Jackson Curtis

Deep Learning Final Project Write-Up

For the deep learning project I worked in collaboration with Dr. Hartman, Josh Meyers, and Matt Oehler on making a GAN that could generate policy data for car insurance data. The goal was to be able to make realistic looking data that could be published in an actuarial journal without compromising the privacy of those in the real dataset. We used a dataset on French auto liability data made available through the CASdatasets package in R. Although we shared ideas and resources, we each built our own GANs independently to compare.

The project had a few interesting twists that made it different than the face generation assignment. First, we were trying to generate discrete categorical data such as whether the car ran on gas or diesel, what brand it was, etc. Much of our inspiration for what we did came from a paper where they did a similar thing for medical records (<https://arxiv.org/abs/1703.06490>). They suggested using an autoencoder to map the one hot encoded discrete variables to a continuous space, and then train the GAN on the continuous representation before reversing the mapping to get the actual generated data. This seemed to work pretty well although it’s questionable whether the mapping really made it more continuous or just spread the discrete points over Rn.

Another innovation that they suggested that we implemented was a way to avoid mode collapse. The method is called “mini-batch averaging” and it attaches summary statistics about the entire minibatch (which we used 1,000) to each sample of the batch and lets the discriminator see what the sample statistics are when making its decision. This way, if the generator generates one type of sample only, the mean and variance will not be anything like the actual data and it will be easily dismissed.

A surprising amount of work went into writing the code to actually be able to analyze whether what our generator was generating was reasonable. The generator generated something that had to be decoded by the autoencoder, then the continuous variables had to be backtransformed from the mean and standard deviation to the original scales, the discrete variables had to be argmaxed over all the possible categories to see which category it was actually selecting (and we had to make sure that it wasn’t trying to select two categories at once—such as being gas and diesel). I then wrote this all to a csv so I could compare generated results to the original dataset in R.

Looking at the data we were able to generate gave us promising results. The distribution of the data almost always had the right mean and variance, which isn’t surprising given our mini-batch averaging. On the continuous variables, the shape of the distribution tended to be a lot more bell-shaped than the skewed real data. The discrete variables looked pretty good overall. They seemed to have a bit of mode collapse where categories that happened about 1% of the time in the real data never happened in the fake data. Also, the correlations among the continuous variables looked very close to what was actually observed, usually within one or two percent.

More work would need to be done on getting the generator to generate realistically skewed distributions instead of the normalish looking distributions it currently generates.