price discrimination. Surely he would have charged a single passenger only 150 gold, and five extra people on the ship meant nothing relative to the ship's actual weight. We took the offer, though, because it was the only choice we had.

I realized something, though. When the captain said he needed strong men, everyone had looked towards me. Only at that point did I interpret *why*.

I had shown them my amazing strength, and now they expected me to be able to haul large crates and pull ropes that were thicker than my neck with immeasurable ease.

I was pretty much screwed.

I quickly sifted through recovery ideas, mindlessly handing over the money I owed. I could slip on something the moment I got on the ship, or perhaps make it seem like one of the ship's crew accidentally fell on my or bludgeoned me or something. Anything to break a leg or an arm as an excuse.

How did I get into this mess? I was just pretending to be strong to impress some adventurers.

Wait- pretending. That's how I can get through this. I can *fake* a wound- or, better yet, I can pretend to *help* on the ship. That way, I'll seem useful.

And so I did. We made it all the way to Gardor on the other side of the sea, and no one suspected a thing.

Writer's Notebook: The Curse of Dimensionality 18 March 2013

This should, in theory, be a quick entry. The Curse of Dimensionality isn't a hard concept to grasp— there's some background you'll need to know, though. Let's see how efficiently I can do this.

A neural network is a group of computational functions that uses probability and statistics in an attempt to simulate the workings of an actual neural system. Neural networks are the basis for handwriting recognition systems, voice recognition systems, and NLP (natural language processing) systems. Generally, a neural network is given a "training set" of data. In a handwriting recognition example, that data might be a bunch of pictures of handwritten sentences, *along with* the actual text written.

Neural networks use their training data to try to assess the probabilities of certain possible results. Generally, when they are informed that they produced correct output, they will add the new data to their training set.

Now, suppose an example in which the neural network needs to produce a "y" value given an "x" value on a 2-dimensional grid. The network will have training data for x/y pairs, and will need to compare the likelihoods of certain y values.

Here's the problem: it is much simpler to process the data if the graph is divided into vertical "slices". Given an x-value, the function can peek into its training data for the closest slice, and then interpolate a more detailed y-value from an average provided by training data. This y-value won't be too accurate, though, if the slices are too big.

That's where dimensionality comes in. Surely, we will have more precise results if we use smaller slices. We increase dimensionality, or the slice precision, and we get more precise answers. The "curse" here is that, for each slice there must be at least one y-value— otherwise, if an x-value input falls into that slice, the network will not be able to provide a result at all.

Thus, we keep broader thresholds for our slices in neural computing so that the "curse" doesn't get us. Sure, we want precision, but too much precision leads down the dark hole that