An Information Theoretical Transferability Metric

Fred Guth
fredguth@fredguth.com
Departamento de Ciência de Computação, Universidade de Brasília
Brasília, DF, Brazil

ABSTRACT

Deep Learning in Computer Vision deserves a lot of its success to the simple Transfer Learning technique of fine-tunning, using a pre-trained model as feature extractor for a new task, which unlocks the floodgates of very big datasets like ImageNet[12] even to the most modest classification task. It is so ubiquitous that not doing it is considered foolhardy [15]. Despite of that, little is known on when and to what extend fine-tunning works. Decisions on which pre-trained model to use are *ad hoc* and automatic model selection seems far from reality. Fortunately, a recent theoretical effort [1, 2, 4–9, 14, 17–20, 25] have shed some light upon these matters using Information Theory. An interesting result is that Fisher Information Matrices can be used as task embeddings and distances between them can be seen as the properties of "closeness" and "transferability". In this work, we present an (hopefully) accessible review of this theoretical *tour de force*.

CCS CONCEPTS

• **Computing methodologies** → **Transfer learning**.

KEYWORDS

transfer learning, automation, deep learning, complexity, task2vec, information theory, learning theory

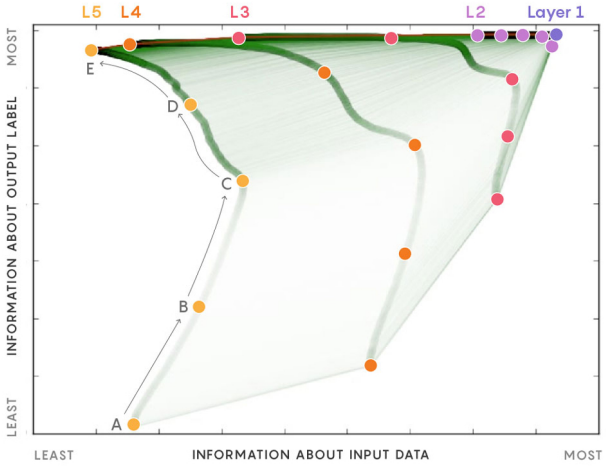
1 INTRODUCTION

Despite all success of supervised Deep Learning in a myriad of applications, humans still show a much better generalization ability from very few samples (Goodfellow et al. [11]). The huge demand for well labeled data, which are expensive and difficult to obtain, lead to a growing interest for Transfer Learning, which harness previously acquired knowledge to new tasks so that they can be learned in a more effective and efficient way. It is notoriously known that one of the main reasons for deep learning success is due to the fact that transfer learning works very well for it.

Indeed, Transfer learning became so ubiquitous in the Deep Learning context that Mahajan et al. [15] claims that training from *tabula rasa*, instead of using a pre-trained ImageNet[12] model as feature extractor, is considered foolhardy in Computer Vision.

A recent survey on Research Frontiers in Transfer Learning [13] points that this simple transfer learning technique, also known as fine-tunning, is the most used approach in recent publications. This confirms Torrey and Shavlik [21] view that transfer has been applied on an *ad hoc* basis. Moreover, this context imply a new constraint on deep learning usage in practice: experts capable of selecting models and datasets to build feature extractors for new tasks.

This same survey presents as open problems the lack of:



A **INITIAL STATE:** Neurons in Layer 1 encode everything about the input data, including all information about its label. Neurons in the highest layers are in a nearly random state bearing little to no relationship to the data or its label.

B **FITTING PHASE:** As deep learning begins, neurons in higher layers gain information about the input and get better at fitting labels to it.

C **PHASE CHANGE:** The layers suddenly shift gears and start to "forget" information about the input.

D **COMPRESSION PHASE:** Higher layers compress their representation of the input data, keeping what is most relevant to the output label. They get better at predicting the label.

E **FINAL STATE:** The last layer achieves an optimal balance of accuracy and compression, retaining only what is needed to predict the label.

Source: Reproduced from Wolchover [23].

Figure 1: Illustration of Tishby's conjecture in the Information Plane.

(a) theory to support why, when and to what extent fine-tunning works;
(b) transferability metrics, which could lead to meta-learning algorithms that automatic select the most promising pre-trained model for a new task;

This two open problems are the subject of this paper.

The best paper at CVPR 2018 (Zamir et al. [24]) proposes a experimental evaluation of transferability, creating a taxonomic map for task transfer learning. This leaves open the possibility of creating

meta-learning algorithms to automatically select models for new tasks based on previous empirical results.

The method is highly computational intensive, though. The total number of transfer functions trained to build a taxonomy of 26 vision tasks was around 3000 and took 47,886 GPU hours[24, §4 Experiments].

It is desired for such metric to be more easily computed. One recent approach that promises to fill this gap is Achille et al. [2, Task2Vec], where the authors use the Fisher Information Matrix (FIM), a method widely used in physics to forecast empirical results of hypothesis, to generate a task embedding, which conveys the task complexity and can be used to generate transferability metrics. Unfortunately, the AWS* sponsored research did not make available its implementation and experimental data as open-source.

1.1 Overview

In sections 2 and 3, we give some background and motivation for sections 4 and 5, which present the theory behind task embeddings and are the crux of the paper. Section 6 presents the application of the theory that motivated our interest in the first place, how to use information theoretical supported transferability measures. Section 7 summarize the findings and suggest future venues of inquiry.

1.2 Related Work

A superficial and joyful overview of the Information Bottleneck Theory can be found in Wolchover [23]. For a more in depth and broad overview of IB and its applications we refer to Hafez-Kolahi and Kasaei [14]. Cover and Thomas [10] is an superb source on the foundations of Information Theory and covers most (if not all) concepts used here in a very pleasant way. For more on information/task complexity we suggest [5, 8].

2 PRELIMINARIES AND NOTATIONS

Let $X \in \mathcal{X}$ be a random variable of the input space (e.g., an image) and $Y \in \mathcal{Y}$ a random variable that we want to infer (e.g, a label), which is therefore referred as our *task*. We consider that X and Y are continuous, but represented in a finite precision machine, therefore quantized into discrete values. A dataset is a finite set of m samples $\mathcal{D} = \{(x_i, y_i)\}^m$.

We will make frequent use of the following basic information theoretic concepts[10]: $H(X) = \mathbb{E}_p[\log \frac{1}{p(x)}]$ † is the Shannon entropy of the random variable X, not to be confused with $\mathcal{H}$, the hypothesis space in classical learning theory. The conditional entropy is $H(X|Y) = \mathbb{E}_{\hat{y}} H(X|Y = \hat{y}) = H(X, Y) - H(Y)$. We denote $p(X, Y)$ as the joint probability of X and Y, and the corresponding mutual information (MI or information gain) $I(X; Y) = I(Y; X)$ is defined as:

$$I(X;Y) = \mathrm{KL}(\, p(Y,X) \,\|\, p(Y)p(X)\,) = \mathbb{E}_X \,\mathrm{KL}(\, p(Y|X) \,\|\, p(Y)\,), \quad (1)$$

where $\mathrm{KL}(\, p \,\|\, q\,) = \mathbb{E}_p[log\frac{p}{q}]$ is the Kullbach-Leibler divergence (KL-divergence) between the the probability distributions p and q, which normally denote the true distribution of a variable and the estimated distribution respectively. Cross-entropy is:

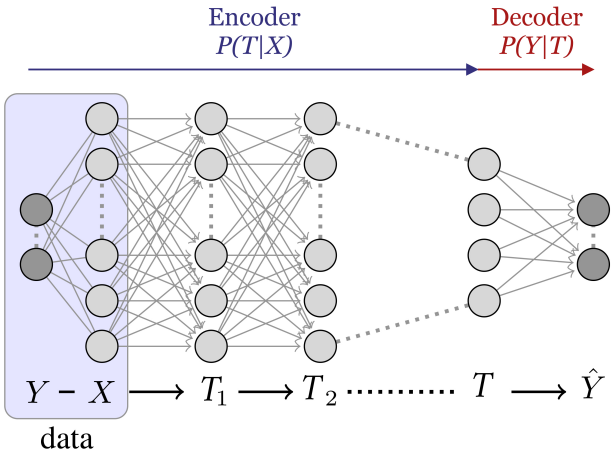
$$H_{p,q}(X) = \mathbb{E}_p[-\log q(X)] = H_p(X) + \mathrm{KL}(\, p(X) \,\|\, q(X)\,). \quad (2)$$

Total correlation TC(T), a.k.a. the multi-variate mutual information is:

$$TC(T) = \mathrm{KL}(\, p(T) \,\|\, \textstyle\prod_i p(T_i)\,)$$

$$TC = 0 \iff T_1 \perp T_2 \perp \ldots \perp T_k,$$

where $p(T_i)$ are the marginal distributions of the components of T. A DNN model (figure 2) is a succession of K layers, each k^{th} layer denoted as $T_k = \phi_k(W_k T_{k-1})$, where W_k represent the weight vector and ϕ_k a non-linear function (usually ReLU) in the k^{th} layer.



Source: Adapted from Shwartz-Ziv and Tishby [17].

Figure 2: Deep Neural Network layers as representations of the input.

Given the weights, the layers form a Markov chain of successive internal representations of the input variable $X : Y \to X \to T_1 \to T_i \to \ldots \to T_K \to \hat{Y}$, i.e. $p(T|X, Y) = p(T|X)$ and their MI obey a chain of Data Processing Inequalities (DPI)[10, 17], therefore $I(Y; X) \geq I(Y; T)$. The DNN can be seen as an encoder from $X \to T$, $p(T|X)$, and a decoder from $T \to \hat{Y}$, $p(Y|T)$.

3 LEARNING THEORY HAS FAILED DEEP

Learning Theory studies the complexity of learning algorithms, not only in terms of steps but also in the number of samples needed to guarantee true error bounds (generalization bounds). It started with the seminal work of Valiant [22] introducing the PAC Learning model.

The generalization bounds obtained by PAC Learning is [16, Theorem 2.13]:

$$\varepsilon^2 < \frac{\log |\mathcal{H}_\epsilon| + \log \delta^{-1}}{2m} \quad (3)$$

where:

- ε is the tolerance margin between training and generalization errors. The generalization error measures the accuracy of the algorithm for previously unseen data, the test error;

* Amazon Web Services † Despite the notation, it is not a function of X, but a function its distribution p(X)

$|\mathcal{H}_\epsilon|$ is the cardinality of the ϵ-cover of the hypothesis space. Typically, it is assumed that $|\mathcal{H}_\epsilon| \sim \frac{1}{\epsilon}^d$;

δ is the confidence margin;

m is the number of training samples, a.k.a. the sample complexity;

d is the Vapnik–Chervonenkis dimension of the hypothesis space. In the case of neural networks it is $O(|\theta|log|\theta|)$, where $|\theta|$ is the number of parameters in the network.

The generalization error is bounded by a function of the hypothesis space and the dataset sizes.

3.1 Criticism

The main criticism on the current state of Learning Theory in the context of DNNs are:

(1) the bounds are too loose and, therefore, not very valuable in practice;
(2) it depends on the model (size of hypothesis space), not only on the problem;
(3) its preference for simpler algorithms (smaller hypothesis space) does not explain the fact that larger and deeper Deep Neural Networks (DNNs) usually achieve better accuracy and generalization.

4 AN INFORMATION THEORY OF DEEP LEARNING

Tishby and Zaslavsky [20] propose a new Learning Theory of Deep Learning based on Shannon's Information Theory. Therefore, it is important to establish some context.

Tasks can be easy or difficult depending on how information is represented, a classical example is that it is much easier to calculate using hindu-arabic numerals than with roman numerals. Thus, it is reasonable to think of supervised trained DNNs as performing representation learning, where the last layer, the head, is typically a softmax regression classifier and all previous layers just learn to provide a good representation for this last classifier[11, Chapter 15].

Another way is to think of the last layer as decoding a message that was encoded by the rest of the network (see figure 2). In this view, each layer can be seen as a single random variable, T_i, and the network as the communication channel itself, $p(T|X)$, to which all Shannon information properties apply.

4.1 Desiderata for representations

Let T denote a representation of X, that is optimal to the task Y, meaning that T captures and exposes only the information from X which is relevant to Y. Ideally, this representation should be [6]:

a statistic: a function $T \sim p(T|X)$;

sufficient: $I(Y;T) = I(Y;X)$, so there is no loss in relevant Y information;

minimal: $I(T;X)$ is minimized, so that it retains as little of X as possible. This means there is an encoding from X to T that keep only relevant information;

invariant: to the effect of nuisances N, where $N \perp Y \rightarrow I(N;Y) = 0 \rightarrow I(T;N) = 0$ means that if N does not have information about Y, there should not be information of N in the representation T, otherwise the classifier could fit to spurious correlations;

maximally disentangled: no information will be present in the correlations between components of T.

4.2 The Information Bottleneck

The Information Bottleneck is a method for finding minimal sufficient statistics developed by Tishby et al. [19]:

$$\begin{aligned} T^* = \arg\min_T \quad & I(T;X) \\ \text{s.t.} \quad & I(T;Y) = I(X;Y) \end{aligned} \tag{4}$$

Applying the lagrangian relaxation, we have:

$$\begin{aligned} T^* &= \arg\min_T \mathcal{L} \\ \mathcal{L} &= \min_{q(T|X)} I(T;X) - \beta\, I(T;Y), \beta > 0 \end{aligned} \tag{IB}$$

where β is the Lagrange multiplier. Tishby and Zaslavsky [20] used the Information Bottleneck (IB) to formulate the deep learning goal as an information trade-off between sufficiency and minimality, accuracy and generalization, prediction and compression.

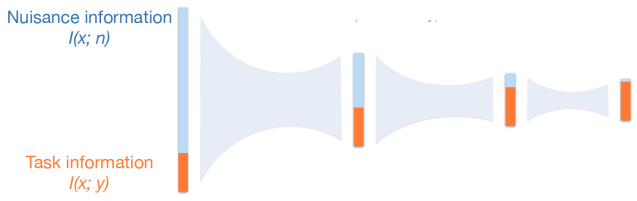
4.3 Emerging Properties of DNNs

It is interesting to notice that it is possible to rewrite (Appendix A.1) the IB formulation as:

$$\mathcal{L} = \min_q \underbrace{H_{p,q}}_{\text{cross-entropy}}(Y|T) + \beta \underbrace{I(T;X)}_{\text{regularizer}}, \beta > 0 \quad \text{(IB Lagrangian)}$$

and the cross-entropy, the most successful loss function for classification tasks, naturally emerges, as does a not usual regularizer in a second term.

4.3.1 Invariance and minimality. Achille and Soatto [6] demonstrate that by using SGD there is in fact an implicit compression of information, showing the regularizer is there, but implicit. Also, they show that by enforcing the minimization of the information about the input representation $I(T;X)$, invariance and disentanglement naturally emerges as well, satisfying the desiderata(§4.1).



Source: Adapted from Achille [1].

Figure 3: Stacking layers improve generalization.

The intuition for the emergence of invariance is quite simple to understand. The architecture enforces a dimensionality reduction by stacking layers and/or applying pooling, while the algorithm is trying to maintain the information about the labels (figure 3). Besides, techniques like dropout and batch-normalization add noise to the training. Noise in the communication channel reduces its capacity, therefore, reduces the upper bound to $I(X;T)$. If $I(X;T)$ is being reduced while $I(Y;T)$ is maintained, what is being reduced is $I(T;N)$, where $N \perp Y$.

Another way to reduce the information is to train with the regularizer of eq. (IB Lagrangian) explicitly. This is what Achille and Soatto [7] propose.

4.4 Tishby's Conjecture: Learning is forgetting

Shwartz-Ziv and Tishby [17] conjectures that DNNs work in two phases: (1) a **Fitting Phase** where the DNN rapidly fit to the training labels; and (2) a **Compression Phase** where it spends most of the time, compressing the inputs and therefore "forgetting" as much of X that it can, without losing information about Y.

To support this view, they present the Information Plane (Figure 1), where darker colors represent later epochs in the training (and the darkening of the colors is in constant rate to the number of epochs). Its experimental setup, however, has been challenged and no consensus has been reached [8]. Regardless, Tishby's conjecture is the basis of a new Learning Theory that is worth attention.

4.5 Information Theoretical Bounds

As the current Learning Theory does not explain well documented observed facts §3.1, Tishby and Zaslavsky [20] propose a change of focus from worse case model-dependent distribution-independent PAC bounds to typical data-dependent model-free bounds. In other words, instead of bounding using the expressivity of the hypothesis space ($\mathcal{H}$), bounding by the limits of the compression of the input data.

In this new information learning theory, the complexity of a learning problem only depends on the problem itself, the data and the information it conveys, and not on any property of the algorithm chosen to solve it, which is a very interesting proposition.

In the IB formulation, $p(T|X)$ is a stochastic mapping between each value $x \in \mathcal{X}$ to a value $t \in \mathcal{T}$, inducing a partition of $\mathcal{X}$, a quantization of $\mathcal{X}$ into a codebook $\mathcal{T}$. In terms of Shannon's Information Theory, it can be seen as a communication from X to T.

A reliable communication is one that require that different input sequences produce disjoint output sequences. As $|\mathcal{X}|$ is quite large, we can use the concept of typicality. The typical set, $\mathcal{T}_\epsilon$, is the set of distributions that are near the true distribution. Empirical probability distributions that are non typical have exponentially smaller probability. The current learning theory works for any distribution of data, but $|\mathcal{T}_\epsilon| \ll |\mathcal{X}|$.

A typical n-sequence of T symbols can, from the ϵ-partition property (AEP, [10, Theorem 3.1.2]), represent $\sim 2^n H(X|T)$ possible X n-sequences. And again from AEP, there are $\sim 2^n H(X)$ different n-sequences of X. Therefore, as we have to ensure that no two sequences of X maps to the same sequence of T, the total number of disjoint subsets T_i of X is upper bounded by $2^{n(H(X)-H(X|T)} = 2^{nI(X;T)}$ [18]. From equation 3, we now can bound the generalization error. Let T_ϵ be the space of ϵ-partitions of X:

$$
\begin{aligned}
|\mathcal{H}| \leq 2^{|\mathcal{X}|} &\rightarrow 2^{|\mathcal{T}_\epsilon|} \\
|\mathcal{T}_\epsilon| &= 2^{I(T;X)} \\
\epsilon^2 &< \frac{2^{I(T;X)} + log\delta^{-1}}{2m}
\end{aligned} \tag{5}
$$

Every k bits of compression have the same effect to the error as 2^k samples.

4.6 IB Achille's heel

The formulation presented in eq. (IB Lagrangian) is a representation of yet not observed future data (the activations of future X). It is sort of a wishful thinking, as we only have past data.

A common criticism to IB is that a totally valid minimization of $I(T;X)$ is to memorize the index of input training data that map to a certain label. Although keeping little information, this obviously will not generalize. We need a function obtained in training that is guaranteed to work in test time.

This is basically what Achille and Soatto [6] propose with the Information in the Weights Bottleneck.

5 INFORMATION IN THE WEIGHTS

5.1 Deep Learning Reality

Deep Learning is usually associated with DNNs, but it is only one of its components:

(1) DNN architecture
(2) Optimizer (SGD)
(3) Dataset
(4) Loss function, usually:

$$
\begin{aligned}
\mathcal{L}(W) &= H_{p,q}(Y|X,W) \\
&= H_{p,q}(D|W)
\end{aligned}
$$

where, as a reminder, p is the true distribution and q is the model's approximation of the true distribution.

Not only the architecture is important to current Deep Learning success. As it was already mentioned and will be seen shortly, in this information theoretical view the stochastic of SGD plays a crucial role, so does the choice of cross-entropy for the loss function and the use of large datasets.

A known problem, though, is that DNNs are prone to overfitting. Actually, Zhang et al. [26] shows that state-of-the-art convolutional deep neural networks can easily fit a random labeling of training data.

5.2 Overfitting

To understand why DNNs overfit, Achille and Soatto [6] propose to decompose the cross-entropy. Assuming the dataset is sampled from some generative model $p(D|\theta)$:

$$
\boldsymbol{H_{p,q}(D|W)} = H_p(D,\theta) + I(\theta;D|W) + \mathbb{E}\,\mathrm{KL}(\,p\,\|\,q\,) - \underbrace{I(D;W|\theta)}_{overfitting}
$$

A naive idea to eliminate overfitting would be to rewrite the loss to eliminate the overfitting term:

$$\mathcal{L}(W) = H_{p,q}(D|W) + I(D;W|\theta)$$

To calculate $I(D;W|\theta)$, true distribution p_θ is needed, which we are just trying to approximate with q in training. Hence we are presented with chicken-egg problem. Rather, one can add a Lagrangian multiplier to upper bound $I(D;W|\theta)$:

$$\boldsymbol{\mathcal{L}(W) = H_{p,q}(D|W) + \underbrace{\beta I(D;W)}_{\text{Information in the Weights}}} \qquad \text{(new IBL)}$$

Remarkably, this has the same form of the IB Lagrangian equation.

5.3 A new Information Lagrangian

The question now is if this new Lagrangian emerged from trying to eliminate overfitting is somehow related to the IB Lagrangian in the activations presented before.

$$\underset{\text{input}}{X} \longrightarrow \underset{\text{activations}}{T} \longrightarrow \underset{\text{label}}{Y}$$

$$\min_{q(T|X)} \mathcal{L}(W) = H_{p,q}(Y|T) + I(T;X) \qquad \text{(Activations IB)}$$

$$\underset{\text{dataset}}{D} \longrightarrow \underset{\text{weights}}{W} \longrightarrow \underset{\text{real distribution}}{p_{(Y|X)}}$$

$$\min_{q(T|X)} \mathcal{L}(W) = H_{p,q}(D|W) + I(D;W) \qquad \text{(Weights IB)}$$

Intuitively, the *Weights IB* seems to be the dual to the *Activations IB*, as $I(T;X)$ which measures the complexity of the activations representation, can be defined by the amount of weight in the network: low or zero weights will connect to the activations that are not in T^* which minimizes $I(T;X)$.

Achille and Soatto [6, Corollary C.8] have proved that indeed $I(T;X) \leq I(W;D)$. In other words, the amount of information needed to be memorized to minimally represent the dataset is an upper bound to the amount of information needed to guarantee invariance (a small error in test data).

As $I(W;D)$ can be calculated, this development allows one to explicitly regularize the training. That is exactly what [7, Information Dropout] proposes.

5.4 Shannon vs. Fisher Information

By using eq. (1), eq. (new IBL) can be rewritten as:

$$\boldsymbol{\mathcal{L}(W) = H_{p,q}(D|W) + \beta\,\mathrm{KL}\Big(\underbrace{q(W|D)}_{\text{training output}} \,\|\, \underbrace{p(W)}_{\text{fixed prior}}\Big)}$$

In other words, $I(W;D)$ is the divergence of the conditional model distribution $q(W|D)$ and the expected prior averaging all datasets. If, instead, we assume an isotropic gaussian[‡] as the prior, the information in the weights when W_* is minimal, is given by:

$$\mathrm{KL}(\, q(W|D) \,\|\, p(W)\,) = \frac{1}{2}\left(\log|\mathcal{F}m| + \cancel{\log \lambda^2 I} + \frac{\cancel{W_*^2}}{\cancel{\lambda^2} I} \right),$$

where the canceled terms are the ones that do not depend on $q(W|D)$ and can be ignored, $\log|\mathcal{F}|$ is the log-determinant of Fisher Information Matrix of the weights and m is the number of samples in the dataset.

This is quite interesting as it gives us an analytical and fast calculation of a bound to $I(W;D)$:

$$I(T;X) \leq I(D;W) \leq \log|\mathcal{F}(W^*)| \qquad (6)$$

5.5 Information, Flat Minima and Generalization

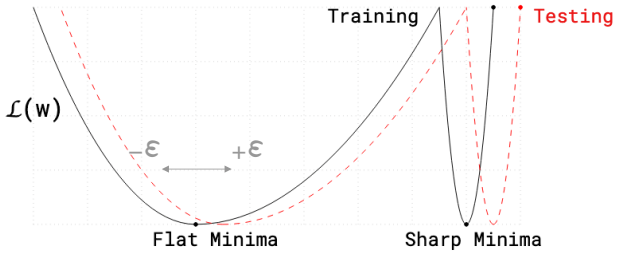


Figure 4: Information in the weights explain the preference of SGD for flat minima.

The relation of low Fisher Information in the weights and good generalization in the activations bring us another interesting insight. It is a well documented phenomena that DNNs trained with SGD tend to find flat minima, meaning saddle points where some noise in the test data will minimally affect the loss (see figure 4). Here we have a theoretical reason why this is happening. SGD is implicitly regularizing the information in the weights and that is the same as looking for minimas with low Fisher Information, which are Flat Minimas.

6 TOPOLOGY OF LEARNING TASKS

To this day, transferability is either measured experimentally [24] or inferred subjectively by experts according to tasks "proximity". Given an analytical transferability measure, obtained directly from the data in a cost-effective way, with experimentally proved prediction ability, automatic selection of source tasks as feature extractors for target tasks (auto-DL) is a simple search in the topology of learning tasks.

This illustrates the importance of building such topology. In other words, we need to know:

- What is the complexity of a learning task?
- How far or close are two tasks?
- How difficult it is to transfer from one task to another?

The aim of this section is to show such a transferability metric based on the information theory developments explained in previous sections.

6.1 Complexity of Learning Tasks

Intuitively, the complexity of a learning task is related to its best expected true error. This is exactly what what the Information in the Weights (section 5) give us.

[‡] An isotropic gaussian is one where the covariance matrix is represented by $\Sigma = \lambda^2 I$.

From eq. (5) and eq. (6), we have:

$$\epsilon^2 \propto \frac{2^{I(X;T)}}{2m} \propto \frac{2^{\log|\mathcal{F}|)}}{2m} \propto \frac{|\mathcal{F}|}{m}$$

Given a fixed architecture, the amount of information in the weight (Fisher Information) measures how much "memorization" was used to fit the model. High information in the weights suggest more "difficult" tasks. The Fisher Matrix (FIM) measure the resilience of the loss due to perturbation in the weights (see figure 4). If a weight accept more noise, it is less important and there is no need to "memorize" it. Also, as seen in 5.5, this amount of noise has direct correspondence to generalization. Using this intuition, [3, Task2Vec] uses the diagonal of the FIM as an embedding that represent the task itself. Since the FIM can be too noisy when trained from few examples, the diagonal of the FIM is used as it is considered a more simple and robust representation.

Different choices of fixed architectures, however, produce FIMs that are not comparable. To address this, a standard "probe" network pre-trained on ImageNet is used. The FIM of the probe represents the canonical task t_0 from which other tasks are compared. The embedding of a new task t_i is obtained by re-training only the classifier layer $p(T;Y)$, which usually can be done efficiently, and then computing the FIM for the feature extractor parameters.

6.2 Measures in the Task Topology

Achille et al. [3, Task2Vec] propose two measures in the task topology: (1) Relatedness, a semantic distance which they call taxonomic distance; and (2) Transferability, which they call transfer distance.

6.2.1 Relatedness: semantic distance. Experts base their subjective *ad hoc* choices of which tasks to use as source for a specific target to the idea of "proximity" or "closeness" of tasks (how far or close are two tasks). Thus, a measure of semantic distance with clear operational meaning is highly desired, that metric is the symmetric distance of task embeddings:

Definition 6.1 (Symmetric distance). *The symmetric distance between normalized embeddings measures its "closeness":*

$$d_{sym}(F_a, F_b) = d_{cos}\Big(\frac{F_a}{F_a+F_b}, \frac{F_b}{F_a+F_b}\Big),$$

where d_{cos} is the cosine distance, F_a and F_b are the two task embeddings, and the division is element-wise.

6.2.2 Transferability. Transferability (or fine-tunning gain) from a task t_a to a task t_b is the difference in expected performance between a model trained for task b from a fixed initialization, t_0, and the performance of a solution to t_a fine-tuned for t_b:

$$D_{\text{f-t}}(t_a \to t_b) = \frac{\mathbb{E}[\ell_{a\to b}] - \mathbb{E}[\ell_b]}{\mathbb{E}[\ell_b]},$$

where expectations are taken over all trainings, ℓ_b is the final test error obtained by training task b from initialization, and $\ell_{a\to b}$ is the error when starting from a solution to task a fine-tuned for task b. Hence, transferability depends on the similarity between two tasks and the complexity of the first. Indeed, the fact that pre-training in ImageNet has become a *de facto* standard [15] is due to its high complexity.

This intuition suggest the following measure for transferability:

Definition 6.2 (Asymmetric distance). *The asymmetric distance[§] between two normalized embeddings measures its transfer gain of using one as the source of the other:*

$$d_{asym}(t_a \to t_b) = d_{sym}(t_a, t_b) - \alpha d_{sym}(t_a, t_0),$$

where t_0 is the canonical embedding, and α is an hyperparameter.

In their experiments, the best value of α ($\alpha = 0.15$ when using a ResNet-34 pre-trained on ImageNet as the probe network) was considered robust to the feature-extractor selection meta-task.

7 DISCUSSION

In this work we presented the information theoretical support for the use of Fisher Information Matrices as task embeddings with which it is possible to measure the closeness and the transferability between tasks.

This theory is full of counter-intuitive and elucidative explanations for several puzzling phenomena. It shows that:

a) Noise in training and architectural bottlenecks are helpful;
b) Stacking layers increase generalization;
c) There is no "curse of dimensionality", as the complexity of a task is related not to the number of parameters of a model, but only to the compression limits of the data itself;
d) Reducing information in the weights during training yields smaller error in the activations in testing;
e) Tishby's conjecture is right and learning is forgetting, what maybe his information plane cannot prove, but certainly Achille and Soatto proof of the bound of the information in the weight to the information in the activation does;
f) The preference of SGD for flat minima can be explained by an implicit regularizer that tries to constraint the amount of information in the weights;
g) A FIM can be used as a representation of a Task;
h) It is possible to calculate asymmetric "distances" between tasks as transferability metrics in a cost efficient way;
i) There is a potential path towards deep learning automation.

At the same time the this theoretical *tour de force* lifts the veil of several aspects of deep learning, it also opens new venues of inquiry like how to use the Fisher Information Matrix to not only decide task sources but also which layer has the right amount of semantic for the target task in question; how does the choice of the learning rate affect the noise in the channel and if it can explain the phenomenon of "super-convergence"; How good is the analytical measures of complexity and measurability proposed compared to the empirically obtained ones in Zamir et al. [24]. These are questions that will certainly be subject of our future investigations.

REFERENCES

[1] Alessandro Achille. 2019. CS103 - Topics in Representation Learning, Information Theory and Control. https://alexachi.github.io/cs103/index.html [Online; accessed on June 20th, 2019].
[2] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2Vec: Task Embedding for Meta-Learning. *arXiv preprint arXiv:1902.03545* (2019).
[3] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. 2019.

[§] Not properly a distance as asymmetric distance is an oxymoron and it also can be negative.

Task2Vec: Task Embedding for Meta-Learning. *CoRR* abs/1902.03545 (2019). arXiv:1902.03545 http://arxiv.org/abs/1902.03545
[4] Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. 2019. The Information Complexity of Learning Tasks, their Structure and their Distance. *arXiv preprint arXiv:1904.03292* (2019).
[5] Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. 2019. The Information Complexity of Learning Tasks, their Structure and their Distance. *CoRR* abs/1904.03292 (2019). arXiv:1904.03292 http://arxiv.org/abs/1904.03292
[6] Alessandro Achille and Stefano Soatto. 2017. On the Emergence of Invariance and Disentangling in Deep Representations. *CoRR* abs/1706.01350 (2017). arXiv:1706.01350 http://arxiv.org/abs/1706.01350
[7] Alessandro Achille and Stefano Soatto. 2018. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2018), 2897–2905.
[8] Alessandro Achille and Stefano Soatto. 2019. Where is the Information in a Deep Neural Network? arXiv:cs.LG/1905.12213
[9] William Bialek, Ilya Nemenman, and Naftali Tishby. 2001. Predictability, complexity, and learning. *Neural computation* 13, 11 (2001), 2409–2463.
[10] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA.
[11] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org/
[12] M. Guillaumin and V. Ferrari. 2012. Large-scale knowledge transfer for object localization in ImageNet. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. https://doi.org/10.1109/cvpr.2012.6248055
[13] Fred Guth. 2019. *Research Frontiers in Transfer Learning: a systematic review with a quantitative approach*. Technical Report. UnB.
[14] Hassan Hafez-Kolahi and Shohreh Kasaei. 2019. Information Bottleneck and its Applications in Deep Learning. *arXiv preprint arXiv:1904.03743* (2019).
[15] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 181–196.
[16] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. The MIT Press.
[17] Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the Black Box of Deep Neural Networks via Information. *CoRR* abs/1703.00810 (2017). arXiv:1703.00810 http://arxiv.org/abs/1703.00810
[18] Noam Slonim. 2002. *The information bottleneck: Theory and applications*. Ph.D. Dissertation. Hebrew University.
[19] Naftali Tishby, Fernando C. Pereira, and William Bialek. 1999. The Information Bottleneck Method. 368–377.
[20] Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*. IEEE, 1–5.
[21] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI Global, 242–264.
[22] L. G. Valiant. 1984. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing - 84*. ACM Press. https://doi.org/10.1145/800057.808710
[23] Natalie Wolchover. 2017. New Theory Cracks Open the Black Box of Deep Learning. https://www.quantamagazine.org/new-theory-cracks-open-the-black-box-of-deep-learning-20170921/
[24] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling Task Transfer Learning. *CoRR* abs/1804.08328 (2018). arXiv:1804.08328 http://arxiv.org/abs/1804.08328
[25] Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. 2018. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences* 115, 31 (2018), 7937–7942.
[26] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *CoRR* abs/1611.03530 (2016). arXiv:1611.03530 http://arxiv.org/abs/1611.03530

Appendices

A PROOFS

A.1 Information Bottleneck Lagrangian

$$\begin{aligned} \min_{p(T|X)} \quad & I(T;X) \\ \text{s.t.} \quad & I(T;Y) \le I(X;Y) \end{aligned}$$

Can be rewritten:

$$\begin{aligned} I(T;Y) &\le I(X;Y) \\ H(Y) - H(Y|T) &\le H(Y) - H(Y|X) \\ H(Y|X) - H(Y|T) &\le 0 \end{aligned}$$

$$\begin{aligned} \min_{p(T|X)} \quad & I(T;X) \\ \text{s.t.} \quad & H(Y|X) - H(Y|T) \le 0 \end{aligned}$$

Using the Lagrangian multiplier, it can be relaxed as:

$$\gamma > 0, \quad \min_{p(T|X)} \quad I(T;X) + \gamma H(Y|T)$$

$$\frac{1}{\gamma} = \beta > 0, \quad \min_{p(T|X)} \quad H(Y|T) + \beta I(T;X) \qquad \square$$

A.2 Cross-Entropy Decomposition

We want to prove that:

$$H_{p,q}(D|W) = H_p(D,\theta) + I(\theta;D|W) + \mathbb{E}\,\mathrm{KL}(\,p\,\|\,q\,) - I(D;W|\theta)$$

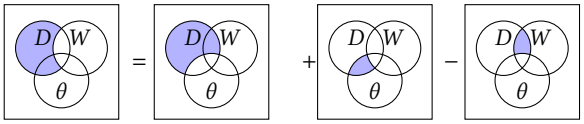
From eq. eq. (2):

$$H_{p,q}(D|W) = H_p(D|W) + \mathrm{KL}(\,p(D|W)\,\|\,q(D|W)\,)$$

Therefore, we only need to prove that:

$$H_p(D|W) = H_p(D|\theta) + I(D|W;\theta) - I(D;W|\theta)$$

Which is clear with the help of the following Venn diagrams:



$$H_p(D|W) = H_p(D|\theta) \qquad +I(D|W;\theta) - I(D;W|\theta) \qquad \square$$