

---

layout: post title: "title" image: assets/images/x.gif image\_alt: "x" published: false

Fresh off of winning TODO from the OpenAI retro contest, I wanted to keep exploring more AI topics. Somebody told me that the best way to learn was reproducing other people's papers. Not wanting to learn any more Python than I had to, I decided to try to tackle some existing work with [TensorFlow.js](#).

GANs or [Generative Adversarial Networks](#) have always interested me, mainly from the examples I had seen of interesting generated things like [Magic cards](#) to [stock images](#) so I decided to try to reproduce the simplest GAN tutorial I could find with TensorFlow.js

In [An introduction to Generative Adversarial Networks](#) Jonh Glover walks through some of the theory behind GANs and then implements a basic example of generating a 1 dimensional distribution based on [work by Eric Jang](#). Since I'm an engineer and not a physicist, I am going to start with the implementation and then use that to pragmatically learn the theory.

The first step is to be able to generate a 1D distribution. John used what appears to be basic NumPy wizardry, and I am already starting to feel the burn of trying to redo this in a new language. Luckily, I have all of JavaScript's world of packages at my disposal, and a quick google search led me to the [gaussian](#) package. After that, getting the values wasn't too hard.

```
function dataDistribution(N) {  
  let mean = 4;  
  let variance = 0.5;  
  let distribution = gaussian(mean, variance);  
  let i = 0;  
  let samples = [];  
  while (i < N) {  
    samples.push(distribution.ppf(Math.random()));  
    i++;  
  }  
  samples.sort();  
  return samples;  
}
```



TODO make a codepen with a graph of the distribution.