realistic example of $\mathbf{y}$ given $\mathbf{x}$, without any need for a specific paired $\mathbf{y}$ as is traditionally needed in supervised learning. More detail on unsupervised translation for images is given in Section 25.7.5.

21.7.2 Transfer learning and multitask learning

In **transfer learning**, experience with one learning task helps an agent learn better on another task. For example, a person who has already learned to play tennis will typically find it easier to learn related sports such as racquetball and squash; a pilot who has learned to fly one type of commercial passenger airplane will very quickly learn to fly another type; a student who has already learned algebra finds it easier to learn calculus.

Transfer learning

We do not yet know the mechanisms of human transfer learning. For neural networks, learning consists of adjusting weights, so the most plausible approach for transfer learning is to copy over the weights learned for task A to a network that will be trained for task B. The weights are then updated by gradient descent in the usual way using data for task B. It may be a good idea to use a smaller learning rate in task B, depending on how similar the tasks are and how much data was used in task A.

Notice that this approach requires human expertise in selecting the tasks: for example, weights learned during algebra training may not be very useful in a network intended for racquetball. Also, the notion of copying weights requires a simple mapping between the input spaces for the two tasks and essentially identical network architectures.

One reason for the popularity of transfer learning is the availability of high-quality pretrained models. For example, you could download a pretrained visual object recognition model such as the ResNet-50 model trained on the COCO data set, thereby saving yourself weeks of work. From there you can modify the model parameters by supplying additional images and object labels for your specific task.

Suppose you want to classify types of unicycles. You have only a few hundred pictures of different unicycles, but the COCO data set has over 3,000 images in each of the categories of bicycles, motorcycles, and skateboards. This means that a model pretrained on COCO already has experience with wheels and roads and other relevant features that will be helpful in interpreting the unicycle images.

Often you will want to freeze the first few layers of the pretrained model—these layers serve as feature detectors that will be useful for your new model. Your new data set will be allowed to modify the parameters of the higher levels only; these are the layers that identify problem-specific features and do classification. However, sometimes the difference between sensors means that even the lowest-level layers need to be retrained.

As another example, for those building a natural language system, it is now common to start with a pretrained model such as the ROBERTA model (see Section 24.6), which already "knows" a great deal about the vocabulary and syntax of everyday language. The next step is to fine-tune the model in two ways. First, by giving it examples of the specialized vocabulary used in the desired domain; perhaps a medical domain (where it will learn about "myocardial infarction") or perhaps a financial domain (where it will learn about "fiduciary responsibility"). Second, by training the model on the task it is to perform. If it is to do question answering, train it on question/answer pairs.

One very important kind of transfer learning involves transfer between simulations and the real world. For example, the controller for a self-driving car can be trained on billions