A Survey of Transfer Learning for Convolutional Neural Networks

Ricardo Ribani* and Mauricio Marengoni†
Programa de Pós-Graduação em Engenharia Elétrica e Computação
Universidade Presbiteriana Mackenzie, São Paulo, SP 01302–000
Email: rribani@hotmail.com, mmarengoni@mackenzie.br†*

***Abstract*—Transfer learning is an emerging topic that may drive the success of machine learning in research and industry. The lack of data on specific tasks is one of the main reasons to use it, since collecting and labeling data can be very expensive and can take time, and recent concerns with privacy make difficult to use real data from users. The use of transfer learning helps to fast prototype new machine learning models using pre-trained models from a source task since training on millions of images can take time and requires expensive GPUs. In this survey, we review the concepts and definitions related to transfer learning and we list the different terms used in the literature. We bring the point of view from different authors of prior surveys, adding some more recent findings in order to give a clear vision of directions for future work in this field of research.**

***Keywords*-Transfer Learning; Convolutional Neural Networks; Deep Learning;**

I. Introduction

Machine learning models are commonly designed to resolve specific tasks and work in isolation. In these cases, the model should be constructed and trained from scratch and requires a lot of data. Depending on the kind of problem to be solved or the type of model, millions of samples may be required. Transfer learning is a method used to transfer knowledge acquired from one task to resolve another. This procedure can help for example to improve accuracy or reduce the time of training. A different task solved with a similar problem can be a starting point to extract learned information and fine-tune the model, reducing the isolation paradigm of having two different problems.

In general, complex problems require a lot of data but the situation gets complicated when it needs to be labeled for supervised learning. It's very hard to label a vast amount of data which may require time, tools and people to label it, and even after having it done there is no guarantee that all the labels are right. In order to ensure the quality of the labels in a dataset some more amount of work is needed, like double checking or choosing the best of three labels.

With the advances of social networks, there is a lot of content being shared by users across the internet, but there is also a big concern with privacy. It's just not possible to use the data from users without explicit permission. Recent news have reported that big datasets got taken down by the issues involving the General Data Protection Regulation (GDPR) [1], which makes the life of companies working with building artificial intelligence solutions way more complicated. The GDPR is an European law on data protection that aims to give control to individuals over their personal data, which forces companies that collect data from users to say when, how and the purpose of collection. Also, users can request to remove their data from the company's database at any time.

Recently, some very good datasets have been created for application in different tasks like classification, object recognition and semantic segmentation. The ImageNet [2] which is a large-scale hierarquical image database, organized according to the WordNet hierarchy and has more than 14 million images annotated in more than 21k synsets. For the ImageNet Large Scale Visual Recognition Challenge [3], a smaller set of 1.4 million images with 1000 categories was created to benchmark object category classification and detection. The MSCOCO dataset [4] which is focused on Object Recognition with the goal of advancing the state-of-the-art in the field, containing 2.5 million labeled instances of objects and 328k images in 91 categories. And recently released by Google AI, there is the Open Images Dataset [5] with 9.2 million images annotated for classification, object detection and visual relationship detection. These are just a few examples that can be used for transfer learning.

There are many other public datasets available on the internet for use in different tasks, but it's very common to have a problem to solve in the same domain that is not labeled. For example, in the ImageNet dataset there are two categories of guitar: "acoustic guitar" and "electric guitar". What if I want to train a model to recognize different types of guitar like stratocaster, telecaster or les paul? By having a smaller dataset labeled with this data, it's possible to just use the ImageNet to train a deep learning model and then fine-tune it using a smaller set of images with the desired labels. The most used machine learning frameworks already provide pre-trained models using different datasets, which make things way easier.

Most people that start to work with machine learning don't know what is transfer learning and hence don't know how much it can be beneficial. On the other hand, there are people

2474-0705/19/$31.00 ©2019 IEEE
DOI 10.1109/SIBGRAPI-T.2019.00010

that already have some expertise in machine learning and know what is transfer learning but don't know how or when to use it. If wrongly applied, transfer learning can penalize the accuracy of the model, which is called negative transfer [6].

We bring here a study in the field of transfer learning applied to recent deep learning models, which has been growing faster and has delivered applications in different areas with state-of-the-art results. The advances in deep learning technology have enabled research in different new areas like deep reinforcement learning, deep unsupervised learning, learning visualization, generative adversarial networks, transfer learning, and many others.

In this survey, we'll cover the theory of transfer learning and review the different techniques applied to it. In section II, we bring an overview and the definitions used in the literature [6]–[9]. In section III, the different transfer learning strategies are presented: inductive, transductive and unsupervised. The section IV covers the motivation for usage and explanation for when to use and when to not use it, exploring what kind of knowledge should be transferred depending on the task, and also how to transfer. In section V, we discuss about transfer learning applied to deep learning. In section VI, we present different types of transfer learning. And finally, in section VII we cover trends and directions for future work.

II. Overview and Definition

The idea that people can apply knowledge already learned from one task to another, faster and with better success, led to the study of transfer learning. Humans have an intrinsic ability to transfer knowledge between tasks, for example on how to transfer knowledge from knowing how to skateboard to learn how to surf. This motivation was discussed in the NIPS-1995 workshop "Learning to Learn" [10] in order to discuss progress in knowledge consolidation and transfer. The topic was recently reinforced by Andrew Ng in a tutorial of NIPS-2016 called "Nuts and bolts of building AI applications using Deep Learning", who mentioned that transfer learning will be the next driver of machine learning commercial success [11].

The idea of transfer learning has been discussed in the literature since 1990's with different names: learning to learn [10], life-long learning [12], knowledge transfer [13], inductive transfer [14], multi-task learning [15], knowledge consolidation [16], context-sensitive learning [17], knowledge-based inductive bias [18], meta learning [19], and incremental or cumulative learning [20]. In fact, all of these methods aim to extract knowledge from one or more tasks and apply to a different new task, with an exception for the multi-task learning that usually learns both tasks simultaneously.

As usually defined by different authors [6]–[8], [21], learning-based models need labeled data for training, and it's almost impossible to train a supervised learning method in a target domain with just a few labeled data. In semi-supervised learning it is possible to have a large amount of unlabeled data and a small set of labeled data to build a classifier, assuming that source and target tasks have the same distribution of data. But in these cases, once the feature space changes the model is not applicable anymore and needs to be retrained with new data. The transfer learning methods and research has emerged to deal with this problem, allowing to transfer knowledge between different domains or tasks.

Figure 1 [9] shows the difference between a traditional learning process of two different tasks and the same tasks learned using a transfer learning method, which the task 1 is fully trained using a large dataset and task 2 uses the knowledge obtained from task 1 to improve accuracy and learn faster.

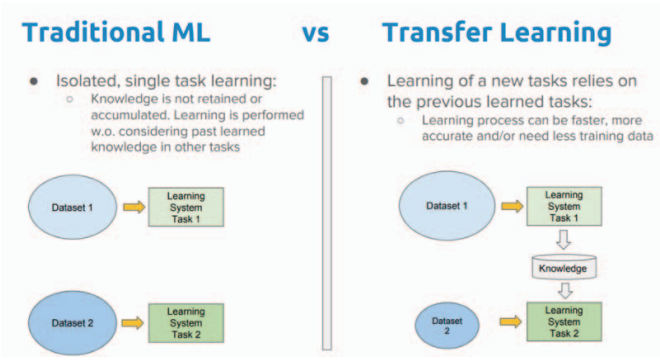

Figure 1. Difference between traditional learning process of two different tasks and transfer learning. In transfer learning the task 1 is fully trained using a large dataset and task 2 uses the knowledge obtained from task 1 to improve accuracy and learn faster. [9]

For reviewing the concept of transfer learning we get the definitions from [6], that split it into two high level concepts: Domain and Task. Most of the authors use the same definitions [7]–[9], [21].

A domain $\mathcal{D}$ is most related to the distribution of the data to be used for training. It's given by a feature space χ and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \chi$. In this case χ is the entire space of possible features, x_i is the i^{th} feature contained in some samples of the training data and X is a particular learning sample with a subset of features from χ. For considering two domains as different, they must have different feature spaces or different marginal probability distributions.

A task $\mathcal{T}$ is defined by a set of all possible labels $\mathcal{Y}$ and a model or predictive function $f(\cdot)$ that predicts a corresponding label based on a given domain $\mathcal{D}$. The task is denoted by $\mathcal{T} = \{\chi, f(\cdot)\}$ and can be learned from the training data, which consists in a collection of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ is used to predict a label y or $f(x)$ from an instance x.

Having a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a

target domain $\mathcal{D}_T$ and a learning task $\mathcal{T}_T$, *transfer Learning* is defined by [6] as a process that will help to improve the function $f_T(\cdot)$ in the target learning task based on the knowledge obtained from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $D_S \neq D_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

From the definition of domain and task, in Figure 2 we present the same transfer learning process from Figure 1 [22] but putting each term from the definition in its place. Both source and target domains share the same feature space, for example when both domains consist of only images.

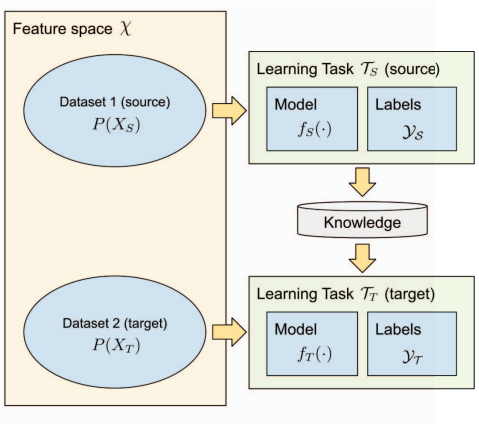

Figure 2. Representation of transfer learning from the definition of domain and task. In this case, both source and target domains shares the same feature space, while the marginal probability distributions are different. For example, each of them has its own dataset of different but related images to resolve two different tasks. Each task has its own model and labels.

Putting this definition in the context of computer vision and deep learning, let's consider χ all possible values that can be placed as input tensors of a convolutional neural network. Considering a domain with images, χ can contain a range of integers $\{0, ..., 255\}$ or floats $\{0, ..., 1\}$. $P(X)$ will be a distribution of values contained in χ that produces a set of images. In an image classification problem, $\mathcal{Y}$ have all the possible labels, for example cats or dogs. For the same task, $f(\cdot)$ will be the deep learning model trained to classify an image x into a label y, that can be one of the labels in $\mathcal{Y}$. The training process consists on update this model based on the domain $\mathcal{D}$. Using transfer learning, this model can be improved based on the knowledge obtained from a source domain $\mathcal{D}_S$ and a source task $\mathcal{T}_S$, as shown in Figure 2.

When the source and target domains are the same, and the source and target learning tasks are the same, it becomes a traditional machine learning problem. In this case, transfer learning is not applicable because both source and target are trying to resolve exactly the same problem.

III. Transfer Learning Strategies or Settings

Three important questions are discussed by some authors of different surveys regarding strategies for transfer learning: "What to transfer", "When to transfer" and "How to transfer" [6]–[9] . The first one is related to understand which part of knowledge from source can be used to improve the performance on the target, it's important to identify which part of the knowledge is specific or not. The algorithms that needs to be develop to transfer this knowledge will answer the "How to transfer" question. And the decision of transfer or not is made by the "When to transfer", that asks in which situation source domain and target domain are not enough related or won't help to improve the performance on the target, resulting in *negative transfer*.

Based on the definitions of transfer learning and based on different existing scenarios between source and target, [6] categorizes transfer learning under three settings: inductive transfer learning, transductive transfer learning and unsupervised transfer learning. In more recent studies these setting are also called strategies [9]. Figure 3 gives a good representation of these settings depending on data availability on source and target, considering that both are related.

A. Inductive Transfer Learning

This setting refers to the use case where the source and target tasks are different and domains can be related or not. Some labeled data in the target is needed to induce the target objective predictive model $f_T(\cdot)$, and using the inductive biases from the source domain this setting tries to improve the task in the target. [6] explains that depending on the labeled data availability from the source it can be split into two use cases, when the problem is similar to the multi-task learning or self-taught learning.

One common type of inductive learning is the idea of getting a pre-trained deep learning model using a source domain and task, and fine-tuning different layers for the target task [23], as presented in Figure 4, in the model at the right. For example, getting a pre-trained model with the ImageNet dataset, then fine-tuning it to recognize different classes of objects.

Another type is the multi-task learning itself, where multiple tasks are learned simultaneously from the same input [15], for example, from the same input image, the model will learn to perform a semantic segmentation and a depth estimation.

If there is not enough labeled data in the source domain, the inductive transfer learning is similar to a self-taught learning [24]. In this case, the information from the source domain cannot be used directly, because the label spaces may be different, but the unlabeled data can be used to improve the performance on the target task. The source task can be trained to recognize patterns in the unlabeled data and then these patterns can be used within a supervised learning task in the target.

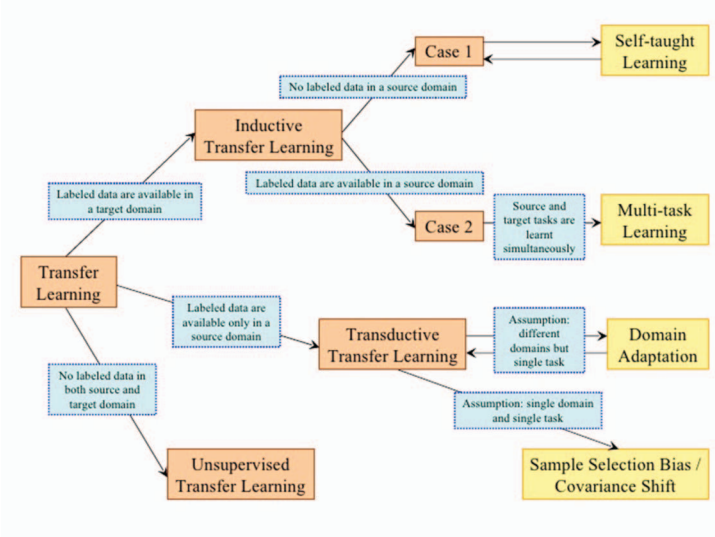

Figure 3. This diagram gives an overview of different transfer learning settings presented by [6].

B. Transductive Transfer Learning

In this setting, both source and target tasks are similar but domains are different. There is no labeled data available in the target, only in the source, which is similar to semi-supervised learning. When the feature spaces between domains are the same, $\chi_S = \chi_T$, but the marginal probability distributions of input data are different, $P(X_S) \neq P(X_T)$, the transductive transfer learning is related to a domain adaptation [25], sample selection bias [26] or covariate shift [27]. One example of transductive transfer is the use of synthetic images in a source task to boost performance on a target task with real-world images.

C. Unsupervised Transfer Learning

This setting is similar to the *inductive transfer learning*, the tasks are different between the target and the source, but the unsupervised transfer learning focus on resolve unsupervised learning tasks in the target domain, like clustering and dimensionality reduction [28], [29]. The labeled data is unavailable in both domains.

IV. Transfer Learning Approaches

Using the three described settings, there are different approaches that can be applied in terms of "What to transfer" [6]. The first approach is called *instance transfer*, a very common use case where parts of the source domain can be reused in the target domain for learning [30]–[38]. This process can be called as re-weighting [6]. In some cases, the data from the source domain can be reused as part of the target domain to improve the target task.

The second approach is called *feature representation transfer*, where the learned feature representations from the source are used to improve the target task. In this use case, it is expected that the data in the source domain to be enough to generate a good feature representation for the target. The knowledge is encoded into a feature representation. A good example of this approach is the use of well know pre-trained models with the ImageNet [2] dataset that became popular with the recent advances in the field of deep learning. Recent researches has shown that intermediate representations learned from ImageNet also provides substantial gains over hand-engineered features when transferred to other tasks [39]–[41]. Other experiments have shown that the CNNs pre-trained with ImageNet dataset are biased towards texture [42], causing negative transfer.

Next, there is the *parameter transfer* approach, where source and target tasks shares some parameters or prior distribution of hyper-parameters of the models [43]–[47]. The transferred knowledge is, in fact, the shared parameters between the models. It is different of multi-task learning, where both tasks are learned simultaneously. In parameter transfer there are two models, and just some of the parameters may be different.

The last one is the *relational knowledge transfer*, that different of the three above deals with the transfer of knowledge between relational domains [48]–[50]. In this approach, there is an assumption that the data in the source and target contains similar relationship and the knowledge to be transferred is the relationship among the data. It's common for models that use social network data to implement relational transfer learning.

Table I shows the relationship between different transfer learning settings and what to transfer.

V. Transfer Learning Applied to Deep Learning

With the advances in the field of deep learning, research related to transfer learning has focused on this kind of

Table I
RELATIONSHIP BETWEEN DIFFERENT TRANSFER LEARNING SETTINGS AND WHAT TO TRANSFER. [6]

	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
Instance Transfer	X	X	
Feature Representation Transfer	X	X	X
Parameter Transfer	X		
Relational Knowledge Transfer	X		

neural network architecture [7], [22], [39], [51]–[53]. A recent paper reviews transfer learning methods for deep learning [8] and categorize it into four approaches. The first one is called *instance-based* transfer, which is exactly the same idea described by [6] for instance transfer learning, where parts of the source domain can be reused in the target domain for learning. The second approach is called *mapping-based* transfer, which uses data from both source and target domains to create a new data space for training. Next, the author describes the *network-based* transfer, which reuses part of the neural network pre-trained in the source domain. This approach is the same described by [6] for the feature representation transfer. Finally, the author briefly introduces the *adversarial-based* transfer [54]–[59], which applies a technology inspired by the generative adversarial networks (GANs) [60] to find generative features suitable for source and target domains.

It's known that deep learning models requires a lot of labeled data to be trained, but this problem can be reduced using different transfer learning approaches, like learning useful representations from unlabeled data or transfer learned representations from a related task. The author of the book "Hands-On Transfer Learning with Python" [9] relates the deep learning models to the inductive transfer learning setting, described here in the section III-A, using the inductive biases of the source task to improve the target task. Instead of training a deep learning model from scratch with a lot of data and taking days of training, it's possible to get a model trained on a different domain for a different source task and adapt it to the desired target task.

There are many deep learning networks that has achieved state of the art performance within different tasks, like the VGG [61], Inception [62], [63], ResNet [64], MobileNet [65], SSD [66], and many others. Most of these models are publicly available. In many cases, the authors share details of the implementation and source code or even make the pre-trained models available for download with the most used deep learning frameworks. The key point on having these models available is that they already achieved a stable performance on specific tasks and can be safely used for transfer learning with related tasks.

Deep learning systems are commonly built with a sequence of convolutional layers and max-pooling layers to learn hierarchical representations of the data. These layers work as feature extractor connected to the final layers, responsible to perform a specific task, like classification or regression. Usually, these final layers contain fully connected neurons. This architecture allows us to use a pre-trained model to perform transfer learning by replacing the last layers and then fine-tuning the model for a target task, which is an implementation of the *instance transfer* approach.

Still using a pre-trained model, it is possible to freeze the first layers, making part of the source model to work as a feature extractor for the last layers in the target task, which is an implementation of the *feature representation* approach. Also, the deep learning architecture allows us to use a mixture of both approaches by selecting which layers to freeze of fine-tune, which can be useful depending on the source and target domains. These variations are shown in Figure 4.

In order to decide between fine-tuning or freeze, it's important to consider the distribution of data. Consider a source model pre-trained using thousands of classes of photographies. If the target domain also contains photographies in the same classes to resolve a different task, probably *freezing* layers will give good results because of common feature representations contained in both source and target domains. However, if the target domain contains images of documents, probably it will be necessary to unfreeze and *fine-tune* some or all layers in order to make the model to adapt to the new distribution contained in the target domain. Due to the hierarchical representation learned by the deep learning models, the first layers commonly recognize basic representations, like edges and corners, and the following layers learn more complex representations, like parts of objects. In other words, a dataset of documents will not have complex representations contained in photographies, but parts of text existing in the figures may have edges or corners that can be beneficial to be transferred. Thus depending on the distribution of the data between the source and target domains, different layers can be frozen to improve the performance on the target task.

VI. TYPES OF TRANSFER LEARNING

The research of transfer learning appear in the literature with different names, sometimes focused on different techniques and sometimes just with a different name. [9] describes five types of transfer learning and explain that all of them has the same goal to try to solve a target task using source task-domain knowledge. In addition to these five types, we describe here one more variants of transfer

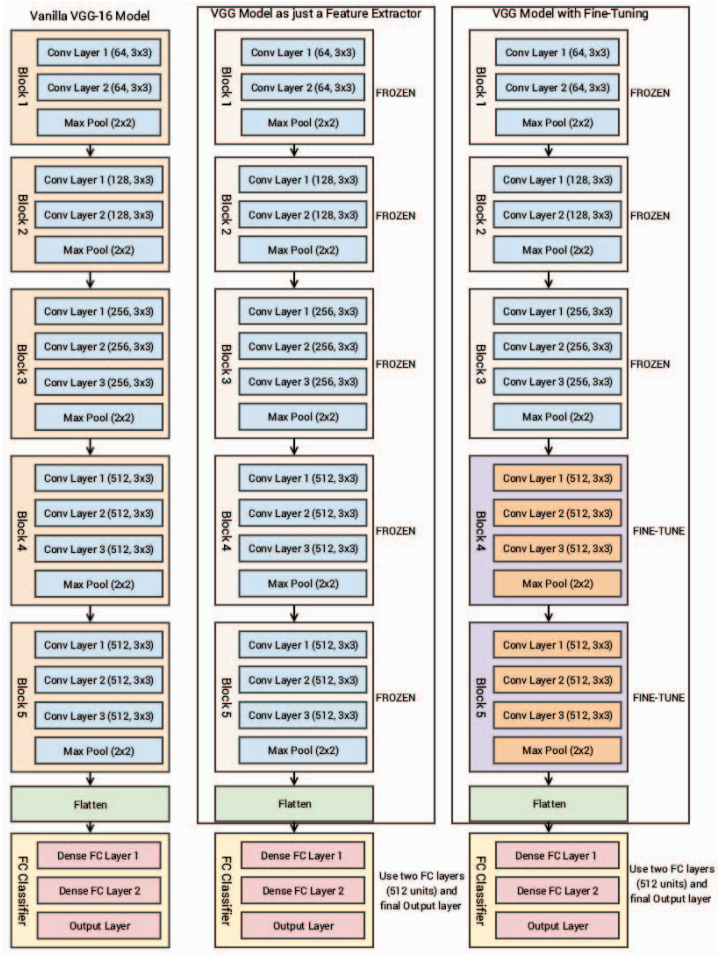

Figure 4. Difference between fine-tune and freeze layers using a VGG model. [9]

learning that we consider promising for the future of deep learning and machine learning in general, the meta-learning.

Domain Adaptation: Refers to the scenario where the feature spaces are similar between source and target but the marginal probability distributions are different, $P(X_S) \neq P(X_T)$. There is a shift between both domains that requires to implement this technique prior to use transfer learning. For example, consider a task of crowd counting and a source dataset of synthetic crowd scene, and a target dataset of real-world crowd scene, both represent the same content, but a domain adaptation technique such as a style transfer may be needed to generate photorealistic images and use transfer learning in this scenario [67], [68]. By not applying domain adaptation in this example, may cause the target learning task to not generalize well on real-world images. A promising technology to be used for this is the adversarial-based learning [51], [54], which is a domain adaptation method of transfer learning that uses the same concept developed for the generative adversarial networks (GANs) [60] to find generative features suitable for source and target domains.

Domain Confusion: This technique takes advantage of the feature representation learned by deep learning models to learn domain-invariant features and improve transferability accross domains. Instead of making the model to learn any representation, the idea is to push representations in both domains to be as similar as possible. There are recent deep learning architectures focused on implementing this kind of technique, for example adding a domain confusion loss in addition to the classification loss [51] or aligning the distributions of the source and target features in an unsupervised manner [22].

Multitask Learning: Although not being exclusive of deep learning applications, it's commonly used in this kind of system due to the simplicity of implementation and the advantage of easily split feature extraction and classification layers. Multitask learning is a paradigm in machine learning and its aim to leverage useful information contained in

multiple related tasks to help improve the generalization performance of all the tasks [69]. This is a different type of transfer learning, where both source and target tasks are learned at the same time, taking advantage of the common features existing in both domains to obtain the best balance in performance for both tasks.

One-Shot Learning: Different of human learning, machine learning systems requires a lot of data to learn from it. A person can easily identify an object after having seen it for the first time, but not the existing machines. Recent studies compared the capacity of learn from a single example between the machines and humans [70]. One-shot learning is a variant of transfer learning, that tries to infer the required output based on just one or a few training examples. The "One-Shot Learning" term first appeared in a paper that presented this technique applied with a bayesian approach [71], but this technique has recently being explored using deep learning models [72]–[75].

Zero-Shot Learning: This technique aim to intelligently apply previously learned knowledge to help future recognition tasks. In particular, a major topic in this research area is building recognition models capable of recognizing novel visual categories that have no associated labelled training samples. It uses class attributes as aside information and transfer knowledge from source classes with labelled samples [76]–[78], it can also be referred as *translated learning* [79]. For example, presenting a picture of a zebra to the system that have never seen a zebra but has seen a horse and also taught that a zebra looks like a horse but with stripes.

Meta-Learning: The idea was first described by J. Schmidhuber as *learning to learn* in the late 1980s [80] and Y. Bengio et al. in early 1990s [19]. It has also been studied with modern approaches, for neural networks optimization [81], to find better network architectures [82], in few-shot image recognition [83] (related to one-shot learning and zero-shot learning) and fast reinforcement learning [84], [85]. These systems are trained with different variety of tasks such as they can learn new tasks faster and with few shot examples, based only on prior knowledge on how to resolve these tasks. An interesting area of research using meta-learning is the Model-Agnostic Meta-Learning (MAML) [86], which is intended to apply the idea with different tasks not tied to the model architecture.

VII. Conclusion

The idea of transfer learning is not new and can be used in both traditional machine learning and deep learning applications and both people in industry and academia can take advantage of it to improve the performance of trained models. In the NIPS 2016, Andrew Ng mentioned that transfer learning will be the next driver of machine learning commercial success [11]. We covered in this survey some background related to transfer learning to help in the understanding of concepts, definition, strategies and types of transfer learning.

An open question in the field of transfer learning is related to avoiding negative transfer, where the transfer of knowledge from the source to the target does not lead to any improvement. Avoiding negative transfer is crucial to implementing transfer learning. There is a need to study transferability between source domains or tasks and target domains or tasks [6]. There can be various reasons for negative transfer, such as cases when the source task is not sufficiently related to the target task, or if the transfer method could not leverage the relationship between the source and target tasks very well [9].

One new technique that has gained attention is called Generative Adversarial Networks (GANs) [60], which is considered the most interesting idea in the last 10 years in ML by most researchers. Considering the potential of this idea and the impressive results published in many papers, there is also an opportunity for improvement using transfer learning with these models. It can be useful to transfer knowledge learned by the generators or discriminators across different domains, and to make the training faster. Also, the idea has been inspired new methods for transfer learning, like using adversarial models for domain adaptation [51], [54] and generating photo-realistic images from synthetic data [59].

Collecting labeled data is tedious, time-consuming, and susceptible to GDPR concerns. Methods of transfer learning that help to train models with small amount of data are very promising areas of research, like one-shot learning, zero-shot learning, and meta-learning. But there are still open challenges when comparing this kind of transfer knowledge to humans capacity, that can learn quickly from just a few samples of data.

Acknowledgment

The authors wish to thank Fundo Mackenzie de Pesquisa (Mack-pesquisa) from Mackenzie Presbyterian University for the financial support for this research.

References

[1] European Parliament and Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), "Council regulation (EU) no 2016/679," 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014. [Online]. Available: http://arxiv.org/abs/1409.0575

[4] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[5] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *arXiv:1811.00982*, 2018.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[7] K. Weiss, T. M. Khoshgoftaar, and D. Wang Background, "A survey of transfer learning," *Journal of Big Data*, 2016. [Online]. Available: https://journalofbigdata.springeropen.com/track/pdf/10.1186/s40537-016-0043-6

[8] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *CoRR*, vol. abs/1808.01974, 2018. [Online]. Available: http://arxiv.org/abs/1808.01974

[9] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Packt Publishing, 2018. [Online]. Available: https://books.google.com.br/books?id=aPFsDwAAQBAJ

[10] R. Caruana, D. L. Silver, J. Baxter, T. M. Mitchell, L. Y. Pratt, and S. Thrun, "Learning to learn: knowledge consolidation and transfer in inductive systems," 1995.

[11] A. Y. Ng, "Nuts and bolts of building applications using deep learning," NIPS 2016, 2016. [Online]. Available: https://nips.cc/Conferences/2016/Schedule?showEvent=6203

[12] D. L. Silver and R. E. Mercer, "The task rehearsal method of life-long learning: Overcoming impoverished data," in *Advances in Artificial Intelligence*, R. Cohen and B. Spencer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 90–101.

[13] J. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 7304–7308.

[14] U. Rückert and S. Kramer, "Kernel-based inductive transfer," in *Machine Learning and Knowledge Discovery in Databases*, W. Daelemans, B. Goethals, and K. Morik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 220–233.

[15] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[16] D. Silver, "The consolidation of task knowledge for lifelong machine learning," 03 2013.

[17] D. L. Silver and R. Poirier, "Context-sensitive mtl networks for machine lifelong learning," in *FLAIRS Conference*, 2007.

[18] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *ICML*, 1993.

[19] Y. Bengio, S. Bengio, and J. Cloutier, "Learning a synaptic learning rule," in *IJCNN-91-Seattle International Joint Conference on Neural Networks*, vol. ii, July 1991, pp. 969 vol.2–.

[20] S. Thrun and L. Pratt, Eds., *Learning to Learn*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[21] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence," *Know.-Based Syst.*, vol. 80, no. C, pp. 14–23, May 2015. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2015.01.010

[22] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, pp. 2058–2065. [Online]. Available: http://dl.acm.org/citation.cfm?id=3016100.3016186

[23] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. S. Feris, "Spottune: Transfer learning through adaptive fine-tuning," *CoRR*, vol. abs/1811.08737, 2018. [Online]. Available: http://arxiv.org/abs/1811.08737

[24] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 759–766. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273592

[25] H. Daumé, III and D. Marcu, "Domain adaptation for statistical classifiers," *J. Artif. Int. Res.*, vol. 26, no. 1, pp. 101–126, May 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622559.1622562

[26] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 114–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015425

[27] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, Oct. 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0M-4136355-5/1/6432c256e0be03b1503bbf79e4e91d1a

[28] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 200–207. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390182

[29] Z. Wang, Y. Song, and C. Zhang, "Transferred dimensionality reduction," in *Machine Learning and Knowledge Discovery in Databases*, W. Daelemans, B. Goethals, and K. Morik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 550–565.

[30] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 193–200. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273521

[31] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'07. AAAI Press, 2007, pp. 540–545. [Online]. Available: http://dl.acm.org/citation.cfm?id=1619645.1619732

[32] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[33] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 264–271. [Online]. Available: https://www.aclweb.org/anthology/P07-1034

[34] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 505–512. [Online]. Available: http://doi.acm.org/10.1145/1102351.1102415

[35] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06. Cambridge, MA, USA: MIT Press, 2006, pp. 601–608. [Online]. Available: http://dl.acm.org/citation.cfm?id=2976456.2976532

[36] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 81–88. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273507

[37] M. Sugiyama, S. Nakajima, H. Kashima, P. v. Bünau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07. USA: Curran Associates Inc., 2007, pp. 1433–1440. [Online]. Available: http://dl.acm.org/citation.cfm?id=2981562.2981742

[38] Wei Fan, I. Davidson, B. Zadrozny, and P. S. Yu, "An improved categorization of classifier's sensitivity on sample selection bias," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Nov 2005, pp. 4 pp.–.

[39] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" May 2018.

[40] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 647–655. [Online]. Available: http://proceedings.mlr.press/v32/donahue14.html

[41] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 512–519. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2014.131

[42] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," nov 2018. [Online]. Available: http://arxiv.org/abs/1811.12231

[43] N. D. Lawrence and J. C. Platt, "Learning to learn with the informative vector machine," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 65–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015382

[44] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task gaussian process prediction," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07. USA: Curran Associates Inc., 2007, pp. 153–160. [Online]. Available: http://dl.acm.org/citation.cfm?id=2981562.2981582

[45] A. Schwaighofer, V. Tresp, and K. Yu, "Learning gaussian process kernels via hierarchical bayes," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, ser. NIPS'04. Cambridge, MA, USA: MIT Press, 2004, pp. 1209–1216. [Online]. Available: http://dl.acm.org/citation.cfm?id=2976040.2976192

[46] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 109–117. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014067

[47] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 283–291. [Online]. Available: http://doi.acm.org/10.1145/1401890.1401928

[48] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'07. AAAI Press, 2007,

pp. 608–614. [Online]. Available: http://dl.acm.org/citation.cfm?id=1619645.1619743

[49] R. Liang, W. Xie, W. Li, H. Wang, J. J. Wang, and L. Taylor, "A novel transfer learning method based on common space mapping and weighted domain matching," *CoRR*, vol. abs/1608.04581, 2016. [Online]. Available: http://arxiv.org/abs/1608.04581

[50] J. Davis and P. Domingos, "Deep transfer via second-order markov logic," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 217–224. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553402

[51] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, Jan. 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=2946645.2946704

[52] M. Wulfmeier, A. Bewley, and I. Posner, "Addressing appearance change in outdoor robotics with adversarial domain adaptation," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1551–1558, 2017.

[53] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *ArXiv*, vol. abs/1412.3474, 2014.

[54] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," *stat*, vol. 1050, 12 2014.

[55] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 1180–1189. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045244

[56] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 4068–4076. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.463

[57] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2962–2971.

[58] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Domain adaptation with randomized multilinear adversarial networks," *CoRR*, vol. abs/1705.10667, 2017. [Online]. Available: http://arxiv.org/abs/1705.10667

[59] A. Dundar, M. Liu, T. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," *CoRR*, vol. abs/1807.09384, 2018. [Online]. Available: http://arxiv.org/abs/1807.09384

[60] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969125

[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for Large-Scale image recognition," Sep. 2014.

[62] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[63] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[65] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[66] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[67] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," *CoRR*, vol. abs/1903.03303, 2019. [Online]. Available: http://arxiv.org/abs/1903.03303

[68] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Lim, and R. Chellappa, "Unsupervised domain adaptation for semantic segmentation with gans," *CoRR*, vol. abs/1711.06969, 2017. [Online]. Available: http://arxiv.org/abs/1711.06969

[69] Y. Zhang and Q. Yang, "A survey on multi-task learning," *CoRR*, vol. abs/1707.08114, 2017. [Online]. Available: http://arxiv.org/abs/1707.08114

[70] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, pp. 1332–1338, 2015.

[71] Li Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, April 2006.

[72] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap, "One-shot learning with memory-augmented neural networks," *CoRR*, vol. abs/1605.06065, 2016. [Online]. Available: http://arxiv.org/abs/1605.06065

[73] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," 2015.

[74] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 3637–3645. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157504

[75] E. Zakharov, A. Shysheya, E. Burkov, and V. S. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *CoRR*, vol. abs/1905.08233, 2019. [Online]. Available: http://arxiv.org/abs/1905.08233

[76] Y. Fu, T. Xiang, Y. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition," *CoRR*, vol. abs/1710.04837, 2017. [Online]. Available: http://arxiv.org/abs/1710.04837

[77] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," *CoRR*, vol. abs/1611.05088, 2016. [Online]. Available: http://arxiv.org/abs/1611.05088

[78] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly," *CoRR*, vol. abs/1707.00600, 2017. [Online]. Available: http://arxiv.org/abs/1707.00600

[79] W. Dai, Y. Chen, G. rong Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 353–360.

[80] J. Schmidhuber, "Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook," Diploma Thesis, Technische Universitat Munchen, Germany, 14 May 1987. [Online]. Available: http://www.idsia.ch/~juergen/diploma.html

[81] K. Li and J. Malik, "Learning to optimize neural nets," *CoRR*, vol. abs/1703.00441, 2017. [Online]. Available: http://arxiv.org/abs/1703.00441

[82] R. Negrinho and G. J. Gordon, "Deeparchitect: Automatically designing and training deep architectures," *CoRR*, vol. abs/1704.08792, 2017. [Online]. Available: http://arxiv.org/abs/1704.08792

[83] B. Hariharan and R. B. Girshick, "Low-shot visual object recognition," *CoRR*, vol. abs/1606.02819, 2016. [Online]. Available: http://arxiv.org/abs/1606.02819

[84] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl^2: Fast reinforcement learning via slow reinforcement learning," *CoRR*, vol. abs/1611.02779, 2016. [Online]. Available: http://arxiv.org/abs/1611.02779

[85] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *CoRR*, vol. abs/1611.05763, 2016. [Online]. Available: http://arxiv.org/abs/1611.05763

[86] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th ICML*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70, 2017, pp. 1126–1135.