Image Compression with an Autoencoder Neural Network

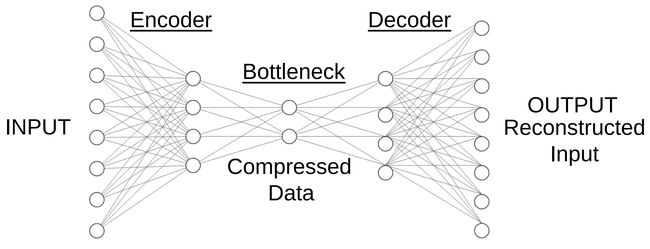
By Xavier Markowitz and Austin Perera

Introduction

An autoencoder is a special type of network that compresses and decompresses an inputted data sample. Unlike other neural networks, an autoencoder attempts to reconstruct exactly the input it was given. This can be used for image denoising, image creation (similar to that of Dall-E), data augmentation, and much more.

An autoencoder is made of two main parts:
The encoder - compresses while learning the important features of the data. This abstract learnt representation is called the latent space .

The decoder - attempts to reconstruct the original data given the compressed representation.



The bottleneck is the point at which the data is the most compressed. As the amount of nodes in the bottleneck decreases, the network is forced to compress the data more, thus further limiting the latent space feature representation.

- How does bottleneck size affect the final generated image?

- Can we see a link between the latent space representation and generated image?

Methods and Hypothesis

We use two image datasets:
- MNIST - provides a good environment to analyze how the network recognizes lines and shapes
- STL-10 - allow us to see how the network reconstructs colorful images

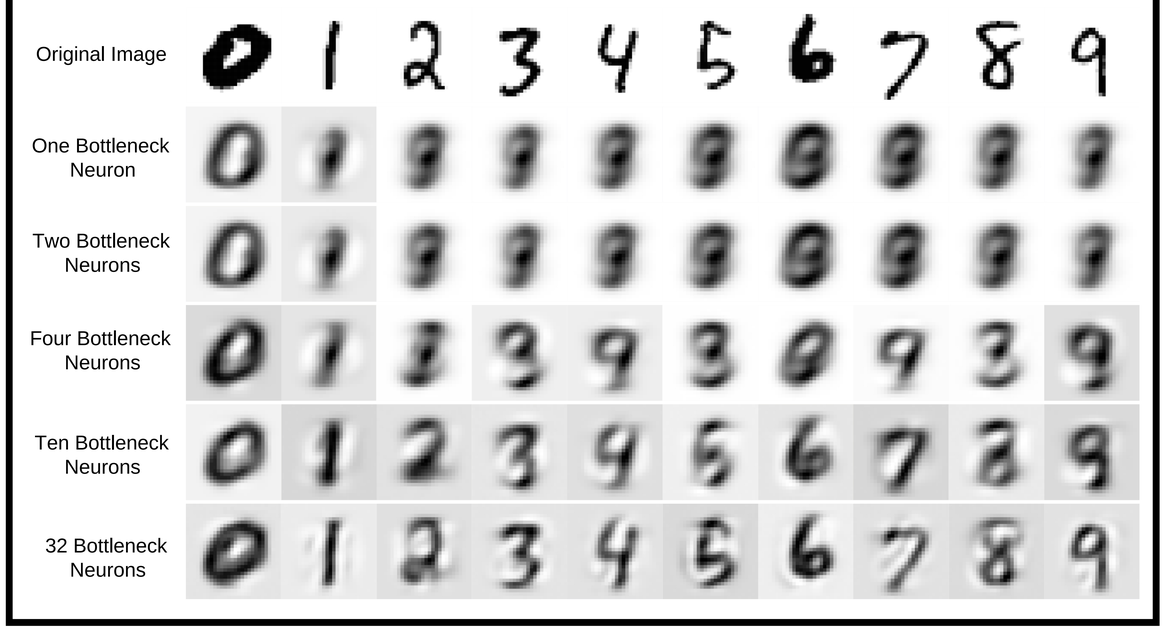
Train five different versions of our autoencoder network, each with a different amount of neurons in the bottleneck. The sizes were 1, 2, 4, 10, and 32.

To control for the bottleneck, we change no other parameters between each network
- Learning rate: .0001
- Early stopping patience: 3
- Mini-batch size: 750

We hypothesize that we will be able to compress then recreate the images from our dataset very closely. Reducing the bottleneck size will create a flatter, but still recognizable image output.

Results

MNIST Tests



- The generated outputs from one neuron bottleneck show the network being forced to create the most general output for each data sample because of the limited feature representation.

- As neurons are added to the bottleneck, generated images become more individual, and contain more recognizable features from the original.

- The generated outputs for STL-10 are more abstract and less defined than the outputs for MNIST. Our network is able to approximate color and color regions, but not the complex details present in the STL data.

- There is a clear outlier present in the 3rd column, last row of the generated STL-10 images. We are not completely sure what the origin of this is.

STL-10 Tests

Results Cont.

Table of average one neuron bottleneck values from network trained on MNIST:

0	0.6731367573674236
1	1.8448098856972177
2	1.25390981851947
3	1.2561716698774017
4	1.3548389480215954
5	1.2284700513353797
6	1.1456833659109342
7	1.4792098000007845
8	1.288045598297271
9	1.3867185656332048

- Notice the value for zero is dramatically different from other latent space values.

- Compare this data with the generated MNIST images for one neuron. It is clear that from just one bottleneck neuron, the network understood that zero was very different physically from the other numbers.

Conclusions

- There is a clear relationship between the values in the latent space and the generated images.

- As the amount of bottle neurons increases, the network is able to understand and reconstruct more specific individual information about each image.

- Our network was not very good at recreating the complex images depicted in the STL-10 dataset.

Further Research

For further study we are going to look into creating a convolution based autoencoder for two reasons:
- Reduce number of trained weights - reducing computation resources
- Greater ability to maintain details of the original in generated images.

Acknowledgments

We would like to extend our thanks to Colby College, the Colby Computer Science Department, and Professor Oliver Layton for making this project possible.