Using Neural Networks for Data Generation

Jim Slone
Data 622

In this assignment, we constructed a very rudimentary feedforward neural network, which was used to make predictions based on the dataset. This same network architecture, with some modifications, can also be used as the backbone of a generative model that can construct synthetic samples based on the existing data given to it.

One method by which this standard feedforward neural network architecture can generate new samples is to sample the latent space present in the network's hidden layers, sample vectors from the hidden layers, and perturb them before putting them through a decoder to convert them back to a sample. This can also be done more directly by sampling from the data itself and perturbing the values for the various features present. In essence, the trained information present in the network encodes information about the distribution of samples and their corresponding target value, which allows for the network to be sampled itself to generate new samples. Another method by which this can be achieved is by starting with a target example and feeding the output vector back through the network in reverse to generate a new sample. All of these methods would benefit by adding perturbations, or a small amount of random noise, to the data vectors used as the basis of generation, to ensure that new, non-redundant, samples are being generated. Additionally, it likely will be required to replace the final SOFTMAX activation function in the network with a linear activation function that results in a vector of the input features for a given sample, allowing direct generation of samples. This framework would be classified as an autoencoder, with an encoding part mapping the input vectors onto an intermediate latent space, and a decoding part which maps these latent vectors to output vectors that serve as new samples. This is traditionally done with an encoding latent space which compresses the features, having less parameters than the network has input features.

Another architecture of which can be used for the generation of synthetic samples is a GAN, or Generative Adversarial Network. This architecture is composed of two main components: a Generator and a Discriminator. The Generator is responsible for mapping randomized noise vectors to vectors of input features, generating new samples. The samples are then passed along to the Discriminator, which serves to classify whether the data was generated or not. This process leads to the Discriminator learning better performance in identifying synthetic data and the Generator learning better performance in making more convincing data to "dupe" the Discriminator. This process creates a regressive positive feedback loop which results in the network generating samples of which are closer and closer to the true distribution of the latent space for the input features present in the provided dataset.