**Учебный материал 1**

Машинное обучение

Machine learning is a rapidly growing field of study whose primary concern is the design and analysis of algorithms which enable computers to learn. While still a young discipline,

with much more awaiting to be discovered than is currently known, today machine learning can be used to teach computers to perform a wide array of useful tasks. This includes tasks like the automatic detection of objects in images (a crucial component of driver-assisted and self-driving cars), speech recognition (which powers voice command technology), knowledge discovery in the medical sciences (used to improve our understanding of complex diseases), and predictive analytics (leveraged for sales and economic forecasting).

To get a big picture sense of how machine learning works we begin by discussing a simple toy machine learning problem: teaching a computer how to distinguish between pictures of cats from those with dogs.

This will allow us to informally describe the procedures used to solve machine learning problems in general.

To teach a child the difference between “cat” versus “dog”, parents (almost!) never give their children some kind of formal scientific definition to distinguish the two; i.e., that a dog is a member of Canis Familiaris species from the broader class of Mammalia, and that a cat while being from the same class belongs to another species known as Felis Catus. No, instead the child is naturally presented with many images of what they are told are either “dogs” or “cats” until they fully grasp the two concepts. How do we know when a child can successfully distinguish between cats and dogs? Intuitively, when they encounter new cats and dogs, and can correctly identify each new example. Like human beings, computers can be taught how to perform this sort of task in a similar manner. This kind of task, where we aim to teach a computer to distinguish between different types of things, is referred to as a *classification* problem in machine learning.

1. **Collecting data** Like human beings, a computer must be trained to recognize the difference between these two types of animal by learning from a batch of examples, typically referred to as a *training set* of data. Figure 1.1 shows such a training set consisting of a few images of different cats and dogs. Intuitively, the larger and more diverse the training set the better a computer (or human) can perform a learning task, since exposure to a wider breadth of examples gives the learner more experience.



**2. Designing features** Think for a moment about how you yourself tell the difference between images containing cats from those containing dogs. What do you look for in order to tell the two apart? You likely use color, size, the shape of the ears or nose, and/or some combination of these *features* in order to distinguish between the two. In other words, you do not just look at an image as simply a collection of many small square pixels. You pick out details, or features, from images like these in order to identify what it is you are looking at. This is true for computers as well. In order to successfully train a computer to perform this task (and any machine learning task more generally) we need

to provide it with properly designed features or, ideally, have it find such features itself.

This is typically not a trivial task, as designing quality features can be very application dependent. For instance, a feature like “number of legs” would be unhelpful in discriminating between cats and dogs (since they both have four!), but quite helpful in telling cats and snakes apart. Moreover, extracting the features from a training dataset can also be challenging. For example, if some of our training images were blurry or taken from a perspective where we could not see the animal’s head, the features we designed might not be properly extracted.

However, for the sake of simplicity with our toy problem here, suppose we can easily extract the following two features from each image in the training set:

1. *size of nose*, relative to the size of the head (ranging from small to big);

2. *shape of ears* (ranging from round to pointy).

Examining the training images shown in Fig. 1.1, we can see that cats all have *small* noses and *pointy* ears, while dogs all have *big* noses and *round* ears. Notice that withthe current choice of features each image can now be represented by just two numbers: a number expressing the relative nose size, and another number capturing the pointiness or round-ness of ears. Therefore, we now represent each image in our training set in a 2-dimensional *feature space* where the features “nose size” and “ear shape” are the horizontal and vertical coordinate axes respectively, as illustrated in Fig. 1.2. Because our designed features distinguish cats from dogs in our training set so well the feature representations of the cat images are all clumped together in one part of the space, while those of the dog images are clumped together in a different part of the space.



**3. Training a model** Now that we have a good feature representation of our training data the final act of teaching a computer how to distinguish between cats and dogs is a simple geometric problem: have the computer find a line or *linear model* that clearly separates the cats from the dogs in our carefully designed feature space.1 Since a line (in a 2-dimensional space) has two parameters, a slope and an intercept, this means finding the right values for both. Because the parameters of this line must be determined based on the (feature representation) of the training data the process of determining proper parameters, which relies on a set of tools known as *numerical optimization,* is referred to as the training of a model.

Figure 1.3 shows a trained linear model (in black) which divides the feature space into cat and dog regions. Once this line has been determined, any future image whose feature representation lies above it (in the blue region) will be considered a cat by the computer, and likewise any representation that falls below the line (in the red region) will be considered a dog.



**4. Testing the model** To test the efficacy of our learner we now show the computer a batch of previously unseen images of cats and dogs (referred to generally as a *testing* *set* of data) and see how well it can identify the animal in each image. In Fig. 1.4 we show a sample testing set for the problem at hand, consisting of three new cat and dog images. To do this we take each new image, extract our designed features (nose size and ear shape), and simply check which side of our line the feature representation falls on. In this instance, as can be seen in Fig. 1.5 all of the new cats and all but one dog from the testing set have been identified correctly.



The misidentification of the single dog (a Boston terrier) is due completely to our choice of features, which we designed based on the training set in Fig. 1.1. This dog has been misidentified simply because its features, a small nose and pointy ears, match those of the cats from our training set. So, while it first appeared that a combination of nose size and ear shape could indeed distinguish cats from dogs, we now see that our training set was too small and not diverse enough for this choice of features to be completely effective.

To improve our learner, we must begin again. First, we should collect more data, forming a larger and more diverse training set. Then we will need to consider designing more discriminating features (perhaps eye color, tail shape, etc.) that further help distinguish cats from dogs. Finally, we must train a new model using the designed features, and test it in the same manner to see if our new trained model is an improvement over the old one.

**The pipeline of a typical machine learning problem**

Let us now briefly review the previously described process, by which a trained model was created for the toy task of differentiating cats from dogs. The same process is used to perform essentially all machine learning tasks, and therefore it is worthwhile to pause for a moment and review the steps taken in solving typical machine learning problems.

We enumerate these steps below to highlight their importance, which we refer to all together as the general pipeline for solving machine learning problems, and provide a picture that compactly summarizes the entire pipeline in Fig. 1.6.

