



Figure 6.8 The top row shows examples of natural images of size 64×64 pixels, whereas the bottom row shows randomly generated images of the same size obtained by drawing pixel values from a uniform probability distribution over the possible pixel colours.

improvement. Effectively, neural networks learn a set of basis functions that are adapted to data manifolds. Moreover, for a particular application, not all directions within the manifold may be significant. For example, if we wish to determine only the orientation, and not the position, of the object in [Figure 6.7](#), then there is only one relevant degree of freedom on the manifold and not three. Neural networks are also able to learn which directions on the manifold are relevant to predicting the desired outputs.

Another way to see that real data is confined to low-dimensional manifolds is to consider the task of generating random images. In [Figure 6.8](#) we see examples of natural images along with examples of synthetic images of the same resolution generated by sampling each of the red, green, and blue intensities at each pixel independently at random from a uniform distribution. We see that none of the synthetic images look at all like natural images. The reason is that these random images lack the very strong correlations between pixels that natural images exhibit. For example, two adjacent pixels in a natural image have a much higher probability of having the same, or very similar, colour, than would two adjacent images in the random examples. Each of the images in [Figure 6.8](#) corresponds to a point in a high-dimensional space, yet natural images cover only a tiny fraction of this space.

6.1.4 Data-dependent basis functions

We have seen that simple basis functions that are chosen independently of the problem being solved can run into significant limitations, particularly in spaces of high dimensionality. If we want to use basis functions in such situations, then one approach would be to use expert knowledge to hand-craft the basis functions in a