Transfer Learning in Computer Vision Tasks: Remember Where You Come From

Xuhong Li[a], Yves Grandvalet[a], Franck Davoine[a], Jingchun Cheng[b], Yin Cui[c], Hang Zhang[d], Serge Belongie[c], Yi-Hsuan Tsai[e], Ming-Hsuan Yang[f]

[a] *Alliance Sorbonne Université, Université de technologie de Compiègne, CNRS, Heudiasyc, UMR 7253, Compiègne, France.*
[b] *Tsinghua University*
[c] *Cornell University*
[d] *Amazon Inc*
[e] *NEC Labs America*
[f] *University of California at Merced*

Abstract

Fine-tuning pre-trained deep networks is a practical way of benefiting from the representation learned on a large database while having relatively few examples to train a model. This adjustment is nowadays routinely performed so as to benefit of the latest improvements of convolutional neural networks trained on large databases. Fine-tuning requires some form of regularization, which is typically implemented by weight decay that drives the network parameters towards zero. This choice conflicts with the motivation for fine-tuning, as starting from a pre-trained solution aims at taking advantage of the previously acquired knowledge. Hence, regularizers promoting an explicit inductive bias towards the pre-trained model have been recently proposed. This paper demonstrates the versatility of this type of regularizer across transfer learning scenarios. We replicated experiments on three state-of-the-art approaches in image classification, image segmentation, and video analysis to compare the relative merits of regularizers. These tests show systematic improvements compared to weight decay. Our experimental protocol put forward the versatility of a regularizer that is easy to implement and to operate that we eventually recommend as the new baseline for future approaches to transfer learning relying on fine-tuning.

Keywords: Transfer learning, parameter regularization, computer vision

1. Introduction

The L^2 parameter regularization, also known as weight decay, is commonly used in machine learning and especially when training deep neural networks. This simple regularization scheme restricts the capacity of the trained model by restraining the effective size of the search space during optimization, implicitly driving the parameters towards the origin. When having no a priori knowledge about the "true solution", the origin is an arbitrary yet reasonable choice. However, the parameters can be driven towards any value of the parameter space, and better results should be obtained for a value closer to the true one [12, Section 7.1.1].

Li et al. [21] recently proposed to use the pre-trained model as an a priori knowledge about the "true solution" in transfer learning: the starting point (*-SP*) should be used in place of the origin as the reference for parameter regularization. Figure 1 illustrates this scheme in a simple case that corresponds to linear regression. The left-hand side plot (a) represents the weight decay regularizer when starting from the origin (no fine-tuning): the optimizer goes towards the solution of the unregularized risk and stops when reaching the boundary of the admissible set. The center plot (b) represents the weight-decay regularizer when starting from the pre-trained solution, assumed to be in the vicinity of the solution to the unregularized risk: the optimizer reaches the previous solution; all the benefit of the pre-trained solution is lost. The right-hand side plot (c) represents the L^2*-SP* regularizer: the optimizer converges towards a solution between the pre-trained solution and the solution of the unregularized risk; the memory of the pre-trained solution is preserved. Note that the very same scenario is unlikely with non-convex deep models, but it seems dangerous to rely on non-convexity to prevent forgetting.

Li et al. [21] showed that the *-SP* regularizers improve transfer learning in image classification. However, weight decay still remains predominantly used in computer vision for fine-tuning, maybe due to the limited experimental evidence of the benefit of *-SP* regularizers. In this paper, we aim at demonstrating the general usefulness of L^2*-SP* for transfer learning, by providing novel evidences showing that the simple L^2*-SP* regularizer is applicable in a very wide scope.

The following sections provide the necessary background material regarding transfer learning (Section 2), regularizers (Section 3) and the tested approaches (Section 4). We then report in Section 5 a wide variety of experiments, dealing with state-of-the-art transfer learning schemes for image

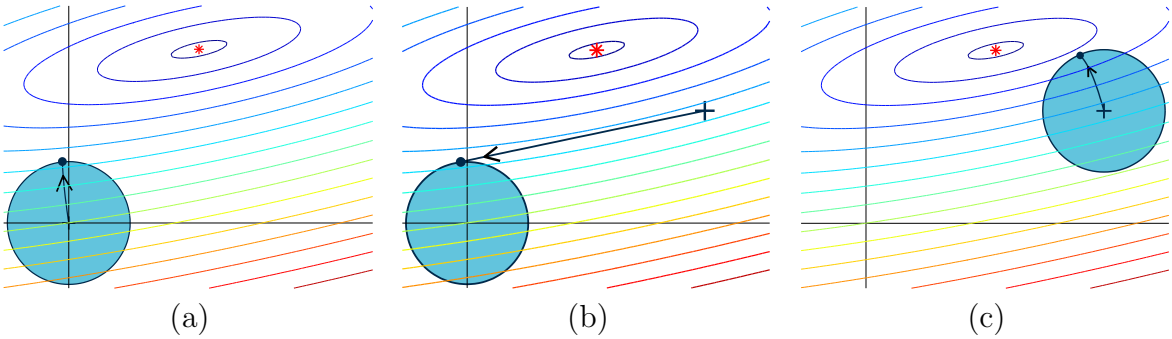

Figure 1: Inadequacy of the standard L^2 regularization for transfer learning. Each plot shows the same 2D parameter space in a simple transfer learning situation. The red star represents the minimum of the unregularized risk for the target problem; the black cross is the starting point of the optimization process, and the black point represents the result of a gradient-like optimizer, with intermediate solutions represented by the black segment. The ellipses represent the contour levels of the target problem, and the large blue circle represents the effective search domain defined by the regularizer (admissible set). The sub-figures correspond to: (a) the standard learning process with L^2 regularization (no fine-tuning), (b) the fine-tuning process with L^2 regularization, (c) the fine-tuning process with L^2-*SP* regularization.

classification (DSTL [7], short for *domain similarity for transfer learning*), image segmentation (EncNet [45]) and video analysis (SegFlow [5]). In order to avoid any form of experimental bias, all the experiments reported in this paper were carried out under the exact conditions of the original transfer learning schemes, by the main authors of the original approaches, who introduced a single modification in the fine-tuning protocol, by replacing L^2 by L^2-*SP* regularization. These experimental results show consistent improvement for L^2-*SP* regularization, demonstrating the versatility of the -*SP* regularizers for fine-tuning across network structures, datasets, and vision problems. These improvements are marginal to moderate, but should not be neglected since they come with minimal computing overhead. We thus claim that L^2-*SP* regularization should be adopted as a baseline, or in combination with other schemes in all vision applications relying on transfer learning.

2. Related Work

2.1. Inductive Transfer Learning

Regarding transfer learning, we follow the nomenclature of Pan and Yang [32]. A domain corresponds to the feature space and its distribution, whereas

a task corresponds to the label space and its conditional distribution with respect to features. The initial learning problem is defined on the source domain and the source task, whereas the new learning problem is defined on the target domain and the target task. According to domain and task settings during the transfer, Pan and Yang categorized several types of transfer learning problems. *Inductive transfer learning* is the situation where the target domain is identical to the source domain and the target task is different from the source task. Many recent applications of convolutional networks, like image classification [11, 7], object detection [35, 46, 24], object instance segmentation [14, 15, 25], image segmentation [5, 45, 4], depth estimation [26, 6], optical flow [16, 5, 20], human action recognition [47, 3], and person re-identification [34, 17], rely on transfer learning in the inductive transfer learning setting. All these approaches start with some model pre-trained on a source domain for image classification and fine-tune them on the target domain for a different task. They show state-of-the-art results in a challenging transfer learning setup, as going from classification to object detection or image segmentation requires notable modifications in the architecture of the network.

The success of these approaches relies on the generality of the representations that have been learned from a large database like ImageNet [8]. Yosinski et al. [44] quantify the transferability of these pieces of information in different layers, i.e. the first layers learn general features, the middle layers learn high-level semantic features and the last layers learn the features that are very specific to a particular task. Overall, the learned representations can be conveyed to related but different domains and the network parameters are reusable for different tasks.

2.2. Parameter Regularizers for Transfer Learning

Parameter regularization is widespread in deep learning. L^2 regularization has been used for a long time as a simple method for preventing overfitting by limiting the norm of the parameter vector. Besides L^2, other penalties have been proposed, such as max-norm regularization [38], which is found especially helpful when using dropout, or the orthonormal regularizer [43], which forces each kernel in one convolution layer to have minimum correlation with others.

The L^2-*SP* regularization we consider in this paper has been used in lifelong learning [22, 18] to cope with the catastrophic forgetting problem. A similar regularizer has also been used in domain adaptation for vision [36],

speaker adaptation [23, 31], and neural machine translation [1]. In brief, L^2-*SP* is an explicit inductive bias towards the initial parameters that has been proved to be useful in different transfer learning scenarios. However, weight decay remains predominantly used in the computer vision community for fine-tuning.

3. A Reminder on -*SP* regularizers

The regularizers that were recently proposed in [21] apply to the vector $\boldsymbol{w} \in \mathbb{R}^n$ containing all the network parameters that are to be adapted to the target task. The regularized objective function $\tilde{J}$ that is to be optimized is the sum of the standard objective function J and the regularizer $\Omega(\boldsymbol{w})$. In practice, J is usually the negative log-likelihood, so that the criterion $\tilde{J}$ could be interpreted in terms of maximum *a posteriori* estimation, where the regularizer $\Omega(\boldsymbol{w})$ would act as the log prior of $\boldsymbol{w}$. More generally, the minimization of $\tilde{J}$ is a trade-off between the data-fitting term and the regularization term.

***L*^{2}** *penalty.* The current baseline penalty for transfer learning is the L^2 penalty, also known as weight decay:

$$\Omega(\boldsymbol{w}) = \frac{\alpha}{2} \|\boldsymbol{w}\|_2^2 \quad , \tag{1}$$

where α is the regularization parameter setting the strength of the penalty and $\|\cdot\|_p$ is the p-norm of a vector.

L*^{2}*-SP. Let $\boldsymbol{w}^0$ be the parameter vector of the model pre-trained on the source problem, acting as the starting point (*-SP*) in fine-tuning. Using this initial vector as the reference in the L^2 penalty, we get:

$$\Omega(\boldsymbol{w}) = \frac{\alpha}{2} \left\|\boldsymbol{w} - \boldsymbol{w}^0\right\|_2^2 \quad . \tag{2}$$

Typically, the transfer to a target task requires some modifications of the network architecture used for the source task, such as on the last layer used for predicting the outputs. Then, there is no one-to-one mapping between $\boldsymbol{w}$ and $\boldsymbol{w}^0$, and we use two penalties: one for the part of the target network that shares the architecture of the source network, denoted $\boldsymbol{w}_{\mathcal{S}}$, the other one for the novel part, denoted $\boldsymbol{w}_{\bar{\mathcal{S}}}$. The compound penalty then becomes:

$$\Omega(\boldsymbol{w}) = \frac{\alpha}{2} \left\|\boldsymbol{w}_{\mathcal{S}} - \boldsymbol{w}^0_{\mathcal{S}}\right\|_2^2 + \frac{\beta}{2} \|\boldsymbol{w}_{\bar{\mathcal{S}}}\|_2^2 \quad . \tag{3}$$

Several other -*SP* regularizers exist, like L^2-*SP-Fisher* or *Group-Lasso-SP* [21], but we focus here on comparing L^2 with L^2-*SP*, because of the simplicity and efficiency of L^2-*SP* compared to the other -*SP* regularizers.

4. Transfer Learning Approaches

In this section, we present the three recent representative approaches of transfer learning with convolutional networks that will be used to compare the L^2 and L^2-*SP* regularizers. They cover a variety of setups, and their protocols rely at least partly on fine-tuning, originally implemented with weight decay.

4.1. EncNet

Zhang et al. [45] designed a context-encoding module to extract the relation between the object categories and their global semantic context in the image, so as to emphasize the frequent objects in one context and de-emphasize the rare ones. The proposed module explicitly captures contextual information of the scene using sparse encoding and learns a set of scaling factors, by which the feature maps are then rescaled for selectively highlighting the class-dependent feature channels. For an image segmentation problem, the features highlighted by the semantic context facilitate the pixel-wise prediction and improve the recognition of small objects. Meanwhile, an auxiliary loss is computed from the encoded features to better extract the contextual information.

We refer to this approach as *EncNet*, following [45]. It relies on a pre-trained ResNet [13] that is then evaluated on the PASCAL-Context dataset [10] for image segmentation.

4.2. SegFlow

Cheng et al. [5] constructed a network architecture, named *SegFlow*, with two branches for simultaneously (i) segmenting video frames pixel-wisely and (ii) computing the optical flow in videos. The segmentation branch is based on ResNet [13] transformed into a fully-convolutional structure, while the optical flow branch is an encoder-decoder network [9]. Both segmentation and optical flow branches have feature maps at multiple scales, enabling connections between the two tasks. Gradients from both tasks can pass through the two branches, and the last representations in feature space are shared.

SegFlow is initialized with two pre-trained networks: ResNet [13] for the encoding of the segmentation branch and FlowNetS [9] for the optical flow branch. It is then fine-tuned on the DAVIS 2016 dataset [33] for video object segmentation and the Scene Flow datasets [28] for optical flow.

4.3. Domain Similarity for Transfer Learning

Cui et al. [7] measure the similarity between the source domain, supposed to cover a broad range of objects, and the target domain, supposed to be more specific. They use the Earth Mover's Distance to compute the similarity between the source categories and the target domains. They then choose the top k categories of the source domain that best cover the target domains to pre-train the network from scratch.

The networks are pre-trained on subsets of ImageNet [8] and iNaturalist [41]. Here, we use their Subset B. This pre-trained network is then fine-tuned on different target databases (see Table 2). We refer to this approach as DSTL, the abbreviation of *domain similarity for transfer learning*.

5. Experiments

In this section, we experiment the three approaches, *i.e.* EncNet [45], SegFlow [5], and DSTL [7], with their original L^2 penalty implementation, and compare their performances with the L^2-*SP* penalty, in the very same experimental conditions. To ensure perfect replication of the original protocol, these experiments were carried out by the main authors of the original papers to avoid any kind of "competence bias".

When these experiments are carried out using the L^2 penalty, the fluctuations from the original results are only due to the inherent randomness of the stochastic learning process. When using the L^2-*SP* penalty, this is the only piece of code that was changed, and the very same conditions are observed otherwise, so as to ensure that differences in performances are only due to the differences between the two regularization approaches during fine-tuning.

5.1. Experimental Setup

Datasets. The characteristics of the source and target datasets are summarized in Table 1 and Table 2 respectively, including the size of each dataset, the task related and the approach that uses this dataset. Note that DSTL includes a scheme for building a source domain by selecting a subset of classes that are the most relevant to the target task, but it follows a fine-tuning transfer learning protocol nevertheless.

dataset	#images/scenes	#classes	task addressed	note	approach
ImageNet [8]	∼1.2M	1000	image classification	object-centered	all
iNaturalist [41]	∼675K	5,089	image classification	natural categories	DSTL
Flying Chairs [9]	∼2K scenes	-	optical flow	synthetic	SegFlow

Table 1: Source datasets: number of examples, number of classes, type of task addressed, and the approach(es) relying on this dataset as a source task.

dataset	#images/seq.	#classes	task addressed	approach
PASCAL-Context [29]	∼10K	59	image segmentation	EncNet
Scene Flow [28]	32 sequences	-	optical flow	SegFlow
DAVIS 2016 [33]	50 sequences	-	video segmentation	SegFlow
CUB200 [42]	∼11K	200	image classification	DSTL
Flowers102 [30]	∼10K	102	image classification	DSTL
Stanford Cars [19]	∼16K	196	image classification	DSTL
Aircraft [27]	∼10K	100	image classification	DSTL
Food101 [2]	∼100K	101	image classification	DSTL
NABirds [40]	∼50K	555	image classification	DSTL

Table 2: Target datasets: number of examples, number of classes, type of task addressed, and approach that uses this dataset as a target task.

Network Structures. The source task is usually a classification task, which alleviates the labeling burden. Conventionally, if the target task is also classification, like DSTL, the fine-tuning process starts by replacing the last layer with a new one, randomly generated, with a size defined by the number of classes in the target task. The modification of the network structure in quite light in this situation.

In contrast, for image segmentation and optical flow estimation, where the objectives differ radically from image classification, the source network needs to be modified, typically by adding a decoder part, which is much more involved than a single fully connected layer. Here, we follow exactly the original papers regarding the modifications of the network architectures.

Training Details. For training details, we refer the reader to the original papers [5, 45, 7], since again, nothing was changed on this matter. Note that L^2 and L^2-*SP* are only applied to weights in convolutional and fully connected layers: the normalization layers or the biases are not penalized to follow the usual fine-tuning protocol.

Evaluation Metrics. We briefly recall the evaluation metrics used for measuring the performance on these tasks. In image classification and segmentation, *accuracy* or *pixel accuracy* is defined as the ratio of correctly predicted examples or pixels to the total.

In segmentation, performance is evaluated by the *mean intersection over union* (mean IoU or mIoU). The intersection over union (IoU) compares two sets: the set of pixels that are predicted to be of a given category and the set of pixels that truly belong to this category. It measures the discrepancy between the two sets as the ratio of their intersection to their union. The mIoU is the mean of IoUs over all categories.

In optical flow, performance is evaluated by the average *endpoint error* (EPE) is defined as the average L^2 distance between the estimated optical flow and the ground truth at each pixel.

5.2. Experimental Results

Table 3 compares the results of fine-tuning with L^2 and L^2-*SP* of all approaches on their specific target tasks. We readily observe that fine-tuning with L^2-*SP* in place of L^2 consistently improves the performance, whatever the task, whatever the approach. Some of these improvements are marginal, but we recall that, compared to the L^2 fine-tuning baseline, L^2-*SP* only

approach	**target dataset**	**task**	**metric**	L^2	L^2-*SP*
EncNet-50	PASCAL-Context	image seg.	mIoU	50.84	51.17
EncNet-101	PASCAL-Context	image seg.	mIoU	54.10	54.12
SegFlow*	DAVIS	video seg.	IoU	65.5	66.2
SegFlow	DAVIS	video seg.	IoU	67.4	68.0
SegFlow	Monkaa *Final*	optical flow	EPE	7.90	7.17
SegFlow	Driving *Final*	optical flow	EPE	37.93	30.31
DSTL	CUB200	image classif.	accuracy	88.47	89.19
DSTL	Flowers102	image classif.	accuracy	97.21	97.68
DSTL	Stanford Cars	image classif.	accuracy	90.19	90.67
DSTL	Aircraft	image classif.	accuracy	85.89	86.83
DSTL	Food101	image classif.	accuracy	88.16	88.75
DSTL	NABirds	image classif.	accuracy	87.64	88.32

Table 3: Experimental results. For all metrics except EPE, higher is better. SegFlow marked with '*' does not use the optical flow branch. The optical flow results of SegFlow are evaluated on two subsets of Scene Flow databases [28], i.e. Monkaa and Driving.

requires an extra subtraction operation per weight during training, and that it has no extra computational cost in evaluation.

EncNet. The EncNet-50 and EncNet-101 are based on ResNet-50 and ResNet-101 [13] respectively, pre-trained on ImageNet [8]. The networks are fine-tuned on PASCAL-Context [29] for image segmentation, and their performances are measured by pixel accuracy and mIoU. There is no improvement in mIoU brought by L^2-*SP* for EncNet-101 and only a marginal one for EncNet-50: the importance of the choice of the regularizer is not crucial here thanks to the number of training examples that are highly informative owing to the pixelwise segmentation.

Table 4 provides more details, with the average test pixel accuracy and mIoU obtained with several values of the hyper-parameters α, β in Equation (3) and compared to the best solution obtained with L^2 fine-tuning. Pixel accuracy, which is the criterion that is actually used during training, is always improved by L^2-*SP*, even for suboptimal choices of the regularization parameters, but it is more noteworthy here to observe that it is relatively safer to increase the α/β ratio than to decrease it. In other words, for controlling the complexity of the overall network, being more conservative on

approach	α	β	**accuracy**	**mIoU**
EncNet-50 - L^2	1e-4	1e-4	79.09	50.84
EncNet-50 - L^2-*SP*	1e-4	1e-3	79.10	50.31
EncNet-50 - L^2-*SP*	1e-3	1e-4	79.18	51.12
EncNet-50 - L^2-*SP*	1e-4	1e-4	**79.20**	**51.17**
EncNet-101 - L^2	1e-4	1e-4	80.70	54.10
EncNet-101 - L^2-*SP*	1e-4	1e-4	**80.81**	**54.12**

Table 4: EncNet pixel accuracy and mIoU on the PASCAL-Context validation set according to regularization hyper-parameters.

the pre-trained part of the network is a better option than being more constrained on its novel part.

SegFlow. As for the segmentation performance of SegFlow, we have conducted two experiments: fine-tuning without the optical flow branch (denoted SegFlow* in Table 3) and fine-tuning the entire model. Both options are evaluated on the DAVIS target task [33]. The segmentation branch and the optical flow branch of SegFlow are pre-trained on ImageNet [8] and FlyingChairs [9] respectively. When applicable, both branches are regularized by L^2 fine-tuning or towards the pre-trained values by L^2-*SP* fine-tuning. The benefits of L^2-*SP* are again systematic and higher for SegFlow*, where less data is fed to the network during fine-tuning.

For the optical flow estimation, we also observe systematic benefits of L^2-*SP* (recall that, for the EPE measure, the lower, the better), with again higher impact for the smaller target training set *Driving*, that only comprises 8 scenes. Table 5 reports additional results on two subsets of the Scene Flow Dataset [28]: *Monkaa*, based on an animated short film, with 24 scenes, and *Driving*, containing 8 realistic driving street scenes. There are two versions for both datasets: a *Clean* version, which has no motion blur and atmospheric effects and a *Final* version, with blurring and effects. Table 5 displays results for different choices of β when using L^2-*SP* during fine-tuning. Compared to L^2, L^2-*SP* performs better on a wide range of β values, covering several orders of magnitude, showing that suboptimal choices of (α, β) still allow for substantial reductions in errors. Hence, fine-tuning SegFlow with L^2-*SP* does not require an intensive search of hyper-parameters.

	β	Monkaa		Driving	
		Clean	*Final*	*Clean*	*Final*
SegFlow-L^2		7.94	7.90	37.91	37.93
SegFlow-L^2-*SP*	1e0	7.55	7.60	34.20	35.17
SegFlow-L^2-*SP*	1e-1	**7.10**	**7.17**	31.11	30.31
SegFlow-L^2-*SP*	1e-2	7.41	7.52	**30.57**	**30.14**

Table 5: Average endpoint errors (EPEs) on the two subsets of the Scene Flow dataset according to the regularization hyper-parameter β (for $\alpha = 0.1$). The evaluations are performed on the validation set of the Monkaa and Driving datasets and use both *forward* and *backward* samples.

DSTL. The source datasets used here for DSTL are subsets of ImageNet [8] and iNaturalist [41], containing 585 categories slightly biased towards bird and dog breeds, i.e. Subset B in [7]. Inception-V3 [39] is pre-trained on this subset and then fine-tuned on six target datasets. See Table 2 for the details of datasets.

Table 3 displays that fine-tuning with L^2-*SP* improves systematically upon the L^2 baseline. We again investigate the sensitivity to hyper-parameters in Table 6 on the validation set of CUB200 [42], which is a dataset of 200 bird species. As previously, the improved performance of the L^2-*SP* penalty spans a wide area of values, and large α/β ratios are preferable to small ones.

approach	α	β	**accuracy**
DSTL - L^2	4e-5	4e-5	88.47
DSTL - L^2-*SP*	1e-4	1e-4	89.07
DSTL - L^2-*SP*	1e-3	1e-3	**89.19**
DSTL - L^2-*SP*	1e-3	1e-2	88.53
DSTL - L^2-*SP*	1e-2	1e-3	89.12
DSTL - L^2-*SP*	1e-1	1e-3	89.00

Table 6: DSTL classification accuracy using the Inception-V3 network on the CUB200 validation set according to regularization hyper-parameters.

5.3. Analysis and Discussion

Behavior across Network Structures. L^2*-SP* behaves very well across all tested network structures. EncNet is based on ResNet [13] and modified by adding a context encoding module. The segmentation branch of SegFlow is based on ResNet transformed to fully convolutional; the flow branch is FlowNetS [9], which is a variant of VGG [37]. For DSTL, Inception-V3 [39] is used. Throughout the various network structures and the diversity of problem addressed, we consistently observe better performances with fine-tuning using L^2*-SP*.

Choosing α *and* β*.* The selection of the regularization parameters α and β of Equation (3) does not require a precise search; a rule of thumb is to favor large α/β ratio rather than small ones: when transfer helps, the pre-trained weights are relevant, and α can be set to a large value without imposing detrimental constraints. As for β, which applies to the randomly initialized weights in the new layers, a large β impedes the necessary adaptation.

Bias-Variance Analysis. We propose here a simple bias-variance analysis for the tractable case of linear regression. Consider the squared loss function $J(\boldsymbol{w}) = \frac{1}{2}\|\mathbf{X}\boldsymbol{w} - \boldsymbol{y}\|^2$, where $\boldsymbol{y} \in \mathbb{R}^n$ is a vector of continuous responses, and $\mathbf{X} \in \mathbb{R}^{n\times p}$ is the matrix of predictor variables. We use the standard assumptions of the fixed design case, that is: (i) $\boldsymbol{y}$ is the realization of a random variable $\mathbf{Y}$ such that $\mathbb{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{w}^*$, $\mathbb{V}[\mathbf{Y}] = \sigma^2\mathbf{I}_n$, and $\boldsymbol{w}^*$ is the vector of true parameters; (ii) the design is fixed and orthonormal, that is, $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$. We also assume that the reference we use for L^2*-SP*, *i.e.* $\boldsymbol{w}^0$, is not far away from $\boldsymbol{w}^*$ (since it is the minimizer of the unregularized objective function on a large data set): $\boldsymbol{w}^0 = \boldsymbol{w}^* + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$, the difference between the two parameters, is supposed to be relatively small, *i.e.* $\|\boldsymbol{\varepsilon}\| \ll \|\boldsymbol{w}^*\|$.

We consider the three estimates $\widehat{\boldsymbol{w}} = \text{argmin}_{\boldsymbol{w}} J(\boldsymbol{w})$, $\widehat{\boldsymbol{w}}^{\text{L}^2} = \text{argmin}_{\boldsymbol{w}} J(\boldsymbol{w}) + \frac{\alpha}{2}\|\boldsymbol{w}\|_2^2$ and $\widehat{\boldsymbol{w}}^{\text{SP}} = \text{argmin}_{\boldsymbol{w}} J(\boldsymbol{w}) + \frac{\alpha}{2}\|\boldsymbol{w} - \boldsymbol{w}^0\|_2^2$. Their closed-form formulations, expectations and variances are given in Table 7. Without any regularization, the least squares estimate $\widehat{\boldsymbol{w}}$ is unbiased, but with the largest variance. With the L^2 regularizer, variance is decreased by a factor of $1/(1+\alpha)^2$ but the squared bias is $\|\boldsymbol{w}^*\|^2\alpha^2/(1+\alpha)^2$. The L^2*-SP* regularizer benefits from the same decrease of variance and suffers from the smaller squared bias $\|\boldsymbol{\varepsilon}\|^2\alpha^2/(1+\alpha)^2$. It is thus a better option than L^2 (provided the assumption $\|\boldsymbol{\varepsilon}\| \ll \|\boldsymbol{w}^*\|$ holds), it is also always better than the least squares estimate provided $\|\boldsymbol{\varepsilon}\| < p\sigma^2$ and otherwise better than this estimate for sufficiently small α, that is for $\alpha < 2p\sigma^2/(\|\boldsymbol{\varepsilon}\|^2 - p\sigma^2)$.

	$\widehat{\boldsymbol{w}}$	$\widehat{\boldsymbol{w}}^{\mathrm{L}^2}$	$\widehat{\boldsymbol{w}}^{\mathrm{SP}}$
closed-form	$\mathbf{X}^\mathsf{T}\boldsymbol{y}$	$\frac{1}{1+\alpha}\mathbf{X}^\mathsf{T}\boldsymbol{y}$	$\frac{1}{1+\alpha}\mathbf{X}^\mathsf{T}\boldsymbol{y} + \frac{\alpha}{1+\alpha}\boldsymbol{w}^0$
$\mathbb{E}$	$\boldsymbol{w}^*$	$\frac{1}{1+\alpha}\boldsymbol{w}^*$	$\boldsymbol{w}^* + \frac{\alpha}{1+\alpha}\boldsymbol{\varepsilon}$
$\mathbb{V}$	$\sigma^2\mathbf{I}_p$	$\left(\frac{\sigma}{1+\alpha}\right)^2\mathbf{I}_p$	$\left(\frac{\sigma}{1+\alpha}\right)^2\mathbf{I}_p$

Table 7: Three estimates of the solution of a simple linear regression problem using different regularizers, and their expectations $\mathbb{E}$ and variances $\mathbb{V}$.

6. Conclusion

This paper provides new evidences on the relevance of the L^2-*SP* regularization in transfer learning. We report experiments with three representative state-of-the-art transfer learning approaches of computer vision: EncNet [45], SegFlow [5], and DSTL [7]. Our protocol avoids any distortion or bias by handing over the experiments to the original authors of these approaches. This protocol was made possible thanks to the ease of implementation of the L^2-*SP* regularization, whose adjustment is facilitated by the relative robustness with regard to the tuning of the regularization parameter. A general safe rule is to favor values of α that are larger than β.

Our experiments demonstrate that fine-tuning with L^2-*SP* regularization, used in place of the standard weight decay, is effective and versatile: not a single comparison is in favor of fine-tuning with L^2 regularization. These conclusions are not surprising, considering that fine-tuning is motivated by assuming the proximity of the solutions to the source and target problems. Our analysis in a simplified linear setting confirms that, when this assumption is true, that is, when fine-tuning is expected to perform better than learning from scratch, L^2-*SP* is always better than L^2. We thus conclude that L^2-*SP* regularization should be the baseline when fine-tuning for transfer learning in computer vision applications.

Acknowledgments

This work was carried out with the supports of the China Scholarship Council and of a PEPS grant through the DESSTOPT project jointly managed by the National Institute of Mathematical Sciences and their Interac-

tions (INSMI) and the Institute of Information Science and their Interactions (INS2I) of the CNRS, France.

References

[1] Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, 2017.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, pages 446–461, 2014.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[5] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. SegFlow: Joint learning for video object segmentation and optical flow. In *IEEE International Conference on Computer Vision (ICCV)*, pages 686–695, 2017.

[6] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[7] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4109–4118, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2758–2766, 2015.

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[11] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10–19, 2017.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. MIT Press, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[15] Hexiang Hu, Shiyi Lan, Yuning Jiang, Zhimin Cao, and Fei Sha. FastMask: Segment multi-scale object candidates in one shot. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 991–999, 2017.

[16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017.

[17] Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1062–1071, 2018.

[18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 554–561, 2013.

[20] Hoang-An Le, Anil S Baslamisli, Thomas Mensink, and Theo Gevers. Three for one and one for three: Flow, segmentation, and surface normals. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[21] Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning (ICML)*, pages 2830–2839, 2018.

[22] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, pages 614–629, 2016.

[23] Hank Liao. Speaker adaptation of context dependent deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7947–7951. IEEE, 2013.

[24] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[25] Pauline Luc, Camille Couprie, Yann LeCun, and Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[26] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 155–163, 2018.

[27] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[28] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[29] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 891–898, 2014.

[30] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 722–729, 2008.

[31] Tsubasa Ochiai, Shigeki Matsuda, Xugang Lu, Chiori Hori, and Shigeru Katagiri. Speaker adaptive training using deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6349–6353. IEEE, 2014.

[32] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[33] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016.

[34] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[35] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[36] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):801–814, 2019.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[38] Nathan Srebro and Adi Shraibman. Rank, Trace-Norm and Max-Norm. In *Conference on Learning Theory (COLT)*, volume 5, pages 545–560. Springer, 2005.

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[40] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2015.

[41] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8769–8778, 2018.

[42] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[43] Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5075–5084, 2017.

[44] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NIPS)*, pages 3320–3328, 2014.

[45] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7151–7160, 2018.

[46] Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 528–537, 2018.

[47] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. MiCT: Mixed 3D/2D convolutional tube for human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 449–458, 2018.