Programming Assignment 3: Forward Noisification

Due Feburary 24, 2023

In the repository you have starter code for the third programming assignment, 'pa3.py.' In this assignment, you will work through constructing a fully connected deep neural network with arbitrary dimensions. You will fix code to compute a forward pass "from the ground up" as well as with torch modules. (We will discuss later *why* to use torch, but for now an appetizer may assuage unbounded curiosity: torch handles differentiation "automatically" so that you do not have to manually compute derivatives by hand.)

You will not (yet) train this model, but merely verify that your manual construction matches the torch construction. In particular, you will see how to specify the architecture *and* model defining parameters. Completion of this code should result in plots which empirically validate equivalence of your forward methods, as in the following plots, which plot the output and error between the torch and ground up model.
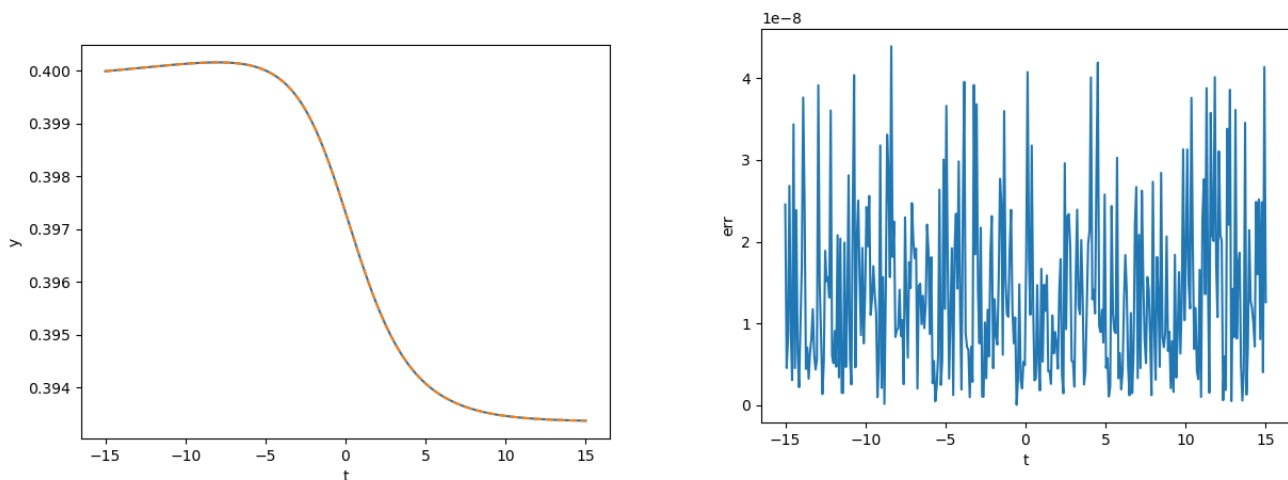
Your output will almost certainly look different, because one of these models is initialized randomly (and the other, of course, initialized to match the first).

A secondary (and almost equivalent in importance) objective is gaining practice debugging ML code. Common pain points include type handling (converting e.g. numpy arrays to torch tensors and back) and dimension matching. Therefore, the code as provided to you is practically complete, but a code demon swept through and inserted a number of bugs which you will need to step-wise fix in order for the code to run correctly. Most errors are independent and fairly clearly identified by error messages, and I have put a pointer for the one particularly tricky one whose fix may involve an earlier correction, depending on how you handled it. This will give you preliminary exposure to setting up and using a debugger (such as exists in VS Code), so that you can step through the code line by line and check that it is behaving as you expect.