High-ratio image compression: an exploration of autoencoder hyperparameter selection to minimise reconstruction error

Hassan R. S. Andrabi

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc odio nisl, tempus eget cursus et, consectetur id tellus. Praesent pharetra sodales eleifend. Quisque nec blandit nisl. Curabitur sit amet odio quam. Mauris sodales sollicitudin diam, non pulvinar felis semper a. Nunc quis sapien hendrerit, pretium ipsum ac, aliquet lacus. Curabitur facilisis sit amet erat ut suscipit.

1 Introduction

Ours is the age of information: where digital data is ubiquitous, and computational resources are in consequently constrained supply. In this climate, a fundamental problem persists in the efficient management of data to limit the computational expense of information storage and exchange. On this account, data-compression techniques seek to alleviate the computational burden of sending and storing digital information by compressing data to representations of reduced size. In particular, practical applications of such techniques can be broadly divided as employing either lossless, or lossy compression. Lossless techniques comprise the more common class of compression algorithms, which target perfect reconstruction of input data from compressed formats. While such techniques offer uncompromised data-accuracy between compressed and uncompressed formats, recent reports show that practical applications rarely achieve compression ratios in excess of two- to four-times the size of the original data [4]. On the other hand, lossy data-compression techniques are able to achieve exceptionally high compression ratios at the expense of degraded data accuracy. High compression-ratios substantially lower file sizes, and thereby reduce the computational expense of storing and sending information. Accordingly, in scenarios where speed of information exchange is paramount and minor degradation in data-accuracy is unlikely to be noticed by end-users (such as in the case of image-based exchanges), lossy compression becomes an attractive approach.

To this end, among the most successful non-probabalistic approaches to lossy compression is through the use of autoencoder networks. Autoencoders are representation-learning

applications of unsupervised artificial neural networks (ANNs), that seek to learn mappings of high-dimensional data to meaningful lower-dimensional representations. By learning fundamental representations of data, autoencoders enable fast transformation between highly compressed and uncompressed formats, and thereby enable storage and exchange of large files in compressed and computationally inexpensive formats. Indeed, several works demonstrate the ability of autoencoders to learn compressed representations of image data, with promising reconstruction accuracy [1, 2, 5]. Despite the interest autoencoder networks have generated, little documentation of the effect of variation in model hyperparameters has emerged to this date. Recognising this, the analysis in this note seeks to evaluate the reconstruction accuracy of various autoencoder network architectures to image compression of high-dimensional image data. To this end, the present analysis will investigate variations in reconstruction-error resulting from modifications to compression-ratios enforced by the autoencoder, and inclusion or exclusion of bias nodes in the network. In particular, the analysis will address the following research questions:

> **RQ1**: How does the selection of compression ratio influence reconstruction accuracy of compressed images?
>
> **RQ2**: How does the inclusion/exclusion of bias nodes affect reconstruction accuracy of compressed images?

The remainder of the paper is structures as follows. Section 2 provides an overview of the dataset used in this analysis. Section 3 describes the general experimental framework, and the employed model estimation methodology. Section 4 presents and discusses the results. Finally, Section 5 concludes the paper.

2 Dataset

The analysis in this note employs the MNIST Database of Handwritten Digit Images (available online: `http://yann.lecun.com/exdb/mnist/`) [3]. The dataset contains a normalised subset of 70,000 annotated images of handwritten digits collected from the much larger NIST database. The MNIST dataset modifies instances of handwritten digits in the original NIST database to ensure an equal distribution of instances with respect to the circumstances of their collection. In particular, the dataset contains an equal number of instances of handwritten digits collected US census bureaus workers, and from US high school students. In total, 35,000 instances are collected from each category of writer respectively.

Each digit in the dataset is normalised in grey-scale (0-255), and aligned by translating the centre of pixel-mass to be positioned at the centre of a 28x28 pixel grid. For the purpose of model estimation, the dataset is partitioned into a training set of 60,000 images, containing equal proportions of instances collected from either category of writer. The remaining 10,000 instances comprise the validation set, and are used for model evaluation. Example instances of images from this dataset are presented in 1. The dataset is balanced with respect to the distribution of digit classes. Class distributions across the dataset are visualised in 2.

3 Experimental method

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc odio nisl, tempus eget cursus et, consectetur id tellus. Praesent pharetra sodales eleifend. Quisque nec blandit nisl. Curabitur sit amet odio quam. Mauris sodales sollicitudin diam, non pulvinar felis semper a. Nunc quis sapien hendrerit, pretium ipsum ac, aliquet lacus. Curabitur facilisis sit amet erat ut suscipit.

4 Results

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc odio nisl, tempus eget cursus et, consectetur id tellus. Praesent pharetra sodales eleifend. Quisque nec blandit nisl. Curabitur sit amet odio quam. Mauris sodales sollicitudin diam, non pulvinar felis semper a. Nunc quis sapien hendrerit, pretium ipsum ac, aliquet lacus. Curabitur facilisis sit amet erat ut suscipit.

5 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc odio nisl, tempus eget cursus et, consectetur id tellus. Praesent pharetra sodales eleifend. Quisque nec blandit nisl. Curabitur sit amet odio quam. Mauris sodales sollicitudin diam, non pulvinar felis semper a. Nunc quis sapien hendrerit, pretium ipsum ac, aliquet lacus. Curabitur facilisis sit amet erat ut suscipit.

References

[1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2016.

[2] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Deep convolutional autoencoder-based lossy image compression. In *2018 Picture Coding Symposium (PCS)*, pages 253–257. IEEE, 2018.

[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

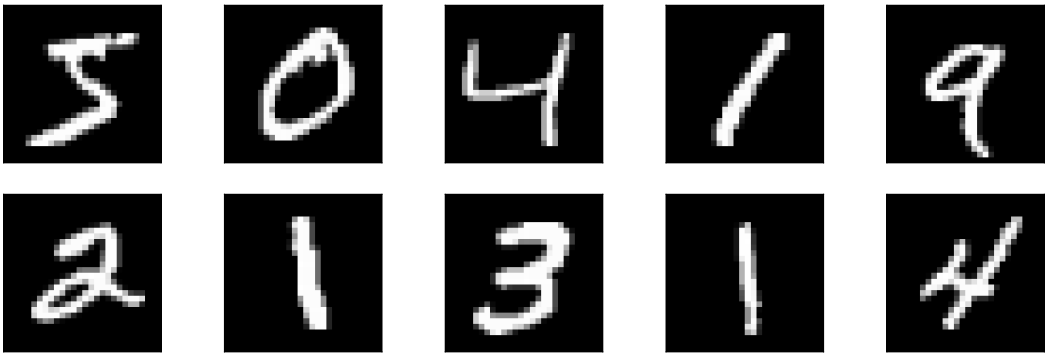
[4] Sparsh Mittal and Jeffrey S Vetter. A survey of architectural approaches for data compression in cache and main memory systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1524–1536, 2015.

[5] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017.

List of Figures

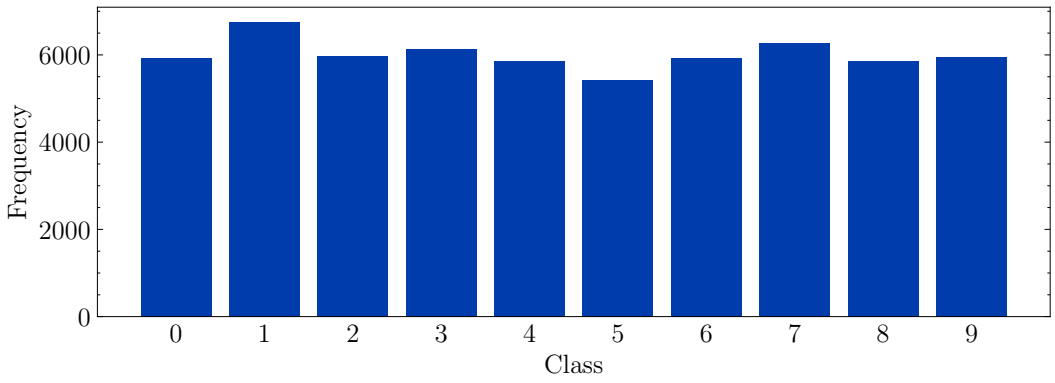
1 Examples of MNIST instances 6
2 Distribution of classes in MNIST dataset 6

Figure 1: Examples of MNIST instances



Notes: Example instances from the MNIST dataset. Each instance is normalised in grey-scale (0-255), and aligned by translating the centre of pixel-mass to be positioned at the centre of a 28x28 pixel grid.

Figure 2: Distribution of classes in MNIST dataset



Notes: Class distribution of digit instances (0 - 9) in the full MNIST dataset, across both training (N=60,000) and validation (N=10,000) partitions.