UE21CS343BB2

Topics in Deep Learning

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD), Centre for Data Sciences & Applied Machine Learning (CDSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack: Devang Saragoi, Teaching Assistant

PES UNIVERSITY

Unit 1: Introduction to Deep Learning

Transfer Learning

Devang Saraogi
Teaching Assistant

Introduction to Transfer Learning

Humans have an inherent ability to transfer knowledge across tasks. What we acquire as knowledge while learning about one task, we utilize in the same way to solve related tasks. The more related the tasks, the easier it is for us to transfer, or cross-utilize our knowledge.

Some simple examples would be,

Know how to ride a cycle	→	Learn how to ride a motorbike
Know how to play classic piano	→	Learn how to play jazz piano
Know math and statistics	→	Learn machine learning

In each of the above scenarios, we don't learn everything from scratch when we attempt to learn new aspects or topics. We transfer and leverage our knowledge from what we have learnt in the past!

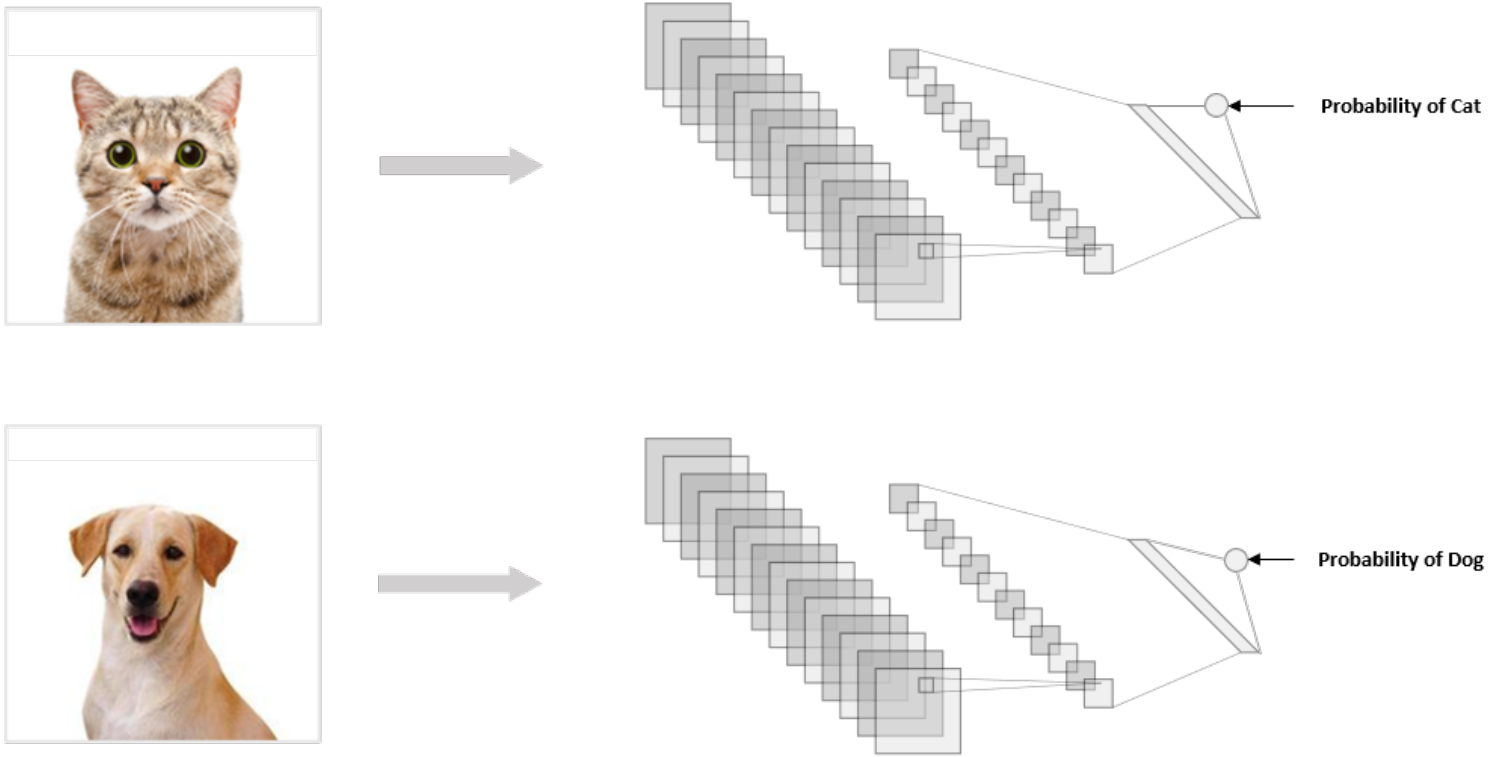
Transfer Learning

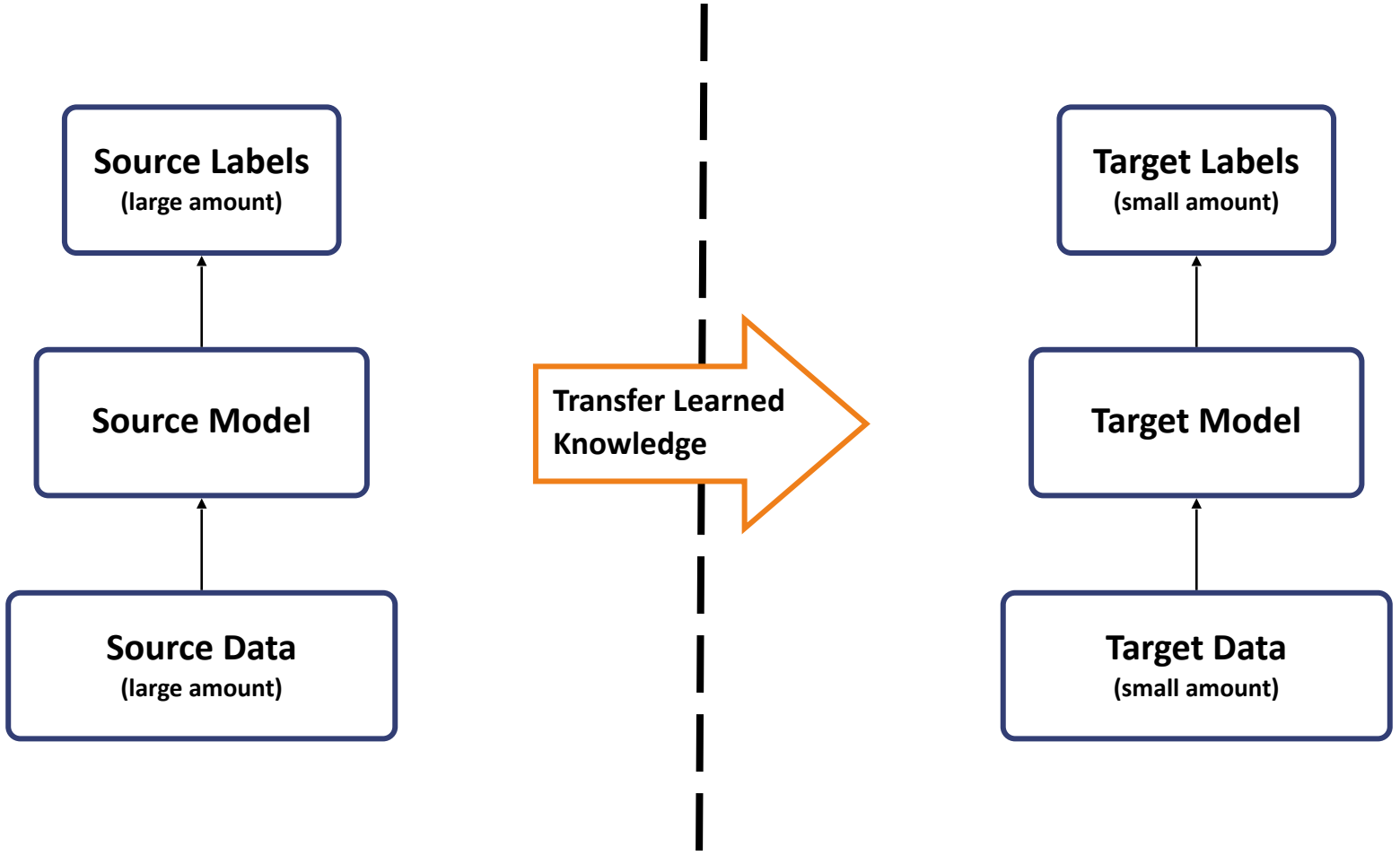
Transfer Learning is a technique where a model developed for one task is reused or repurposed for a different but related task.

- The knowledge of an already trained machine learning model is **transferred** to a different but closely linked problem throughout transfer learning.
- It can be understood as an **optimization** strategy that enables accelerated progress and enhanced performance while modelling the problem.
- Transfer learning is not exclusively an area of study for deep learning but is a popular tool, given that deep learning **demands substantial resources and data to train models**.

Transfer Learning

For example, if you trained a simple classifier to predict whether an image contains a cat, you could use the model's training knowledge to identify other animals such as dogs.

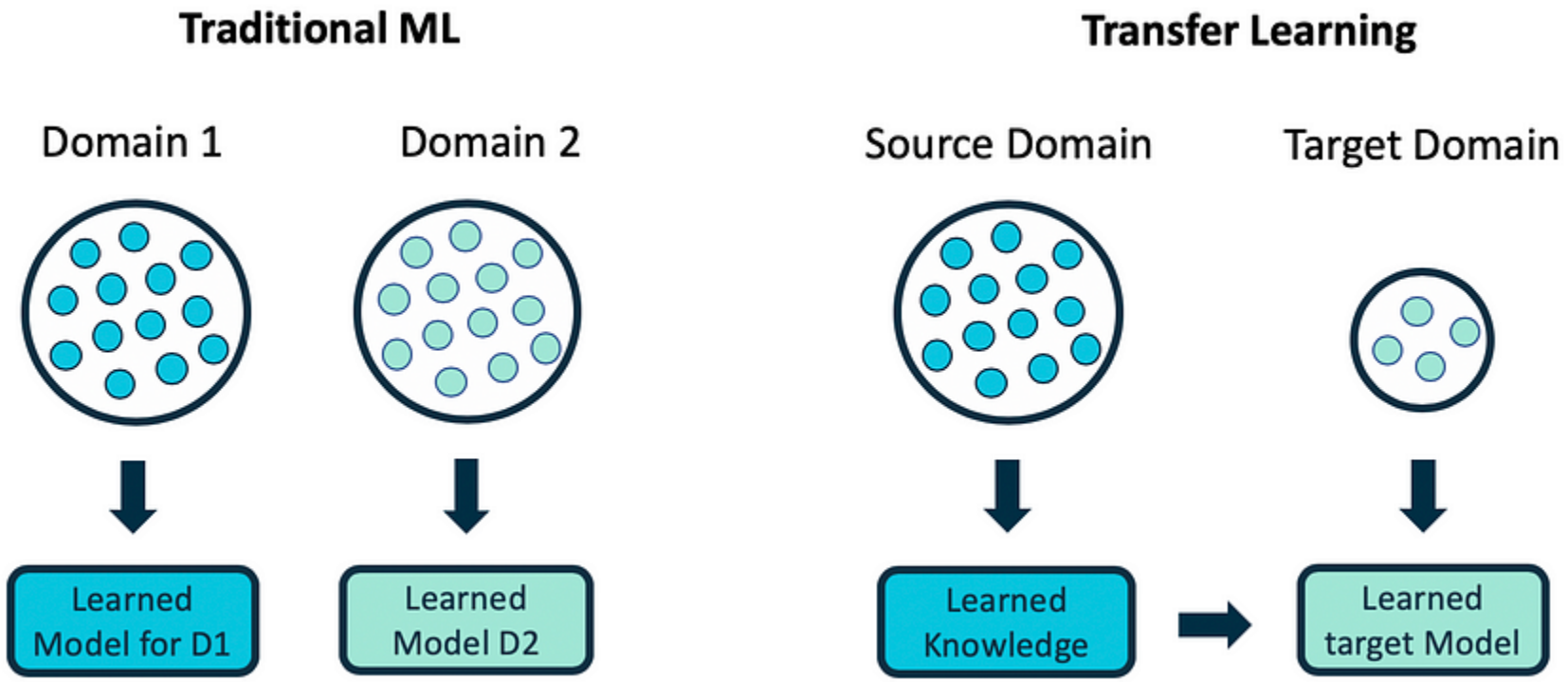



Source Labels
(large amount)
Source Model
Source Data
(large amount)
Transfer Learned
Knowledge
Target Labels
(small amount)
Target Model
Target Data
(small amount)

Traditional Machine Learning vs. Transfer Learning

Traditional Machine Learning	Transfer Learning
isolated training approach	knowledge is transferred
computationally expensive	computationally efficient
large amounts of data is required	small dataset is efficient
takes time to achieve optimal performance	achieves optimal performance faster

Traditional Machine Learning vs. Transfer Learning



Formal Definition of Transfer Learning

A **Domain** consists of two components

$$D = \{X, P(X)\}$$

where,

- feature space: X
- marginal distribution: $P(X), X = \{x_1, \ldots, x_n\}, x_i \in X$

For a given domain (D) , a **Task** is defined by two components

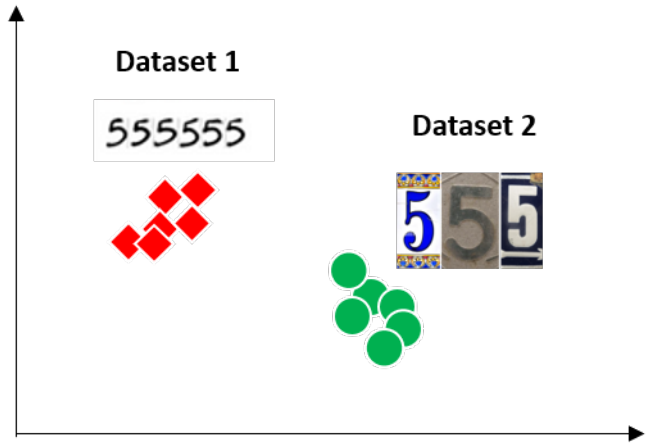
$$T = \left\{ Y, P\left(Y \middle| X\right) \right\} = \{Y, \eta\} \qquad Y = \{y_1, \ldots, y_n\}, y_i \in Y$$

where,

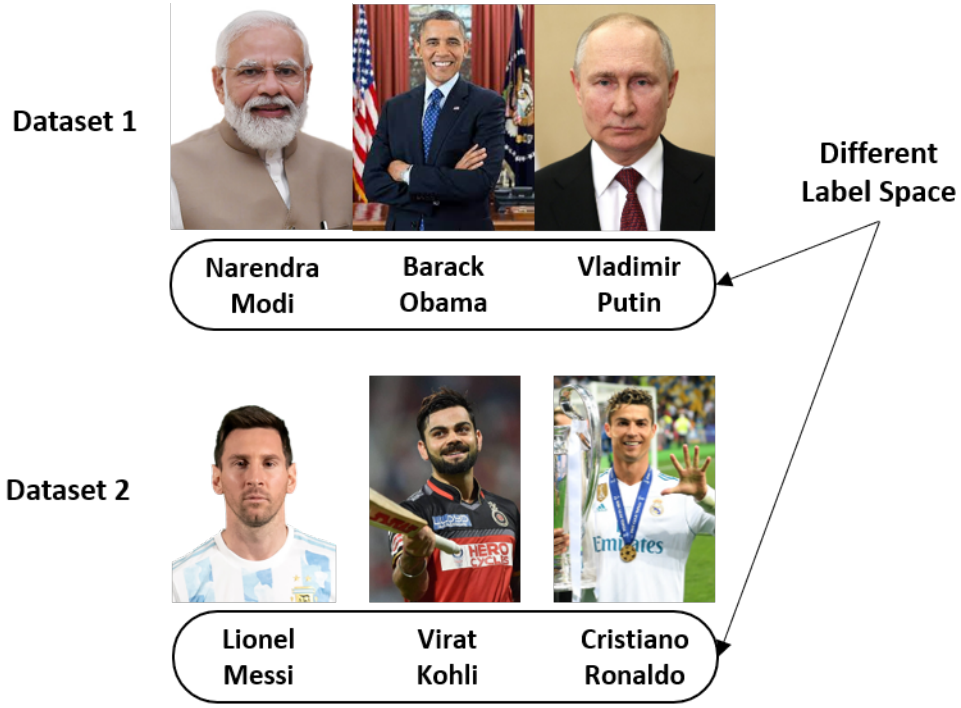
- label space: Y
- a predictive function (η), learned from features vectors/ label pairs, $(x_i , y_i), x_i \in X, y_i \in Y$
- for each vector in the domain, (η) predicts its corresponding label: $\eta(x_i) = y_i$

Domain and Task in Transfer Learning

If two domains are different, they may have different **feature spaces** or **different marginal distributions**



If two tasks are different, they may have different **label spaces** or **different conditional distributions**



Objective of Transfer Learning

Given a source domain (D_s), a corresponding source task (T_s), as well as a target domain (D_T), and a target task (T_T), the objective of transfer learning now is to enable us to learn the target conditional probability distribution $P(Y_T | X_T)$ in D_T with the information gained from D_s and T_s where $D_s \neq D_T$ or $T_s \neq T_T$.

In most cases, a limited number of labeled target examples, which is exponentially smaller than the number of labeled source examples are assumed to be available.

Objective of Transfer Learning

Given source and target domains (D_s) and (D_T) where $D = \{X, P(X)\}$ and source and target tasks (T_s) and (T_T) where $T = \left\{ Y, P\left(Y \middle| X\right) \right\}$ source and target conditions can vary in four ways, which we will illustrate in the following again using a document classification example

- **(** $X_s \neq X_T$ **)** - The feature spaces of the source and target domain are different, e.g. the documents are written in two different languages. In the context of natural language processing, this is generally referred to as cross-lingual adaptation.
- **(** $P(X_s) \neq P(X_T)$ **)** - The marginal probability distributions of source and target domain are different, e.g. the documents discuss different topics. This scenario is generally known as domain adaptation.
- **(** $Y_s \neq Y_T$ **)** - The label spaces between the two tasks are different , e.g. documents need to be assigned different labels in the target task. In practice, this scenario usually occurs with scenario 4, as it is extremely rare for two different tasks to have different label spaces, but exactly the same conditional probability distributions.
- ($P(Y_s \mid X_s) \neq P(Y_T \mid X_T)$) - The conditional probability distributions of the source and target tasks are

Transfer Learning Strategies

Different transfer learning strategies and techniques are applied based on the domain of the application, the task at hand, and the availability of data. Before deciding on the strategy of transfer learning, it is crucial to have an answer of the following questions:

- **Which** part of the knowledge can be transferred from the source to the target to improve the performance of the target task?
- **When** to transfer and when not to, so that one improves the target task performance/ results and does not degrade them?
- **How** to transfer the knowledge gained from the source model based on our current domain/task?

Traditionally, transfer learning strategies fall under three major categories depending upon the task domain and the amount of labeled/unlabeled data present.

Transfer Learning Strategies

Inductive Transfer Learning

Inductive Transfer Learning requires the source and target domains to be the same, though the specific tasks the model is working on are different.

The algorithms try to use the knowledge from the source model and apply it to improve the target task. The pre-trained model already has expertise on the features of the domain and is at a better starting point than if we were to train it from scratch.

Inductive transfer learning is further divided into two subcategories depending upon whether the source domain contains labeled data or not. These include multi-task learning and self-taught learning, respectively.

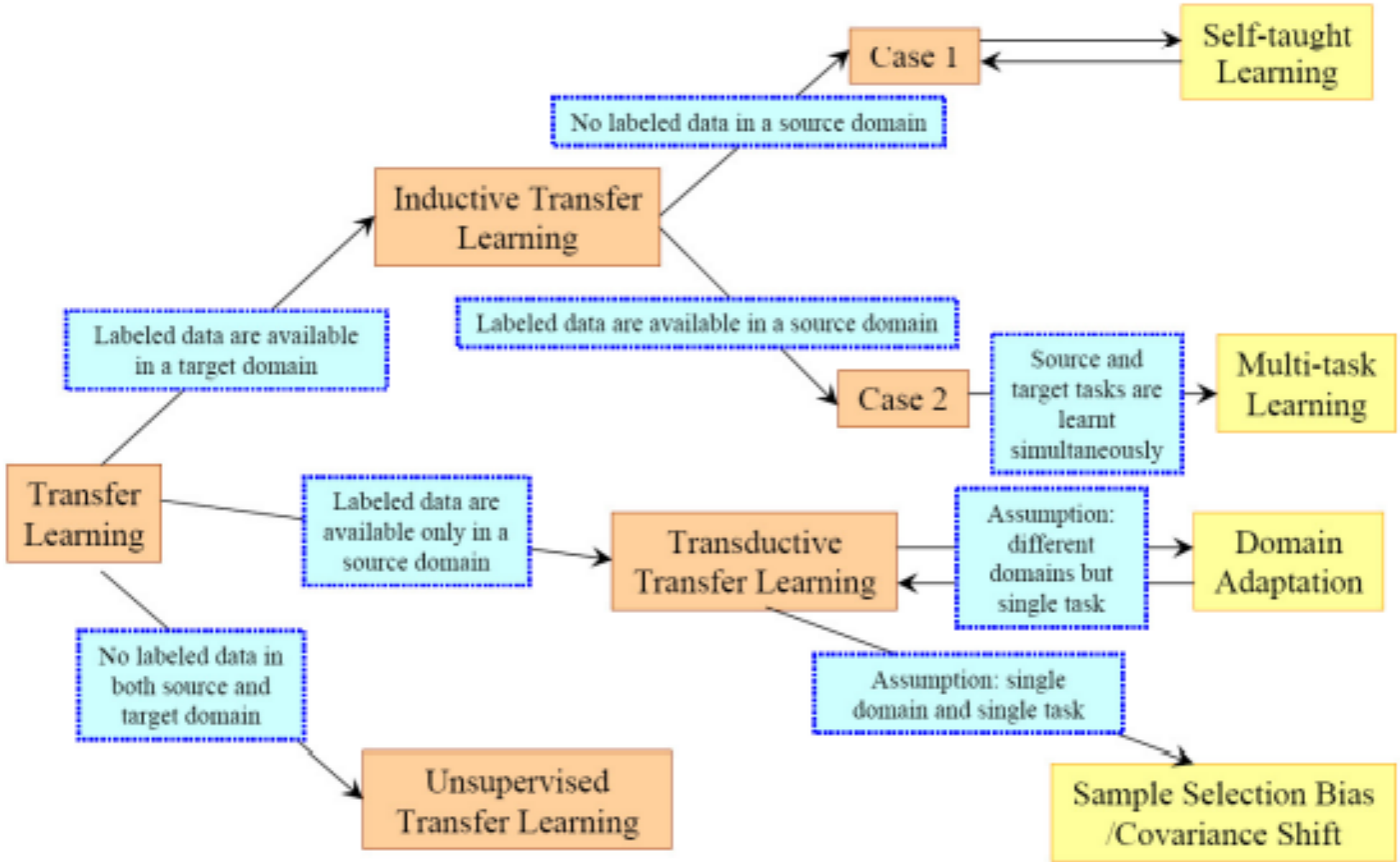
Transfer Learning Strategies

Transductive Transfer Learning

Scenarios where the domains of the source and target tasks are not exactly the same but interrelated uses the Transductive Transfer Learning strategy. One can derive similarities between the source and target tasks. These scenarios usually have a lot of labeled data in the source domain, while the target domain has only unlabeled data.

Unsupervised Transfer Learning

Unsupervised Transfer Learning is similar to Inductive Transfer learning. The only difference is that the algorithms focus on unsupervised tasks and involve unlabeled datasets both in the source and target tasks.

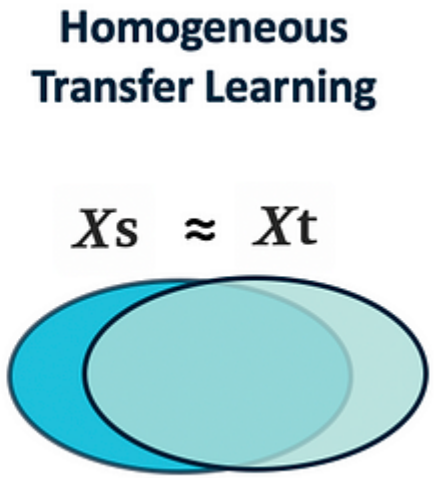

Transfer Learning
Labeled data are available in a target domain
Inductive Transfer Learning
No labeled data in a source domain
Case 1
Self-taught Learning
Labeled data are available in a source domain
Case 2
Source and target tasks are learnt simultaneously
Multi-task Learning
Labeled data are available only in a source domain
Transductive Transfer Learning
Assumption: different domains but single task
Domain Adaptation
Assumption: single domain and single task
Sample Selection Bias /Covariance Shift
No labeled data in both source and target domain
Unsupervised Transfer Learning

Transfer Learning Approaches

Transfer Learning approaches can be categorized into different approaches based on the similarity of domains, independent of the type of data samples present during training.

Homogeneous Transfer Learning

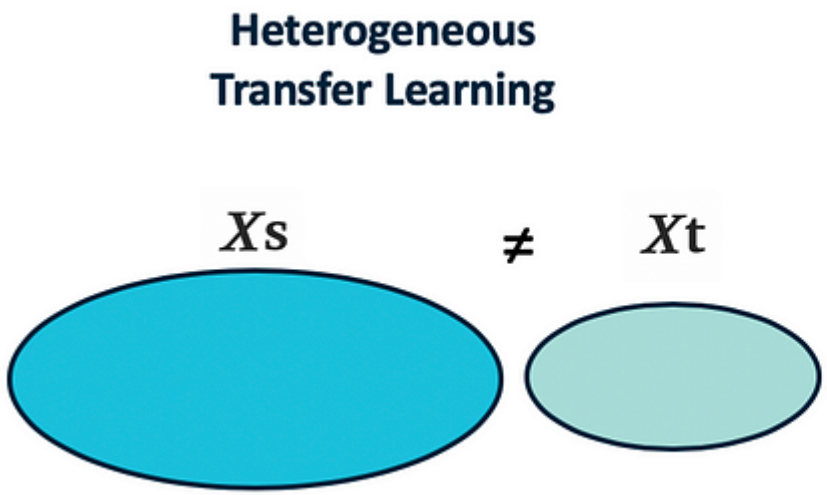
- proposed to handle situations where the domains are of the same feature space
- domains have only a slight difference in marginal distributions and are adjusted to by correcting the sample selection bias or covariate shift.



Transfer Learning Approaches

Heterogenous Transfer Learning

- usual approaches don't account the difference in the feature spaces between source and target domains
- it is challenging to collect labeled source domain data with the same feature space as the target domain
- solves the issue of source and target domains having differing feature spaces and other concerns like differing data distributions and label spaces
- applied in cross-domain tasks such as cross-language text categorization, text-to-image classification etc.



Types of Transferable Components

Instance Transfer

Instance transfer reuses knowledge acquired from the source domain to enhance the performance of the model on a target task. **Specific instances or data points from the source domain are deemed to be relevant to the target domain.**

- the central idea is to transfer information from source domain to target domain, but the direct reuse of source domain data is not always practical or advantageous
- involves the selective identification and utilization of specific instances from the source domain that align with the characteristics and requirements of the target task
- this approach acknowledges that certain instances from the source domain can be more relevant when combined with the target data to achieve improved results

Types of Transferable Components

Instance Transfer

For example, if a model is trained on urban scenes (source domain) and needs to adapt to a new city (target domain), certain instances from the source domain might include images with architectural features, street layouts, or objects commonly found in both urban environments.

Traffic light is a common element in urban settings worldwide and can be identified as a “certain instance”.

Types of Transferable Components

Feature Representation Transfer

This transfer approach acts as a bridge between domains. Abstract features are extracted from the source domain data. These **features capture general patterns and relationships** that are applicable to the target domains as well.

The approach can be further divided into two sub categories…

Asymmetric Transfer: Source features are transformed to fit the target feature space. Loss of information might happen due to domain differences.

Symmetric Transfer: A common latent feature space is identified, and both source and target features are transformed into this shared representation.

Types of Transferable Components

Feature Representation Transfer

- state of the art models trained on enormous amounts of data (eg. VGG16) becomes the feature extraction engine. This model have mastered extracting generic, robust features relevant to understanding the data.
- instead of using the entire pre-trained model, different layers of the models are utilized; earlier layers capture low level features like edges and textures while later layers encode more complex patterns
- a custom model is shaped around the borrowed layers of the pre-trained model, typically smaller and simpler with a focus learning the relationship between the extracted features and target outputs
- sometimes, fine tuning is required to align transferred functionality with the target domain; this could include adjusting weights of later layers

Types of Transferable Components

Feature Representation Transfer

For example, if a model is initially trained to recognize objects in natural images (source domain), the knowledge gained in learning features like edges, textures, or shapes can be transferred to a new task of recognizing medical images (target domain) where the underlying visual patterns may also share similarities.

Types of Transferable Components

Parameter Transfer

Parameter-based transfer learning is a technique where knowledge is transferred between source and target domain models **directly through their shared parameters (weights and biases)**. This leverages the assumption that models for related tasks often share some underlying structure or patterns captured by these parameters.

There are two main approaches to parameter sharing…

Hard Weight Sharing: The weights and biases of the pre-trained source model are directly copied for the target model.

Soft Weight Sharing: The model is expected to be close to the already learned features and is usually penalized if its weights deviate significantly from a given set of weights.

Types of Transferable Components

Relational Based Transfer

Relational-based transfer learning focuses on transferring knowledge between source and target domains by **explicitly learning and leveraging the relationships between data points**.

- this contrasts with approaches that transfer individual features, instances, or parameters
- relational methods aim to capture logical rules and connections within the data, enabling knowledge transfer even for non-IID* data with complex relationships
- for example, understanding the relationships between speech elements in a male voice can assist in analyzing sentences from other voices

**IID - Independent and Identically Distributed*

PES UNIVERSITY

Transfer Learning Strategies and Types of Transferable Components

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

PES UNIVERSITY

Unit 1: Introduction to Deep Learning

Transfer Learning for Deep Learning

Devang Saraogi
Teaching Assistant

Transfer Learning for Deep Learning

Myth

Deep Learning is not possible unless you have a million labelled examples for the task at hand.

Reality

- Useful representations can be learned from unlabelled data.
- You can train on a nearby **surrogate objective** for which it is easy to generate labels.
- You can transfer learned representations from a related task.

Transfer Learning for Deep Learning

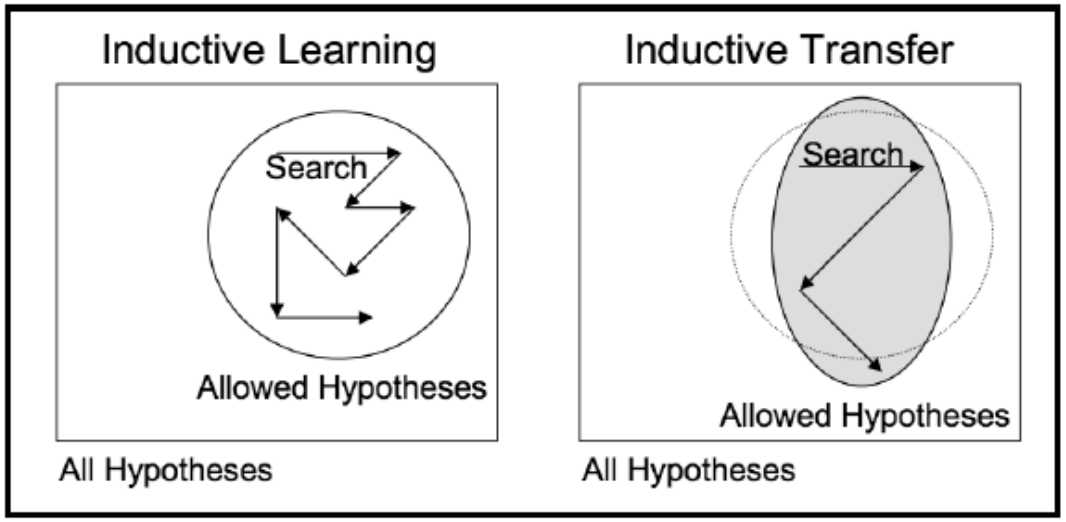
Deep learning models are representative of what is known as **inductive learning**.

- Inductive learning algorithms infer a mapping from a set of training examples
- For instance, classification models learn a mapping between input features and class labels and in order for a model to generalize well on unseen data, the model functions based on a set of assumptions which are related to the distribution of the training data
- These set of assumptions are known as inductive bias and this bias influences what is learned by the model on the given task and domain.

Transfer Learning for Deep Learning

Inductive transfer techniques utilize the inductive biases of the source task to assist the target task.

This can be done in different ways, such as by adjusting the inductive bias of the target task by limiting the model space, narrowing down the hypothesis space, or making adjustments to the search process itself with the help of knowledge from the source task.



PES UNIVERSITY

Deep Transfer Learning

Deep learning has made considerable progress in recent years. Pre-trained models form the basis of transfer learning in the context of deep learning (deep transfer learning). Let's look at the two most popular strategies for deep transfer learning.

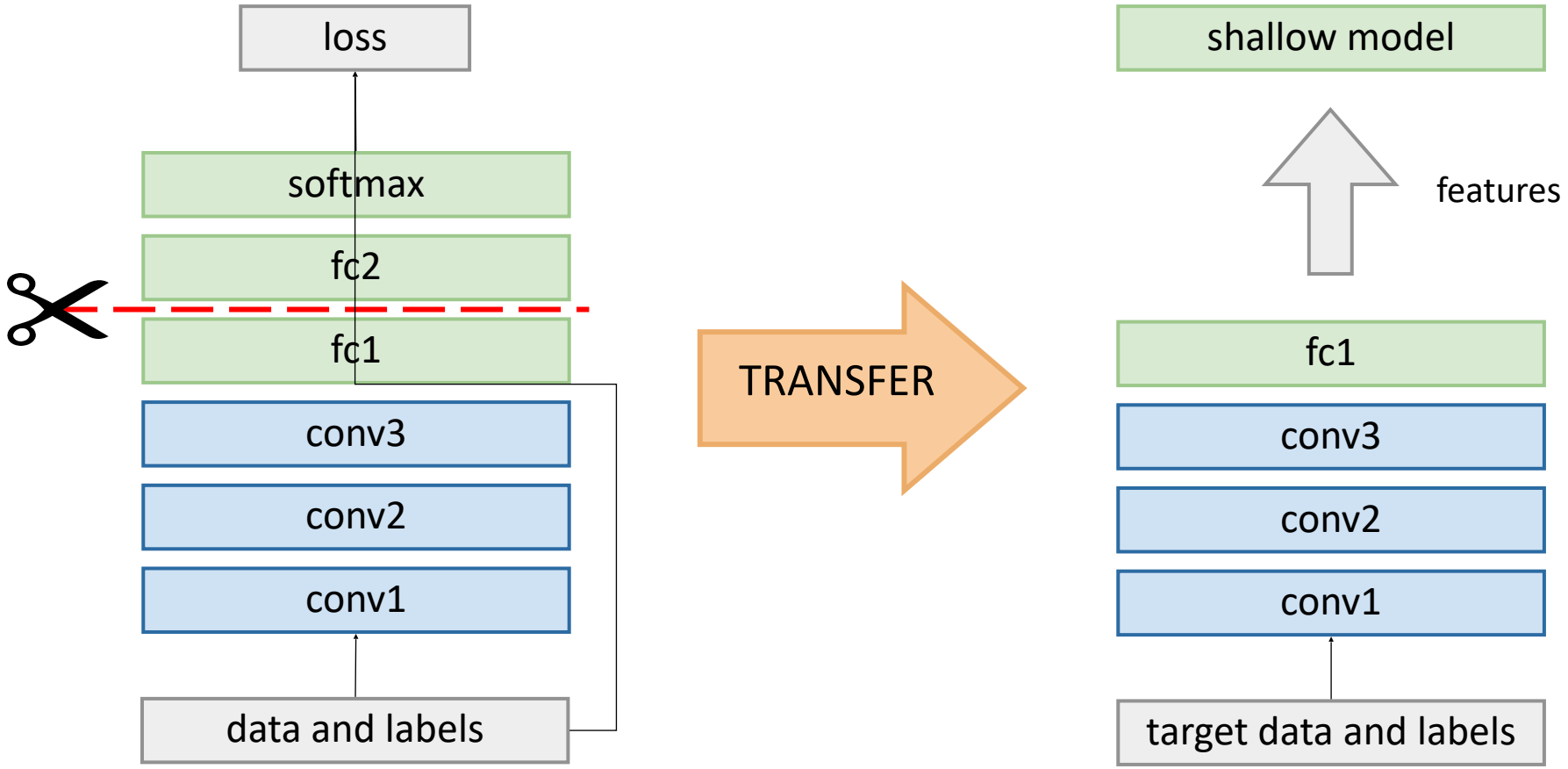
- **Off-the-shelf Pre-trained Models as Feature Extractors**
- **Fine Tuning Off-the-shelf Pre-trained Models**

Deep Transfer Learning

Off-the-shelf Pre-trained Models as Feature Extractors

Deep learning systems and models are layered architectures that learn different features at different layers (hierarchical representations of layered features). These layers are then finally connected to a last layer (usually a fully connected layer, in the case of supervised learning) to get the final output.

This layered architecture allows us to utilize a pre-trained network (such as Inception v3 or VGG) without its final layer as a fixed feature extractor for other tasks.



The key idea here is to just leverage the pre-trained model's weighted layers to extract features but **not to update** the weights of the model's layers during training with new data for the new task.

Deep Transfer Learning

Fine Tuning Off-the-shelf Pre-trained Models

This technique requires more human intervention and it **goes beyond replacing the final layer**. In this technique, some of the previous layers are **selectively retrained**.

- neural network are highly configurable architectures with a number of hyperparameters
- initial layers capture more generic features and as we progress towards the final layer, the focus shifts towards the specific task at hand
- we can **freeze (fix weights)** certain layers while retraining or **fine tune** the rest of the layers based on the requirement

Question: Is freezing layers and using the model as a feature extractor enough? Or finetuning is also required?

Freeze or Fine Tuned ?

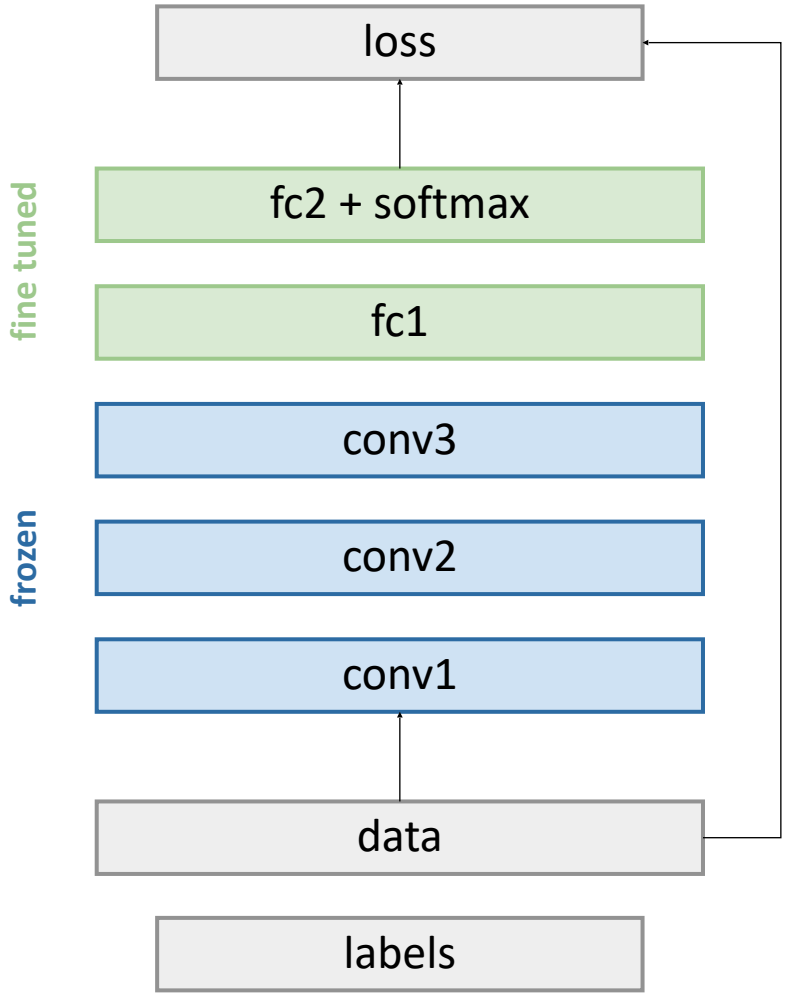
Bottom *n* layers can be either frozen or fine-tuned.

- **Freeze:** Not Updated during Backpropagation
- **Fine-Tuned:** Updated during Backpropagation

Based on Target Task

- **Freeze :** target task labels are scarce; want to avoid overfitting
- **Fine-Tuned:** Plenty of target task labels

Tip: Learning rates can be set to different for each layer to understand the tradeoff between freezing and fine tuning



Pre-trained Models

The entire concept of transfer learning is dependent on the presence of pre-trained models**, more importantly pre-trained models that perform exceptionally well on source tasks.**

Luckily, the deep learning world believes in sharing and advancing together. Today, many state-of-the-art models have been openly shared.

- pre-trained models usually cater to computer vision or natural language processing related tasks
- they are usually shared in different variants (small, medium, big) but contain millions of parameters achieved during training
- they can be either downloaded from the internet or called as an object from deep learning related Python libraries

Pre-trained Models

Pre-trained models for Computer Vision

- VGG16, VGG19
- ResNet, ResNet50
- MobileNet
- SAM (Segment Anything Model), FastSAM
- YOLO suite of models
- OpenPose, Google MediaPipe
- DETR (Detection Transformers)
- ViT (Vision Transformer)
- ByteTrack

Pre-trained models for Natural Language Processing

- word2Vec
- GLoVe (Global Vectors for Word Representation)
- BERT (Bidirectional Encoder Representations from Transformers)
- GPT (Generative Pretrained Transformer)
- ELMo (Embeddings from Language Models)
- RoBERTa (Robustly Optimized BERT)
- T5 (Text-to-Text Transfer Transformer)
- XLNet (eXtreme Language understanding Network)

Types of Deep Transfer Learning

Domain Adaptation

Domain adaption is usually referred to in scenarios where the marginal probabilities between the source and target domains are different, such as $P(X_s) \neq P(X_T)$.

- there is a shift or drift in the data distribution of the source and target domains that requires tweaks to transfer the learning
- for instance, a corpus of movie reviews labeled as positive or negative would be different from a corpus of product-review sentiments. A classifier trained on movie-review sentiment would see a different distribution if utilized to classify product reviews
- thus, domain adaptation techniques are utilized in transfer learning in these scenarios

Types of Deep Transfer Learning

Domain Confusion

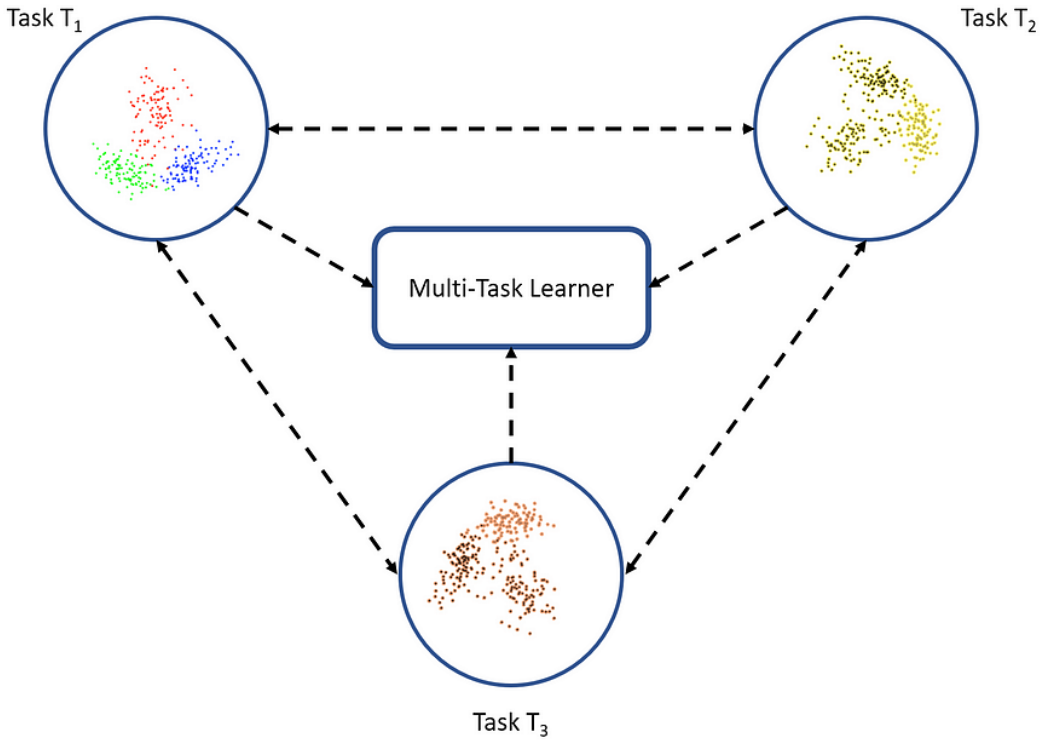
Different layers in a model capture different sets of features. This can be utilized to learn domain-invariant features and improve their transferability across domains.

- instead of allowing the model to learn any representation, we nudge the representations of both domains to be as similar as possible
- this can be achieved by applying certain pre-processing steps directly to the representations themselves
- **the basic idea behind this technique is to add another objective to the source model to encourage similarity by confusing the domain itself, hence domain confusion.**

Types of Deep Transfer Learning

Multitask Learning

Multitask learning is a slightly different flavor of the transfer learning world. In the case of multitask learning, **several tasks are learned simultaneously without distinction between the source and targets**. In this case, the learner receives information about multiple tasks at once, as compared to transfer learning, where the learner initially has no idea about the target task.



Types of Deep Transfer Learning

One-shot Learning

Deep learning systems are data-hungry by nature, such that they need many training examples to learn the weights. This is one of the limiting aspects of deep neural networks, though such is not the case with human learning.

For instance, once a child is shown what an apple looks like, they can easily identify a different variety of apple (with one or a few training examples); this is not the case with ML and deep learning algorithms.

One-shot learning is a variant of transfer learning, **where the required output is tried and inferred based on just one or a few training examples.** This is essentially helpful in real-world scenarios where it is not possible to have labeled data for every possible class (if it is a classification task), and in scenarios where new classes can be added often.

Types of Deep Transfer Learning

Zero-shot Learning

Zero-shot learning is another extreme variant of transfer learning, **which relies on no labeled examples to learn a task.** Zero-data learning or zero-short learning methods, make clever adjustments during the training stage itself to exploit additional information to understand unseen data.

In a book on Deep Learning, present zero-shot learning as a scenario where three variables are learned, such as the traditional input variable $(\boldsymbol{x})$, the traditional output variable $\left(\boldsymbol{y}\right)$ and the additional random variable that describes the task $(\boldsymbol{T})$. The model is thus trained to learn the conditional probability distribution of $\boldsymbol{P}\left(\boldsymbol{y}\middle|\boldsymbol{x},\ \boldsymbol{T}\right)$.

Zero-shot learning comes in handy in scenarios such as machine translation, where we may not even have labels in the target language.

PES UNIVERSITY

Recap

- Transfer learning models focus on storing knowledge gained while solving one problem and applying it to a different but related problem.
- Instead of training a neural network from scratch, many pre-trained models can serve as the starting point for training. These pre-trained models give a more reliable architecture and save time and resources.
- Transfer learning is used in scenarios where there is not enough data for training or when we want better results in a short amount of time.
- Transfer learning involves selecting a source model similar to the target domain, adapting the source model to the target model before transferring the knowledge, and training the source model to achieve the target model.
- It is common to fine-tune the higher-level layers of the model while freezing the lower levels as the basic knowledge is the same that is transferred from the source task to the target task of the same domain.
- In tasks with a small amount of data, if the source model is too similar to the target model, there might be an issue of overfitting. To prevent overfitting, it is essential to tune the learning rate, freeze some layers from the source model, or add linear classifiers while training the target model can help avoid this issue.

PES UNIVERSITY

References

- https://machinelearningmastery.com/transfer-learning-for-deep-learning/
- https://www.v7labs.com/blog/transfer-learning-guide
- https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a
- https://medium.com/georgian-impact-blog/transfer-learning-part-1-ed0c174ad6e7

UE21CS343BB2

Topics in Deep Learning

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD), Centre for Data Sciences & Applied Machine Learning (CDSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack: Devang Saragoi, Teaching Assistant