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

PES UNIVERSITY

Ack: Devan Saragoi, Teaching Assistant

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!

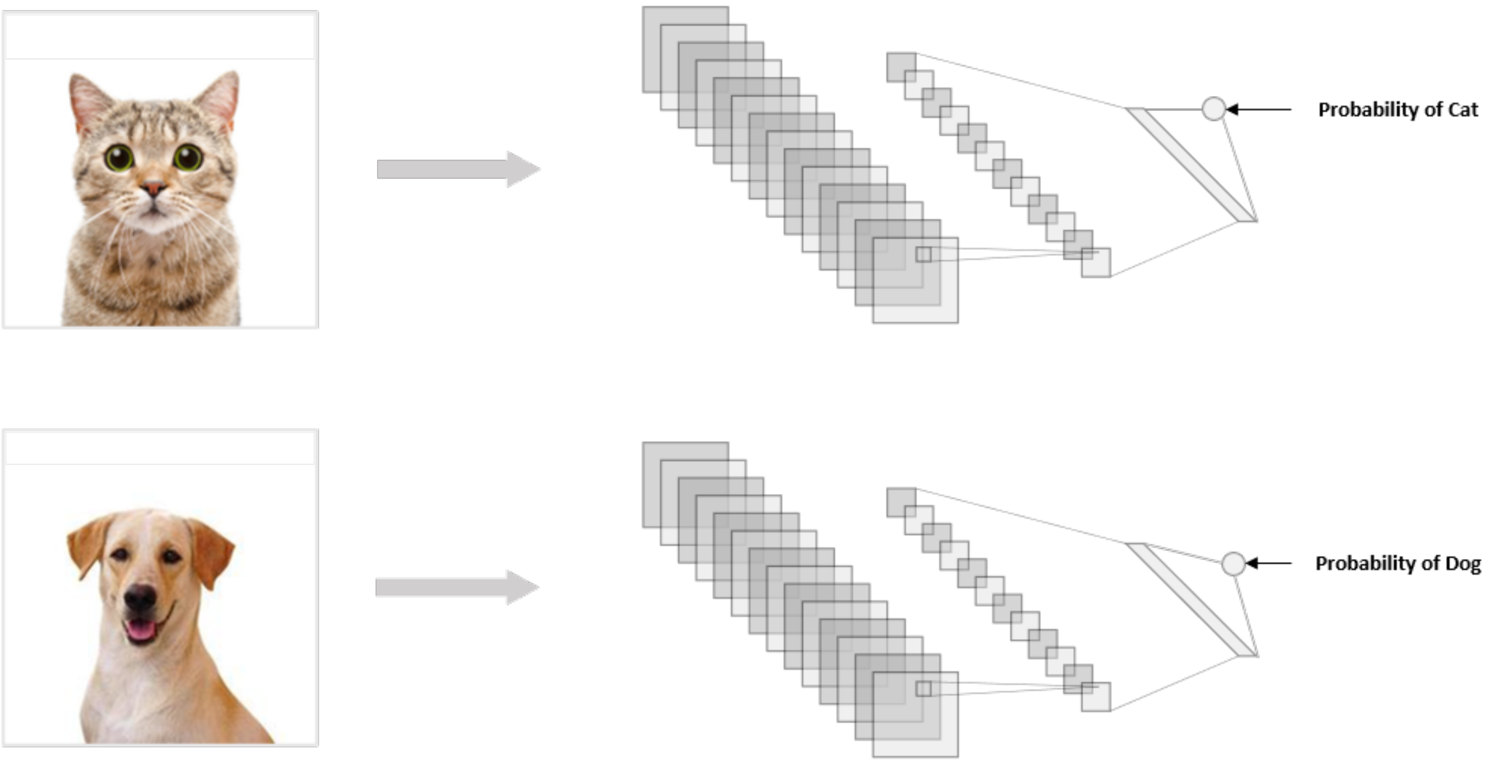
Transfer Learning

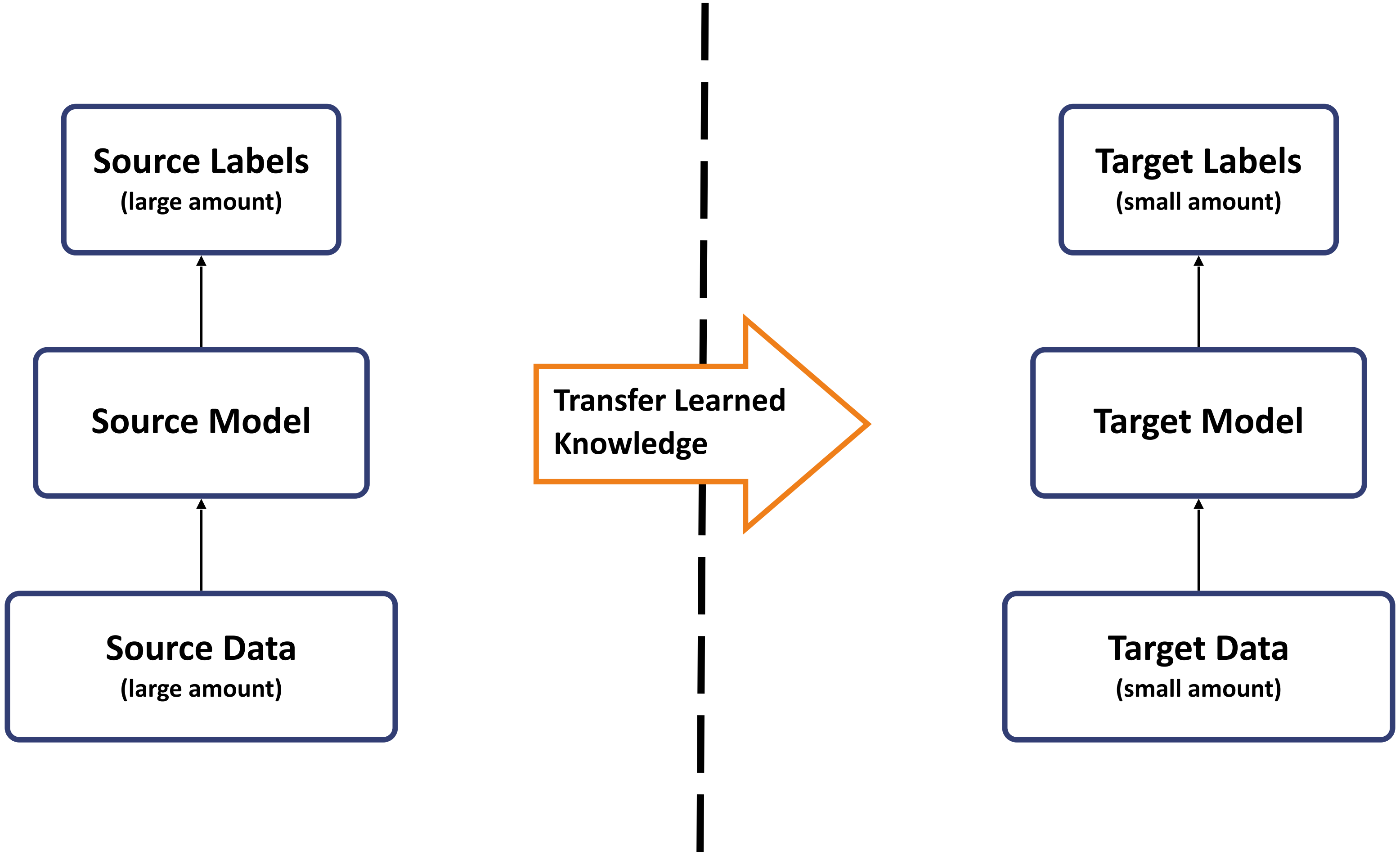
Transfer Learning is a technique where a model developed for one task is reused or repurposed for a different but related task.

- 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.

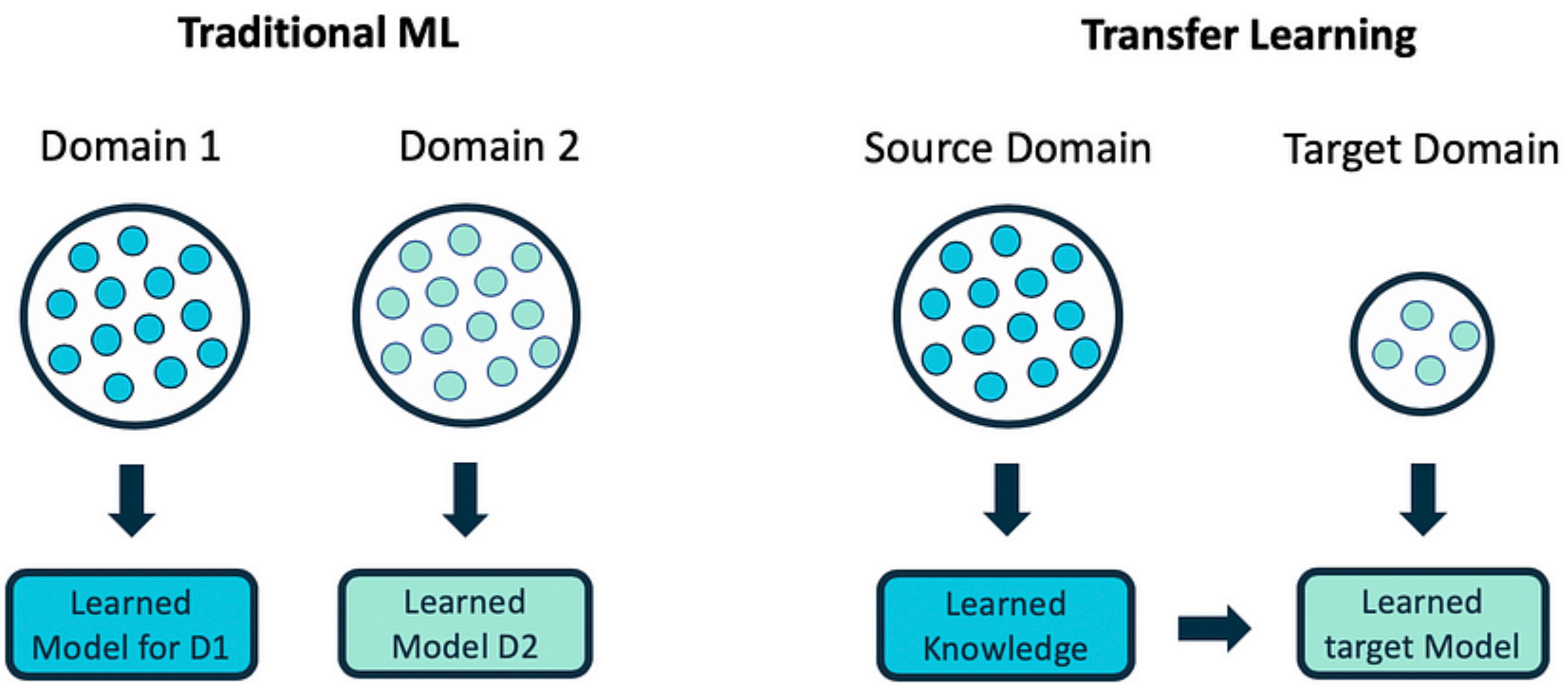



Source Labels
(large amount)
Source Model
Source Data
(large amount)
Transfer Learned Knowledge
Target Labels
(small amount)
Target Model
Target Data
(small amount)

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

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

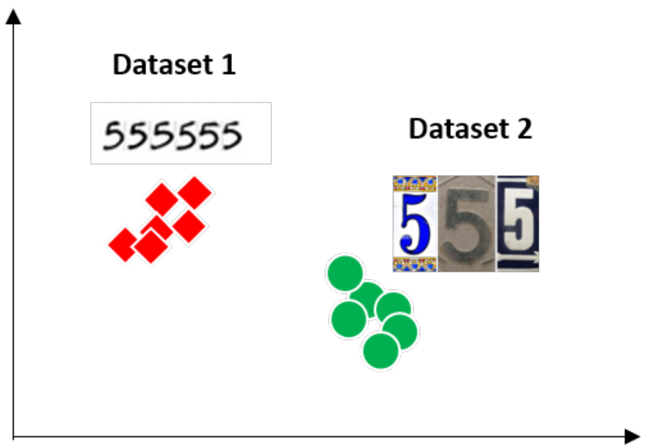
where,

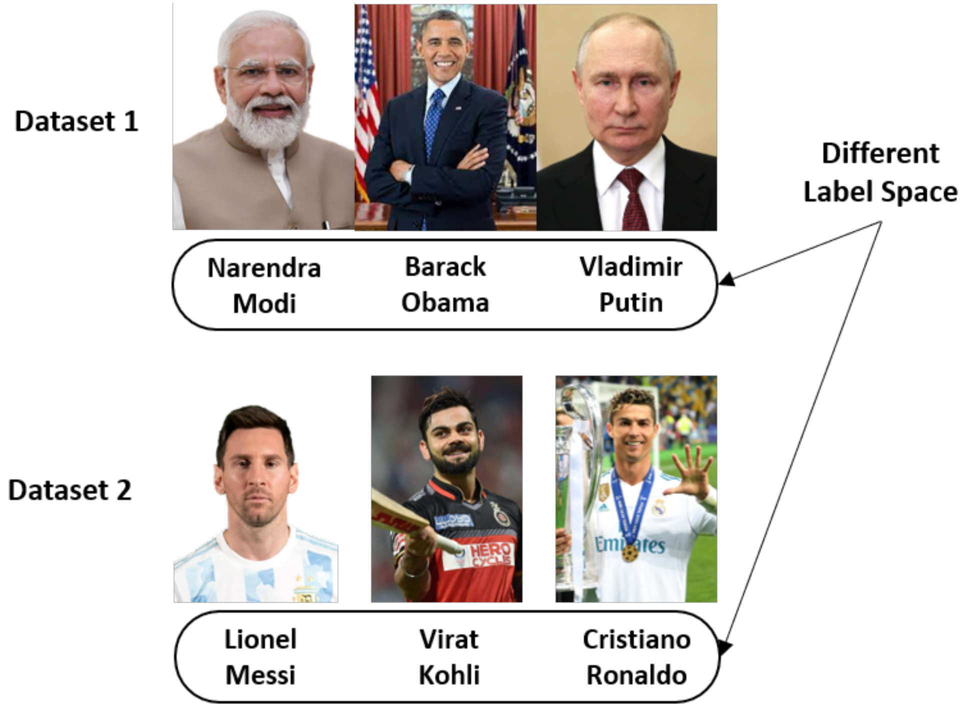
- label space: Y
- a predictive function (η), learned from features vectors/ label pairs, $(x_i , y_i), x_i \in X, y_i \in Y$
- for each vector in the domain, (η) predicts its corresponding label: $\eta(x_i) = y_i$

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**





PES UNIVERSITY

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 $P(Y_T \mid X_T)$ 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.
- **(** $P(Y_s \middle| X_s) \neq P(Y_T \middle| X_T)$ **) -** 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

PES UNIVERSITY

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.

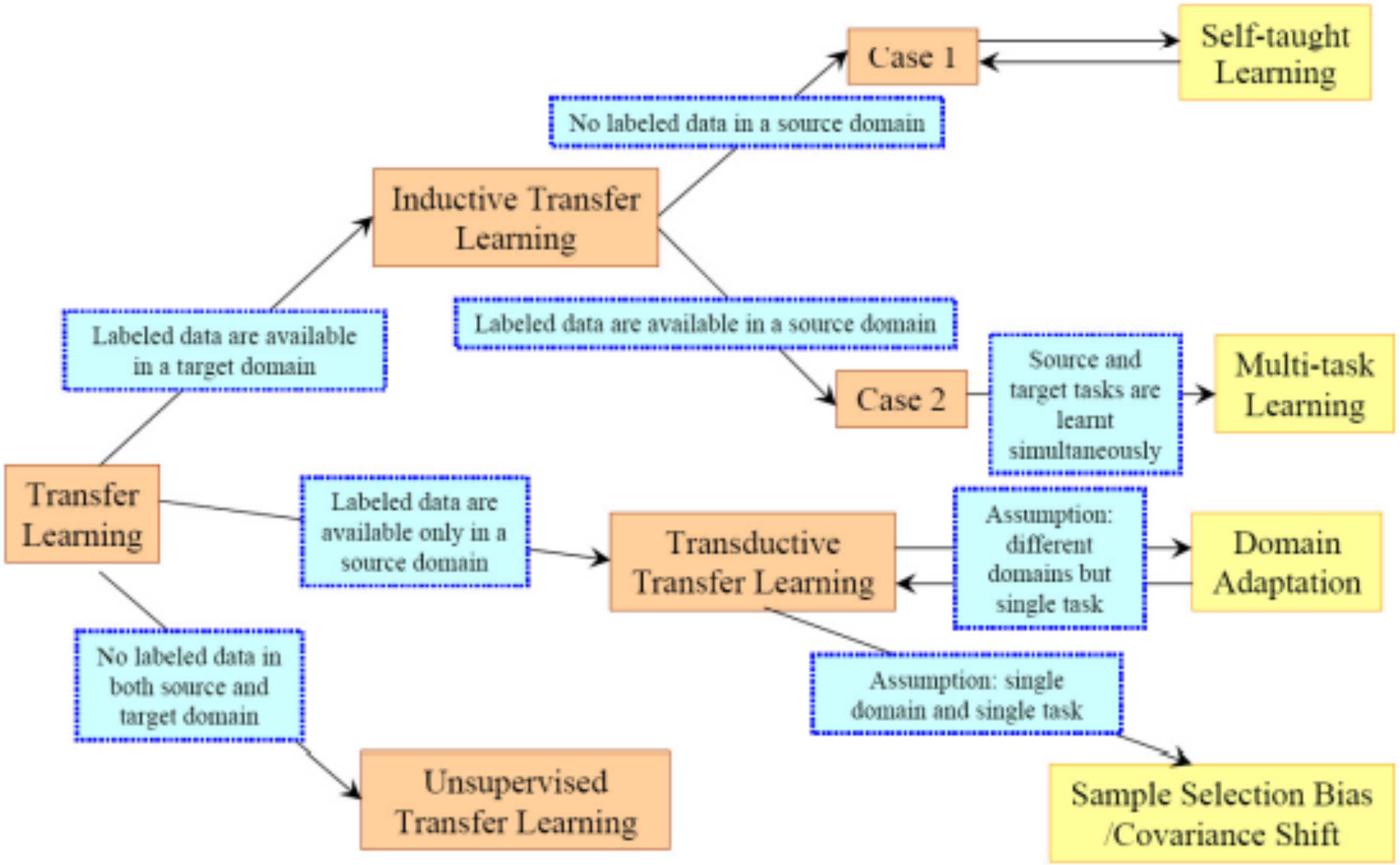
Transfer Learning Strategies

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.


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

PES UNIVERSITY

Recap

- Activation Functions are used to introduce **non-linearity** in the network.
- A neural network will almost always have the **same activation function in all hidden layers**. This activation function should be **differentiable** so that the parameters of the network are learned in backpropagation.
- While selecting an activation function, you must consider the problems it might face: vanishing and exploding gradients.
- Use Softmax or Sigmoid function for the classification problems.

Further Reading

In addition to the popular activation functions mentioned earlier, such as the Sigmoid and ReLU functions, there are many other activation functions that can be used in neural networks based on the specific problem statement and desired optimization of the learning process.

To gain a deeper understanding and familiarize yourself with different activation functions, you can refer to the following link:

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: Devan Saragoi, Teaching Assistant