19.3.1 Measure Twice, Model Once

Beks has been at the foundation for five years, so by now she knows that writing the checks is the easy part. The hard part is figuring out how much they should be and to whom they should go! In the same way, most of the work in building complicated models doesn't go into the actual coding. Instead, it's figuring out what is going on with the data, cleaning it, and getting it organized. Luckily for Alphabet Soup, Beks always does her due diligence.

When building a computational model, most of the design effort is not writing code to build the complex model. Rather, most of the effort in computational model building is preprocessing and cleaning up the input data. Neural networks are no exception to this rule. In fact, neural networks tend to require the most preprocessing of input data compared to all other statistical and machine learning models. This is because neural networks are really good at identifying patterns and trends in data; therefore, they are susceptible to getting stuck when looking at abstract or raw data. When data has many categorical values, or large gaps between numerical values, a neural network might think that these variables are less important (or more important) than they really are. As a result, the neural network may ignore other variables that should provide more meaningful information to the model.

For example, if a bank wanted to build a neural network model to identify if a company was eligible for a loan, it might look at factors such as a company's net worth. If the bank's input dataset contained information from large fortune 500 companies, such as Google and Facebook, as well as small mom-and-pop stores, the variability in net worth would be outrageous. Without normalizing the input data, a neural network could look at net worth as being a strong indicator of loan eligibility, and as a result, could ignore all other factors, such as debt-to-income ratio, credit status, or requested loan amount. Instead, if the net worth was normalized on a factor such as number of employees, the neural network would be more likely to weigh other factors more evenly to net worth. This would result in a neural network model that assesses loan eligibility more fairly, without introducing any additional risk.

In the next few pages, we'll look at different preprocessing steps that can prepare input data for training neural network models. By the end of this section, we'll no longer need to use dummy input data—we'll be able to apply neural network models to any dataset!