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. Now, more than ever before, individuals possess unprecedented access to vast quantities of data. However, as the volume of information available for consumption increases, so too does the demand for more efficient and cost-effective mechanisms for information exchange. Regardless of the quantity or utility of information, no one will ask for it, learn from it, or act upon it unless it can be acquired in a timely and inexpensive manner. Indeed, 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 large volumes of digital information by compressing data to representations of reduced size. Most practical applications of compression occur through lossless techniques (such as ZIP and gzip) which enable perfect reconstruction of input data from compressed formats. While lossless compression offers 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]. In contrast, lossy approaches to data compression are able to achieve exceptionally high compression ratios through the use of inexact approximations and partial data exclusion techniques. While compression achieved in this way invariably degrades data-accuracy, consequent reductions in file size significantly reduce the computational expense of storing and sending information. Moreover, well-designed approaches to lossy compression can achieve remarkably high compression-ratios before degradation is noticed by end-users. In scenarios where speed of information exchange outweighs minor degradation in data-accuracy (such as in the case of real-time multimedia communication), lossy compression becomes an attractive approach.

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 capacity of various autoencoder network architectures to compress and subsequently reconstruct high-dimensional image data of handwritten digits. To this end, the present analysis will investigate variations in reconstruction-error resulting from: (1) modifications to compression-ratios enforced by the autoencoder; and (2) the inclusion of bias nodes within 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 this paper is structured as follows. Section 2 briefly outlines the intuition behind autoencoder networks. Section 3 describes the dataset used in this analysis, and provides an overview of the general experimental framework. Section 4 presents and discusses the results. Finally, Section 5 concludes the paper.

2 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:

$$\begin{aligned} z &= E_\phi(x) && (1) \\ &= \sigma(W_E x + b_E), && (2) \end{aligned}$$

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:

$$\begin{aligned} x' &= D_\phi(z) \qquad (3)\\ &= \sigma(W_D z + b_D), \qquad (4) \end{aligned}$$

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:

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

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 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.

4 Experimental method

The following section presents an overview of the autoencoder network architectures and estimation methodology employed to evaluate the role of compression and bias hyperparameters in reconstruction accuracy of encoded images. To this end, the analysis in this note adopts a fundamentally simple experimental framework, whereby a generic autoencoder architecture is iteratively modified and reviewed for changes in reconstruction performance. This generic architecture (illustrated in Figures 3 and 4) consists of an input layer with 784 nodes; a single fully-connected hidden layer with a variable number of nodes activated by the Sigmoid function; an optional bias node; and an output layer with 784 nodes—again, activated by the Sigmoid function.

To assess the role of compression ratios in reconstruction performance, the generic autoencoder architecture is sequentially modified by varrying number of nodes in the hidden layer 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, 56x, 28x, 14x, and 7x respectively. Each architecture is modelled twice: once with the inclusion of a trainable weights to a constant bias unit, and once with bias weights set to zero. Figures 3 and 4 present diagrams of the generic autoencoder architecture with and without trainable bias weights.

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 a 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.

5 Results

Autoencoder networks generate compressed representations of data by propagating inputs through *dimension bottlenecks*, wherein high-dimensional inputs are compressed to lower-dimensional representations. The intensity of the dimension bottleneck imposed by an autoencoder is encapsulated by its *compression ratio* — that is, the ratio between the number of nodes in the input layer, and the number of nodes in the smallest hidden layer. In particular, the greater the intensity of dimension bottleneck, the greater the quantity of data that must be discarded to produce low-dimension representations, and the greater the consequent degradation in data-accuracy of reconstructed inputs. Demonstrating this notion, Figures 6 and 7 present example image reconstructions for networks of varying compression ratios. While low compression ratios (such as 7x and 14x) produce highly legible reconstructions of the original handwritten images, reconstruction accuracy diminishes drastically as compression ratios increase beyond 14x, likely as a reflection of the quantity of data that must be discarded to produce such highly compressed repreesntations.

Formally, the present analysis documents monotonically increasing reconstruction error in response to increasing compression ratios imposed by autoencoder networks (Table 1, Figure 5). In both cases of inclusion and exclusion of bias units, the most accurate reconstruction of input images was achieved by networks with 114 hidden nodes, and the least drastic compression ratios (MSE of 0.0085 and 0.0138 with and without inclusion of bias units respectively). While doubling compression ratios from 7x to 14x only resulted in minor degradation in reconstruction accuracy (MSE of 0.0160 and 0.0208 with and without inclusion of bias units respectively), compression ratios beyond 14x were associated with markedly higher reconstruction error compared to best observed performance.

Notably, however, average reconstruction error across the validation dataset masks substantial heterogeneity in model performance with respect to particular data subclasses. Figure 8 shows mean-squared error disaggregated by annotated label of instances processed by autoencoder networks. Between-class differences in MSE suggest that estimated autoencoder networks exhibit varied proficiency in reconstructing particular digits. Specifically, digits with simple, linear structures — such as 1s and 7s — are more accurately reconstructed from compressed representations by all models, in comparison to digits with complex, curved structures – such as 3s, 6s, and 8s.

Unlike the effect of compression ratios, the inclusion of bias units within network architectures elicited mixed changes in reconstruction performance. Firstly, reconstruction accuracies for networks with high compression ratios (392x and 196x) were relatively unaffected by inclusion or exclusion of bias units (Table 1, Figure 5). Secondly, while architectures with lower compression ratios exhibited noticeable differences in reconstruction accuracy when implemented with and without bias units, the direction of these differences were inconsistent across the range of estimated models. Such inconsistencies in the direction of affect resulting from imposition bias units are more likely an artefact of the stochastic nature of model training, rather than indicative of true reconstruction performance of underlying architectures.

6 Conclusion

In this paper, I evaluate the capacity of various autoencoder network architectures to compress and subsequently reconstruct high-dimensional images of handwritten digits. I assess the reconstruction accuracy of 14 distinct autoencoders architectures, each imposing a distinct combination of compression ratio and bias-unit parameterisations. The obtained results show that reconstruction accuracy of encoded images monotonically diminishes with the imposition of higher compression-ratios. Moreover, the accuracy with which autoencoder networks reproduce input images is shown to depend on the complexity of structures in the original image. Finally, the inclusion or exclusion of bias units within autoencoder network architectures is documented to inconsistently affect image reconstruction performance.

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.

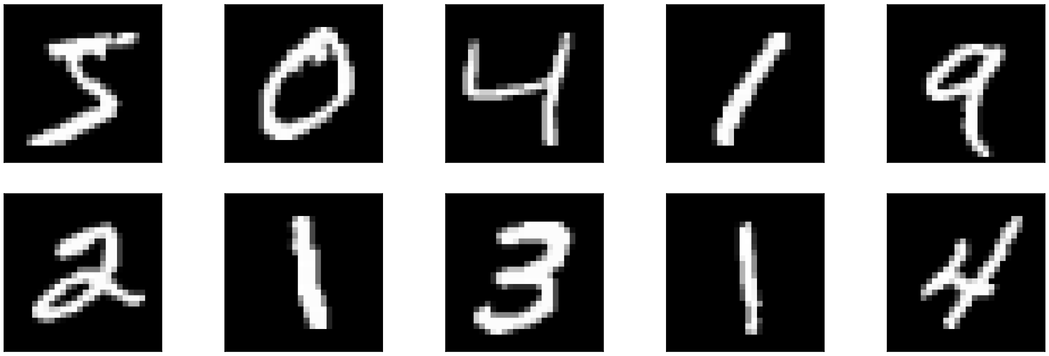
[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[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.

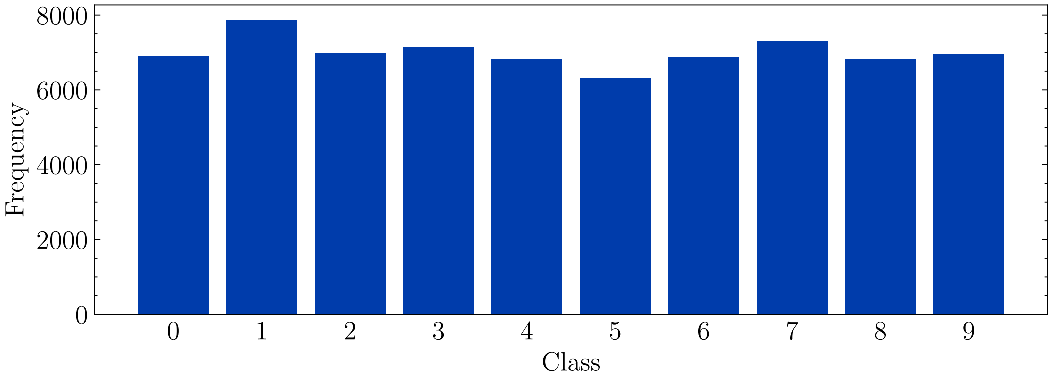
List of Figures

Figure 1: Examples of MNIST instances



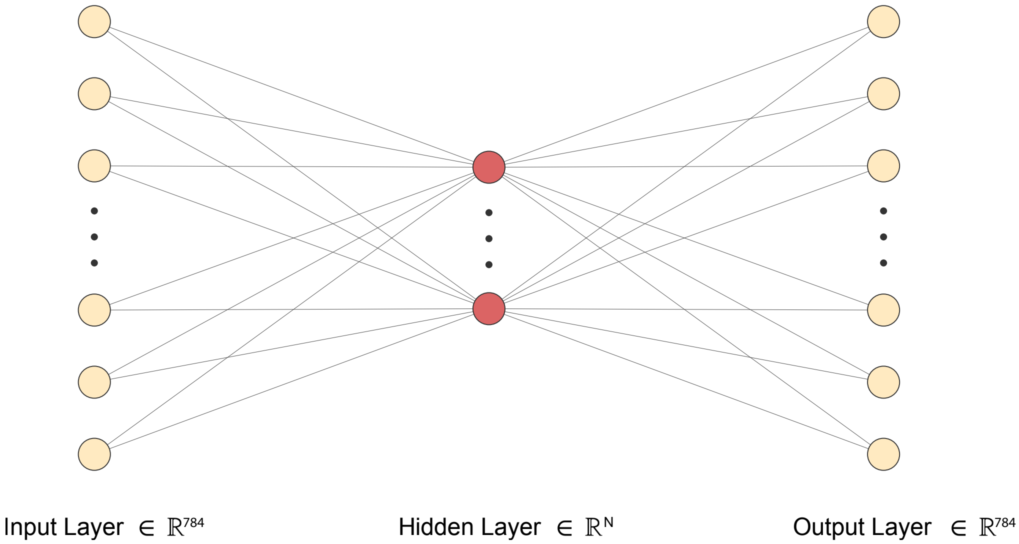
Notes: Example instances from the MNIST dataset. Each instance is normalised in grey-scale (0-255), and aligned by translating the centre of pixel-mass to be positioned at the centre of a 28x28 pixel grid.

Figure 2: Distribution of classes in MNIST dataset



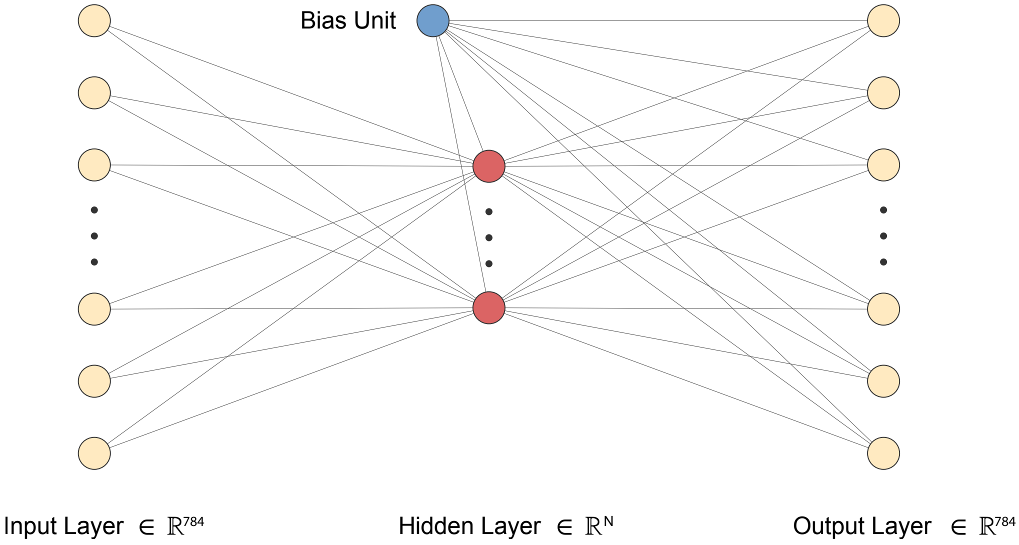
Notes: Class distribution of digit instances (0 - 9) in the full MNIST dataset, across both training (N=60,000) and validation (N=10,000) partitions.

Figure 3: Generic autoencoder architecture, without bias


Input Layer $\in \mathbb{R}^{784}$
Hidden Layer $\in \mathbb{R}^{N}$
Output Layer $\in \mathbb{R}^{784}$

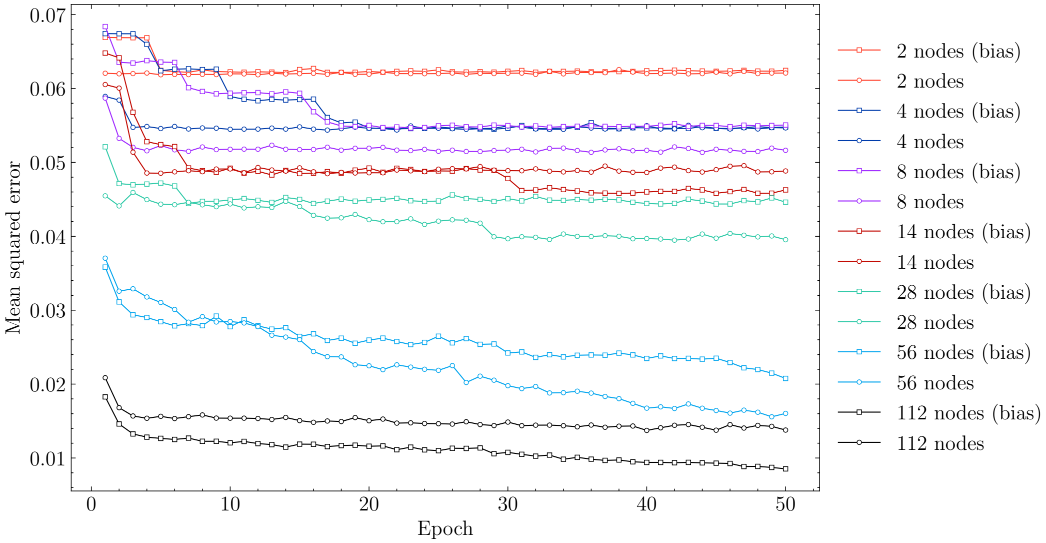
Notes: Input and output layers each contain 784 nodes respectively, in congruence with the dimensions of MNIST image instance vectors. Intermediate nodes are omitted for ease of interpretability. Hidden layers contain a variable number of nodes, N, varied across seven values: 2, 4, 8, 14, 28, 56, and 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



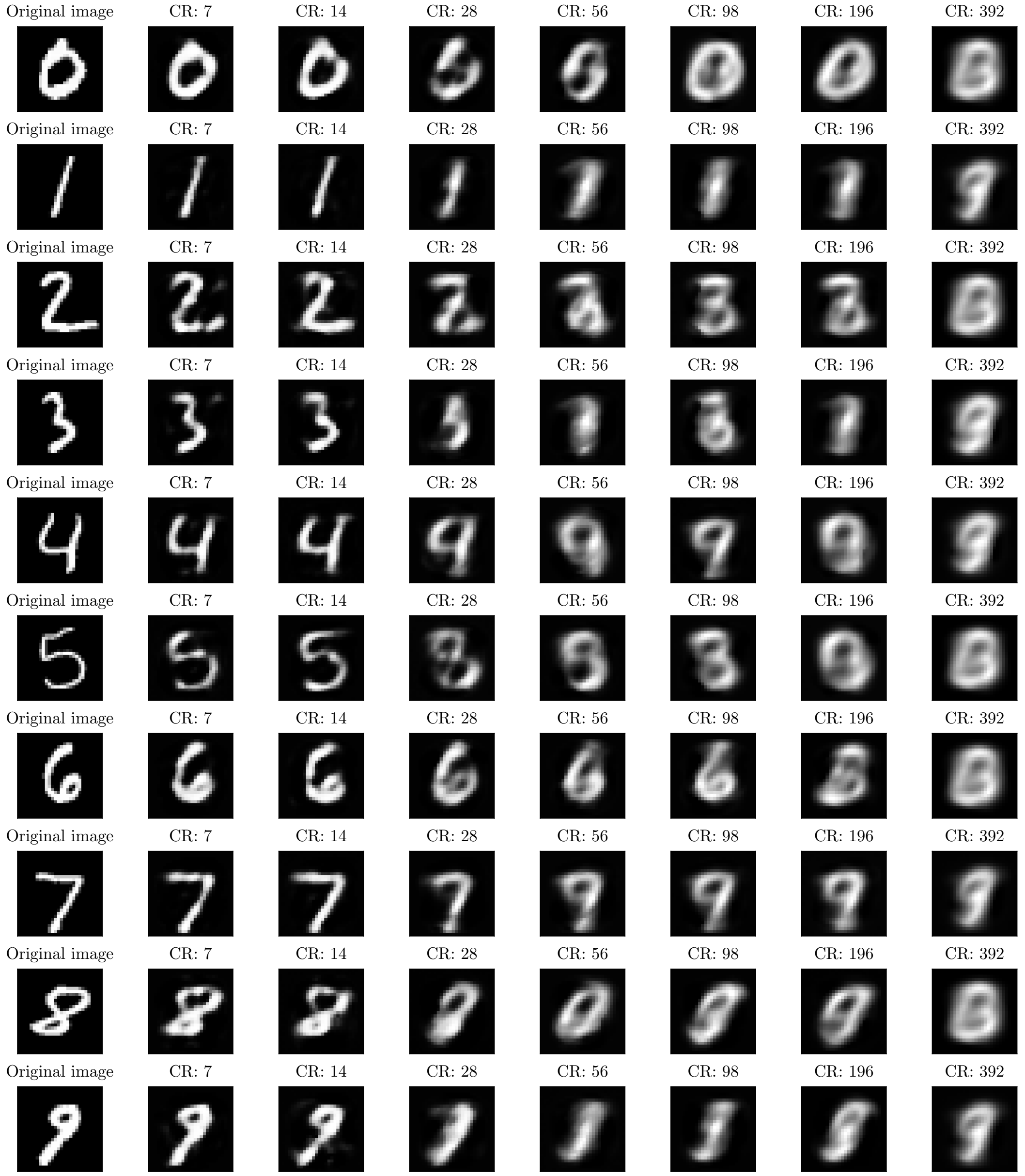
Notes: Input and output layers each contain 784 nodes respectively, in congruence with the dimensions of MNIST image instance vectors. Intermediate nodes are omitted for ease of interpretability. Hidden layers contain a variable number of nodes, N, varied across seven values: 2, 4, 8, 14, 28, 56, and 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



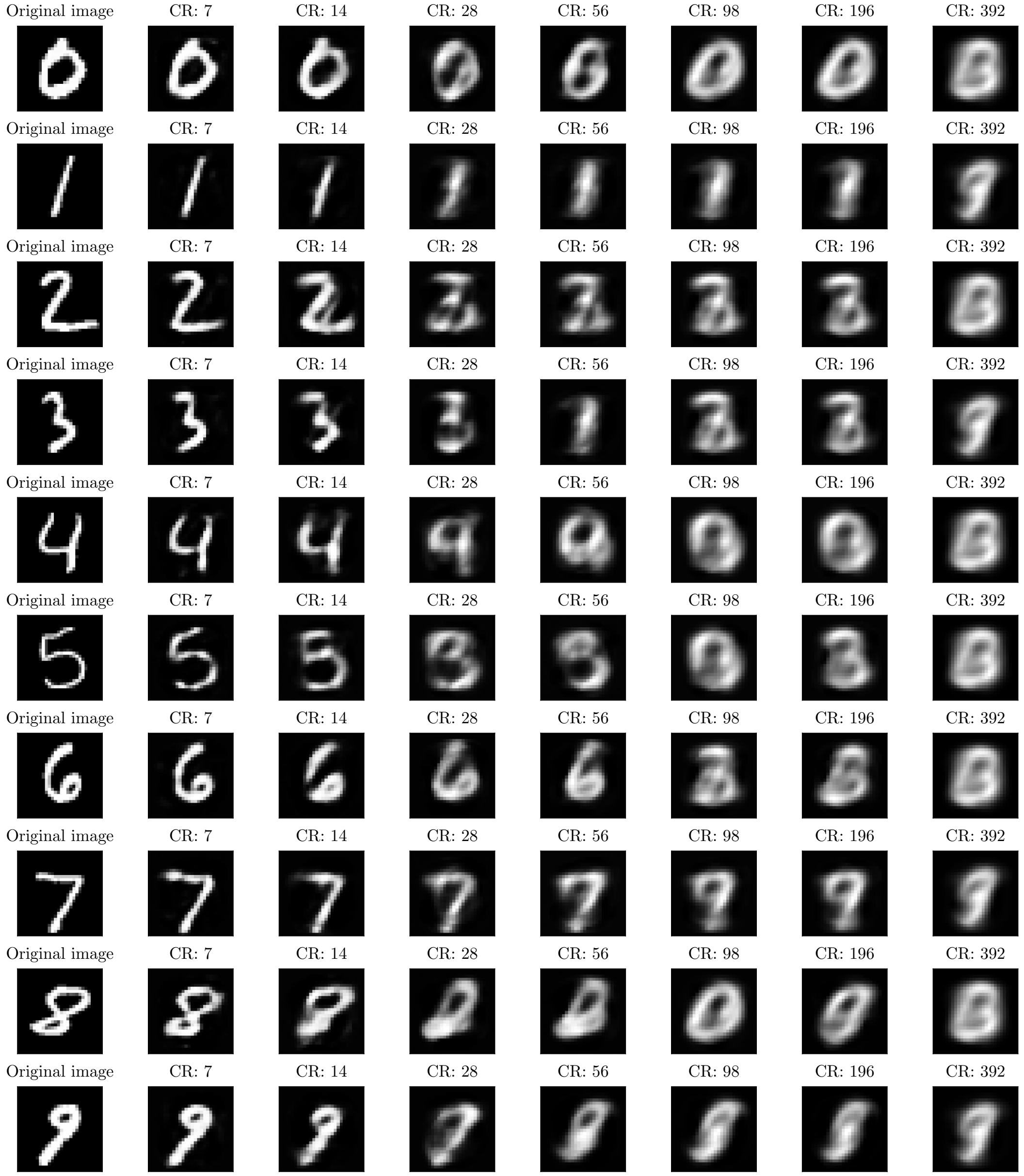
Notes: Mean-squared error (MSE) of autoencoder networks across validation partition of MNIST dataset. MSE was calculated through forward propogation of MNIST validation instances (N=10,000) using trained weights at each discrete training timestep. Learning rate for each model was set at 0.01.

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



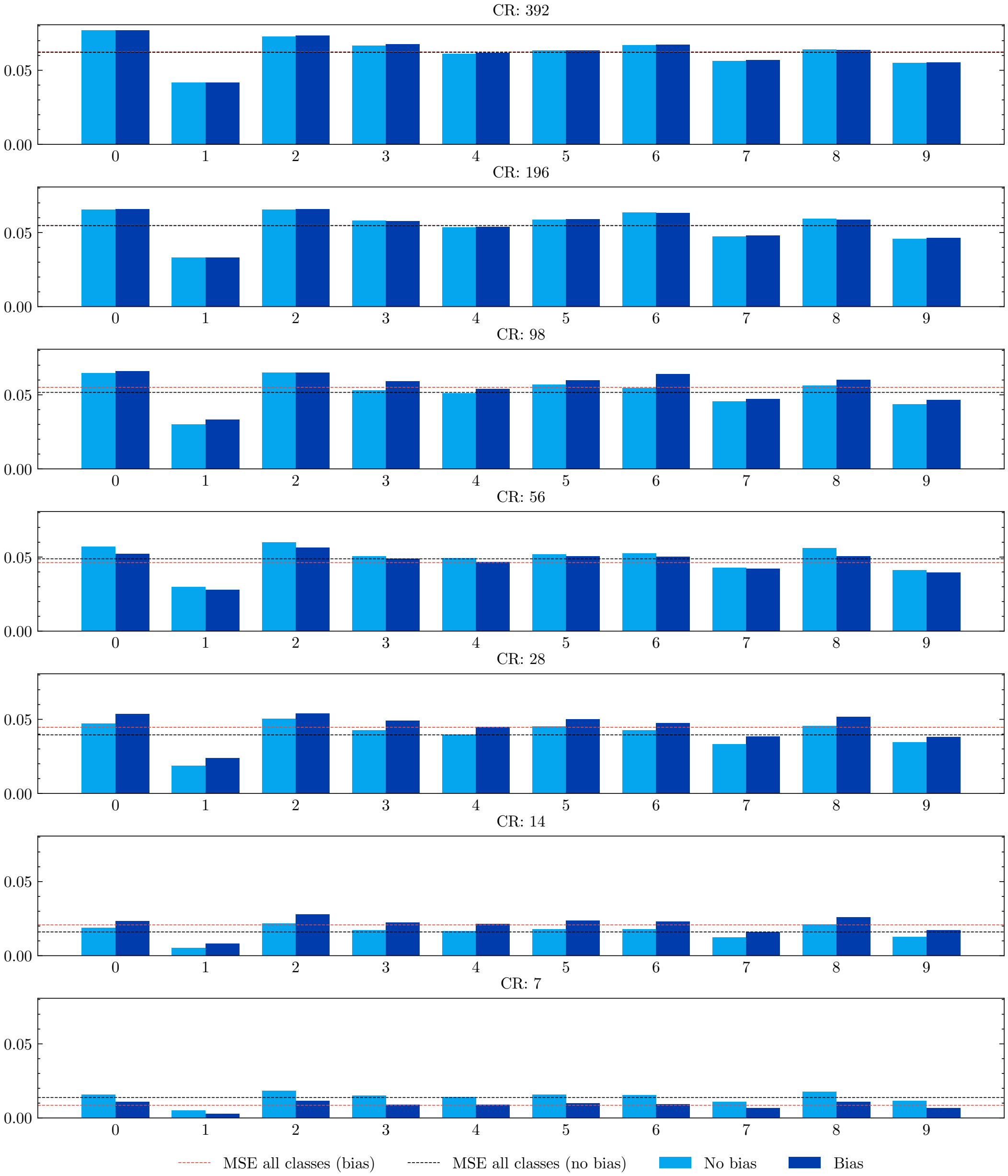
Notes: Reconstructed images are generated by encoding and decoding a stimulus image using a trained autoencoder networks with varying hidden layers, and bias set to zero. Columns separate reconstructed images by compression ratio (CR). Rows separate reconstructed images by the input stimulus (uncompressed) 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.

Figure 8: Mean-squared error (MSE) for reconstructed images disaggregated by MNIST class



Notes: Mean squared error (MSE) across validation partition of MNIST dataset (N=10,000), disaggregated by annotated class of image instance. MSE is calculated after training models for 50 epochs. Autoencoder network architectures are split by compression ratio (CR), and inclusion of bias units.

List of Tables

Table 1: 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: Mean squared error (MSE) across validation partition of MNIST dataset (N=10,000) for reconstructed images. MSE is calculated after training models for 50 epochs.