Lautum Regularization for Semi-supervised Transfer Learning

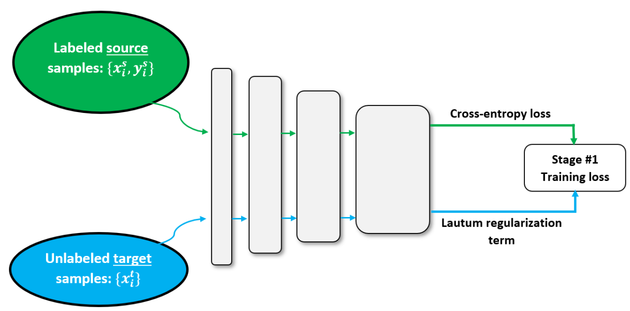
Daniel Jakubovitz
School of Electrical Engineering, Tel Aviv University,
Tel Aviv, Israel
danielshaij@mail.tau.ac.il

Miguel R. D. Rodrigues
Department of Electronic and Electrical Engineering, University College London,
London, United Kingdom
m.rodrigues@ucl.ac.uk

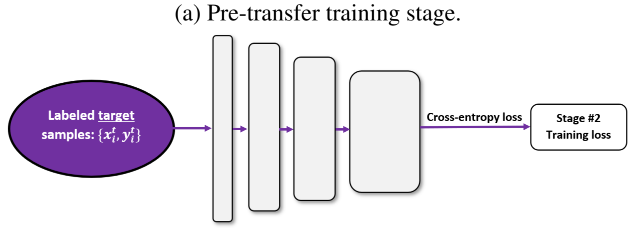
Raja Giryes
School of Electrical Engineering, Tel Aviv University,
Tel Aviv, Israel
raja@tauex.ac.il

Abstract

Transfer learning is a very important tool in deep learning as it allows propagating information from one "source dataset" to another "target dataset", especially in the case of a small number of training examples in the latter. Yet, discrepancies between the underlying distributions of the source and target data are commonplace and are known to have a substantial impact on algorithm performance. In this work we suggest a novel information theoretic approach for the analysis of the performance of deep neural networks in the context of transfer learning. We focus on the task of semi-supervised transfer learning, in which unlabeled samples from the target dataset are available during the network training on the source dataset. Our theory suggests that one may improve the transferability of a deep neural network by imposing a Lautum information based regularization that relates the network weights to the target data. We demonstrate in various transfer learning experiments the effectiveness of the proposed approach.



(a) Pre-transfer training stage.



(b) Post-transfer training stage.

Figure 1: Our semi-supervised transfer learning technique applying Lautum regularization. Omitting the blue part in the first training stage (top) gives standard transfer learning.

1. Introduction

Machine learning algorithms have lately come to the forefront of technological advancements, providing state-of-the-art results in a variety of fields [5]. However, alongside their incredible performance, these methods suffer from sensitivity to data discrepancies - any inherent difference between the training data and the test data may result in a substantial decrease in performance. Moreover, to obtain good performance a large amount of labeled data is necessary for their training. Such a substantial amount of labeled data is often either very expensive or simply unobtainable.

One popular approach to mitigate this issue is using

”transfer learning”, where training on a small labeled target dataset is improved by using information from another large labeled ”source” dataset of a different problem. A common method for transfer learning uses the result of training on the source as initialization for the training on the target, thereby improving the performance on the latter [2].

Transfer learning has been the focus of much research attention along the years. Plenty of different approaches have been proposed to encourage a more effective transfer from a source dataset to a target dataset. Many of them aim at obtaining better system robustness to environment changes, so as to allow an algorithm to perform well even under some variations in the settings (e.g. changes in lighting conditions in computer vision tasks). Sometimes this is achieved at the expense of diminishing the performance on the original task or data distribution. Other works take a more targeted approach and directly try to reduce algorithms' generalization error by decreasing the difference in their performance on specific source and target datasets [16].

The underlying assumption in transfer learning is that the source and target tasks are in some sense related. In this case, an effective technique may obtain substantial performance improvement when the relevant knowledge from the source data distribution is extracted to improve performance on the target data distribution. The effectiveness of such a transfer depends heavily on the relation between the source and target datasets as well as on the number of training samples in each. It is often the case that the target dataset has a large number of samples, though only a few of those samples are labeled. In this scenario a semi-supervised learning approach could prove to be beneficial by making good use of the available unlabeled samples for training.

Contribution. We consider the mostly unexplored case of semi-supervised transfer learning in which plenty of labeled examples from a source distribution are available along with just a few labeled examples from a target distribution; yet, we are provided also with a large number of unlabeled samples from the latter. This setup combines transfer learning and semi-supervised learning, where both aim at obtaining improved performance on a target dataset with a small number of labeled examples. Here we suggest to combine both methodologies to gain the advantage of both of them. This setting represents the case where the learned information from a large labeled source dataset is used to obtain good performance when transferring to a mostly unlabeled target set, where the unlabeled examples of the target are available at the training time on the source.

To do so, we provide a theoretical derivation that leads to a novel semi-supervised technique for transfer learning, which makes a good use of the knowledge within the source dataset to obtain improved performance on the target dataset. We take an information theoretic approach to examine the cross-entropy test loss of machine learning methods. We decompose it to several different terms that account for different aspects of its behavior. This derivation leads to a new regularization term, which we call ”Lautum regularization” as it relies on the maximization of the Lautum information [18] between unlabeled data samples drawn from the target distribution and the learned model weights.

Figure 1 provides a general illustration of our approach. Notice that in the second training stage we may also apply a semi-supervised learning technique instead of only fine-tuning, which can improve performance even further as we show in the experiments. We corroborate the effectiveness of our approach with experiments of semi-supervised transfer learning for neural networks on image classification tasks. We examine the transfer in two cases: from the MNIST dataset to the notMNIST dataset (which consists of the letters A-J in grayscale images) and from the CIFAR-10 dataset to 10 specific classes of the CIFAR-100 dataset. We compare our results to the Temporal Ensembling method [14], a state-of-the-art method for semi-supervised training (applied in a transfer learning setup), and to standard transfer learning. We demonstrate the advantage of our method and also show how it may be combined with existing semi-supervised techniques to improve performance even further.

2. Related Work

Plenty of works exist in the literature on transfer learning, semi-supervised learning and using information theory for the analysis of machine learning algorithms. We hereby overview the ones most relevant to our work.

Transfer learning. Transfer learning [19, 25] is a useful training technique when the goal is to adapt a learning algorithm, which was trained on a source dataset, to perform well on a target dataset that is potentially very different in content compared to the source. This technique can provide a significant advantage when the number of training samples in the source dataset is large compared to a small number in the target, where the knowledge extracted from the source dataset may be relevant also to the target.

The work in [30] relates to a core question in transfer learning: which layers in the network are general and which are more task specific, and precisely how transferability is affected by the distance between two tasks. In a recent work [15] an analytic theory of how knowledge is transferred from one task to another in deep linear networks is presented. A metric is given to quantify the amount of knowledge transferred between a pair of tasks.

Practical approaches for improving performance in transfer learning settings have been proposed in many works. In [28], transfer learning in the context of regression problems is examined. A transfer learning algorithm which does not assume that the support of the target distribution is contained in the support of the source distribution is proposed. This notion leads to a more flexible transfer. In [29] the framework of "learning to transfer" is proposed in order to leverage previous transfer learning experiences for better transfer between a new pair of source and target datasets. In [31] the structural relation between different visual tasks is examined in the feature space. The result is a taxonomic map that enables a more efficient transfer learning with a reduced amount of labeled data.

All of the above works differ from ours in their core approach - they address the discrepancy between the source and target data either in the input space or in the feature space, yet disregard the effect of the chosen loss function and its impact on the mitigation of this discrepancy. In contrast, our work is focused on the mathematical analysis of the cross-entropy loss which is commonly used in classification tasks. We apply our theoretical analysis in a semi-supervised transfer learning setting, which is a generally under-explored task.

Semi-supervised learning. Semi-supervised learning [33] is typically used when there is little labeled data for training, yet more unlabeled data is available. The literature on semi-supervised training is vast and describes a variety of techniques for performing effective semi-supervised learning that would make good use of the available unlabeled data in order to improve model performance. Most of these techniques rely on projecting the relation between the available labeled samples and their labels to the unlabeled samples and the model's predicted labels for them.

In [6] minimum entropy regularization is proposed. This technique modifies the cross-entropy loss used for training in order to encourage a deep neural network to make confident predictions on unlabeled data. In [8] a new framework for semi-supervised training of neural networks called "associative learning" is proposed. In this framework "associations" are made between the embeddings of the available labeled data and the unlabeled data. An optimization process is then used to encourage correct "associations", which make better use of the unlabeled data. In [24], a method is proposed for combining several different semi-supervised learning techniques using Bayesian optimization. In [20] a semi-supervised framework that allows labeled training data privacy is proposed. In this framework, knowledge is transferred from teacher models to a student model in a semi-supervised manner, thereby precluding the student from gaining access to the labeled training data which is available to the teachers.

Two recent works that employ semi-supervised training techniques are [10] and [13]. In [10] a semi-supervised deep kernel learning model is presented for regression tasks. In [13] a GAN based method is presented. It is proposed to estimate the tangent space to the learned data manifold using GANs, infer the relevant invariances and then inject these into the learned classifier during training. In [17] various semi-supervised learning algorithms are evaluated on real-world applications, yet no specific attention is paid to the transfer learning case and the effects of fine-tuning a pre-trained network.

Semi-supervised transfer learning is a generally under-explored field with just a few works focusing on it in the literature. The work in [7] examines semi-supervised transfer learning for sentiment classification. In [32] semi-supervised transfer learning is examined for different training strategies and model choices. Several observations regarding the application of existing semi-supervised methods in transfer learning settings are made.

In our experiments we compare our results to the state-of-the-art semi-supervised technique of Temporal Ensembling [14]. Temporal Ensembling relies on forming ensemble predictions during training using the outputs of the network in different training epochs under the application of various regularization and augmentation techniques. This leads to a better prediction on unlabeled data, which provides target labels for training.

Information theory and machine learning. Information Theory has lately been used to give theoretical insight into the intricacies of machine learning algorithms. In [26], the Information Bottleneck framework has been presented. This framework formalizes the trade-off between algorithm sufficiency (fidelity) and complexity. It has been analyzed in various works such as [22] and [3]. Following works [27, 23] made the specific relation to deep learning, explicitly applying the principles of the information bottleneck to deep neural networks. In [21] several of the claims from [23] are examined and challenged. In [4] useful methods for the computation of information theoretic quantities are proposed for several deep neural network models.

The closest work to ours is [1] in which an information theoretic approach is used in order to decompose the cross-entropy train loss of a machine learning algorithm into several separate terms. It is suggested that overfitting the training data is mathematically encapsulated in the mutual information between the training data labels and the learned model weights, i.e. this mutual information essentially represents the ability of a neural network to *memorize* the training data [9]. Consequently, a regularizer that prevents overfitting is proposed, and initial results of its efficiency are presented. However, unlike this work we propose a different decomposition of the cross-entropy *test* loss and make the relation to semi-supervised transfer learning. We propose a regularizer which leads to an improved semi-supervised transfer technique and present experimental results that corroborate our theoretical analysis.

3. The cross-entropy loss - an information theory perspective

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a training set with N training samples that is used to train a learning algorithm with a set of weights w. We assume that given $\mathcal{D}$, the learning algorithm selects a particular hypothesis from the hypothesis class according to the distribution $p(w|\mathcal{D})$. In the case of a neural network, selecting the hypothesis is equivalent to training the network on the data.

We denote by $w_{\mathcal{D}}$ the model parameters which were learned using the training set $\mathcal{D}$, and by $f(y|x, w_{\mathcal{D}})$ the learned classification function which given the weights $w_{\mathcal{D}}$ and a D-dimensional input $x \in \mathbb{R}^D$ calculates a K-dimensional output $y \in \mathbb{R}^K$ that represents the probability of any one of the K possible classes to be the correct class of the given input. The learned classification function is tested on data drawn from the true underlying distribution $p(x, y)$. Ideally, the learned classification function $f(y|x, w_{\mathcal{D}})$ would highly resemble the ground-truth classification $p(y|x)$.

With these definitions, we turn to analyze the cross-entropy loss used predominantly in classification tasks. In our derivations we used several information theoretic measures which we present hereafter. Let X, Y be two random variables with respective probability density functions $p(x), p(y)$. We present the definitions of three information theoretic measures which are relevant to our derivations.

Definition 1 (Mutual information) *The mutual information between X and Y is defined by*

$$I(X;Y) = \iint p(x,y) \log \left\{ \frac{p(x,y)}{p(x)p(y)} \right\} dxdy. \quad (1)$$

The Mutual information captures the dependence between two random variables. It is the Kullback-Leibler divergence between the joint distribution and the product of the marginal distributions. The following is the Lautum information:

Definition 2 (Lautum information) *The Lautum information between X and Y is*

$$L(X;Y) = \iint p(x)p(y) \log \left\{ \frac{p(x)p(y)}{p(x,y)} \right\} dxdy. \quad (2)$$

This measure is the Kullback-Leibler divergence between the product of the marginal distributions and the joint distribution. Similar to the mutual information, the Lautum information is related to the dependence between two random variables. However, it has different properties than the mutual information, as outlined in [18]. The last definition is of the differential entropy of a random variable.

Definition 3 (Differential entropy) *The differential entropy of a random variable X is defined by*

$$H(X) = -\int p(x) \log p(x) dx. \quad (3)$$

Main theoretical result. Having these definitions, we turn to present our main theoretical result.

Theorem 1 *For a classification task with ground-truth distribution $p(y|x)$, training set $\mathcal{D}$, learned parameters $w_{\mathcal{D}}$ and learned classification function $f(y|x, w_{\mathcal{D}})$, the expected cross-entropy loss of a machine learning algorithm on the test distribution is equal to*

$$\mathbb{E}_{w_{\mathcal{D}}} \left\{ KL(p(y|x) || f(y|x, w_{\mathcal{D}})) \right\} + H(y|x) - L(w_{\mathcal{D}}; x). \quad (4)$$

Note that KL signifies the Kullback-Leibler divergence and that we treat the training set $\mathcal{D}$ as a fixed parameter whereas $w_{\mathcal{D}}$ and the examined test data (x, y) are treated as random variables. In accordance with Theorem 1, the three terms that compose the expected cross-entropy test loss represent three different aspects of the loss of a learning algorithm performing a classification task:

- **Classifier mismatch** $\mathbb{E}_{\mathbf{w}_{\mathcal{D}}}\mathbf{KL}\left(\mathbf{p}(\mathbf{y}|\mathbf{x})||\mathbf{f}(\mathbf{y}|\mathbf{x}, \mathbf{w}_{\mathcal{D}})\right)$: measures the deviation of the learned classification function $f(y|x, w_{\mathcal{D}})$ from the true classification of the data $p(y|x)$. It is measured by the KL-divergence, which is averaged over all possible instances of w given the training set $\mathcal{D}$. This term essentially measures the ability of the weights learned from $\mathcal{D}$ to represent the true distribution of the data.

- **Intrinsic Bayes error** $\mathbf{H}(\mathbf{y}|\mathbf{x})$: represents the inherent uncertainty of the labels given the data samples.

- **Lautum information** between $w_{\mathcal{D}}$ and x $\mathbf{L}(\mathbf{w}_{\mathcal{D}}; \mathbf{x}) = \mathbb{E}_{\mathbf{w}_{\mathcal{D}}}\{\mathbf{KL}(\mathbf{p}(\mathbf{x})||\mathbf{p}(\mathbf{x}|\mathbf{w}_{\mathcal{D}}))\}$: represents the dependence between $w_{\mathcal{D}}$ and x. It essentially measures how much $p(x|w_{\mathcal{D}})$ deviates from $p(x)$ on average over the possible values of $w_{\mathcal{D}}$.

Our formulation in Theorem 1 suggests that a machine learning algorithm, which is trained relying on empirical risk minimization, implicitly aims at maximizing the Lautum information $L(w_{\mathcal{D}}; x)$ in order to minimize the cross-entropy loss. At the same time, the algorithm aspires to minimize the KL-divergence between the learned classification function and the ground-truth classification of the data. The intrinsic Bayes error cannot be minimized and remains the inherent uncertainty of the task.

Namely, the formulation in (4) suggests that encouraging a larger Lautum information between the data samples and the learned model weights would be beneficial for reducing the model's test error on unseen data drawn from $p(x, y)$.

We hereby present the main steps of the proof of Theorem 1. The full proof is given in Appendix A.

Proof Sketch. The expected cross-entropy loss of the learned classification function $f(y|x, w_{\mathcal{D}})$ on the test distribution $p(x, y)$ is given by

$$\mathbb{E}_{(x,y)\sim p(x,y)}\mathbb{E}_{w\sim p(w|\mathcal{D})}\{-\log f(y|x, w)\}. \quad (5)$$

Explicitly, (5) can be written as [1]

$$-\iiint p(x, y)p(w|\mathcal{D}) \log f(y|x, w)dxdydw. \quad (6)$$

We henceforth omit the multiple integral signs and the multiple differentials for convenience where the integration variables are implicit from the context.

To compare the learned classifier with the true distribution of the data, we further develop (6) as follows:

$$= -\int p(x, y)p(w|\mathcal{D}) \log \left\{ \frac{f(y|x, w)}{p(y|x, w)} p(y|x, w) \right\} \quad (7)$$

$$\begin{aligned} = &-\int p(x, y)p(w|\mathcal{D}) \log \left\{ \frac{f(y|x, w)}{p(y|x, w)} \right\} \\ &- \int p(x, y)p(w|\mathcal{D}) \log p(y|x, w). \end{aligned} \quad (8)$$

Using the definitions of the differential entropy and Lautum information of random variables, we obtain the expression for the expected cross-entropy loss on the test distribution in (4).

4. Lautum information based semi-supervised transfer learning

We turn to show how we may apply our developed theory on the task of semi-supervised transfer learning. In standard transfer learning, which consists of pre-transfer and post-transfer stages, a neural network is trained on a fully labeled source dataset and then fine-tuned on a smaller labeled target dataset. In semi-supervised transfer learning, which we study here, we assume that an additional large set of unlabeled examples from the target distribution is available during training on the source data.

Semi-supervised transfer learning is highly beneficial in scenarios where the available target dataset is only partially annotated. Using the unlabeled part of this dataset, which is usually substantially bigger than the labeled part, has the potential of considerably improving the obtained performance. Thus, if this unlabeled part is a-priori available, then using it from the beginning of training can potentially improve the results.

[1]Since the values of y are discrete it is more accurate to sum instead of integrate over them. Yet, for the simplicity of the proof we present the derivations using integration.

For using the unlabeled samples of the target dataset during the pre-transfer training on the source dataset, we leverage the formulation in (4). Considering its three terms, it is clear that by using unlabeled samples the classifier mismatch cannot be minimized due to the lack of labels for these samples [2]; the intrinsic Bayes error is a characteristic of the task and cannot be minimized either; yet, the Lautum information does not depend on the labels and can therefore be calculated and maximized.

When the Lautum information is calculated between the model weights and data samples drawn from the target distribution, its maximization would encourage the learned weights to better relate to these samples, and by extension to better relate to the underlying probability distribution from which they were drawn. Therefore, it is expected that an enlarged Lautum information will yield an improved performance on the target's test set.

Accordingly, we aim at maximizing $L(w_{\mathcal{D}}; x)$ during training. When performing the pre-transfer training, the cross-entropy loss is calculated using the labeled source samples. Maximizing $L(w_{\mathcal{D}}; x)$, which is based on samples drawn from the target distribution, before the transfer would make the learned weights more inclined towards good performance on the target set right from the beginning.

In the post-transfer stage, the cross-entropy loss is calculated using the labeled target samples, and therefore implicitly maximizes $L(w_{\mathcal{D}}; x)$ by itself. We have empirically observed that explicitly maximizing the Lautum information between the unlabeled target data and the model weights during the post-transfer training in addition to (or instead of) during pre-transfer training does not lead to improved results.

To summarize, our semi-supervised transfer learning approach optimizes two goals at the same time: (i) minimizing the classifier mismatch $\mathbb{E}_{w_{\mathcal{D}}}\{KL\left(p(y|x)||f(y|x, w_{\mathcal{D}})\right)\}$, which is achieved using the labeled data both for the source and the target datasets during pre-transfer and post-transfer training respectively; and (ii) maximizing the Lautum information $L(w_{\mathcal{D}}; x)$, which is achieved explicitly using the unlabeled target data during pre-transfer training, and in the post-transfer stage implicitly through the minimization of the cross-entropy loss which is evaluated on the labeled target data. Figure 1 summarizes our training scheme.

4.1. Estimating the Lautum information

Following the above, we are interested in using the Lautum information as a regularization term, which we henceforth refer to as "Lautum regularization". Since computing the Lautum information between two random variables requires knowledge of their probability distribution functions

[2]Potentially, we could perform pre-transfer training also using the labeled target samples (e.g. with weight sharing). Yet, due to the size imbalance we do not perform it.

(which are high-dimensional and hard to estimate), we assume that $w_{\mathcal{D}}$ and x are jointly Gaussian with zero-mean. Even though this may seem like an arbitrary assumption, it nevertheless provides ease of computation and good experimental results as shown hereafter in Section 5.

Since we only have one instance of the network weights at any specific point during training, we use the network features as a proxy for the network weights in the calculation of the Lautum information, instead of using the weights themselves. Namely, we use the network's output (its pre-softmax logits) when the input is x as a proxy for the network weights $w_{\mathcal{D}}$. This way we have in every training iteration a number of samples equivalent to the size of our training mini-batch, instead of only one sample which would not allow any stable estimation to be made.

As shown in [18], the Lautum information between two jointly Gaussian random variables (w, x) with covariance

$$\begin{bmatrix} \Sigma_w & \Sigma_{wx} \\ \Sigma_{xw} & \Sigma_x \end{bmatrix}, \tag{9}$$

where $\Sigma_x \succ 0$ and $\Sigma_w \succ 0$, is given by

$$\begin{aligned} L(w;x) = &\log\left\{\det(I - \Sigma_x^{-1}\Sigma_{xw}\Sigma_w^{-1}\Sigma_{wx})\right\} \\ &+ 2tr((I - \Sigma_x^{-1}\Sigma_{xw}\Sigma_w^{-1}\Sigma_{wx})^{-1} - I). \end{aligned} \tag{10}$$

The covariance matrix of our target dataset Σ_x is evaluated once before training using the entire target training set, whereas $\Sigma_w, \Sigma_{wx}, \Sigma_{xw}$ are evaluated during training using the current mini-batch in every iteration. All of these matrices are estimated using standard sample covariance estimation based on the current mini-batch in every iteration, e.g.

$$\Sigma_x = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} (x_i - \mu_x)(x_i - \mu_x)^T, \tag{11}$$

and

$$\Sigma_{xw} = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} (x_i - \mu_x)(w_i - \mu_w)^T, \tag{12}$$

where

$$\mu_x = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} x_i, \qquad \mu_w = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} w_i$$

represent the sample mean values of x and w respectively (i.e. their average values in the current mini-batch) and N_{batch} denotes the mini-batch size. The dimensions of these matrices are $\Sigma_x \in \mathbb{R}^{D\times D}, \Sigma_w \in \mathbb{R}^{K\times K}, \Sigma_{xw} \in \mathbb{R}^{D\times K}, \Sigma_{wx} \in \mathbb{R}^{K\times D}$. Note that $\Sigma_{wx} = \Sigma_{xw}^T$ hence only one of these matrices has to be calculated from the samples in every training iteration. Since these are high-dimensional matrices, obtaining a numerically stable sample estimate would require a large amount of data, which would require the use of a very large mini-batch. This constraint poses both a hardware problem (since standard GPUs cannot fit mini-batches of thousands of examples) and a potential generalization degradation (as smaller mini-batches have been linked to improved generalization [11]).

To overcome this issue, we use a standard exponentially decaying moving-average estimation of the three matrices $\Sigma_w, \Sigma_{wx}, \Sigma_{xw}$ in order to obtain numerical stability. We denote by α the decay rate, and get the following update rule for the three covariance matrices in every training iteration:

$$\Sigma_{(n)} = \alpha\Sigma_{(n-1)} + (1-\alpha)\Sigma_{batch}, \tag{13}$$

where n denotes the training iteration and Σ_{batch} denotes the sample covariance matrix calculated using the current mini-batch. Using an exponentially decaying moving-average calculation is particularly of high importance for Σ_w, which is inverted to compute the Lautum regularization term.

4.2. Training with Lautum regularization

Our loss function for pre-transfer training is therefore:

$$Loss = \sum_{i=1}^{N}\sum_{k=1}^{K} -y_{ik}^s \log f_k(x_i^s) - \lambda L(w_{\mathcal{D}}; x^t). \tag{14}$$

Note that the the cross-entropy loss is calculated using labeled samples from the source training set (which we denote by the s superscript) whereas the Lautum regularization term is calculated using unlabeled samples from the target training set (which we denote by the t superscript). Also note that y_i represents the ground truth label of the sample x_i; $f(x_i)$ represents the network's estimated post softmax label for that sample; and $L(w_{\mathcal{D}}; x)$ is calculated according to (10). We emphasize that the Lautum regularization term is subtracted and not added to the cross-entropy loss since we aim at *maximizing* the Lautum information during training.

Our loss function for post-transfer training consists of a standard cross-entropy loss:

$$Loss = \sum_{i=1}^{N}\sum_{k=1}^{K} -y_{ik}^t \log f_k(x_i^t). \tag{15}$$

Note that at this stage the cross-entropy loss, which is calculated using labeled target samples, inherently includes the Lautum term of the target data (see Theorem 1). Our training scheme is illustrated in Figure 1.

5. Experiments

In order to demonstrate the advantages of semi-supervised transfer learning with Lautum regularization we perform several experiments on image classification tasks using deep neural networks (though our theoretical derivations apply also to other machine learning algorithms).

5.1. Experimental setup

We train deep neural networks and perform transfer learning from our original source dataset to our target dataset. In our experiments we use the original labeled source training set as is and split the target training set into two parts. The first part is very small and contains labeled samples, whereas the second part consists of the remainder of the target training set and contains unlabeled samples only (the labels are discarded). The performance is evaluated on the post transfer accuracy on the target test set.

As presented in Section 4.2, our training consists of two stages which are illustrated in Figure 1. First, we train the network using our fully labeled source training set while using Lautum regularization with the unlabeled samples from our target training set. We use the same mini-batch size both for the calculation of the cross-entropy loss (using labeled source samples) and for the computation of the Lautum regularization term (using unlabeled target samples). We also use an exponentially decaying moving average to obtain numerical stability in the estimation of the the covariance matrices $\Sigma_w, \Sigma_{wx}, \Sigma_{xw}$, as outlined in (13). The matrix Σ_x is calculated once before training and remains constant all throughout it.

Second, we perform a transfer to the target set by training (fine-tuning) the entire network using the labeled samples from the target training set, where the mini-batch size remains the same as before. As in [32], we fine-tune the entire network since this best fits the settings of semi-supervised learning. As mentioned above, we do not apply the Lautum regularization at this stage as we empirically found that it does not improve the results.

We examine 4 different methods of transfer learning: (1) standard supervised transfer which uses the labeled samples only. (2) Temporal Ensembling semi-supervised learning as outlined in [14], applied in a transfer learning setting. Temporal Ensembling is applied in the post-transfer training stage, where the weights are initialized using the network trained on the source dataset. (3) Lautum regularization - our technique as described in Section 4. (4) The combination of Lautum regularization and Temporal Ensembling. Note that the Lautum regularization is applied before the transfer whereas Temporal Ensembling is applied after it.

We perform our experiments on the MNIST and CIFAR-10 datasets. For MNIST we examine the transfer to the notMNIST dataset, which consists of 10 classes representing the letters A-J. The notMNIST dataset is similar to the MNIST dataset in its grayscale styling and image size, yet it differs in content. For CIFAR-10 we examine the transfer to 10 specific classes of the CIFAR-100 dataset (specifically, classes 0, 10, 20,...,90 of CIFAR-100, which we reclassified as classes 0, 1, 2,...,9 respectively). These CIFAR-100 classes are different than the corresponding CIFAR-10 ones in content. For example, class 0 in CIFAR-10 represents airplanes whereas class 0 in CIFAR-100 represents beavers etc.

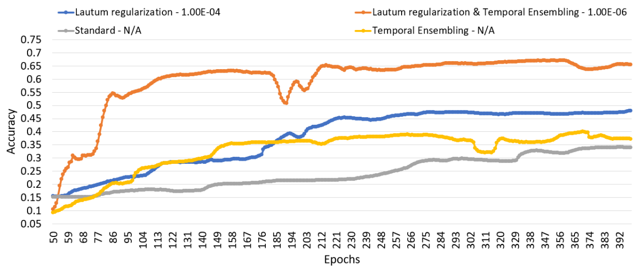
Both in the MNIST $\rightarrow$ notMNIST case and the CIFAR-10 $\rightarrow$ CIFAR-100 (10 classes) case we use the same CNN as in [14]. The architecture of the network is illustrated in Appendix B. The results of an ablation study we have conducted are in Appendix C.

5.2. MNIST to notMNIST results

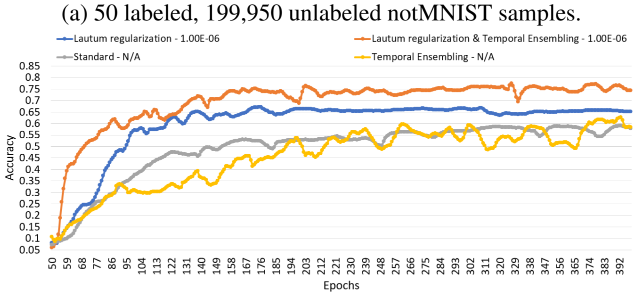
In order for the input images to fit the network's input we resized the MNIST and notMNIST images to 32x32 pixels and transformed each of them to RGB format. Training was done using an Adam optimizer [12] and a mini-batch size of 50 inputs. With this network and using standard supervised training on the entire MNIST dataset we obtained a test accuracy of 99.01% on MNIST.

For each of the 4 transfer learning methods we examined three different splits of the notMNIST training dataset which consists of 200,000 samples to an unlabeled part and a labeled part: (1) unlabeled part of 199,950 samples and a labeled part of 50 samples; (2) unlabeled part of 199,900 samples and a labeled part of 100 samples; (3) unlabeled part of 199,800 samples and a labeled part of 200 samples. All three options use very few labeled samples in order to fairly represent realistic semi-supervised learning scenarios - in all three options 99.9% or more of the training data is unlabeled. We used a decay rate of $\alpha = 0.999$ for the exponentially decaying moving average estimation of the 3 covariance matrices $\Sigma_w, \Sigma_{wx}, \Sigma_{xw}$, and a different value of λ (which controls the weight of the Lautum regularization) in each scenario which we found to provide a good balance between the cross-entropy loss and the Lautum regularization. Using these settings we obtained the results shown in Figure 2. Note that this figure shows the notMNIST test accuracy in the post-transfer epochs only. Also note that the appropriate values of λ tend to be quite small as the values of the Lautum information tend to be large.

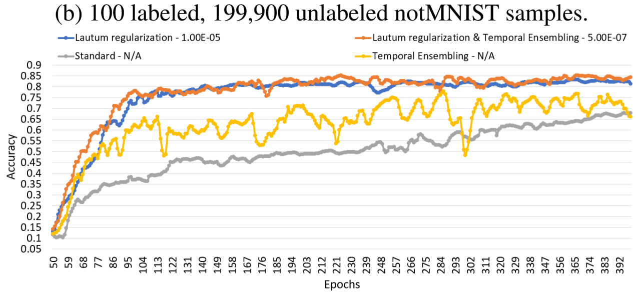
It is noticeable from the results that using Lautum regularization encourages an initial post-transfer fast rise in the test accuracy on the target set. In addition, the combination of Lautum regularization and Temporal Ensembling yields even improved results, though the gap compared to Lautum regularization alone is closed as training continues in the case of 200 labeled samples. In general, the Temporal Ensembling method by itself does not yield very competitive results compared to standard training on the target training set. We conjecture the reason for this being this method's lack of ability to leverage semi-supervised fine-tuning for improved performance in a transfer learning setting.



(a) 50 labeled, 199,950 unlabeled notMNIST samples.

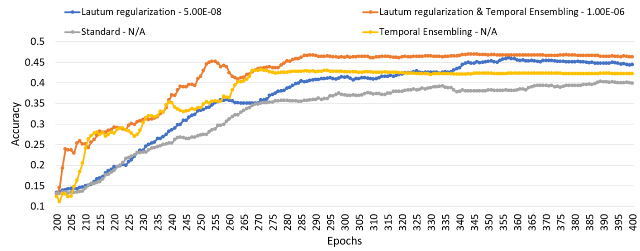


(b) 100 labeled, 199,900 unlabeled notMNIST samples.

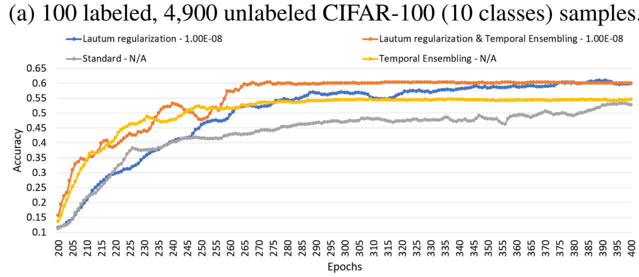


(c) 200 labeled, 199,800 unlabeled notMNIST samples.

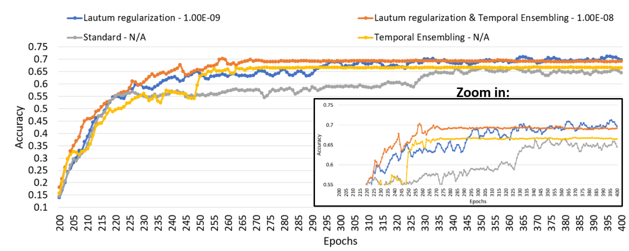
Figure 2: Post-transfer test accuracy on notMNIST test set for MNIST $\rightarrow$ notMNIST semi-supervised transfer learning. Results are given for different amounts of labeled samples from the notMNIST dataset. The legend depicts: training mode, λ value (when Lautum regularization is used).



(a) 100 labeled, 4,900 unlabeled CIFAR-100 (10 classes) samples.



(b) 200 labeled, 4,800 unlabeled CIFAR-100 (10 classes) samples.



(c) 500 labeled, 4,500 unlabeled CIFAR-100 (10 classes) samples.

Figure 3: Post-transfer test accuracy on CIFAR-100 (10 classes) test set for CIFAR-10 $\rightarrow$ CIFAR-100 (10 classes) semi-supervised transfer learning. Results are given for different amounts of labeled samples from the CIFAR-100 (10 classes) dataset. The legend depicts: training mode, λ value (when Lautum regularization is used).

5.3. CIFAR-10 to CIFAR-100 (10 classes) results

For CIFAR-10 $\rightarrow$ CIFAR-100 (10 classes) we used the CNN from [14] as we did for MNIST $\rightarrow$ notMNIST. Training was done using an Adam optimizer [12] and a mini-batch size of 100 inputs. With this network and using standard supervised training on the entire CIFAR-10 dataset we obtained a test accuracy of 85.09% on CIFAR-10.

Our target set consists of 10 classes of the CIFAR-100 dataset. Accordingly, our training target set consists of 5,000 samples and our test target set consists of 1,000 samples. We examined the same 4 transfer learning techniques as in the MNIST $\rightarrow$ notMNIST case, where for each we examined three different splits of the CIFAR-100 (10 classes) training dataset to an unlabeled part and a labeled part: (1) unlabeled part of 4,900 samples and a labeled part of 100 samples; (2) unlabeled part of 4,800 samples and a labeled part of 200 samples; (3) unlabeled part of 4,500 samples and a labeled part of 500 samples. All three options use a small number of labeled samples in order to fairly represent realistic semi-supervised learning scenarios - in all three options 90% or more of the data is unlabeled. We used a decay rate of $\alpha = 0.999$ for the exponentially decaying moving average estimation of the 3 covariance matrices $\Sigma_w, \Sigma_{wx}, \Sigma_{xw}$, and a different value of λ (which controls the weight of the Lautum regularization) in each scenario which we found to provide a good balance between the cross-entropy loss and the Lautum regularization. Using these settings we obtained the results shown in Figure 3. Note that this figure shows the CIFAR-100 (10 classes) test accuracy in the post-transfer epochs only. Also note that in this case as well the appropriate values of λ tend to be quite small as the values of the Lautum information tend to be large.

Similar to the MNIST $\rightarrow$ notMNIST case, it is noticeable from the results that using Lautum regularization in the CIFAR-10 $\rightarrow$ CIFAR-100 (10 classes) case improves the post-transfer performance on the target test set and outperforms Temporal Ensembling in all the three examined target training set splits.

6. Conclusions

We proposed a new semi-supervised transfer learning approach for machine learning algorithms that use the cross-entropy loss. Our approach is backed by information theoretic derivations and exemplifies how one could make good use of unlabeled samples along with just a few labeled samples to improve performance on the target dataset. Our approach relies on the maximization of the Lautum information between unlabeled samples from the target set and an algorithm's learned features by using the Lautum information as a regularization term. As shown, the maximization of the Lautum information minimizes the cross-entropy test loss on the target set and thereby improves performance as indicated by our experimental results. We have also shown that our approach surpasses the performance of a prominent state-of-the-art semi-supervised learning technique in a transfer learning setting.

Future work will focus on alternative approximations of the Lautum information which could potentially yield better performance or reduce the additional computational overhead it introduces. In addition, our formulation has the potential to be applied in other tasks as well, such as multi-task learning or domain adaptation. Incorporating techniques to mitigate the effects of training using an imbalanced dataset could also be of interest. We defer these directions to future research.

Acknowledgment. This work was supported by the ERC-StG SPADE grant.

A. Proof of Theorem 1

Let us reiterate Theorem 1 before formally proving it.

Theorem 1 *For a classification task with ground-truth distribution $p(y|x)$, training set $\mathcal{D}$, learned parameters $w_{\mathcal{D}}$ and learned classification function $f(y|x, w_{\mathcal{D}})$, the expected cross-entropy loss of a machine learning algorithm on the test distribution is equal to*

$$\mathbb{E}_{w_{\mathcal{D}}}\left\{KL(p(y|x)||f(y|x,w_{\mathcal{D}}))\right\} + H(y|x) - L(w_{\mathcal{D}};x). \tag{16}$$

Proof. The expected cross-entropy loss of the learned classification function $f(y|x, w_{\mathcal{D}})$ on the test distribution $p(x, y)$ is given by

$$\mathbb{E}_{(x,y)\sim p(x,y)}\mathbb{E}_{w\sim p(w|\mathcal{D})}\{-\log f(y|x,w)\}. \tag{17}$$

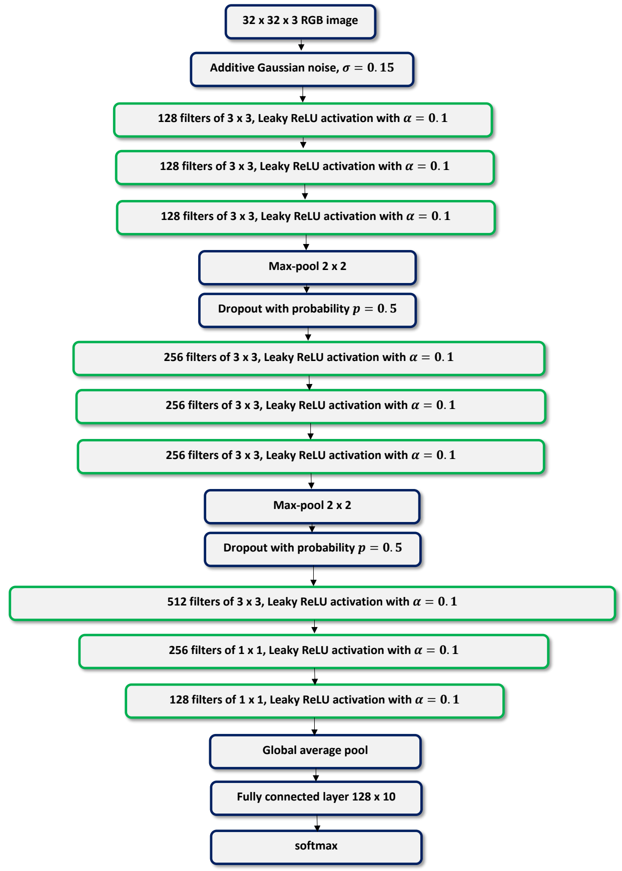


Figure 4: The network architecture used in our experiments.

Explicitly, (17) can be written as [3]

$$-\iiint p(x,y)p(w|\mathcal{D})\log f(y|x,w)dxdydw. \tag{18}$$

We henceforth omit the multiple integral signs and the multiple differentials for convenience where the integration variables are implicit from the context.

To compare the learned classifier with the true distribution of the data we develop (18) further as follows:

$$= -\int p(x,y)p(w|\mathcal{D})\log\left\{\frac{f(y|x,w)}{p(y|x,w)}p(y|x,w)\right\} \tag{19}$$

$$\begin{aligned} = &-\int p(x,y)p(w|\mathcal{D})\log\left\{\frac{f(y|x,w)}{p(y|x,w)}\right\} \\ &-\int p(x,y)p(w|\mathcal{D})\log p(y|x,w). \end{aligned} \tag{20}$$

[3] Since the values of y are discrete it is more accurate to sum instead of integrate over them. Yet, for the simplicity of the proof we present the derivations using integration.

Let us denote the two obtained expressions by † and ⋆, and the model weights learned from the training set $\mathcal{D}$ as $w_{\mathcal{D}}$. We get the following for (20):

$$\begin{aligned} &= \underbrace{- \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ \frac{f(y|x,w_{\mathcal{D}})}{p(y|x,w_{\mathcal{D}})} \right\}}_{\dagger} \\ & \underbrace{- \int p(x,y)p(w_{\mathcal{D}}) \log p(y|x,w_{\mathcal{D}})\,.}_{\star} \end{aligned} \tag{21}$$

We separate the derivations of the two terms in (21). First, we develop the term ⋆ further:

$$\star = - \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ \frac{p(x,y,w_{\mathcal{D}})}{p(x,w_{\mathcal{D}})} \right\} \tag{22}$$

$$\begin{aligned} = &- \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ \frac{p(x,y,w_{\mathcal{D}})}{p(x,y)p(w_{\mathcal{D}})} \right\} \\ &- \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ p(x,y)p(w_{\mathcal{D}}) \right\} \\ &- \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ \frac{p(x)p(w_{\mathcal{D}})}{p(x,w_{\mathcal{D}})} \right\} \\ &+ \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ p(x)p(w_{\mathcal{D}}) \right\}. \end{aligned} \tag{23}$$

Using the definitions of the differential entropy and the Lautum information we can reformulate (23) as follows:

$$\begin{aligned} \star = &L(w_{\mathcal{D}};(x,y)) \\ &+ H(w_{\mathcal{D}}) + H(x,y) \\ &- L(w_{\mathcal{D}};x) \\ &- [H(x) + H(w_{\mathcal{D}})] \end{aligned} \tag{24}$$

$$= L(w_{\mathcal{D}};(x,y)) + H(y|x) - L(w_{\mathcal{D}};x). \tag{25}$$

We next analyze the expression in †:

$$\dagger = - \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ \frac{f(x,y,w_{\mathcal{D}})}{p(x,y,w_{\mathcal{D}})} \cdot \frac{p(x,y)p(w_{\mathcal{D}})}{p(x,y)p(w_{\mathcal{D}})} \right\} \tag{26}$$

$$\begin{aligned} &= \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ \frac{p(x,y,w_{\mathcal{D}})}{p(x,y)p(w_{\mathcal{D}})} \right\} \\ &+ \int p(x,y)p(w_{\mathcal{D}}) \log \left\{ \frac{p(x,y)p(w_{\mathcal{D}})}{f(x,y,w_{\mathcal{D}})} \right\} \end{aligned} \tag{27}$$

$$\begin{aligned} &= -L(w_{\mathcal{D}};(x,y)) + KL(p(x,y)||f(x,y|w_{\mathcal{D}})) \\ &= -L(w_{\mathcal{D}};(x,y)) + KL(p(y|x)||f(y|x,w_{\mathcal{D}})). \end{aligned} \tag{28}$$

Plugging the expressions we got for ⋆ from (25) and for † from (28) into (21) we obtain the expression in (16).

B. The CNN architecture used in our experiments

As described in the paper, both in the MNIST → notMNIST case and the CIFAR-10 → CIFAR-100 (10 classes) case we used the same CNN as in [14]. The architecture is illustrated in Figure 4.

C. Ablation study

Lautum regularization is applied during pre-transfer training on the source data, which is done using a standard cross-entropy loss. The hyper-parameter λ controls the weight of the Lautum regularization in the loss function used for pre-transfer training.

The influence of λ on the post-transfer notMNIST test accuracy in the MNIST → notMNIST case when Lautum regularization is applied is depicted in Figure 5.

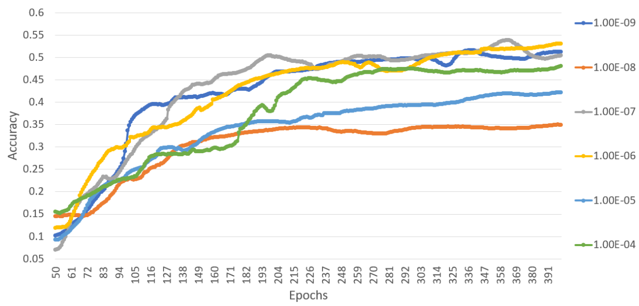
The influence of λ on the post-transfer notMNIST test accuracy in the MNIST → notMNIST case when Lautum regularization and Temporal Ensembling [14] are applied is depicted in Figure 6.

The influence of λ on the post-transfer CIFAR-100 (10 classes) test accuracy in the CIFAR-10 → CIFAR-100 (10 classes) case when Lautum regularization is applied is depicted in Figure 7.

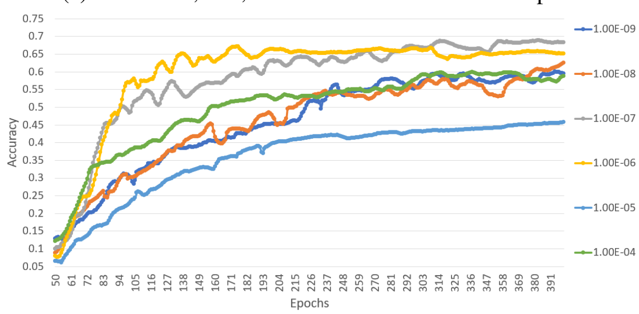
The influence of λ on the post-transfer CIFAR-100 (10 classes) test accuracy in the CIFAR-10 → CIFAR-100 (10 classes) case when Lautum regularization and Temporal Ensembling [14] are applied is depicted in Figure 8.

The results show that the dependence on the value of λ is larger when there are less labeled samples available for training in the target set. We emphasize that the values of λ that provide a good balance with the cross-entropy loss, which is evaluated on labeled samples from the source data, are typically quite small, as the values of the Lautum information tend to be large. In general, when the value of λ is too large or too small the obtained performance on the target test set is sub-optimal. However, the right value of λ provides a substantial improvement in performance. Even though in the main paper we have shown the results for different values of λ, one may choose just a single value per dataset. For example, the value of $\lambda = 10^{-6}$ in the MNIST → notMNIST case and the value of $\lambda = 10^{-8}$ in the CIFAR-10 → CIFAR-100 (10 classes) case yield a substantial performance improvement for all the examined amounts of labeled samples.

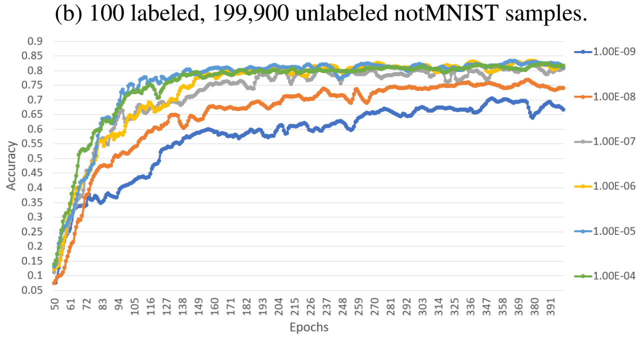
In addition, note that in the case of CIFAR-10 → CIFAR-100 (10 classes) the benefit of combining Lautum regularization with Temporal Ensembling, compared to only using Lautum regularization, is in the rapid convergence to good performance rather than a substantial increase in accuracy.



(a) 50 labeled, 199,950 unlabeled notMNIST samples.

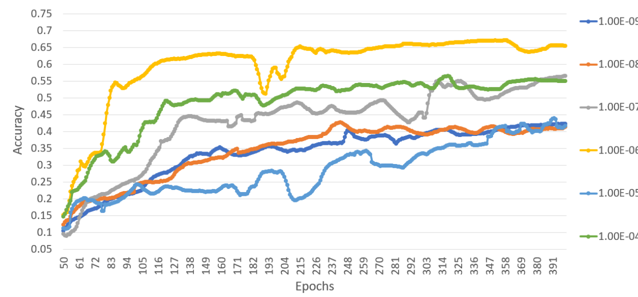


(b) 100 labeled, 199,900 unlabeled notMNIST samples.

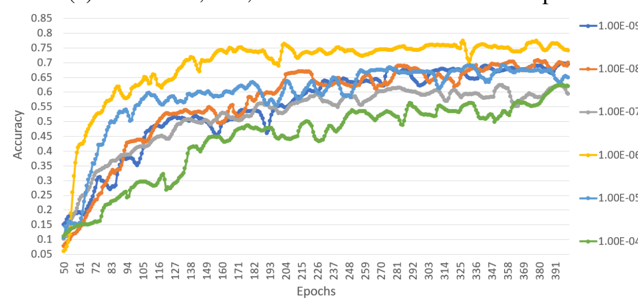


(c) 200 labeled, 199,800 unlabeled notMNIST samples.

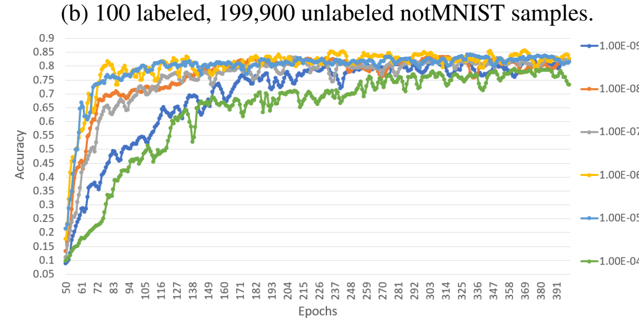
Figure 5: Influence of the hyper-parameter λ on post-transfer test accuracy on the notMNIST test set for MNIST $\rightarrow$ notMNIST semi-supervised transfer learning **with Lautum regularization**. Results are given for different amounts of labeled samples from the notMNIST dataset. The legend depicts the value of λ.



(a) 50 labeled, 199,950 unlabeled notMNIST samples.

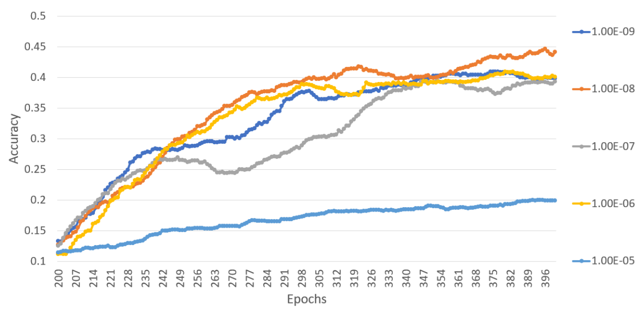


(b) 100 labeled, 199,900 unlabeled notMNIST samples.

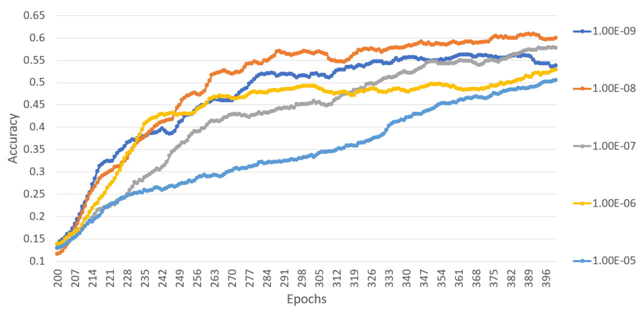


(c) 200 labeled, 199,800 unlabeled notMNIST samples.

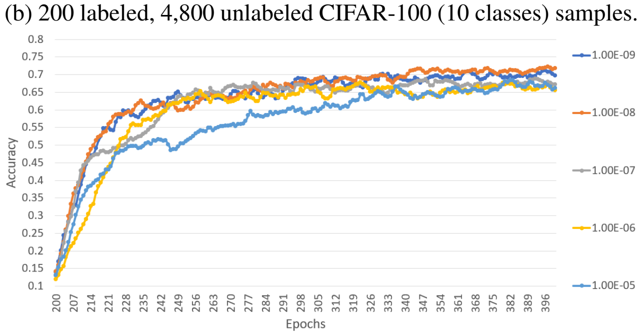
Figure 6: Influence of the hyper-parameter λ on post-transfer test accuracy on the notMNIST test set for MNIST $\rightarrow$ notMNIST semi-supervised transfer learning **with Lautum regularization and Temporal Ensembling**. Results are given for different amounts of labeled samples from the notMNIST dataset. The legend depicts the value of λ.



(a) 100 labeled, 4,900 unlabeled CIFAR-100 (10 classes) samples.

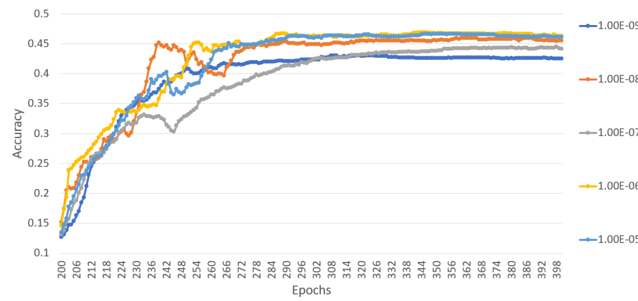


(b) 200 labeled, 4,800 unlabeled CIFAR-100 (10 classes) samples.

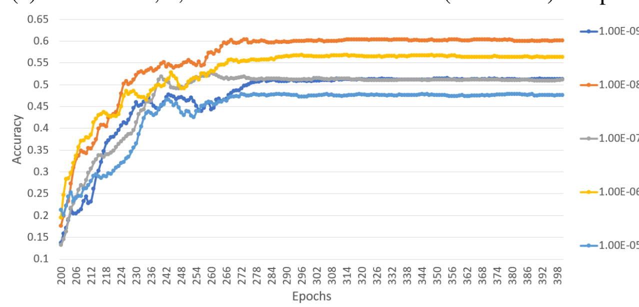


(c) 500 labeled, 4,500 unlabeled CIFAR-100 (10 classes) samples.

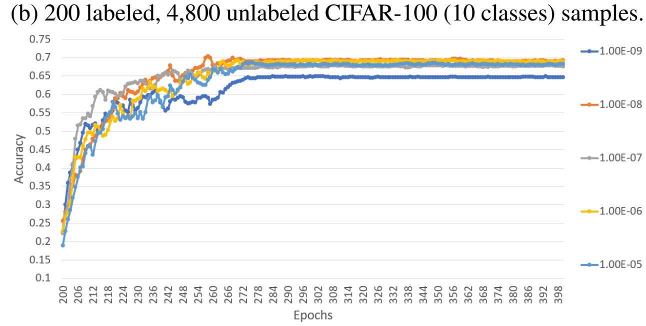
Figure 7: Influence of the hyper-parameter λ on post-transfer test accuracy on the CIFAR-100 (10 classes) test set for CIFAR-10 $\rightarrow$ CIFAR-100 (10 classes) semi-supervised transfer learning **with Lautum regularization**. Results are given for different amounts of labeled samples from the CIFAR-100 (10 classes) dataset. The legend depicts the value of λ.



(a) 100 labeled, 4,900 unlabeled CIFAR-100 (10 classes) samples.



(b) 200 labeled, 4,800 unlabeled CIFAR-100 (10 classes) samples.



(c) 500 labeled, 4,500 unlabeled CIFAR-100 (10 classes) samples.

Figure 8: Influence of the hyper-parameter λ on post-transfer test accuracy on the CIFAR-100 (10 classes) test set for CIFAR-10 $\rightarrow$ CIFAR-100 (10 classes) semi-supervised transfer learning **with Lautum regularization and Temporal Ensembling**. Results are given for different amounts of labeled samples from the CIFAR-100 (10 classes) dataset. The legend depicts the value of λ.

References

[1] A. Achille and S. Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19, 2018. 3

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 647–655, Bejing, China, 22–24 Jun 2014. PMLR. 2

[3] N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate information bottleneck. In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001*, pages 152–161, 2001. 3

[4] M. Gabrié, A. Manoel, C. Luneau, j. barbier, N. Macris, F. Krzakala, and L. Zdeborová. Entropy and mutual information in models of deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1821–1831. Curran Associates, Inc., 2018. 3

[5] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. 1

[6] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 529–536. MIT Press, 2005. 3

[7] R. Gupta, S. Sahu, C. Espy Wilson, and S. Narayanan. Semi-supervised and transfer learning approaches for low resource sentiment classification. In *Proceedings of ICASSP*, April 2018. 3

[8] P. Häusser, A. Mordvintsev, and D. Cremers. Learning by association - A versatile semi-supervised training method for neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 626–635, 2017. 3

[9] D. Jakubovitz, R. Giryes, and M. R. D. Rodrigues. Generalization error in deep learning. *arXiv*, abs/1808.01174, 2018. 3

[10] N. Jean, S. M. Xie, and S. Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5322–5333. Curran Associates, Inc., 2018. 3

[11] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017. 6

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7, 8

[13] A. Kumar, P. Sattigeri, and T. Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5534–5544. Curran Associates, Inc., 2017. 3

[14] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 2, 3, 7, 8, 10

[15] A. K. Lampinen and S. Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In *ICLR*, 2019. 2

[16] X. LI, Y. Grandvalet, and F. Davoine. Explicit inductive bias for transfer learning with convolutional networks. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2825–2834, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 2

[17] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 3239–3250, 2018. 3

[18] D. P. Palomar and S. Verdu. Lautum information. *IEEE Trans. Inform. Theory*, 54(3):964–975, March 2008. 2, 4, 6

[19] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. 2

[20] N. Papernot, M. Abadi, lfar Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*, 2017. 3

[21] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D., and T. D. D. Cox. On the information bottleneck theory of deep learning. In *ICLR*, 2018. 3

[22] O. Shamir, S. Sabato, and N. Tishby. Learning and generalization with the information bottleneck. *Theor. Comput. Sci.*, 411(29-30):2696–2711, 2010. 3

[23] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017. 3

[24] H. Sun, W. W. Cohen, and L. Bing. Semi-supervised learning with declaratively specified entropy constraints. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4430–4440. Curran Associates, Inc., 2018. 3

[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning. In *ICANN*, 2018. 2

[26] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. pages 368–377, 1999. 3

[27] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pages 1–5, 2015. 3

[28] X. Wang and J. Schneider. Flexible transfer learning under support and model shift. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1898–1906. Curran Associates, Inc., 2014. 2

[29] Y. Wei, Y. Zhang, J. Huang, and Q. Yang. Transfer learning via learning to transfer. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine*

Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 5085–5094, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 2

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014. 2

[31] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[32] H.-Y. Zhou, A. Oliver, J. Wu, and Y. Zheng. When semi-supervised learning meets transfer learning: Training strategies, models and datasets. *arXiv*, abs/1812.05313, 2018. 3, 7

[33] X. Zhu. Semi-supervised learning literature survey. In *University of Wisconsin-Madison, Tech. Rep. 1530*, 2006. 3