Do autoencoders need a bottleneck for anomaly detection?

Bang Xiang Yong, Alexandra Brintrup

Institute for Manufacturing, University of Cambridge, UK

Abstract

A common belief in designing deep autoencoders (AEs), a type of unsupervised neural network, is that a bottleneck is required to prevent learning the identity function. Learning the identity function renders the AEs useless for anomaly detection. In this work, we challenge this limiting belief and investigate the value of non-bottlenecked AEs.

The bottleneck can be removed in two ways: (1) overparameterising the latent layer, and (2) introducing skip connections. However, limited works have reported on the use of one of the ways. For the first time, we carry out extensive experiments covering various combinations of bottleneck removal schemes, types of AEs and datasets. In addition, we propose the infinitely-wide AEs as an extreme example of non-bottlenecked AEs.

Their improvement over the baseline implies learning the identity function is not trivial as previously assumed. Moreover, we find that non-bottlenecked architectures (highest AUROC=0.857) can outperform their bottlenecked counterparts (highest AUROC=0.696) on the popular task of CIFAR (inliers) vs SVHN (anomalies), among other tasks, shedding light on the potential of developing non-bottlenecked AEs for improving anomaly detection.

Keywords: `autoencoders, anomaly detection, bottleneck, unsupervised neural network`

1. Introduction

Numerous works have demonstrated the successful use of autoencoders (AEs), a type of unsupervised neural network (NN), for anomaly detection [1]. AEs are optimised to reconstruct a set of training data with minimal error. When given anomalous data which have high dissimilarity from the training data, the AEs reconstruct them with high error. Therefore, the reconstruction error is a measure of data anomalousness; by placing a threshold, we can effectively classify data points as inliers or anomalies.

Extant works claim that AEs will trivially learn the identity function when no constraints are placed [2, 3]. If this were to occur, AEs will perfectly reconstruct any inputs (regardless whether it is anomalous or not), and hence the reconstruction loss will be low for all inputs, leading to unreliable anomaly detection. To prevent this, it is common to impose a bottleneck in the architecture, resulting in an undercomplete architecture: the output of the encoder has much lower dimensions than the input. However, most works describe the need for a bottleneck analogically and report only the empirical performance of bottlenecked AEs, without comparing them against non-bottlenecked AEs [4, 5, 6, 7, 2].

Why should we care about non-bottlenecked AEs? By limiting to bottlenecked architectures, we miss the potential of achieving better performance with non-bottlenecked AEs. Therefore, in this work, we study the use of non-bottlenecked AEs for anomaly detection. We investigate combinations of ways for removing the bottleneck, including (1) expanding the latent dimensions and (2) introducing skip connections. Furthermore, we propose the infinitely-wide AEs as an extreme example. Extensive experiments demonstrate the empirical success of non-bottlenecked AEs in detecting anomalies over the baseline and the bottlenecked AEs, indicating the non-bottlenecked AEs have failed to learn the identity function, contrary to conventional belief.

We suggest that rethinking about AEs is needed. In this effort, we adopt the probabilistic formulation of Bayesian autoencoders (BAEs), viewing them as regularised density estimators that benefit from having higher expressivity allowed by non-bottlenecked architectures (see Fig. 1 for an example). The Bayesian framework also provides a sound foundation for theoretical analysis of these architectures in future work.

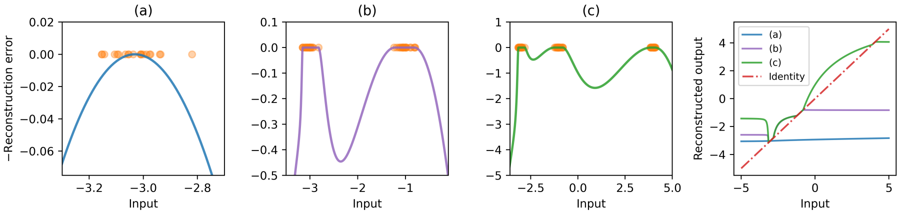


Fig. 1: (a-c) Negative reconstruction errors (log-likelihood) from BAEs with five layers of infinitely many parameters on 1D toy datasets, resembling reasonable density estimation. Orange dots represent the training data points. (d) The reconstructed outputs (last panel) clearly differ from the identity function. All layers use GELU [8] as activation functions, except the last, which uses the sigmoid function; min-max scaler [9] is used.

This paper is organised as follows: Section 2 formulates AEs from a Bayesian perspective and describes ways to remove the bottleneck. Our experimental setup is described in Section 3 followed by results and discussion in Section 4. We relate to previous works in Section 5 and state our limitations in Section 6. We close with a summary and future directions in Section 7.

2. Methods

2.1. Bayesian autoencoders

Suppose we have a set of data $X^{train} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ...\mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^{\mathrm{D}}$. An AE is an NN parameterised by θ, and consists of two parts: an encoder for mapping input data $\mathbf{x}$ to a latent embedding, $\mathbf{z} = f_{\text{encoder}}(\mathbf{x})$, and a decoder f_{decoder} for mapping the latent embedding to a reconstructed signal of the input $\hat{\mathbf{x}}$ (i.e. $\hat{\mathbf{x}} = f_\theta(\mathbf{x}) = f_{\text{decoder}}(f_{\text{encoder}}(\mathbf{x}))$) [10].

Bayes' rule can be applied to the parameters of the AE to create a BAE,

$$p(\theta|X^{train}) = \frac{p(X^{train}|\theta)\, p(\theta)}{p(X^{train})}, \tag{1}$$

where $p(X^{train}|\theta)$ is the likelihood and $p(\theta)$ is the prior distribution of the AE parameters. The log-likelihood for a diagonal Gaussian distribution is,

$$\log p(\mathbf{x}|\theta) = -(\frac{1}{D}\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i - \hat{x_i})^2 + \frac{1}{2}\log\sigma_i^2) \tag{2}$$

where σ_i^2 is the variance of the Gaussian distribution. For simplicity, we use an isotropic Gaussian likelihood with $\sigma_i^2 = 1$ in this study, since the negative log-likelihood (NLL) is proportional to the mean-squared error (MSE) function. We employ an isotropic Gaussian prior distribution, effectively leading to L_2 regularisation.

Since Equation 1 is analytically intractable for a deep NN, various approximate methods have been developed such as Stochastic Gradient Markov Chain Monte Carlo (SGHMC) [11], Monte Carlo Dropout (MCD) [12], Bayes by Backprop (BBB) [13], and anchored ensembling [14] to sample from the posterior distribution. In contrast, a deterministic AE has its parameters estimated using maximum likelihood estimation (MLE) or maximum a posteriori (MAP) when regularisation is introduced. The variational autoencoder (VAE) [15] and BAE are AEs formulated differently within a probabilistic framework: in the VAE, only the latent embedding is stochastic while the f_{encoder} and f_{decoder} are deterministic and the model is trained using variational inference; on the other hand, the BAE, as an unsupervised Bayesian neural network (BNN), has distributions over all parameters of f_{encoder} and f_{decoder}. In short, the training phase of BAE entails using one of the sampling methods to obtain a set of approximate posterior samples $\{\hat{\theta}_m\}_{m=1}^{M}$.

Then, during the prediction phase, we use the posterior samples to compute M estimates of the NLL. The predictive density of a new data point $\mathbf{x}^*$ can be approximated as the mean of the posterior NLL estimates,

$$p(\mathbf{x}^*|X^{train}) = \mathbb{E}_{\theta}[-\log p(\mathbf{x}^*|\theta)\, p(\theta|X^{train})] \approx -\frac{1}{M}\sum_{m=1}^{M} p(\mathbf{x}^*|\hat{\theta}_m) \tag{3}$$

For convenience, we denote $p(\mathbf{x}^*|X^{train})$ as $\mathbb{E}_{\theta}$ [NLL]. The Bayesian formulation allows us to view AEs as regularised probability density estimators: they model the training data distribution, assigning lower density scores to data which have higher dissimilarity from the training data.

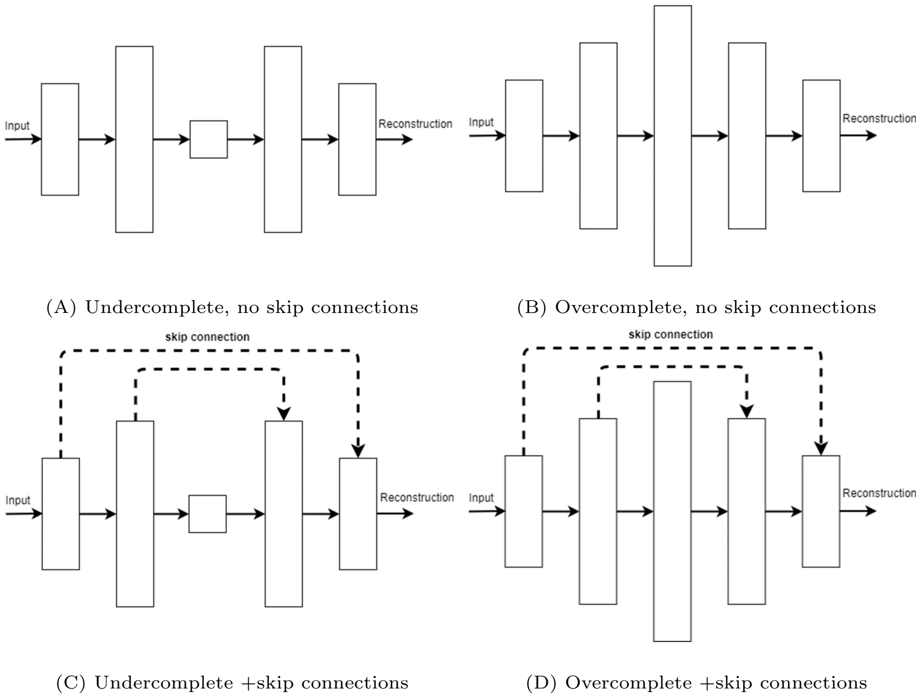
2.2. How to remove the bottleneck?

The identity function is successfully learnt when $f_\theta(\mathbf{x}) = \mathbf{x}$ holds true for all $\mathbf{x}$ and therefore the reconstruction loss or NLL is always 0, rendering it useless for distinguishing anomalies from inliers. In an effort to mitigate this, a bottleneck is implemented at the latent layer (encoder's final layer) by having the latent dimensions smaller than the input dimensions, $\dim(\mathbf{z}) < \dim(\mathbf{x})$, and there is no way for any output of the intermediate layers to bypass the bottleneck layer. It is straightforward to eliminate the bottleneck by doing the opposite: (1) simply expand the size of the latent dimensions to $\dim(\mathbf{z}) \geq \dim(\mathbf{x})$, also known as an overcomplete architecture, and/or (2) introduce long-range skip connections from the encoder to the decoder akin to a U-Net architecture [16], thereby allowing each layer's data flow to bypass the bottleneck; for clarity, see Table 1 and Fig. 2.

Table 1: Categorising architectures into with or without a bottleneck depends on the latent dimensions and the presence of skip connections.

Architecture type	Latent dimensions	Skip connections
Bottlenecked		
A	Undercomplete	✗
Non-bottlenecked		
B	Undercomplete	✓
C	Overcomplete	✗
D	Overcomplete	✓

Why skip connections? Skip connections allow a better flow of information in NNs with many layers, leading to a smoother loss landscape [17] and easier optimisation, without additional computational complexity [18]. Recent works [19, 20, 21] have reported that AEs with skip connections outperform those without on image anomaly detection. In preventing the skip-AEs from



(A) Undercomplete, no skip connections

(B) Overcomplete, no skip connections

(C) Undercomplete +skip connections

(D) Overcomplete +skip connections

Fig. 2: Architectures of bottlenecked (type A) and non-bottlenecked (type B, C and D) AEs. Each block represents an NN layer. Its width indicates the relative number of parameters.

learning the identity function, Collin et al. [20] and Baur et al. [19] have implemented a denoising scheme and a dropout mechanism, respectively. Notably, Baur et al. [19] have reported that random weight initialisation alone is sufficient to prevent learning the identity function, rendering the dropout redundant.

Infinitely-wide BAE. In the infinite-width limit, a fully-connected BNN is equivalent to a neural network Gaussian process (NNGP) [22]. The results have been extended to modern architectures such as convolutional NNs, recurrent NNs, and transformers [23, 24, 25] in recent years. We propose extending the NNGP to the AE to create an infinitely-wide BAE (BAE-∞), which opposes the conventional bottleneck design. Viewing the BAE as a density estimator motivates this; it is not unconventional for density estimators to have infinite parameters as they benefit from higher expressivity to model an arbitrary distribution well [26, 27]. There are two primary advantages of the NNGP: having

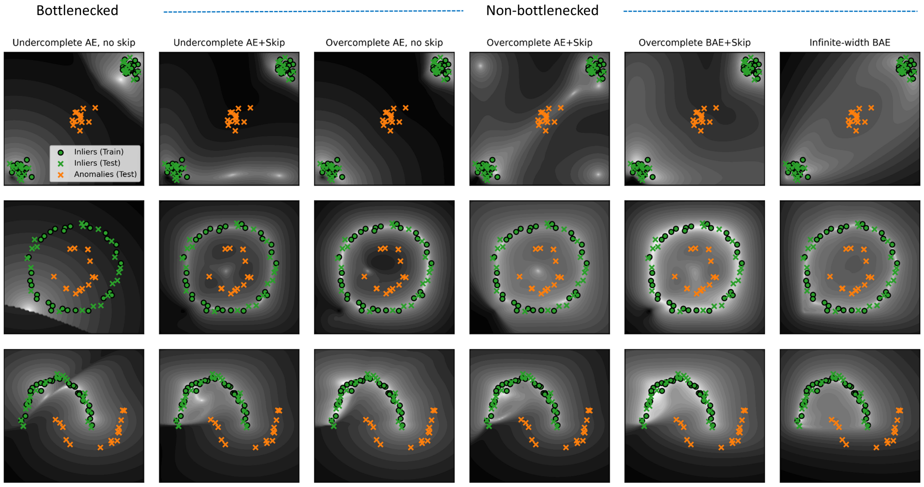


Fig. 3: $\mathbb{E}_{\theta}$ [NLL] using deterministic AE and BAEs with bottlenecked and non-bottlenecked architectures. Brighter region has lower $\mathbb{E}_{\theta}$ [NLL] values and log-scale of contour is used to increase visibility. The encoder architecture has fully-connected layers with nodes of 2-50-50-50-dim($\mathbf{z}$) where dim($\mathbf{z}$)=1 for undercomplete and dim($\mathbf{z}$)=100 for overcomplete architectures. We use SELU activation [28] for every layer and sigmoid activation for the final layer. The BAE-∞ has a similar number of layers with infinite parameters. The darker contours away from the training points show that the deterministic AE and BAE do not learn the identity function despite being overparameterised and having skip connections.

a closed-form solution and modelling a BNN with infinitely many parameters. The first facilitates a theoretical understanding by linking to the well-studied GP model, and the second *potentially* improves performance since deep NNs succeed over traditional ML models via increasing model parameters [1]. Nonetheless, empirically, infinite NNs do not always outperform finite NNs; reasons for their underperformance remain an active research topic [29, 30]. Another drawback is their computational complexity of $\mathcal{O}(N^3)$, where N is the number of training examples, reducing scalability to large datasets.

Surprisingly, when we examine the behaviours of AEs on 2D toy data sets (Fig. 3), we find that the identity function is not learnt despite using various types of non-bottlenecked AEs. Consequently, this observation on low-dimensional data implies it is more unlikely to learn the identity function on high-dimensional data due to higher degrees of freedom.

We suggest several reasons hindering AEs from the identity mapping: high degree of non-linearity in the AE and regularisation induced by mini-batching, the deep learning optimiser (e.g. Adam [31]) and the prior over parameters. Since these are usually implicit in training the AE, no additional, explicit efforts are necessary (e.g. denoising or dropout mechanisms).

3. Experimental Setup

The full code for reproducing results will be released upon the acceptance of this paper.

3.1. Datasets

Several publicly available datasets are included in our experiments. For image data, we use pairs of unrelated datasets (inliers vs anomalies) commonly used in previous works: FashionMNIST [32] vs MNIST [33], and CIFAR [34] vs SVHN [35]. For tabular data, we use eight datasets from the Outlier Detection Datasets (ODDS) collection [36]: Cardio, Lympho, Optdigits, Ionosphere, Pendigits, Thyroid, Vowels and Pima. For sensors data, we use the ZeMA [37] and STRATH [38] datasets gathered from industrial environments.

Table 2: Sensors used for each task in ZeMA and STRATH datasets.

		ZeMA	STRATH
Tasks	Target subsystem	Sensors	Sensors
(i)	Cooler	Temperature (TS4)	Position (L-ACTpos)
(ii)	Valve	Temperature (TS4)	Speed (A-ACTspd)
(iii)	Pump	Pressure (PS6)	Servo (Feedback-SPA)
(iv)	Accumulator	Temperature (TS4)	Sensors from tasks (i-iii)

In the ZeMA dataset, the tasks are to detect deterioration in the subsystems of a hydraulic test rig (see Table 2 for the sensor-subsystem pairs). We consider the data from the healthiest state of the subsystems as inliers and the rest as anomalies. In the STRATH dataset, different sensors are used in each task to detect defective parts manufactured from a radial forging process consisting of heating and forging phases. Geometric measurements of each forged part are available as target quality indicators. To label the anomalies, we focus our analysis on the *38 diameter@200* by applying the Tukey's fences method [39] on the absolute difference between the measured and nominal dimensions.

3.2. Preprocessing

For image data, we use the default split of train-test sets and rescale pixel values to [0,1]. For the ODDS, ZeMA and STRATH datasets, we split the inliers into train-test sets of 70:30 ratio with random shuffling, and include all anomalies in the test set. We apply min-max scaling [9] with care to prevent train-test bias by fitting the scaler to the train set instead of the entire inlier set. For ZeMA, we downsample the pressure sensor to 1Hz and use the temperature data as provided; for STRATH, we downsample the data by tenfold and segment only the forging phase.

3.3. Models

We train variants of AEs: deterministic AE, VAE, and BAEs with several inference methods: MCD, BBB and anchored ensembling. All models use isotropic Gaussian priors over weights. The number of posterior samples is set to $M = 100$ for the VAE, BAE-MCD and BAE-BBB, while $M = 10$ for the BAE-Ensemble. We set a fixed learning rate of 0.001 for STRATH; the learning rates for the other datasets are searched with a learning rate finder, employing a cyclic learning rate [40]. The Adam optimiser is used and the training epochs for FashionMNIST, CIFAR, ODDS, ZeMA and STRATH are 20, 20, 300, 100, 100, respectively. The weight decay is set to 1×10^{-11} for FashionMNIST and CIFAR, and 1×10^{-10} for ODDS, ZeMA and STRATH.

Table 3: Encoder architecture of finite-width AEs. The decoder is a reflection of the encoder, in which the Conv1D and Conv2D layers are replaced by Conv1D- and Conv2D-Transpose layers. The leaky ReLu [41] is used as the activation function with a slope of 0.01 while the sigmoid function is used at the decoder's final layer.

(a) FashionMNIST and CIFAR

Layer	Output channels/nodes	Kernel	Strides
Conv2D	10	2 x 2	2 x 2
Conv2D	32	2 x 2	1 x 1
Reshape	-	-	-
Dense	100	-	-
Dense	Latent dimensions	-	-

(b) ODDS

Layer	Output channels/nodes	Kernel	Strides
Linear	Input dimensions ×4	-	-
Linear	Input dimensions ×4	-	-
Linear	Latent dimensions	-	-

(c) ZeMA and STRATH

Layer	Output channels/nodes	Kernel	Strides
Conv1D	10	8	2
Conv1D	20	2	2
Reshape	-	-	-
Linear	1000	-	-
Linear	Latent dimensions	-	-

The architectures of non-finite AEs are described in Table 3. All architectures, except for ZeMA, apply layer normalisation [42] before the activation function. Leaky ReLu [41] is used as the intermediate layers' activation function while the sigmoid function is used at the final output layer. The bias terms are turned off for each layer. The size of latent dimensions is set to the flattened input dimensions multiplied by factors of $\times\frac{1}{10}, \times\frac{1}{2}, \times 1$, including $\times 2$ (for FashionMNIST and CIFAR) and $\times 10$ (for ODDS, ZeMA and STRATH). For the BAE-∞ , we implement the NNGP with seven infinitely-wide dense layers (including the encoder and decoder) using the Neural Tangent Kernel library [43] for all datasets. During testing, we evaluate the area under the receiver-operating characteristic curve (AUROC) [44] scores of the $\mathbb{E}_\theta$ [NLL] on the test set consisting of inliers and anomalies.

4. Results and discussion

In Table 4, most non-bottlenecked models (type B, C and D) beat the baseline with mean AUROC ≥ 0.8. This observation indicates the identity function has not been learnt despite being overparameterised and having skip connections. Also, their positive average treatment effect (ATE) improves over the bottlenecked models (type A) with type D models showing the highest ATE. In addition, the best mean AUROC scores on most datasets have been achieved by the non-bottlenecked models, except on FashionMNIST vs MNIST, which has the bottlenecked model (BAE-BBB, type A) marginally beating the second-best model (BAE-BBB, type C) by 0.1 AUROC.

Focusing on CIFAR vs SVHN, our results provide new insights into previous works which reported poor performances [45, 46]. Notably, the best non-bottlenecked model (BAE-Ensemble, type B, AUROC=0.849) and the BAE-∞ (AUROC=0.771) outperform the best bottlenecked model (BAE-Ensemble, type A, AUROC=0.696). These results imply the poor performance could be fixed if previous works were to consider non-bottlenecked architectures.

Switching from a deterministic AE to a BAE improves performance as the best performing BAEs achieve the highest AUROC scores on all datasets. The performance gain is attributed to Bayesian model averaging [47], which accounts for uncertainty in model parameters. The best BAEs also outperform the VAEs, evidencing the advantage of addressing the uncertainty over parameters of the entire model instead of considering only the latent layer.

Although the BAE-∞ does not score the highest AUROC, on a positive note, there are specific tasks on which the BAE-∞ outperforms other models with the highest median AUROC (e.g. see Fig. 4 on Cardio, Thyroid, Pendigits and ZeMA(iii)) and with low variability in performance. However, the gain over the finite-width BAEs is not demonstrated on some tasks of ZEMA and STRATH.

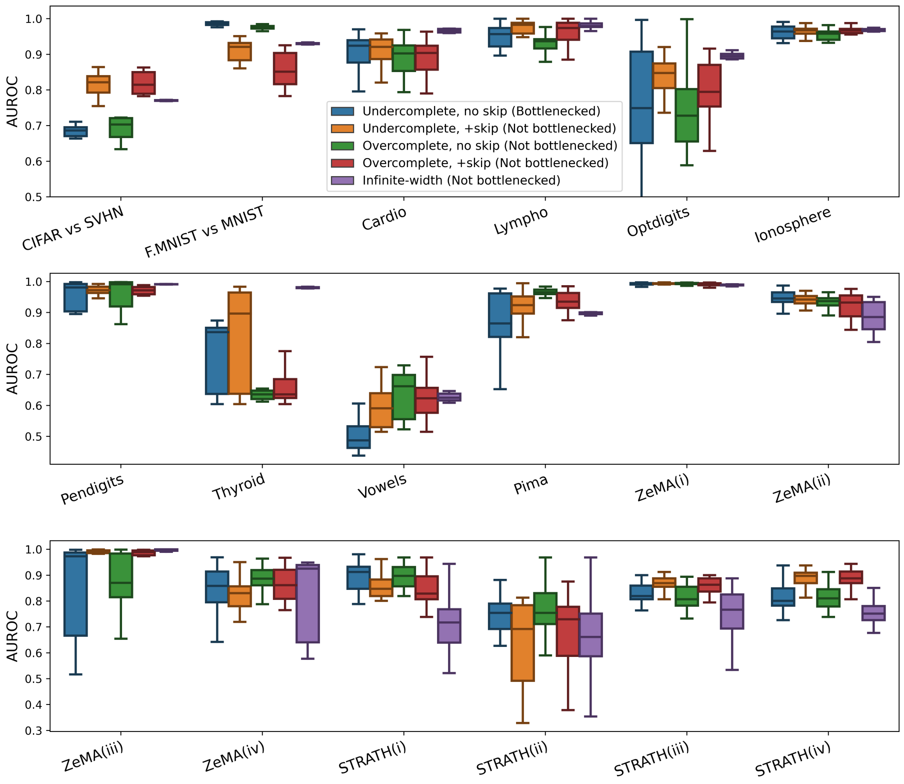


Fig. 4: AUROC scores of bottlenecked and non-bottlenecked BAEs on various datasets. Results are shown for the finite-width BAE-Ensemble and the infinite-width BAE.

Table 4: Mean ± standard error AUROC scores for deterministic AEs, VAEs and BAEs with bottlenecked and non-bottlenecked architectures. Model with highest mean AUROC is bolded for each dataset.

	—Bottlenecked—	———Not bottlenecked———		
	A	B	C	D
Model	Undercomplete	Undercomplete	Overcomplete	Overcomplete
	No skip	+ skip	No skip	+ skip
CIFAR vs SVHN, four runs				
Deterministic AE	0.686 ± 0.006	0.816 ± 0.013	0.692 ± 0.012	0.820 ± 0.011
VAE	0.398 ± 0.006	0.822 ± 0.012	0.428 ± 0.004	0.840 ± 0.008
BAE-MCD	0.613 ± 0.012	0.809 ± 0.014	0.626 ± 0.012	0.834 ± 0.004
BAE-BBB	0.672 ± 0.006	0.831 ± 0.016	0.639 ± 0.007	0.826 ± 0.006
BAE-Ensemble	0.696 ± 0.009	**0.849 ± 0.005**	0.699 ± 0.008	0.838 ± 0.005
BAE-∞	-	-	0.771 ± 0.001	-
FashionMNIST vs MNIST, four runs				
Deterministic AE	0.986 ± 0.002	0.912 ± 0.012	0.976 ± 0.002	0.858 ± 0.019
VAE	0.793 ± 0.02	0.895 ± 0.014	0.913 ± 0.008	0.884 ± 0.021
BAE-MCD	0.983 ± 0.001	0.712 ± 0.033	0.984 ± 0.001	0.783 ± 0.048
BAE-BBB	**0.992 ± 0.001**	0.848 ± 0.027	0.99 ± 0.002	0.901 ± 0.009
BAE-Ensemble	0.985 ± 0.003	0.920 ± 0.006	0.977 ± 0.001	0.928 ± 0.009
BAE-∞	-	-	0.930 ± 0.002	-
ODDS, eight datasets, ten runs				
Deterministic AE	0.819 ± 0.014	0.872 ± 0.011	0.833 ± 0.013	0.848 ± 0.012
VAE	0.795 ± 0.014	0.876 ± 0.008	0.819 ± 0.013	0.883 ± 0.008
BAE-MCD	0.829 ± 0.015	0.891 ± 0.010	0.819 ± 0.013	0.856 ± 0.011
BAE-BBB	0.892 ± 0.011	0.913 ± 0.007	**0.919 ± 0.007**	0.917 ± 0.007
BAE-Ensemble	0.857 ± 0.012	0.886 ± 0.010	0.850 ± 0.011	0.86 ± 0.012
BAE-∞	-	-	0.913 ± 0.013	-
ZeMA, four tasks, ten runs				
Deterministic AE	0.879 ± 0.024	0.930 ± 0.01	0.911 ± 0.014	0.908 ± 0.019
VAE	0.879 ± 0.015	0.917 ± 0.016	0.876 ± 0.019	0.922 ± 0.016
BAE-MCD	0.892 ± 0.017	0.920 ± 0.015	0.862 ± 0.023	0.936 ± 0.008
BAE-BBB	0.888 ± 0.019	0.939 ± 0.007	0.895 ± 0.019	0.931 ± 0.009
BAE-Ensemble	0.938 ± 0.007	0.961 ± 0.005	0.928 ± 0.007	**0.963 ± 0.004**
BAE-∞	-	-	0.926 ± 0.01	-
STRATH, four tasks, ten runs				
Deterministic AE	0.817 ± 0.010	0.811 ± 0.015	0.819 ± 0.01	0.819 ± 0.012
VAE	0.825 ± 0.009	0.831 ± 0.012	0.831 ± 0.008	0.824 ± 0.014
BAE-MCD	0.838 ± 0.008	0.808 ± 0.016	0.838 ± 0.007	0.822 ± 0.014
BAE-BBB	0.847 ± 0.008	0.834 ± 0.007	0.848 ± 0.008	0.833 ± 0.008
BAE-Ensemble	0.839 ± 0.008	0.850 ± 0.007	0.835 ± 0.007	**0.855 ± 0.006**
BAE-∞	-	-	0.717 ± 0.009	-
Mean	0.825 ± 0.01	0.866 ± 0.012	0.835 ± 0.009	**0.867 ± 0.012**
ATE	-	0.041	0.010	**0.042**

5. Related work

Several works have investigated the use of skip connections in AEs for tasks such as image denoising [48, 49] and audio separation [50]. Our work differs from current works on skip-AEs for anomaly detection [21, 19, 20]: we have investigated a wider range of non-bottlenecked AEs, in which skip-AEs are only one type, and experimented with more datasets.

Snoek et al. [51] has proposed the autoencoder with an infinitely-wide decoder while keeping its encoder finite, and demonstrates its effectiveness for supervised classification and learning latent representations. Nguyen et al. [52] has theoretically studied infinitely-wide and shallow (two layers) AEs, providing insights into their behaviours. To the best of our knowledge, we are the first to propose a deep (seven layers) BAE-∞ with all layers being infinitely-wide, and provide empirical results on anomaly detection.

Radhakrishnan et al. and Zhang et al. [53, 54] have observed that overcomplete AEs exhibit *memorisation*, a phenomenon where the AEs reconstruct the closest training examples instead of the inputs. We suggest that a possible link exists between memorisation and the success of detecting anomalies using overcomplete AEs: when given an anomalous input, the AEs reconstruct the closest training example of inliers. This leads to a more discriminating, larger reconstruction error with the anomalous input than if the input were to be an inlier.

6. Limitations

Our study has focused on unsupervised anomaly detection and implies nothing about other use cases (e.g. clustering and dimensionality reduction), for which a bottleneck is necessary. Our experiments have covered various data types, however, there may exist datasets where learning the identity function is trivial for the AE. While we lack theoretical proof that non-bottlenecked AEs never learn the identity function, the contrary is true; there is no proof, to the best of our knowledge, that they always learn the identity function.

7. Conclusion

With visualisations on low-dimensional toy data and extensive experiments covering high-dimensional datasets for anomaly detection, we find that non-bottlenecked AEs (including the BAE-∞) can perform reasonably well over the baseline. The major implications of our work are (1) learning the identity function is not as trivial as previously assumed and (2) modellers should not restrict to only bottlenecked architectures since non-bottlenecked architectures can perform better.

In light of the potential of non-bottlenecked AEs, future work should develop more variants. The closed-form solutions of BAE-∞ can facilitate theoretical work on understanding and proving the conditions for not learning the identity function. Possible directions include understanding the connection between BAEs as predictive density models and kernel density estimation [55, 56].

References

[1] G. Pang, C. Shen, L. Cao, A. V. D. Hengel, Deep learning for anomaly detection: A review, ACM Computing Surveys 54 (2) (mar 2021). `doi:10.1145/3439950`.
URL `https://doi.org/10.1145/3439950`

[2] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 4393–4402.
URL `https://proceedings.mlr.press/v80/ruff18a.html`

[3] A. Ng, et al., Sparse autoencoder, CS294A Lecture notes 72 (2011) (2011) 1–19.

[4] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, Autoencoder-based network

anomaly detection, in: 2018 Wireless Telecommunications Symposium (WTS), IEEE, 2018, pp. 1–5.

[5] J. Chow, Z. Su, J. Wu, P. Tan, X. Mao, Y. Wang, Anomaly detection of defects on concrete structures with the convolutional autoencoder, Advanced Engineering Informatics 45 (2020) 101105. `doi:https://doi.org/10.1016/j.aei.2020.101105`.
URL `https://www.sciencedirect.com/science/article/pii/S1474034620300744`

[6] S. Kim, W. Jo, T. Shon, Apad: Autoencoder-based payload anomaly detection for industrial ioe, Applied Soft Computing 88 (2020) 106017. `doi:https://doi.org/10.1016/j.asoc.2019.106017`.
URL `https://www.sciencedirect.com/science/article/pii/S1568494619307999`

[7] A. Mujeeb, W. Dai, M. Erdt, A. Sourin, One class based feature learning approach for defect detection using deep autoencoders, Advanced Engineering Informatics 42 (2019) 100933. `doi:https://doi.org/10.1016/j.aei.2019.100933`.
URL `https://www.sciencedirect.com/science/article/pii/S1474034619301259`

[8] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), arXiv preprint arXiv:1606.08415 (2016).

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[10] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[11] T. Chen, E. Fox, C. Guestrin, Stochastic gradient hamiltonian monte carlo, in: International Conference on machine learning, 2014, pp. 1683–1691.

[12] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, 2016, pp. 1050–1059.

[13] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: International Conference on Machine Learning, PMLR, 2015, pp. 1613–1622.

[14] T. Pearce, F. Leibfried, A. Brintrup, Uncertainty in neural networks: Approximately bayesian ensembling, in: International conference on artificial intelligence and statistics, PMLR, 2020, pp. 234–244.

[15] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[16] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[17] H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the loss landscape of neural nets, arXiv preprint arXiv:1712.09913 (2017).

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[19] C. Baur, B. Wiestler, S. Albarqouni, N. Navab, Bayesian skip-autoencoders for unsupervised hyperintense anomaly detection in high resolution brain mri, in: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), IEEE, 2020, pp. 1905–1909.

[20] A.-S. Collin, C. De Vleeschouwer, Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise, in: 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 7915–7922. doi:10.1109/ICPR48806.2021.9412842.

[21] J. Kim, J. Ko, H. Choi, H. Kim, Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder, Sensors 21 (15) (2021). doi:10.3390/s21154968.
URL https://www.mdpi.com/1424-8220/21/15/4968

[22] R. M. Neal, Priors for infinite networks, in: Bayesian Learning for Neural Networks, Springer, 1996, pp. 29–53.

[23] R. Novak, L. Xiao, J. Lee, Y. Bahri, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, J. Sohl-Dickstein, Bayesian deep convolutional networks with many channels are gaussian processes, arXiv preprint arXiv:1810.05148 (2018).

[24] G. Yang, Wide feedforward or recurrent neural networks of any architecture are gaussian processes, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.

[25] J. Hron, Y. Bahri, J. Sohl-Dickstein, R. Novak, Infinite attention: Nngp and ntk for deep attention networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 4376–4386.

[26] T. Chen, J. Morris, E. Martin, Probability density estimation via an infinite gaussian mixture model: application to statistical process monitoring, Journal of the Royal Statistical Society: Series C (Applied Statistics) 55 (5) (2006) 699–715.

[27] B. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, R. Kumar,

Density estimation in infinite dimensional exponential families, Journal of Machine Learning Research 18 (2017).

[28] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, in: Proceedings of the 31st international conference on neural information processing systems, 2017, pp. 972–981.

[29] L. Aitchison, Why bigger is not always better: on finite and infinite neural networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 156–164.

[30] J. Lee, S. Schoenholz, J. Pennington, B. Adlam, L. Xiao, R. Novak, J. Sohl-Dickstein, Finite versus infinite neural networks: an empirical study, Advances in Neural Information Processing Systems 33 (2020).

[31] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[32] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).

[33] L. Deng, The mnist database of handwritten digit images for machine learning research, IEEE Signal Processing Magazine 29 (6) (2012) 141–142.

[34] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

[35] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 2011.

[36] S. Rayana, ODDS library (2016).
URL http://odds.cs.stonybrook.edu

[37] T. Schneider, S. Klein, M. Bastuck, Condition monitoring of hydraulic systems Data Set at ZeMA (Apr. 2018). doi:10.5281/zenodo.1323611. URL https://doi.org/10.5281/zenodo.1323611

[38] C. Tachtatzis, G. Gourlay, I. Andonovic, O. Panni, Sensor data set radial forging at afrc testbed v2 (Sep. 2019). doi:10.5281/zenodo.3405265. URL https://doi.org/10.5281/zenodo.3405265

[39] J. W. Tukey, et al., Exploratory data analysis, Vol. 2, Reading, Mass., 1977.

[40] L. N. Smith, Cyclical learning rates for training neural networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2017, pp. 464–472.

[41] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

[42] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[43] R. Novak, L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein, S. S. Schoenholz, Neural tangents: Fast and easy infinite neural networks in python, in: International Conference on Learning Representations, 2020. URL https://github.com/google/neural-tangents

[44] F. Melo, Area under the ROC Curve, Springer New York, New York, NY, 2013, pp. 38–39. doi:10.1007/978-1-4419-9863-7_209.

[45] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, B. Lakshminarayanan, Do deep generative models know what they don't know?, in: International Conference on Learning Representations, 2019.

[46] H. Choi, E. Jang, A. A. Alemi, Waic, but why? generative ensembles for robust anomaly detection, arXiv preprint arXiv:1810.01392 (2018).

[47] M. Hinne, Q. F. Gronau, D. van den Bergh, E.-J. Wagenmakers, A conceptual introduction to bayesian model averaging, Advances in Methods and Practices in Psychological Science 3 (2) (2020) 200–215. doi:10.1177/2515245919898657.
URL https://doi.org/10.1177/2515245919898657

[48] G. Zhao, J. Liu, J. Jiang, H. Guan, J.-R. Wen, Skip-connected deep convolutional autoencoder for restoration of document images, in: 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 2935–2940. doi:10.1109/ICPR.2018.8546199.

[49] X. Mao, C. Shen, Y.-B. Yang, Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 29, Curran Associates, Inc., 2016.
URL https://proceedings.neurips.cc/paper/2016/file/0ed9422357395a0d4879191c66f4faa2-Paper.pdf

[50] J.-Y. Liu, Y.-H. Yang, Denoising auto-encoder with recurrent skip connections and residual regression for music source separation, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 773–778. doi:10.1109/ICMLA.2018.00123.

[51] J. Snoek, R. Adams, H. Larochelle, On nonparametric guidance for learning autoencoder representations, in: Artificial Intelligence and Statistics, PMLR, 2012, pp. 1073–1080.

[52] T. V. Nguyen, R. K. Wong, C. Hegde, Benefits of jointly training autoencoders: An improved neural tangent kernel analysis, IEEE Transactions on Information Theory (2021).

[53] A. Radhakrishnan, K. Yang, M. Belkin, C. Uhler, Memorization in overparameterized autoencoders, arXiv preprint arXiv:1810.10333 (2018).

[54] C. Zhang, S. Bengio, M. Hardt, M. C. Mozer, Y. Singer, Identity crisis: Memorization and generalization under extreme overparameterization, arXiv preprint arXiv:1902.04698 (2019).

[55] M. Rosenblatt, Remarks on Some Nonparametric Estimates of a Density Function, The Annals of Mathematical Statistics 27 (3) (1956) 832 – 837. doi:10.1214/aoms/1177728190.
URL https://doi.org/10.1214/aoms/1177728190

[56] E. Parzen, On estimation of a probability density function and mode, The annals of mathematical statistics 33 (3) (1962) 1065–1076.