Introduction to Transfer Learning

Agenda

- Problems with training CNNs
- Transfer Learning
- Transfer Learning on ImageNet

Problems with training CNNs

The Ideal CNN model

- Some of the traits of an **ideal and effective Convolutional Neural Network** are as follows:
  - Low bias
  - Low variance
  - High performance metrics (Recall / Precision / F1 Score)
- To achieve these measures of success for CNNs, we usually require:
  - **A large, diverse** and **high quality "labeled"** image dataset,
  - **High computational power** to train deep convolutional neural networks for complex images.
- However, as we will realize, it is far from straightforward to acquire both the above requirements that are needed to build robust and generalizable Convolutional Neural Networks.

Problems with training CNNs

1. **The first problem** has to do with the requirement of a **large and diverse labeled image dataset**:

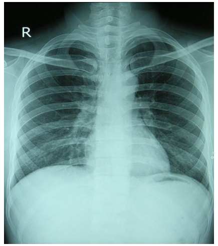
   - In many domains **getting such a large, labelled dataset can be very difficult.**

     Some examples of this:

     i. **Medical Images** (X-Rays, CT-Scans, Thermograms, Skin Patch Images)
     ii. **Agricultural Images** (Crop Monitoring, Poultry Farming, Fruit & Vegetable Counting)
     iii. **Manufacturing Images** (Defect Detection, Images of Safety and Security SOPs)

   - There are challenges with image acquisition such as patient privacy and the prohibitive cost of generating scans using medical equipment in the medical domain, or the labor-intensive, expensive task of acquiring such images in agriculture and manufacturing.

   - We also **cannot always use Data Augmentation** to sidestep the problem of image acquisition, because the **training images need to be diverse, meaningful and of high quality.**



Problems with training CNNs

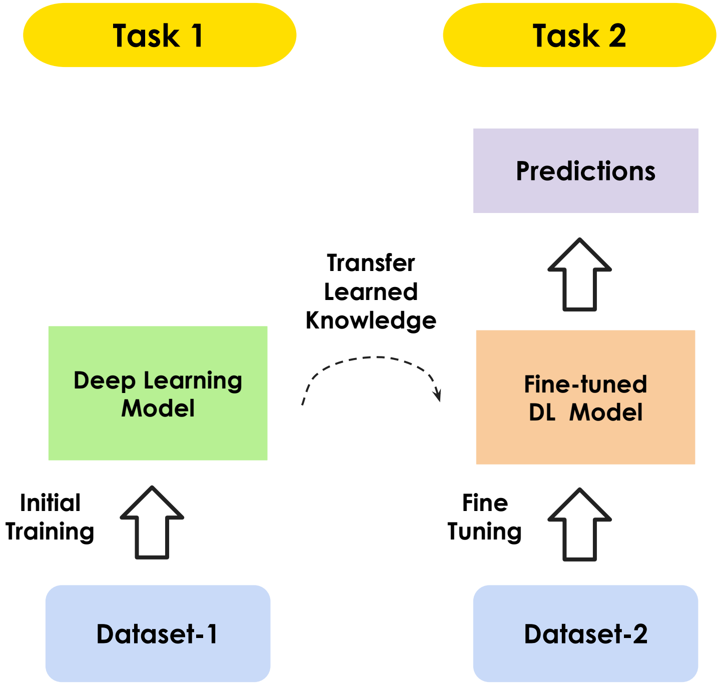
2. **The second problem** is the **computational cost of training deep CNNs** for state-of-the-art performance with minimal latency.

   Even though they are far more computationally efficient than ANNs, training complex and deep CNN models is still a fairly difficult endeavor. It requires **Graphical Processing Units (GPUs), Cloud Computing Resources** and/or other ideas from accelerated computing hardware. This can be **expensive to acquire** and may not always be feasible for individual Deep Learning practitioners or algorithmic researchers, who may need to iterate and experiment with model architectures.

The industry discovered a simple solution to both these problems with CNNs:
Transfer Learning!

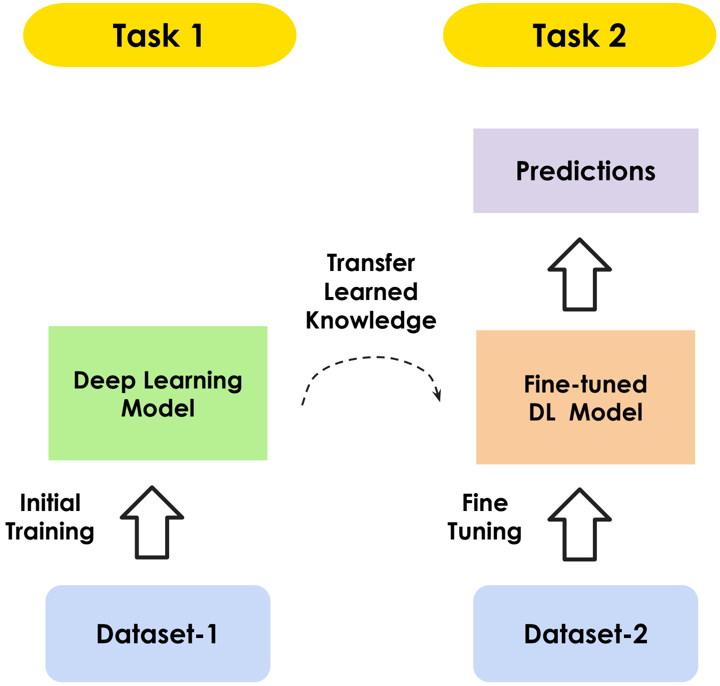
Transfer Learning

What is Transfer Learning?



- Let's say we have **two similar tasks**: **Task 1 & Task 2**
- Task 1 has a large, diverse and high-quality labeled training dataset, and can be used to train a high-performing CNN. **However, let's say we are interested in predictions for Task 2, which perhaps does not have the luxury of such a dataset, and only has a smaller, less diverse labeled dataset.**
- The idea then, is to build a high–performing CNN for Task 1 using the rich dataset available for Task 1, and then **"transfer" that knowledge** (the weights, parameters and architecture) **to Task 2**, such that we don't need to train a model for Task 2 from scratch, and we can use the lower-quality dataset for Task 2 just to fine-tune the final model for Task 2.
- **That is the reason this idea is called "Transfer Learning".**

What is Transfer Learning?



- **The model would initially be trained using the high-quality Dataset 1 to make predictions for Task 1.**
- With repeated forward and backward propagations and good hyperparameter selection, we would be able to **create an excellent model for Task 1.**
- **This model can then be "transferred" to Task 2**, and we would then **use Dataset 2 to fine-tune the predictions for Task 2.**
- The labels/classes of Task 1 & Task 2 will likely be different, so **the input and output layers of the model would need to be changed accordingly**. However **the bulk of the layers** of what could be a deep model **would be "frozen"**, and we could choose to **train and adjust the parameters of only the last few layers** in the model for Task 2.
- **Through this, we would be able to build a high-performance model for Task 2, despite not having the dataset to do so.**

Some terms related to Transfer Learning

- **Pre-trained Model:** A model that was previously trained by someone for a particular task, and is now being transferred and utilized to make predictions for the same task or another task
- **Fine-tuning**: Fine-tuning refers to importing a pre-trained model along with its weights and biases, and **continuing its training with a new dataset.**
  - Here, instead of initializing the weights and biases with random distributions, **we initialize them with the weights and biases of the pre-trained model.**
- **Transfer Learning:** Transfer Learning is when a pre-trained model is directly used with or without fine-tuning, to make predictions for another task that is similar to the task that the model was originally trained on.
  - To make such predictions, **the feature extraction layer can be "frozen" and reused**, and **only the last prediction layers may be altered** according to the labels in the new dataset.

The benefits of Transfer Learning

- At a broader level, **Transfer Learning represents a way of reusing the intelligence gained** from a high-quality, pre-existing dataset to make predictions on new tasks, which may not have datasets of the same quality as the original task the model was trained on. It represents a significant step on the path to a more generalizable form of Artificial Intelligence.

- **Transfer Learning has been a game-changer in Deep Learning and Artificial Intelligence**, because it means individual practitioners can simply "import" the latest state-of-the-art Deep Learning architectures from industry giants such as Google and Microsoft and directly utilize them for their own prediction tasks. The idea is, of course, not just restricted to Computer Vision. It has also been utilized in other areas of AI research, such as Natural Language Processing.

- **Since these models were trained on huge datasets for a long period of time, it would normally be difficult for other researchers to replicate on their own.** But through Transfer Learning this replication is far easier, and both the earlier problems of CNNs - the lack of access to large, high-quality labeled datasets, and the computational cost of training deep CNN models, can successfully be sidestepped by importing and just fine-tuning these state-of-the-art architectures. The model has already learnt good parameters from a rich dataset, and only a few layers of the architecture need to be re-trained and fine-tuned, significantly decreasing the computational cost.

Why does Transfer Learning work in CNNs?

- As we have learnt in CNNs, **the initial convolutional layers of the model extract basic features** from the images, and **as we go deeper** into the architecture, **complex features are extracted.**
- The filters trained initially for extracting the basic features for Task 1, can **directly be used for Task 2 to extract similar features**, because the understanding of basic features like edges is **common to many image processing tasks.** This is what helps us prevent the process of training our whole model once again for Task 2 and instead, **just transfer the learnings from Task 1 to Task 2.**
- So in Task 2, we only have to fine-tune the last fully-connected layers used for prediction, by utilizing the dataset we have for Task 2 to update the weights at the prediction stage. As only fine-tuning is required, and we don't need to train the model from scratch, this can be done with a small dataset and limited computational resources.

Freezing / Keeping the feature extraction part constant

Fine-tuning only the last fully-connected layers for prediction

When can we use Transfer Learning?

- Transfer Learning is beneficial in situations where we have:
  - A limited amount of labeled training data
  - A limited amount of computational power
- However, it is worth keeping in mind the initial premise of Transfer Learning - **Task 1 and Task 2 have to be similar. The level of success of Transfer Learning is correlated with the level of similarity between the two tasks.** The more dissimilar the tasks, the more work we have to do on the imported model - instead of simply fine-tuning the last few layers for example, we may also need to fine-tune some of the latter Convolutional layers or even the whole Convolutional Neural Network, with only the architecture of the network then being reused and not the weights.

For more dissimilar tasks, we may need to fine-tune the weights of the feature extraction stage in addition to those of the prediction stage.

Transfer Learning on ImageNet

ImageNet

- ImageNet is a large, open-source image dataset containing nearly 14 million labeled images, belonging to over 20,000 categories. Each category usually consists of over a hundred images.
- Some of the image classes in ImageNet comprise various animals, birds, reptiles, cars, buses, flowers, plants, and many other objects.
- Owing to the complexity of this multi-class classification problem, **ImageNet has been one of the most important open-source contributors to have advanced the state-of-the-art in Deep Learning methods for Computer Vision.** The annual ImageNet competition - **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**, has been a testing ground for organizations to evaluate and improve the performance of their state-of-the-art model architectures.

Some of the image classes in ImageNet:

Dog

Flower

Bird

Plane

Transfer Learning on ImageNet

- **Due to the large volume of diverse, high-quality labeled images on ImageNet**, it has functioned as an excellent "Task 1" for Transfer Learning models, that can then be imported and fine-tuned into other specialized tasks that may only have smaller, more limited datasets.
- Over time, many research organizations, including industry leaders such as Google and Microsoft, have trained deep, complex Convolutional Neural Networks on ImageNet. The deep models trained on this dataset have been imported and have further been used in many other image prediction tasks, by leveraging Transfer Learning.
- **These models have proven to be highly useful, and have achieved good performance metrics in many other tasks / problem domains where the availability of such large, labeled training datasets is difficult.**
- Some examples of these famous model architectures are VGG-16/19, InceptionNets and ResNets.

Thank You