Andy Liu

Variational Autoencoders are probabilistic, regularized versions of autoencoders that can be used to generate new data, rather than just for dimensionality reduction/data representation. Rather than encoding input as a single point like a normal autoencoder, a VAE encodes input as a distribution over the latent space. A point from this distribution is then sampled and decoded, the reconstruction error computed, and the error backpropagated through the network. In this way, VAEs avoid autoencoders' issues with overfitting. Usually, the encoded distributions are chosen to be normal; however, VAEs work with basically any distribution. Both the covariance and mean of a distribution returned are regularized in order to avoid overfitting (as otherwise, a VAE could just return a very narrow distribution approximating a single point).

I was able to adapt code from this paper using variational autoencoders for collaborative filtering (in this case, it was applied to movie rating data to generate recommendations). I modified the Python notebook that was used (test.ipynb) so that it would work with Tensorflow 2 and Python 3, as it was written with many deprecated packages. (Note that you will have to modify the file paths on the test.ipynb notebook and download the data used at this link). The code currently uses a softmax regression, but it can probably be tweaked further in the future to work with a normal distribution and financial data, if necessary. The paper claimed a multinomial VAE model yielded a recall@20 of 0.395, a recall@50 of 0.537, and an NDCG@100 of 0.426. By running the VAE portions of the Python notebook, I was able to verify the paper's results, computing a Recall@20 of 0.39535, a Recall@50 of 0.5354, and an NDCG@100 of 0.42592, which are comparable to the published results.