What is Transfer Learning?

Traditionally, a machine learning model works well when you have enough training data and then apply it on test data that has the same features and probability distribution. When either the features of the data or the goal of your model changes, it is necessary to train a new model from scratch that is able to handle these changes. Transfer learning aims to solve this problem by transferring some of the knowledge or information gained with your first model to the new one, without having to collect or label more data.

The features of the data and their marginal probability distributions are called the domain, the labels and the objective function used to estimate their conditional probability distribution given the features are called the task. There are different categories of transfer learning depending on whether it is domain or the task that changes, or whether you are dealing with a supervised or unsupervised learning problem.

We will focus on an inductive transfer learning problem, where the target task is different to the source task and there is labeled data for both the source and target domains. Suppose we are trying to build a model to classify images of ants and bees, but we only have less than 250 of them. We first trained a convolutional neural network with 5 layers (resnet19 architecture) from scratch.

If we transfer the parameters from a model trained on the ImageNet dataset, with a different domain and task, by “freezing” all of the model’s weights on every layer but the last one, and then retrain the last layer’s weights with our 250 images, we can achieve an AUC of 0.97, improving accuracy by 33% and reducing computing time by 57%. With this approach, the first four layers act as a feature extraction process for the new images and we are only concerned about training the last layer.