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. 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 enable 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 enable smaller file sizes, and thereby reduce the computational expense of storing and sending information. Accordingly, in scenarios where speed of information exchange outweighs minor degradation in data-accuracy (such as in the case of multimedia messaging), lossy compression becomes an attractive approach.

To this end, among the most successful non-probabalistic approaches to lossy compression occur 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 of data between highly compressed and uncompressed formats. This, in turn, enables computationally inexpensive storage and exchange of large files. 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 empirical evidence on 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 for 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 Figure 1. The dataset is balanced with respect to the distribution of digit classes. Class distributions across the dataset are visualised in Figure 2.

3 Experimental method

The following section provides a conceptual overview of the intuition behind the autoencoder, and thereafter outlines model architectures and estimation methodology employed to evaluate the role of compression and bias hyperparameters in reconstruction accuracy of encoded images.

3.1 Intuition of the autoencoder

While conceptually simple, autoencoders are capable of learning powerful representations of high-dimensional input data. The general intuition underpinning autoencoder networks is as follows: for an arbitrary high-dimensional input, x, an autoencoder first attempts to encode the dimensions of the input instance to a lower-dimensional space through a learned mapping, E_ϕ, such that the encoded message, z, is given by:

$$z = E_\phi(x) \tag{1}$$
$$= \sigma(W_E x + b_E), \tag{2}$$

where σ is an activation function, such as a Sigmoid function, W_E is a matrix of incrementally trained encoding weights, and b_E is a vector of incrementally trained encoding bias parameters. The ratio between the respective dimensions of the input instance, x, and the encoded lower-dimensional representation, z, represents the *compression ratio* of the model.

Thereafter, the autoencoder attempts to reconstruct the original high-dimensional representation from the encoded message, z, through a concurrently learned mapping, D_ϕ, such that the reconstructed representation, x', is given by:

$$x' = D_\phi(z) \tag{3}$$
$$= \sigma(W_D z + b_D), \tag{4}$$

where W_D is a matrix of incrementally trained decoding weights, and b_D is a vector of incrementally trained decoding bias parameters. The accuracy of the reconstructed image, x' is thereafter compared to the original high-dimensional input, x, by way of a loss function, ϵ. Mean squared error (MSE) is a common choice of loss function in autoencoder networks, such that the loss, ϵ, is given by:

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_i')^2, \tag{5}$$

where N is the number of instances in the training partition of the dataset. As the autoencoder progresses through the training phase, E_ϕ and D_ϕ are incrementally learned through backpropogation of reconstruction error.

3.2 Model estimation methodology

To evaluate the role of compression and bias hyperparameters in autoencoder network architectures, the analysis in this note adopts a fundamentally simple experimental framework.

A generic autoencoder architecture is implemented, consisting of an input layer with 784 nodes; a single fully-connected hidden layer with a variable number of nodes; and an output layer with 784 nodes. A Sigmoid activation function is used to calculate activations in hidden and output nodes. To assess the role of compression ratios in reconstruction performance, the number of nodes in the hidden layer is varied across seven values: 2, 4, 8, 14, 28, 56, and 112. This generates seven distinct autoencoder architectures, with corresponding compression ratios of 392x, 196x, 98x, 49x, 24.5x, 12.25x, and 6.125x respectively. Each architecture is modelled twice: once with the inclusion of a constant bias unit, and once with bias set to zero. Figures 3 and 4 present diagrams of the generic autoencoder architecture with bias units set to zero and one respectively.

In total, the employed experimental framework results in fourteen distinct combinations of autoencoder architectures, spanning across seven parameterisations of compression ratios, and two parameterisations of bias nodes. Each model is trained on the training partition of the MNIST dataset (N=60,000) for 50 epochs. Learning rate is set at 0.01, and invariant between models. At the end of each epoch, reconstruction accuracy of each autoencoder is assessed by calculating average mean squared error (MSE) across the validation partition of the training dataset (N=10,000). Traces of MSE for each model architecture throughout the training phase are presented in Figure 5.

4 Results

Reconstruction error of decoded images monotonically increased with increasing compression ratios (Table 1, Figure 5). This result is unsurprising, as increasing the number of dimensions in compressed representations of data enables more refined encoding of initial high-dimensional inputs.

As can be expected, the best reconstruction performance was achieved by models with 128 hidden nodes (compression ratio: 6.125).

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.

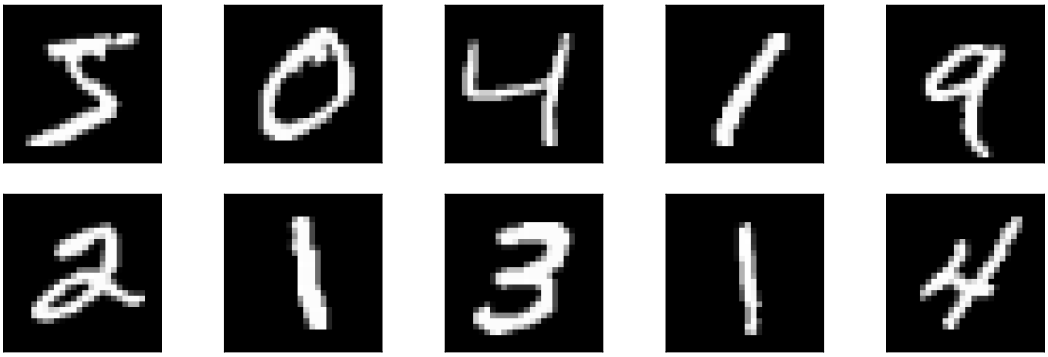
[4] Sparsh Mittal and Jeffrey S Vetter. A survey of architectural approaches for data compression in cache and main memory systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1524–1536, 2015.

[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

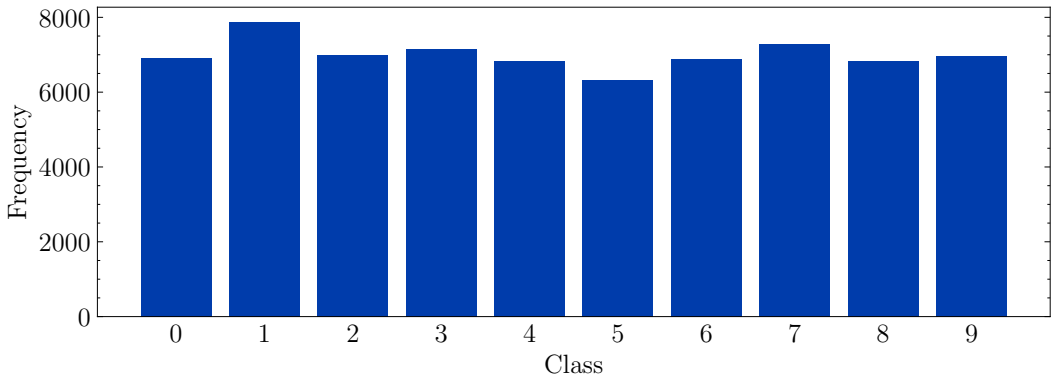
1 Examples of MNIST instances 7
2 Distribution of classes in MNIST dataset 7
3 Generic autoencoder architecture, without bias 8
4 Generic autoencoder architecture, with bias 9
5 Reconstruction mean-squared error throughout training phase, disaggregated by network architecture 10
6 Reconstructed instances disaggregated by compression ratio (no bias units) . 11
7 Reconstructed instances disaggregated by compression ratio (including bias units) 12

Figure 1: Examples of MNIST instances



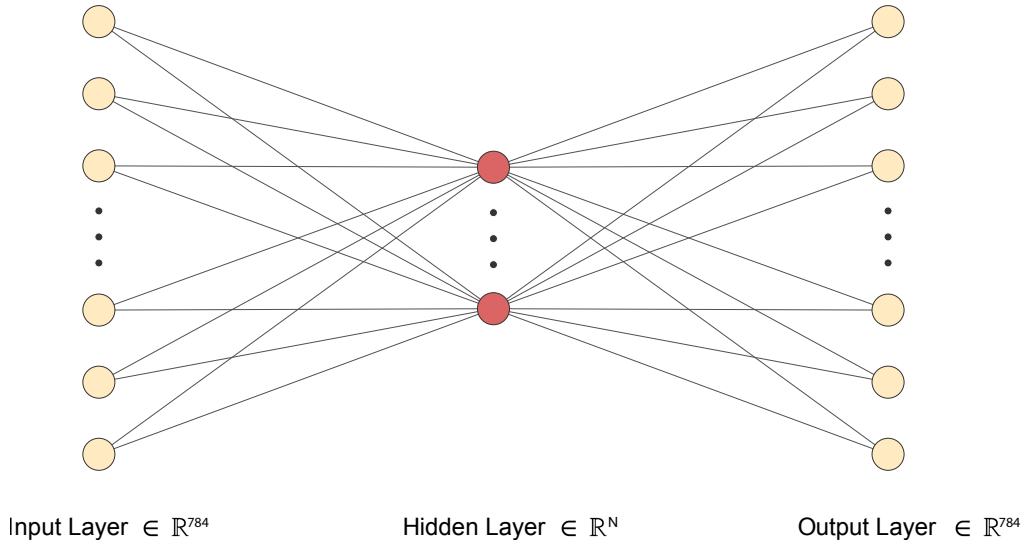
Notes: Example instances from the MNIST dataset. Each instance is normalised in grey-scale (0-255), and aligned by translating the centre of pixel-mass to be positioned at the centre of a 28x28 pixel grid.

Figure 2: Distribution of classes in MNIST dataset



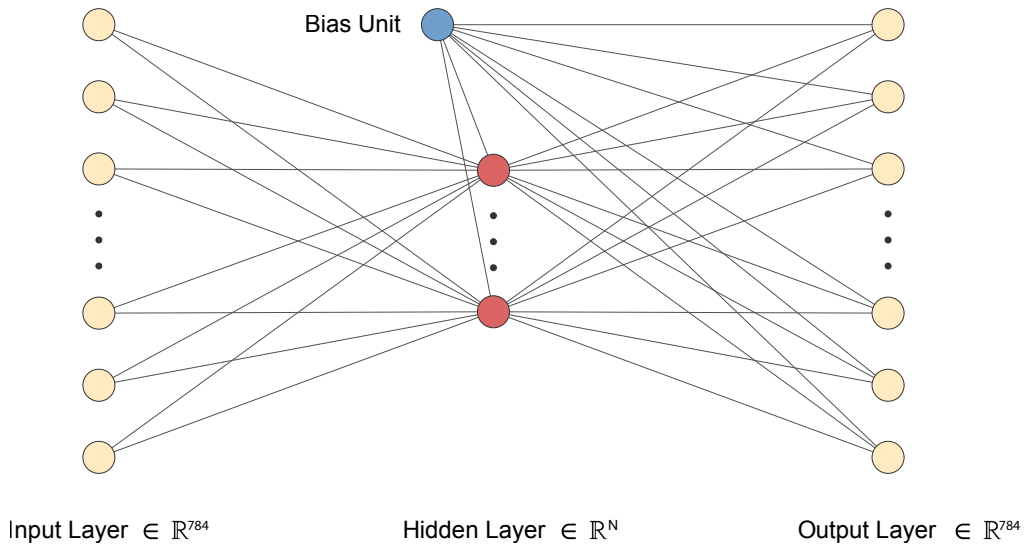
Notes: Class distribution of digit instances (0 - 9) in the full MNIST dataset, across both training (N=60,000) and validation (N=10,000) partitions.

Figure 3: Generic autoencoder architecture, without bias



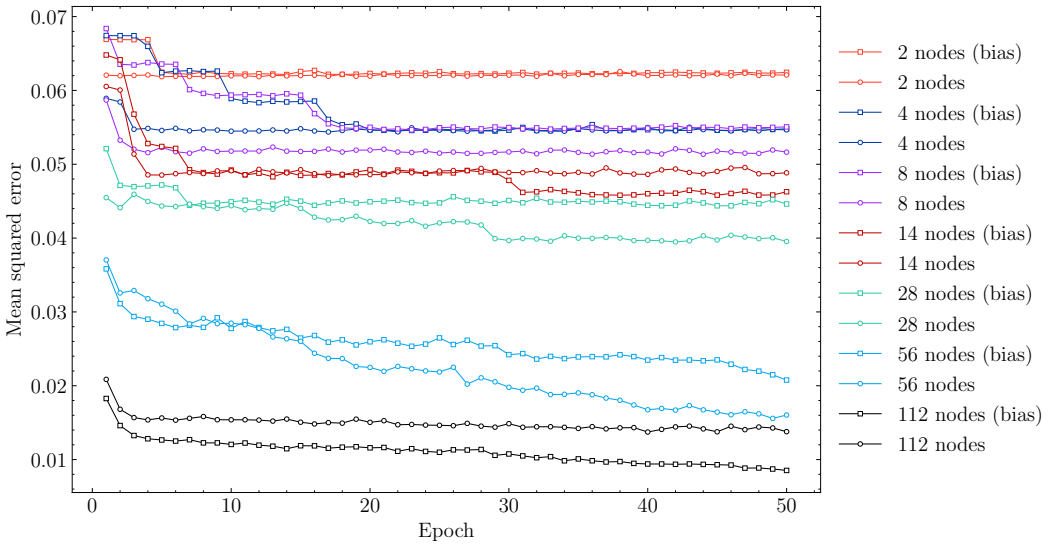
Notes: Input and output layers each contain 784 nodes respectively, in congruence with the dimensions of MNIST image instance vectors. Intermediate nodes are omitted for ease of interpretability. Hidden layers contain a variable number of nodes, N, varied across seven values: 2, 4, 8, 14, 28, 56, and 112. Encoding of inputs occurs through weighting of activations between input and hidden layers. Image reconstruction occurs through weighting of activations between hidden and output layers.

Figure 4: Generic autoencoder architecture, with bias



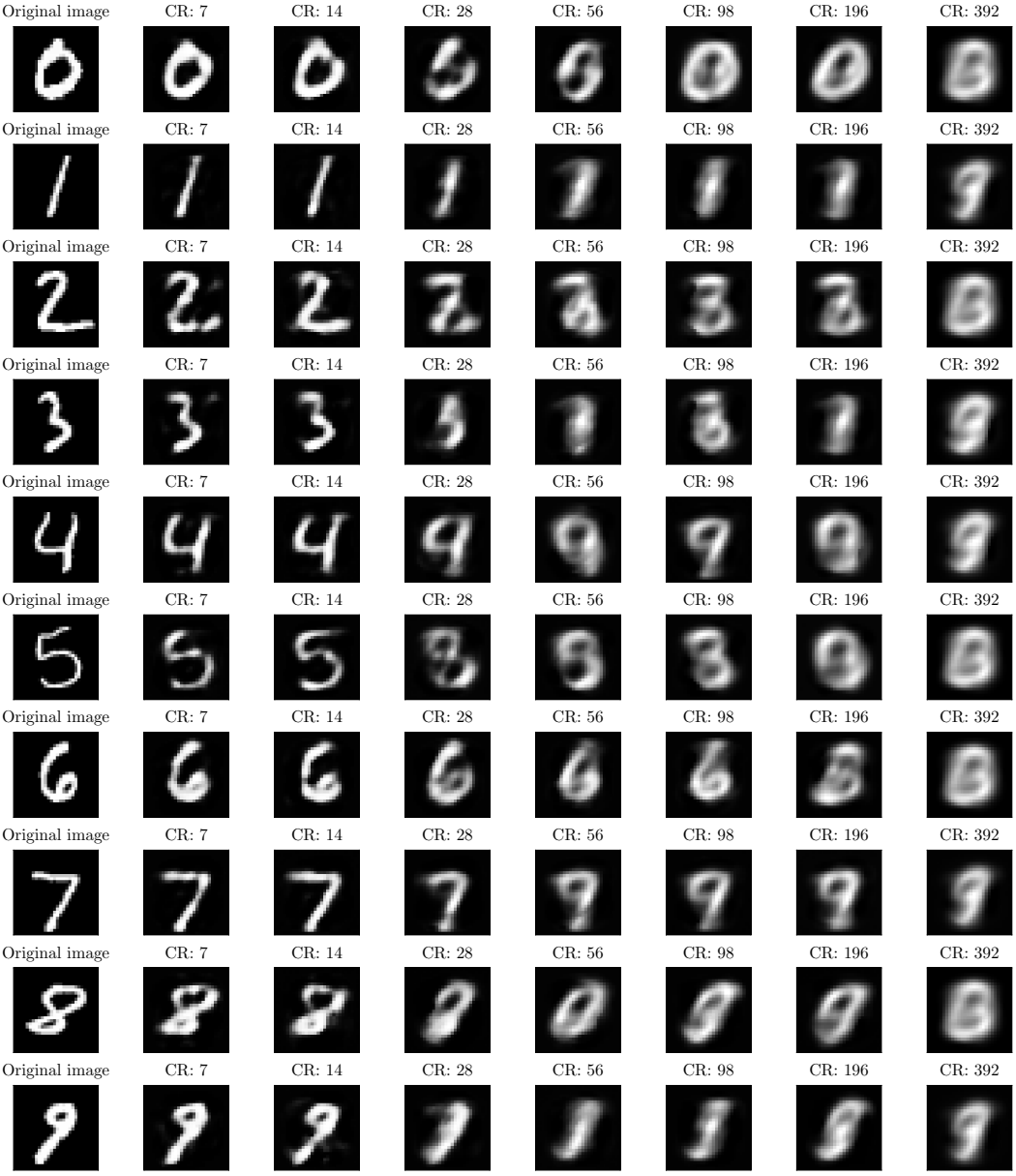
Notes: Input and output layers each contain 784 nodes respectively, in congruence with the dimensions of MNIST image instance vectors. Intermediate nodes are omitted for ease of interpretability. Hidden layers contain a variable number of nodes, N, varied across seven values: 2, 4, 8, 14, 28, 56, and 112. A constant bias affects activations in hidden and output nodes differentially through learned bias weights. Encoding of inputs occurs through weighting of activations between input and hidden layers. Image reconstruction occurs through weighting of activations between hidden and output layers.

Figure 5: Reconstruction mean-squared error throughout training phase, disaggregated by network architecture



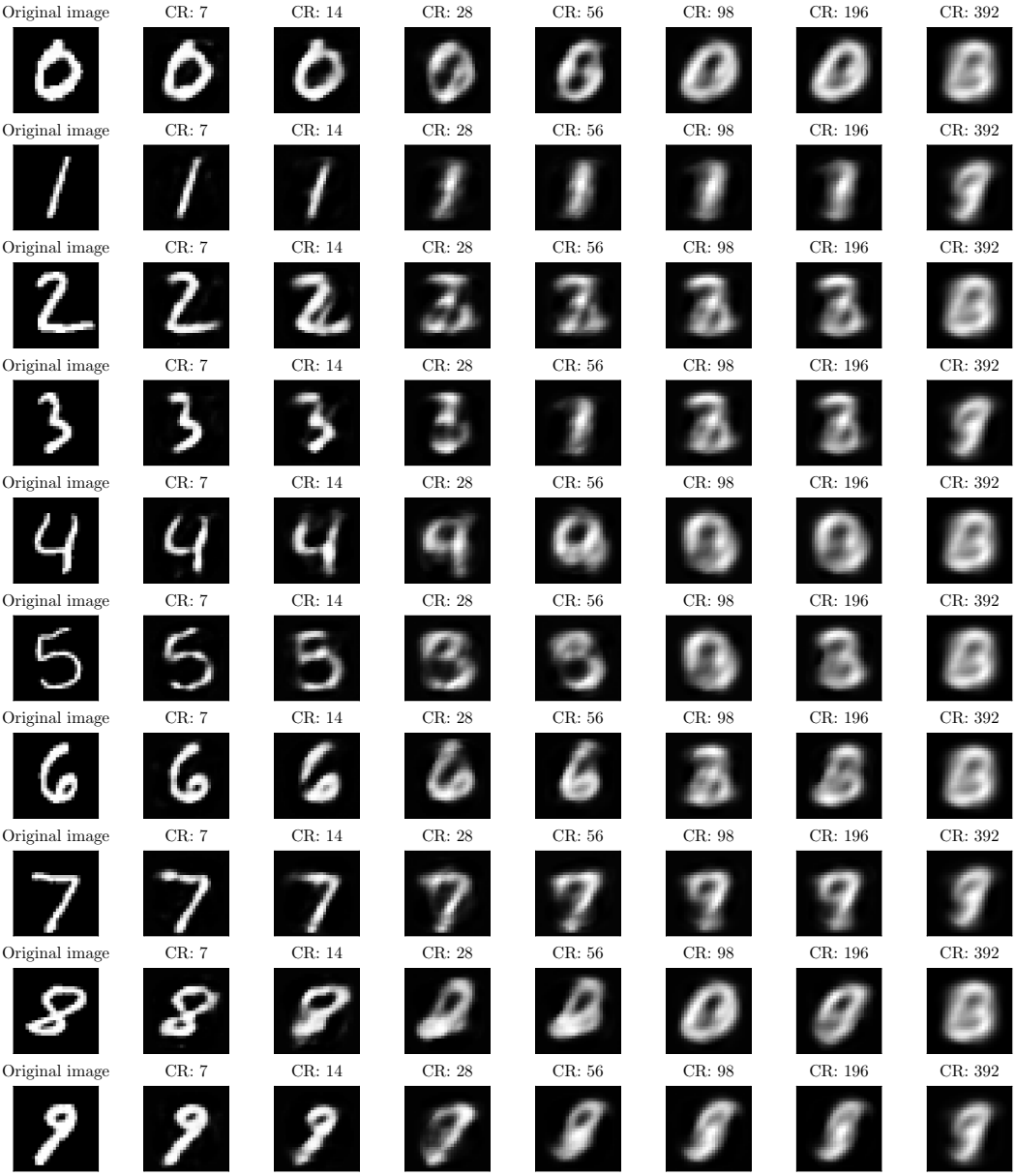
Notes: Mean-squared error (MSE) of autoencoder networks across validation partition of MNIST dataset. MSE was calculated through forward propogation of MNIST validation instances (N=10,000) using trained weights at each discrete training timestep. Learning rate for each model was set at 0.01.

Figure 6: Reconstructed instances disaggregated by compression ratio (no bias units)



Notes: Reconstructed images are generated by encoding and decoding a stimulus image using a trained autoencoder networks with varying hidden layers, and bias set to zero. Columns separate reconstructed images by compression ratio (CR). Rows separate reconstructed images by the input stimulus (uncompressed) image. The first entry in each row is the stimulus image used to generate the following reconstructions.

Figure 7: Reconstructed instances disaggregated by compression ratio (including bias units)



Notes: Reconstructed images are generated by encoding and decoding a stimulus image using a trained autoencoder networks with varying hidden layers, and bias units included. Columns separate reconstructed images by compression ratio (CR). Rows separate reconstructed images by the input stimulus (uncompressed) image. The first entry in each row is the stimulus image used to generate the following reconstructions.

List of Tables

1 Average mean-squared error (MSE) for reconstructed images 14

Table 1: Average mean-squared error (MSE) for reconstructed images

Model	MSE
Compression ratio: 392	0.0621
Compression ratio: 392 (bias)	0.0624
Compression ratio: 196	0.0547
Compression ratio: 196 (bias)	0.0548
Compression ratio: 98	0.0516
Compression ratio: 98 (bias)	0.0551
Compression ratio: 56	0.0488
Compression ratio: 56 (bias)	0.0463
Compression ratio: 28	0.0395
Compression ratio: 28 (bias)	0.0446
Compression ratio: 14	0.0160
Compression ratio: 14 (bias)	0.0208
Compression ratio: 7	0.0138
Compression ratio: 7 (bias)	0.0085

Notes: Average mean squared error (MSE) across validation partition of MNIST dataset (N=10,000) for reconstructed images. Average MSE is calculated after training models for 50 epochs.