МИНИСТЕРСТВ ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)
ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР ИМ. А.А. ДОРОДНИЦЫНА РАН
КАФЕДРА "ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Колесов Александр Сергеевич

Состязательный метод дообучения нейронной сети в задаче переноса информации

Выпускная квалификационная работа магистра

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

Москва—2022

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]; [55]; [32]), 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 ([50]).

When the amount of training samples for a target task is insufficient, fine-tuning can suffer from Catastrophic forgetting ([12]) and Negative transfer ([50]). 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, ([55]) 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 ([32]) , 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; 18; 10), domain adaptation (8) and became the core of new generative models (26; 33). 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; 18; 36; 37; 44), while Wasserstein-2 distance is used by generative models for finding the best mapping between distributions (26; 33; 27).

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 (32; 6; 55). 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 (36; 2; 18; 34; 35; 37) 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.

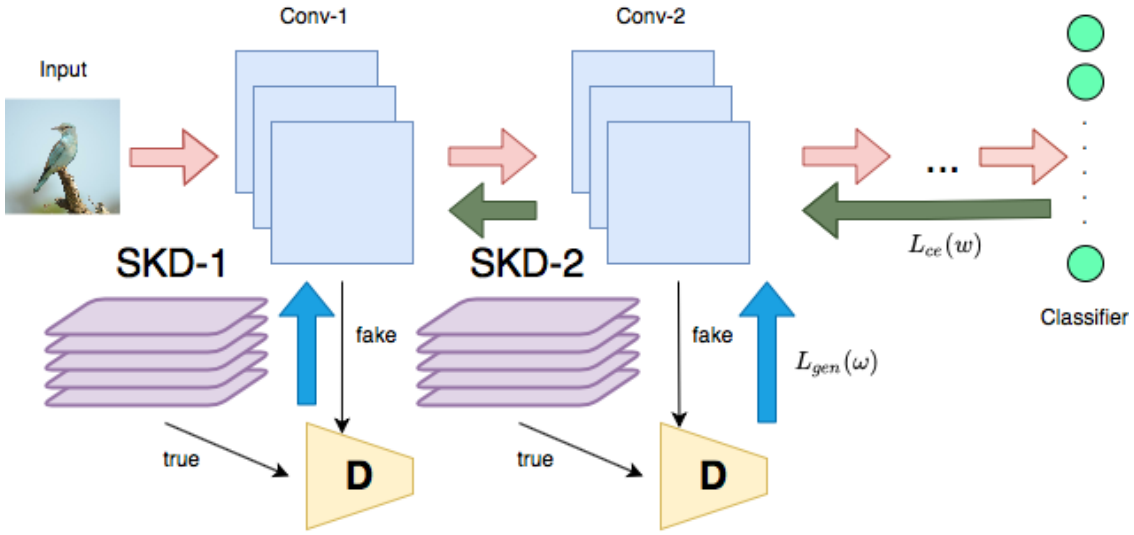


Figure 1.1: The learning process for generator through training steps. SKD is set of source's filters from a correspond layer. D are discriminators that distinguish generator's filters as fake from SKD's filters as true. $L_{ce}(w)$ is cross-entropy loss, while $L_{gen}(w)$ is adversarial generator loss.

Notation. We work in the $\mathbb{R}^D$ space that is endowed with the Euclidean norm $||\cdot||_2$. We denote the set of Borel probability distributions on $\mathbb{R}^D$ with finite first moment by $\mathcal{P}_1(\mathbb{R}^D)$. We use $\Pi(\mathbb{P}, \mathbb{Q})$ to denote the set of probability distributions on $\mathbb{R}^D \times \mathbb{R}^D$ with marginals $\mathbb{P}$ and $\mathbb{Q}$. All the integrals are computed over $\mathbb{R}^D$, if not stated otherwise. We write $f \oplus g \leq ||\cdot||_2$ for functions $f, g : \mathbb{R}^D \to \mathbb{R}$ if $\forall x, y \in \mathbb{R}^D$ satisfy $f(x) + g(y) \leq ||x - y||_2$. We denote a C-Lipschitz function by $||f||_L \leq C$.

Chapter 2

Problem statement

Definition 2.1 The function $\mathrm{f}(x, w) : \mathbb{X} \times \mathbb{W} \to \mathbb{Y}$ 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]; [53]), One-Shot Learning ([11]), Zero-Shot Learning ([46]), Transfer Learning ([50]; [6]; [32]; [55]), 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. We consider ResNet architectures ([20]) of DNNs that are composed of two main parts. The first part is termed as "encoder": $\mathrm{f}_{enc}(\mathrm{x}, \mathrm{w}) : \mathbb{X} \times \mathbb{W} \to \mathbb{Q}$, where $\mathbb{Q}$ is a latent space of encoder's features. The second component is referred to as "classifier" : $\mathrm{f}_{cl}(\mathrm{q}, \mathrm{w}) : \mathbb{Q} \times \mathbb{W} \to \mathbb{Y}$. 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 (19).

The modern algorithms of Transfer Learning (55; 6; 32) 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 (52) 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 (54) 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 (55; 32; 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; 55; 32). 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 (32; 55; 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) (55), for rigorous regularization we inhibit catastrophic forgetting problem, whereas exacerbate negative transfer problem (50). 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(55) 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 (32) 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

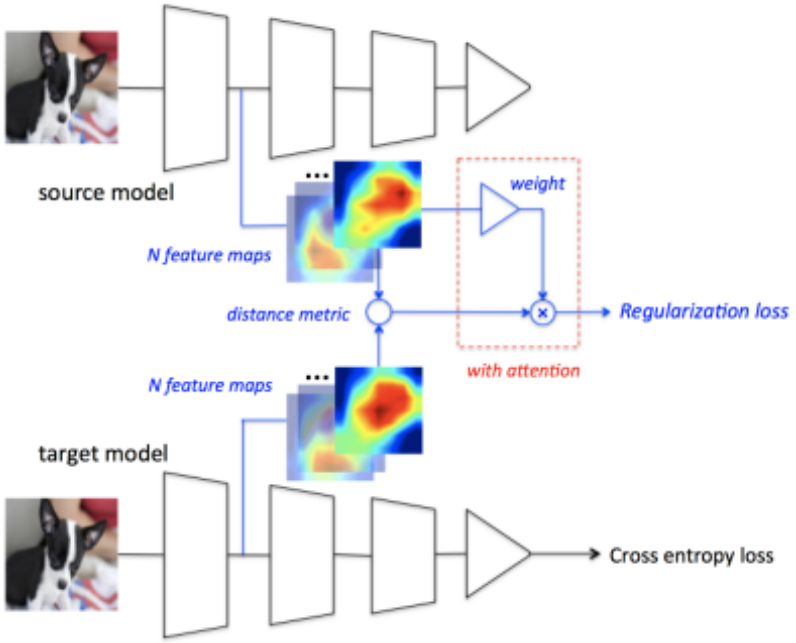


Figure 2.1: The scheme of transfer learning method DELTA.

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_{enc}^{tgt} 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 (55; 32) 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 (32) or (55) or $L2$-penalty with the offered loss function that reflects the fact, that singular values of f_{enc}^{tgt} shoudn't be high enough. In figure 2.2, one can look at the scheme of this method.

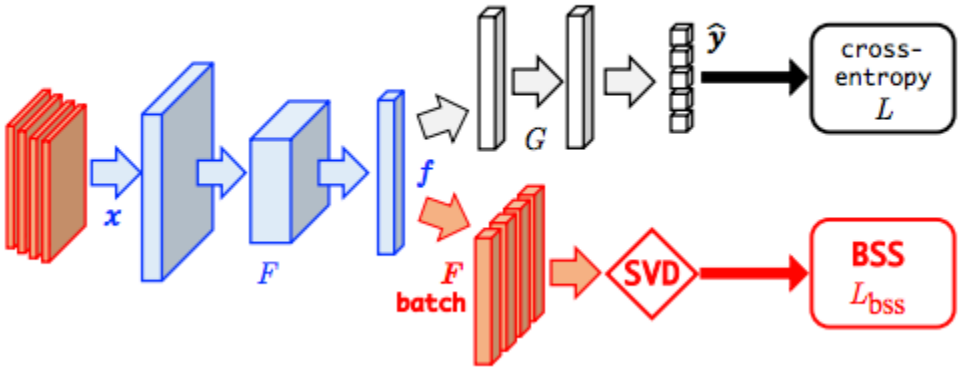


Figure 2.2: The scheme of Batch spectral shrinkage .

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 (57) 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 (55; 32) and the method, that is based on $L2$-penalty.

2.2 Background on Optimal Transport

Primal Formulation. For $\mathbb{P}, \mathbb{Q} \in \mathcal{P}_1(\mathbb{R}^D)$, the Monge's formulation of the Wasserstein-1 ($\mathbb{W}_1$) distance is

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \stackrel{def}{=} \inf_{T\sharp\mathbb{P}=\mathbb{Q}} \int ||x - T(x)||_2 d\mathbb{P}(x), \tag{2.5}$$

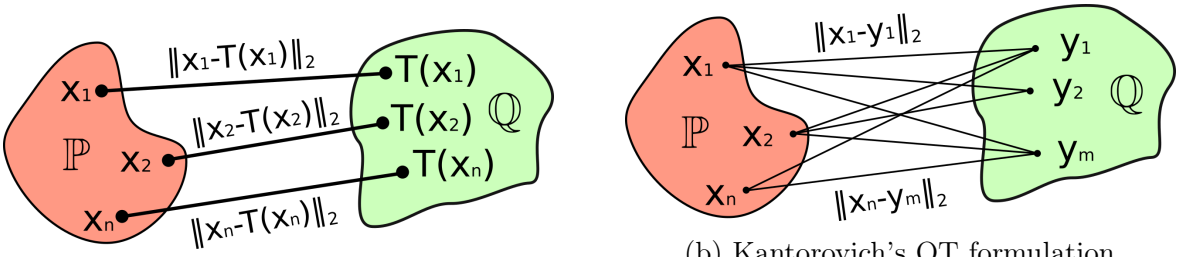
where $\inf$ is taken over all measurable functions $T : \mathbb{R}^D \to \mathbb{R}^D$ (deterministic transport maps) that map $\mathbb{P}$ to $\mathbb{Q}$.

Nevertheless, in Monge's formulation (2.5) , there is not always a deterministic transport map. To solve this issue Kantorovich (23) proposes to consider stochastic transport plans. Thereby, he 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. 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 probabilistic 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 (43).



(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. For $\mathbb{P}, \mathbb{Q} \in \mathcal{P}_1(\mathbb{R}^D)$, the dual formulation of $\mathbb{W}_1$ is given by (51, Th.5.10):

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

Using the definition of c-transform (51) , one can alternatively express (2.7) as

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

where $f^c(y) = \min_{x \in \mathbb{R}^D} [||x - y||_2 - f(x)]$ is the c-transform. In accordance with , the equation (2.8) can be further restricted by 1-Lipschitz functions. In this case, $f^c(y)$ becomes $-f(y)$. Then, another formulation of $\mathbb{W}_1$ is :

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

The final dual formulation is the most popular, because is used in WGANs (36; 2; 18).

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 (35). 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 $\mathbb{W}_1$ (39, 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 (49), 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.

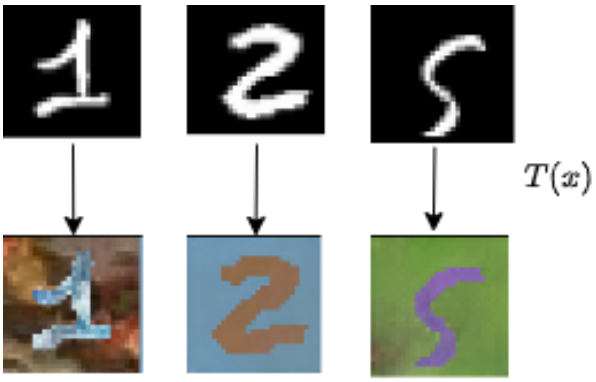


Figure 2.4: transport plan $T(x)$ in Domain Adaptation problem between distributions of samples from MNIST dataset and samples from USPS dataset .

- Using the gradient $\partial \mathbb{W}_1(\mathbb{P}_\alpha, \mathbb{Q})/\partial\alpha$ to update generative models. In, 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 data distribution and $\mathbb{P}_\alpha$ is a learned parametric distribution. The data distribution $\mathbb{P}_\alpha$ is most often generated from simple fixed latent s-dimensional ($s < D$) distribution $\mathbb{S}$ by a neural network "generator" $G_\alpha : \mathbb{R}^s \to \mathbb{R}^D$. 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:

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

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.

Quantitative evaluation of OT solvers. Existing solvers are typically tested as the loss in WGANs without evaluating its actual OT performance. The quality of the generated samples is evaluated by standard metrics such as FID (21) or IS (41). These metrics do not provide understanding about the quality of the solver itself, since they depend on components of the model that are not related to OT.

In (39; 35; 47), the authors use discrete $\mathbb{P}, \mathbb{Q}$ to show that some solvers imprecisely compute $\mathbb{W}_1$. Their approach is not applicable to evaluation of the OT gradient, as ∇f^* is ill-defined in the discrete case. For example, when $\mathbb{P} = \delta_0$, $\mathbb{Q} = \delta_1$, it holds that $f^* = -[x]_+$ is an optimal critic, but it is not even differentiable at $x = 0 = \text{Supp}(\mathbb{P})$. The existence of the OT gradient is studied, e.g., in (22).

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 (55; 6; 32).

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 (56), 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:

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

Moreover, the auxiliary variational lower bound is its the general cost, that is composed of term 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:

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

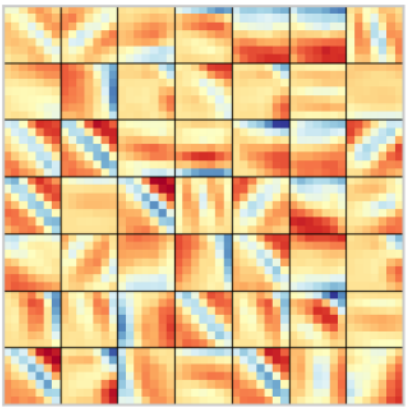
where I identity d-dimensional covariance matrix, while d is the dimensionality of the weight $w_{i,l}$. Undoubtedly, this factorization is sufficiently simple and doesn't probably reflect the ground truth distribution $p(w)$. Thus, this fact is the additional judgment to consider the Wasserstein-1 $\mathbb{W}_1$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

We follow common concepts (4; 50) and introduce the following definition of parameters from a source model f^{src}.

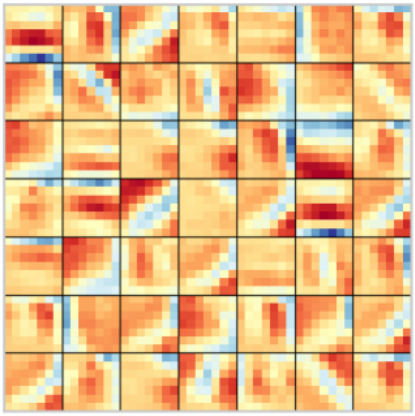
Definition 3.1 The parameters from a distribution of parameters of l-th model's layer $p_l(w)$ is referred to as SKD(Source Kernel Distribution) of l-th layer.

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; 55; 6; 32; 50) 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 3.15 , 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 Vanilla 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 (55; 6; 32) 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 (32; 6; 55; 57) 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 (55; 57; 6) or feature maps (32) 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,

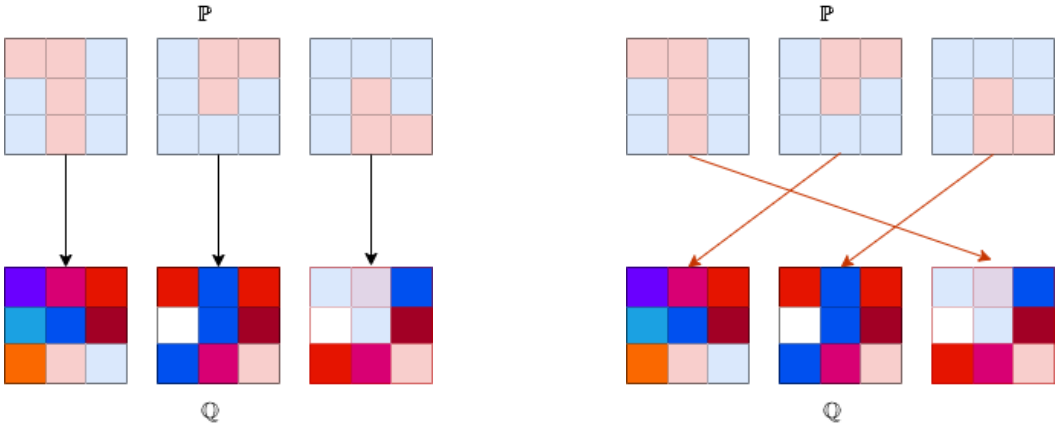
$$\Omega(\mathrm{w}) = \beta||\mathrm{w}_{enc}^{tgt} - \mathrm{w}_{enc}^{src}||_2^2 + \alpha||\mathrm{w}_{cl}^{tgt}||_2^2.$$

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" (25; 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 (55; 32). Moreover, they experimentally shows up at all improvement their scheme of the exception of modern transfer learning techniques.

- DELTA : The final method (32) 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 (48). 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; 32; 55; 57) 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_{enc}^{tgt} as a distribution of this parameters $\mathbb{Q}$ and f_{enc}^{src} 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 conisder penalization schemes as distance between distributions of parameters $\mathbb{P}$ and $\mathbb{Q}$ respectively in accordance with (4; 55; 6; 32; 50), then one can formulate the following important lemma for the following analysis of the proposed method:

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 trsnfer 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:

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

Prove.

3.3 1-lipschitz networks

The dual formulation of (2.9) constraint 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; 18; 36; 1; 38; 37; 42; 15; 10). The main challenge of these approaches is to enforce 1-Lipschitz continuity for a neural network $f_\theta : \mathbb{R}^D \to \mathbb{R}$ or accurately compute c-transfrom decomposition. There are many approaches to provide this property for a neural network. We give brief description and comparisons of these methods in table 1.

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.

⌊Lip-Clip⌋ The authors of (3) propose to approximate f by a neural network f_θ, whose space of parameters is a compact, for instance, D-dimensional cube $\Theta = [-c, c]^D$. Then, f_θ is Lipschitz continuous with some unknown constant M ,i.e., $||f_\theta||_L \leq M$. 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:

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

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

⌊Lip-GP⌋ The authors of (18) prove , that under mild assumptions on the OT plan π^*, the norm of optimal critic $||\nabla_x f^*(x)||_2$ equals to 1 in transport rays. Thus, they regularize f_θ for violating abovementioned constraint by the "Gradient penalty" method as:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \sup_{\theta} [\int f_\theta(x) d\mathbb{P}(x) - \int f_\theta(y) d\mathbb{Q}(y) + \lambda \mathcal{R}(f_\theta)] \quad \lambda > 0. \tag{3.3}$$

The penalization $\mathcal{R}(f_\theta)$ is :

$$\int (||\nabla_r f_\theta(r)||_2 - 1)^2 d\mu(r),$$

where μ is the probability distribution that describes random variable $r = xt + (1-t)y$ with $x \sim \mathbb{P}$ and $y \sim \mathbb{Q}$.

⌊Lip-LP⌋ In (38), the authors demonstrate that the training of WGANs with "Gradient penalty" suffers from an instability and high magnitudes of the regularization term. To overcome this issue, they introduce "Lipschitz penalty" the penalization method for 1-Lipschitz's constraint as:

$$\int (\max\{0, ||\nabla_r f(r)||_2 - 1\})^2 d\mu(r),$$

showing that proposed regularization has less values, than the "Gradient penalty" constraint.

⌊Lip-SN⌋ The authors of (36) claim, that (18; 38) 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 $||f_\theta||_L \leq 1$.

⌊Lip-SO⌋ The authors of (1) show, that a neural network with spectral normalization (36) 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 (36) 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 (42; 15; 10), the authors optimize an unconstrained regularized dual form

of (2.9)

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{\theta,\phi}[\int f_\theta(x)d\mathbb{P}(x) - \int g_\phi(y)d\mathbb{Q}(y) - \mathcal{R}(f_\theta, g_\phi)]. \tag{3.4}$$

The entropic and L_2 regularizer $\mathcal{R}$ (42, Eq.5) softly penalizes dual potentials f_θ, g_ϕ for disobeying the constraint $f \oplus g \leq ||\cdot||_2$.

⌊MM:B⌋ The authors of (34) consider the dual formulation (??) with the inner problem as:

$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \sup_{\theta}[\int f_\theta(x)d\mathbb{P}(x) - \int \min_{x\in\mathbb{R}^D}[||x-y||_2 - f_\theta(y)]d\mathbb{Q}(y)]. \tag{3.5}$$

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

⌊MM⌋ The authors of (37) introduce a minimax reformulation of (??). 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:

$$W_1(\mathbb{P},\mathbb{Q}) = \max_{\theta}[\int f_\theta(y)d\mathbb{Q}(y) - \min_{\psi}\int\{f_\theta(H_\psi(x)) + ||H_\psi(x) - x||_2\}d\mathbb{P}(x)]. \tag{3.6}$$

In accordance with (29, Lemma 4), the optimal "mover" H^* is the OT plan from $\mathbb{P}$ to $\mathbb{Q}$.

⌊MM:R⌋ One may also recover the OT gradient ∇f^* from the OT map T^*. Consider the form

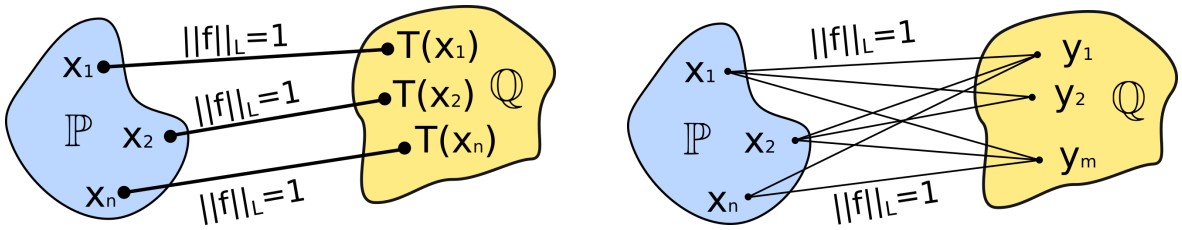
$$\mathbb{W}_1(\mathbb{P},\mathbb{Q}) = \max_{g}\int \min_{T}\left[\|T(x)-x\|_2 - g(T(x))\right]d\mathbb{P}(x) + \int g(y)d\mathbb{Q}(y), \tag{3.7}$$

which is a reversed (27) version of (??), i.e., the roles of $\mathbb{P},\mathbb{Q}$ are swapped and $T : \mathbb{R}^D \to \mathbb{R}^D$. For some optimal saddle points (g^*, T^*) of (3.7) it holds that T^* is an OT map (29, Lemma 4), (14, Lemma 2). With mild assumptions on $\mathbb{P},\mathbb{Q}$, one may recover T^* and use the

equality $\nabla f^*(x) = \frac{x - T^*(x)}{\|x - T^*(x)\|_2}$ (? , §1) to obtain the OT gradient. An alternative is to use $\nabla f^*(x) = -\nabla g^*(x)$.

Solver	Related works	Parametrizations of potentials	Quantitatively tested as OT	Tested in GANs
Maximin ⌊MM⌋	(37)	$f_\theta : \mathbb{R}^D \to \mathbb{R}$ $h_\psi : \mathbb{R}^D \to \mathbb{R}^D$		Three-player WGAN (37)
Regularized ⌊Reg⌋	(42; 44; 15; 10)	$f_\theta : \mathbb{R}^D \to \mathbb{R}$ $g_\phi : \mathbb{R}^D \to \mathbb{R}$	Gaussian case (26)	Ent.-regularized WGAN (42)
Enforcing Lipschitz ⌊Lip⌋	(3; 18; 36; 1)	$f_\theta : \mathbb{R}^D \to \mathbb{R}$		WGAN (3), WGAN-GP (18), SN-GAN (36), Sort-Out (1)

While these approaches provide 1-Lipschitz continuity for the function on all space, the authors of (43, Lemma 3.2) argue, that this function should be strongly 1-Lipschitz on transport rays 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, one has to develop such neural networks that satisfy to the property of strongly 1-Lipschitz continuity.



(a) trongly 1 Lipschitzness on transport rays (b) Strongly 1 Lipschitzness on transport rays

Figure 3.3: Strongly 1 Lipschitzness on transport rays

Let's consider one-layer dense network $f_\theta : \mathbb{R}^D \to \mathbb{R}$. In according to the definition, to provide strongly 1-lipschitz continuity, it takes the norm of gradient of $f(x)$ by inputs should be 1.

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

Obviously, having normalized weight matrix by its norm, one can get strongly 1-Lipschitz continuity for function. Clearly, such network has poor generalization and is linear function. To solve this problem, one need add non-linearity behaviour to this model. Let's consider

one-linear dense net $f_\theta : \mathbb{R}^D \to \mathbb{R}^k$, where $k < D$. To provide non-linearity behaviour, one can pick up the only output from k. For instance, one can build mini-max tree upon k output neurons.

Having solved the problem of linearity, one has to overcome problem of poor generalization. We prove that the following functions possess strongly 1-Lipschitz continuity.

Lemma 3.2 Let $f, g \in Lip_1$. Then, the following fucntions are strongly 1-Lipschitz too:

$$\max(f, g), \quad \min(f, g)$$

prove.

Lemma 3.3 Let $f, g \in Lip_1$. Then , the function $t : \mathbb{R}^{2D} \to \mathbb{R}$ that is defined as:

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

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

prove. -$\nabla_x f(x) = [\frac{\partial f}{\partial x_1} \quad ... \quad \frac{\partial f}{\partial x_n}]_{1xn}$

-$\nabla_y g(y) = [\frac{\partial g}{\partial y_1} \quad ... \quad \frac{\partial g}{\partial y_n}]_{1xn}$

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

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

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

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

Lemma 3.4 Let $f, g \in Lip_1$. Then, its composition $f(g(\cdot))$ is strongly 1-Lipschitz continuity

prove. - Derivative scalar valued function is

$$\frac{\partial U}{\partial f} = [\frac{\partial U}{\partial f_1} \cdots \frac{\partial U}{\partial f_n}]_{1xn}$$

- Derivative of vector function is

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ldots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

- Chain rule for jacobian

$$\nabla_x U(f(x)) = \frac{\partial U}{\partial f}\frac{\partial f}{\partial x}$$

- Then

$$\nabla_x U(f(x)) = [(\frac{\partial U}{\partial f_1}\frac{\partial f_1}{\partial x_1} + \cdots + \frac{\partial U}{\partial f_n}\frac{\partial f_n}{\partial x_1}) \ldots\ldots (\frac{\partial U}{\partial f_1}\frac{\partial f_1}{\partial x_n} + \cdots + \frac{\partial U}{\partial f_n}\frac{\partial f_n}{\partial x_n})]_{1xn}$$

- Then for strongly 1 Lipschitz function

$$(\frac{\partial U}{\partial f_1}\frac{\partial f_1}{\partial x_1} + \cdots + \frac{\partial U}{\partial f_n}\frac{\partial f_n}{\partial x_1})^2 + \cdots + (\frac{\partial U}{\partial f_1}\frac{\partial f_1}{\partial x_n} + \cdots + \frac{\partial U}{\partial f_n}\frac{\partial f_n}{\partial x_n})^2 = 1$$

- Then in case of 2 dimensional space

$$(\frac{\partial U}{\partial f_1}\frac{\partial f_1}{\partial x_1} + \frac{\partial U}{\partial f_2}\frac{\partial f_2}{\partial x_1})^2 + (\frac{\partial U}{\partial f_1}\frac{\partial f_1}{\partial x_2} + \frac{\partial U}{\partial f_2}\frac{\partial f_2}{\partial x_2})^2 = 1$$

- If we use $U(f) = \mathsf{max}(f_1, f_2)$ or $U(f) = \mathsf{min}(f_1, f_2)$, then f shoul be strongly 1 Lipschitz by components

$$(\frac{\partial U}{\partial f_1})^2 \quad ((\frac{\partial f_1}{\partial x_1})^2 + (\frac{\partial f_1}{\partial x_2})^2) = 1$$

Having created neural network with the desired property, one can compare computation abilities between the proposed method and another aforementioned approaches in the table. In the experimental session, we compare the Wasserstein-1 distance that is calculated by the modern methodologies with our method. It is worth noticing, that there are a lot of approaches ([35]; [47]; [39]), that theoretically and experimentally prove, that $\mathbb{W}_{\not\Vdash}$ is not the general cost of WGANs. For instance, the authors of ([39]) notice, that the Wasserstein-1 distance between two discrete distributions is changed with changing of the batch-size, while models and another parameters of the experiment are the same. Also, The authors of ([35])

Lemma 3.5 (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 $\mathcal{I}(\mathbb{P}, \mathbb{Q}) \stackrel{def}{=} \int \|x - y\|_2 d\mathbb{P}(x) d\mathbb{Q}(y)$ is the average pairwise distance between $\mathbb{P}, \mathbb{Q}$, and $\mathcal{E}^2(\mathbb{P}, \mathbb{Q}) \stackrel{def}{=} \mathcal{I}(\mathbb{P}, \mathbb{Q}) - \frac{1}{2}\mathcal{I}(\mathbb{P}, \mathbb{P}) - \frac{1}{2}\mathcal{I}(\mathbb{Q}, \mathbb{Q})$ is (the square of) energy distance (40).

[Proof of Lemma 3.5] Consider a trivial transport plan $\pi = \mathbb{P} \times \mathbb{Q}$. Its transport cost is $\mathcal{I}(\mathbb{P}, \mathbb{Q})$. Since π is not necessarily an optimal plan, from definition (??) of $\mathbb{W}_1$ we have $\mathcal{I}(\mathbb{P}, \mathbb{Q}) \geq \mathbb{W}_1(\mathbb{P}, \mathbb{Q})$.

Consider a function $k(x, y) \stackrel{def}{=} \frac{1}{2}\|x\|_2 + \frac{1}{2}\|y\|_2 - \frac{1}{2}\|x - y\|_2$. It is a positive definite symmetric kernel (45, Definition 13 & Proposition 14). Therefore, there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathbb{R}^D \to \mathcal{H}$ 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 (??) and obtain

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

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

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

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 $\|\cdot\|^2_{\mathcal{H}}$. 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

[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.

Maximum Mean Discrepancy (MMD) w.r.t. the kernel k (45, Definition 10). For the kernel k in view, the squared MMD between $\mathbb{P}, \mathbb{Q}$ is the squared energy distance (45, §2).

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 learned the neural network f^{src}, we get optimized set of parameters that we will use as prior distribution for the following steps. 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 (34) might be as a source dataset, while ImageNet, which is composed of 1000 different classes, might be as a target dataset (28). Clearly, since amount of classes differ from each other, it means, that $\mathbb{Y}_s$ not equals to $\mathbb{Y}_t$. It is worth saying, that f_{cl}^{src}, that maps hidden representation of f_{enc}^{src} to $\mathbb{Y}_s$, is useless for transferring, because label spaces of source dataset 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 3.1. Namely, we assume, that prior distribution is fully-factorized by layers without independence in a layer with the exception of (4). We use the Wasserstein-1 $\mathbb{W}_1$ distance as regularization term for the transfer learning problem. The main advantage of using this metric between two distributions $\mathbb{P}$ and $\mathbb{Q}$ is ability to compute it without analytical form of distributions. To get unbiased estimation of this distance, one has to get the optimal transport plan T^*, that map samples from $\mathbb{P}$ to $\mathbb{Q}$. Thus, having found an optimal transport plan, we need only in samples to compute the $\mathbb{W}_1(\mathbb{P},\mathbb{Q})$.

The following step of transfer learning process is to transform the information from f_{enc}^{src} to another network f_{enc}^{tgt} with the same neural architecture. Let's assume $f_{enc}^{src} : \mathbb{X}_s \to \mathbb{Q}_s$ is the encoder with L layers. Then, having proposed independence of weights in the model, one can represent parameters of j layer as samples from prior distribution $p_j(w)$. Meanwhile,

parameters of j layer of f_{enc}^{tgt} are represented by their distribution $q_j(w)$. Thus, we have:

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

Undoubtedly, we would like to adapt parameters of f^{tgt} to get high accuracy score in target-dataset $\mathcal{T}$. To solve this problem, the minimization of cross-entropy loss function is often used. Thus, having considered target dataset $\mathcal{T}$ and a neural network f^{tgt} . one can write it as follow:

$$\mathcal{L}_{ce}(w) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f^{tgt}(x_i, w)) \quad \mathcal{T} = (x_i, y_i)_{i=1}^{N} : x_i \in \mathbb{X}_t, y_i \in \mathbb{Y}_t \tag{3.13}$$

However, we use prior knowledge of paramters from f_{enc}^{src} as $p(w)$ to get distribution of filters $q(w)$ of f_{enc}^{tgt} , thereby providing the fast convergence with the exception of current modern techniques of transfer learning (32; 55; 6). Thus, we approximate one's parameters of f_{enc}^{tgt} by others from f_{enc}^{src}. Nevertheless, we don't approximate their strictly as (32; 55; 6), thereby they deteriorate flexibility of models to new data domains. As we prove in lemma 3.1, these contemporary methods constitutes 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 optimal transport plan T^* to compute ground-truth the Wasserstein-1 distance $\mathbb{W}_1$ between these distributions.

we use the adversarial framework that was firstly described in (16). The network $f_{enc}^{tgt}(x_i, w)$ play a role of a generator that makes filters during the training process. To maintain the paradigm of adversarial learning, such filters for each layer is called as fake and its distribution is as fake distribution $q(w)$. Then, filters from the f_{enc}^{src} is termed as real and distribution at each layer is real distribution $p(w)$. From the point of adversarial game, one should make fake distribution similar to real distribution. To reach this destination, it is necessary to define strongly 1-ipschitz discriminator $D_j(w, \theta)$ for each layer with own learned parameters as $\theta \in \mathbb{R}^s$ that takes filters w and has to distinguish true from fake. The goal of our generator $f_{enc}^{tgt}(x_i, w)$ is to mislead $D_j(w, \theta)$ by own fake filters from j-th layer, while minimizes cross-entropy loss. In accordance with (?), the j-th element of second loss term of the networks is as follows:

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

Thus, the full loss $\mathcal{L}_f$ of the network $f(x_i, w)$ on a target database is sum of cross-entropy loss and generator's loss

$$\mathcal{L}_f = \mathcal{L}_{ce} + \sum_j \beta_j \mathcal{L}_{gen}^j \tag{3.14}$$

Chapter 4

Numerical experiments

4.1 The Wasserstein-1 computation experiments

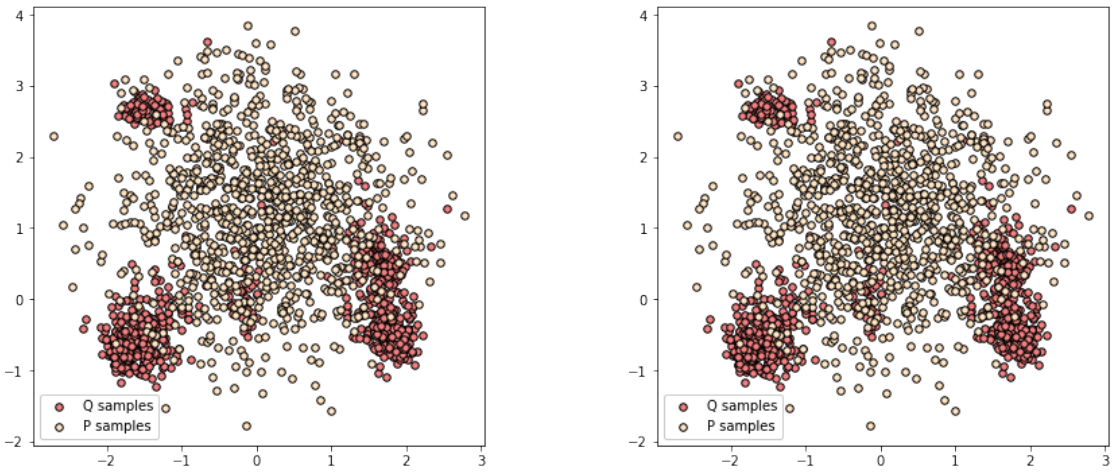
The first experimental session deals with the computation of ground-truth Wasserstein-1 distance between two distributions. It is worth noticing, that there are a lot of methods, that doubt in the computation of WGANs (47; 35; 39). However, these approaches use discrete optimal transport for unbiased estimating of the Wasserstein-1 distance and cannot generalize own judgments for continuous case. We offer the following framework for unbiased estimating of $\mathbb{W}_1$ for any continuous dual solvers (37; 34; 15; 44; 18; 39; 38). in accordance with the definition (2.6), the optimal transport cost 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 the OT plan. Thus, in order to get unbiased estimation of the optimal transport cost , one should get access to the optimal transport plan.

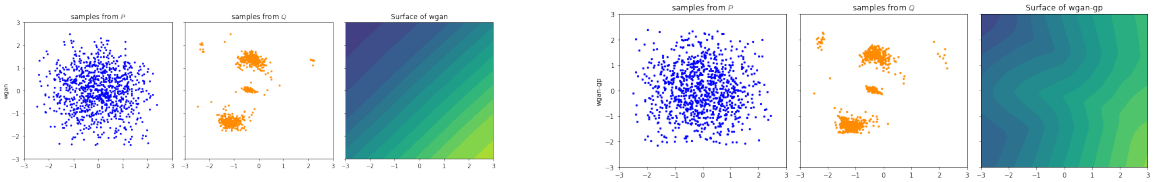
Thus, we can generate two distributions with known the known optimal transport plan. Precisely, we sample samples x from simple base distribution as standard normal distribution $\mathcal{N}(0, I)$. Then, in accordance with (43, Lemma 3.2), one can take strongly 1-Lipschitz function, that we created above, and all points will move from x to another location y, that are sampled from unknown distribution $\mathbb{Q}$. Thus, we know precise correspondence between samples from distributions and in according to this lemma , this correspondence is built by optimla transport plan $\pi^*(x, y)$. It is worth to recall, that to estimate the transport cost , we need only in samples from optimal transport plan $\pi^*(x, y)$. Thus, we can calculate the optimal transport cost.

To train on the proposed pairs of samples such methods as (36; 18; 2; 44; 34; 35) , we need in infinite samples from this distributions. To reach this destination, we use sampler from standard normal distribution as the initial sampler from $\mathbb{P}$ distribution , while sampler from



(a) PCA: $\mathbb{R}^4$ (b) PCA: $\mathbb{R}^{16}$

Figure 4.1: Principal compounds of $\mathbb{P}$ and $\mathbb{Q}$

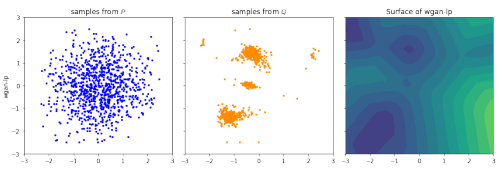


(a) 1a (b) 1b

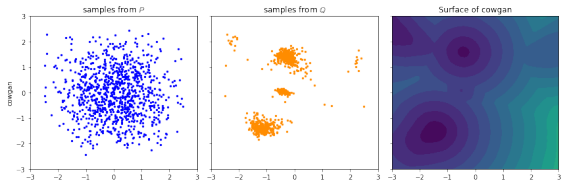
Figure 4.2: plots of....

$\mathbb{Q}$ is samples from the initial sampler , that is moved by strongly 1-Lipschitz continuity function. In figure 4.1a is demonstrated samples from $\mathbb{P}$ and $\mathbb{Q}$, that are obtained by the proposed method in principal components of samples from $\mathbb{Q}$ in case of 4-dimensional space, while 4.1b in case of 16-dimensional space. Having infinite samples from $\mathbb{P}$ and $\mathbb{Q}$, we can evaluate WGANs methods and compute unbiased estimation of the Wasserstein-1 distance. However, it is worth saying, that the aforementioned methods use for not samples from plan $\pi^*(x, y)$, while ,indeed, samples from marginal distributions $\mathbb{P}$ and $\mathbb{Q}$ correspondingly. Thus, we can evaluate these methods in the computation experiment and compare estimations, that are obtained by the aforementioned approaches with lower and upper estimations for $\mathbb{W}_1$ from lemma 3.5. We provide the following tables for each dimensionality and for each methods.

In case of 2-dimensional experiment, one can look at dual surface of WGANs and compare with the true surface of strongly 1-Lipschitz function, that makes this pairs of distributions.

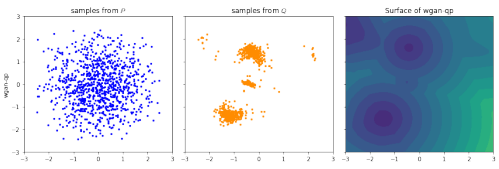


(a) 1a

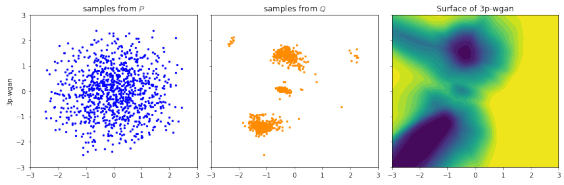


(b) 1b

Figure 4.3: plots of....

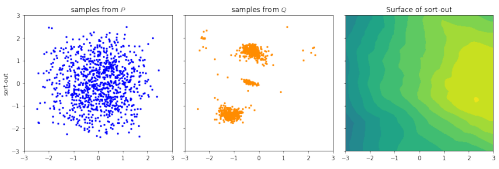


(a) 1a

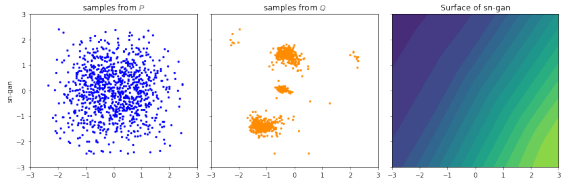


(b) 1b

Figure 4.4: plots of....

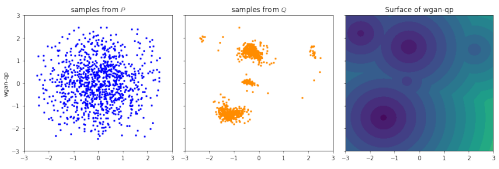


(a) 1a

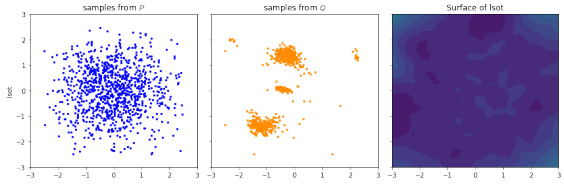


(b) 1b

Figure 4.5: plots of....



(a) 1a



(b) 1b

Figure 4.6: plots of....

4.2 Transfer Learning Experiment

We evaluate the proposed technique for the transfer learning problem and compare with the current modern solutions (32; 55; 6) in the classification task. We use CIFAR-100 dataset (30) 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 (17) 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 knowledge $p(w)$, that is extracted from f_{enc}^{src} is useful for optimization parameters of f_{enc}^{tgt}.

- 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 (17).

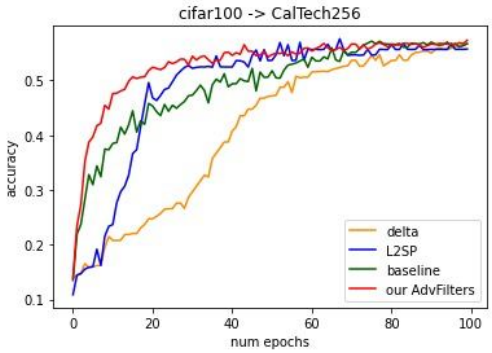
- We use generic subsample of generic objects from ImageNet dataset (31)

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 metric of quality of a DNN:

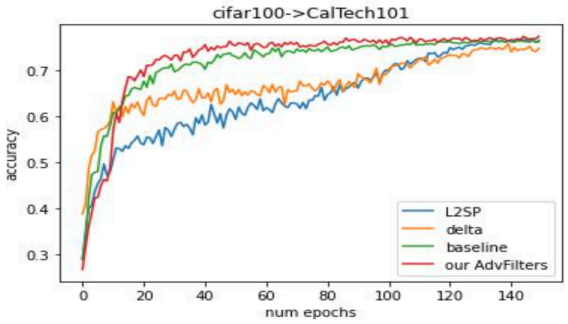
$$Acc(\mathrm{x}) = 1 - \frac{1}{m}\sum_{i=1}^{m}[\mathrm{f}^{src}(\mathrm{x}_i, \mathrm{w}^{src}) \neq \mathrm{y}_i].$$

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 (24) 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_{cl}^{src} and remains only f_{enc}^{src}, whose parameters are used as prior knowledge. We initialize parameters of f_{enc}^{tgt} by parameters one of 5 f_{enc}^{src} models , which were trained on a source dataset. Meanwhile, we initialize parameters of f_{cl}^{tgt} by fixed values , for instance by ones as it was mentioned in (6). To support the concept of adversarial training (16; 2), we optimize parameters of the critic g_ϕ by maximization the $\mathbb{W}_1$ distance during several steps to get the optimal critic, while the parameters of the generator f^{tgt} is



(a) 1a



(b) 1b

Figure 4.7: plots of....

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.

Chapter 5

Discussion and conclusion

Summarize your work in this section.

1. Summary of the main results of the work that is consistent with the Aim and Objectives.

2. Overall position on the global research landscape.

3. Comparative critical analysis: what you have deduced from the findings and how these results relate to previous research or other studies.

4. Research limitations.

Acknowledgements

Write here acknowledgments of financial assistance for the conduct of research and to specific individuals who contributed to the science. Dedications are not recommended and must reference scientific contributions.

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] Gazdieva, M., Rout, L., Korotin, A., Filippov, A., and Burnaev, E. Unpaired image super-resolution with optimal transport maps. arXiv preprint arXiv:2202.01116 (2022).

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in neural information processing systems (2017), pp. 6626–6637.

[22] Houdard, A., Leclaire, A., Papadakis, N., and Rabin, J. On the existence of optimal transport gradient for learning generative models. arXiv preprint arXiv:2102.05542 (2021).

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

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

[25] 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.

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

[27] 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).

[28] Korotin, A., Li, L., Solomon, J., and Burnaev, E. Continuous wasserstein-2 barycenter estimation without minimax optimization. In International Conference on Learning Representations (2021).

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

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

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

[32] 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).

[33] 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).

[34] 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).

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

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

[37] 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).

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

[39] 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.

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

[41] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs. In Advances in neural information processing systems (2016), pp. 2234–2242.

[42] 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).

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

[44] 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).

[45] 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.

[46] 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).

[47] 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).

[48] 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.

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

[50] 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.

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

[52] 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.

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

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

[55] 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.

[56] 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).

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