07-280: Transfer Learning, Representation Learning, Pretraining, Fine Tuning

You have trained a model for task A using plenty of data that was available for task A. Suppose, for example, task A is to classify: input = face image, output $\in$ {"Messi", "Not Messi"}.

You now have to accomplish another task B. E.g., input = face image, output $\in$ {"Ronaldo", "Not Ronaldo"}. What do you do?

One option is to train a model from scratch for task B. This would require a lot of compute as well as data for task B. In many real-world settings, you may not have so much data.

Idea: Transfer learning — Modify model for task A using much lower (additional) compute and data to address task B.

We will now discuss ways of doing this.

Representation Learning

A representation of an input $x \in \mathcal{X}$ is a vector $g(x) \in \mathbb{R}^d$ produced by some function $g : \mathcal{X} \to \mathbb{R}$. E.g., g could be obtained as the first certain number of layers of a neural network. In fact, in CNNs, it has been observed that each layer represents some concepts (simpler concepts like edges in initial layers and features like eyes, nose in later layers).

Such a representation can be obtained from task A. Take the neural network learnt in task A and remove the last certain number of layers (e.g., remove the final layer).

The representation should be such that it makes the downstream task, task B, easy. For task B, we now train a function $h \colon \mathbb{R}^d \to \mathcal{Y}$. h operates on the output of g (neural net).

$$x \longrightarrow \boxed{g} \longrightarrow \boxed{h} \longrightarrow \text{output}$$

Ideally, in the space $\mathbb{R}^d$ (which is the output of g), the data of task B becomes linearly separable, and h is simply a linear classifier. The function g, which was obtained from task A, does all the heavy lifting. We obtain weights for h using much less compute (it is a much smaller neural network, often 1-layer) and data for task B.

Fine Tuning

Modify parameters of the neural network for task A via backprop using training data of task B. The learning rate is kept small to avoid forgetting what was learnt when training for task A. Sometimes weights of first few layers are kept frozen and only later layers are updated.

Pretraining

Training a model on a task with large data for the purpose of modifying it for other downstream tasks.

E.g., Task A: Given an image with some obscured pixels, predict the values of those pixels. This does not need any labeled data—the "labeled data" for this task can be obtained computationally from unlabeled images (x = image with obscured pixels, y = original image).

This task by itself does not have value. However, the representations learnt here can be very useful.

Foundation model: A pretrained model trained using a large corpus of data so that it can learn broad representations and be adapted for a wide variety of tasks.