Learning Lower-Dimensional Embeddings of CyTOF Data with Deep Autoencoders

Krishnan Srinivasan

Abstract

Autoencoders are an unsupervised learning approach of using neural networks to perform data compression, and can be trained to perform PCA or spectral graph clustering [3, 9, 8]. In this project, we investigate the use of deep sparse and denoising autoencoders as a method of data compression in biology to learn lower-dimensional embeddings and clusterings of single-cell mass cytometry data across different cell types. To validate the model, we first trained our model to learn embeddings for the MNIST dataset and data generated from multivariate gaussian distributions, and compared the embeddings to those generated by shallow, single hidden layer sparse and denoising autoencoders. Next, we experimented with how well the generated embeddings would cluster data from cells that are of the same type. For this, we trained the model on a blood dataset from Dr. Kang’s lab in the Yale School of Medicine, and measured how well k-means clustered labels from the embedding matched the true labels. We then compared our clustering performance to the results from a newly introduced clustering algorithm called Phenograph, which develops the Louvain method clustering approach to work on high-dimensional single-cell data [5]. Finally, we present some visualizations generated by the autoencoder after being trained on a clinical blood dataset from the Yale New Haven Hospital. Using our model, we plotted the middle layer of the autoencoder in two dimensions to highlight structural similarities and differences in the embeddings for blood samples from young and old patients after a kidney transplant. With these plots, we intend to further develop the model to help identify other kinds of phenotypic differences between cell samples from different patients.
1 Introduction

Traditionally, autoencoders have been used as methods to pretrain deep neural networks as a way of learning useful intermediate representations of the data, and initializing the weights of the network to start closer to a setting that can realize and use these representations in training [1]. Different types of autoencoders, sparse, denoising, and contractive, have also been developed as manifold learning methods to learn invariant features of an input space [7]. We investigate the uses of autoencoders in this context and apply it to reduce the dimensionality of CyTOF data and study the structure of the resulting embeddings. This paper is broken into three sections. The first provides a quick background of the mechanics of sparse and denoising autoencoders. The next section details the architecture of the deep autoencoder we use for mass cytometry (CyTOF) data, including the implementation details. Finally, we discuss how the model performs on test datasets, clustering CyTOF data, and visualizing CyTOF data.

2 Autoencoder Background

2.1 Simple Autoencoder

A basic autoencoder consists of two components, a fully-connected layer known as the “encoder,” and a fully-connected layer called the “decoder” [2]. The goal of the autoencoder is to train these two layers to reduce the input dimensions down to a lower dimensional encoding, and then return back to the dimension of the input using the decoder layer, as shown in Figure 1. The network is trained with a loss function that computes how well the original input was reconstructed from the encoding, so that the encoder learns how to reduce the dimensionality of the original data while preserving the necessary information required for the decoder to reconstruct the original input. Common choices for the loss function include squared error, or the cross-entropy of the reconstruction, which roughly measures the number of bits of information that are preserved by the reconstruction. Sparse and denoising autoencoders build on this idea, and make modifications to the architecture and loss function to learn invariant features better than a simple autoencoder can.

2.2 Sparse and Denoising Autoencoders

Another way to extract invariant features from high dimensional data is by using a sparse encoding instead of a lower dimensional encoding. This involves changing the model above in two ways. The first is removing the constraint that the encoding layer has to have fewer dimensions than the original data. Given some way to enforce sparsity on the encoding, it is possible for the middle layer to have
Figure 1: A traditional autoencoder

more dimensions than the original input, much like a multiplexer creates a sparse output from an dense input. The second difference is the change in loss function that enables this sparsity. There are many ways to approach the sparsity constraint of the encoding, and for the purposes of this project, we chose to penalize the l1-norm of the encoding, so in order to minimize the loss function, the autoencoder would need to find a way to make the encoding layer sparse, in addition to reconstructing the original input well.

Denoising autoencoders offer yet another approach to manifold learning, with the main difference being that they corrupt the data before it is passed into the network. This is typically done by adding random noise sampled from a Gaussian distribution to the original input. Then, simply train the autoencoder (using either a sparse or simple approach) to reconstruct something as close to the original input, as possible, while eliminating the noise.

3 Implementation

The simple, 3-layer autoencoders that are described above (with input, encoding, and decoding layers), while useful for illustrating the concept of autoencoders, and are not sufficient for the task of learning an embedding of higher dimensional biological data. For this reason, by making the model deeper and performing a series of nonlinear transformations on the input, a better result can be found. The same approach can be used, for example, to unravel a multi-dimensional swiss roll into two dimensions using an autoencoder, a common task in nonlinear dimensionality reduction. Figure 2 shows the deep sparse and denoising autoencoder architectures we used for the experiments, including the activation function and number of nodes used in each layer. To make the encoding layers sparse, one addition to the model is an l1 regularization term of the activations of an encoding layer being added to the loss function. This way, the sparse layers learn to compress their input into a sparse signal with only a few non-zero values. Another important detail is that we scaled the values of each column of input between 0 and 1, so that we could use the sigmoid activation function in the last layer.
4 Experiments

4.1 Testing the Model

To validate the deep sparse and denoising autoencoders, we first trained them on randomly generated samples from a multivariate Gaussian distribution to form a middle layer encoding that could separate samples from different univariate distributions into two dimensions, as shown by Figures 3a and 3b. Lastly, we ensured that our deep autoencoder successfully embedded the MNIST dataset [4], a basic benchmark standard of autoencoder embeddings, by creating a visualization that could visually cluster the different labeled digits into a 2 dimensional plot. See Figures 3c and 3d for the comparison of the embeddings from the deep and simple sparse autoencoders.

4.2 Clustering Healthy Blood Data by Cell Type

Next, we gauged how well the autoencoder finds encodings of the dataset that would allow us to identify different cell types. For this task, we used a healthy blood dataset from Dr. Kang’s lab, with labeled cell types for each sample. Then, the autoencoder is used to make a sparse and 2-dimensional encoding of the each sample, and each set of transformed samples is clustered using k-means clustering. We used the number of labeled cell types for k. We compare the clustering results, which measure cluster accuracy using the Rand score between the true and predicted clusters, between our model and a recently published cell labeling/clustering algorithm called Phenograph [5]. The results are shown in Table 1.
Figure 3: (a) Random samples generated from 3D multivariate normal, plotting first two dimensions. (b) Embedding generated by training autoencoder on random samples. (c) Embedding from a regular sparse autoencoder. (d) Embedding from a deep sparse autoencoder

<table>
<thead>
<tr>
<th>Middle Layer Size</th>
<th>Phenograph</th>
<th>Sparse Layer Clust.</th>
<th>Middle Layer Clust.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>39 %</td>
<td>53 %</td>
<td>53 %</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>69 %</td>
<td>69 %</td>
</tr>
</tbody>
</table>

Table 1: Comparing Rand Scores between true clusters and clusters from Phenograph and Sparse/Middle Layer codes

4.3 Visualizing Blood Data of Kidney Transplant Patients

Finally, we demonstrate the ability of autoencoders to visualize CyTOF data and uncover underlying phenotypic differences between cell samples from different pa-
Figure 4: (a) Embedding showing a branching structure. (b) Embedding of the other patient type with less branching. Both are colored with respect to activations along different CyTOF channels.

patients. For this task, we used blood data from old and young kidney transplant patients, and compared the structure of the 2 dimensional embeddings from different autoencoders that were each trained on samples from one patient. As can be seen from Figure 4, of the 8 patients that we observed data from, each of which was from a different age group, there are clear structural differences between the embeddings of samples from young patients and old patients (Note: we have currently held out the age of the patients in our plots, and only know that there are two distinct age groups corresponding to our results. We are currently in the process of tweaking the model and creating more plots that can show a more clear distinction between the two groups). The distinct separation between the different embedding structures seems promising, and has helped us identify a possible application for this kind of model in the future.

5 Future Work

Our results so far have allowed us to justify the use-cases for autoencoders as a way of learning and visualizing lower-dimensional embeddings of biological data. There are many next steps from this point, starting with developing a more rigorous methodology in testing the results of our application to use the learned embedding to locate cell groups. For example, we can start to compare our embeddings to what is output by t-SNE (t-distributed stochastic neighbor embedding) [6], and other embedding/clustering methods. We are also looking at extensions into visualizing and potentially imputing rna-seq data, which is known for having huge errors with missing values.
6 Conclusion

We have begun to show some of the viable applications of visualizing and clustering CyTOF data through embeddings learned by deep autoencoders. The initial results are encouraging, as the model was able to cluster well by cell types compared to existing methods, and the visualizations are useful at identifying phenotypic traits. However, one critical area for improvement moving forward is the methodology used to show these results, as it is still largely difficult to prove using any existing theoretical framework that the visual embeddings or clusters are accurate and informative. As I move toward the continuation of this project in the next term of coursework, we will continue to develop the procedures for evaluation, and will begin to try other deep learning methods, with a possible look into GANs to impute missing data.

References