Generalization bounds for transfer learning using neural network features

Anonymous Authors[1]

Abstract

Understanding generalization for deep neural networks is a fundamental problem in modern machine learning theory. Generalization bounds are particularly important for deep transfer learning, where features extracted from a model trained on a data-rich source task (e.g. ImageNet) are used in a downstream (often data-poor) target task. Strikingly, these features can be obtained using unsupervised learning with similarity-based contrastive losses. We prove generalization bounds for deep transfer learning, in the case where the target dataset is independent of the source dataset and the classifiers are linear.

1. Introduction

Deep Neural Network (DNN)s have recently gained prominence as more effective than traditional machine learning methods. A pressing theoretical question is to understand *generalization* for DNNs, in the sense of ensuring that model accuracy does not degrade on unseen inputs. While these results are available for traditional machine learning models, the corresponding results are much more limited for deep models (Neyshabur et al., 2017; Kawaguchi et al., 2017).

In this article we identify a setting where we can apply statistical learning theory to provide generalization bounds for models which use *features* coming from a DNN. These bounds apply in the setting of deep transfer learning.

We can regard a DNN classifier as composed of two parts: a deep feature extraction layer, followed by a final classification "model". We thus break the generalization question into two parts:

1. Do the features learned by DNNs generalize?
2. Do models trained on (fixed) DNN features generalize?

By generalization, here we mean in-distribution generalization in the precise sense of statistical learning theory, which provides sample-dependent bounds on the generalization gap (Mohri et al., 2012).

1.1. Contributions

To our knowledge we are the first to present theoretical results proving learning bounds on models trained using deep features via transfer learning. Our analysis applies to transfer learning because in this setting, we can apply classification bounds using deep features representation *as if they are generic features*. In other words, we can treat the features as encoding prior knowledge about the task, despite the fact that they are learned using a deep model.

Our main result is as follows:

> Models with bounded Rademacher complexity trained using (fixed) DNN features on downstream tasks generalize.

The models include regularized linear classifiers, kernel classifiers, and shallow neural networks. We bound the expected loss (true risk) of the models in terms of the empirical loss and the number of training samples, independent of the dimension of the data (Theorem 4). Loosely stated, this means these types of transfer models do not overfit. Empirically, this result corresponds to a bound on the gap between the test and training loss (or error).

Remarkably, this result does not require proving that the features themselves generalize: since the bound is on the generalization gap, it applies equally when training accuracy is high or low.

2. Related work

2.1. Generalization: whether, how, and when

While it is now established that DNNs are exceptionally good at fitting data, this was not always the case. Very early neural networks with no hidden layers were effective learners, but were incapable of fitting the simple XOR function (Goodfellow et al., 2016). Later theoretical work (Barron, 1993; Leshno et al., 1993) showed that networks (with a hidden layer) were *expressive* enough to fit any function. These results ensured that models *could* fit data, but did not go so far as to ensure that they *would*. The next step was to show *how* algorithms for DNNs could fit data ensuring high training accuracy.

The remaining question is that of generalization, in the sense

of bounding the gap between expected loss and the empirical loss. A number of recent works address this question, but it remains unresolved. Neyshabur et al. (2015) proposed inductive bias of Stochastic Gradient Descent (SGD) optimization as an explanation for the generalization of neural networks: while there may be lots of bad minima (which overfit), the algorithm finds the good minima (which don't overfit). Hardt et al. (2016) extended the algorithmic stability approach, which provides generalization bounds for convex models (Xu & Mannor, 2012) to DNNs. However, the bounds, which depend on the number of steps of SGD, are loose in the non-convex case. Further work on loss landscapes (Choromanska et al., 2015; Keskar et al., 2017; Li et al., 2018) made empirical and theoretical progress in this area, but this research area has lost momentum (no pun intended).

Recent work on generalization in the over-parameterized regime (Belkin et al., 2018; Bartlett et al., 2017; Poggio et al., 2018; Belkin et al., 2018; 2020; Liang et al., 2020; 2019) revisits the question of *how*: making the point that neural networks (or kernels) can fit labels perfectly and still generalize. This makes the point that the conventional bias-variance trade-off need not apply in this context.

The hypothesis space complexity approach (Mohri et al., 2012) is a powerful approach to generalization bounds for classical machine learning models (reviewed in section 4). The empirical Rademacher complexity is a measure of the ability of a hypothesis class to fit random labels. The Rademacher complexity of a hypothesis leads to bounds on the generalization gap. Zhang et al. (2017) showed that DNNs have high (poor) Rademacher complexity. However bounds on the Rademacher complexity are useful for shallow models with bounded weights (Bartlett, 1998; Yin et al., 2019).

One line of research seeks surrogate complexity measures which do apply to DNNs. Keskar et al. (2017) propose robustness as a surrogate complexity measure, however Neyshabur et al. (2017) showed that this is not sufficient. Jiang et al. (2019) proposes geometric margin, and Liang et al. (2019) proposes information capacity and geometry as surrogates complexity measures. Jiang et al. (2020) propose empirical measures which are predictive of generalization. Arjovsky et al. (2019) shows how to distinguish between statistical associations that correspond to causal connections, and those that are just spurious correlations. Long & Sedghi (2020) propose generalization bounds based on additional parameters.

Jacot et al. (2018); Li et al. (2019); Arora et al. (2019); Shankar et al. (2020) study Neural Tangent Kernels, relating two-layer neural networks to a kernel, in the infinite width limit. Similar themes are further explored in (Ghorbani et al., 2019; Allen-Zhu et al., 2019; Domingos, 2020). This theoretically appealing mathematical modelling approach makes reasonable approximations which are more amenable to analysis. However, it stops short of providing generalization bounds for finite width DNNs.

2.2. Transfer learning

Prior to deep feature learning, determining relevant features was studied in statistical experimental design (Steinberg & Hunter, 1984; Chaloner & Verdinelli, 1995). Early machine learning models required choosing individual features and testing their relevance to classification (Blum & Langley, 1997; Kohavi et al., 1997). As data sizes and data dimensions became large, feature selection became onerous. Kernel methods (Hofmann et al., 2008) allow learning models to select from an infinite number of features. Rather than learning features, the features are determined by the selection of the kernel (Rasmussen & Williams, 2006).

The effectiveness of features learned by CNNs led to the "Pre-train and fine-tune" paradigm which emerged in computer vision about a decade ago (Razavian et al., 2014; Yosinski et al., 2014; Donahue et al., 2014; Oquab et al., 2014; Jia et al., 2014; Girshick et al., 2014). Kolesnikov et al. (2020) shows how efficient this approach can be compared to training.

Features are often extracted from a deep model trained on ImageNet (Deng et al., 2009) and used on a variety of downstream tasks, such as object detection, image classification and segmentation, and various medical imaging tasks (Neyshabur et al., 2020). The downstream data sets can be quite different: for example, CHEXPERT (Irvin et al., 2019), a medical imaging dataset of chest x-rays considering different diseases and DOMAINNET (Peng et al., 2019) datasets that are specifically designed to probe transfer learning in diverse domains which range from real images to sketches, clipart and painting samples.

Shallow transfer learning (Pan & Yang, 2010; Weiss et al., 2016; Zhuang et al., 2020), in contrast, is restricted to similar tasks. In this setting, domain-adaptation algorithms (Mansour et al., 2008) often involve model, rather than feature, transfer. Perrot & Habrard (2015) study transfer learning of linear models. See also generalization results for multi-task learning (Liu et al., 2017) and for transfer learning under model shift (Wang & Schneider, 2015).

Transfer learning is also effective in natural language processing (NLP) (Raffel et al., 2019), (Devlin et al., 2019). Pre-training is often performed using unsupervised learning on unlabeled data such as word embeddings (Mikolov et al., 2013; Pennington et al., 2014). These vectors are trained using a similarity-based loss, leading to semantically similar words mapping to geometrically close vectors (Mikolov et al., 2013). Hence, our transfer learning generalization

bound is useful across multiple fields of machine learning.

Linear classification is the benchmark for measuring the effectiveness of feature maps (He et al., 2020; Hénaff et al., 2019; Chen et al., 2020) for classification on the same dataset. Kornblith et al. (2019) propose a protocol for transfer learning, where features are compared using linear models, fine-tuned models, and randomly initialized models.

Early work by Razavian et al. (2014) presented the surprising result that transfer learning through linear classifiers using pre-trained features outperformed highly tuned models on a wide range of visual classification tasks on various datasets. However, it is recently found that fine tuning significantly outperforms transferred linear models, which nevertheless outperform shallow models (Chen et al., 2020; Hénaff et al., 2019).

There is limited theoretical work on deep transfer learning. Galanti et al. (2016) presents a theoretical framework using the concept of distribution of distributions. Lampinen & Ganguli (2019) studies deep linear network generalization dynamics. Liu et al. (2019) study how the loss landscape of deep representations impacts transfer learning. Wu et al. (2020) study theoretically how the alignment of data corresponding to different tasks impacts transfer learning. However none of these works provide bounds on the generalization gap in the setting of transfer learning.

3. Notation and setup

3.1. Notation

Let $x \in \mathcal{X}$ be an input, where $\mathcal{X} \subset [0,1]^d$. We consider the target set $\mathcal{Y} = \mathbb{R}$ for regression, and, later, $\mathcal{Y} = \{1, \ldots, k_c\}$ for classification. The dataset $S_m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ consists of m samples (x_i, y_i), drawn i.i.d. from $\rho(x, y)$. We also write $\rho(x)$ for the marginal. Let $\mathcal{H} = \{f(\cdot, \theta)\}$ be a class of hypothesis functions $f : \mathcal{X} \to \mathcal{Y}$, parameterized by θ.

Consider the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$. We assume that $\ell \in [0, 1]$, a mild assumption designed to avoid additional constants in the bounds. The expected loss $\mathcal{L}(f)$ and the empirical loss $\widehat{\mathcal{L}}_{S_m}(f)$ of the hypothesis $f \in \mathcal{H}$ are:

$$\mathcal{L}(f) = \mathbb{E}_{(x,y)\sim\rho}\ell(f(x, \theta), y)$$
$$\widehat{\mathcal{L}}_{S_m}(f) = \frac{1}{m}\sum_{i=1}^{m} \ell(f(x_i, \theta), y_i)$$

3.2. Multi-class classification

We consider the case, $\mathcal{Y} = \{1, \ldots, k_c\}$, of k_c classes. The scoring method for multi-class classification is to use k_c scoring functions,

$$f(x, \theta) = (f_1(x, \theta_1), \ldots, f_{k_c}(x, \theta_{k_c})). \tag{1}$$

The classification is given by the highest scoring function,

$$y_f(x, \theta) = \arg\max_{j=1,\ldots,k_c} f_j(x, \theta_j) \tag{2}$$

We write $y_f(x, \theta)$ for the classifier, to emphasize that it comes from the scoring function f. The relevant classification loss is the error, given by the zero-one loss: $\ell_{0,1}(y, y') = 0$ if $y = y'$ and 1 otherwise. Since this loss is not amenable to optimization, a surrogate loss, defined on the score functions, $\ell : \mathbb{R}^{k_c} \times \mathcal{Y} \to R^+$, is used instead. In order for bounds on the surrogate loss to lead to bounds on the classification loss, it should satisfy

$$\ell_{0,1}(y_f(x, \theta), y) \le \ell(f(x, \theta), y). \tag{3}$$

In order to apply learning bounds, the surrogate loss ℓ should be L_ℓ-Lipschitz as a function of its first input:

$$|\ell(y_1, y) - \ell(y_2, y)| \le L_\ell \|y_1 - y_2\| \quad \forall y \in \mathcal{Y}$$

For example, the γ-hinge and γ-margin loss satisfy (3) and have Lipschitz constant $L_\ell = 1/\gamma$, (Mohri et al., 2012). We show in section 8 that this result can also be applied to the KL-softmax loss.

3.3. Transfer learning

Let $S^{\text{source}}_{m'}$ be a source dataset generated from m' i.i.d. samples of ρ^{source}. Assume the model $f^{\text{source}}(x, \theta^*)$ has been trained on the data $S^{\text{source}}_{m'}$. Write $f^{\text{source}} = g \circ \phi^{\text{source}}$ as the composition of a feature map, $\phi^{\text{source}}(x, \theta^*) : \mathcal{X} \to \mathbb{R}^N$, where N is the number of features, and a (possibly linear) classifier g. The feature map, ϕ^{source}, encodes the prior knowledge learned from the source data that may be useful for the target task. In the sequel, we drop θ^* and write $\phi^{\text{source}}(x)$, since the weights will not change in transfer learning. The target task shares the input space, $\mathcal{X}$, and has target labels $\mathcal{Y}^{\text{target}}$, which may be different.

Given the feature map ϕ^{source}, define the bounded linear hypothesis class using these features as:

$$\mathcal{H}^{\text{transfer}}_\Lambda = \{f(x, \theta) = \theta \cdot \phi^{\text{source}}(x) \mid \|\theta\|_2 \le \Lambda\} \tag{4}$$

We write $\mathcal{G}$ to denote a generic hypothesis class defined on the target features, $\mathcal{G} = \{g(\cdot, w) : \mathbb{R}^N \to \mathcal{Y}^{\text{target}}\}$, later it will be a hypothesis class with bounded Rademacher complexity. Given $\mathcal{G}$, define a hypothesis class consisting of functions $f : \mathcal{X} \to \mathcal{Y}^{\text{target}}$ by composition with the feature map,

$$\mathcal{H}^{\text{transfer}} = \{g \circ \phi^{\text{source}} : \mathcal{X} \to \mathcal{Y}^{\text{target}} \mid g \in \mathcal{G}\} \tag{5}$$

4. Background statistical learning theory

Statistical learning theory (Mohri et al., 2012; Wainwright, 2019) can be used to obtain dimension-independent sample

complexity bounds for the expected loss (generalization) of a learning algorithm. These are bounds which depend on the number of samples m, but not on the dimension of the underlying data. The theory builds on the concentration of measure results (valid for sampling), and extends these results to learning (valid for learning algorithms).

The bounds we present here are fundamental. Stronger bounds can be obtained, see (Bartlett & Mendelson, 2002; Wainwright, 2019) and the references therein.

4.1. Concentration of measure

McDiarmid's inequality applies to (deterministic) functions which satisfy a bounded difference inequality. If the loss function $\ell \in [0,1]$, then

$$\mathbb{P}\left(\mathcal{L}(f(x),y) - \widehat{\mathcal{L}}_S(f(x),y) \geq \epsilon\right) \leq \exp\left(-2m\epsilon^2\right) \quad (6)$$

for any $\epsilon > 0$ (with a similar result for the other inequality). Setting $\delta = 2\exp\left(-m\epsilon^2/2\right)$ and solving for ϵ allows us to restate the result as

$$\mathcal{L}(f(x),y) \leq \widehat{\mathcal{L}}_S(f(x),y) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

with probability $\geq 1-\delta$, for any $\delta > 0$.

McDiarmid's inequality is a generalization of Hoeffdings inequality (Wainwright, 2019) which provides bounds on the difference between the empirical and true value of a sampled value (rather than function). Consider the simple experiment of flipping a possibly biased coin, whose true probability of heads is p. Let $\widehat{p}_m$ be the average number of heads after m tosses.

$$\mathbb{P}\left(p - \widehat{p}_m \geq \epsilon\right) \leq \exp\left(-2m\epsilon^2\right)$$

for any $\epsilon > 0$ (with a similar result for the other inequality). Thus, applying a similar argument as before, we have the inequality

$$p \leq \widehat{p}_m + \sqrt{\frac{\log(1/\delta)}{2m}} \quad (7)$$

with probability $\geq 1-\delta$, for any $\delta > 0$. Comparing (7) with (8) and Theorem 1 makes the point mentioned above that learning bounds extend sampling bounds with a complexity term.

4.2. Hypothesis space complexity

Classical generalization bounds estimate the difference between the empirical loss and the expected loss of a learning algorithm. A learning algorithm is a function which maps datasets, S_m, to hypotheses, $f_{\mathcal{A}(S)}$. For bounded deterministic functions,

$$\mathcal{L}(f(x),y) \leq \widehat{\mathcal{L}}_S(f(x),y) + \sqrt{\frac{\log(1/\delta)}{2m}} \quad (8)$$

with probability $\geq 1-\delta$, for any $\delta > 0$. For details refer to section 4.

Empirical risk minimization (ERM) corresponds to the algorithm,

$$f_{\mathcal{A}(S)} = \arg\min_{f\in\mathcal{H}} \widehat{\mathcal{L}}_S(f)$$

Now the learned function $f_{\mathcal{A}(S)}$ depends on the samples, S_m, so it is a *random* function, which means the inequality (8) does not apply. The *hypothesis space complexity* approach allows us to replace the generalization gap for the *learned* hypothesis with the worst-case generalization gap for function in the hypothesis space.

$$\mathcal{L}(f_{\mathcal{A}(S)}) - \widehat{\mathcal{L}}_S(f_{\mathcal{A}(S)}) \leq \sup_{f\in\mathcal{H}} \left(\mathcal{L}(f) - \widehat{\mathcal{L}}_S(f)\right) \quad (9)$$

The term on the right is bounded using the the Rademacher complexity, $\mathfrak{R}_m(\mathcal{G})$, of the losses composed with the hypotheses, $\mathcal{G} = \{\ell(f(x),y) \mid f \in \mathcal{H}\}$. which we discuss in the next section.

4.3. Rademacher complexity

The Rademacher complexity $\mathfrak{R}_m(\mathcal{G})$ measures the probability that some function $f \in \mathcal{H}$ is able to fit samples of size m with random labels (Koltchinskii & Panchenko, 2000; Bartlett & Mendelson, 2001; 2002).

The empirical Rademacher complexity, $\widehat{\mathfrak{R}}_{S_m}(\mathcal{H})$ of the hypothesis class $\mathcal{H}$ on the data S_m, is defined by (Mohri et al., 2012; Bartlett & Mendelson, 2002)

$$\widehat{\mathfrak{R}}_{S_m}(\mathcal{H}) = \mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i)\right]$$

where the expectation is over the Rademacher variables σ_i which are i.i.d. on $\{-1,+1\}$. It can be interpreted, (e.g. for h taking values in $[-1,1]$), as the ability of the hypothesis class to fit random labels on the dataset S_m.

The Rademacher complexity $\mathfrak{R}_m(\mathcal{H})$ of the hypothesis class $\mathcal{H}$ on the distribution $\rho(x)$, is given by taking expectations over the i.i.d. samples S_m of size m of the empirical Rademacher complexity,

$$\mathfrak{R}_m(\mathcal{H}) = \mathop{\mathbb{E}}_{S\sim\rho^m}\left[\widehat{\mathfrak{R}}_S(\mathcal{H})\right]$$

For example, for the bounded linear hypothesis space, $\mathcal{H}_\Lambda = \{x \mapsto w\cdot x \mid \|w\|_2 \leq \Lambda\}$, the Rademacher complexity is estimated by

$$\mathfrak{R}_m(\mathcal{H}_\Lambda) \leq \frac{r\Lambda}{\sqrt{m}}. \quad (10)$$

where $r = \max_{i=1}^m \|x_i\|$. The Rademacher complexity of DNNs is poor (Zhang et al., 2017). Below, in subsection 7.1 and subsection 7.2, we present well-established estimates of the Rademacher complexity of kernel and shallow neural network hypothesis spaces.

4.4. Generalization bound for regression

Going back to (9), when the loss is L_ℓ-Lipschitz, Talagrand's lemma gives

$$\mathfrak{R}_m(\mathcal{G}) \leq L_\ell \mathfrak{R}_m(\mathcal{H}).$$

Combining these arguments leads to the following theorem.

Theorem 1. *Let $f \in \mathcal{H}$ and ℓ be L_ℓ-Lipschitz as a function of f. Then for any $\delta > 0$,*

$$\mathcal{L}(f(x), y) \leq \widehat{\mathcal{L}}_S(f(x), y) + 2L_\ell \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

with probability $\geq 1 - \delta$.

Note that (8) applies to deterministic functions, and the inequality in Theorem 1 involves the Rademacher complexity.

4.5. Generalization bound for classification

In this section, we review results on generalization bounds for classification problems using Rademacher complexity. The standard reference for the binary classification case is (Boucheron et al., 2005). For the extension to the multi-class case, we refer to the textbook (Mohri et al., 2012). While this section is a synthesis of textbook results, it will be fundamental for the results which follow.

The bound (3) above requires that the zero-one loss be bounded by the surrogote loss. One way to enforce (3) is using the margin. Define the margin of the scoring function vector (1), by

$$t(f(x), y) = f_y(x) - \max_{j \neq y} f_j(x). \tag{11}$$

The hinge loss penalizes scoring functions which have a margin less than $\gamma > 0$. Let $H_\gamma(t) = \max(1 - t/\gamma, 0)$, then the γ-margin hinge loss is given by

$$\ell_\gamma(f(x), y) = H_\gamma(t(f(x), y))$$

The loss is $\frac{1}{\gamma}$-Lipschitz, and satisfies (3).

Theorem 2. *Consider the k_c-class learning problem, using a L-Lipschitz loss, which satisfies (3). Suppose f has the form (2), where each $f_i \in \mathcal{H}$. For any $\delta > 0$,*

$$\mathcal{L}_{0,1}(y_f(x), y) \leq \widehat{\mathcal{L}}_S(f(x), y) + 4k_c L_\ell \, \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

with probability $\geq 1 - \delta$.

5. Rademacher complexity for feature transfer to linear classifier

Our first, and main result, is simply the observation that using deep neural network features ϕ^{source} as feature maps on downstream tasks does not affect the Rademacher complexity of the models. These models include linear classifiers, as well as kernels and regularized shallow neural networks, denoted generically by $\mathcal{G}$.

Let $S_m = \{(x_1, y_1), \ldots (x_m, y_m)\}$ be drawn i.i.d. from the target distribution $\rho^{\text{target}}(x, y)$.

Theorem 3. *Suppose ρ^{target} is different from ρ^{source}. The Rademacher complexity of $\mathcal{H}^{\text{transfer}}$, defined by (5), is*

$$\mathfrak{R}_m(\mathcal{H}^{\text{transfer}}) = \mathfrak{R}_m(\mathcal{G})$$

In particular, for the bounded linear feature hypothesis space, $\mathcal{H}_\Lambda^{\text{transfer}}$, defined by (4),

$$\mathfrak{R}_m(\mathcal{H}_\Lambda^{\text{target}}) \leq \frac{r\Lambda}{\sqrt{m}} \tag{12}$$

where $r = \max_{i=1}^m \|\phi^{\text{source}}(x_i)\|$.

Proof. The first result is on the Rademacher complexity of $\mathcal{H}^{\text{transfer}}$. Recall that $\phi^{\text{source}}(x) = \phi^{\text{source}}(x, w^*)$ is a random function which depends on the samples in $S_{m'}^{\text{source}}$.

Recall that the empirical Rademacher complexity is:

$$\widehat{\mathfrak{R}}_{S_m}(\mathcal{H}^{\text{transfer}}) = \mathbb{E}\left[\sup_{h \in \mathcal{H}^{\text{transfer}}} \frac{1}{m} \sum_{i=1}^m \sigma_i h(x_i)\right]$$

where the expectation is over the Rademacher variables σ_i i.i.d. on $\{-1, +1\}$. Here $h(x_i) = g(\phi^{\text{source}}(x_i))$. Since the function values $g(z_i)$ where $z_i = \phi^{\text{source}}(x_i)$, and $h(x_i)$ are the same, we have

$$\mathfrak{R}_{S_m}(\mathcal{H}^{\text{transfer}}) = \widehat{\mathfrak{R}}_{\phi \circ S_m}(\mathcal{G})$$

where $\phi \circ S_m = \{\phi(x_1), \ldots \phi(x_m)\}$.

The Rademacher complexity is given by taking expectations over samples of size m of the data of the empirical Rademacher complexity,

$$\widehat{\mathfrak{R}}_m(\mathcal{H}^{\text{transfer}}) = \mathop{\mathbb{E}}_{S \sim (\rho^{\text{target}})^m}\left[\widehat{\mathfrak{R}}_S(\mathcal{H}^{\text{transfer}})\right]$$

Again, from the point of view of ρ^{target}, the mapping ϕ^{source} is deterministic. So the result follows by noting that the mapping is just a deterministic change of variables, as written above for the empirical Rademacher complexity.

The result (12) follows from the first part, along with the standard result on complexity of linear hypothesis with bounded weights (6), which can be found, for example, in Theorem 5.10 of (Mohri et al., 2012). □

6. Generalization bounds for deep transfer learning using linear classifiers

We now present Theorem 4 and Theorem 5, bounds on the generalization gap for regression, and classification with k_c classes, respectively.

These theorems are stated in two parts. For the special case of linear classifiers, we present a precise bound. We then allow for other hypothesis classes $\mathcal{G}$, such as kernels and regularized shallow neural networks. In those cases, we can apply the Rademacher complexity bounds presented in subsection 7.1 and subsection 7.2, respectively.

6.1. Generalization bound for regression

Theorem 4. *Let the loss ℓ be L_ℓ-Lipschitz. Given the dataset S_m, generated i.i.d. from ρ^{target}. Assume ρ^{target} is different from ρ^{source}. Set $r = \sup_{x \in S_m} \|\phi^{\text{source}}(x)\|$. Suppose $f \in \mathcal{H}^{\text{transfer}}$ uses the hypothesis class $\mathcal{G}$ defined by (5). Then for any $\delta > 0$,*

$$\mathcal{L}(f) \leq \widehat{\mathcal{L}}_{S_m}(f) + 2L_\ell \mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

with probability $\geq 1 - \delta$. In particular, for $f \in \mathcal{H}^{\text{transfer}}_\Lambda$, the bound is

$$\mathcal{L}(f) \leq \widehat{\mathcal{L}}_{S_m}(f) + 2L_\ell \frac{r\Lambda}{\sqrt{m}} + \sqrt{\frac{\log(1/\delta)}{2m}} \qquad (13)$$

Proof. The proof of Theorem 4 follows from Theorem 1 and Theorem 3. □

The main inequality in Theorem 4 (13) is a textbook generalization bound, applied in our setting. The novelty is that it can be applied with deep neural network features.

The quantity of interest, on the left hand side of the inequality, is the unknown expected loss of the model f which uses the deep features ϕ^{source}. The first quantity on the right is the known empirical loss of the model. The last term on the right is the same term that appears in statistical sampling: it represents the error between an empirical mean and the expectation, even when no learning has occurred (see section 4). Because the bound is probabilistic, the parameter δ represent the trade-off between a tighter bound and a higher probability of the bound holding.

The Lipschitz constant L_ℓ is $1/\gamma$ for margin loss with margin γ. Note this does not involve the Lipschitz constant of the model f at all.

The middle term on the right is important: it is a bound on the complexity of the model. When applied to the linear hypothesis space, we obtain three terms: r, Λ, and m. For typical feature vectors r is order 1. In the case of (normalized) cosine similarity based features, we have $r = 1$. The constant, Λ, represents the norm of the weight of the linear classifier, which can be controlled using $\|\theta\|^2$ regularization. Finally m is the number of samples in the target dataset. The $\sqrt{m}$ factor also appears in sampling. In subsection 7.1 and subsection 7.2, we give another interpretation of the Rademacher complexity bounds for $\mathcal{H}^{\text{transfer}}_\Lambda$, taking $\mathcal{G}$ as a kernel and shallow network, respectively.

In the case of deep neural networks, the Rademacher complexity is so high that the bound is vacuous. However, for a shallow model, with bounded weights, it can be controlled.

6.2. Generalization bound for classification

Next we present the bound in the multi-class classification case. The theorem bounds the true classification error in terms of the empirical surrogate loss. It applies to the hinge or margin loss. We show in section 8 that this result can also be applied to the KL-softmax loss, multiplied by the constant $1/\log 2$.

Theorem 5. *Let the loss ℓ be L_ℓ-Lipschitz, and satisfy (3). Given the dataset S_m, generated i.i.d. from ρ^{target}. Assume ρ^{target} is different from ρ^{source}. Set $r = \sup_{x \in S_m} \|\phi^{\text{source}}(x)\|$.*

Suppose each $f_i \in \mathcal{H}^{\text{transfer}}$, for $i = 1, \ldots, k_c$, where k_c is the number of classes. Let y_f be the classifier (2). Then for any $\delta > 0$,

$$\mathcal{L}_{0,1}(y_f) \leq \widehat{\mathcal{L}}_{S_m}(f) + 4k_c L_\ell \mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

with probability $\geq 1 - \delta$. In particular, when each $f_i \in \mathcal{H}^{\text{transfer}}_\Lambda$,

$$\mathcal{L}_{0,1}(y_f) \leq \widehat{\mathcal{L}}_{S_m}(f) + 4k_c L_\ell \frac{r\Lambda}{\sqrt{m}} + \sqrt{\frac{\log(1/\delta)}{2m}} \qquad (14)$$

Proof. The proof of Theorem 5 follows from Theorem 2 and Theorem 3. □

The main difference in this theorem, compared to Theorem 4, is that the zero-one loss is on the left-hand side of the inequality, while the surrogate loss is on the right. The other difference is in the second term on the right hand side: where previously the coefficient was $2L_\ell$, and we had a single linear function, now we are in the multi-class case with k_c classes, and f is a vector of k_c linear functions (all with the same weight norm bound Λ). This is the reason that the factor k_c appears in the coefficient. For the remainder of the terms, the discussion in the previous case applies.

7. Rademacher complexity for other models

In this section, we present the Rademacher complexity bounds for kernels and regularized shallow neural networks.

7.1. Rademacher complexity for Kernels

Consider an abstract Reproducing Kernel Hilbert space (RKHS), $\mathbb{H}$, with an inner product written as $\langle \phi, \phi' \rangle$, for $\phi, \phi' \in \mathbb{H}$. Refer to (Wainwright, 2019), Chapter 12. A feature map, ϕ is a function $\phi : \mathcal{X} \to \mathbb{H}$.

Define the kernel

$$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

which, by definition, is positive semi definite (PSD).

Kernel classification is to map data to $\mathbb{H}$, which in this context, is referred to as the feature space. If a rich enough feature space can be found, then linear classification is possible. Thus the kernel hypothesis space, with bounded weight norms, is given by

$$\mathcal{H}_{\Lambda}^{\text{Ker}} = \{ f(x, \theta) = \langle \phi(x), \theta \rangle \mid \|\theta\|_{\mathbb{H}} \leq \Lambda \}.$$

The representation theorem allows us to write $\theta = \sum_{i=1}^{m} \alpha_i \phi(x_i)$, for some coefficients α_i, so that

$$f(x, \theta) = \sum_{i=1}^{m} \alpha_i \langle \phi(x), \phi(x_i) \rangle = \sum_{i=1}^{m} \alpha_i K(x_i, x)$$

allowing kernel computations to replace high (or infinite) dimensional inner products.

Let $r = \sup_{x \in X} K(x, x)$, then

$$\mathfrak{R}_m(\mathcal{H}_{\text{Ker}}) \leq \frac{r\Lambda}{\sqrt{m}} \tag{15}$$

In applications of the result above, the supremum in the definition of r can also be replaced with the maximum over the dataset.

7.2. Rademacher complexity of regularized shallow neural networks

For regularized shallow neural networks, Rademacher complexity bounds are available (Bartlett, 1998; Bartlett & Mendelson, 2001; Mohri et al., 2012). This result allows such networks to be used as hypothesis classes, $\mathcal{G}$, in the transfer hypothesis class (5).

As a precise example, for a two layer network of the form

$$\mathcal{H}_{\Lambda_1, \Lambda_2}^{DNN} = \left\{ x \mapsto \sum_{j=1}^{n_2} \theta_j \sigma(u_j \cdot x) \right\},$$
$$\text{for } \|\theta\|_1 \leq \Lambda_1, \|u_j\|_2 \leq \Lambda_2, j \in [n_2]$$

we have (Bartlett, 1998; Bartlett & Mendelson, 2001; Mohri et al., 2012)

$$\mathfrak{R}_m(\mathcal{H}_{\Lambda_1, \Lambda_2}^{DNN}) \leq \frac{\Lambda_1 \Lambda_2 r}{\sqrt{m}} \tag{16}$$

where, as before r is a bound on $\|x\|_2$. However, extending these bounds to deeper networks requires that the corresponding products of the norms be under control, which puts a sevee restriction on the norms of the matrices in each layer. More sophisticated generalization bounds for neural networks are available, for example (Bartlett et al., 2017).

The above complexity measures allow us to apply the main theorems for these hypothesis classes.

8. Classification bounds using the cross-entropy loss

In this section, we show we can use the cross-entropy loss as a surrogate loss in the classification result Theorem 5.

The cross-entropy loss (or the composition of the Kullback-Liebler divergence with the softmax function) is a popular loss for DNNs.

$$\ell_{KL-SM}(f, y) = -\log(\text{softmax}(f_y))$$

The following lemma shows that we can apply Theorem 5 to linear classifiers with the KL-softmax loss, just by multiplying the loss by a constant.

Lemma 6. *Let $\ell_2(f, y) = \frac{1}{\log 2} \ell_{KL-SM}(f, y)$ be a constant multiple of the standard KL-softmax loss. Then $\ell_2(f, y)$ satisfies (3), and is $2/\log 2$-Lipschitz.*

Proof. When the classification is correct, (3) holds, since the loss is non-negative. When the classification is incorrect, the margin, t, defined by (11), is non-positive. In this case, we will show that $\ell_{KL-SM}(f, y) \geq \log 2$.

Rewrite the loss as

$$\ell_{KL-SM}(f, y) = \log \left(\frac{\sum_i e^{f_i}}{e^{f_y}} \right) = \log \left(1 + \sum_{i \neq y} e^{(f_i - f_y)} \right)$$

Write $f_m = \max_i f_i$, so the margin is given by $t = f_y - f_m$.

$$\ell_{KL-SM}(f, y) \geq \log \left(1 + \exp(f_m - f_y) \right) \geq \log(2)$$

since the margin is negative, giving the desired result.

For the Lipschitz constant, compute the gradient:

$$\begin{aligned} \nabla_f \ell_{KL-SM}(f, y) &= \nabla_f \left(f_y + \log \left(\sum_i \exp(f_i) \right) \right) \\ &= e_y + \frac{\exp f_i}{\sum_j \exp(f_j)} \\ &= e_y + \text{softmax}(f) \end{aligned}$$

The two vectors on the last line each have norm bounded by one, so $\|\nabla \ell_{KL-SM}(f, y)\| \leq 2$. Since the Lipschitz constant of a function is bounded by the supremum of the gradient norm, we obtain the desired result. □

9. Discussion

9.1. Can fine-tuning fit random labels?

Zhang et al. (2017) makes the point that neural networks can fit random data, which means that the Rademacher complexity of neural networks as a hypothesis class is high. However, we establish bounds on the generalization gap using neural networks as features, and low complexity models as classifiers (independent of the labels). This means, for example, that using ImageNet features as ϕ^{source} and a linear classifier, we ensure that the generalization gap (as approximated by the difference in train loss and test loss) is small on, say, CIFAR10. This is true in both the cases of using real and random labels.

On the other hand, the example from (Zhang et al., 2017) leaves open the possibility that fine tuning on downstream data (training the full network without freezing any layers) could also fit random labels. In this case, it would be worthwhile to study this further to better understand the extent of the trade off between loss and overfitting, on fine tuning versus linear classifiers.

As can be seen in Figure 1, a ResNet18 network (He et al., 2016) pre-trained on ImageNet and (orange) transferred to CIFAR10 is prone to smaller generalization gap than (blue) fine-tuned on CIFAR10, in both cases of training on (a) true labels and (b) random labels. In our work, we offer the bound on the generalization gap for the transfer network, not on the fine-tuned one.

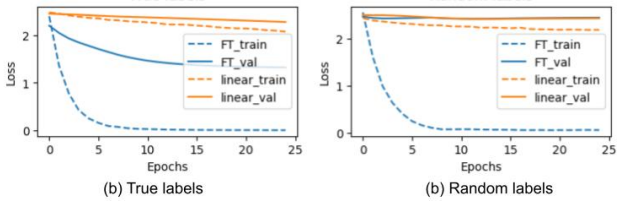

Figure 1. Generalization gap of fine-tuning (blue) is greater than that of transfer learning (orange), in both cases of (a) True labels and (b) Random labels.

9.2. Extensions to other related work

The analysis of the overparameterized regime (Belkin et al., 2018) can be understood in the context of a feature map. In this case, perfect fit for features corresponds to better features: there is no notion of overfitting of features, since the downstream classification layer is regularized to control the norm of the weight vectors. Neural Tangent kernel analysis (Jacot et al., 2018) could be applied to a model using a DNN feature map.

The performance and generalization gap between the methods could be partly closed with either more expressive models which still have bounded Rademacher complexity. For example, using efficient kernels (Rudi et al., 2017; Sterge et al., 2020), where the theory still applies, or shallow neural networks with regularized weights (Bartlett, 1998; Bartlett & Mendelson, 2002).

10. Conclusion

In this work, we presented generalization bounds for models trained on (fixed) deep neural network features via transfer learning. These models include typically used linear classifiers, kernels, and shallow neural networks. We first built on concepts from statistical learning theory to identify generalization bounds (difference between expected loss and empirical loss) in the transfer learning setting using Rademacher complexity, in the cases of regression (Theorem 1) and classification (Theorem 2).

We then identified the Rademacher complexities of linear classifiers (Theorem 3 (12)), Kernels (Theorem 4 (15)), and regularized shallow neural networks (Theorem 5 (16)) using empirical surrogate loss. These Rademacher complexity measures allow us to compute the generalization bounds, such as shown in Equations (13), (14). We also showed that Theorem 5 can be used for shallow neural networks with cross-entropy loss using Theorem 6.

Hence, we proved that classifiers with bounded Rademacher complexity used on transferred features from deep neural networks do not overfit, i.e. we can bound the generalization gap in transfer learning.

References

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 6155–6166, 2019. URL http://papers.nips.cc/paper/8847-learning-and-generalization-in-overparameterized-neural-networks-going-beyond-two-layers.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019. URL http://arxiv.org/abs/1907.02893.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. In Wallach, H. M.,

Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8139–8148, 2019. URL http://papers.nips.cc/paper/9025-on-exact-computation-with-an-infinitely-wide-neural-net.

Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory*, 39(3):930–945, 1993. doi: 10.1109/18.256500. URL https://doi.org/10.1109/18.256500.

Bartlett, P. L. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. In Helmbold, D. P. and Williamson, R. C. (eds.), *Computational Learning Theory, 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 16-19, 2001, Proceedings*, volume 2111 of *Lecture Notes in Computer Science*, pp. 224–240. Springer, 2001. doi: 10.1007/3-540-44581-1_15. URL https://doi.org/10.1007/3-540-44581-1_15.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6240–6249, 2017. URL http://papers.nips.cc/paper/7204-spectrally-normalized-margin-bounds-for-neural-networks.

Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 540–548. PMLR, 2018. URL http://proceedings.mlr.press/v80/belkin18a.html.

Belkin, M., Hsu, D., and Xu, J. Two models of double descent for weak features. *SIAM J. Math. Data Sci.*, 2 (4):1167–1180, 2020. doi: 10.1137/20M1336072. URL https://doi.org/10.1137/20M1336072.

Blum, A. L. and Langley, P. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

Boucheron, S., Bousquet, O., and Lugosi, G. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

Chaloner, K. and Verdinelli, I. Bayesian experimental design: A review. *Statistical Science*, pp. 273–304, 1995.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pp. 192–204, 2015.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Domingos, P. Every model learned by gradient descent is approximately a kernel machine. *arXiv preprint arXiv:2012.00152*, 2020.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recog-

nition. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pp. 647–655. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/donahue14.html.

Galanti, T., Wolf, L., and Hazan, T. A theoretical framework for deep transfer learning. *Information and Inference: A Journal of the IMA*, 5(2):159–209, 04 2016. ISSN 2049-8764. doi: 10.1093/imaiai/iaw008. URL https://doi.org/10.1093/imaiai/iaw008.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. Linearized two-layers neural networks in high dimension. *CoRR*, abs/1904.12191, 2019. URL http://arxiv.org/abs/1904.12191.

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 580–587. IEEE Computer Society, 2014. doi: 10.1109/CVPR.2014.81. URL https://doi.org/10.1109/CVPR.2014.81.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In Balcan, M. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1225–1234. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/hardt16.html.

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

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9726–9735. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00975. URL https://doi.org/10.1109/CVPR42600.2020.00975.

Hénaff, O. J., Srinivas, A., Fauw, J. D., Razavi, A., Doersch, C., Eslami, S. M. A., and van den Oord, A. Data-efficient image recognition with contrastive predictive coding. *CoRR*, abs/1905.09272, 2019. URL http://arxiv.org/abs/1905.09272.

Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *The annals of statistics*, pp. 1171–1220, 2008.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Jacot, A., Hongler, C., and Gabriel, F. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018. URL http://papers.nips.cc/paper/8076-neural-tangent-kernel-convergence-and-generalization-in-neural-networks.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R. B., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Hua, K. A., Rui, Y., Steinmetz, R., Hanjalic, A., Natsev, A., and Zhu, W. (eds.), *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, pp. 675–678. ACM, 2014. doi: 10.1145/2647868.2654889. URL https://doi.org/10.1145/2647868.2654889.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. Predicting the generalization gap in deep networks with margin distributions. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HJlQfnCqKX.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learn-

ing: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Kohavi, R., John, G. H., et al. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pp. 491–507. Springer, 2020. doi: 10.1007/978-3-030-58558-7_29. URL https://doi.org/10.1007/978-3-030-58558-7_29.

Koltchinskii, V. and Panchenko, D. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pp. 443–457. Springer, 2000.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 2661–2671. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00277. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Kornblith_Do_Better_ImageNet_Models_Transfer_Better_CVPR_2019_paper.html.

Lampinen, A. K. and Ganguli, S. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryfMLoCqtQ.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. doi: 10.1016/S0893-6080(05)80131-5. URL https://doi.org/10.1016/S0893-6080(05)80131-5.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6391–6401, 2018. URL http://papers.nips.cc/paper/7875-visualizing-the-loss-landscape-of-neural-nets.

Li, Z., Wang, R., Yu, D., Du, S. S., Hu, W., Salakhutdinov, R., and Arora, S. Enhanced convolutional neural tangent kernels. *CoRR*, abs/1911.00809, 2019. URL http://arxiv.org/abs/1911.00809.

Liang, T., Poggio, T. A., Rakhlin, A., and Stokes, J. Fisher-rao metric, geometry, and complexity of neural networks. In Chaudhuri, K. and Sugiyama, M. (eds.), *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pp. 888–896. PMLR, 2019. URL http://proceedings.mlr.press/v89/liang19a.html.

Liang, T., Rakhlin, A., et al. Just interpolate: Kernel "ridgeless" regression can generalize. *Annals of Statistics*, 48 (3):1329–1347, 2020.

Liu, H., Long, M., Wang, J., and Jordan, M. I. Towards understanding the transferability of deep representations. *CoRR*, abs/1909.12031, 2019. URL http://arxiv.org/abs/1909.12031.

Liu, T., Tao, D., Song, M., and Maybank, S. J. Algorithm-dependent generalization bounds for multi-task learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(2):227–241, 2017. doi: 10.1109/TPAMI.2016.2544314. URL https://doi.org/10.1109/TPAMI.2016.2544314.

Long, P. M. and Sedghi, H. Generalization bounds for deep convolutional neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1e_FpNFDr.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21:1041–1048, 2008.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 3111–3119, 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012. ISBN 978-0-262-01825-8. URL http://mitpress.mit.edu/books/foundations-machine-learning-0.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6614.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30:5947–5956, 2017.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0607f4c705595b911a4f3e7a127b44e0-Abstract.html.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 1717–1724. IEEE Computer Society, 2014. doi: 10.1109/CVPR.2014.222. URL https://doi.org/10.1109/CVPR.2014.222.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22 (10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 1406–1415. IEEE, 2019. doi: 10.1109/ICCV.2019.00149. URL https://doi.org/10.1109/ICCV.2019.00149.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In Moschitti, A., Pang, B., and Daelemans, W. (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543. ACL, 2014. doi: 10.3115/v1/d14-1162. URL https://doi.org/10.3115/v1/d14-1162.

Perrot, M. and Habrard, A. A theoretical analysis of metric hypothesis transfer learning. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1708–1717. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/perrot15.html.

Poggio, T. A., Kawaguchi, K., Liao, Q., Miranda, B., Rosasco, L., Boix, X., Hidary, J., and Mhaskar, H. Theory of deep learning III: explaining the non-overfitting puzzle. *CoRR*, abs/1801.00173, 2018. URL http://arxiv.org/abs/1801.00173.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X. URL https://www.worldcat.org/oclc/61285753.

Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 512–519. IEEE Computer Society, 2014. doi: 10.1109/CVPRW.2014.131. URL https://doi.org/10.1109/CVPRW.2014.131.

Rudi, A., Carratino, L., and Rosasco, L. FALKON: an optimal large scale kernel method. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3888–3898, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/05546b0e38ab9175cd905eebcc6ebb76-Abstract.html.

Shankar, V., Fang, A., Guo, W., Fridovich-Keil, S., Ragan-Kelley, J., Schmidt, L., and Recht, B. Neural kernels without tangents. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings*

of Machine Learning Research, pp. 8614–8623. PMLR, 2020. URL http://proceedings.mlr.press/v119/shankar20a.html.

Steinberg, D. M. and Hunter, W. G. Experimental design: review and comment. *Technometrics*, 26(2):71–97, 1984.

Sterge, N., Sriperumbudur, B., Rosasco, L., and Rudi, A. Gain with no pain: Efficiency of kernel-pca by nyström sampling. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3642–3652. PMLR, 2020. URL http://proceedings.mlr.press/v108/sterge20a.html.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wang, X. and Schneider, J. G. Generalization bounds for transfer learning under model shift. In Meila, M. and Heskes, T. (eds.), *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, pp. 922–931. AUAI Press, 2015. URL http://auai.org/uai2015/proceedings/papers/123.pdf.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

Wu, S., Zhang, H. R., and Ré, C. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylzhkBtDB.

Xu, H. and Mannor, S. Robustness and generalization. *Machine Learning*, 86(3):391–423, 2012. doi: 10.1007/s10994-011-5268-1. URL https://doi.org/10.1007/s10994-011-5268-1.

Yin, D., Ramchandran, K., and Bartlett, P. L. Rademacher complexity for adversarially robust generalization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7085–7094. PMLR, 2019. URL http://proceedings.mlr.press/v97/yin19b.html.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3320–3328, 2014. URL http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.