Chpt7 Telling birds from airplanes

Augest 22, 2021

7.2.7 The limits of going fully connected

Let's reason about what using a linear module on 1D view of our images entails. It's like taking every single input value - that is, every single component in our RGB image - and computing a linear combination of it with all the other values of every output feature. On one hand, we are allowing for the combination of any pixel with every other pixel in the image being potentially relevant for our task. On the other hand, we aren't utilizing the relative position of neighbouring or far away pixels, since we are treating the image as one big vector of numbers.

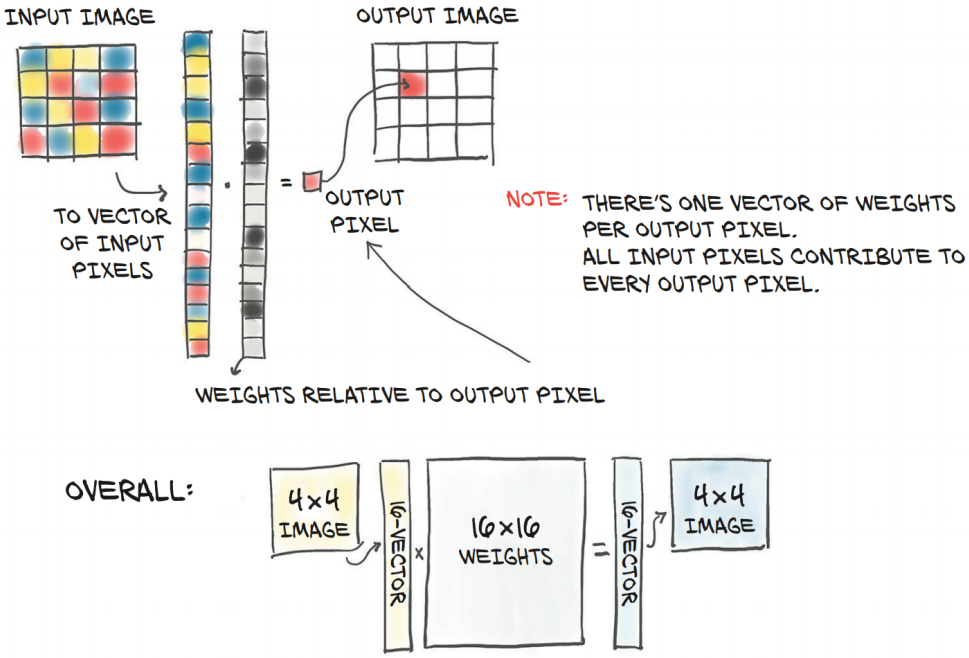

Figure 7.15 Using a fully connected module with an input image: every input pixel is combined with every other to produce each element in the output.

However, this *data argumentation* strategy comes at a cost: the number of hidden features - that is, of parameters - must be large enough to store the information about all of these translated replicas. Due to a mismatch between our problem and our network structure, we end up overfitting our training data, rather than learning the generalized features of what we want the model to detect.

1. Datasets and DataLoaders provide a simple yet effective abstraction for loading and sampling datasets.

2. For a classification task, using the softmax funciton on the output of a network produces values that satisfy the requirements for being interpreted as probabilities. The ideal loss function for classification in this case is obtained by using the output of softmax as the input of a non-negative

log likelihood function. **The combination of softmax and such loss is called cross-entropy** in Pytorch.

3. Simple models can created using nn.Sequentail