Sharing to learn and learning to share - Fitting together Meta-Learning, Multi-Task Learning, and Transfer Learning : A meta review

Richa Upadhyay richa.upadhyay@ltu.se
Luleå University of Technology,
Department of Computer Science, Electrical and Space Engineering,
Embedded Intelligent Systems Lab, 97187 Luleå, Sweden.

Ronald Phlypo ronald.phlypo@gipsa-lab.grenoble-inp.fr
University Grenoble Alpes, CNRS, Grenoble INP,
GIPSA-lab, 38000 Grenoble, France

Rajkumar Saini rajkumar.saini@ltu.se
Luleå University of Technology,
Department of Computer Science, Electrical and Space Engineering,
Embedded Intelligent Systems Lab, 97187 Luleå, Sweden.

Marcus Liwicki marcus.liwicki@ltu.se
Luleå University of Technology,
Department of Computer Science, Electrical and Space Engineering,
Embedded Intelligent Systems Lab, 97187 Luleå, Sweden.

Editor:

Abstract

Integrating knowledge across different domains is an essential feature of human learning. Learning paradigms like transfer learning, meta learning, and multi-task learning reflect the human learning process by exploiting the prior knowledge for new tasks, encouraging faster learning and good generalization for new tasks. This article gives a detailed view of these learning paradigms and their comparative analysis. The weakness of a learning algorithm turns out to be the strength of another, and thereby merging them is a prevalent trait in the literature. This work delivers a literature review of the articles that combines two algorithms to accomplish multiple tasks. A global generic learning network, an ensemble of meta learning, transfer learning, and multi-task learning, is also introduced here, along with some open research questions and directions for future research.

Keywords: Multi-task learning, Meta learning, Transfer learning, Knowledge sharing, Heterogeneous tasks, multi-modal inputs, Generalization on unseen tasks

1. Introduction

Machine Learning (ML) continuously draws inspiration from human cognition and decision making to develop more human-like, neurally-weighted algorithms (Fong et al. (2017)). One of the state-of-the-art models and successful tools in computer vision is the Convolutional Neural Networks (CNNs), which are inspired by the biological vision and neural activity. The traditional ML algorithms follow a single task learning approach wherein they are trained to solve only one task at a time. If there is a need to accomplish another task, the network needs to be re-trained on a new dataset. This type of learning is quoted as *isolated learning* by Liu (2017). It does not utilize and preserve any prior information from the previous learning for a future task. But humans don't learn anything from scratch, and they can rapidly learn new concepts due to their inherent potential of seamlessly sharing the acquired knowledge across tasks. The more related the tasks are, the easier it is to re-utilize the knowledge. For example, while learning to drive a car, the knowledge acquired while riding a bike or a motorbike comes in useful. The transfer of prior knowledge enables humans to learn quickly and accurately in few instances because humans have a bias that similar tasks have similar solutions (to some extent) therefore, they acquire the concept by focusing on learning the differences between the features of the tasks. Some of the learning techniques in ML or Deep Learning (DL), such as transfer learning, meta learning, Multi-Task Learning (MTL), lifelong learning, etc, are inspired from such human capability, where the aim is to transfer the learnings of one task to another task rather than training it from scratch.

1.1 What is a Task ?

Before moving further with the discussion about these information sharing learning paradigms, it is vital to understand the definition of a *task*. Formally task can be defined as a piece of work performed to fulfill a purpose. In this article, while discussing the learning paradigms, it should be noted that task does not refer to the process of learning. In fact, learning helps to acquire the competence required to execute a task. The most common kinds of tasks performed by various ML or DL algorithms are classification, regression, segmentation, machine translation, anomaly detection, dimension reduction, and several others. If two tasks are similar e.g., both are classification tasks, they are referred as homogeneous tasks. But if they are different e.g., one is classification and the other is segmentation, they are termed as heterogeneous tasks.

It should be noted that task and domain are considered different in this article. When there is only one task from each domain, researchers interchangeably use the term's tasks and domain. In this article, domain refers to the data distribution from where the training or test data is sampled. It is possible to have multiple similar types or diverse tasks in a domain. Overall it can be said that the tasks can be categorized depending on the labels or ground truth in the case of supervised learning, whereas the domain is related to the feature space of the data.

1.2 Knowledge transfer in machine learning algorithms

Learn one task at a time; this is a generic approach in the field of ML. Big problems are disassociated into smaller independent tasks that are learned distinctly, and combined results are presented. Caruana (1993) introduces MTL, wherein it is proposed that in order to have a better performance, all the tasks should be trained simultaneously. The underlying concept is that if all the smaller tasks share their learning, they may find it easier to learn than learning in isolation. This idea of *Multi-task learning* is very similar to the human vision system. For example, while looking at a scene, the brain is not just able to identify objects. Indeed, it can also segment them, understand them, identify people, classify the weather, and several other things from a single visual. In ML classification, segmentation, identification, etc are independent tasks, but MTL proposes to accomplish all these tasks jointly by exploiting the fact that in the above example, from the tasks point of view these are non identical but closely related tasks i.e., segmentation of a human from non-human in the picture will further aid in identifying the person.

Similarly, learning a complex task like riding a bike relies on the motor skills the human develops when they learn to walk as a baby. The task is therefore learned by integrating a considerable amount of prior knowledge across tasks. Also, while learning generalized concepts across many tasks, humans evolve the ability to learn fast and in fewer instances. These concepts are the bedrock for *transfer learning*, *lifelong learning* as well as *meta learning*. The approach of how the prior knowledge is introduced while training makes these learning algorithms mutually distinct.

1.3 Scope of this work

The emphasis of this article is on learning algorithms that use previously learned knowledge while learning an unseen but related task. Therefore, it mainly discussed three learning paradigms: multi-task learning, meta-learning, and transfer learning. Other related learning algorithms like Lifelong Learning (LL), online learning, and Reinforcement Learning (RL) are not within this work's scope because their fundamental objective is slightly different, i.e., progressive learning process. Also, reinforcement learning and online learning are solely restricted to single task learning only, while the others involve multiple tasks. LL has the key characteristics of consistent knowledge accumulation across several tasks and reusing it while learning a new task which makes it closely related to meta learning. However, the system architecture of LL (Chen et al. (2018b)) is fundamentally different from the other learning paradigms; it may require an ensemble of many learning algorithms and various knowledge representation methods. As for the evaluation of LL, many tasks and datasets are required to review the algorithm's performance, while learning the sequence of tasks is of great significance On the grounds of these differences, LL is excluded from this study.

1.4 Types of knowledge transfer

The information sharing between different tasks can result in two types of knowledge transfers: positive and negative transfer (Crawshaw (2020b); Zhang et al. (2021b)). Positive transfer is when the information shared between the tasks aids in improving the performance of the tasks, while the negative transfer is when the performance of the tasks suffers due to information flow within the tasks. Negative transfer is also known as destructive interference; it occurs because possibly even related tasks may have contradictory requirements, and when these are in a knowledge sharing setting, improving the performance of one task may hinder the performance of another. Negative transfer is a significant issue in multiple tasks learning concepts like transfer learning, MTL, etc. Therefore, to support positive transfer, it is important for an algorithm to fulfill two complementary objectives: retain task-specific knowledge, and better generalization across tasks. Accomplishing such goals is not at all that straightforward; consequently, it is an active area of research.

1.5 Contribution

The following are the important contributions of this work.

- An outline of the three learning algorithms, i.e., transfer learning, MTL, and meta learning, together with a comparative study of these learning paradigms focusing on their strengths and weakness.
- A detailed survey of the current research in multi-task meta learning, meta transfer learning, and multi-task transfer learning. This article highlights instances in the literature where because of abuse of terminology, these algorithms are often misunderstood as similar or derivatives of each other.
- This article proposes a novel generic learning network, an approach to combine these three learning paradigms in such a manner that gives the flexibility to utilize either all the three learning algorithms or combinations of two or just one of them as required.
- This work also presents the open questions in research related to knowledge sharing in multiple tasks learning arrangement.

1.6 Notations used in this article

Before moving to the next section it is important to know some of the notations used in this article. These are:

- Variables in calligraphic $(\mathcal{D}, \mathcal{X}, \mathcal{Y})$ - sample space
- Variables in italic (D, x, y) - one sample of the space
- p_i - joint distribution of i over the space
- $p(A)$ - marginal probability distribution of A
- $p(A|B)$ - conditional probability distribution of event A given B

1.7 Structure

The structure of the article is as follows: Sec. 2 gives the definitions and fundamentals of the learning paradigms along with supporting the explanation with suitable examples and also a comparative study. A detailed literature study of the learning algorithms when used together is presented in the Sec. 3. While Sec. 4 explains the reason of fusing these algorithms and also introduces a global learning network. At last Sec. 5 concludes the article with some open questions and directions for future research.

2. Learning paradigms

This section will elaborate on the three learning paradigms this article is keying on, particularly transfer learning, MTL, and meta learning. Along with an explicit definition, a comparison of these algorithms is also detailed below. But before this, an introduction to conventional supervised learning and its limitations, which direct to the reason for using information sharing algorithm, is also discussed.

2.1 Supervised Learning

The objective of ML algorithms is to learn from experiences. Here learning refers to improving by experience at a particular task (Mitchell (1997)). In supervised learning, which is one of the primary categories of ML algorithms, these experiences happen to be labeled datasets, wherein the algorithm is fed with the inputs and the expected output, and the goal is to learn a mapping from the input to the output. This article discusses the various knowledge transferring learning algorithms in the supervised sense only.

Definitions

Consider a domain $\mathcal{D}$, which is a combination of the input feature space $\mathcal{X}$, output space $\mathcal{Y}$, and an associated probability distribution $p(x, y)$, i.e., $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}, p(x, y)\}$. Here $p(x, y)$ represent the joint probability distribution over the feature-label space, and can be decomposed as $p(x, y) = p(x)p(y|x)$ or $p(x, y) = p(y)p(x|y)$, where $p(.)$ is the marginal distribution and $p(.|.)$ is the conditional distribution. The joint probability is used because the learning algorithms implicitly assume that each sample (x_i, y_i) is drawn from a joint distribution. In supervised setting the dataset D consists of input-output pairs, i.e., $D = \{(x_i, y_i)_{i=1}^{n}\}$, where x_i is a m-dimensional feature vector and y_i is the response or output variable which can be either a categorical variable or a real-valued scalar and n is the number of labelled samples.

In supervised learning, for a specific domain $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}, p(x, y)\}$, the aim is to learn a predictive function $\mathcal{F}_\theta(x)$ from the training data $\{(x_i, y_i)_{i=1}^{n}\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, and θ are the function parameters. From a probabilistic point of view, $\mathcal{F}_\theta(x)$ can also be considered as the conditional probability distribution $p(y|x)$. In case of a classification task, when the response variable y is categorical, i.e., $y \in \{1, ...C\}$ and C is the number of categories, supervised learning aims to predict;

$$\hat{y} = \mathcal{F}_\theta(x) = \underset{c \in [\![1,C]\!]}{\arg\max} \quad p(y = c|x) \tag{1}$$

This is known as Maximum A Posteriori (MAP) i.e., the most probable class label.

In conventional supervised learning, it is assumed to solve one task using one dataset, and so, the dataset is divided into development and test set. Here the test set represents the future unseen data, but both the sets are drawn from the same distribution (or the same domain), and both are labeled. But in reality, such a dataset is difficult to find, and therefore the supervised learning algorithm fails to generalize. For example, a model is trained using the handwritten digits dataset but is further used for inference on license plate images to detect the numbers automatically. So, for the model to perform on the data from another distribution, it needs to be trained again from the beginning using the new data. This loss of previous learning while learning new information is often referred to as catastrophic forgetting (Hasselmo (2017)). It is the biggest shortcoming in traditional supervised learning, which is said to be overcome in learning algorithms like transfer learning, MTL, meta learning, lifelong learning, etc, by sharing information between tasks.

2.2 Transfer Learning

Most ML models work under the assumption that the training and testing data are drawn from the same distribution (domain). If the distribution is changed, then the model needs to be trained again from scratch. Transfer learning helps to overcome this issue. Transfer learning refers to exploiting what has already been learned in one setting to improve the learning in another setting Goodfellow et al. (2016). Information transfer happens from the source domain (transfers knowledge to other tasks) to the target domain (use knowledge from other tasks).

Assuming only one source domain $\mathcal{D}_s = \{\mathcal{X}_s, \mathcal{Y}_s, p_s\}$ and target domain $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{Y}_t, p_t\}$ with p_s and p_t the joint distributions of the source and target data, respectively. Let source data be $D_s = \{\ (x_{s_i}, y_{s_i})_{i=1}^{n_s}\}$, where $x_{s_i} \in \mathcal{X}_s$ and $y_{s_i} \in \mathcal{Y}_s$ are the data instances and associated labels respectively. Similarly let the target data be $D_t = \{\ (x_{t_i}, y_{t_i})_{i=1}^{n_t}\}$, where $x_{t_i} \in \mathcal{X}_t$ and $y_{t_i} \in \mathcal{Y}_t$. Usually in transfer learning it is considered that $0 \leq n_t << n_s$ i.e., the source data is much larger than the target data.

Definition of transfer learning, as discussed by Pan et al. (2010):

> For a given source domain $\mathcal{D}_s$, and target domain $\mathcal{D}_t(\neq \mathcal{D}_s)$, transfer learning aims to improve the learning of the target prediction function $\mathcal{F}_{\theta t}(x_t)$ in domain $\mathcal{D}_t$ using the knowledge gained by performing a task in domain $\mathcal{D}_s$. The task in the source domain is to learn the source predictive function $\mathcal{F}_{\theta s}(x_s)$ from the training data D_s. Also the dataset D_s is not accessed during transfer.

In general, two domains are different if they differ in at least one of their components, i.e., input space, label space, or the probability density function. Based on these conditions, transfer learning can be classified as follows;

- Inductive transfer learning - in this $\mathcal{Y}_s \neq \mathcal{Y}_t$, and it doesn't matter whether the source and target input feature space are the same or different. Here there are two possibilities;

    1. *source domain has a lot of labeled data*; this is similar to Multi-task learning (Caruana (1997)) (discussed in section 2.3). But there is a difference, inductive transfer learning shares the knowledge of the source task to improve the learning of only the target task, while multi-task learning jointly learns both the task and attempts to improve the performance of both.
    2. *source domain has no labeled data*; this converges to Self-taught learning (Raina et al. (2007)). In the case of unlabelled data, the source task is to learn a good feature representation and use these learned feature representations to accomplish the target task.

- Transductive transfer learning (Arnold et al. (2007)) - in this $\mathcal{Y}_s = \mathcal{Y}_t$ but $\mathcal{X}_s \neq \mathcal{X}_t$, as well as it is assumed that the target domain has unlabelled data at the time of training. There can further be two cases-

    1. $\mathcal{X}_s \neq \mathcal{X}_t$ i.e., feature space is different
    2. $\mathcal{X}_s = \mathcal{X}_t$, but $p_s(x) \neq p_t(x)$ i.e., the marginal probability distribution is different for both the domains. This case is related to domain adaptation (Farahani et al.

(2020)). Furthermore, covariance shift is a condition in domain adaptation when along with $p_s(x) \neq p_t(x)$, the conditional distributions are constant $p_s(y|x) = p_t(y|x)$. And data drift or concept shift is the case when $p_s(x) = p_t(x)$, while $p_s(y|x) \neq p_t(y|x)$. These are the types of domain shifts explained by Farahani et al. (2020).

- Unsupervised transfer learning - there is no labeled data in both the domains and the aim is to solve unsupervised learning tasks in the target domain, for example, clustering, dimension reduction etc. using a large amount of data in the source domain. Self-taught clustering (Dai et al. (2008)) is one such instance of unsupervised transfer learning.

There are two primary approaches for transferring information between source and target tasks. The first is feature extraction; it uses the source model architecture and model parameters to extract good data features from the target domain to accomplish the target task. The second approach is fine-tuning; similar to the first, the source model shares its parameters and architecture with the target task, but the source parameters only serve as initialization to the target network, and further training is required using the target domain data. Very often, only the higher (last) layers of the network are modified and are trained, i.e., the parameters are altered to adapt to the new data, while the parameters of the lower (initial) layers are frozen, i.e., same as that from the source model. Therefore, the pre-trained source model is said to be fine-tuned according to the target data and fine-tuning also helps to improve generalization. Usually, transfer of learning, along with fine-tuning leads to better performance than training the target task from scratch.

Example of transfer learning

Consider a classification task of identifying cats from dogs as the target task, with limited training data. And a source task, say of identifying the breeds of dogs having a larger dataset than the target task. Since the conventional supervised machine learning models are data-hungry, it won't be easy to accomplish the target task with less data, assuming that the source task with larger data than the target performs sufficiently well. In this scenario, knowledge transfer from the source to target, if performed effectively, may enhance the performance of the target task. A common approach to knowledge transfer is using the model as well as the parameters of the source to extract features of the target data and appending a classifier, which will learn to classify the images of dogs and cats.

2.3 Multi-task Learning

MTL as explained by Caruana (1997), is an inductive transfer approach that exploits the domain information in the training data of related tasks as inductive bias to improve the generalization of all the tasks. The underlying theory of MTL is, the information gained while learning one task can help the other task to learn better. In MTL all the tasks are trained (or learned) together, Maurer et al. (2016) explains that shared representations significantly improve the performance of the tasks as compared to learning task individually.

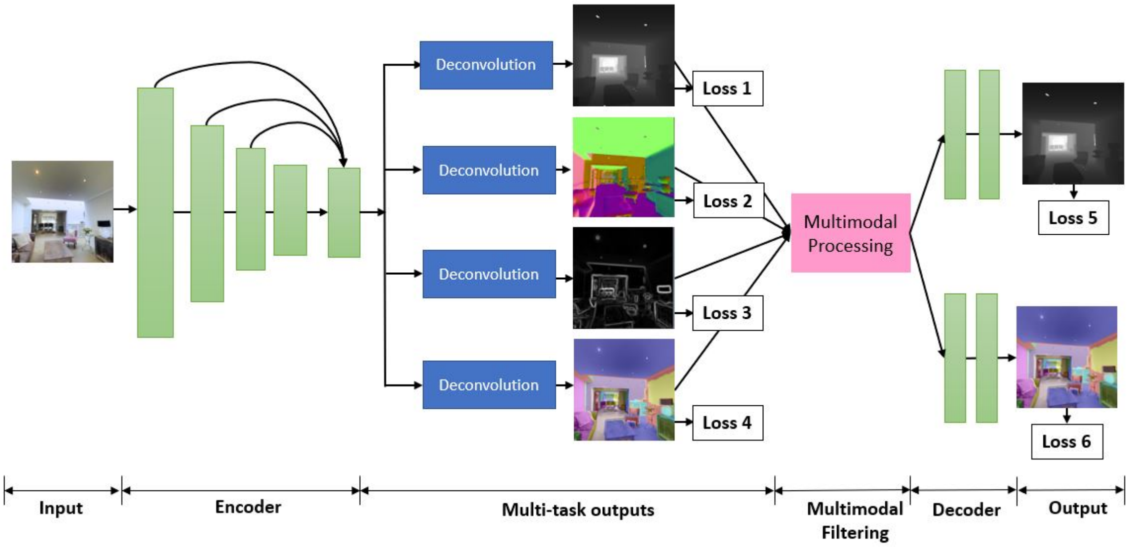


Figure 1: An illustration of the PAD-Net architecture proposed by Xu et al. (2018), there are four primary tasks of monocular depth estimation, semantic segmentation, finding surface normal and contour prediction and further the outputs are integrated for the prediction of two output tasks of depth estimation and scene parsing. Here Loss 1 - Loss 6 represent the optimization losses for the various tasks. All the images in the figure are from the dataset Taskonomy (Zamir et al. (2018))

Definition of MTL, as discussed by Chen et al. (2018b):

Consider, $\mathcal{T}$ is an ensemble of N related but not identical tasks i.e., $\mathcal{T} = \{T_1, T_2....T_N\}$ and each task $T_i \in \mathcal{T}$ has training data D_i^{Tr}. MTL aims to jointly learn these multiple tasks $\{T_1, T_2....T_N\}$ in order to maximize the performance of all the N tasks.

The objective of MTL is to learn optimal parameters θ^*, so as to minimize the combined loss $\mathcal{L}$ across each task. It can be expressed as:

$$\theta^* = \min_{\theta \in \Theta = \cup_{i=1}^{n} \Theta_i} \sum_{i=1}^{N} \mathcal{L}_i(\theta_i, D_i^{Tr}) \tag{2}$$

As discussed in section 1.4, in multi-task arrangement despite the tasks being related negative transfer can exist. It depends on the information sharing between the tasks and can be controlled by better MTL architecture designs and task relationship learning. In recent years there has been significant research on creating shared architectures for MTL, the article by Crawshaw (2020b) gives a survey of common deep MTL architectures used in the computer vision, natural language processing, reinforcement learning, etc. The deep MTL architectures can be divided into two types of modules (Goodfellow et al. (2016)):

Generic modules: these are shared across all tasks and the parameters are benefited from data of all tasks (correspond to $\Theta_g = \bigcap_i \Theta_i$);

Task-specific modules: these are dedicated modules for each task and the parameters are benefited from the instances of the particular task (corresponds to $\Theta_i - \Theta_g$).

Note, in this context modules are combinations of layers of neural network or the Convolutional Neural Network (CNN). As a result of these modules, the parameters are divided as shared parameters and task specific parameters. The best performance of MTL models is achieved only when there is balanced sharing, because too much sharing can cause negative transfer and too little sharing can inhibit the effective leveraging of information between tasks. Therefore, in order to create an effective MTL architecture it is important to analyze how to combine the shared modules (layers) and task specific modules and what portion of model's parameters will be shared between tasks. In conventional MTL, the parameter sharing approach is classified as (Crawshaw (2020a))-

- *hard parameter sharing* - model weights (or parameters) are shared between multiple tasks and each weight is modified to minimize multiple loss functions. This can be achieved as a result of MTL architectures.

- *soft parameter sharing* - tasks have separate weights and distance between the weights of models for every task added to the joint loss function is minimized, similar to introducing a regularization term in the combined loss. Therefore there is no sharing of parameters explicitly, rather models of different tasks are forced to have similar parameters. This is often introduced when there is negative transfer between the task and need to share less. The various optimization techniques help to achieve soft parameter sharing.

Example of MTL

Consider a dataset with images of natural scenes. Every image is labelled for a number of tasks like scene classification, semantic segmentation, instance segmentation and pixel wise depth values for depth estimation. For conventional machine learning all these are four different tasks, and it is required to train different models for each of the problems. MTL, however, exploits the facts that all the tasks are related and use the same input image. Therefore, they can be trained together in a MTL architecture, so that they share the representations between tasks and encourage the model to generalize better than single task learning. A similar architecture is proposed by Xu et al. (2018) in Fig .1. Usually in MTL, it is also believed that there is one main task and rest are auxiliary tasks which only contribute to improve the performance of the main task. In this article, there is no such assumption of main and auxiliary task. Furthermore in the above example, additional tasks like learning image compression and decompression, colorization of gray-scale images, denoising of images, etc can also be integrated in the architecture.

2.4 Meta Learning

Meta Learning better known as "*Learning to learn*" (Thrun and Pratt (1998)), is a learning paradigm that aims to improve the learning of new tasks with lesser data and computation, by exploiting the experience gained over multiple training episodes for various tasks. The conventional ML algorithm employs multiple data instances for better model predictions, while meta learning uses multiple learning instances to improve the performance of a learning algorithm.

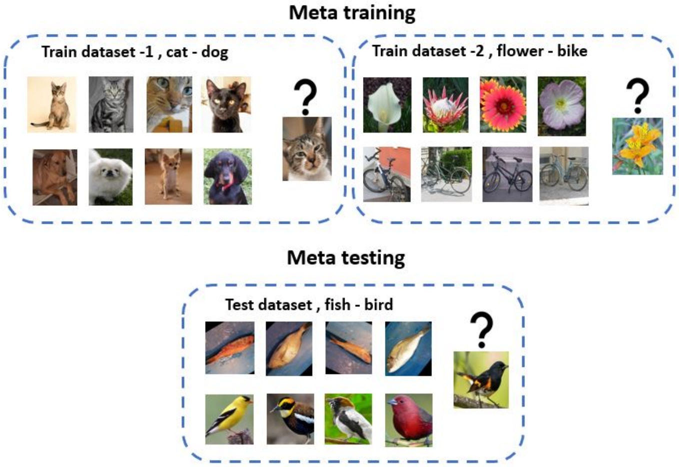

Meta training
Train dataset -1 , cat - dog
?
Train dataset -2 , flower - bike
?
Meta testing
Test dataset , fish - bird
?

Figure 2: An example of meta learning illustrating 4 shot 2 class image classification

Meta learning can be defined as:

Assuming a set of M source tasks $\mathcal{T}_s$, sampled from a distribution $p(\mathcal{T})$, with source datasets D_s. And Q target tasks $\mathcal{T}_t$ with target datasets D_t. Meta learning is to train a model on the M source tasks using data D_s, such that it generalizes well on a new unseen target task, which leads to:

- Computational efficiency - faster training of the target task using data D_t^{train}
- Data efficiency - good training with less target data instances
- Effective knowledge transfer - good performance on the target test data D_t^{test}

Let a task $\mathcal{T}$ be defined as, $\mathcal{T} = \{\mathcal{L}, D\}$, where $D = \{(x_1, y_1), ..(x_N, y_N)\}$ is training dataset and $\mathcal{L}$ is the loss function. For a single task conventional supervised ML algorithm, the aim of learning a model $\hat{y} = f_\theta(x)$ parameterized by θ, is accomplished by solving:

$$\theta^*(\mathcal{T}) = \arg\min_\theta \mathcal{L}(\mathcal{D}; \theta, \phi) \tag{3}$$

Here ϕ denotes the "how to learn" assumptions (Hospedales et al. (2020)), for example the optimizer for θ, choice of hyper-parameters, etc. A pre-specified ϕ can help to achieve significant performance as compared to the case when it is absent.

Meta learning involves learning a generic algorithm by training over several tasks, that enables each new task to learn better than the previous. Therefore,for a distribution of task $p(\mathcal{T})$, meta learning becomes:

$$\min_\phi \mathop{\mathbb{E}}_{\mathcal{T}\sim p(\mathcal{T})} \mathcal{L}(D; \theta^\star(\mathcal{T}), \phi) \tag{4}$$

Where $\mathcal{L}(D; \theta^\star(\mathcal{T}), \phi)$ evaluates the model's performance trained using ϕ on task $\mathcal{T}$. $\theta^\star(\mathcal{T})$ is the optimal parameter learnt for task $\mathcal{T}$. Here the parameter ϕ is the *meta knowledge* or across task knowledge (Hospedales et al. (2020)).

To solve the meta learning problem, assuming M source tasks $\mathcal{T}_s$ sampled from $p(\mathcal{T})$, having dataset $D_s = \{(D_s^{train}, D_s^{val})^{(1)}, ..., (D_s^{train}, D_s^{val})^{(M)}\}$, with train (support) and validation (query) sets. Also, Q target tasks with data $D_t = \{(D_t^{train}, D_t^{test})^{(1)}, ..., (D_t^{train}, D_t^{test})^{(Q)}\}$, i.e., each task with train and test set.

The meta learning objective in eq.[4] is obtained in two stages,

- Meta training - The meta training stage can be posed as a bi-level optimization problem, where one optimization contains another optimization as a constraint. Here an inner learning algorithm solves a task, defined by dataset $D_s^{train(i)}$ and objective function $\mathcal{L}^{task}$. While in meta training an outer (meta) algorithm updates the inner algorithm in order to improve the outer objective $\mathcal{L}^{meta}$. So, meta training can be formulated as:
(outer Objective)

$$\phi^* = \arg\min_{\phi} \mathop{\mathbb{E}}_{\mathcal{T}_s \sim p(\mathcal{T})} \mathcal{L}^{meta}\ (\theta^{*(i)}(\phi), \phi, D_s^{val(i)}) \tag{5}$$

Where,
(inner objective)

$$\theta^{*(i)}(\phi) = \arg\min_{\theta}\ \mathcal{L}^{task}\ (\theta, \phi, D_s^{train(i)}) \tag{6}$$

ϕ^* has all the information of source tasks (or data) to solve new tasks. So, the inner objective corresponds to task specific learning, while the outer objective corresponds to multiple task learning.

- Meta testing - The meta testing stage is often referred as *adaptation* stage. This stage uses the meta knowledge or meta parameters (ϕ^*), to train the model on unseen target tasks. For an i$^{\text{th}}$ target task, meta testing involves training on the $D_t^{train(i)}$ to minimize the loss $\mathcal{L}^{test}$ and evaluating the performance on $D_t^{test(i)}$, for optimal parameter $\theta^{*(i)}$ given by:

$$\theta^{*(i)} = \arg\min_{\theta}\ \mathcal{L}^{test}(\theta, \phi_\star, D_t^{train(i)}) \tag{7}$$

However, MTL can be also be seen as a special case of meta learning if $\theta = \phi$ in the meta training phase, as there will be only one optimization objective and multiple tasks for training. Despite this similarity, there are many differences that persists between meta learning and MTL which are discussed in Sec. 2.5.

Example of meta learning

Fig. 2, shows an example of meta learning, where the source tasks are: Task 1- classification between cats and dogs, and Task 2- classifying flowers from bikes. The target task is to classify between images of fish and bird which the model has not seen during the meta training phase. This is a classic example of 4 shot 2 class image classification, where the objective is to learn to identify the categories only by 4 images and every tasks has 2 classes or labels. So, meta learning enables the model to learn fast in a few instances of dogs and otters images during meta testing by utilizing the meta knowledge gained from meta training of the source tasks. The number of source tasks can be increased in order to achieve better generalization.

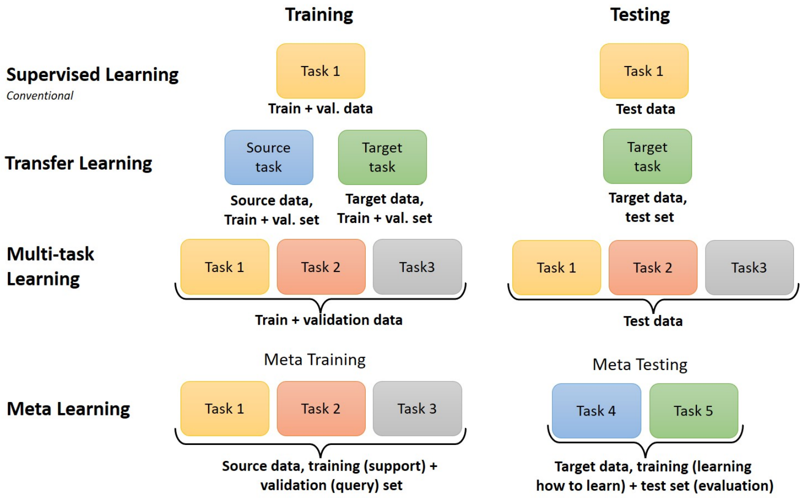


Figure 3: A comparative representation of the learning paradigms

2.5 Comparisons

From the above sections which details about the learning paradigms, it is clear that all the three transfer knowledge between tasks, the difference lies in how, when and what knowledge is shared between the tasks. This section presents a comparative view of the three learning paradigms. It might not cover all the differences but focuses on the significant dissimilarities and also similarities.

Transfer learning and MTL

The tasks involved in transfer learning and MTL can be heterogeneous, i.e., the nature of the tasks can be different like classification, segmentation, regression, etc. These learning paradigms can share features and parameters between the tasks. Because of the inductive transfer approach, MTL is also considered as a type of transfer learning (Pan et al. (2010)), but they are very different. MTL learns many tasks together, while it is not the case in transfer learning, where first the source task is trained, and the information is transferred for learning the target task. In other words, training tasks in transfer learning are sequential, while in MTL, tasks are usually trained simultaneously (or jointly). In MTL, the goal is to generalize the performance of all the tasks, while in transfer learning, the focus is only on the generalization of the target domain. Transfer learning is a logical explanation for multi-task learning but not vice versa.

Meta learning and Transfer learning

In both these learning paradigms, the target tasks are trained after the successful training and testing of the source task, i.e., sequentially. Both algorithms aim to achieve better generalization on the target task. The key difference between both lies in the optimization algorithm. In transfer learning there is no meta objective while deriving priors (parameters) from learning the source task. In contrast, in meta learning the priors are extracted as a result of the outer optimization and these are evaluated while learning a new task. Only model parameters are shared in transfer learning, while meta learning transfers a variety of meta representations. Also, in meta learning the meta training (source) and meta testing (target) conditions must match, e.g., if the source tasks are binary classification problems, then the target task necessarily should be binary classification. On the contrary, transfer learning is possible on diverse tasks.

Meta learning and Multi-task learning

Like transfer learning, MTL has single level optimization, i.e., no meta objective. MTL aims to solve fixed number of known tasks, whereas meta learning deals with solving unseen tasks. Meta learning, as discussed earlier, only works with homogeneous tasks and same modality of inputs during meta training and testing. At the same time, MTL can handle heterogeneous tasks and also a variety of inputs. The source tasks are trained sequentially for many iterations during meta training in meta learning. A new unseen task is learned at the time of meta testing; on the contrary, in MTL, the training and testing are performed for all the tasks jointly. Therefore, the knowledge is shared between the tasks at the time of training in MTL, but for meta learning, the model saves the prior knowledge from the source tasks to be used during meta testing the target tasks.

3. Ensemble of the learning paradigms

The strength and weaknesses of the learning paradigms detailed in Sec. 2 are summarized in Tab. 1. Each algorithm has some drawbacks, which are overcome by another learning algorithm. Like meta learning supports homogeneous tasks and unimodal input, MTL may integrate heterogeneous tasks and multimodal inputs. Similarly, introducing a new task is much simpler in meta learning and transfer learning than MTL. For these reasons, these algorithms are coupled together in the literature to utilize their best features. This section will discuss the research performed in various ensembles of these algorithms.

3.1 Research in Multi-task meta learning

Insights from both meta learning and multitask learning can be fused to achieve the best of both worlds, i.e., efficient training of multiple heterogeneous tasks, a feature of MTL, and quickly adapting new tasks, a feature of meta learning. Thereby offering a fast, efficient, and adaptive learning mechanism. There have been many studies that demonstrate employing both of these learning paradigms together.

The article by Chen et al. (2018a) proposes a function level information sharing scheme for MTL, which employs a meta Long Short-Term Memory (LSTM) i.e., shared across all tasks and a basic LSTM which is task specific and the parameters are generated depending on the current factors by the meta LSTM. Here the meta LSTM is considered the prior knowledge while the basic LSTM is observed as the posterior knowledge. It focuses on two Natural Language Processing (NLP) tasks, i.e., text classification and sequence tagging. Overall the architecture they propose is a multitask architecture, and the two level optimization scheme is from meta learning. But, the performance is not evaluated for an unseen task. Nevertheless, they show that the performance of the proposed network is much better

Table 1: Strength and Weaknesses

Paradigms	Strengths	Weakness
Multi-task learning	- Can handle heterogeneous tasks and multimodal inputs - Aim is to enhance performance of all tasks	- Requires a huge dataset for training - On adding a new task have to retrain the network, as it may require architectural changes
Meta learning	- Easy addition of new tasks - Less training data required for a new unseen task - Robust generalization across tasks - Meta learning objective (bi-level optimization)	- Works for homogeneous tasks only, as source and target tasks must match - Only focuses on enhancing performance of target task
Transfer learning	- Less training data required for target task - Good for feature extraction	- Pretrained models tend to overfit on target task - Complex source models with millions of parameters, not always required for target tasks - Only focuses on enhancing performance of target task

than applying MTL and single task learning. Because adding meta learning to MTL leads to meta-knowledge transfer, which shares semantic composition function across tasks. In neural representation learning of text sequences, the semantic composition function is of extreme significance.

Zhang et al. (2021a) introduce a multimodal meta multitask learning approach for rumor detection and stance detection in social media, illustrated in Fig. 4. The multimodal inputs here are text and images, converted into embeddings and fused to the model. This article discusses the feature level, meta level, and task level challenges related to the problem statement. Therefore, suggests sharing the higher meta knowledge between tasks, i.e., sharing features in the initial layers of the model and further no sharing in the task specific lower layers. Here, the feature sharing between the tasks is considered meta-knowledge. Therefore, the article's title mentions meta multitask, but there is no meta learning applied. It neither solves the bi-level meta objective nor evaluates the performance on unseen tasks.

The Parameters Read-Write Networks (PRaWNs) are introduced by Liu and Huang (2018), which presents a novel concept of communication between the tasks in a multi-task setting. The idea here is to allow various tasks to pass the gradients explicitly, restricting the tasks to update parameters constantly. Therefore the gradients are passed between tasks

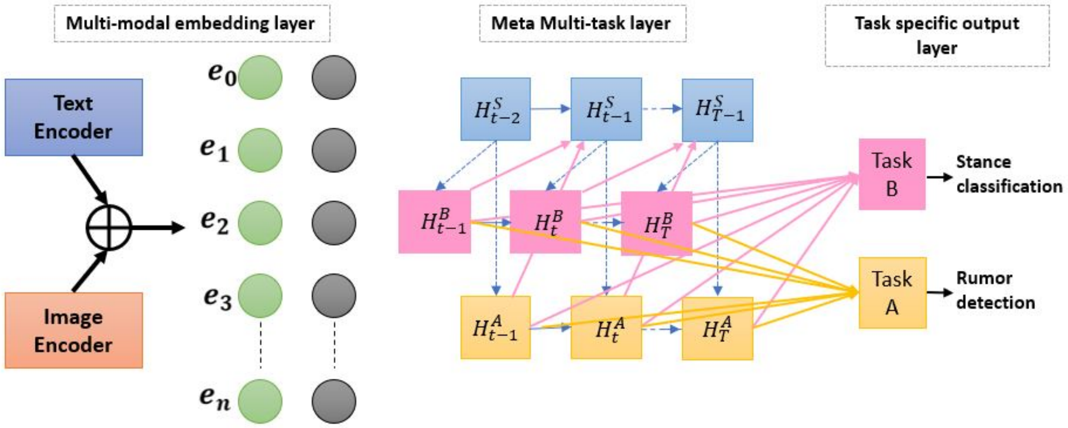


Figure 4: A multi-modal meta MTL framework proposed by Zhang et al. (2021a) for rumor detection in social media.

pairwise and list-wise, where list-wise gradients account for task relatedness. The features learned in the shared space are usually entangled. This proposed model helps untangle features in different domains, which as a result, the given use-case avoids the escape of private information in the shared space. Liu and Huang (2018) investigate the proposed network for text classification, sequence tagging, and image aesthetic assessment tasks for in-task (same dataset) and out-of-task (new dataset) settings. It is concluded that the introduced gradient passing mechanism for out-of-task settings shows better generalization to unseen tasks.

In a similar work, the authors Tarunesh et al. (2021) offer an approach to learn interactions between languages and tasks by employing meta learning in a multi-task scenario. Question answering, parts of speech tagging, name entity recognition, paraphrase identification, and natural language inference are the multitasks in consideration for this work. They exploit the fact that these tasks can be benefited from each other and also cross lingual embeddings in multiple language learning can be helpful in the case of languages with limited data. Therefore the meta information in this work is the task relatedness plus the language relatedness. They also propose sampling techniques like heuristic and parameterized sampling to be integrated into meta learning to improve the performance of the model. This approach can be very useful for languages with limited data resources as it performs significantly well on the zero-shot new target languages.

Personalized dialogue generation is improved by using Meta and MTL by Lee et al. (2021), which seeks to overcome the issue of a large dataset for every person and including pre-defined persona information. In this work, an auxiliary task of persona reconstruction is added only at the meta training stage to gather the persona information as meta-knowledge for the dialogue generation task. Thereby making the model able to generate dialogues for new users at the time of meta testing. This introduces two frameworks, Multi-Task Meta-Learning (MTML), which combines losses from both the tasks. and Alternating Multi-Task Meta-Learning (AMTML), which operates alternatively on the tasks of generation and persona reconstruction. Another similar work by Chen and Zhu (2020) aims at text style transfer with limited data, basically paraphrasing text from one writing style to another. It employs the Model Agnostic Meta Learning (MAML) (Finn et al. (2017)) algorithm for few shot text style transfer, and a task corresponds to a pair of styles, so viewing style transfer between each pair as a domain specific task, hence employing MTL. This enables to transfer (to and from) writing styles which have small training data and also data which the model hasn't seen before due to the meta learning framework. The proposed methodology outperforms the state-of-art results in text style transfer.

Lekkala and Itti (2020) combine concepts from meta learning, MTL and visual attention for image classification and estimation of depth, vanishing point, and surface normal form a single input image. The flexible attention mechanism is used to adapt the network for a particular task, and the features in the generic modules (or backbone) of the network are weighted according to the importance of the specific task. The task specific modules (or layers) are adapted to unseen tasks by employing the Almost No Inner-Loop MAML (ANI-MAML) (Raghu et al. (2020)) training procedure, hence introducing the role of meta learning in this work. The attention mechanism and meta learning made it possible for the tasks heads to learn to adapt new unseen tasks in lesser data instances. Along with learning the task specific representations, the primary issue was to learn task-invariant representations, the MTL architecture solves this by providing inductive bias on selected features as directed by the attention procedure.

Cai et al. (2020) introduces meta learning and MTL for speech emotion recognition. In this work, the emotion classification for each user is considered the auxiliary tasks used for meta training, also known as the multi-train stage. Next is the knowledge transfer stage wherein, the meta information from the training stage is used to train and evaluate the model for a new user. So, this article aims to model the relationship between the auxiliary tasks, i.e., users, and transfer knowledge to the target task. This is very much the concept of meta learning as discussed in Sec. 2; this work considers the multiple tasks during training as MTL and therefore mentions meta multi-task learning in the title.

The article by Liu et al. (2020) solves the challenge of data shortage and disease diversity in mortality prediction of rare diseases by employing a multi-task architecture along with the meta learning optimization scheme MAML (Finn et al. (2017)). Here the multiple tasks are detecting the temporal occurrences of rare diseases, and the input is multi-modal, i.e., text, images, signals, etc. A learning methods Ada-SiT (Adaptation to Similar Task) is introduced in this work, in which the task similarity is measured during meta training and this is used to share initialization for faster adaptation of new tasks.

All articles discussed above are the ones that appear (and are relevant) in the 'in-title' search for meta learning and MTL on google scholar. Some articles by Chen et al. (2018a); Cai et al. (2020); Zhang et al. (2021a); Liu and Huang (2018); Lee et al. (2021) refer the combination as *Meta Multi-task Learning* which focuses on improving multi-task learning by employing meta learning to gather meta knowledge (e.g., task relationships) to transfer it to new task. While a few by Lee et al. (2021); Chen and Zhu (2020); Lekkala and Itti (2020) mention *Multi-task Meta Learning* that focus on upgrading meta learning by introducing multi-task learning for allowing training with heterogeneous tasks as well as efficient training mechanism for better learning of features. In general, there is no logical explanation of the taxonomy used for employing the two algorithms together. Moreover, they can be used interchangeably as integrating these learning mechanisms is to leverage the qualities of both meta learning and MTL.

There are also a few articles by Bronskill et al. (2020); Ghadirzadeh et al. (2021); Kedia and Chinthakindi (2021); Tian et al. (2019); Lin et al. (2019); Krueger et al. (2020b) which by abuse of terminology refer to the training of multiple tasks in meta training stage as MTL, thereby considering the work as meta multi-task learning. Also, a few articles (Bansal et al. (2020); Zou and Lu (2020); Zintgraf et al. (2020); Zhou et al. (2020); Krueger et al. (2020a); Guo et al. (2020); Kim and Pavlovic (2020, 2021)) in the google scholar search for meta multi-task learning (or multi-task meta learning) merely mention these terms in related work or future work sections. In a similar search, a handful of articles (Retyk (2021); Ghosh et al. (2020a,b); Li et al. (2009)) focusing on meta multi-task reinforcement learning are also present. However, these are not discussed here as it is beyond the scope of this work.

3.2 Research in multi-task transfer learning

Transfer learning and MTL usually differ in how and when the information is shared between tasks. However, they can be used together in the following ways,

1. Transfer learning enables to extract features for tasks by employing models pre-trained for some other related tasks on a large dataset, which can further be used in a multi-task setting. For instance, when there are multiple related target tasks and thereby employing MTL to solve them, and source task is conventional supervised learning.

2. Another way of fusing these algorithms is when there are multiple source tasks such that MTL can be applied and the knowledge is shared with the target task in terms of model parameters, features, etc.

In the past, these two learning paradigms were used together a number of times. Ye et al. (2018) aim to predict the pharmacokinetic parameters by learning a model for quantitative structural activity relationships of drugs. The four parameters, i.e., oral bioavailability, plasma protein binding rate, apparent volume of distribution at steady-state, and elimination half-life, ought to be estimated in this work. These four parameters are considered multiple tasks. It follows the first approach discussed above of feature extraction and multiple tasks in the target domain. At first, a pretrained model is learned on the extensive bioactivity data set, the knowledge from which is further used in the multi-task DeepPharm model proposed in the article, as depicted in Fig. 5. The integrated multi-task and transfer learning approach proved to enhance the generalization of the model as compared to the conventional models, as well as overcome the issue of lack of sufficient and high quality data in Absorption, Distribution, Metabolism, and Excretion (ADME) evaluation. Such a well generalized model may be helpful to perform the ADME calculations on the new drug structures using the DeepPharm model, i.e., making inference only, as there is no need of training the model again on different data. This article opens a new dimension of research drug discovery and development using combining DL algorithms.

An example of the second type of fusion mentioned above, i.e., when the source task is MTL from which the learned information is transferred to a target task, is illustrated in an article by Hasan et al. (2020). The aim of this article is fault diagnosis of rolling

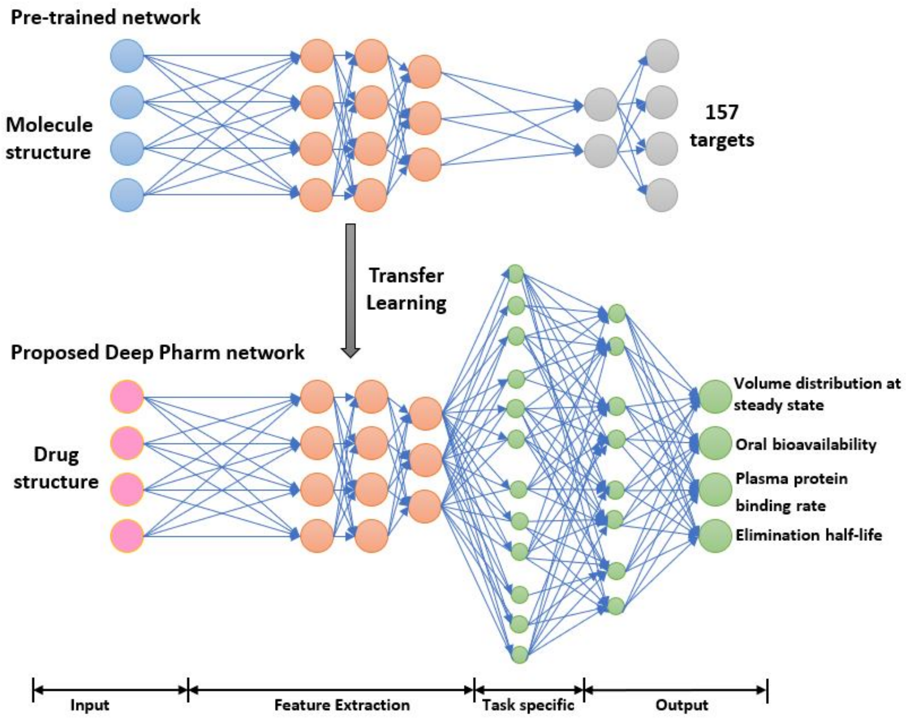


Figure 5: Illustration of the integration of transfer learning and MTL (Ye et al. (2018))

element bearings under uncertain working conditions. Firstly, MTL is enforced to determine the speed and health type of the machines (in general) by using the bi-spectrum based analysis of the vibration signals as inputs. Further, transfer learning is used to enhance the classification performance by using the proposed MTL-CNN as the source task and identifying the bearing faults under severe conditions as the target task. Since the faulty bearing data in extreme conditions is usually less (as this is a type of anomaly), the transfer learning technique is advantageous. The pre-trained model for extraction features in the target dataset in a multi-task setting leads to good performance on the unseen target data.

In Cruz et al. (2020), pretrained transfer learning models like BERT (Devlin et al. (2019)), ULMFiT (Howard and Ruder (2018)) and GPT-2 (Radford et al. (2019)), for fake news detection. Along with this, it also includes an auxiliary language modeling task to adapt to the writing style of the downstream task of fake news detection. Therefore, multi-task fine-tuning is introduced in this work by combining losses from both the tasks. In a similar work by Dong et al. (2019), transfer learning and MTL are used together for Named Entity Recognition (NER) on Chinese Electronic Medical Records (EMR). It trains a bi-directional LSTM in the general (source) domain and further uses the acquired knowledge from the source domain to improve the performance of NER in Chinese EMR, i.e., target domain. Parts-of-Speech (POS) tagging and NER are the two tasks in the target domain which are trained alternatively, so that knowledge form one task may enhance the knowledge gained by the other task. Since this work aims NER, the POS is treated as an auxiliary task that aids in better learning of Chinese NER. On similar grounds, Taslimipoor et al.

(2019) intend to classify the multi-word expressions with the help of two auxiliary tasks of dependency arcs and labels. To attain this, it utilizes knowledge shared by the pretrained model for POS and dependency parse tags for two different languages. This is because the target language has limited resources, and cross-lingual transfer learning helps overcome this issue.

Multi-task transfer learning is also exercised together in the article by Du et al. (2020) for neural decoding, which decodes the brain activity to reconstruct visual information. To achieve this, first, the fMRI voxel features of the brain are decoded into CNN features by using Structured Multi-output Regression (SMR) (i.e., Voxel2Unit). Further, the predicted CNN features are converted to an image using introspective conditional generation (i.e., Unit2Pixel). Multiple CNN fetaures are decoded from the FMRI data using SMR, and every single output prediction is considered a task, thereby applying MTL for the Voxel2Unit process. For the Unit2Pixel process, a pretrained CNN called AlexNet (Krizhevsky et al. (2012)), trained on large image data ImageNet (Deng et al. (2009)) is used for image reconstruction as a part of deep generative models, particularly a combination of Variational Auto-Encoders (VAEs) (Kingma and Welling (2014)) and Generative Adversarial Nets (GANs) (Goodfellow et al. (2014)).

The article by Nguyen et al. (2021) analyzes the characters of a digital Japanese comic named Manga, using Bi-modal inputs, i.e., the graphics and text information. It uses pretrained networks like BERT (Devlin et al. (2019)) and ResNet (He et al. (2015)) for text and visual feature extractor, respectively, in other words, to get the feature embeddings to combine data from both the modes. Further use these embeddings in a multi-task architecture for character retrieval, identification, and clustering tasks. These three tasks are independent; each can be accomplished by training three different models using the same images and text (uni-modal inputs can also be used). However, since they share common multi-modal inputs (image and text), they are suitable for MTL. Certainly, sharing of parameters between tasks improves the performance of all the tasks compared to when they are trained isolated. The strength of MTL i.e., handling multi-modal inputs and multiple tasks and transfer learning i.e., good feature representations is cherished in this work.

Qu et al. (2019), exploits a few-shot Dirichlet-Net based MTL for hyper-spectral image classification. An encoder-decoder network to reconstruct hyper-spectral images is used, and the representations are shared using the encoder in a classifier network. This ensemble of encoder-decoder and classifier is termed as MTL in this article. The encoder takes as input two images of different domains and extracts both representative and discriminative vectors representations from both the domains; these encoder network parameters are shared with the classification network for extracting features of the input image patches to predict output labels. The important contribution of this article is the extraction of shared representations from the objects in different domains by applying transfer learning.

Another exciting application by Dong and Khosla (2021), aims to estimate the virus-human protein interactions by pre-training a source model UniRep (Alley et al. (2019)) to produce the features or protein embeddings. Due to the scarcity of training data for virus-human protein interaction, the article first trains a network on powerful statistical protein representations, i.e., source task. And the target task performs MTL by extracting the embeddings using the source network to find human protein-protein interactions (PPI) and human virus PPI. Likewise, Aydin and Erdem (2019) use VGGNet (Simonyan and Zisserman (2015)) for pre-training, and the extracted features are used for detection and identification of copper and plastic wires buried in the ground, using Ground Penetrating radar (GPR) scans. Identifying the type of soil (wet or dry) in the target domain is also added, making it a MTL application. It is observed that in many of the articles, such as Cruz et al. (2020); Dong and Khosla (2021); Simonyan and Zisserman (2015), a source network is trained on a large dataset of one domain, which is exploited by the target dataset of another domain having less data resources. Thereby transfer learning helps to find good feature representations leading to better generalization. Along with the multi-task architecture, making it possible to train many tasks jointly.

Some articles that appear in search results for MTL and transfer learning on google scholar discuss these learning paradigms separately which is not the aim of this article. This section focuses on articles that have used the two learning algorithms together. Xu and Yang (2011), give a detailed survey of how transfer learning and MTL individually are used in bio-informatics. Kamath et al. (2019) is a book chapter that discusses transfer learning types and describes MTL as a variant of transfer learning. While Sun et al. (2019b); Ricci et al. (2017) details types of multi-view transfer learning and multi-view MTL. The articles Wang and Pineau (2015); Maurer et al. (2014) give theoretical concepts of improving the performance of MTL and transfer learning in general.

3.3 Research in Meta transfer learning

Meta learning and transfer learning are very similar like both have source tasks and unseen target tasks, and the objective is a better generalization on the target task. However, the optimization scheme differs; meta learning has a two-level optimization, whereas transfer learning does not. However, an alliance of transfer learning with meta learning is practiced in many pieces of research in the literature.

As already discussed in Sec. 3.2 transfer learning can be crucial in extracting data features using a pre-trained model and therefore can integrate with other learning algorithms. The article by Sun et al. (2019a, 2020) follows a similar approach for performing few-shot learning. Transfer learning is used during meta training by extracting representations of images trained on a very large dataset like MiniImageNet (Vinyals et al. (2016)). The meta transfer learning proposed in this work answers two critical questions, i.e., what to transfer and how to transfer. The Deep Neural Network (DNN) parameters trained on the large-scale data answer what to transfer, whereas the scaling and shifting operations learned for each task introduced in this work refer to how to transfer. The meta knowledge from the training stage is shared in the testing stage on the Fewshot-CIFAR 100 (Oreshkin et al. (2018)) dataset to learn new tasks using fewer data instances. They also propose a hard task meta batch strategy in which, rather than randomly picking meta training tasks, the algorithm resamples the hard tasks based on the past validation accuracy and failure. The use of pre-trained DNN was proved to be very useful for tailoring the learning experience for unseen tasks.

Likewise, Soh et al. (2020) also use transfer learning together with the optimization based meta learning method MAML (Finn et al. (2017)) for zero shot super-resolution of images. For faster adaptation and better generalization on new tasks, meta learning helps learn effective initial parameters during training. At the time of meta testing, it takes only a few gradient steps to learn the image specific information, even in case of external (or new) data instance. The learning strategy introduced in this work learns initialization parameters with reference to different blur conditions, making it possible to adapt to new (unseen) blur kernels quickly. It uses a large-scale dataset, ImageNet (Deng et al. (2009)), for transfer learning and meta training; during meta testing, low resolution images are used to train a model with a corresponding blur kernel. Therefore, the model can be trained for multiple types of blur kernels at the testing stage.

A recent article by Willard et al. (2021) adopts meta transfer learning for predicting the dynamics of water temperature of un-monitored lakes. Since the data for the un-monitored lakes is insufficient and all the deep learning models need a lot of data to learn, this work utilizes the adequately available data of the monitored lakes to learn the models and transfer knowledge to the less resourceful domain. It exploits the monitored lakes' data by extracting important characteristics and using two source models, i.e., process-based and process-guided deep learning, on each monitored lake and evaluating the models' performance, thereby applying transfer learning for the un-monitored lakes. This meta knowledge of features and performance of the models of the monitored lakes is used to select the best model for the un-monitored lakes based on the lowest predicted error. These pre-trained models outperform on the target un-monitored lake compared to when no transfer of learning is performed.

For vehicle tracking using Unmanned Aerial Vehicle (UAV) Song et al. (2020), uses a pretrained model for vehicle tracking on ground images and employ the model to adapt to the drone view images. From the deep learning viewpoint, vehicle tracking in UAV is an under-explored research area, as these videos have significantly less labeled data. To overcome this problem of data availability, transfer learning is employed. A large ground view vehicle tracking dataset is used to train a model, which is then used by the drone-view dataset, therefore transferring features across landscapes. Meta learning is used to adaptively extract shared features between both domains (drone and ground view). Therefore, transfer learning helps to overcome the data scarcity problem, while meta learning solves the issue of domain shift.

A similar issue of difference in distributions during training and testing is solved by using transfer and meta learning for down-link beam-forming by Yuan et al. (2020). As in conventional supervised learning, it is assumed that the training and test data belong to the same distribution. If the distribution of the test data changes, the model's performance is unsatisfactory. To address the are changes in the wireless environment and for faster adaptation to new unseen data, this work proposes two models based on deep transfer learning and meta learning. The results in the article prove that the performance of the proposed algorithm is significantly better than the traditional deep learning models.

The 'transfer of meta information' (which is solely meta learning) is also termed as meta transfer learning in articles such as Nguyen et al. (2018); Bastani et al. (2021); Duan et al. (2021); Aiolli (2012). Like Nguyen et al. (2018), the meta information for facial emotion classification on one dataset is transferred to another dataset. Bastani et al. (2021) discuss transferring knowledge across experiments for dynamic pricing applications. Aiolli (2012) introduces a variant of meta learning which learns how to learn kernels from data and share the sequence of transformations to find a kernel for a new task. These are solely meta learning applications, but due to abuse of terminology, they fall under the category of meta transfer learning, when actually it refers to simply transfer of meta learnings (or knowledge).

4. Discussion - Why these learning paradigms should be used together?

As discussed in Sec. 2.5, the learning algorithms have some strengths and drawbacks, and when the union of two algorithms is employed, it helps to overcome a few weaknesses. Like the integration of multi-task and meta learning helps in good generalization for unseen tasks, enables faster learning for new tasks, and makes it possible to handle heterogeneous tasks, overall makes it possible to introduce a new task in the multi-task and also have different types of tasks in meta learning. Nevertheless, at the same time, the issue of making architectural changes for a new task in MTL still and large data requirements remains a challenge. For the combination of meta and transfer learning, faster learning with fewer data and good performance on a new task are certainly the strengths of the merger. However, the disadvantage that these algorithms only focus on improving the target tasks and are not able to have multi-modal inputs, cannot be omitted.

Similarly, the merger of transfer learning with MTL enables handling multi-modal inputs and various heterogeneous tasks, requiring less data for training and aiming to boost all the tasks' performance (because of MTL). Nevertheless, at the same time suffer from the problem of being unable to integrate an unseen task and good generalization (as often pre-trained models tend to overfit).

A reliable solution that may help beat the issues mentioned above is an ensemble of the three learning paradigms, i.e., MTL, meta learning, and transfer learning. The primary objectives of this ensemble are :

1. Good performance on a new unseen task (due to meta learning)
2. Ability to handle multi-modal inputs and heterogeneous tasks (due to MTL)
3. Require less training data and good feature representation for learning (due to transfer learning)

There are many approaches in which this ensemble may be implemented. One of them, as illustrated in Fig. 6, is when the architecture of the model is with respect to MTL, it uses transfer learning; the pre-trained models for better feature extraction for various inputs, and then the backbone network and multi-head modules are trained using a two-level optimization that is employed in meta learning. Assuming the inputs are multimodal, the pre-trained networks (checkered boxes) extract essential features from the inputs, and these are processed accordingly in the multimodal embedding layer before forwarding to the backbone network, similar to what is presented by Zhang et al. (2021a); Team et al. (2021). The backbone network refers to the generic layers of the neural network or CNN, whose input combines embeddings from various inputs. The output from the backbone network goes into the task-specific layers. The architecture of every task-specific layer can differ as required by the task. Therefore, in the meta training stage, the multi-task architecture helps to learn many tasks together, thereby improving the performance of all the tasks. When new tasks (Task-4 and Task-5) are introduced in the meta testing stage, it is easy to learn in fewer data instances. The inductive bias of the multi-task architecture assists in

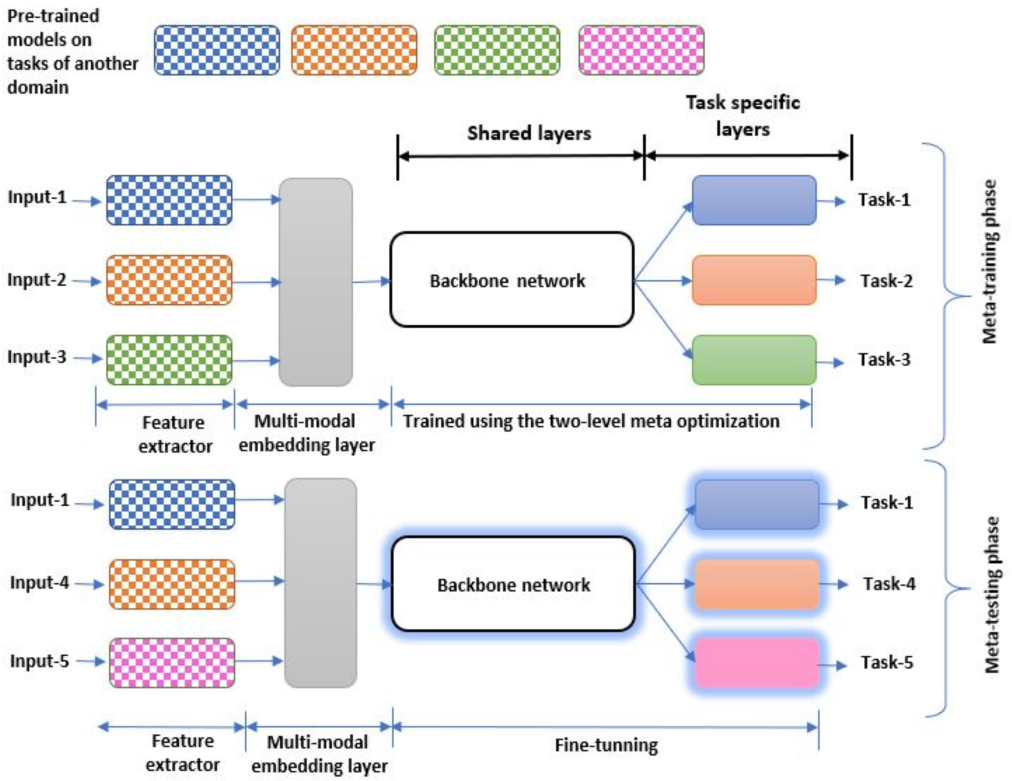


Figure 6: Proposed implementation, that ensembles MTL, meta learning and transfer learning. Here the checkered rectangle boxes represent the pretrained layers (or model), the solid rectangle boxes are the layers trained from scratch and the glowing rectangle boxes denote the layers which are fine-tuned, i.e., the parameters from the source model act as initialization for the target network.

better generalization than single task learning. Also, in this stage, it allows for both feature extraction and fine-tuning variants of transfer learning. This proposed implementation gives the liberty to add new heterogeneous tasks and allows for multi-modal inputs; transfer learning and meta learning also enable learning with fewer data samples during meta testing. MTL helps to improve the performance of both the source and the target tasks. Hence, it complies with all the three objectives mentioned above.

A variant of the implementation in Fig. 6 is shown in Fig. 7. It represents the instance when there is one input and many tasks related to it (similar to the illustration in Fig. 1). Here a multi-task source architecture can be meta trained for many different datasets and similar tasks. Later the network is used for some data of a different domain, and the meta information from the training phase will be helpful for improved learning of the target tasks. The trained network on the source data can be employed as a feature extractor i.e., case -1 in Fig. 7 or fine-tuned for the target data i.e., case-2 in Fig. 7, is the transfer learning point of view. Another way of viewing this approach is from the meta learning point of view, wherein the training of the source tasks can be considered as the meta training stage,

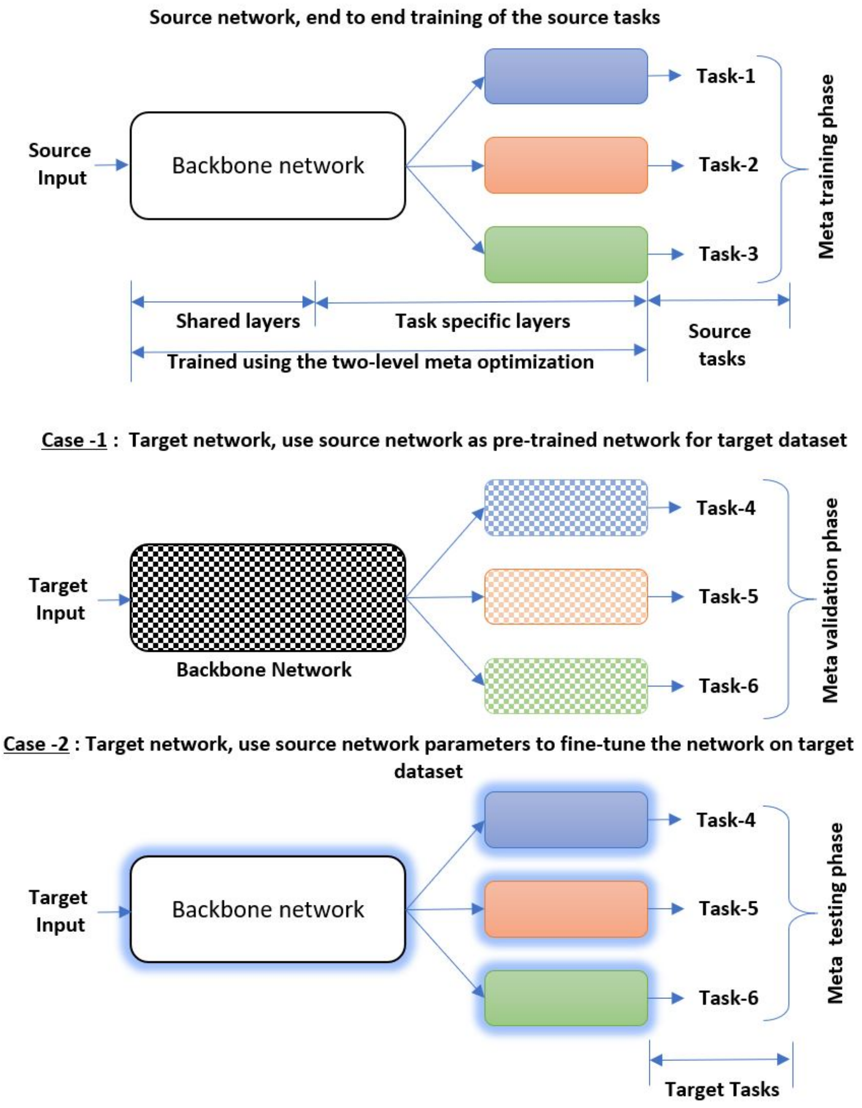

Source network, end to end training of the source tasks
Source Input
Backbone network
Task-1
Task-2
Task-3
Meta training phase
Shared layers
Task specific layers
Source tasks
Trained using the two-level meta optimization
Case -1 : Target network, use source network as pre-trained network for target dataset
Target Input
Backbone Network
Task-4
Task-5
Task-6
Meta validation phase
Case -2 : Target network, use source network parameters to fine-tune the network on target dataset
Target Input
Backbone network
Task-4
Task-5
Task-6
Meta testing phase
Target Tasks

Figure 7: A variant of the proposed implementation in Fig. 6, it represents the instance when there is a single input and multiple tasks. The case-1 and 2 can be understood from two point of views one is transfer learning and the other is meta learning, while the architecture represents MTL. Here the checkered rectangle boxes represent the pretrained layers (or model), the solid rectangle boxes are the layers trained from scratch and the glowing rectangle boxes denote the layers which are fine-tuned for unseen target data.

the case-1 can be treated as the meta validation stage because there is no training. It is only making inference on a new data, and the case-2 is nothing but the meta testing stage which uses the initial parameters from the source model and then trains the network on new target dataset.

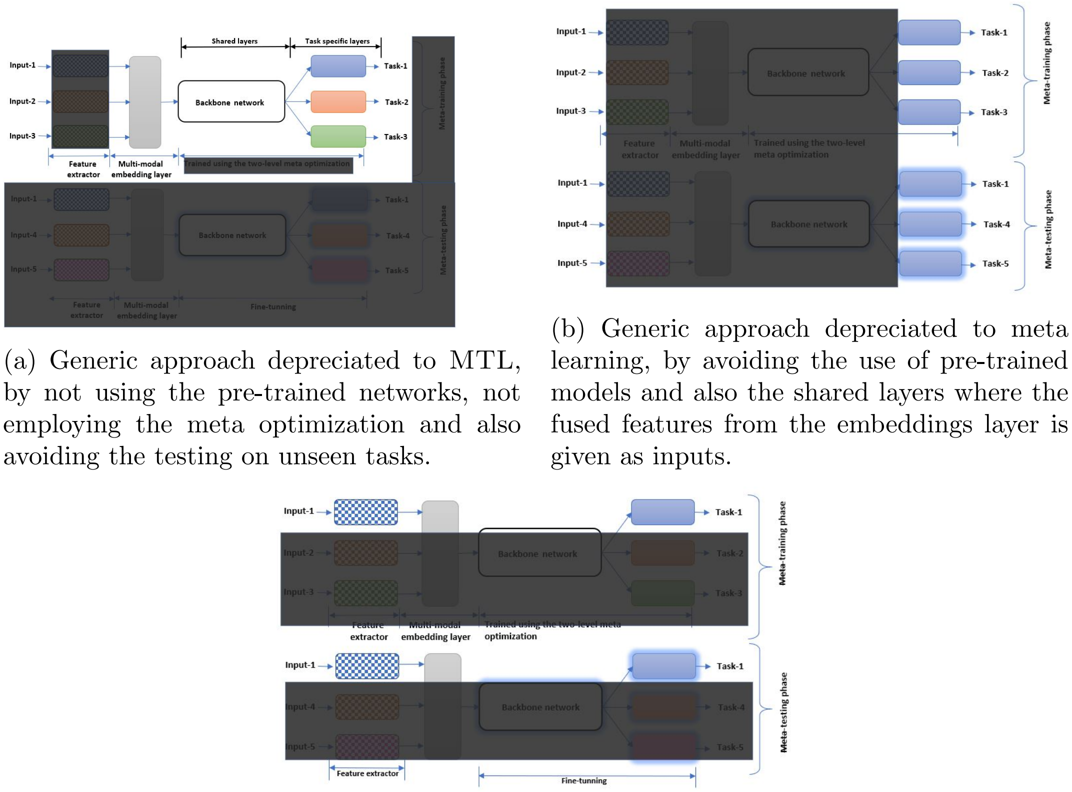
Fig. 6 is a plausible approach of how to fuse the learning paradigms. Adhering to the three primary objectives, indeed there can be several other ways of accomplishing this. Not to forget, the approach should exploit all the advantages of these learning algorithms and try to overcome their weakness of each. Also, these approaches depend on the use-case and the available dataset, i.e., the number of tasks, modality of the data, amount of data samples, etc. To the best of our knowledge, there is no such work in the literature that employs the ensemble of meta learning, transfer learning, and MTL. Indeed, the related work or future scope section of many articles mentions these three learning algorithms, but none illustrate applying them together.

5. Open questions and future work

This article discusses the fundamentals of transfer learning, MTL, and meta learning gives a comparative view on these three algorithms, details the literature study when pairs of these algorithms are employed together, and also proposes a few approaches of how all the three learning paradigms can be jointly used to overcome the drawbacks of each other. Since the proposed approaches are a combination of transfer learning, meta learning, and MTL, along with the liberty to employ multi-modal inputs and heterogeneous tasks, this study introduces the ensemble as *Multi-modal Multi-task Meta Transfer Learning (*3MTL*)*. These suggested techniques in Sec. 4 may overcome most of the drawbacks faced by the learning algorithms when used in isolation. To investigate this, this survey article enlists a few open research questions. These are:

1. How is the proposed 3MTL approach better than single task learning?
2. In 3MTL, is there a possibility for modular learning? In particular, for an unseen task, can the network automatically choose, based on the meta knowledge, which part of the network should be trained rather than training the whole network and how to exploit the shared structure (using transfer learning) ?
3. Do similar tasks automatically resort to similar sub-architecture training?
4. How can meta learning and transfer learning help to automatically alter the network architecture for a new task?
5. In case of multi-modal inputs, analyze the contribution of each modality for the outcome.

This work therefore proposes a novel generic approach to implement 3MTL in Fig. 6, keeping in mind the three objectives required for the ensemble. It is possible to reduce the global approach to any learning paradigm, as shown in Fig. 8. For example, as illustrated in Fig. 8a, only MTL can be achieved by not introducing the new tasks, by avoiding the use of pre-trained models and also by not employing the bi-level meta optimization scheme, i.e., when $\theta = \phi$ in the meta training phase Sec. 2.4, narrows down to only one optimization objective and multiple tasks for training. Fig. 8b demonstrates how the generic implementation can be reduces to meta learning, by adding no pre-trained models, no embedding layer to fuse the embeddings and also no shared layers, as every isolated input is trained for a task using two level optimization in the training phase and in the meta testing phase new tasks can be introduced. On the same conditions, if even the optimization is cut-



(a) Generic approach depreciated to MTL, by not using the pre-trained networks, not employing the meta optimization and also avoiding the testing on unseen tasks.

(b) Generic approach depreciated to meta learning, by avoiding the use of pre-trained models and also the shared layers where the fused features from the embeddings layer is given as inputs.

(c) Generic approach depreciated to transfer learning, by not using the shared backbone network and the meta optimization scheme, along with using the pre-trained models and fine-tuned layers as required for various tasks.

Figure 8: An illustration of reducing the generic structure of Fig. 6 to individual learning paradigms. The black(hidden) area represent that the blocks underneath are disabled.

down to single level optimization, along with giving the ability to use the pre-trained and fine tune networks (howsoever required), the approach is now shortened to transfer learning as in Fig. 8c. It can be considered as a global learning structure, which gives the flexibility to choose between the learning algorithms and also combinations of them.

Since the global network introduced in this work makes it possible to choose the elements required to learn the task, it will undoubtedly be worthwhile to meta-train the network to learn to evaluate which elements will do justice to a task, rather than employing all the learning algorithms. Because there is a possibility that jointly learning is not possible, and in such a scenario, the idea is to learn to activate only parts of the network, thereby foreseeing a formulation of the meta-meta learning algorithm. In consequence, as future research, it will be interesting to explore how these learning paradigms together share to learn and learn to share, along with knowing when to share and thereby making it possible to develop more human-like learning techniques.

References

Fabio Aiolli. Transfer learning by kernel meta-learning. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 81–95, Bellevue, Washington, USA, 02 Jul 2012. PMLR. URL `https://proceedings.mlr.press/v27/aiolli12a.html`.

Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-only deep representation learning. *bioRxiv*, 2019. doi: 10.1101/589333. URL `https://www.biorxiv.org/content/early/2019/03/26/589333`.

Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 77–82, 2007. doi: 10.1109/ICDMW.2007.109.

Enver Aydin and Seniha Esen Yüksel Erdem. Transfer and multitask learning using convolutional neural networks for buried wire detection from ground penetrating radar data. In Steven S. Bishop and Jason C. Isaacs, editors, *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV*, volume 11012, pages 259 – 270. International Society for Optics and Photonics, SPIE, 2019. URL `https://doi.org/10.1117/12.2518875`.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. Self-supervised meta-learning for few-shot natural language classification tasks, 2020.

Hamsa Bastani, David Simchi-Levi, and Ruihao Zhu. Meta dynamic pricing: Transfer learning across experiments, 2021.

John Bronskill, Jonathan Gordon, James Requeima, Sebastian Nowozin, and Richard Turner. TaskNorm: Rethinking batch normalization for meta-learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1153–1164. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/bronskill20a.html`.

Ruichu Cai, Kaibin Guo, Boyan Xu, Xiaoyan Yang, and Zhenjie Zhang. Meta multi-task learning for speech emotion recognition. In *INTERSPEECH*, pages 3336–3340, 2020.

Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ICML'93, page 41–48, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1558603077.

Rich Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997. ISSN 0885-6125. doi: 10.1023/A:1007379606734. URL `https://doi.org/10.1023/A:1007379606734`.

Junkun Chen, Xipeng Qiu, Pengfei Liu, and Xuanjing Huang. Meta multi-task learning for sequence modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018a.

Xiwen Chen and Kenny Q. Zhu. St^2: Small-data text style transfer via multi-task meta-learning, 2020.

Zhiyuan Chen, Bing Liu, Ronald Brachman, Peter Stone, and Francesca Rossi. *Lifelong Machine Learning*. Morgan & Claypool Publishers, 2nd edition, 2018b. ISBN 1681733021.

Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020a.

Michael Crawshaw. Multi-task learning with deep neural networks: A survey, 2020b.

Jan Christian Blaise Cruz, Julianne Agatha Tan, and Charibeth Cheng. Localization of fake news detection via multitask transfer learning. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2596–2604, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://www.aclweb.org/anthology/2020.lrec-1.316`.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. In *Proceedings of the 25th international conference on Machine learning*, pages 200–207, 2008.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Ngan Thi Dong and Megha Khosla. A multitask transfer learning framework for novel virus-human protein interactions. *bioRxiv*, 2021. doi: 10.1101/2021.03.25.437037. URL `https://www.biorxiv.org/content/early/2021/03/26/2021.03.25.437037`.

Xishuang Dong, Shanta Chowdhury, Lijun Qian, Xiangfang Li, Yi Guan, Jinfeng Yang, and Qiubin Yu. Deep learning for named entity recognition on chinese electronic medical records: Combining deep transfer learning with multitask bi-directional lstm rnn. *PLOS ONE*, 14(5):1–15, 05 2019. doi: 10.1371/journal.pone.0216046. URL `https://doi.org/10.1371/journal.pone.0216046`.

Changde Du, Changying Du, Lijie Huang, Haibao Wang, and Huiguang He. Structured neural decoding with multitask transfer learning of deep neural network representations. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2020. doi: 10.1109/TNNLS.2020.3028167.

Tiehang Duan, Mihir Chauhan, Mohammad Abuzar Shaikh, Jun Chu, and Sargur Srihari. Ultra efficient transfer learning with meta update for cross subject eeg classification, 2021.

Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A brief review of domain adaptation, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/finn17a.html`.

Ruth Fong, Walter Scheirer, and David Cox. Using human brain activity to guide machine learning, 2017.

Ali Ghadirzadeh, Xi Chen, Petra Poklukar, Chelsea Finn, Mårten Björkman, and Danica Kragic. Bayesian meta-learning for few-shot policy adaptation across robotic platforms, 2021.

Ahana Ghosh, Sebastian Tschiatschek, Hamed Mahdavi, and Adish Singla. Towards deployment of robust cooperative ai agents: An algorithmic framework for learning adaptive policies. In *19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pages 447–455, May 2020a. URL `http://eprints.cs.univie.ac.at/6587/`.

Ahana Ghosh, Sebastian Tschiatschek, Hamed Mahdavi, and Adish Singla. Towards deployment of robust ai agents for human-machine partnerships, 2020b.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3854–3863. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/guo20e.html`.

Md Junayed Hasan, Muhammad Sohaib, and Jong-Myon Kim. A multitask-aided transfer learning-based diagnostic framework for bearings under inconsistent working conditions. *Sensors*, 20(24), 2020. ISSN 1424-8220. doi: 10.3390/s20247205. URL `https://www.mdpi.com/1424-8220/20/24/7205`.

Michael E. Hasselmo. Avoiding Catastrophic Forgetting, jun 2017. ISSN 1879307X. URL `http://dx.doi.org/10.1016/j.tics.2017.04.001`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pages 1–1, may 2020. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3079209.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

Uday Kamath, John Liu, and James Whitaker. *Transfer Learning: Scenarios, Self-Taught Learning, and Multitask Learning*, pages 463–493. Springer International Publishing, Cham, 2019. ISBN 978-3-030-14596-5. doi: 10.1007/978-3-030-14596-5_10. URL https://doi.org/10.1007/978-3-030-14596-5_10.

Akhil Kedia and Sai Chetan Chinthakindi. Keep learning: Self-supervised meta-learning for learning from inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 63–77, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.6.

Minyoung Kim and Vladimir Pavlovic. Recursive inference for variational autoencoders, 2020.

Minyoung Kim and Vladimir Pavlovic. Reducing the amortization gap in variational autoencoders: A bayesian random function approach, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

David Krueger, Tegan Maharaj, and Jan Leike. Hidden incentives for auto-induced distributional shift, 2020a.

David Scott Krueger, Tegan Maharaj, Shane Legg, and Jan Leike. Hidden incentives for self-induced distributional shift, 2020b. URL https://openreview.net/forum?id=SJeFNlHtPS.

Jing Yang Lee, Kong Aik Lee, and Woon Seng Gan. Generating personalized dialogue via multi-task meta-learning, 2021.

Kiran Lekkala and Laurent Itti. Attentive feature reuse for multi task meta learning, 2020.

Hui Li, Xuejun Liao, and Lawrence Carin. Multi-task reinforcement learning in partially observable stochastic environments. *Journal of Machine Learning Research*, 10(40):1131–1186, 2009. URL http://jmlr.org/papers/v10/li09b.html.

Yiting Lin, Bineng Zhong, Guorong Li, Sicheng Zhao, Ziyi Chen, and Wentao Fan. Localization-aware meta tracker guided with adversarial features. *IEEE Access*, 7:99441–99450, 2019. doi: 10.1109/ACCESS.2019.2930550.

Bing Liu. Lifelong machine learning: A paradigm for continuous learning. *Front. Comput. Sci.*, 11(3):359–361, June 2017. ISSN 2095-2228. doi: 10.1007/s11704-016-6903-6. URL https://doi.org/10.1007/s11704-016-6903-6.

Luchen Liu, Zequn Liu, Haoxian Wu, Zichang Wang, Jianhao Shen, Yiping Song, and Ming Zhang. Multi-task learning via adaptation to similar tasks for mortality prediction of diverse rare diseases. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2020: 763–772, 2020.

Pengfei Liu and Xuanjing Huang. Meta-learning multi-task communication. *arXiv preprint arXiv:1810.09988*, 2018.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning, 2014.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016. URL `http://jmlr.org/papers/v17/15-242.html`.

Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, Inc., USA, 1 edition, 1997. ISBN 0070428077.

Dung Nguyen, Kien Nguyen, Sridha Sridharan, Iman Abbasnejad, David Dean, and Clinton Fookes. Meta transfer learning for facial emotion recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3543–3548, 2018. doi: 10.1109/ICPR.2018.8545411.

Nhu-Van Nguyen, Christophe Rigaud, Arnaud Revel, and Jean-Christophe Burie. Manga-mmtl: Multimodal multitask transfer learning for manga character analysis. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, *Document Analysis and Recognition – ICDAR 2021*, pages 410–425, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86331-9.

Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

Sinno Jialin Pan, Qiang Yang, Wei Fan, and Sinno Jialin Pan (ph. D. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

Ying Qu, Razieh Kaviani Baghbaderani, and Hairong Qi. Few-shot hyperspectral image classification through multitask transfer learning. In *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5, 2019. doi: 10.1109/WHISPERS.2019.8920992.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rkgMkCEtPB`.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 759–766, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273592. URL https://doi.org/10.1145/1273496.1273592.

Federico Retyk. *On Meta-Reinforcement Learning in task distributions with varying dynamics.* PhD thesis, UPC, Facultat d'Informàtica de Barcelona, Departament de Ciències de la Computació, Apr 2021. URL http://hdl.handle.net/2117/348143.

Elisa Ricci, Yan Yan, Anoop K. Rajagopal, Ramanathan Subramanian, Radu L. Vieriu, Oswald Lanz, and Nicu Sebe. Chapter 4 - exploring multitask and transfer learning algorithms for head pose estimation in dynamic multiview scenarios. In Vittorio Murino, Marco Cristani, Shishir Shah, and Silvio Savarese, editors, *Group and Crowd Behavior for Computer Vision*, pages 67–87. Academic Press, 2017. ISBN 978-0-12-809276-7. doi: https://doi.org/10.1016/B978-0-12-809276-7.00005-9. URL https://www.sciencedirect.com/science/article/pii/B9780128092767000059.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Wenfeng Song, Shuai Li, Yuting Guo, Shaoqi Li, Aimin Hao, Hong Qin, and Qinping Zhao. Meta transfer learning for adaptive vehicle tracking in uav videos. In Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve, editors, *MultiMedia Modeling*, pages 764–777, Cham, 2020. Springer International Publishing. ISBN 978-3-030-37731-1.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a.

Qianru Sun, Yaoyao Liu, Zhaozheng Chen, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning through hard tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. doi: 10.1109/TPAMI.2020.3018506.

Shiliang Sun, Liang Mao, Ziang Dong, and Lidan Wu. *Multiview Transfer Learning and Multitask Learning*, pages 85–104. Springer Singapore, Singapore, 2019b. ISBN 978-981-13-3029-2. doi: 10.1007/978-981-13-3029-2_7. URL https://doi.org/10.1007/978-981-13-3029-2_7.

Ishan Tarunesh, Sushil Khyalia, Vishwajeet Kumar, Ganesh Ramakrishnan, and Preethi Jyothi. Meta-learning for effective multi-task and multilingual modelling. *arXiv preprint arXiv:2101.10368*, 2021.

Shiva Taslimipoor, Omid Rohanian, and Le An Ha. Cross-lingual transfer learning and multitask learning for capturing multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 155–161, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5119. URL `https://aclanthology.org/W19-5119`.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents, 2021.

S. Thrun and L.Y. Pratt, editors. *Learning To Learn.* Kluwer Academic Publishers, Boston, MA, 1998.

Bing Tian, Yong Zhang, J. Wang, and Chunxiao Xing. Hierarchical inter-attention network for document classification with multi-task learning. In *IJCAI*, 2019.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf`.

Boyu Wang and Joelle Pineau. Online boosting algorithms for anytime transfer and multitask learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015. URL `https://ojs.aaai.org/index.php/AAAI/article/view/9607`.

Jared D. Willard, Jordan S. Read, Alison P. Appling, Samantha K. Oliver, Xiaowei Jia, and Vipin Kumar. Predicting water temperature dynamics of unmonitored lakes with meta-transfer learning. *Water Resources Research*, 57(7), Jun 2021. ISSN 1944-7973. doi: 10.1029/2021wr029579. URL `http://dx.doi.org/10.1029/2021WR029579`.

Dan Xu, Wanli Ouyang, Xiaogang Wang, and N. Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.

Q. Xu and Qiang Yang. A survey of transfer and multitask learning in bioinformatics. *J. Comput. Sci. Eng.*, 5:257–268, 2011.

Zhuyifan Ye, Yilong Yang, Xiaoshan Li, Dongsheng Cao, and Defang Ouyang. An integrated transfer learning and multitask learning approach for pharmacokinetic parameter prediction. *Molecular Pharmaceutics*, 16(2):533–541, Dec 2018. ISSN 1543-8392. doi: 10.1021/acs.molpharmaceut.8b00816. URL `http://dx.doi.org/10.1021/acs.molpharmaceut.8b00816`.

Yi Yuan, Gan Zheng, Kai-Kit Wong, Björn Ottersten, and Zhi-Quan Luo. Transfer learning and meta learning-based fast downlink beamforming adaptation. *IEEE Transactions on Wireless Communications*, 20(3):1742–1755, 2020.

Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

Huaiwen Zhang, Shengsheng Qian, Quan Fang, and Changsheng Xu. Multi-modal meta multi-task learning for social media rumor detection. *IEEE Transactions on Multimedia*, 2021a.

Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer, 2021b.

Wei Zhou, Yiying Li, Yongxin Yang, Huaimin Wang, and Timothy M. Hospedales. Online meta-critic learning for off-policy actor-critic methods, 2020.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning, 2020.

Yayi Zou and Xiaoqi Lu. Gradient-em bayesian meta-learning, 2020.