a quick rundown on basis functions

Basis functions come up all the time in science, so it's important to get at least an intuitive understanding of them.

First of all, functions and vectors are extremely similar objects, and so we often use the terms them interchangeably in mathematics. If this is a bit confusing, check out the Week 2 video tutorial on the supplementary page of the course website.

Basically the whole idea motivating the use of basis functions is the understanding that there are different ways to represent a function, and sometimes one representation is more useful than another.

For example, let's say I have a big bag of functions, which we'll denote by the set $\{f_i(x)\}$, and each function is defined on the interval $[0, 10]$. Further, let's suppose that all of the functions in our set have a very similar structure: $f_i(x) = a_i sin(x) + b_i x^2$. That is, each function is made up of a weighted sum of two building block functions $sin(x)$ and x^2, and the weights a_i and b_i are called the coefficients.

Now, let's say I randomly draw a function from my big bag of functions, and this function happens to be $f_i(x) = 5sin(x) - 2x^2$. If I were to call up a friend on the phone and describe this function, how would I do it? One way would be to tell my friend the value of the function at every single possible x. This is impossible because there are an infinite number of different x's, but I could probably do a pretty good job if I told my friend the value of the function at $x = 0, x = .001, x = .002, ..., x = 9.998, x = 9.999, x = 10$. So I would say to my friend, "$f(x = 0) = 0, f(x = .001) = .005, f(x = .002) = .01, ...$", the moral of the story being that it would take a very long time, especially if I drew several functions from my bag and had to describe each one in such a manner.

What would be a better way to do this? Well, the easy thing to do would be to first say, "hey friend, every function in my bag has the form $f_i(x) = a_i sin(x) + b_i x^2$." Then, if I

randomly drew the function $f_i(x) = 5sin(x) - 2x^2$, all I would have to tell my friend would be, "$a_i = 5, b_i = -2$". Thus, by representing each function by its coefficients I have made the problem much easier. In this case I have reduced the number of things I have to say to describe the function. (However, keep in mind that this strategy requires knowing the structure of all of the functions.)

Now, using the building block functions $sin(x)$ and x^2, I can build all of the functions in my bag, but I certainly can't build all functions defined over the interval $[0, 10]$. In order to do that, I would have to have a lot more building blocks (an infinite number in fact!). Luckily, there do exist such infinite sets of building blocks. One set is the set of functions associated with the Fourier transform. In this case, each building block is a wave-like function (i.e., a sinusoid) with a frequency slightly different from all the rest. So it turns out that you can represent any function by the coefficients of Fourier building blocks. If your set of building blocks completely span the set of functions you intend to represent, then these building blocks are called basis functions.

In the sparse coding problem, we start with a set of functions, which in this case are images $\{f_i(x, y)\}$, and we try to find a good set of basis functions to describe that set. The magic of the algorithm is in how you define "good". They define "good" in two ways: 1. If you use a smaller number of basis functions than the number of pixels in your image (which you would like to for the sake of efficiency), there will usually be a little bit of error between the true image and its representation using the coefficients of the basis functions. The goal is to make this error as small as possible. 2. In general, you want to choose your basis functions so that you need as few basis functions as possible/the smallest values of the coefficients to represent each image. This is because the coefficients correspond to neural activities, and it is more efficient to have as few neurons active as possible.

The way these two rules are implemented is in the cost function. The sum over squared terms is the error between the reconstructions and the true images, and the term with the λ in it is the penalty for having too many neurons too active at once. The best set of basis functions is then the one that minimizes this cost.