**Transfer Learning**

Conventional machine learning and deep learning models, so far, have been traditionally designed to work in stand alone isolation. These models are trained to solve some tasks that are specific.The models have to be built and trained from scratch once the data distribution changes.

*Transfer learning is usually a scenario where what has been learned in one state is exploited to improve generalization in another state.*

To understand transfer learning we need to understand the stark difference between the traditional concept of making and training machine learning models, and using a methodology following transfer learning concepts.

Traditional learning is task specific and is purely isolated , the type of dataset and training performed creates isolated models . No knowledge can be transferred from one model to another.

In transfer learning, you can use the retained knowledge (features, weights etc) from beforehand or previously trained models for training of newer models and even create solutions for problems that have less data for the newer task.

Let’s try to understand the condescending explanation with the help of an illustration. Let’s just assume our task is to classify objects in images within a subset domain of a restaurant.Let ***T1*** be the mark to define the current task in its scope. For the given dataset we train the model to obtain the classification of restaurants .Traditional supervised learning ML algorithms fails when the data points or observation present in the dataset is lower or not sufficient enough for the model to learn well or generalise to a solution. Let task ***T2*** be to classify or identify the objects in a park or a cafe. Theoretically we should be able to apply the model trained for ***T1,*** in this problem scope but practically the model faces some performance degradation, which is collectively and liberally termed as the model's bias towards the training or problem domain.

Transfer Learning thus enables us to apply knowledge gained from previously learned tasks and utilize them to different but contextually related tasks. Generally if we have sufficient enough data for task ***T1*** the knowledge gained(weights, baises etc) can be transferred to task ***T2.***

A formal definition of Transfer Learning. A domain, ***D***, is defined as a two-element tuple consisting of feature space, ***x***, and marginal probability, ***P(Χ)***, where ***Χ*** is a sample data point. Thus, we can represent the domain mathematically as ***D = {x,* *P(Χ)}***

Domain contains two things D = {x, P(X)}.

Feature space: X

Marginal Distributions: P(X), X= {x1, x2,.....xn} belongs X

The function can be denoted as P(y|x) from a probabilistic point of view,

Y can be two label tuple.

For a given domain D, a task is defined by two components:

T = {Y, P(Y|X)} = {Y, n}; Y = {y1,................yn}, yi belongs y

A label space: Y

A predictive function n. Learned from feature vector/label pairs, (xi, yi)

For each feature vector in the domain, predicts its corresponding label: n(xi) = yi

Transfer learning Strategies

* **Inductive Transfer learning:** In this case, where the source and target domains are the same, yet the source and target tasks are different from one another.
* **Unsupervised Transfer Learning:** it is similar to inductive transfer learning but with the scenario of having an unsupervised domain for the task.
* **Transductive Transfer Learning:** this is the scenario where there are similarities in the source and the target but the task domains are different to one another.