

---

# Learning Dynamics of Linear Denoising Autoencoders

---

Arnu Pretorius<sup>1,2</sup> Steve Kroon<sup>1,2</sup> Herman Kamper<sup>3</sup>

## Abstract

Denoising autoencoders (DAEs) have proven useful for unsupervised representation learning, but a thorough theoretical understanding is still lacking of how the input noise influences learning. Here we develop theory for how noise influences learning in DAEs. By focusing on linear DAEs, we are able to derive analytic expressions that exactly describe their learning dynamics. We verify our theoretical predictions with simulations as well as experiments on MNIST and CIFAR-10. The theory illustrates how, when tuned correctly, noise allows DAEs to ignore low variance directions in the inputs while learning to reconstruct them. Furthermore, in a comparison of the learning dynamics of DAEs to standard regularised autoencoders, we show that noise has a similar regularisation effect to weight decay, but with faster training dynamics. We also show that our theoretical predictions approximate learning dynamics on real-world data and qualitatively match observed dynamics in nonlinear DAEs.\*

## 1. Introduction

The goal of unsupervised learning is to uncover hidden structure in unlabelled data, often in the form of latent feature representations. One popular type of model, an autoencoder, does this by trying to reconstruct its input (Bengio et al., 2007). Autoencoders have been used in various forms to address problems in machine translation (Chandar et al., 2014; Tu et al., 2017), speech processing (Elman & Zipser, 1987; Zeiler et al., 2013), and computer vision (Rifai et al., 2011;

Larsson, 2017), to name just a few areas. Denoising autoencoders (DAEs) are an extension of autoencoders which learn latent features by reconstructing data from *corrupted* versions of the inputs (Vincent et al., 2008). Although this corruption step typically leads to improved performance over standard autoencoders, a theoretical understanding of its effects remains incomplete. In this paper, we provide new insights into the inner workings of DAEs by analysing the learning dynamics of linear DAEs.

We specifically build on the work of Saxe et al. (2013a;b), who studied the learning dynamics of deep linear networks in a supervised regression setting. By analysing the gradient descent weight update steps as time-dependent differential equations (in the limit as the learning rate approaches a small value), Saxe et al. (2013a) were able to derive exact solutions for the learning trajectory of these networks as a function of training time. Here we extend their approach to linear DAEs. To do this, we use the expected reconstruction loss over the noise distribution as an objective (requiring a different decomposition of the input covariance) as a tractable way to incorporate noise into our analytic solutions. This approach yields exact equations which can predict the learning trajectory of a linear DAE.

Our work here shares the motivation of many recent studies (Advani & Saxe, 2017; Pennington & Worah, 2017; Pennington & Bahri, 2017; Nguyen & Hein, 2017; Dinh et al., 2017; Louart et al., 2017; Swirszcz et al., 2017; Lin et al., 2017; Neyshabur et al., 2017; Soudry & Hoffer, 2017; Pennington et al., 2017) working towards a better theoretical understanding of neural networks and their behaviour. Although we focus here on a theory for *linear* networks, such networks have learning dynamics that are in fact *nonlinear*. Furthermore, analyses of linear networks have also proven useful in understanding the behaviour of nonlinear neural networks (Saxe et al., 2013a; Advani & Saxe, 2017).

First we introduce linear DAEs (§2). We then derive analytic expressions for their nonlinear learning dynamics (§3), and verify our solutions in simulations (§4) which show how noise can influence the shape of the loss surface and change the rate of convergence for gradient descent optimisation. We also find that an appropriate amount of noise can help DAEs ignore low variance directions in the input while learning the reconstruction mapping. In the remainder of

<sup>1</sup>Computer Science Division, Stellenbosch University, South Africa <sup>2</sup>CSIR/SU Centre for Artificial Intelligence Research, <sup>3</sup>Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa. Correspondence to: Steve Kroon <kroon@sun.ac.za>.

*Proceedings of the 35<sup>th</sup> International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

\*Code to reproduce all the results in this paper is available at: [https://github.com/arnupretorius/lindaedynamics\\_icml2018](https://github.com/arnupretorius/lindaedynamics_icml2018)

the paper, we compare DAEs to standard regularised autoencoders and show that our theoretical predictions match both simulations (§5) and experimental results on MNIST and CIFAR-10 (§6). We specifically find that while the noise in a DAE has an equivalent effect to standard weight decay, the DAE exhibits faster learning dynamics. We also show that our observations hold qualitatively for nonlinear DAEs.

## 2. Linear Denoising Autoencoders

We first give the background of linear DAEs. Given training data consisting of pairs  $\{(\tilde{\mathbf{x}}_i, \mathbf{x}_i), i = 1, \dots, N\}$ , where  $\tilde{\mathbf{x}}$  represents a corrupted version of the training data  $\mathbf{x} \in \mathbb{R}^D$ , the reconstruction loss for a single hidden layer DAE with activation function  $\phi$  is given by

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^N \|\mathbf{x}_i - W_2 \phi(W_1 \tilde{\mathbf{x}}_i)\|^2.$$

Here,  $W_1 \in \mathbb{R}^{H \times D}$  and  $W_2 \in \mathbb{R}^{D \times H}$  are the weights of the network with hidden dimensionality  $H$ . The learned feature representations correspond to the latent variable  $\mathbf{z} = \phi(W_1 \tilde{\mathbf{x}})$ .

To corrupt an input  $\mathbf{x}$ , we sample a noise vector  $\epsilon$ , where each component is drawn i.i.d. from a pre-specified noise distribution with mean zero and variance  $s^2$ . We define the corrupted version of the input as  $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$ . This ensures that the expectation over the noise remains unbiased, i.e.  $\mathbb{E}_\epsilon(\tilde{\mathbf{x}}) = \mathbf{x}$ .

Restricting our scope to linear neural networks, with  $\phi(a) = a$ , the loss in expectation over the noise distribution is

$$\begin{aligned} \mathbb{E}_\epsilon[\mathcal{L}] &= \frac{1}{2N} \sum_{i=1}^N \|\mathbf{x}_i - W_2 W_1 \mathbf{x}_i\|^2 \\ &\quad + \frac{s^2}{2} \text{tr}(W_2 W_1 W_1^T W_2^T), \end{aligned} \quad (1)$$

See the supplementary material for the full derivation.

## 3. Learning Dynamics of Linear DAEs

Here we derive the learning dynamics of linear DAEs, beginning with a brief outline to build some intuition.

The weight update equations for a linear DAE can be formulated as time-dependent differential equations in the limit as the gradient descent learning rate becomes small (Saxe et al., 2013a). The task of an ordinary (undercomplete) linear autoencoder is to learn the identity mapping that reconstructs the original input data. The matrix corresponding to this learned map will essentially be an approximation of the full identity matrix that is of rank equal to the input dimension. It turns out that tracking the temporal updates of this mapping represents a difficult problem that involves dealing with

coupled differential equations, since both the on-diagonal and off-diagonal elements of the weight matrices need to be considered in the approximation dynamics at each time step.

To circumvent this issue and make the analysis tractable, we follow the methodology introduced in Saxe et al. (2013a), which is to: (1) decompose the input covariance matrix using an eigenvalue decomposition; (2) rotate the weight matrices to align with these computed directions of variation; and (3) use an orthogonal initialisation strategy to diagonalise the composite weight matrix  $W = W_2 W_1$ . The important difference in our setting, is that additional constraints are brought about through the injection of noise.

The remainder of this section outlines this derivation for the exact solutions to the learning dynamics of linear DAEs.

### 3.1. Gradient descent update

Consider a continuous time limit approach to studying the learning dynamics of linear DAEs. This is achieved by choosing a sufficiently small learning rate  $\alpha$  for optimising the loss in (1) using gradient descent. The update for  $W_1$  in a single gradient descent step then takes the form of a time-dependent differential equation

$$\begin{aligned} \tau \frac{d}{dt} W_1 &= \sum_{i=1}^N W_2^T (\mathbf{x}_i \mathbf{x}_i^T - W_2 W_1 \mathbf{x}_i \mathbf{x}_i^T) \\ &\quad - \epsilon W_2^T W_2 W_1 \\ &= W_2^T (\Sigma_{xx} - W_2 W_1 \Sigma_{xx}) - \epsilon W_2^T W_2 W_1. \end{aligned}$$

Here  $t$  is the time measured in epochs,  $\tau = \frac{N}{\alpha}$ ,  $\epsilon = N s^2$  and  $\Sigma_{xx} = \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T$ , represents the input covariance matrix. Let the eigenvalue decomposition of the input covariance be  $\Sigma_{xx} = V \Lambda V^T$ , where  $V$  is an orthogonal matrix and denote the eigenvalues  $\lambda_j = [\Lambda]_{jj}$ , with  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D$ . The update can then be rewritten as

$$\begin{aligned} \tau \frac{d}{dt} W_1 &= W_2^T V (\Lambda - V^T W_2 W_1 V \Lambda) V^T \\ &\quad - \epsilon W_2^T W_2 W_1. \end{aligned}$$

The weight matrices can be rotated to align with the directions of variation in the input by performing the rotations  $\bar{W}_1 = W_1 V$  and  $\bar{W}_2 = V^T W_2$ . Following a similar derivation for  $W_2$ , the weight updates become

$$\begin{aligned} \tau \frac{d}{dt} \bar{W}_1 &= \bar{W}_2^T (\Lambda - \bar{W}_2 \bar{W}_1 \Lambda) - \epsilon \bar{W}_2^T \bar{W}_2 \bar{W}_1 \\ \tau \frac{d}{dt} \bar{W}_2 &= (\Lambda - \bar{W}_2 \bar{W}_1 \Lambda) \bar{W}_1^T - \epsilon \bar{W}_2 \bar{W}_1 \bar{W}_1^T. \end{aligned}$$

### 3.2. Orthogonal initialisation and scalar dynamics

To decouple the dynamics, we can set  $W_2 = V D_2 R^T$  and  $W_1 = R D_1 V^T$ , where  $R$  is an arbitrary orthogonal matrix

and  $D_2$  and  $D_1$  are diagonal matrices. This results in the product of the realigned weight matrices

$$\overline{W}_2 \overline{W}_1 = V^T V D_2 R^T R D_1 V^T V = D_2 D_1$$

to become diagonal. The updates now reduce to the following scalar dynamics that apply independently to each pair of diagonal elements  $w_{1j}$  and  $w_{2j}$  of  $D_1$  and  $D_2$  respectively:

$$\tau \frac{d}{dt} w_{1j} = w_{2j} \lambda_j (1 - w_{2j} w_{1j}) - \varepsilon w_{2j}^2 w_{1j} \quad (2)$$

$$\tau \frac{d}{dt} w_{2j} = w_{1j} \lambda_j (1 - w_{2j} w_{1j}) - \varepsilon w_{2j} w_{1j}^2. \quad (3)$$

Note that the same dynamics stem from gradient descent on the loss given by

$$\ell = \sum_{j=1}^D \frac{\lambda_j}{2\tau} (1 - w_{2j} w_{1j})^2 + \sum_{j=1}^D \frac{\varepsilon}{2\tau} (w_{2j} w_{1j})^2. \quad (4)$$

By examining (4), it is evident that the degree to which the first term will be reduced will depend on the magnitude of the associated eigenvalue  $\lambda_j$ . However, for directions in the input covariance  $\Sigma_{xx}$  with relatively little variation the decrease in the loss from learning the identity map will be negligible and is likely to result in overfitting (since little to no signal is being captured by these eigenvalues). The second term in (4) is the result of the input corruption and acts as a suppressant on the magnitude of the weights in the learned mapping. Our interest is to better understand the interplay between these two terms during learning by studying their scalar learning dynamics.

### 3.3. Exact solutions to the dynamics of learning

As noted above, the dynamics of learning are dictated by the value of  $w = w_2 w_1$  over time. An expression can be derived for  $w(t)$  by using a hyperbolic change of coordinates in (2) and (3), letting  $\theta$  parameterise points along a dynamics trajectory represented by the conserved quantity  $w_2^2 - w_1^2 = \pm c_0$ . This relies on the fact that  $\ell$  is invariant under a scaling of the weights such that  $w = (w_1/c)(c w_2) = w_2 w_1$  for any constant  $c$  (Saxe et al., 2013a). Starting at any initial point  $(w_1, w_2)$  the dynamics are

$$w(t) = \frac{c_0}{2} \sinh(\theta_t), \quad (5)$$

with

$$\theta_t = 2 \tanh^{-1} \left[ \frac{(1-E)(\zeta^2 - \beta^2 - 2\beta\delta) - 2(1+E)\zeta\delta}{(1-E)(2\beta + 4\delta) - 2(1+E)\zeta} \right]$$

where  $\beta = c_0 \left(1 + \frac{\varepsilon}{\lambda}\right)$ ,  $\zeta = \sqrt{\beta^2 + 4}$ ,  $\delta = \tanh\left(\frac{\theta_0}{2}\right)$  and  $E = e^{\zeta \lambda t / \tau}$ . Here  $\theta_0$  depends on the initial weights  $w_1$  and  $w_2$  through the relationship  $\theta_0 = \sinh^{-1}(2w/c_0)$ . The

derivation for  $\theta_t$  involves rewriting  $\tau \frac{d}{dt} w$  in terms of  $\theta$ , integrating over the interval  $\theta_0$  to  $\theta_t$ , and finally rearranging terms to get an expression for  $\theta(t) \equiv \theta_t$  (see the supplementary material for full details). To derive the learning dynamics for different noise distributions, the corresponding  $\varepsilon$  must be computed and used to determine  $\beta$  and  $\zeta$ . For example, sampling noise from a Gaussian distribution such that  $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ , gives  $\varepsilon = N\sigma^2$ . Alternatively, if  $\epsilon$  is distributed according to a zero-mean Laplace distribution with scale parameter  $b$ , then  $\varepsilon = 2Nb^2$ .

## 4. The Effects of Noise: a Simulation Study

Since the expression for the learning dynamics of a linear DAE in (5) evolve independently for each direction of variation in the input, it is enough to study the effect that noise has on learning for a single eigenvalue  $\lambda$ . To do this we trained a scalar linear DAE to minimise the loss  $\ell_\lambda = \frac{\lambda}{2}(1 - w_2 w_1)^2 + \frac{\varepsilon}{2}(w_2 w_1)^2$  with  $\lambda = 1$  using gradient descent. Starting from several different randomly initialised weights  $w_1$  and  $w_2$ , we compare the simulated dynamics with those predicted by equation (5). The top row in Figure 1 shows the exact fit between the predictions and numerical simulations for different noise levels,  $\varepsilon = 0, 1, 5$ .

The trajectories in the top row of Figure 1 converge to the optimal solution at different rates depending on the amount of injected noise. Specifically, adding more noise results in faster convergence. However, the trade-off in (4) ensures that the fixed point solution also diminishes in magnitude.

To gain further insight, we also visualise the associated loss surfaces for each experiment in the bottom row of Figure 1. Note that even though the scalar product  $w_2 w_1$  defines a linear mapping, the minimisation of  $\ell_\lambda$  with respect to  $w_1$  and  $w_2$  is a non-convex optimisation problem. The loss surfaces in Figure 1 each have an unstable saddle point at  $w_2 = w_1 = 0$  (red star) with all remaining fixed points lying on a minimum loss manifold (cyan curve). This manifold corresponds to the different possible combinations of  $w_2$  and  $w_1$  that minimise  $\ell_\lambda$ . The paths that gradient descent follow from various initial starting weights down to points situated on the manifold are represented by dashed orange lines.

For a fixed value of  $\lambda$ , adding noise warps the loss surface making steeper slopes and pulling the minimum loss manifold in towards the saddle point. Therefore, steeper descent directions cause learning to converge at a faster rate to fixed points that are smaller in magnitude. This is the result of a sharper curving loss surface and the minimum loss manifold lying closer to the origin.

We can compute the fixed point solution for any pair of initial starting weights (not on the saddle point) by taking

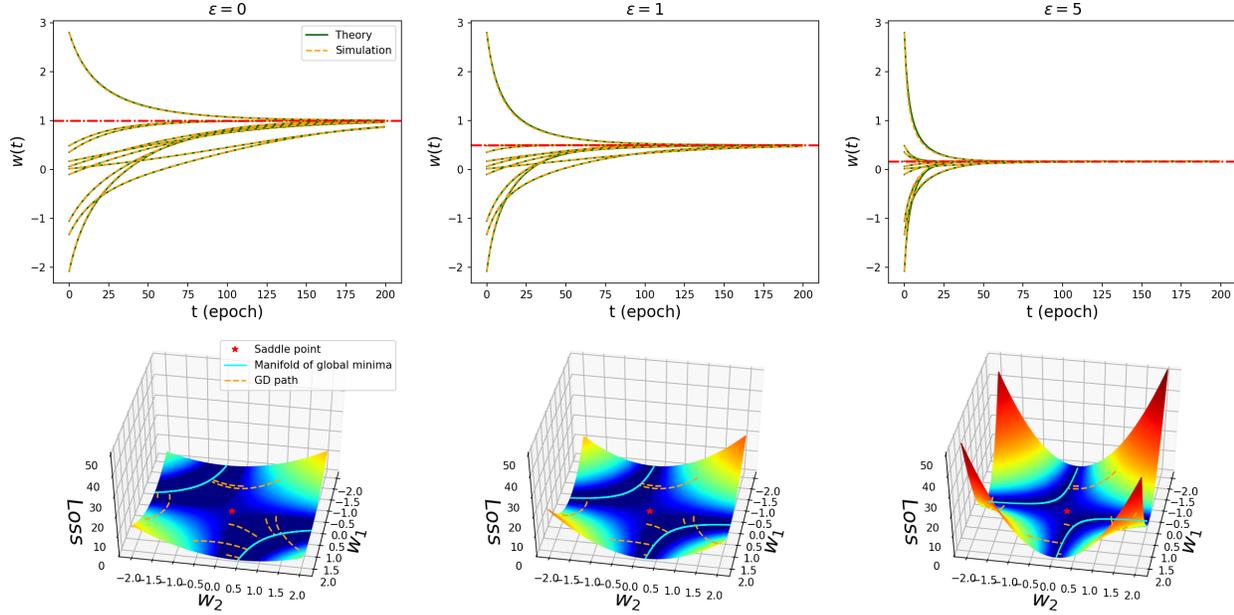


Figure 1. Learning dynamics, loss surface and gradient descent paths for linear denoising autoencoders. **Top:** Learning dynamics for each simulated run (dashed orange lines) together with the theoretically predicted learning dynamics (solid green lines). The red line in each plot indicates the final value of the resulting fixed point solution  $w^*$ . **Bottom:** The loss surface corresponding to the loss  $\ell_\lambda = \frac{\lambda}{2}(1 - w_2w_1)^2 + \frac{\varepsilon}{2}(w_2w_1)^2$  for  $\lambda = 1$ , as well as the gradient descent paths (dashed orange lines) for randomly initialised weights. The cyan hyperboloids represent the global minimum loss manifold that corresponds to all possible combinations of  $w_2$  and  $w_1$  that minimise  $\ell_\lambda$ . **Left:**  $\varepsilon = 0, w^* = 1$ . **Middle:**  $\varepsilon = 1, w^* = 0.5$ . **Right:**  $\varepsilon = 5, w^* = 1/6$ .

the derivative

$$\frac{d\ell_\lambda}{dw} = -\frac{\lambda}{\tau}(1 - w) + \frac{\varepsilon}{\tau}w,$$

and setting it equal to zero to find  $w^* = \frac{\lambda}{\lambda + \varepsilon}$ . This solution reveals the interaction between the input variance associated with  $\lambda$  and the noise  $\varepsilon$ . For large eigenvalues for which  $\lambda \gg \varepsilon$ , the fixed point will remain relatively unaffected by adding noise, i.e.,  $w^* \approx 1$ . In contrast, if  $\lambda \ll \varepsilon$ , the noise will result in  $w^* \approx 0$ . This means that over a distribution of eigenvalues, an appropriate amount of noise can help a DAE to ignore low variance directions in the input data while learning the reconstruction. In a practical setting, this motivates the tuning of noise levels on a development set to prevent overfitting.

## 5. The Relationship Between Noise and Weight Decay

It is well known that adding noise to the inputs of a neural network is equivalent to a form of regularisation (Bishop, 1995). Therefore, to further understand the role of noise in linear DAEs we compare the dynamics of noise to those of explicit regularisation in the form of *weight decay* (Krogh & Hertz, 1992). The reconstruction loss for a linear weight

decayed autoencoder (WDAE) is given by

$$\frac{1}{2N} \sum_{i=1}^N \|\mathbf{x}_i - W_2 W_1 \mathbf{x}_i\|^2 + \frac{\gamma}{2} (\|W_1\|^2 + \|W_2\|^2) \quad (6)$$

where  $\gamma$  is the penalty parameter that controls the amount of regularisation applied during learning. Provided that the weights of the network are initialised to be small, it is also possible (see supplementary material) to derive scalar dynamics of learning from (6) as

$$w_\gamma(t) = \frac{\xi E_\gamma}{E_\gamma - 1 + \xi/w_0}, \quad (7)$$

where  $\xi = (1 - N\gamma/\lambda)$  and  $E_\gamma = e^{2\xi t/\tau}$ .

Figure 2 compares the learning trajectories of linear DAEs and WDAEs over time (as measured in training epochs) for  $\lambda = 2.5, 1, 0.5$  and  $0.1$ . The dynamics for both noise and weight decay exhibit a sigmoidal shape with an initial period of inactivity followed by rapid learning, finally reaching a plateau at the fixed point solution. Figure 2 illustrates that the learning time associated with an eigenvalue is negatively correlated with its magnitude. Thus, the eigenvalue corresponding to the largest amount of variation explained is the quickest to escape inactivity during learning.

The colour intensity of the lines in Figure 2 correspond to the amount of noise or regularisation applied in each run,

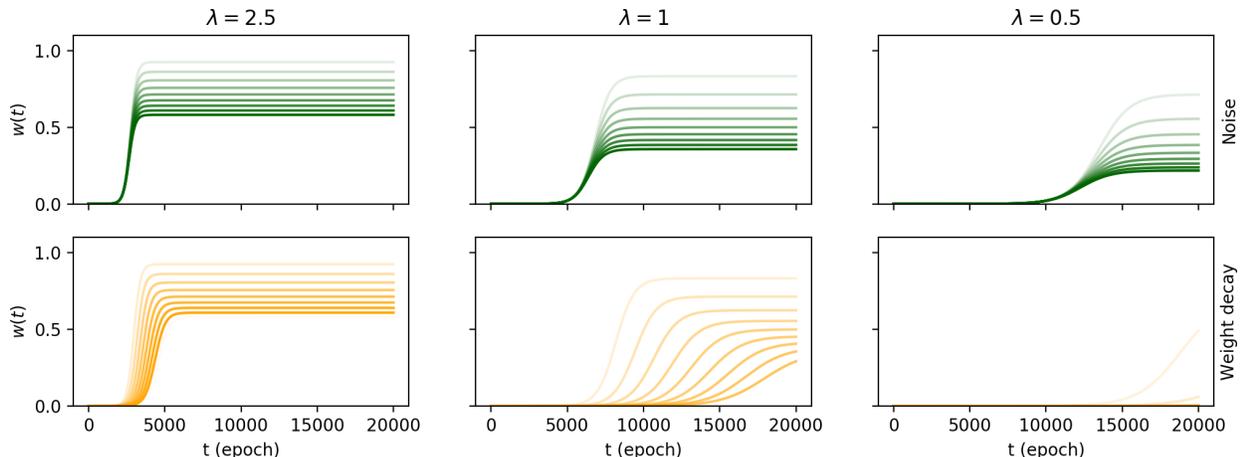


Figure 2. Theoretically predicted learning dynamics for noise compared to weight decay for linear autoencoders. **Top:** Noise dynamics (green), darker line colours correspond to larger amounts of added noise. **Bottom:** Weight decay dynamics (orange), darker line colours correspond to larger amounts of regularisation. **Left to right:** Eigenvalues  $\lambda = 2.5, 1$  and  $0.5$  associated with high to low variance.

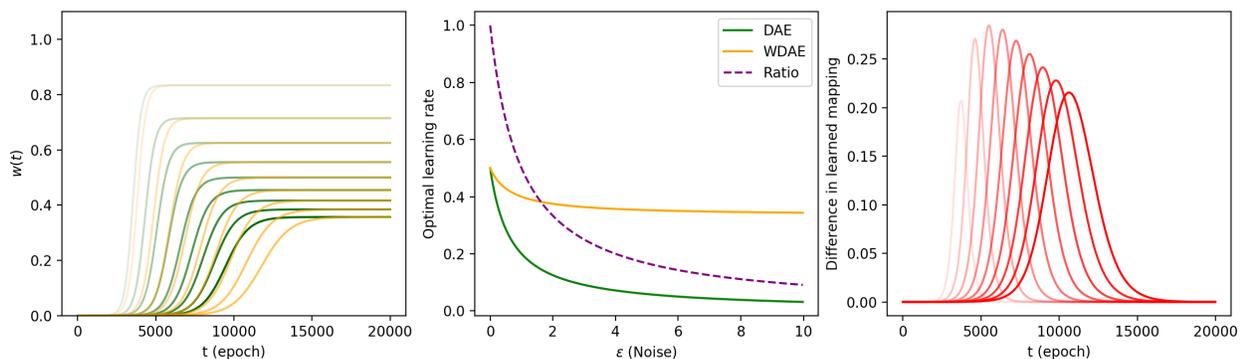


Figure 3. Learning dynamics for optimal discrete time learning rates ( $\lambda = 1$ ). **Left:** Dynamics of DAEs (green) vs. WDAEs (orange), where darker line colours correspond to larger amounts noise or weigh decay. **Middle:** Optimal learning rate as a function of noise  $\varepsilon$  for DAEs, and for WDAEs using an equivalent amount of regularisation  $\gamma = \lambda\varepsilon/(\lambda + \varepsilon)$ . **Right:** Difference in mapping over time.

with darker lines indicating larger amounts. In the continuous time limit with equal learning rates, when compared with noise dynamics, weight decay experiences a delay in learning such that the initial inactive period becomes extended for every eigenvalue, whereas adding noise has no effect on learning time. In other words, starting from small weights, noise injected learning is capable of providing an equivalent regularisation mechanism to that of weight decay in terms of a constrained fixed point mapping, but with zero time delay.

However, this analysis does not take into account the practice of using well-tuned stable learning rates for discrete optimisation steps. We therefore consider the impact on training time when using optimised learning rates for each approach. By using second order information from the Hessian as in Saxe et al. (2013a), (here of the expected reconstruction loss with respect to the scalar weights), we relate the optimal learning rates for linear DAEs and WDAEs,

where each optimal rate is inversely related to the amount of noise/regularisation applied during training (see supplementary material). The ratio of the optimal DAE rate to that for the WDAE is

$$R = \frac{2\lambda + \gamma}{2\lambda + 3\varepsilon}. \quad (8)$$

Note that the ratio in (8) will essentially be equal to one for eigenvalues that are significantly larger than both  $\varepsilon$  and  $\gamma$ , with deviations from unity only manifesting for smaller values of  $\lambda$ .

Furthermore, weight decay and noise injected learning result in equivalent scalar solutions when their parameters are related by  $\gamma = \frac{\lambda\varepsilon}{\lambda + \varepsilon}$  (see supplementary material). This leads to the following two observations. First, it shows that adding noise during learning can be interpreted as a form of weight decay where the penalty parameter  $\gamma$  adapts to each direction of variation in the data. In other words, noise essentially makes use of the statistical structure of

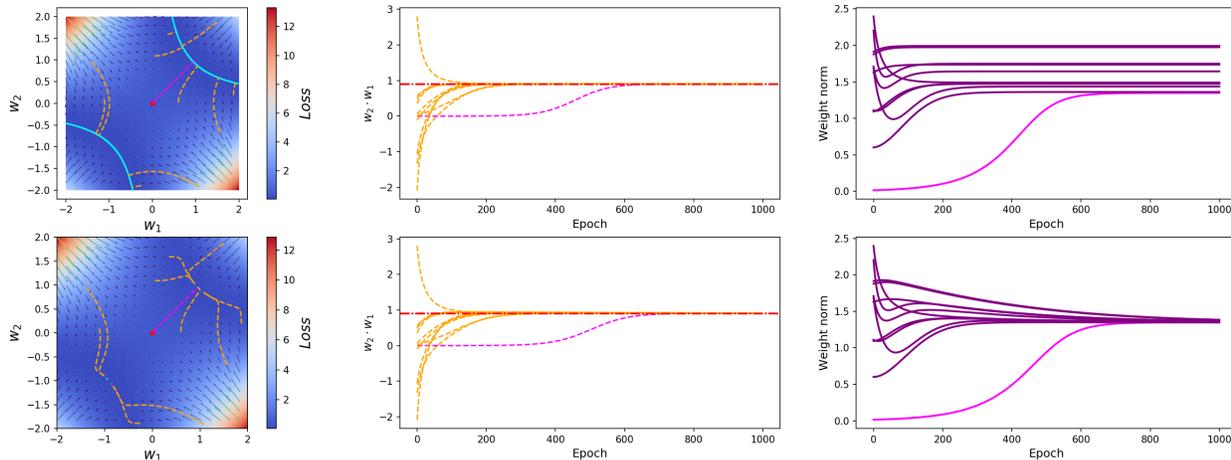


Figure 4. The effect of noise versus weight decay on the norm of the weights during learning. **Left:** Two-dimensional loss surface  $\ell_\lambda = \frac{\lambda}{2}(1 - w_2 w_1)^2 + \frac{\varepsilon}{2}(w_2 w_1)^2 + \frac{\gamma}{2}(w_2^2 + w_1^2)$ . Gradient descent paths (orange/magenta dashed lines), minimum loss manifold (cyan curves), saddle point (red star). **Middle:** Simulated learning dynamics. **Right:** Norm of the weights over time for each simulated run. **Top:** Noise with  $\lambda = 1, \varepsilon = 0.1$  and  $\gamma = 0$ . **Bottom:** Weight decay with  $\lambda = 1, \varepsilon = 0$  and  $\gamma = \lambda(0.1)/(\lambda + 0.1) = 0.091$ . The magenta line in each plot corresponds to a simulated run with small initialised weights.

the input data to influence the amount of shrinkage that is being applied in various directions during learning. Second, together with (8), we can theoretically compare the learning dynamics of DAEs and WDAEs, when both equivalent regularisation and the relative differences in optimal learning rates are taken into account.

The effects of optimal learning rates (for  $\lambda = 1$ ), are shown in Figure 3. DAEs still exhibit faster dynamics (left panel), even when taking into account the difference in the learning rate as a function of noise, or equivalent weight decay (middle panel). In addition, for equivalent regularisation effects, the ratio of the optimal rates  $R$  can be shown to be a monotonically decreasing function of the noise level, where the rate of decay depends on the size of  $\lambda$ . This means that for any amount of added noise, the DAE will require a slower learning rate than that of the WDAE. Even so, a faster rate for the WDAE does not seem to compensate for its slower dynamics and the difference in learning time is also shown to grow as more noise (regularisation) is applied during training (right panel).

### 5.1. Exploiting invariance in the loss function

A primary motivation for weight decay as a regulariser is that it provides solutions with smaller weight norms, producing smoother models that have better generalisation performance. Figure 4 shows the effect of noise (top row) compared to weight decay (bottom row) on the norm of the weights during learning. Looking at the loss surface for weight decay (bottom left panel), the penalty on the size of the weights acts by shrinking the minimum loss manifold down from a long curving valley to a single point (associ-

ated with a small norm solution). Interestingly, this results in gradient descent following a trajectory towards an “invisible” minimum loss manifold similar to the one associated with noise. However, once on this manifold, weight decay begins to exploit invariances in the loss function to changes in the weights, so as to move along the manifold down towards smaller norm solutions. This means that even when the two approaches learn the exact same mapping over time (as shown by the learning dynamics in the middle column of Figure 4), additional epochs will cause weight decay to further reduce the size of the weights (bottom right panel). This happens in a stage-like manner where the optimisation first focuses on reducing the reconstruction loss by learning the optimal mapping and then reduces the regularisation loss through invariance.

### 5.2. Small weight initialisation and early stopping

It is common practice to initialise the weights of a network with small values. In fact, this strategy has recently been theoretically shown to help, along with early stopping, to ensure good generalisation performance for neural networks in certain high-dimensional settings (Advani & Saxe, 2017). In our analysis however, what we find interesting about small weight initialisation is that it removes some of the differences in the learning behaviour of DAEs compared to regularised autoencoders that use weight decay.

To see this, the magenta lines in Figure 4 show the learning dynamics for the two approaches where the weights of both the networks were initialised to small random starting values. The learning dynamics are almost identical in terms of their temporal trajectories and have equal fixed

points. However, what is interesting is the implicit regularisation that is brought about through the small initialisation. By starting small and making incremental updates to the weights, the scalar solution in both cases end up being equal to the minimum norm solution. In other words, the path that gradient descent takes from the initialisation to the minimum loss manifold, reaches the manifold where the norm of the weights happen to also be small. This means that the second phase of weight decay (where the invariance of the loss function would be exploited to reduce the regularisation penalty), is not only no longer necessary, but also does not result in a norm that is appreciably smaller than that obtained by learning with added noise. Therefore in this case, learning with explicit regularisation provides no additional benefit over that of learning with noise in terms of reducing the norm of the weights during training.

When initialising small, early stopping can also serve as a form of implicit regularisation by ensuring that the weights do not change past the point where the validation loss starts to increase (Bengio et al., 2007). In the context of learning dynamics, early stopping for DAEs can be viewed as a method that effectively selects only the directions of variation deemed useful for generalisation during reconstruction, considering the remaining eigenvalues to carry no additional signal.

## 6. Experimental Results

To verify the dynamics of learning on real-world data sets we compared theoretical predictions with actual learning on MNIST and CIFAR-10. In our experiments we considered the following linear autoencoder networks: a regular AE, a WDAE and a DAE.

For MNIST, we trained each autoencoder with small randomly initialised weights, using  $N = 50000$  training samples for 5000 epochs, with a learning rate  $\alpha = 0.01$  and a hidden layer width of  $H = 256$ . For the WDAE, the penalty parameter was set at  $\gamma = 0.5$  and for the DAE,  $\sigma^2 = 0.5$ . The results are shown in Figure 5 (left column).

The theoretical predictions (solid lines) in Figure 5 show good agreement with the actual learning dynamics (points). As predicted, both regularisation (orange) and noise (green) suppress the fixed point value associated with the different eigenvalues and, whereas regularisation delays learning (fewer fixed points are reached by the WDAE during training when compared to the DAE), the use of noise has no effect on training time.

Similar agreement is shown for CIFAR-10 in the right column of Figure 5. Here, we trained each network with small randomly initialised weights using  $N = 30000$  training samples for 5000 epochs, with a learning rate  $\alpha = 0.001$  and a hidden dimension  $H = 512$ . For the WDAE, the

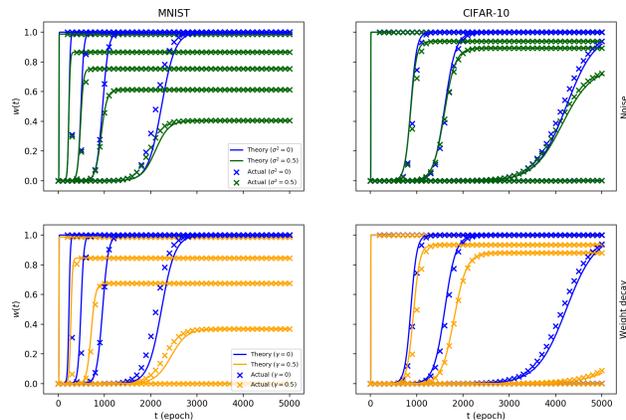


Figure 5. Learning dynamics for MNIST and CIFAR-10. Solid lines represent theoretical dynamics and ‘x’ markers simulated dynamics. Shown are the mappings associated with the set of eigenvalues  $\{\lambda_i, i = 1, 4, 8, 16, 32\}$ , where the remaining eigenvalues were excluded to improve readability. **Top:** Noise: AE (blue) vs. DAE with  $\sigma^2 = 0.5$  (green). **Bottom:** Weight decay: AE (blue) vs. WDAE with  $\gamma = 0.5$  (orange). **Left:** MNIST. **Right:** CIFAR-10.

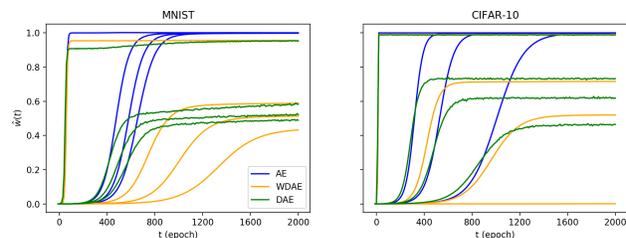


Figure 6. Learning dynamics for nonlinear networks using ReLU activation. AE (blue), WDAE (orange) and DAE (green). Shown are the mappings associated with the first four eigenvalues, i.e.  $\{\lambda_i, i = 1, 2, 3, 4\}$ . **Left:** MNIST **Right:** CIFAR-10.

penalty parameter was set at  $\gamma = 0.5$  and for the DAE,  $\sigma^2 = 0.5$ .

Next, we investigated whether these dynamics are at least also qualitatively present in nonlinear autoencoder networks. Figure 6 shows the dynamics of learning for nonlinear AEs, WDAEs and DAEs, using ReLU activations, trained on MNIST ( $N = 50000$ ) and CIFAR-10 ( $N = 30000$ ) with equal learning rates. For the DAE, the input was corrupted using sampled Gaussian noise with mean zero and  $\sigma^2 = 3$ . For the WDAE, the amount of weight decay was manually tuned to  $\gamma = 0.0045$ , to ensure that both autoencoders displayed roughly the same degree of regularisation in terms of the fixed points reached. During the course of training, the identity mapping associated with each eigenvalue was estimated (see supplementary material), at equally spaced intervals of size 10 epochs.

The learning dynamics are qualitatively similar to the dy-

namics observed in the linear case. Both noise and weight decay result in a shrinkage of the identity mapping associated with each eigenvalue. Furthermore, in terms of the number of training epochs, the DAE is seen to learn as quickly as a regular AE, whereas the WDAE incurs a delay in learning time. Although these experimental results stem from a single training run for each autoencoder, we note that wall-clock times for training may still differ because DAEs require some additional time for sampling noise. Similar results were observed when using a tanh nonlinearity and are provided in the supplementary material.

## 7. Related Work

There have been many studies aiming to provide a better theoretical understanding of DAEs. Vincent et al. (2008) analysed DAEs from several different perspectives, including manifold learning and information filtering, by establishing an equivalence between different criteria for learning and the original training criterion that seeks to minimise the reconstruction loss. Subsequently, Vincent (2011) showed that under a particular set of conditions, the training of DAEs can also be interpreted as a type of score matching. This connection provided a probabilistic basis for DAEs. Following this, a more in-depth analysis of DAEs as a possible generative model suitable for arbitrary loss functions and multiple types of data was given by Bengio et al. (2013).

In contrast to a probabilistic understanding of DAEs, we present here an analysis of the learning process. Specifically inspired by Saxe et al. (2013a), as well as by earlier work on supervised neural networks (Oppen, 1988; Sanger, 1989; Baldi & Hornik, 1989; Saad & Solla, 1995), we provide a theoretical investigation of the temporal behaviour of linear DAEs using derived equations that exactly describe their dynamics of learning. Specifically for the linear case, the squared error loss for the reconstruction contractive autoencoder (RCAE) introduced in Alain & Bengio (2014) is equivalent to the expected loss (over the noise) for the DAE. Therefore, the learning dynamics described in this paper also apply to linear RCAEs.

For our analysis to be tractable we used a marginalised reconstruction loss where the gradient descent dynamics are viewed in expectation over the noise distribution. Whereas our motivation is analytical in nature, marginalising the reconstruction loss tends to be more commonly motivated from the point of view of learning useful and robust feature representations at a significantly lower computational cost (Chen et al., 2014; 2015). This approach has also been investigated in the context of supervised learning (van der Maaten et al., 2013; Wang & Manning, 2013; Wager et al., 2013). Also related to our work is the analysis by Poole et al. (2014), who showed that training autoencoders with noise (added at different levels of the network architecture),

is closely connected to training with explicit regularisation and proposed a marginalised noise framework for noisy autoencoders.

## 8. Conclusion and Future Work

This paper analysed the learning dynamics of linear denoising autoencoders (DAEs) with the aim of providing a better understanding of the role of noise during training. By deriving exact time-dependent equations for learning, we showed how noise influences the shape of the loss surface as well as the rate of convergence to fixed point solutions. We also compared the learning behaviour of added input noise to that of weight decay, an explicit form of regularisation. We found that while the two have similar regularisation effects, the use of noise for regularisation results in faster training. We compared our theoretical predictions with actual learning dynamics on real-world data sets, observing good agreement. In addition, we also provided evidence (on both MNIST and CIFAR-10) that our predictions hold qualitatively for nonlinear DAEs.

This work provides a solid basis for further investigation. Our analysis could be extended to nonlinear DAEs, potentially using the recent work on nonlinear random matrix theory for neural networks (Pennington & Worah, 2017; Louart et al., 2017). Our findings indicate that appropriate noise levels help DAEs ignore low variance directions in the input; we also obtained new insights into the training time of DAEs. Therefore, future work might consider how these insights could actually be used for tuning noise levels and predicting the training time of DAEs. This would require further validation and empirical experiments, also on other datasets. Finally, our analysis only considers the training dynamics, while a better understanding of generalisation and what influences the quality of feature representations during testing, are also of prime importance.

## Acknowledgements

We would like to thank Andrew Saxe for early discussions that got us interested in this work, as well as the reviewers for insightful comments and suggestions. We would like to thank the CSIR/SU Centre for Artificial Intelligence Research (CAIR), South Africa, for financial support. AP would also like to thank the MIH Media Lab at Stellenbosch University and Praelix (Pty) Ltd for providing stimulating working environments for a portion of this work.

## References

- Advani, M. S. and Saxe, A. M. High-dimensional dynamics of generalization error in neural networks. *arXiv:1710.03667*, 2017.
- Alain, G. and Bengio, Y. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.
- Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.
- Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pp. 153–160, 2007.
- Bengio, Y., Yao, L., Alain, G., and Vincent, P. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pp. 899–907, 2013.
- Bishop, C. M. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.
- Chandar, S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., and Saha, A. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pp. 1853–1861, 2014.
- Chen, M., Weinberger, K., Sha, F., and Bengio, Y. Marginalized denoising auto-encoders for nonlinear representations. In *International Conference on Machine Learning*, pp. 1476–1484, 2014.
- Chen, M., Weinberger, K., Xu, Z., and Sha, F. Marginalizing stacked linear denoising autoencoders. *Journal of Machine Learning Research*, 16:3849–3875, 2015.
- Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. *arXiv:1703.04933*, 2017.
- Elman, J. L. and Zipser, D. Learning the hidden structure of speech. *Journal of the Acoustic Society of America*, 83: 1615–1626, 1987.
- Krogh, A. and Hertz, J. A. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, pp. 950–957, 1992.
- Larsson, G. Discovery of visual semantics by unsupervised and self-supervised representation learning. *arXiv:1708.05812*, 2017.
- Lin, H. W., Tegmark, M., and Rolnick, D. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168:1223–1247, 2017.
- Louart, C., Liao, Z., and Couillet, R. A random matrix approach to neural networks, 2017. *arXiv:1702.05419v2*.
- Neyshabur, B., Tomioka, R., Salakhutdinov, R., and Srebro, N. Geometry of optimization and implicit regularization in deep learning. *arXiv:1705.03071*, 2017.
- Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. *arXiv:1704.08045*, 2017.
- Opper, M. Learning times of neural networks: exact solution for a perceptron algorithm. *Physical Review A*, 38(7): 3824, 1988.
- Pennington, J. and Bahri, Y. Geometry of neural network loss surfaces via random matrix theory. In *International Conference on Machine Learning*, pp. 2798–2806, 2017.
- Pennington, J. and Worah, P. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pp. 2634–2643, 2017.
- Pennington, J., Schoenholz, S., and Ganguli, S. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Processing Systems*, pp. 4788–4798, 2017.
- Poole, B., Sohl-Dickstein, J., and Ganguli, S. Analyzing noise in autoencoders and deep networks. *arXiv:1406.1831*, 2014.
- Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive auto-encoders: Explicit invariance during feature extraction. In *International Conference on Machine Learning*, pp. 833–840, 2011.
- Saad, D. and Solla, S. A. Exact solution for on-line learning in multilayer neural networks. *Physical Review Letters*, 74(21):4337, 1995.
- Sanger, T. D. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989.
- Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120*, 2013a.
- Saxe, A. M., McClelland, J. L., and Ganguli, S. Learning hierarchical category structure in deep neural networks. In *Proceedings of the Cognitive Science Society*, pp. 1271–1276, 2013b.
- Soudry, D. and Hoffer, E. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv:1702.05777*, 2017.

- Swirszcz, G., Czarnecki, W. M., and Pascanu, R. Local minima in training of neural networks. *arXiv:1611.06310*, 2017.
- Tu, Z., Liu, Y., Shang, L., Liu, X., and Li, H. Neural machine translation with reconstruction. In *AAAI Conference on Artificial Intelligence*, pp. 3097–3103, 2017.
- van der Maaten, L., Chen, M., Tyree, S., and Weinberger, K. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pp. 410–418, 2013.
- Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.
- Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pp. 1096–1103, 2008.
- Wager, S., Wang, S., and Liang, P. S. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pp. 351–359, 2013.
- Wang, S. and Manning, C. Fast dropout training. In *International Conference on Machine Learning*, pp. 118–126, 2013.
- Zeiler, M., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., and Hinton, G. E. On rectified linear units for speech processing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

## Supplementary material

The following section provides detail omitted in the paper regarding the derivation of certain equations as well as additional comments.

### A. Expected loss for linear DAEs

We derive the expected reconstruction loss over the noise distribution as presented in (1) in the paper. The expected loss can be written as

$$\mathbb{E}_\epsilon[\mathcal{L}] = \frac{1}{2N} \sum_{i=1}^N \mathbb{E}_\epsilon [ \|\mathbf{x}_i - W_2 W_1 \tilde{\mathbf{x}}_i\|^2 ],$$

where  $\tilde{\mathbf{x}}_i = \mathbf{x}_i + \epsilon_i$ , with  $\epsilon$  sampled from an isotropic noise distribution with component variance  $s^2$ . Let  $SE(\tilde{\mathbf{x}}_i) = \|\mathbf{x}_i - W_2 W_1 \tilde{\mathbf{x}}_i\|^2$  and  $M = W_2 W_1$ . Then

$$\begin{aligned} \mathbb{E}_\epsilon [SE(\tilde{\mathbf{x}}_i)] &= \mathbb{E}_\epsilon [ \|(I - M)\mathbf{x}_i + M(\mathbf{x}_i - \tilde{\mathbf{x}}_i)\|^2 ] \\ &= SE(\mathbf{x}_i) + \mathbb{E}_\epsilon [ \|M(\mathbf{x}_i - \tilde{\mathbf{x}}_i)\|^2 ] \end{aligned}$$

because the cross product terms vanish, since  $\mathbb{E}_\epsilon [\tilde{\mathbf{x}}_i] = \mathbf{x}_i$ :

$$\begin{aligned} 0 &= \mathbb{E}_\epsilon [ \mathbf{x}_i^T (I - M)^T M (\mathbf{x}_i - \tilde{\mathbf{x}}_i) ] \\ &= \mathbb{E}_\epsilon [ (\mathbf{x}_i - \tilde{\mathbf{x}}_i)^T M^T (I - M) \mathbf{x}_i ]. \end{aligned}$$

We also have that

$$\begin{aligned} \|M(\mathbf{x}_i - \tilde{\mathbf{x}}_i)\|^2 &= (\mathbf{x}_i - \tilde{\mathbf{x}}_i)^T M^T M (\mathbf{x}_i - \tilde{\mathbf{x}}_i) \\ &= \text{tr} [ (\mathbf{x}_i - \tilde{\mathbf{x}}_i)^T M^T M (\mathbf{x}_i - \tilde{\mathbf{x}}_i) ] \\ &= \text{tr} [ M(\mathbf{x}_i - \tilde{\mathbf{x}}_i)(\mathbf{x}_i - \tilde{\mathbf{x}}_i)^T M^T ] \\ &= \text{tr} [ M \epsilon_i \epsilon_i^T M^T ] \end{aligned}$$

due to the invariance of the trace under cycle permutation of products. Therefore, in expectation over the noise we have

$$\mathbb{E}_\epsilon [ \|M(\mathbf{x}_i - \tilde{\mathbf{x}}_i)\|^2 ] = \text{tr} [ M(s^2 I) M^T ],$$

and as a result

$$\begin{aligned} \mathbb{E}_\epsilon [\mathcal{L}] &= \frac{1}{2N} \sum_{i=1}^N \|\mathbf{x}_i - W_2 W_1 \mathbf{x}_i\|^2 \\ &\quad + \frac{s^2}{2} \text{tr} (W_2 W_1 W_1^T W_2^T). \end{aligned}$$

### B. Learning dynamics for linear DAEs

We derive the expression for the learning dynamics of a linear DAE as presented in (5) in the paper. As departure point, we start by examining the expected scalar update equations over the noise model for a small learning rate  $\alpha$ , which can be written as

$$\begin{aligned} \tau \frac{d}{dt} w_1 &= w_2 (\lambda - w_2 w_1 \lambda) - \epsilon w_2^2 w_1 \\ \tau \frac{d}{dt} w_2 &= w_1 (\lambda - w_2 w_1 \lambda) - \epsilon w_2 w_1^2. \end{aligned}$$

where  $\tau = \frac{N}{\alpha}$ , with  $N$  representing the number of training samples. Define  $w = w_2 w_1$  and using the product rule the update for  $w$  then becomes

$$\begin{aligned} \tau \frac{d}{dt} w &= \tau [ w_1 \frac{d}{dt} w_2 + w_2 \frac{d}{dt} w_1 ] \\ &= w_1^2 (\lambda - w_2 w_1 (\lambda + \epsilon)) + w_2^2 (\lambda - w_2 w_1 (\lambda + \epsilon)) \\ &= (\lambda - w (\lambda + \epsilon)) (w_1^2 + w_2^2). \end{aligned} \quad (9)$$

Next we make the following hyperbolic change of coordinates

$$\begin{aligned} w_1 &= \sqrt{c_0} \sinh\left(\frac{\theta}{2}\right), w_2 = \sqrt{c_0} \cosh\left(\frac{\theta}{2}\right), \text{ for } w_1^2 < w_2^2 \\ w_1 &= \sqrt{c_0} \cosh\left(\frac{\theta}{2}\right), w_2 = \sqrt{c_0} \sinh\left(\frac{\theta}{2}\right), \text{ for } w_1^2 > w_2^2, \end{aligned}$$

where  $\theta$  parameterises points along the dynamics trajectory represented by  $w_2^2 - w_1^2 = \pm c_0$  (Saxe et al., 2013a). Note that with this change of coordinates we obtain

$$\begin{aligned} w &= c_0 \cosh\left(\frac{\theta}{2}\right) \sinh\left(\frac{\theta}{2}\right) \\ &= c_0 \left( \frac{e^{\frac{\theta}{2}} + e^{-\frac{\theta}{2}}}{2} \right) \left( \frac{e^{\frac{\theta}{2}} - e^{-\frac{\theta}{2}}}{2} \right) \\ &= \frac{c_0}{2} \left( \frac{e^\theta - e^{-\theta}}{2} \right) \\ &= \frac{c_0}{2} \sinh(\theta), \end{aligned}$$

so that

$$dw = \frac{c_0}{2} \cosh(\theta) d\theta.$$

Similarly,

$$\begin{aligned} w_2^2 + w_1^2 &= c_0 \cosh^2\left(\frac{\theta}{2}\right) + c_0 \sinh^2\left(\frac{\theta}{2}\right) \\ &= c_0 \left( \frac{e^{\frac{\theta}{2}} + e^{-\frac{\theta}{2}}}{2} \right)^2 + c_0 \left( \frac{e^{\frac{\theta}{2}} - e^{-\frac{\theta}{2}}}{2} \right)^2 \\ &= \frac{c_0}{4} (e^\theta + 2 + e^{-\theta} + e^\theta - 2 + e^{-\theta}) \\ &= c_0 \left( \frac{e^\theta + e^{-\theta}}{2} \right) \\ &= c_0 \cosh(\theta) \end{aligned}$$

Plugging these results into the update for  $w$  given in (9), yields

$$\frac{\tau c_0 \cosh(\theta)}{2} \frac{d\theta}{dt} = \left( \lambda - \frac{c_0}{2} \sinh(\theta) (\lambda + \epsilon) \right) c_0 \cosh(\theta),$$

and as a result,

$$\tau \frac{d\theta}{dt} = \lambda (2 - \beta \sinh(\theta)),$$

where  $\beta = c_0 \left(1 + \frac{\varepsilon}{\lambda}\right)$ . To solve for  $t$ , we write

$$t = \int_{\theta_0}^{\theta_f} \frac{\tau}{\lambda (2 - \beta \sinh(\theta))} d\theta$$

and integrate:

$$t = \frac{\tau}{\zeta \lambda} \left[ \ln \left( \frac{\zeta + \beta + 2 \tanh(\frac{\theta}{2})}{\zeta - \beta - 2 \tanh(\frac{\theta}{2})} \right) \right]_{\theta_0}^{\theta_f}$$

where  $\zeta = \sqrt{\beta^2 + 4}$  and initial parameter value  $\theta_0 = \sinh^{-1}(2w/c_0)$ . Let  $\delta_0 = \tanh(\frac{\theta_0}{2})$  and  $\delta_f = \tanh(\frac{\theta_f}{2})$ , then

$$t = \frac{\tau}{\lambda \zeta} \ln \frac{(\zeta + \beta + 2\delta_f)(\zeta - \beta - 2\delta_0)}{(\zeta - \beta - 2\delta_f)(\zeta + \beta + 2\delta_0)},$$

so that

$$e^{\lambda \zeta t / \tau} = \frac{(\zeta + \beta + 2\delta_f)(\zeta - \beta - 2\delta_0)}{(\zeta - \beta - 2\delta_f)(\zeta + \beta + 2\delta_0)}.$$

Multiplying by the denominator, expanding, and defining  $E = e^{\lambda \zeta t / \tau}$ , we obtain

$$\begin{aligned} & -2E\delta_f(\zeta + \beta + 2\delta_0) \\ & + E(\zeta^2 + 2\zeta\delta_0 - \beta^2 - 2\beta\delta_0) \\ & = 2\delta_f(\zeta - \beta - 2\delta_0) \\ & + (\zeta^2 - 2\zeta\delta_0 - \beta^2 - 2\beta\delta_0), \end{aligned}$$

which yields

$$\begin{aligned} & \delta_f((1-E)(2\beta + 4\delta_0) - 2(E+1)\zeta) \\ & = (1-E)(\zeta^2 - \beta^2 - 2\beta\delta_0) - 2(1+E)\zeta\delta_0. \end{aligned}$$

Solving for  $\theta_f(t)$ , we obtain the hyperbolic parameter equation

$$\theta_f(t) = 2 \tanh^{-1} \left[ \frac{(1-E)(\zeta^2 - \beta^2 - 2\beta\delta) - 2(1+E)\zeta\delta}{(1-E)(2\beta + 4\delta) - 2(1+E)\zeta} \right]$$

where  $\delta = \tanh(\frac{\theta_0}{2})$ . Using

$$w(t) = \frac{c_0}{2} \sinh(\theta_t),$$

(where  $\theta_t = \theta_f(t)$ ) to track the weight trajectory gives equation (5) in the paper.

### C. Learning dynamics for linear WDAEs

We derive the expression for the learning dynamics of a linear WDAE as presented in (7) in the paper. Reconstruction

loss with weight decay gives the scalar loss associated with an eigenvalue  $\lambda$  as

$$\ell_\gamma = \frac{\lambda}{2\tau} (1 - w_2 w_1)^2 + \frac{N\gamma}{2\tau} (w_1^2 + w_2^2),$$

where  $\gamma$  is the penalty parameter that controls the amount of regularisation that is being applied. The update equations for the weights then follow as

$$\tau \frac{d}{dt} w_1 = w_2(\lambda - w_2 w_1 \lambda) - N\gamma w_1$$

$$\tau \frac{d}{dt} w_2 = w_1(\lambda - w_2 w_1 \lambda) - N\gamma w_2,$$

assuming the initial  $w_2 = w_1$  (which holds approximately for small initial values), we have for  $w = w_2 w_1$  that

$$\begin{aligned} \tau \frac{d}{dt} w & = 2w(\lambda - w\lambda) - 2N\gamma w \\ & = 2w(\lambda - N\gamma - w\lambda). \end{aligned}$$

Thus,

$$\begin{aligned} t & = \int_{w_0}^{w_f} \frac{\tau}{2w(\lambda - N\gamma - w\lambda)} dw \\ & = \frac{\tau}{2} \left[ \frac{\ln(w) - \ln(\lambda - N\gamma - w\lambda)}{\lambda - N\gamma} \right]_{w_0}^{w_f} \\ & = \frac{\tau}{2(\lambda - N\gamma)} \ln \left( \frac{w_f(\lambda - N\gamma - w_0\lambda)}{w_0(\lambda - N\gamma - w_f\lambda)} \right). \end{aligned}$$

Then solving for  $w_f$  gives

$$w_f(t) = \frac{\xi E_\gamma}{E_\gamma - 1 + \xi/w_0},$$

where  $E_\gamma = e^{2\xi t/\tau}$  and  $\xi = (1 - N\gamma/\lambda)$ .

### D. Optimal learning rates

We derive expressions for the optimal learning rates for linear DAEs and WDAEs as presented in (8) in the paper. First, consider the expected scalar DAE loss

$$\ell_\varepsilon = \frac{\lambda}{2\tau} (1 - w_2 w_1)^2 + \frac{\varepsilon}{2\tau} (w_2 w_1)^2.$$

The Hessian of  $\ell_\varepsilon$  is given by

$$H = \begin{bmatrix} \frac{\partial^2 \ell_\varepsilon}{\partial w_1^2} & \frac{\partial^2 \ell_\varepsilon}{\partial w_1 \partial w_2} \\ \frac{\partial^2 \ell_\varepsilon}{\partial w_2 \partial w_1} & \frac{\partial^2 \ell_\varepsilon}{\partial w_2^2} \end{bmatrix},$$

where

$$\frac{\partial^2 \ell_\varepsilon}{\partial w_1^2} = \frac{w_2^2}{\tau} (\lambda + \varepsilon),$$

$$\frac{\partial^2 \ell_\varepsilon}{\partial w_2^2} = \frac{w_1^2}{\tau} (\lambda + \varepsilon),$$

$$\frac{\partial^2 \ell_\varepsilon}{\partial w_1 \partial w_2} = \frac{\partial^2 \ell_\varepsilon}{\partial w_2 \partial w_1} = \frac{2w_2 w_1}{\tau} (\lambda + \varepsilon) - \frac{\lambda}{\tau}.$$

Now, if we assume  $w_2 = w_1$ , and let  $a = \frac{\partial^2 \ell_\varepsilon}{\partial w_1^2} = \frac{\partial^2 \ell_\varepsilon}{\partial w_2^2}$  and  $b = \frac{\partial^2 \ell_\varepsilon}{\partial w_2 \partial w_1}$ , the eigenvalues for the Hessian can be shown to be  $\lambda_H = a - b$  or  $\lambda_H = a + b$ . The second order update for a single weight  $w$  at time  $t$  is then given by

$$w^{t+1} = w^t - \left( \frac{\partial \ell_\varepsilon}{\partial w^t} \right) / \lambda_H,$$

where the maximum  $\lambda_H$ , is when  $w_2 = w_1 = 1$ , such that

$$\begin{aligned} \lambda_H &= \frac{1}{\tau}(\lambda + \varepsilon) + \frac{2}{\tau}(\lambda + \varepsilon) - \frac{\lambda}{\tau} \\ &= \frac{2\lambda + 3\varepsilon}{\tau}. \end{aligned}$$

Therefore, the optimal learning rate is

$$\alpha_\varepsilon = 1/\lambda_H = \frac{\tau}{2\lambda + 3\varepsilon}.$$

For WDAEs with penalty parameter  $\gamma$ , a very similar derivation gives

$$\alpha_\gamma = \frac{\tau}{2\lambda + \gamma}.$$

Taking the ratio of the optimal DAE rate to that for the WDAE gives

$$R = \frac{\alpha_\varepsilon}{\alpha_\gamma} = \frac{2\lambda + \gamma}{2\lambda + 3\varepsilon}.$$

## E. Equivalent scalar solutions

In Section 4 of the paper, the DAE fixed point solution is shown to be

$$w_\varepsilon^* = \frac{\lambda}{\lambda + \varepsilon}.$$

Now if  $w = w_2 w_1$  and  $w_2 = w_1$ , then for WDAE we have that the scalar loss is given by

$$\ell_\gamma = \frac{\lambda}{2\tau}(1-w)^2 + \frac{\gamma}{\tau}w,$$

and

$$\frac{\partial \ell_\gamma}{\partial w} = -\frac{\lambda}{\tau}(1-w) + \frac{\gamma}{\tau}.$$

Setting the above equal to zero and solving gives

$$w_\gamma^* = 1 - \gamma/\lambda.$$

To obtain the value of  $\gamma$  for which the two fixed points are equal, we set  $w_\gamma^* = w_\varepsilon^*$  and solve for  $\gamma$  to find

$$\gamma = \frac{\lambda\varepsilon}{\lambda + \varepsilon}.$$

## F. Estimated dynamics for nonlinear networks

The dynamics for the nonlinear networks trained in Figure 6 in the paper were estimated using the following approach. First, compute

$$\Sigma_{xx} = \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T = V \Lambda V^T,$$

using an eigen-decomposition giving eigenvalues  $\lambda_j$ ,  $j = 1, \dots, D$ . Then at regular intervals compute

$$\hat{\Sigma}_{xx}(t) = \sum_{i=1}^N \mathbf{x}_i \hat{\mathbf{x}}_i(t)^T,$$

where  $\hat{\mathbf{x}}(t)$  is the estimated reconstruction of input at time  $t$  generated by the autoencoder network. Finally, using the following rotation to obtain the diagonal matrix

$$\hat{\Lambda}(t) = V^T \hat{\Sigma}_{xx}(t) V,$$

where the diagonal contains the estimated eigenvalues  $\hat{\lambda}_j(t)$ , we can compute an estimate for the identity mapping associated with each eigenvalue as  $\hat{\lambda}_j(t)/\lambda_j \in [0, 1]$ .

## G. Learning dynamics for tanh autoencoder networks

We investigated the dynamics of learning for nonlinear AEs, WDAEs and DAEs, using tanh activations.

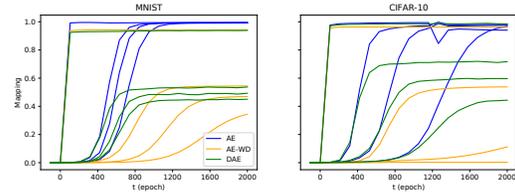


Figure 7. Learning dynamics for nonlinear networks using tanh activation. AE (blue), WDAE (orange) and DAE (green). **Left:** MNIST **Right:** CIFAR-10.

Figure 7 shows the dynamics for these networks trained on MNIST ( $N = 50000$ ) and CIFAR-10 ( $N = 30000$ ) with equal learning rates. For the DAE, the input was corrupted using sampled Gaussian noise with mean zero and  $\sigma^2 = 2$ . For the WDAE, the amount of weight decay was set to  $\gamma = 0.0045$ . During the course of training, the identity mapping associated with each eigenvalue was estimated using the approach described in Section F, at equally spaced intervals of size 100 epochs.