image quality in Figure 4.2 and the decrease of the Generator loss in Figure 4.1 seem to follow the same curves. So the correlation between sample quality and Generator's cost which was mentioned in Section 3.2.4 appears indeed in this implementation.

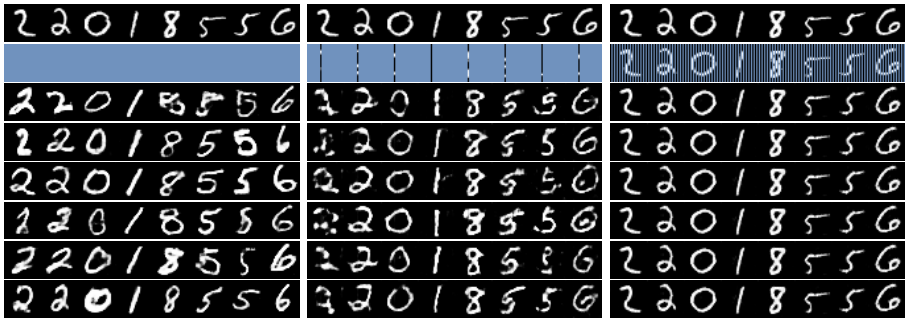

Figure 4.3: (Line 1) MNIST test image with (Line 2) conditional variable and (Line 3-8) generated test images after 100000 training steps

In Figure 4.3 six generated samples of three models are shown which all belong to the same ground-truth of eight images. It can easily be seen that the left model which involves no conditional variable and thus equals the standard Wasserstein GAN generates samples with a lot more variety than the other models. The right model in contrast generates samples which are nearly indistinguishable of each other. This is consistent with the design of the conditional Wasserstein GANs as networks which estimates the distribution $p(x, v)$ by generating samples drawn of this distribution.

Altogether these results suggest that the design of conditional Wasserstein GANs is well adapted for the given problem. In general this might not always be the case since the specific design of the problem greatly affects the success of a network. Specifically the MNIST dataset is as one of the largest datasets with comparatively low-dimensional samples easier to train than the most others.

Appendix A

Abstract & Zusammenfassung

Abstract

This thesis presents the Wasserstein distance which measures the optimal cost of the transport between two probability measures. This distance metrizes the weak convergence in the Wasserstein space and is continuous. Moreover, a Wasserstein variant, called the Earth-Mover distance, is continuous everywhere and differentiable a. e. in parameters θ when measuring the distance between p and p_θ for p_θ being the distribution of a feedforward neural network g_θ.

In addition the application of the Earth-Mover distance in Wasserstein Generative Adversarial Networks (WGANs) is shown. This variant of Generative Adversarial Networks (GANs) is comparatively stable throughout its training process and provides a cost which is strongly correlated with the quality of the generated samples.

This thesis is mainly based on the book 'Optimal Transport, old and new' of Villani [21] and the papers 'NIPS 2016 Tutorial: Generative Adversarial Networks' of Goodfellow [7] and 'Wasserstein GAN' of Arjovsky et al. [1].

Zusammenfassung

In dieser Arbeit geht es um die Wasserstein-Distanz, die die optimalen Transportkosten zwischen zwei Wahrscheinlichkeitsmaßen misst. Diese Distanz ist stetig und eine Konvergenz in der Wasserstein-Distanz ist äquivalent zu einer schwachen Konvergenz im Wassersteinraum. Darüber hinaus ist eine Variante der Wasserstein-Distanz, die Earth-Mover-Distanz, überall stetig und fast überall differenzierbar für die Parameter θ, wenn die Distanz zwischen einer Verteilung p und der Verteilung eines feedforward-Netzes, p_θ, betrachtet wird.

Zusätzlich wird in dieser Arbeit die Anwendung der Earth-Mover Distanz in Wasserstein Generative Adversarial Networks (WGANs) beschrieben. Diese Variante der Generative Adversarial Networks (GANs) ist während des Trainings vergleichsweise stabil und ihre Kostenfunktion korreliert stark mit der Qualität der generierten Daten.

Diese Arbeit basiert auf dem Buch 'Optimal Transport, old and new' von Villani [21] und den Papern 'NIPS 2016 Tutorial: Generative Adversarial Networks' von Goodfellow [7] und 'Wasserstein GAN' von Arjovsky et al. [1].

Bibliography

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *ArXiv e-prints*, January 2017.

[2] Patrick Billingsley. *Convergence of Probability Measures*. Wiley-Interscience, 1999.

[3] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[4] Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review / Revue Internationale de Statistique*, 70(3):419–435, 2002.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[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.

[7] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[9] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Mathematical Programming*. McGraw-Hill, New York, 2 edition, 1995.

[10] Taehoon Kim. Github: Dcgan-tensorflow. https://github.com/carpedm20/DCGAN-tensorflow, 2018. [Online; accessed 22-November-2018].

[11] Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 2009.

[12] Longjie Li, Min Ma, Peng Lei, Xiaoping Wang, and Xiaoyun Chen. A linear approximate algorithm for earth mover's distance with thresholded ground distance. *Mathematical Problems in Engineering*, 2014:1–9, 2014.

[13] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 3 2002.

[14] Frank Nielsen. A family of statistical symmetric divergences based on jensen's inequality. *CoRR*, abs/1009.4004, 2010.

[15] L. J. Ratliff, S. A. Burden, and S. S. Sastry. On the characterization of local nash equilibria in continuous games. *IEEE Transactions on Automatic Control*, 61(8):2301–2307, Aug 2016.

[16] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov 2000.

[17] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[18] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc., 2015.

[19] Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.*, 34(4):66:1–66:11, July 2015.

[20] Allison Toh. Ai podcast: An argument in a bar led to the generative adversarial networks revolutionizing deep learning. https://blogs.nvidia.com/blog/2017/06/08/ai-podcast-an-argument-in-a-bar-led-to-the-generative-adversarial-networks-revolutionizing-deep-learning/, 2017. [Online; accessed 18-November-2018].

[21] Cédric Villani. *Optimal Transport, old and new*. Springer Berlin Heidelberg, June 2008.

[22] Lilian Weng. Github: unified-gan-tensorflow. https://github.com/lilianweng/unified-gan-tensorflow, 2017. [Online; accessed 20-November-2018].

[23] J. Wengenroth. *Wahrscheinlichkeitstheorie*. De Gruyter Lehrbuch. De Gruyter, 2008.