Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)
Физтех-школа прикладной математики и информатики
Кафедра «Интеллектуальные системы»

Колесов Александр Сергеевич

Состязательный метод дообучения нейронной сети в задаче переноса информации

03.04.01 —- Прикладные математика и физика

Выпускная квалификационная работа магистра

Научный руководитель:
к. ф.-м. н. Бахтеев Олег Юрьевич

Москва
2022

Abstract

Inductive transfer learning aims at adapting a pre-trained neural network to target data without access to a source database. Most modern methods suffer from strong regularization between corresponding parameters. An efficient inductive transfer learning method should take enough information from a pre-trained model, but also learn vital patterns from new data. We present the novel technique of transfer learning that is based on optimal transport theory. However, this method requires deep neural networks(DNN) that are strongly 1— Lipschitz continuity functions in according to the theory. To solve this issue, we develop such networks and prove necessary theoretical guarantees for them. We show, that the regularization of the proposed method computes ground-truth the Wasserstein-1 distance between distributions of parameters, while another penalization strategies compute an upper bound. We demonstrate the abilities of the proposed method and compare it with relevant transfer learning techniques.

keywords : Transfer learning, Optimal Transport, Generative models

Contents

Chapter 1

Introduction

Training of a deep neural network (DNN) can be challenging in both small and large data scenarios. The small data has not enough information to reach high performance, whereas the training time is long in the case of large data. Nevertheless, if we have trained a DNN on one domain, which is usually termed as source (6; 53; 29), we can transfer its information to another similar domain, which is referred to as target, by fine-tuning. In other words, having extracted features on source task and initialized parameters of a DNN by pre-trained parameters, one should fine-tune a network to transfer this knowledge to a target task. This approach mitigates issues of different data scenarios and requires relatively smaller training samples to get high accuracy on a target database, rather than learning from scratch (47).

When the amount of training samples for a target task is insufficient, fine-tuning can suffer from Catastrophic forgetting (12) and Negative transfer (47). The first issue is a tendency of forgetting learnt knowledge that can lead to an over-fitting on a target task. The second reflects the fact that not all obtained knowledge from the source is useful for the target domain. For instance, (53) imposes a source-based prior to regularize solutions for a target problem by driving parameters not far away from pre-trained values by L^2 regularization scheme. In (29) , the authors propose feature map regularization, aligning transferable channels in feature maps for most important filters via the attention mechanism.

One of the possible way to overcome problem with negative transfer was discussed in (6). There was found an investigation that spectral components with small singular value of features extracted in high layers aren't transferable. The authors argues, having cut off such spectral components, one can inhibit negative transfer issue. No method focuses on both problems as Catastrophic forgetting and Negative transfer simultaneously. Importantly, these techniques don't use any source's information during training on a target, keeping the source data private and reducing data storage requirements. While the method (?), which is the state-of-the-art in the inductive transfer learning area, utilizes source's labels during training on target database. Supporting data privacy methodology, we will not use this approach for comparing and evaluating its performance in different tasks.

The application optimal transport problems received wide popularity in machine learning. These approaches demonstrated high-quality results in image generation problem (2; 17; 10), domain adaptation (8) and became the core of new generative models (23; 30). The optimal transport problems are usually incorporated in loss function for a DNN , thereby being a Wasserstein distance between distributions . For instance, The Wasserstein-1 distance is the loss function for the most popular generative models as WGAN (2; 17; 33; 34; 41), while Wasserstein-2 distance is used by generative models for finding the best mapping between distributions (23; 30; 24).

Contributions. We propose a novel method of transfer learning with regularization scheme, that is based on the computation of the Wasserstein-1 distance between distribution of parameters from a DNN on source database and distribution of parameters on target domain. The method tries to keep the distribution of filters for target-aimed model similar to that of source-aimed model via the proposed constraints. We prove that such penalization strategy is a lower bound of regularization approaches of any modern transfer learning algorithm (29; 6; 53). We demonstrate the transfer learning algorithm, that is able to compute the ground-truth Wasserstein-1 distance between distributions, thereby providing fast convergence. Also, we encounter with the problem , that contemporary approaches, that use Wasserstein-1 distance as a loss, that indirectly incorporated in the cost function of methods (33; 2; 17; 31; 32; 34) cannot provide 1-Lipschitz continuity for a neural networks. To solve this issue , we develop deep neural networks that are 1-Lipschitz continuous functions, that satisfy theoretical restrictions of the computation of the Wasserstein-1 distance. Moreover, we porve the theorem about lower and upper estimation of the Wasserstein-1 distance and show up at all, that the majority of aforementioned methods cannot sufficiently accurately compute the optimal transport cost. Also, we compare dual surfaces of the methods with the surface of our method and demonstrate, that our technique is able to compute the ground-truth optimal transport cost. Experiments confirm the effectiveness and fast convergence of the novel adversarial technique for transfer learning problem. The proposed method shows state-of-the-art results on several common benchmarks for inductive transfer learning without usage of source's data.

The rest of the paper is organized as follows: In section 2 we state the problem of transfer learning and summarize related works of transfer learning and optimal transport, in Section

3 introduce method with adversarial regularization,proving theorems and lemmas, that theoretically guarantee fast convergence of the proposed method. In Section 4, we demonstrate experimental results and discuss it and conclude the method in the last section.

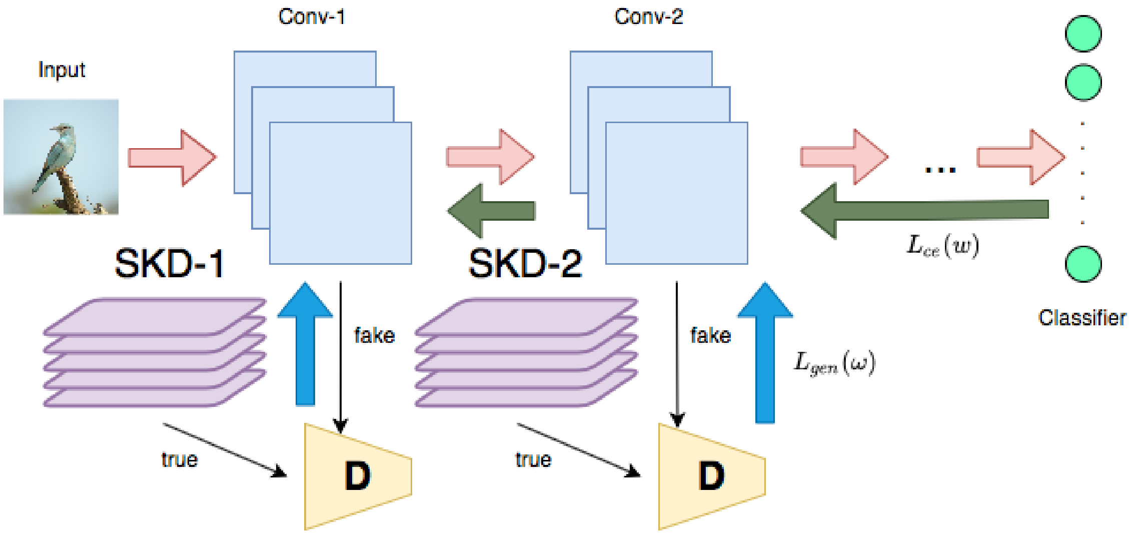


Figure 1.1: The learning process for generator through training steps. SKD is set of source's filters from a correspond layer. D are discriminators that distinguish generator's filters as fake from SKD's filters as true. $L_{ce}(w)$ is cross-entropy loss, while $L_{gen}(w)$ is adversarial generator loss.

Chapter 2

Problem statement

Definition 2.1 The function $\mathrm{f}(x, w) : \mathbb{X} \times \mathbb{W} \rightarrow \mathbb{Y}$ is a Deep Neural Network(DNN), if this function is differentiable by parameters $w \in \mathbb{W}$, where $\mathbb{W}$ is the space of parameters, while $\mathbb{X}$ and $\mathbb{Y}$ are the feature space and the label space correspondingly.

Having defined a neural network, we should introduce the following important concept for the following analysis as source and target datasets.

Definition 2.2 The set of objects and its labels $\mathcal{S} = (\mathrm{x}_i, \mathrm{y}_i)_{i=1}^{n} : \mathrm{x}_i \in \mathbb{X}, \mathrm{y}_i \in \mathbb{Y}$ is a source dataset, if this data are available during the optimization of parameters of a DNN since an initial position $\mathrm{w}_0 \in \mathbb{W}$.

Definition 2.3 The set of objects and its labels $\mathcal{T} = (\mathrm{x}_i, \mathrm{y}_i)_{i=1}^{m} : \mathrm{x}_i \in \mathbb{X}, \mathrm{y}_i \in \mathbb{Y}$ is a target dataset, if this data are available during the optimization of parameters of a DNN since the fixed position $\mathrm{w}_{fix} \in \mathbb{W}$.

In many machine learning scenarios as: Domain Adaptation (8; 51), One-Shot Learning (11), Zero-Shot Learning (43), Transfer Learning (47; 6; 29; 53), we consider a DNN, whose parameters are optimized in one domain (i.e. source dataset) and then evaluate in another one domain (i.e. target dataset). In our paper, we consider the scenario of Transfer Learning problem.

First of all, we start with the statement of the Transfer Learning problem. For example, if we consider ResNet architectures (19) of DNNs that are composed of two main parts. The first part is termed as "encoder": $\mathrm{f}_{enc}(\mathrm{x}, \mathrm{w}) : \mathbb{X} \times \mathbb{W} \rightarrow \mathbb{Q}$, where $\mathbb{Q}$ is a latent space of encoder's features. The second component is referred to as "classifier" : $\mathrm{f}_{cl}(\mathrm{q}, \mathrm{w}) : \mathbb{Q} \times \mathbb{W} \rightarrow \mathbb{Y}$. The main goal of classifier is to decode a hidden features to labels. Thus, the model is the superposition of two models

$$f(x, w) = f_{cl} \odot f_{enc}.$$

Also, it is worth noticing, that we denote model f^{src} if its parameters are optimized in source dataset, while we use f^{tgt} to denote a DNN in target dataset. Therefore, models in source and target domains are given by:

$$f^{src} = f^{src}_{cl} \odot f^{src}_{enc}$$

$$f^{tgt} = f^{tgt}_{cl} \odot f^{tgt}_{enc}$$

The common way for the transfer knowledge from source dataset to target dataset is:

- Optimize parameters f^{src}_{enc} in source dataset $\mathcal{S}$ from an initial values $w_0 \in \mathbb{W}$ to the fixed w_{fix}.
- Optimize parameters f^{tgt}_{enc} in target dataset $\mathcal{T}$ from the fixed value w_{fix}
- Optimize parameters f^{tgt}_{cl} in target dataset $\mathcal{T}$ from the fixed initial position $w^{'}_0$.

It is worth noticing, that having learned the model f^{src}, we remove f^{src}_{cl}, because the space of labels in source $\mathbb{Y}_s$ and in target $\mathbb{X}_s$ are different commonly. As for initial position $w^{'}_0$ for f^{src}_{cl}, one can correspond the fixed vector, that is obtained standard procedure as (18).

The modern algorithms of Transfer Learning (53; 6; 29) is strictly regularization method, thereby they establish direct rigid corresponding between parameters f^{src}_{enc} and f^{tgt}_{enc}. Using such regularization, model is tuned only for the certain target dataset $\mathcal{T}$ and cannot generalize patterns for classification to another ones. One would like to generalize patterns from a source dataset $\mathcal{S}$ to any target dataset. Thus, one can move on to probabilistic view of this problem and consider the posterior distribution $p(w|y, x)$. This posterior distribution reflects the fact, that we know distribution of parameters if we know object and its label from domain. This distributions commonly is defined by the Bayes's formula:

$$p(w|x, y) = \frac{p(y|x, w)p(w)}{\int_{\mathbb{W}} p(y|x, w)p(w)dw}, \tag{2.1}$$

where $p(w)$ is a prior distribution about parameters from a source dataset. However, the denominator of (2.1) is often intractable unless $p(y|x, w)$ and $p(w)$ are conjugate distributions and we can analytically compute this integral. Nevertheless, one can introduce variational distribution q_ϕ with learned parameters $\phi \in \Phi$, where Φ the space of parameters of a neural network q_ϕ. This approximation is an approximation of $p(w|y, x)$. For example, this parameters ϕ might be found as optimal solutions for the following optimization problem:

$$\phi^* = \arg\min_{\phi \in \Phi} \mathbb{D}_{KL}(q_\phi || p(w|y, x)) \tag{2.2}$$

Variational inference (50) is the approach to solve the abovementioned optimization problem with finding variational lower bound $\mathcal{L}(\phi)$

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(w)} \log p(y|x, w) - \mathbb{D}_{KL}(q_\phi(w)||p(w)) \to \max_\phi . \tag{2.3}$$

However, the Kullback-Leybler divergence $\mathbb{D}_{KL}$ is not a metric in probabilistic space $\mathcal{P}(\mathbb{R}^D)$, because this divergence does not satisfy to the triangular inequality and ,moreover, this divergence is not even symmetric. Also, the authors of (52) show, that in case of not overlapping supports of $\mathbb{P}$ and $\mathbb{Q}$ distributions $\mathbb{D}_{KL}$ is not defined. To solve this issues, we consider alternative distance between distribution as Wasserstein-1 $\mathbb{W}_1$. The $\mathbb{W}_1$ is the metric and satisfy the triangular inequality:

$$\forall \mathbb{P}, \mathbb{Q}, \mathbb{S} \in \mathcal{P}_1(\mathbb{R}^D) \quad \Rightarrow \mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \leq \mathbb{W}_1(\mathbb{P}, \mathbb{S}) + \mathbb{W}_1(\mathbb{S}, \mathbb{Q}).$$

Therefore, having substituted the Kullback-Leibler divergence by the Wasserstein-1 distance, one can rewrite the objective for finding the optimal parameters of variational distribution q_ϕ as:

$$\mathcal{L}_W(\phi) = \mathbb{E}_{q_\phi(w)} \log p(y|x, w) - \mathbb{W}_1(q_\phi(w), p(w)) \to \max_\phi . \tag{2.4}$$

In the following sections, we prove the theorem that such regularization scheme is a lower bound of any penalization of current existing Transfer Learning methods (53; 29; 6). Also, we demonstrate, that our penalization scheme compute the ground-truth Wasserstein-1 distance, but not an estimation. Moreover, we demonstrate and prove , how to get such neural networks to accurately compute this optimal transport cost.

In the rest of this section, we review the related works. First of all, we pay our attention to transfer learning approaches and its different regularization methods such as (6; 53; 29). Then, we discuss base theoretical aspects of the optimal transport, the wasserstein-1 distance and application of this in the modern deep learning.

2.1 Background on Transfer Learning

Transfer Learning is the paradigm of machine learning deals with transferring knowledge from source task to a target task. That includes several scenarios : domain adaptation (8) , multi-task learning () and inductive transfer learning (29; 53; 6). Inductive transfer learning is applied in case of there is the label space $\mathbb{Y}_s$ of a source task differs from target label space

$\mathbb{Y}_t$ and labeled target space is available only for evaluating a neural network f^{tgt}, but not on training. Since this situation is more often in the real world, then our investigation focuses on this problem.

The simplest approach of inductive transfer learning is fine-tuning (13) that deals with the following concept. The model f^{src}, which is pre-trained on a source database $\mathcal{S}$, is composed of two main components: feature extractor(encoder) f^{src}_{enc} and classifier f^{src}_{cl}. Then, having another data domain as a target database, one would like to adapt the model to the target database. Then, the fine-tuning approach proposes train new neural network, whose parameters of encoder part are initialized by f^{src}_{enc},by the same loss function with $L2$ regularization of all parameters , that is often called as weight decay. Thus, fine-tuning is aimed to correct feature extractor from source database and train new classifier from scratch on a target database (12).

Nevertheless, using parameters of pre-trained encoder as initialization , which sometimes refers to Starting Point as the Reference (SPAR) (53), for rigorous regularization we inhibit catastrophic forgetting problem, whereas exacerbate negative transfer problem (47). Moreover, there is more detailed investigation the problem that is connected to whether one should transfer the knowledge from a layer on a source task to correspond layer on a target

There are some relevant papers, that is connected to inductive transfer learning, where authors investigated different regularization schemes to accelerate deep transfer learning. For instance, the work(53) offered to apply L^2 norm between parameters of pre-trained feature extractor f^{src}_{enc} and weights of f^{tgt}_{enc} on a target database , while vector of classifier's weights tries to minimize own L^2 norm on a target database, being as weight decay. Thus, avoiding lose the information that was obtained from source task , the method provides model remarkable performance for target tasks and alleviate catastrophic forgetting. Another approach (29) is connected to L^2 regularization of "behavior" of network on a source and target domains. The regularization process constitutes weighted sum by supervised attention mechanism of L^2 norm of difference of layers' outputs from source domain and from target. They consider mapping between output of layers instead of correspond parameters. The key idea is to not equally regularize all correspond feature maps and penalize feature maps in accordance with their importance, that is measured with self-attention mechanism. Thus, the authors of the method tries to tends feature maps of f^{tgt}_{enc} to f^{src}_{enc}, while the second network minimizes the cross-entropy loss simultaneously. The scheme of proposed approach is demonstrated in figure 2.1

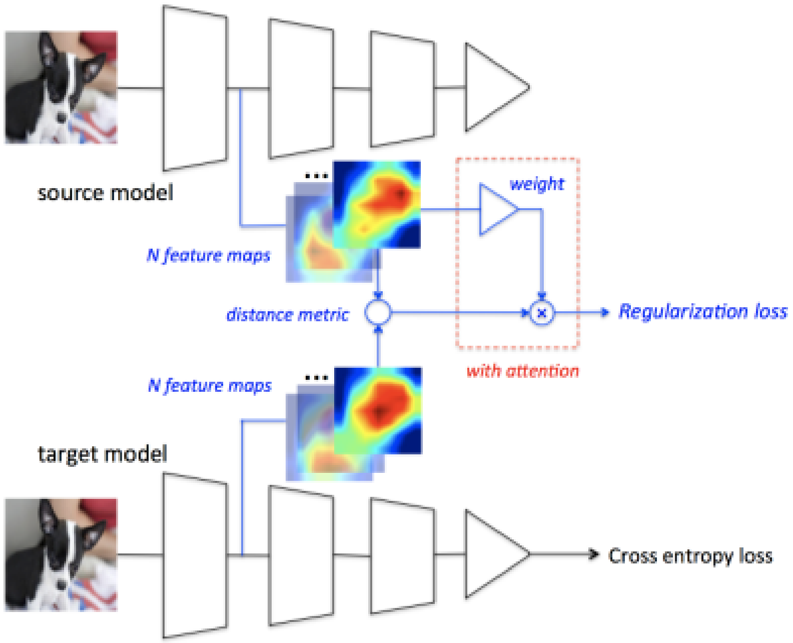


Figure 2.1: The scheme of transfer learning method DELTA (29)

The authors of (6) that spectral components of parameters in high layers with small singular values are not transferable. Thus, they consider outputs of f^{tgt}_{enc} as output of the high layer, then they investigate singular values of this tensor and tries to reset such component, whose singular values are high enough. However, the authors of (6) notice, that one does not consider this regularization scheme without rigid parameter's regularization. Thus, the proposed framework is as the improvement of (53; 29) and their results confirm that. Then, the loss function of this approach is a sum of three loss functions: cross-entropy loss, strict regularization scheme (29) or (53) or $L2$-penalty with the offered loss function that reflects the fact, that singular values of f^{tgt}_{enc} shoudn't be high enough. In figure 2.2, one can look at the scheme of this method.

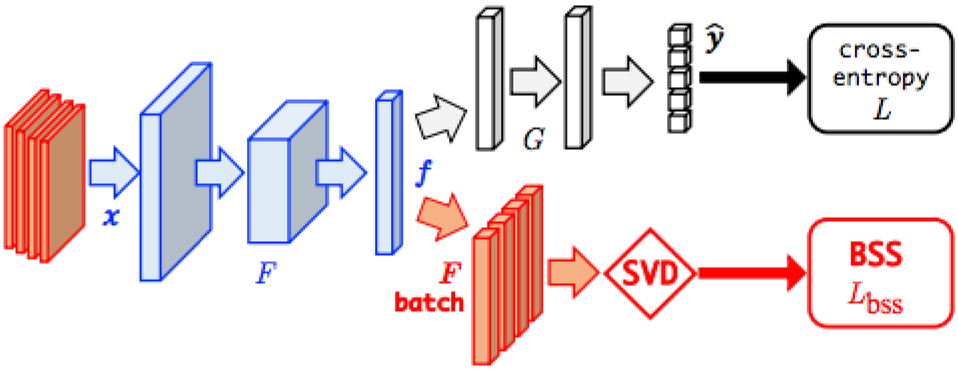


Figure 2.2: The scheme of Batch spectral shrinkage .

However, aforementioned papers consider different connections of feature extractor on a source and on a target database, not taking in consideration classifiers there. Namely, classifier plays an important role for achieving high performance. Unfortunately, there is not transferring of knowledge between task-specific layers above. The authors of article (55) proposed the method that model relationship between categories of source and target databases correspondingly and accelerate deep transfer learning process, obtaining state-of-the-art results on different benchmark datasets. However, this approach utilizes some information from source database as labels to model connections between classifiers. Importantly, such information is often closed by the reason of privacy of data. It is worth noticing, that the pre-trained model is the only tools that is accessed on a target database in case of the data privacy. Thus, not taking in consideration this method that demand an access to source's labels on a target, we propose adversarial framework for deep transfer learning, that outperform other approaches and mitigate base two issues.

Thus, in order to compare our proposed method of transfer learning with another current transfer learning methods, we pay our attention to the following modern methods (53; 29) and the method, that is based on $L2$-penalty.

2.2 Background on Optimal Transport

Primal Formulation. To define primal formulation for finding the Wasserstein-1 distance, we consider two probabilistic measures with finite first moments $\mathbb{P}, \mathbb{Q} \in \mathcal{P}_1(\mathbb{R}^D)$, to provide the existence of finity of the metric. The deterministic optimal transport (36) assumes, that probability mass of one point from $\mathbb{P}$ is completely moved to another one point from $\mathbb{Q}$. This coincidence between points from different measures is described by push-forward operator $T\sharp$ (36), while probability measure $\mathbb{Q}$ is referred to as "target" and is given by $T\sharp\mathbb{P} = \mathbb{Q}$. The deterministic optimal transport defines the Wasserstein-1 distance via deterministic transport plan $T\sharp$ as follow:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \overset{def}{=} \inf_{T\sharp\mathbb{P}=\mathbb{Q}} \int_{\mathbb{R}^D} ||x - T(x)||_2 d\mathbb{P}(x), \tag{2.5}$$

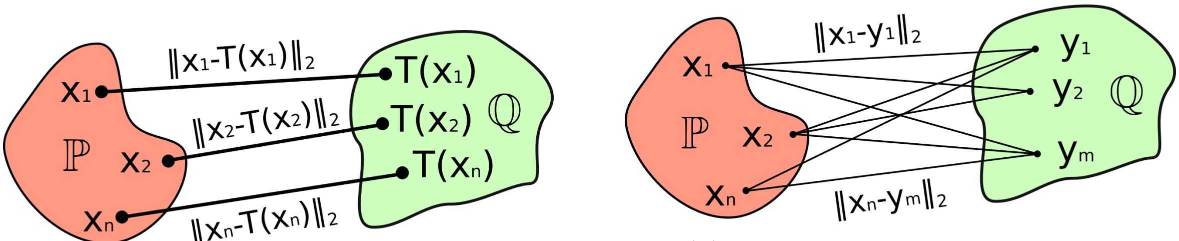
where **inf** is taken over all deterministic transport plans $T : \mathbb{R}^D \to \mathbb{R}^D$ that defines the coincidence between $\mathbb{P}$ and $\mathbb{Q}$.

The aforementioned formulation of the Wasserstein-1 distance has the main disadvantage.

There is not always deterministic coincidence between points from different probability measures. For example, if we consider even two discrete probability measures such that have different support sizes ([36], Remark 2.4), then deterministic transport plan from one to another might be exist, whereas inverse transport plan doesn't. To overcome this obstacle, Kantorovich ([20]) proposed to consider stochastic transport plans $\pi(x, y)$. Thereby, such approach allows probability mass splitting and mass of single point of one distribution might be transferred to several points of another. Since mass of one point spreads between different points of another distribution, then the concept of deterministic transport map is changed to the stochastic transport plan. This stochastic transport plan demonstrate how much probability mass from one point x is splitted between all points y from another distribution. Then, the $\mathbb{W}_1$ distance between $\mathbb{P}$ and $\mathbb{Q}$ is defined by Kantorovich's formulation:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int_{\mathbb{R}^D \times \mathbb{R}^D} ||x - y||_2 d\pi(x, y), \tag{2.6}$$

where $\inf$ is taken over all stochastic transport plans $\pi(x, y)$. The optimal $\pi^* \in \Pi(\mathbb{P}, \mathbb{Q})$ is called the optimal transport plan (OT plan). We call transport ray any non-trivial line $[x, y]$, where $x, y \sim \pi(x, y)$ in according to ([40]).



(a) Monge's OT formulation (2.5).

(b) Kantorovich's OT formulation (2.6)

Figure 2.3: Monge's and Kantorovich's OT fomulations of the Wasserstein-1 distance ($\mathbb{W}_1$).

Dual formulation. The Kantorovich's problem ([20]) is solved by linear programming with $O(n^3)$ algorithm complexity. Nonetheless, such solution is satisfied only for discrete probability measures. In accordance with ([48]) to get a solution for continuous measures, we should take dual formulations of the Kantorovich's problem. Considering two continuous probability measures $\mathbb{P}, \mathbb{Q}$ with finite first moments, the dual formulation with constinuous function f is given by ([49], Th.5.10):

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \sup_{f \oplus g \leq ||\cdot||_2} \int_{\mathbb{R}^D} f(x) d\mathbb{P}(x) + \int_{\mathbb{R}^D} g(y) d\mathbb{Q}(y). \tag{2.7}$$

Using the definition of c-transform (49) , one can alternatively express (2.7) as

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{f} \int_{\mathbb{R}^D} f(x)d\mathbb{P}(x) + \int_{\mathbb{R}^D} f^c(y)d\mathbb{Q}(y), \tag{2.8}$$

where $f^c(y) = \min_{x\in\mathbb{R}^D}[||x-y||_2 - f(x)]$ is the c-transform.

Moving on the space of 1-Lipschitz continuity functions $f : \mathbb{R}^D \to \mathbb{R}$, one can get the final reformulation of the dual Kantorovich's problem in according to (48) as follows:

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{||f||_L\leq 1} \int_{\mathbb{R}^D} f(x)d\mathbb{P}(x) - \int_{\mathbb{R}^D} f(y)d\mathbb{Q}(y). \tag{2.9}$$

The final dual formulation is the most popular, because is used in WGANs (33; 2; 17).

Optimal transport problems. In practice, the $\mathbb{W}_1$ is typically used in the following different three tasks, but not the same:

- Evaluating $\mathbb{W}_1(\mathbb{P},\mathbb{Q})$. Being a metric on $\mathcal{P}_1(\mathbb{R}^D)$, the Wasserstein-1 distance is way to compare probability distributions. In case of discrete distributions, one can compute ground truth Wasserstein-1 distance (32). In the continuous case, we cannot capture all samples, however we can estimate of $\mathbb{W}_1$ by batches of samples from marginal distributions $\mathbb{P}$ and $\mathbb{Q}$. However, such estimation is biased, since for different sizes of batches, we obtain different values of $\mathbb{W}_1$ (37, Fig. 5). In order to calculate unbiased estimation of $\mathbb{W}_1$ is necessary to take points x and y from the optimal transport plan $\pi^*(x,y)$.

- Computing the optimal map T^* or plan π^* . The deterministic transport plan T^* is the good way to find an accordance between samples from distributions. Thus, one can use it as map in problems like domain adaptation (8). Also, the transport plan might be useful for improving generated samples in image-generation tasks (46), thereby there is the better interpolation between $\mathbb{P}$ and $\mathbb{Q}$ distributions. In figure 2.4, one can see the optimal transport map between two distributions, that maps samples from distribution of MNIST samples to the distribution of USPS samples.

- Using the gradient $\partial\mathbb{W}_1(\mathbb{P}_\alpha,\mathbb{Q})/\partial\alpha$ to update generative models. In (33; 17; 1; 3; 41; 31; 28), the authors use implicitly the derivative of $\mathbb{W}_1(\mathbb{P}_\alpha,\mathbb{Q})$ by parameters to update generative models, where $\mathbb{Q}$ is the real-data distribution and $\mathbb{P}_\alpha$ is a learned distribution by the generative model. The learned distribution $\mathbb{P}_\alpha$ is most often generated from fixed hidden s-dimensional ($s < D$) distribution $\mathbb{S}$ by a neural network «generator»

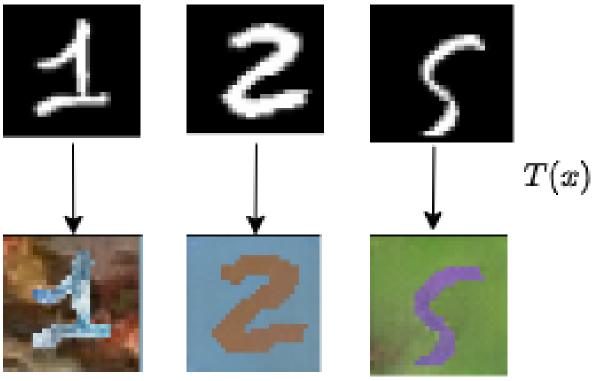


Figure 2.4: transport plan $T(x)$ in Domain Adaptation problem between distributions of samples from MNIST dataset and samples from USPS dataset .

$G_\alpha : \mathbb{R}^s \to \mathbb{R}^D$. The goal is to find parameters α to optimize $\mathbb{W}_1(\mathbb{P}_\alpha, \mathbb{Q})$. The loss function for the generative model is:

$$\mathbb{W}_1(\mathbb{P}_\alpha, \mathbb{Q}) = \int_{\mathbb{R}^s} f^*(G_\alpha(z))d\mathbb{S}(z) + \int_{\mathbb{R}^D} g^*(y)d\mathbb{Q}(y), \tag{2.10}$$

where f^* and g^* are the optimal dual potentials. Then, the derivative of (2.10) is given by :

$$\frac{\partial \mathbb{W}_1(\mathbb{P}_\alpha, \mathbb{Q})}{\partial \alpha} = \int_{\mathbb{R}^s} \lfloor \partial_\alpha G_\alpha(z) \rfloor^T \nabla_G f^*(G_\alpha(z))d\mathbb{S}(z).$$

In practice, the optimal dual potentials f^* and g^* have a parametrization by neural networks f_θ and g_ϕ correspondingly.

In our paper, we consider the final scenario, since the DNN is a generative model.

Chapter 3

Methodology

Firstly, we review about how to get the prior distribution $p(w)$ of parameters of the model f_{enc}^{src} from a source dataset $\mathcal{S}$. We then prove the theorem, that the regularization scheme, which is based on the computation of Wasserstein-1 $\mathbb{W}_1$ distance, is a lower bound for a penalization scheme of any modern transfer learning method (53; 6; 29).

3.1 Prior distribution

In order to transfer information from a source domain $\mathcal{S}$ to a target domain $\mathcal{T}$, it takes prior distribution $p(w)$ of parameters from the encoder of source model f_{enc}^{src}. To get access to this prior distribution, we take a neural network f^{src} and optimize its parameters on the source domain $\mathcal{S}$. Having trained the DNN there, we have an access to parameters of the encoder f_{enc}^{src} in each layer. Since the dependence parameters of a neural network from different layers is weaker , than its connection in one layer (54), then one can represent the prior distribution $p(w)$ of a model f_{enc}^{src} as as a multiplication of independent $p_l(w)$ prior distributions for each layer l:

$$p(w) = \prod_{l=1}^{L} p_l(w).$$

In (4), the authors assume, that a parameter in each layer does not depend on other parameters in this layer. Thus, they represent the prior distribution $p_l(w)$ of a layer l as a product of $p_{i,l}$ prior distributions of each i-th of M parameter of this layer l. Thus, they describe prior distribution of f_{enc}^{src} as follow:

$$p(w) = \prod_{l=1}^{L} \prod_{i=1}^{M} p_{i,l}(w).$$

The authors of (4) optimize an auxiliary variational lower bound that includes of the Kullback-Leybler divergence $\mathbb{D}_{KL}$ between prior distribution $p(w)$ from a source model f_{enc}^{src} and variational distribution $q_\phi(w)$, that describes parameters of f_{enc}^{tgt}. The computation of such divergence requires explicit form of these distributions and its samples. To solve this issue, they make parametrization of the prior distribution $p_{i,l}(w)$ of each weight i-th in each layer l-th as a normal distribution with mean, that equals to this weight and standard deviation as follow:

$$p_{i,l}(w) = \mathcal{N}(w_{i,l}, I_d),$$

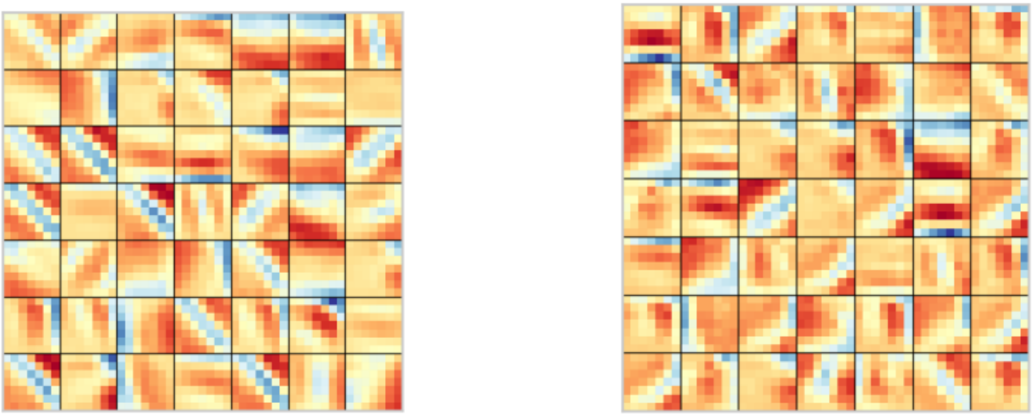
where I identity d-dimensional covariance matrix, while d is the dimensionality of the weight $w_{i,l}$. Undoubtedly, this factorization is sufficiently simple and doesn't probably reflect the ground truth distribution $p(w)$. Thus, this fact is the additional judgment to consider the Wasserstein-1 $\mathbb{W}_1$ distance as the alternative divergence between prior distribution and its approximation.

We follow common concepts (4; 47) and introduce the following definition of parameters from a source model f^{src}.

Definition 3.1 The parameters from a distribution of parameters of l-th model's layer $p_l(w)$ is referred to as SKD (Source Kernel Distribution) of l-th layer.

For example, having selected learned filters from first layer of the optimized model f^{src}_{enc} on a source domain $\mathcal{S}$, we get the group of parameters that correspond to the first group of SKD and is referred to as SKD of first layer in accordance to the aforementioned definition. It is worth noticing, that SKD of the j-th layer constitutes a distribution $p_{j,i}(w)$ of parameters in this layer. The authors of (4; 53; 6; 29; 47) are inclined to assume, that these distributions (SKDs) are a certain prior information for parameters from the correspond layer of f^{tgt}_{enc} on a target database $\mathcal{T}$. In figure 3.1b , one can detach how optimized filters of f^{src}_{enc} , which are obtained by training on MNIST dataset as a source domain, are useful for learning untrained filters of f^{tgt}_{enc} on USPS dataset as a target domain.

As a consequence of that, we optimize parameters of a neural network f^{tgt}_{enc} on a target domain, while the distribution of target's parameters $q_{l,i}(w)$ from l-th layer tends to the distribution $p_{l,i}(w)$ of the source's from the same layer. In other words, we try to make one's parameters closer to other's weights from the correspond SKD. Then, we propose a loss function that consists of two terms. The cross-entropy loss is the first term that is aimed to accurately classify observations from a target database, while the second is penalization, that intends to make one's weights closer to other's.



(a) Learned filters of f_{enc}^{tgt} on USPS dataset.

(b) Parameters from SKD of f_{enc}^{src} on MNIST

Figure 3.1: Learned filters of f_{enc}^{tgt} by (4) with SKD filters of f_{enc}^{src} as a prior distribution

3.2 Wasserstein-1 Regularization.

In this section, we provide theoretical guarantees, that our proposed method is the lower bound of any modern regularization schemes (53; 6; 29) and prove the lemma 3.1, that reflects the fact, that modern transfer learning methods compute biased estimation of divergence between distribution of parameters on a source domain and a target correspondingly.

As was mentioned before in section 2, each current transfer learning approach (29; 6; 53; 55) is viewed as a optimization problem by parameters w of f_{enc}^{tgt}:

$$\min_{w \in \mathbb{W}} \mathcal{L}(f_{cl}^{tgt}(f_{enc}^{tgt}(x_i, w_i)), y_i) + \lambda\Omega(\cdot), \tag{3.1}$$

where $\mathcal{L}(\cdot, \cdot)$ is a loss function as usually cross-entropy loss is used, whereas $\Omega(\cdot)$ is a regularization scheme for parameters (53; 55; 6) or feature maps (29) of models. We briefly review these methods as follows:

- L2-penalty :

  $$\Omega(\mathrm{w}) = \alpha||\mathrm{w}^{tgt}||_2^2$$

  where α - hyperparameter

- L2-SP : The key concept of this method is to strictly tend parameters of f_{enc}^{tgt} to correspond parameters of f_{enc}^{src}. Thus, parameters of the second model inherit patterns and details (in figure 3.1b) of correspond parameters from the first. Thus, the general optimization problem for this method one can view as: f_{enc}^{tgt} to f_{enc}^{src} by $L2$ metric,

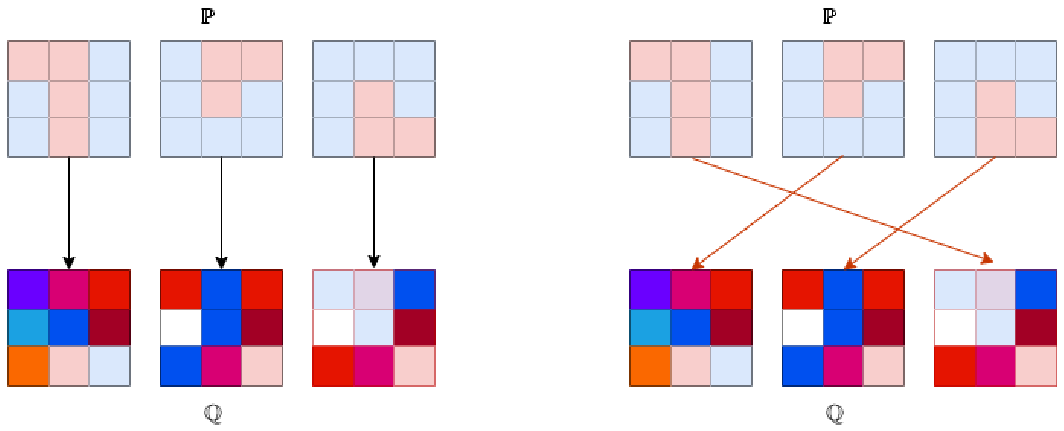
$$\Omega(\mathrm{w}) = \beta||\mathrm{w}_{enc}^{tgt} - \mathrm{w}_{enc}^{src}||_2^2 + \alpha||\mathrm{w}_{cl}^{tgt}||_2^2.$$

Nevertheless, the authors of (6) notice, that such regularization scheme possess strict connection between correspond parameters, thereby not allowing flexibility of model f_{enc}^{tgt}. Adapting such rigorously SKD parameters from a source domain, parameters of f_{enc}^{tgt} are forced to move to correspond parameters f_{enc}^{src}, not trying to generalize this prior information or pick up another parameter from the layer. This issue is referred to as "Catastrophic forgetting" (22; 12). To overcome this problem in case of transfer learning problem, the authors of (6) propose regularization scheme, that reset parameters, whose singular values is sufficiently high. Thus, they propose add their penalization scheme, which is based on linear algebra, to existing methods (53; 29). Moreover, they experimentally shows up at all improvement their scheme of the exception of modern transfer learning techniques.

- DELTA : The final method (29) considers strict penalization between features of f_{enc}^{tgt} and f_{enc}^{src} instead of their parameters. Concretely, they introduce the concept of FM_j^{src} and FM_j^{tgt}, that is a vector of outputs of j-th layer of f^{src} and f^{tgt} correspondingly. Undoubtedly, each feature map of a layer is function of a certain parameters from the same layer (45). Also, it is worth noticing, that method not equally strictly tends ones parameters to another. The authors believe, that there are more important and valuable parameters, than another ones. To solve this issue, they introduce a constant β. This constant reflects the fact, that if parameter or its correspond feature map is valuable , then having reset value of the parameter to zero , it should lead low performance of the network. Thus, they define the following optimization problem for finding optimal parameters of f^{tgt} as:

$$\Omega(\mathrm{w}) = \beta||\mathrm{FM}_{enc}^{tgt}(\mathrm{w}_{enc}^{tgt}) - \mathrm{FM}_{enc}^{src}(\mathrm{w}_{enc}^{src})||_2^2 + \alpha||\mathrm{w}_{cl}^{tgt}||_2^2.$$

Nonetheless, The penalization schemes (5; 29; 53; 55) consider strict corresponding between parameters, trying rigorously and strictly one parameter of f_{enc}^{src} tend to the same parameters of f_{enc}^{tgt}. Thus, while considering a certain layer of f_{enc}^{tgt}, we are forced to approximate parameters by the correspond parameter of the model f_{enc}^{src} without any choice of other parameters from the same layer. As was mentioned above, these penalizations scheme can



(a) Transport plan of Transfer learning between distributions

(b) Optimal transport plan between distributions of parameters

Figure 3.2: Strict transport maps of transfer learning approaches and Optimal transport map

lead to catastrophic forgetting (6).Moreover, considering the set of parameters of f^{tgt}_{enc} as a distribution of this parameters $\mathbb{Q}$ and f^{src}_{enc} as a distribution $\mathbb{P}$. From the point of view of minimization the Wasserstein-1 distance, the abovementioned penalization schemes possess strict connection between samples, thereby they provide the only one trivial identity transport plan T_{id} between samples of these distributions. In figure 3.2a, such transport plan of transfer learning approaches is depicted, while the optimal transport plan between the same distributions is depicted in figure 3.2b.

Once considered these figures, one can see, that the transport plan of transfer learning methods is a map, but not optimal. Thus, if we consider penalization schemes as distance between distributions of parameters $\mathbb{P}$ and $\mathbb{Q}$ respectively in accordance with (4; 53; 6; 29; 47), then one can formulate the following important lemma for the following analysis of the proposed method. Importantly, since parameters has the only one coincidence, while we consider the transfer learning problem, then T_{id} is a transport map of the coincidence.

Lemma 3.1 Let $\mathbb{P}$ and $\mathbb{Q}$ from $\mathcal{P}_1(\mathbb{R}^D)$. The identity transport plan T_{id} is transport map for transfer learning methods with $L2$ penalization schemes. We denote $\hat{\mathbb{W}}_1$ as the estimation of Wasserstein-1 distance by transport plan T_{id}. Then, there is the following estimation:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \leq \hat{\mathbb{W}}_1(\mathbb{P}, \mathbb{Q})$$

[Proof of Lemma 3.1] By the definition of the Wasserstein-1 distance $\mathbb{W}_1$ in accordance with (2.6), to get accurate distance we need take $\inf$ over all measurable plans $\pi \in \Pi(\mathbb{P}, \mathbb{Q})$. Having only one transport plan $\pi = T_{id}$, we can calculate unbiased estimation of W_1. Thus, the formulation with one plan T_{id} satisfies to the following inequality:

$$\hat{W}_1(\mathbb{P}, \mathbb{Q}) = \int_{\mathbb{R}^D \times \mathbb{R}^D} ||x - y||_2 dT_{id}(x, y) \geq \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int_{\mathbb{R}^D \times \mathbb{R}^D} ||x - y||_2 d\pi(x, y) = W_1(\mathbb{P}, \mathbb{Q})$$

3.3 1-lipschitz networks

The dual formulation of (2.9) constraints functions by the class of 1-Lipschitz continuity functions, allowing samples be from marginal distributions $\mathbb{P}$ and $\mathbb{Q}$ correspondingly with the exception of (2.6), where samples should be from the optimal transport plan. This formulation is often used as a loss function of generative models (2; 17; 33; 1; 35; 34; 39; 14; 10). The main challenge of these approaches is to enforce 1-Lipschitz continuity for a neural network $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}$ (17; 35; 33; 2; 3) or accurately compute c-transfrom decomposition (32; 31; 34). There are many approaches to provide this property for a neural network. We give brief description and comparisons of these methods.

The most popular approaches are based on (2.9). The main challenge of these methods is to enforce 1-Lipschitz constraint for the function f.

$\lfloor$Lip-Clip$\rfloor$ The authors of (3) propose to approximate f by a neural network f_θ, whose space of parameters is a compact. As pointed out by (3, §3), the main practical issue is to tune the boundary of a compact set. If the boundary is small enough, then there is a problem of vanishing gradients, because parameters not far away from the initial position. If the boundary is sufficiently large, it is difficult to get optimal critic f^*_θ, because it can request a long time for convergence of critic's parameters. Then, the objective is given by:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \approx \sup_{\theta \in \Theta} [\int_{\mathbb{R}^D} f_\theta(x) d\mathbb{P}(x) - \int_{\mathbb{R}^D} f_\theta(y) d\mathbb{Q}(y)], \tag{3.2}$$

where we update parameters θ with stochastic gradient descent (SGD) over random mini-batches from $\mathbb{P}$ and $\mathbb{Q}$.

$\lfloor$Lip-GP$\rfloor$ The authors of (17) prove, the norm of optimal f^* dual potential $||\nabla_x f^*(x)||_2$ equals to 1 in transport rays. Thus, they regularize dual potential, that is parametrized by feed-forward network f_θ, for violating abovementioned constraint by the "Gradient penalty" method as:

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{\theta}[\int_{\mathbb{R}^D} f_\theta(x)d\mathbb{P}(x) - \int_{\mathbb{R}^D} f_\theta(y)d\mathbb{Q}(y)] + \lambda(||\nabla_t f_\theta(t)||_2 - 1)^2 \tag{3.3}$$

where t is a sample from transport ray, which is defined by $t = xr + (1-r)y$, where r is sampled from standard uniform distribution $\mathcal{U}(0,1)$, between samples x from $\mathbb{P}(x)$ and y from $\mathbb{Q}(y)$.

$\lfloor$Lip-LP$\rfloor$ In (35), the authors propose another penalization scheme, whose main goal is to imporve stability of WGAN's training.

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{\theta}[\int_{\mathbb{R}^D} f_\theta(x)d\mathbb{P}(x) - \int_{\mathbb{R}^D} f_\theta(y)d\mathbb{Q}(y)] + \lambda \max(0, ||\nabla_t f_\theta(t)||_2 - 1)^2 \tag{3.4}$$

They show that proposed regularization has less values, than the "Gradient penalty" constraint, thereby it provides stability of training.

$\lfloor$Lip-SN$\rfloor$ The authors of (33) claim, that (17; 35) cannot globally guarantee correspond constraints. They show, that Lipschitz's norm of a network is bounded by spectral norm of its gradient, proposing another method to enforce 1-Lipschitz continuity for f_θ everywhere. After weight normalization by its spectral norms, f_θ satisfies to condition $||f_\theta||_L \leq 1$.

$\lfloor$Lip-SO$\rfloor$ The authors of (1) show, that a neural network with spectral normalization (33) loses in expressive power, enforcing 1-Lipschitz continuity. They claim, that expressive 1-Lipschitz network must satisfy gradient norm preservation. Proving that gradient norm preservation corresponds to the orthonormal weight matrix, they show that a neural network f_θ with such weight matrix and GroupSort activations (7) is a universal approximation for any 1-Lipschitz function. Alternative strategy to globally enforce 1-Lipschitz continuity is (1). They propose method whereby all singular values of weight matrices equal to 1, whereas (33) restricts the largest singular value. They show, that a neural network f_θ with such weight matrix and GroupSort activations (7) is a universal approximation for any 1-Lipschitz function.

⌊Reg⌋ In (39; 14; 10), the authors solve unconstrained optimization problem for (2.9)

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{\theta,\phi}[\int_{\mathbb{R}^D} f_\theta(x)d\mathbb{P}(x) + \int_{\mathbb{R}^D} g_\phi(y)d\mathbb{Q}(y) - \mathcal{R}(f_\theta, g_\phi)], \tag{3.5}$$

where $\mathcal{R}(\cdot,\cdot)$ is $L2$- regularization term and in according to (41) is defined as:

$$\mathcal{R}(f,g) = -\frac{1}{4\epsilon}\max(0, f(x)+g(y)-||x-y||_2)^2,$$

where ϵ is. a hyper parameter of the method.

⌊MM:B⌋ The authors of (31) consider the dual formulation (2.8) with the inner problem as:

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{\theta}[\int_{\mathbb{R}^D} f_\theta(x)d\mathbb{P}(x) - \int_{\mathbb{R}^D} \min_{x\in\mathbb{R}^D}[||x-y||_2 - f_\theta(y)]d\mathbb{Q}(y)]. \tag{3.6}$$

However, instead of the minimization of the c-transform functional over all $x \in \mathbb{R}^D$, they restrict to the current mini-batch from $\mathbb{P}$.

⌊MM⌋ The authors of (34) introduce a minimax reformulation of (2.8). It includes a function "mover" $H : \mathbb{R}^D \to \mathbb{R}^D$, which learns to predict the result of c-transform for discriminator f_θ:

$$H_\psi(y) = \arg\min_{x\in\mathbb{R}^D}\{||x-y||_2 - f_\theta(x)\} \Rightarrow f^c(y) = ||y - H_\psi(y)||_2 - f_\theta(H_\psi(y)).$$

Thus, the computation of the cost turns to mini-max problem with alternative gradient optimisation strategy:

$$W_1(\mathbb{P},\mathbb{Q}) = \max_{\theta}[\int_{\mathbb{R}^D} f_\theta(y)d\mathbb{Q}(y) - \min_{\psi}\int_{\mathbb{R}^D}\{f_\theta(H_\psi(x)) + ||H_\psi(x) - x||_2\}d\mathbb{P}(x)]. \tag{3.7}$$

In accordance with (25, Lemma 4), the optimal "mover" H^* is the OT plan from $\mathbb{P}$ to $\mathbb{Q}$.

We define simple and strong 1-Lipschitz continuity for a function $f : \mathbb{R}^D \to \mathbb{R}$ as follows.

Definition 3.2 The function $f : \mathbb{R}^D \to \mathbb{R}$ is called 1-Lipschitz function, if :

$$\forall x, y \in \mathbb{R}^D \to \frac{|f(x) - f(y)|}{||x-y||_2} \leq 1,$$

or equivalently:

$$\forall x \in \mathbb{R}^D \to ||\nabla_x f(x)||_2 \leq 1$$

Definition 3.3 The function $f : \mathbb{R}^D \to \mathbb{R}$ is called strong 1-Lipschitz function, if :

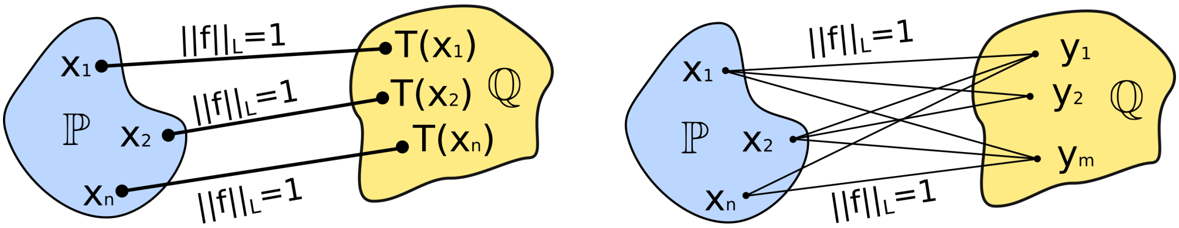
$$\forall x, y \in \mathbb{R}^D \to \frac{|f(x) - f(y)|}{||x - y||_2} = 1,$$

or equivalently:

$$\forall x \in \mathbb{R}^D \to ||\nabla_x f(x)||_2 = 1$$

We denote such class of functions as $f \in Lip_1$ or $||f||_L = 1$.

While aforementioned approaches provide simple 1-Lipschitz continuity for the function f on all space $\mathbb{R}^D$, the authors of (40, Lemma 3.2) argue, that dual potential f function should be strongly 1-Lipschitz on transport rays π between samples from distributions $\mathbb{P}$ and $\mathbb{Q}$. Nevertheless, the aforementioned algorithms cannot give the opportunity to approximate such functions by neural network. To solve this issue, we develop such neural networks that satisfy to the property of strongly 1-Lipschitz continuity everywhere. In figure 3.3a, the deterministic transport map is depicted. In according to (40, Lemma 3.2), the fucntion f should be strongly 1-Lipschitz on transport rays and simple 1-Lipschitz outside of this lines. The same situation is depicted for stochastic optimal transport plans in figure 3.3b.



(a) strongly 1 Lipschitzness on transport rays in deterministic OT

(b) Strongly 1 Lipschitzness on transport rays in stochastic OT

Figure 3.3: Demonstration of lemma 3.2 from (40)

To develop such networks, let's pay our attention to one-layer dense network $f_\theta : \mathbb{R}^D \to \mathbb{R}$, that is $f_\theta(x) = Wx$, where W weight matrix of the network. In according to the definition, to provide strong 1-lipschitz continuity, it takes the norm of gradient of $f(x)$ by inputs should be 1 everywhere.

$$||\nabla_x f(x)||_2 = ||\nabla_x Wx||_2 = ||W||_2 = 1 \tag{3.8}$$

Obviously, having normalized weight matrix by its norm, one can provide strongly 1-Lipschitz continuity for function f. Nonetheless, such network has poor generalization, being a linear function. To overcome this problem, one need add non-linearity behaviour to this model. In accordance with (7), many popular activation functions are not strong 1-Lipschitz continuity , that is why the authors propose different algorithmic approaches as trees and sorts of outputs to reach the desired property. Paying attention to ideas of (7), we develop the similar framework, that provides non-linearity behaviour for the network. We take k (where $k < D$) aforementioned linear strong 1-Lipschitz continuity networks f_θ. Thus, we create a function $f_\theta^{'} : \mathbb{R}^D \rightarrow \mathbb{R}^k$, whose each neuron output is strong 1-Lipschitz continuity function. Proving the lemma 3.3 about maximum and minimum of strong 1-Lipschitz functions, we provide that any mini-max operation upon these k returns one output, that is strong 1-Lipschitz continuity. Then, We consider a mini-max tree upon outputs of the linear layer as the activation function.

Having solved the problem of non-linearity, one has to overcome problem of poor generalization. To reach this destination, we prove three lemmas,that the following functions have strongly 1-Lipschitz continuity.

Lemma 3.2 Let $f, g \in Lip_1$. Then , the function $t : \mathbb{R}^{2D} \rightarrow \mathbb{R}$ that is defined as:

$$t(x, y) = \alpha f(x) + \beta g(y),$$

is strongly 1-Lipschitz, while $\alpha = \sqrt{1 - \beta^2}$

[Proof of Lemma 3.2] 1) In according to the definition 3.3, we should prove that the gradient's norm of $t(x, y)$ equals to 1 everywhere. For this, we write gradients of f and g functions by its inputs as

$$\nabla_x f(x) = [\frac{\partial f}{\partial x_1} \quad ... \quad \frac{\partial f}{\partial x_n}]_{1xn} \quad , \nabla_y g(y) = [\frac{\partial g}{\partial y_1} \quad ... \quad \frac{\partial g}{\partial y_n}]_{1xn}.$$

2) Then, one can represent gradient's norm of $t(x, y)$ by x and y inputs as:

$$\nabla_{x,y} t(x, y) = [\frac{\partial t}{\partial x_1} \quad ... \quad \frac{\partial t}{\partial x_n} \quad \frac{\partial t}{\partial y_1} \quad ... \quad \frac{\partial t}{\partial y_n}]_{1x2n}.$$

Hence, in accordance with the representation of $t(x, y)$ in the statement of the lemma as the sum, we rewrite the gradient's norm as:

$$\nabla_{x,y} t(x, y) = [\frac{\partial(\alpha f + \beta g)}{\partial x_1} \quad ... \quad \frac{\partial(\alpha f + \beta g)}{\partial x_n} \quad \frac{\partial(\alpha f + \beta g)}{\partial y_1} \quad ... \quad \frac{\partial(\alpha f + \beta g)}{\partial y_n}]_{1x2n}.$$

Since f is differentiable only by x and g is defferentiable by y, we write:

$$\nabla_{x,y} t(x,y) = [\frac{\partial(\alpha f)}{\partial x_1} \quad ... \quad \frac{\partial(\alpha f)}{\partial x_n} \quad \frac{\partial(\beta g)}{\partial y_1} \quad ... \quad \frac{\partial(\beta g)}{\partial y_n}]_{1x2n}$$

4) Therefore, we write the gradient's norm of $t(x,y)$ and set it to one , in order to satisfy to the definition 3.3. Using the strong 1-Lipschitz continuity of f and g, we get :

$$||\nabla t(x,y)||^2 = \alpha^2(\sum_i [\frac{\partial f}{\partial x_i}]^2) + \beta^2(\sum_i [\frac{\partial g}{\partial y_i}]^2) = \sqrt{\alpha^2 + \beta^2} = 1$$

Lemma 3.3 Let $f, g \in Lip_1$. Then, the following fucntions are strongly 1-Lipschitz too:

$$\mathsf{max}(f,g), \quad \mathsf{min}(f,g)$$

[Proof of Lemma 3.3] 1) To provide, that $\mathsf{max}(f,g)$ is strongly 1 Lipschitz continuity function, we should show, that the derivative of this function equals to 1 everywhere in accordance with 3.3. Consider the maximum of two stronglty Lipschitz functions:

$$\nabla_x \mathsf{max}(f(x), g(x)) = \begin{cases} \nabla_x f(x) & : f(x) > g(x) \\ \nabla_x g(x) & : f(x) < g(x) \end{cases}$$

In accordance with 1-Lipschitz continuity of f and g , obviously, that $\mathsf{max}(f,g)$ is strongly 1-lipschitz continuity too. The same situation with the $\mathsf{min}(f,g)$.

Having created neural network with the desired property as strong 1-Lipschitz continuity, one can compare computation abilities between the proposed method and another aforementioned approaches. In the experimental session, we compare the Wasserstein-1 distance that is calculated by the modern methodologies against our method. It is worth noticing, that there are a lot of approaches (32; 44; 37), that theoretically and experimentally prove, that $\mathbb{W}_{\not{k}}$ is not the general cost of WGANs. For instance, the authors of (37) notice, that the Wasserstein-1 distance between two discrete distributions is changed with changing of the batch-size of samples, while models and another parameters of the experiment are the same. Also, the authors of (32) demonstrated, that methods, which are based on dual formulation (2.9), are not correct in computation of W_1 distance. Moreover, we prove the following lemma, thereby demonstrate lower and upper bounds for the Wasserstein-1 distance, which are not connected with the metric. In experimental session, we show, that there are lot of methods, that cannot compute even one of these bounds, not saying about the accurate W_1 distance. However, we introduce the following definition before the proof of lemma.

Lemma 3.4 (Upper and lower bounds for $\mathbb{W}_1$) For $\mathbb{P}, \mathbb{Q}$ it holds $\mathcal{E}^2(\mathbb{P}, \mathbb{Q}) \leq \mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \leq \mathcal{I}(\mathbb{P}, \mathbb{Q})$, where $\mathcal{I}(\mathbb{P}, \mathbb{Q}) \overset{def}{=} \int_{\mathbb{R}^D} \int_{\mathbb{R}^D} \|x - y\|_2 d\mathbb{P}(x) d\mathbb{Q}(y)$ is the average pairwise distance between $\mathbb{P}, \mathbb{Q}$, and $\mathcal{E}^2(\mathbb{P}, \mathbb{Q}) \overset{def}{=} \mathcal{I}(\mathbb{P}, \mathbb{Q}) - \frac{1}{2}\mathcal{I}(\mathbb{P}, \mathbb{P}) - \frac{1}{2}\mathcal{I}(\mathbb{Q}, \mathbb{Q})$ is (the square of) energy distance ([38]).

[Proof of Lemma 3.4] Consider a trivial transport plan $\pi = \mathbb{P} \times \mathbb{Q}$. Its estimation of the ground-truth Wasserstein-1 distance as $\mathcal{I}(\mathbb{P}, \mathbb{Q})$. Since π is not necessarily an optimal plan, from definition (2.6) of $\mathbb{W}_1$ we have $\mathcal{I}(\mathbb{P}, \mathbb{Q}) \geq \mathbb{W}_1(\mathbb{P}, \mathbb{Q})$.

Consider a function $k(x, y) \overset{def}{=} \frac{1}{2}\|x\|_2 + \frac{1}{2}\|y\|_2 - \frac{1}{2}\|x - y\|_2$. It is a positive definite symmetric kernel ([42], Definition 13 & Proposition 14). Therefore, there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathbb{R}^D \rightarrow \mathcal{H}$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product in $\mathcal{H}$. We derive

$$\|x - y\|_2 = k(x, x) - 2k(x, y) + k(y, y) = \|\phi(x) - \phi(y)\|^2_{\mathcal{H}}, \tag{3.9}$$

where $\| \cdot \|^2$ is the (squared) norm in $\mathcal{H}$ induced by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. We substitute (3.9) to (2.6) and obtain

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int\limits_{\mathbb{R}^D \times \mathbb{R}^D} ||x - y||_2 d\pi(x, y) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int\limits_{\mathbb{R}^D \times \mathbb{R}^D} ||\phi(x) - \phi(y)||^2_{\mathcal{H}} d\pi(x, y) = \tag{3.10}$$

$$\inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int\limits_{\mathcal{H} \times \mathcal{H}} ||x' - y'||^2_{\mathcal{H}} d\big[(\phi, \phi)\sharp\pi\big](x', y') \geq \inf_{\pi' \in \Pi(\phi\sharp\mathbb{P}, \phi\sharp\mathbb{Q})} \int\limits_{\mathcal{H} \times \mathcal{H}} ||x' - y'||^2_{\mathcal{H}} d\pi'(x', y') = \tag{3.11}$$

$$\mathbb{W}_2^2(\phi\sharp\mathbb{P}, \phi\sharp\mathbb{Q}) \geq \left\| \int_{\mathbb{R}^D} \phi(x) d\mathbb{P}(x) - \int_{\mathbb{R}^D} \phi(y) d\mathbb{Q}(y) \right\|^2_{\mathcal{H}} = \mathcal{E}^2(\mathbb{P}, \mathbb{Q}). \tag{3.12}$$

In transition from line (3.10) to (3.11), we use the change of variables $x' = \phi(x)$ and $y' = \phi(y)$. In line (3.11), we note that instead of searching for an OT plan between $\mathbb{P}$ and $\mathbb{Q}$ in $\mathbb{R}^D$, one may equivalently search for an OT plan between $\phi\sharp\mathbb{P}$ and $\phi\sharp\mathbb{Q}$ in $\mathcal{H}$. However, only plans $\pi' \in \Pi(\phi\sharp\mathbb{P}, \phi\sharp\mathbb{Q})$ should be considered for which there exists $\pi \in \Pi(\mathbb{P}, \mathbb{Q})$ satisfying $(\phi, \phi)\sharp\pi = \pi'$. In the right-hand side of (3.11), we consider the **inf** over all plans between $\phi\sharp\mathbb{P}$, $\phi\sharp\mathbb{Q}$ which is its superset. Thus, in (3.11) we have the inequality. Next, we note that the right-hand side of (3.11) is (the square of) the Wasserstein-2 distance ($\mathbb{W}_2^2$) between $\phi\sharp\mathbb{P}$ and $\phi\sharp\mathbb{Q}$ w.r.t. Hilbert squared norm $\|\cdot\|_{\mathcal{H}}^2$. It is lower bounded by the squared norm of difference between means of distributions (9, §1).[1] It remains to note that means of $\phi\sharp\mathbb{P}$ and $\phi\sharp\mathbb{Q}$ are the kernel mean embeddings of $\mathbb{P}, \mathbb{Q}$. Thus, the squared norm difference is the Maximum Mean Discrepancy (MMD) w.r.t. the kernel k (42, Definition 10). For the kernel k in view, the squared MMD between $\mathbb{P}, \mathbb{Q}$ is the squared energy distance (42, §2). It is worth noticing, that more information about strongly 1—Lipschitz continuity functions will be included in my own publication "Wasserstein GANs are not optimal transport" (unpublished).

3.4 training

Now, one can move on to the following part that deals with the transfer learning concept. As it was mentioned above, the transfer process starts with training f^{src} neural network in source dataset $\mathcal{S}$. Given observations $\{(x_i, y_i)\}_{i=1}^N : x_i \in \mathbb{X}_s, y_i \in \mathbb{Y}_s$ that is N labeled training samples from a source domain. Having optimized parameters of the neural network f^{src}, we get learned set of parameters that we will use as prior distribution $p(w)$ for the training of f^{tgt}. Since, target domain differs from a source domain in such problems as domain adaptation, zero-shot learning, one-shot learning and transfer learning, then $\mathbb{Y}_s$ distinguishes from $\mathbb{Y}_t$. For instance, if we consider the classification problem, then CIFAR-10 (26) might be as a source dataset $\mathcal{S}$, while ImageNet, which is composed of 1000 different classes (27) might be as a target dataset $\mathcal{S}$.Clearly, since amount of classes differ from each other, it means, that $\mathbb{Y}_s$ not equals to $\mathbb{Y}_t$. Then, the model f_{cl}^{src}, that maps hidden representation of f_{enc}^{src} to $\mathbb{Y}_s$, is useless for transferring problem, because label spaces of source datasets and target don't coincide. Thus, having obtained the optimized parameters of f_{enc}^{src}, one can collect prior distribution $p(w)$ in accordance with section 3.1. We assume, that prior distribution $p(w)$ is fully-factorized by layers without independence in a layer of f_{enc}^{src} with the exception of (4). Also, we introduce variational distribution $q(w)$, that is also fully-factorized by each $l-th$ layer

$$q(w) = \prod_{l=1}^{L} q_l(w) \tag{3.13}$$

and describes distribution weights of model f_{enc}^{tgt} in each layer. We use the Wasserstein-1 $\mathbb{W}_1$ distance as regularization term for the transfer learning problem, whose main goal is

[1] For completeness, we note that the lower bound may be further improved by considering the covariances of embedded distributions $\phi\sharp\mathbb{P}, \phi\sharp\mathbb{Q}$, see (9) for details. We use only the means to keep the exposition simple.

approximate prior distribution $p(w)$ by variational distribution $q(w)$. The main advantage of using this metric between two distributions is ability to compute it without analytical form of distributions. Also, one can get unbiased estimation of this distance by batches of datasets, having the optimal transport plan T^{*}, that map samples from prior distribution $p(w)$ to variational distribution $q(w)$.

Considering a j-th from L layers of f_{enc}^{src} and f_{enc}^{tgt}, weights of this layer are sampled from $p_j(w)$ for f_{enc}^{src} and from $q_j(w)$ for f_{enc}^{tgt} correspondingly. Then, one can denote weights from f_{enc}^{src} and f_{enc}^{tgt} as sampples from correspond distributions:

$$w_{encj}^{src} \sim p_j(w), \quad w_{encj}^{tgt} \sim q_j(w).$$

On the one hand, we would like to adapt parameters of f_{enc}^{tgt} to get high performance in target dataset $\mathcal{T}$. To solve this problem, the minimization of cross-entropy loss function is often used (26):

$$\mathcal{L}_{ce}(w) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f^{tgt}(x_i, w)) \quad \mathcal{T} = (x_i, y_i)_{i=1}^{N} : x_i \in \mathbb{X}_t, y_i \in \mathbb{Y}_t, \tag{3.14}$$

where $\ell(\cdot,\cdot)$ is the cross-entropy loss function, while x_i and y_i are a pair object-label from target datset $\mathcal{T}$. On the other hand, we would like to accelerate the training process by transforming the prior distribution $p(w)$ of the trained f_{enc}^{src} model's weights. To solve it, we approximate one's parameters of f_{enc}^{tgt} by others from f_{enc}^{src}. Nevertheless, we don't approximate their strictly as (29; 53; 6), thereby deteriorating flexibility of models to new data domains, that was demonstrated in (6). As we prove in lemma 3.1, the contemporary methods constitute upper estimation of ground-truth the Wasserstein-1 distance between distributions of parameters from differ models. Thus, we propose regularization scheme that is able to find the optimal transport plan T^{*} to compute ground-truth the Wasserstein-1 distance $\mathbb{W}_1$ between prior and variational distributions.

To provide computation of Wasserstein-1 distance between the aforementioned distributions, we should introduce the adversarial framework, which is based on Optimal Transport, and firstly described in (3). This framework requires a network $f(\cdot) : \mathbb{R}^D \to \mathbb{R}$ that is dual Kantorovich's potential in (2.9). This framework is used in the adversarial game (15) between generator $G(z) : \mathbb{Z} \to \mathbb{R}^D$, which make samples from samples of simple distribution, whose support is $\mathbb{Z}$, and critic $f(x)$, which tries to distinguish samples real distribution from samples that are created by the generator. The main goal of generator is to create such samples to fool the critic. In accordance with (3; 2; 17; 33),one can write the loss function as follow mini-max optimization problem:

$$\min_{\psi \in \Psi} \max_{\phi \in \Phi} \mathbb{E}_{z \sim \mathcal{N}(0,I)} f_\phi(G_\psi(z)) - \mathbb{E}_{x \sim p(x)} f_\phi(x), \tag{3.15}$$

where $p(x)$ is groud-thruth distribution of data and optimization is taken over sets of parameters Φ and Ψ for f_ϕ and G_ψ correspondingly.

However, the network $f^{tgt}_{enc}(x_i, w)$ play a role of a generator that makes filters during the training process in target dataset. Then, one can rewrite the mini-max game in our case as

$$\min_{w^{tgt} \in \mathbb{W}} \max_{\theta \in \Theta} \mathbb{E}_{w^{tgt}_{enc} \sim q(w)} f_\theta(w^{tgt}_{enc}) - \mathbb{E}_{w^{src}_{enc} \sim p(w)} f_\theta(w^{src}_{enc}), \tag{3.16}$$

where $\mathbb{W}$ is a set of parameters of f^{tgt}_{enc} and Θ is a set of parameters of strongly 1-Lipschitz continuity network. Then, taking into account (3.14) and fully-factorization of distributions $p(w)$ and $q(w)$, the final optimization problem for our method is postulated as:

$$\max_{\theta \in \Theta} \min_{w^{tgt} \in \mathbb{W}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\mathrm{y}_i | \mathrm{x}_i, w^{tgt}) + \sum_{j=1}^{J} \lambda_j [\mathbb{E}_{q_j(w)} \mathrm{f}_{j\theta_j}(w^{tgt}_{enc_j}) - \mathbb{E}_{p_j(w)} \mathrm{f}_{j\theta_j}(w^{src}_{enc_j})], \tag{3.17}$$

where λ_j is a trade-off between cross-entropy loss and regularization term in j-th layer.

Chapter 4

Numerical experiments

4.1 The Wasserstein-1 computation experiments

The first experimental section deals with the computation of ground-truth Wasserstein-1 distance between two distributions. We compare estimations of $\mathbb{W}_1$ by WGANs (2; 17; 34; 33; 31; 32; 41; 35; 1) and by our proposed method that includes strongly 1-Lipschitz continuity neural networks. Also, we demostrate, that many methods of WGANs cannot ever estimate lower and upper bound of $\mathbb{W}_1$, which were described in lemma 3.4.

It is worth noticing, that there are some papers, that doubt in the true computation of $\mathbb{W}_1$ by WGANs (44; 32; 37). However, these approaches use discrete optimal transport for calculation unbiased estimation of the Wasserstein-1 distance and cannot generalize own approach for continuous case. We offer the following framework for unbiased estimating of $\mathbb{W}_1$ for any continuous dual solvers (34; 31; 14; 41; 17; 37; 35). In accordance with the definition (2.6), the ground-truth Wasserstein-1 distance in Kantrovich's terms is giben by:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \inf_{\pi(x,y)} \int_{\mathbb{R}^D \times \mathbb{R}^D} ||x - y|| d\pi(x, y),$$

where $\pi(x, y)$ is a transport plan. The infimum of $\mathbb{W}_1$ is attained, when a transport plan $\pi(x, y)$ is the optimal transport plan $\pi^*(x, y)$. Thus, in order to get unbiased estimation of the $\mathbb{W}_1$, one should get access to the optimal transport plan π^*.

To overcome this obstacle, we can generate two distributions with the known optimal transport plan between them. Firstly, we generate samples from simple base distribution as standard normal distribution $\mathbb{P} = \mathcal{N}(0, I)$. Then, in accordance with (40, lemma 3.6), one can take strongly 1-Lipschitz function $f(\cdot)$ and this function defines the motion samples from the known $\mathbb{P}$ to a distribution $\mathbb{Q}$. For example, one can look at the dual surface of this strongly 1-Lipschitz function and the motion of samples from $\mathbb{P}$ to $\mathbb{Q}$ in 2 dimensional case in figure 4.1.

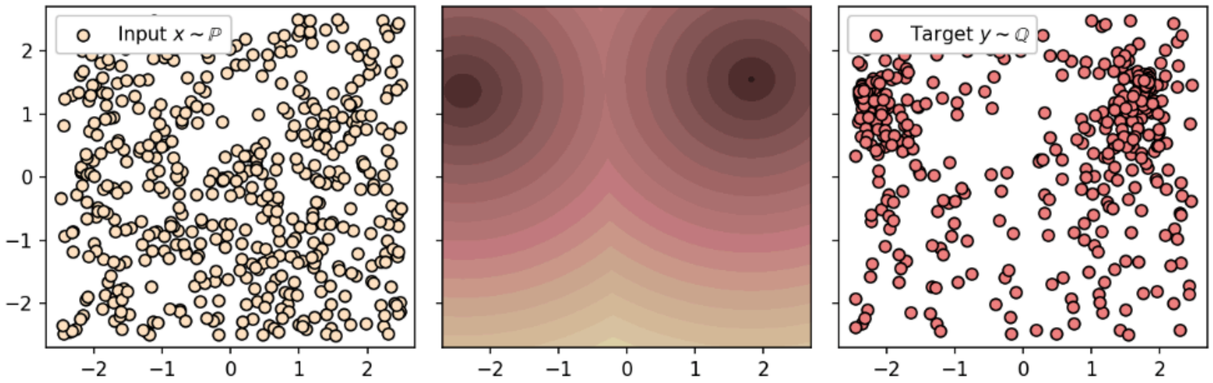
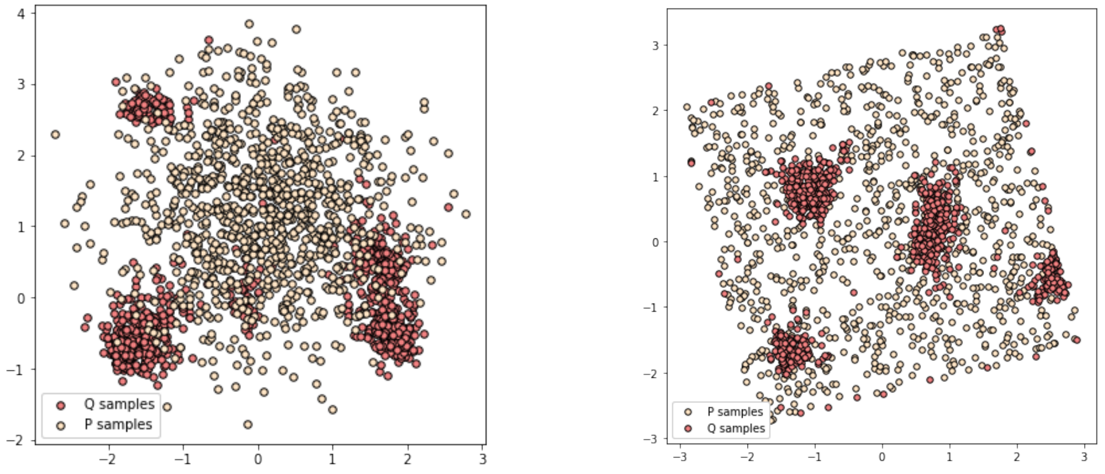


Figure 4.1: Motion of samples from uniform distribution $\mathbb{P} = \mathcal{U}(0, 1)$ to unknown distribution $\mathbb{Q}$ and dual surface of strongly 1-Lipschitz neural network in 2 dimensional case, that defines this motion.



(a) Motion from $\mathbb{P} = \mathcal{N}(0, I)$ to $\mathbb{Q}$

(b) Motion from $\mathbb{P} = \mathcal{U}(0, 1)$ to $\mathbb{Q}$

Figure 4.2: Motion from different $\mathbb{P}$ to different $\mathbb{Q}$

Since the dual surface of $f(\cdot)$ has two funnels, then the majority of samples of $\mathbb{P}$ move to them in accordance with (40, lemma 3.6). Since we know how points move, it means that we know optimal transport plan π^* in according to lemma (40, lemma 3.6). However, It is worth to recall, that to make unbiased estimation of $\mathbb{W}_1$, we need only in samples from the optimal transport plan $\pi^*(x, y)$, not knowing the analytical form of distributions. Thus, we can calculate estimation of the Wasserstein-1 distance.

The proposed method is able to move points from any base distribution $\mathbb{P}$ to a unknown distribution $\mathbb{Q}$. In figure 4.2a is demonstrated motion of samples from $\mathbb{P} = \mathcal{N}(0, I)$ and $\mathbb{Q}$ and standard uniform distribution is $\mathbb{P}$ in 4.2b.

Thus, having samples from different distributions, that are connected by an optimal transport plan, we can calculate unbiased estimation of the W_1 distance and compare the proposed approach with most common approaches (3; 17; 35; 33; 1; 34; 41; 31). How-

Dim	2	4	8	16	32	64	128
Lip-Clip (3)	6.41	3.55	2.02	1.44	0.51	0.04	0.01
Lip-GP (17)	9.25	7.66	6.69	5.05	3.25	2.45	1.77
Lip-LP (35)	9.01	7.38	6.07	5.78	4.20	3.29	1.56
Lip-SN (33)	11.27	10.03	7.94	6.32	5.28	3.19	1.26
Lip-SO (1)	14.67	13.89	12.67	11.00	9.87	4.03	3.09
Lip-WP (28)	14.89	14.21	13.96	11.22	10.07	7.68	3.35
Reg (41)	12.05	10.31	7.99	4.08	2.13	0.11	0.02
MM:B (31)	14.98	13.43	10.24	5.16	1.28	0.19	0.06
MM (34)	14.21	13.21	12.21	11.08	8.42	7.29	6.26
Lower	5.22	4.98	4.57	4.18	3.21	2.39	2.06
True (Our)	15.11	14.33	13.29	13.18	12.22	11.29	11.26
Upper	15.50	17.31	17.98	19.23	20.44	22.29	25.67

Table 4.1: The comparison of calculation of W_1 distance by most common approaches (3; 17; 35; 33; 34; 1; 41; 31) against our proposed framework for the ground-truth computation and computation lower and upper estimations of the Wasserstein-1 distance from lemma 3.4.

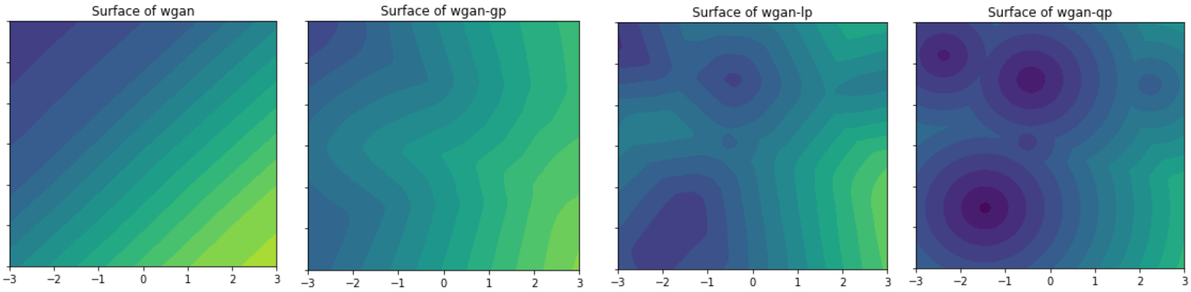


Figure 4.3: Dual surfaces of Lip-Clip (3), Lip-GP (17), Lip-LP (35) and ground-truth real surface

ever, it is worth saying, that WGANs don't use samples from the optimal transport plan $\pi^*(x, y)$ in accordance with (2.9), while they use samples from marginal distributions $\mathbb{P}$ and $\mathbb{Q}$ correspondingly. Thus, we can evaluate and compare estimations of W_1 distance, that are obtained by the aforementioned approaches with lower and upper estimations for $\mathbb{W}_1$ from lemma 3.4. We provide the following tables ?? for each dimensionality and for each methods. We can see, that no one method cannot accurately compute ground-truth the Wasserstein-1 distance with the increasing of the dimensionality. Moreover, some of these methods (41; 3; 31) cannot ever estimate simple lower estimation of $\mathbb{W}_1$.

Moreover, in case of 2-dimensional experiment, the strongly 1-Lipschitz function $f(\cdot)$ is defined on $\mathbb{R}^2$. Then, one can consider surface of this function and compare with the correspond surface of WGANs. In figures 4.3, 4.4 and 4.5, we can see these surfaces, where

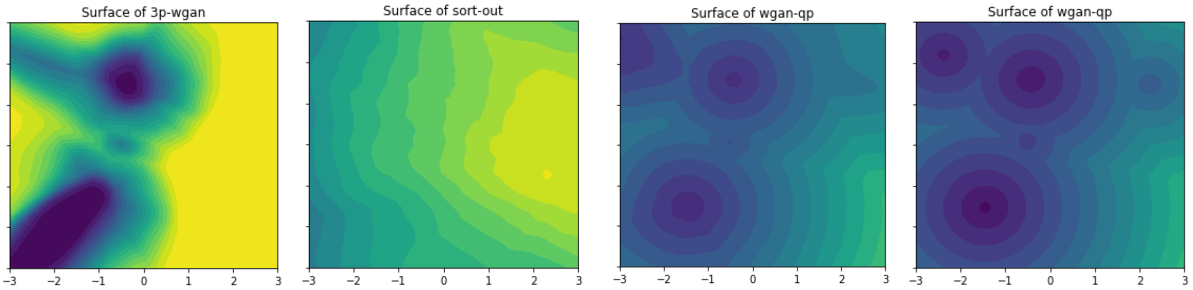


Figure 4.4: Dual surfaces of MM (34), Lip-SO (1), MM:B (31) and ground-truth real surface

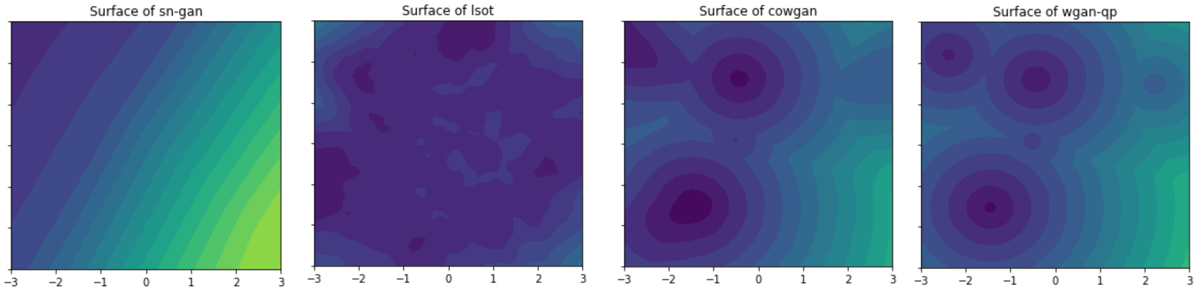


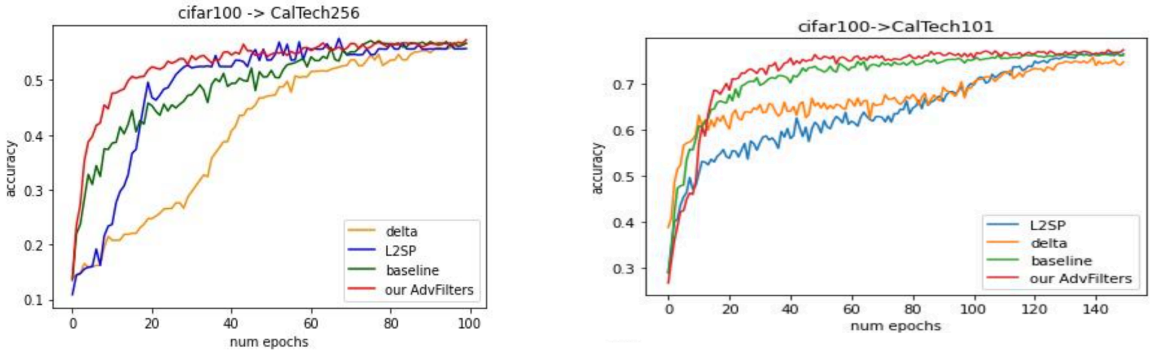
Figure 4.5: Dual surfaces of Lip-SN (33), Reg (41), CoWGAN (28) and ground-truth real surface

axes are coordinates of input samples from $\mathbb{P}$ distribution. Also, one can detach, that Lip-LP (35), Lip-WP (28) and MM:B (31) model dual surfaces, that are sufficiently similar to the ground-truth dual surface of strongly 1-Lipschitz function. In accordance with table 4.1, MM:B (31) and Lip-WP (28) sufficiently well approximate the ground-truth distance, while they also perfectly reconstruct the true dual surface of strongly 1-Lipschitz function. Nonetheless, the ability to well estimate W_1 distance get so much worse with the increasing of the dimensionality. Also, it is worth noticing, that MM (34) and Lip-SO (1) get also worse for computation of the truth Wasserstein-1 distance, but not such drasticlly as MM:B and Lip-WP.

4.2 Transfer Learning Experiment

We evaluate the proposed technique for the transfer learning problem and compare with the current modern solutions (29; 53; 6) in the classification task. We use CIFAR-100 dataset (26) as a source dataset $\mathcal{S} = \{x_i, y_i\}_{i=1}^{N} : x_i \in \mathbb{X}_s, y_i \in \mathbb{Y}_s$, where $\mathbb{Y}_s$ is a 100-component set. We utilize several target $\mathcal{T} = \{x_i, y_i\}_{i=1}^{N} : x_i \in \mathbb{X}_t, y_i \in \mathbb{Y}_t$ to better estimate proposed approach. As a target datasets we use:

- CalTech-256 (16) dataset is often used for generic object recognition. It is worth saying, that objects of this dataset is similar to objects from CIFAR-100 dataset. Thus, prior



(a) CalTech-256 with 60 training samples

(b) CalTech-101 with 60 training samples

Figure 4.6: Accuracy of methods (29; 53; 6) and our proposed method in target datasets CalTech-256 and CalTech-101 with 60 training samples per a class

knowledge $p(w)$, that is extracted from f^{src}_{enc} is useful for optimization parameters of f^{tgt}_{enc}.

- Finally, we use CalTech-101 dataset as yet another target dataset to demostrate performance of the method. This dataset has only objects and has not scenes with the exception of CalTech-256 (16).

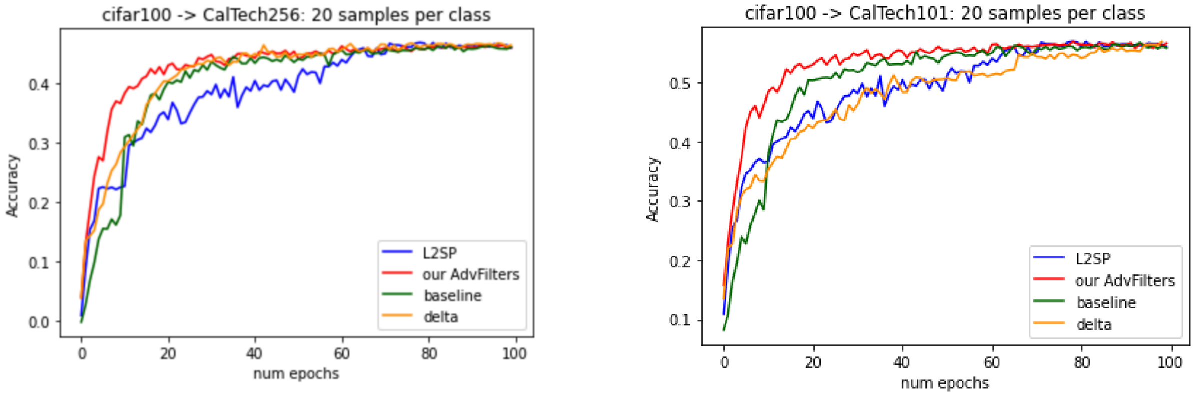
- We use generic subsample of generic objects from ImageNet dataset (27)

First of all, we divide the source dataset $\mathcal{S}$ to two parts. The first part has 70% of objects for optimization of parameters f^{src}, while we use the second to measure quality of the model. We use Accuracy score as a criterion of quality of a DNN:

$$Acc(\mathrm{x}) = 1 - \frac{1}{m}\sum_{i=1}^{m}[\mathrm{f}^{src}(\mathrm{x}_i, \mathrm{w}^{src}) \neq \mathrm{y}_i].$$

For optimization of parameters of f^{src}, we use stochastic gradient descent over random mini-batches from $\mathbb{X}_s$. The size of a batch equals to 64, while we use ADAM (21) with learning rate 0.001 and momentum 0.9. We run 9000 iterations from 5 random initial positions of f^{src} to get more rich set of parameters in each layer, thereby providing more samples for a prior distribution $p(w)$.

Having trained f^{src}, we remove f^{src}_{cl} and remains only f^{src}_{enc}, whose parameters are used as prior knowledge. We initialize parameters of f^{tgt}_{enc} by parameters one of five models f^{src}_{enc}, which were trained on a source dataset. Meanwhile, we initialize parameters of f^{tgt}_{cl} by fixed values, for instance by ones as it was mentioned in (6). To support the concept of adversarial training (15; 2), we optimize parameters of the critic g_ϕ by maximization the $\mathbb{W}_1$



(a) CalTech-256 with 20 training samples

(b) CalTech-101 with 20 training samples

Figure 4.7: Accuracy of methods (29; 53; 6) and our proposed method in target datasets CalTech-256 and CalTech-101 with 20 training samples per a class

Experiment	CalTech101:60	CalTech101:20	CalTech256:60	CalTech256:20
L2-SP (53)	67.45	49.74	50.51	38.68
Delta (29)	67.76	49.58	27.77	41.33
Baseline (47)	71.09	51.77	52.94	40.04
AdvFilters (Our)	72.28	53.18	52.94	42.57

Table 4.2: The comparison of AUC metric for modern transfer learning approaches (53; 29; 47) with our proposed method in 4 different experiments.

distance during several steps to get the optimal critic, while the parameters of the generator f^{tgt} is fixed. We set critic steps equal to 10. Having optimized parameters of the critic, we move on training of f^{tgt} by minimization the same cost, doing one generator update.

We evaluate the score of the proposed method and compare this method with the current transfer learning approaches with two target datasets CalTech-256 and CalTech-101. Also, we measure the accuracy of methods in scenario with two different volume of training objects. For example, we use all 60 training samples per class of CalTech-256 as a target dataset in figure 4.6a, while we utilize only 20 training per class in figure ??. The same situation with target dataset CalTech-101 in figures 4.7b and 4.6b. One can detect, that our method possess fast convergence with the exception of other methods in each experiment. To better demonstrate fast convergence of our method, we use AUC (area under curve) metric, that is area under accuracy curve. Thus, we show the comparing of AUC metric for different methods in different target datasets CalTech-256 and CalTech-101 with volumes of 20 and 60 training samples in the table 4.2, while Cifar-100 is source dataset. One can see, that the proposed method has metric higher, than anyone else in each experiment.

Chapter 5

Conclusion

Thus

- Having proposed and justified the novel method inductive transfer learning, whose regularization term is the Wasserstein-1 distance between parameters from source's model and target's model correspondingly.
- Having proposed and theoretically justified deep neural networks, that has strongly 1-Lipschitz property.
- Having developed method for unbiased estimation of the Wasserstein-1 distance in case of continuous probability distributions.
- Having conducted experiments of transfer learning on several target datasets, that demonstrate fast convergence of the proposed method.

Bibliography

[1] Anil, C., Lucas, J., and Grosse, R. Sorting out lipschitz function approximation. In International Conference on Machine Learning (2019), PMLR, pp. 291–301.

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

[3] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. arXiv preprint arXiv:1701.07875 (2017).

[4] Atanov, A., Ashukha, A., Struminsky, K., Vetrov, D., and Welling, M. The deep weight prior. arXiv preprint arXiv:1810.06943 (2018).

[5] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems (2016), pp. 2172–2180.

[6] Chen, X., Wang, S., Fu, B., Long, M., and Wang, J. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. Advances in Neural Information Processing Systems 32 (2019).

[7] Chernodub, A., and Nowicki, D. Norm-preserving orthogonal permutation linear unit activation functions (oplu). arXiv preprint arXiv:1604.02313 (2016).

[8] Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. IEEE transactions on pattern analysis and machine intelligence 39, 9 (2016), 1853–1865.

[9] Cuesta-Albertos, J. A., Matrán-Bea, C., and Tuero-Diaz, A. On lower bounds for thel 2-wasserstein metric in a hilbert space. Journal of Theoretical Probability 9, 2 (1996), 263–283.

[10] Daniels, G., Maunu, T., and Hand, P. Score-based generative neural networks for large-scale optimal transport. Advances in Neural Information Processing Systems 34 (2021).

[11] Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence 28, 4 (2006), 594–611.

[12] French, R. M. Catastrophic forgetting in connectionist networks. Trends in cognitive sciences 3, 4 (1999), 128–135.

[13] Friederich, S. Fine-tuning.

[14] Genevay, A., Cuturi, M., Peyré, G., and Bach, F. Stochastic optimization for large-scale optimal transport. In Advances in neural information processing systems (2016), pp. 3440–3448.

[15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Advances in neural information processing systems (2014), pp. 2672–2680.

[16] Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset.

[17] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of Wasserstein GANs. In Advances in Neural Information Processing Systems (2017), pp. 5767–5777.

[18] He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (2015), pp. 1026–1034.

[19] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (2016), pp. 770–778.

[20] Kantorovich, L. V. On the translocation of masses. Journal of mathematical sciences 133, 4 (2006), 1381–1382.

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

[22] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences 114, 13 (2017), 3521–3526.

[23] Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. Wasserstein-2 generative networks. In International Conference on Learning Representations (2021).

[24] Korotin, A., Li, L., Genevay, A., Solomon, J., Filippov, A., and Burnaev, E. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. arXiv preprint arXiv:2106.01954 (2021).

[25] Korotin, A., Selikhanovych, D., and Burnaev, E. Neural optimal transport. arXiv preprint arXiv:2201.12220 (2022).

[26] Krizhevsky, A., and Hinton, G. Convolutional deep belief networks on cifar-10. Unpublished manuscript 40, 7 (2010), 1–9.

[27] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25 (2012).

[28] Kwon, D., Kim, Y., Montúfar, G., and Yang, I. Training wasserstein gans without gradient penalties. arXiv preprint arXiv:2110.14150 (2021).

[29] Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L., Chen, Z., and Huan, J. Delta: Deep learning transfer using feature map with attention for convolutional networks. arXiv preprint arXiv:1901.09229 (2019).

[30] Makkuva, A. V., Taghvaei, A., Oh, S., and Lee, J. D. Optimal transport mapping via input convex neural networks. arXiv preprint arXiv:1908.10962 (2019).

[31] Mallasto, A., Frellsen, J., Boomsma, W., and Feragen, A. (q, p)-Wasserstein GANs: Comparing ground metrics for Wasserstein GANs. arXiv preprint arXiv:1902.03642 (2019).

[32] Mallasto, A., Montúfar, G., and Gerolin, A. How well do WGANs estimate the Wasserstein metric? arXiv preprint arXiv:1910.03875 (2019).

[33] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018).

[34] Nhan Dam, Q. H., Le, T., Nguyen, T. D., Bui, H., and Phung, D. Threeplayer Wasserstein GAN via amortised duality. In Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI) (2019).

[35] Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of wasserstein gans. arXiv preprint arXiv:1709.08894 (2017).

[36] Peyré, G., Cuturi, M., et al. Computational optimal transport. Foundations and Trends® in Machine Learning 11, 5-6 (2019), 355–607.

[37] Pinetz, T., Soukup, D., and Pock, T. On the estimation of the Wasserstein distance in generative models. In German Conference on Pattern Recognition (2019), Springer, pp. 156–170.

[38] Rizzo, M. L., and Székely, G. J. Energy distance. wiley interdisciplinary reviews: Computational statistics 8, 1 (2016), 27–38.

[39] Sanjabi, M., Ba, J., Razaviyayn, M., and Lee, J. D. On the convergence and robustness of training GANs with regularized optimal transport. arXiv preprint arXiv:1802.08249 (2018).

[40] Santambrogio, F. Optimal transport for applied mathematicians. Birkäuser, NY 55, 58-63 (2015), 94.

[41] Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. Large-scale optimal transport and mapping estimation. arXiv preprint arXiv:1711.02283 (2017).

[42] Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. The Annals of Statistics (2013), 2263–2291.

[43] Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. Zero-shot learning through cross-modal transfer. Advances in neural information processing systems 26 (2013).

[44] Stanczuk, J., Etmann, C., Kreusser, L. M., and Schonlieb, C.-B. Wasserstein GANs work because they fail (to approximate the Wasserstein distance). arXiv preprint arXiv:2103.01678 (2021).

[45] Sun, Y., Wang, X., and Tang, X. Deep convolutional network cascade for facial point detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (2013), pp. 3476–3483.

[46] Tanaka, A. Discriminator optimal transport. Advances in Neural Information Processing Systems 32 (2019).

[47] Torrey, L., and Shavlik, J. Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, 2010, pp. 242–264.

[48] Villani, C. Topics in optimal transportation. No. 58. American Mathematical Soc., 2003.

[49] Villani, C. Optimal transport: old and new, vol. 338. Springer Science & Business Media, 2008.

[50] Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning 1, 1–2 (2008), 1–305.

[51] Wang, M., and Deng, W. Deep visual domain adaptation: A survey. Neurocomputing 312 (2018), 135–153.

[52] Weng, L. From gan to wgan. arXiv preprint arXiv:1904.08994 (2019).

[53] Xuhong, L., Grandvalet, Y., and Davoine, F. Explicit inductive bias for transfer learning with convolutional networks. In International Conference on Machine Learning (2018), PMLR, pp. 2825–2834.

[54] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? Advances in neural information processing systems 27 (2014).

[55] You, K., Kou, Z., Long, M., and Wang, J. Co-tuning for transfer learning. Advances in Neural Information Processing Systems 33 (2020), 17236–17246.