Project Title: Exploring Representation Learning with Simple Images of Shapes

Kevin Lewis
lewis.3164@osu.edu

Michael Lin
lin.3976@osu.edu

Shashank Raghuraj
raghuraj.2@osu.edu

1 Project Type

Reproducing examples in the textbook

2 Project Introduction

In this project, our group will explore representation learning with simple images of shapes with autoencoders and contrastive learning. Representation learning allows us to automatically discover features from raw data. In this project, we will replicate the textbook examples on autoencoders and contrastive learning, as seen here: https://visionbook.mit.edu/representation_learning.html#experiment-do-autoencoders-learn-useful-representations and https://visionbook.mit.edu/representation_learning.html#sec-representation_learning-expt_designing_embeddings_with_contrastive_learning

3 Project Motivation

This project is interesting for a couple of reasons. First, the learning model does not seem too difficult itself, so we may be able to create both models without using any machine learning libraries. This will be a useful learning tool for understanding more about how these models work behind the scenes. Also, it will be interesting to explore the different things that are possible with the encoded representations (for example, animating a transition between different shapes with autoencoders).

4 Project Plan

4.1 Baseline approach

We will use a simple fully connected neural network as a baseline approach for our project. This is because it should be enough for our purposes, and it is simple enough that we will be able to build it ground up. One potential limitation of this is that using a fully connected network will limit the sizes of the hidden layers. If they were too big, the model may be too big and slow to train effectively.

4.2 Advanced approach

Beyond this, we may add some convolutional layers to see how they impact the performance and accuracy of the model. This will allow us to have a few layers without needing to decrease layer size. We may also add some skip connections in these early layers to emulate a ResNet, in order to explore how this chance affects accuracy.

4.3 Validation plan

We will have a dataset of shapes with different colors, sizes, and rotations to test our network on. We do not currently have access to this dataset, but it will be easy to generate a dataset like it. The textbook used 64,000 images, so we will look to generate a datasest of similar size.

Similarly to the textbook, we will do some nearest-neighbor search to validate our models.

4.4 Computational resources

This model will be simple enough to run on our own computers.

4.5 Library, Tool, etc.

We will write this in Python. The main library we use will likely be either Numpy or CuPy for fast matrix operations, though we will likely write the actual neural network code ourselves. We may also use PyTorch if we choose to go with a machine learning library.

4.6 Estimated baseline algorithm runtime

We are generating the dataset ourself, and since the dataset the texbook used is not too big, we can tune the size of our images so that one epoch worth of training takes at most a couple minutes.

5 Workload

The workload for this project will consists of reading through examples to understand the prupose of autoencoders, and focus on the mathematical foundations they're used with. Next, we'll experiment with simple neural network implementations before diving into more complicated models. The core workload will be the implementation of the model in which we will start by building a baseline fully connected network from scratch. We will spend the remainder of our time conducting experiments to evaluate the performance, accuracy, and representation quality.

6 Ideal result

If everything is successful, the autoencoder model should be able to reconstruct shapes from their compressed form. The contrastive model should produce embeddings that shows similarities between different images. The outcome will not only be a replica of the textbook experiments but also be well optimized.

6.1 Insights

We'll first understand how autoencoders learn compressed representations and how well they preserve essential features during the reconstruction of the object. We'll also learn how to map similar images closer together in the latent space. This project will also help us learn how to build a neural network from scratch.

7 Potential Risk

It is possible we will have difficulty doing the calculus to get the project working. If this is too much of a problem for us, we can switch to using PyTorch, which should handle this for us.

8 Duplication Statement

There are surely autodencoders and contrastive learning models online, although not on the dataset we plan on using. We will not use these models, and we will have to create our own transformations for contrastive learning.