Modelling the influence of data structure on learning in neural networks: the hidden manifold model

Sebastian Goldt, Marc Mézard, Florent Krzakala, Lenka Zdeborová

Modelling the influence of data structure on learning in neural networks: the hidden manifold model

Sebastian Goldt*[1], Marc Mézard[2],
Florent Krzakala[2] and Lenka Zdeborová[1]

[1]Institut de Physique Théorique, CNRS, CEA, Université Paris-Saclay, France
[2]Laboratoire de Physique de l'Ecole Normale Supérieure, Université PSL, CNRS, Sorbonne Université, Université Paris-Diderot, Sorbonne Paris Cité, Paris, France

The lack of crisp mathematical models that capture the structure of real-world data sets is a major obstacle to the detailed theoretical understanding of deep neural networks. Here, we introduce a generative model for data sets that we call the *hidden manifold model* (HMM). The idea is to have high-dimensional inputs lie on a lower-dimensional manifold, with labels that depend only on their position within this manifold, akin to a single layer decoder or generator in a generative adversarial network. We first demonstrate the effect of structured data sets by experimentally comparing the dynamics and the performance of two-layer neural networks trained on three different data sets: (i) an unstructured synthetic data set containing random i.i.d. inputs, (ii) a structured data set drawn from the HMM and (iii) a simple canonical data set containing MNIST images. We pinpoint two phenomena related to the dynamics of the networks and their ability to generalise that only appear when training on structured data sets, and we experimentally demonstrate that training networks on data sets drawn from the HMM reproduces both the phenomena seen during training on real dataset. Our main theoretical result is that we show that the learning dynamics in the hidden manifold model is amenable to an analytical treatment by proving a "Gaussian Equivalence Theorem", opening the way to further detailed theoretical studies. In particular, we show how the dynamics of stochastic gradient descent for a two-layer network is captured by a set of ordinary differential equations that track the generalisation error at all times.

*goldt.sebastian@gmail.com.

1. Introduction

A major impediment for understanding the effectiveness of deep neural networks is our lack of mathematical models for the data sets on which neural networks are trained. This lack of tractable models prevents us from analysing the impact of data sets on the training of neural networks and their ability to generalise from examples, which remains an open problem both in statistical learning theory [1, 2], and in analysing the average-case behaviour of algorithms in synthetic data models [3–5].

Indeed, most theoretical results on neural networks do not model the structure of the training data, while some works build on a setup where inputs are drawn component-wise i.i.d. from some probability distribution, and labels are either random or given by some random, but fixed function of the inputs. Despite providing valuable insights, these approaches are by construction blind to key structural properties of real-world data sets.

Our goal in this paper is to consider a model, the Hidden Manifold Model, amenable to analytical studies, that will capture the most important features of real data sets. We shall show in particular that one can analytically study the learning dynamics in this problem.

To motivate the model, we focus on two types of data structure that can both already be illustrated by considering perhaps the simplest canonical problem of supervised machine learning: classifying the handwritten digits in the MNIST database using a neural network $\mathcal{N}$ [6]. The input patterns are images with 28×28 pixels, so *a priori* we work in the high-dimensional $\mathbb{R}^{784}$. However, the inputs that may be interpreted as handwritten digits, and hence constitute the "world" of our problem, span but a lower-dimensional manifold within $\mathbb{R}^{784}$ which is not easily defined. Its dimension can nevertheless be estimated to be around $D \approx 14$ based on the neighbourhoods of inputs in the data set [7–10]. The intrinsic dimension being lower than the dimension of the input space is a property expected to be common to many real data sets used in machine learning. We should not consider presenting $\mathcal{N}$ with an input that is outside of its world (or maybe we should train it to answer that the "input is outside of my world" in such cases). We will call inputs *structured* if they are concentrated on a lower-dimensional manifold and thus have a lower-dimensional latent representation, which consists of the position of the input on that manifold.

The second type of structure concerns the function of the inputs that is to be learnt, which we will call the learning *task*. We will consider two models: the teacher task, where the label is obtained as a function of the high-dimensional input; and the latent task, where the label is a function of only the lower-dimensional latent representation of the input.

1.1. Main contributions and related work

1. *We experimentally pinpoint two key differences between networks trained in the vanilla teacher-student setup and networks trained on the MNIST task (Sec. 2).* i) Two identical networks trained on the same MNIST task, but starting from different initial conditions, will achieve the same test error on MNIST images, but they learn globally different functions. Their outputs coincide in those regions of input space where MNIST images tend to lie – the "world" of the problem, but differ significantly when tested on Gaussian inputs. In contrast, two networks trained on the teacher task learn the same functions globally to within a small error. ii) In the vanilla teacher-student setup, the test error of a network is stationary during long periods of training before a sudden drop-off. These plateaus are well-known features of this setup [4, 12], but are not observed when training on the MNIST task nor on other data sets used commonly in machine learning.

2. *We introduce the* hidden manifold model *(HMM), a probabilistic model that generates data sets containing high-dimensional inputs which lie on a lower-dimensional manifold and whose labels depend only on their position within that manifold (Sec. 3).* This model is akin to a learnt single layer decoder with random input or a single layer generator of a learnt

structured inputs	inputs that are concentrated on a fixed, lower-dimensional manifold in input space
latent representation	for a structured input, its coordinates in the lower-dimensional manifold
task	the function of the inputs to be learnt
latent task	for structured inputs, labels are given as a function of the latent representation only
teacher task	for all inputs, labels are obtained from a random, but fixed function of the high-dimensional input without explicit dependence on the latent representation, if it exists
MNIST task	discriminating odd from even digits in the MNIST database
vanilla teacher-student setup	Generative model due to [11], where data sets consist of component-wise i.i.d. inputs with labels given by a fixed, but random neural network acting directly on the input
hidden manifold model (HMM)	Generative model introduced in Sec. 3 for data sets consisting of structured inputs (Eq. 6) with latent labels (Eq. 7)

Table 1: Several key concepts used/introduced in this paper.

generative adversarial network (GAN). The input samples, resulting from this model, are structured and their labels depend on their lower-dimensional latent representation only. We experimentally demonstrate that training networks on data sets drawn from this model reproduces both behaviours observed when training on MNIST. Moreover we show that the model displays the recently widely discussed double-descent behaviour, again in agreement to what is observed in MNIST. We also show that the structure of both, input space and the task to be learnt, play an important role for the dynamics and the performance of neural networks.

3. *We show that the hidden manifold model can be studied analytically in a thermodynamic limit using the "Gaussian Equivalence Theorem" (GET) of Sec. 4.* This shows the HMM is not only a better approximation of real data sets, but is also amenable to exact analytical treatment. The GET also allows to study deterministic, or learnt, mapping from the hidden manifold to the data set. These properties, we believe, open the way to many further analytical studies of typical-case behaviour in machine learning problems.

4. *We use the GET to derive asymptotically exact ordinary differential equations governing the learning dynamics of stochastic gradient descent for online learning in the thermodynamic limit (Sec. 5).* These ODEs provide detailed insight into the dynamics of learning and form a starting point for numerous further investigations, and generalise to the HMM the classical analysis for unstructured data [12, 13].

Relation to feature learning and random matrix theory There exists an interesting relation – but also key differences – between the hidden manifold model (HMM) that we propose and random feature learning with unstructured i.i.d. input data [14–16]. Remarkably, random feature learning in the same scaling limit as used in the theoretical part of this paper was analysed in several recent and concurrent works, notably in [17, 18] for ridge regression, and in [19] for max-margin linear classifiers. These papers consider full batch learning, i.e. all samples are used at the same time, which makes one difference from our online (one-pass stochastic gradient descent) analysis.

Another important difference is that we study learning in a neural network with a hidden layer, while the existing works study simpler learning algorithms. Perhaps a more important difference is that in our analysis the features do not need to be random, they can be deterministic, or even be learnt from some data using a GAN or an autoencoder.

The principles underlying the analytic solution presented in the present paper, but also the one of [17–19], boil down to the Gaussian Equivalence Principle, which is stated and used independently in those papers. Special cases of the Gaussian Equivalence Theorem were in fact derived previously using random matrix theory in [20–23], and this equivalent Gaussian covariates mapping was explicitly stated and used in [18, 19]. Very recently, this has been further extended into a broader setting of concentrated vectors encompassing data coming from a GAN in [24, 25], a version closer to our formulation. We discuss the relation of our model to feature learning and these results in more detail in Sec. 6.

Further related work Several works have appreciated the need to model the inputs in the first place, and in particular the need to go beyond the simple component-wise i.i.d. modelling [26–30]. While we will focus on the ability of neural network to generalise from examples, two recent papers studied a network's ability to *store* inputs with lower-dimensional structure and random labels: Chung et al. [31] studied the linear separability of general, finite-dimensional manifolds and their interesting consequences for trained deep neural networks [32], while Rotondo et al. [33] extended Cover's argument [34] to count the number of learnable dichotomies when inputs are grouped in tuples of k inputs with the same label. Recently, Yoshida and Okada analysed the dynamics of online learning for data having an arbitrary covariance matrix, finding an infinite hierarchy of ODE and finding a reduction of the plateau [35].

We also note that several works have compared neural networks trained from different initial conditions on the same task by comparing the different features learnt in vision problems [36–38], but these works did not compare the *functions* learnt by the network.

Accessibility and reproducibility We provide the full code of our experiments and our implementation of the ODEs describing online learning at `https://github.com/sgoldt/hidden-manifold-model` and give necessary parameter values to reproduce our figures beneath each plot.

2. Learning on structured data sets versus unstructured teacher-student model

In this section we compare neural networks trained on two different problems: the *MNIST task*, where one aims to discriminate odd from even digits in the MNIST data set; and the *vanilla teacher-student setup*. In this setup, inputs are drawn as vectors with i.i.d. component from the Gaussian distribution and labels are given by a random, but fixed, neural network acting on the high-dimensional inputs. This model is an example of a teacher task on unstructured inputs. It was introduced by Gardner & Derrida [11] and has played a major role in theoretical studies of the generalisation ability of neural networks from an average-case perspective, particularly within the framework of statistical mechanics [3–5, 13, 39–43], and also in recent statistical learning theory works, e.g. [18, 44–46]. We choose the MNIST data set because it is the simplest widely used example of a structured data set on which neural networks show significantly different behaviour than when trained on synthetic data of the vanilla teacher-student setup. In appendix D.1 we show that the same phenomenology is observed for "fashion MNIST" data set and we expect in many others.

2.1. Learning setup

In order to proceed on the question of what is a suitable model for structured data, we consider the setup of a feedforward neural network with one hidden layer with a few hidden units, as described below. So throughout this work, we focus on the dynamics and performance of fully-connected two-layer neural networks with K hidden units and first- and second-layer weights $\boldsymbol{W} \in \mathbb{R}^{K\times N}$ and $\boldsymbol{v} \in \mathbb{R}^K$, resp. Given an input $\boldsymbol{x} \in \mathbb{R}^N$, the output of a network with parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{v})$ is given by

$$\phi(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_k^K v_k g\left(\boldsymbol{w}_k \boldsymbol{x}/\sqrt{N}\right), \tag{1}$$

where $\boldsymbol{w}_k$ is the kth row of $\boldsymbol{W}$, and $g : \mathbb{R} \to \mathbb{R}$ is the non-linear activation function of the network, acting component-wise. We will focus on sigmoidal networks with $g(x) = \mathrm{erf}(x/\sqrt{2})$, or ReLU networks where $g(x) = \max(0, x)$ (see Appendix D.5).

We will train the neural network on data sets with P input-output pairs $(\boldsymbol{x}_\mu, y^*_\mu)$, $\mu = 1, \ldots, P$, where we use the starred y^*_μ to denote the *true* label of an input $\boldsymbol{x}_\mu$. We train networks by minimising the quadratic training error $E(\boldsymbol{\theta}) = 1/2 \sum_{\mu=1}^P \Delta_\mu^2$ with $\Delta_\mu = \left[\phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}) - y^*_\mu\right]$ using stochastic gradient descent (SGD) with constant learning rate η,

$$\boldsymbol{\theta}_{\mu+1} = \boldsymbol{\theta}_\mu - \eta \nabla_{\boldsymbol{\theta}} E(\theta)|_{\boldsymbol{\theta}_\mu, \boldsymbol{x}_\mu, y^*_\mu}. \tag{2}$$

Initial weights for both layers of sigmoidal networks were always taken component-wise i.i.d. from the normal distribution with mean 0 and variance 1. The initial weights of ReLU networks were also taken from the normal distribution, but with variance 10^{-6} to ensure convergence.

The key quantity of interest is the *test error* or *generalisation error* of a network, for which we compare its predictions to the labels given in a test set that is composed of P^* input-output pairs $(\boldsymbol{x}_\mu, y^*_\mu)$, $\mu = 1, \ldots, P^*$ that are *not* used during training,

$$\epsilon_g^{\mathrm{mse}}(\boldsymbol{\theta}) \equiv \frac{1}{2P^*} \sum_\mu^{P^*} \left[\phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}) - y^*_\mu\right]^2. \tag{3}$$

The test set might be composed of MNIST test images or generated by the same probabilistic model that generated the training data. For binary classification tasks with $y^* = \pm 1$, this definition is easily amended to give the fractional generalisation error $\epsilon_g^{\mathrm{frac}}(\boldsymbol{\theta}) \propto \sum_\mu^{P^*} \Theta\left[-\phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}) y^*_\mu\right]$, where $\Theta(\cdot)$ is the Heaviside step function.

2.2. Learning from real data or from generative models?

We want to compare the behaviours of two-layer neural networks Eq. (1) trained either on real data sets or on unstructured tasks. As an example of a real data set, we will use the MNIST image database of handwritten digits [6] and focus on the task of discriminating odd from even digits. Hence the inputs $\boldsymbol{x}_\mu$ will be the MNIST images with labels $y^*_\mu = 1, -1$ for odd and even digits, resp. The joint probability distribution of input-output pairs $(\boldsymbol{x}_\mu, y^*_\mu)$ for this task is inaccessible, which prevents analytical control over the test error and other quantities of interest. To make theoretical progress, it is therefore promising to study the generalisation ability of neural networks for data arising from a probabilistic generative model.

A classic model for data sets is the *vanilla teacher-student setup* [11], where unstructured i.i.d. inputs are fed through a random neural network called the *teacher*. We will take the teacher to have two layers and M hidden nodes. We allow that $M \neq K$ and we will draw the components of the teacher's weights $\boldsymbol{\theta}^* = (\boldsymbol{v}^* \in \mathbb{R}^M, \boldsymbol{W}^* \in \mathbb{R}^{M\times N})$ i.i.d. from the normal distribution with mean zero and unit variance. Drawing the inputs i.i.d. from the standard normal distribution $\mathcal{N}(\boldsymbol{x}; 0, \boldsymbol{I}_N)$, we will take

$$y^*_\mu = \phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}^*) \tag{4}$$

for regression tasks, or $y^*_\mu = \mathrm{sgn}\left[\phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}^*)\right]$ for binary classification tasks. This is hence an example of a teacher task. In this setting, the network with K hidden units that is trained using SGD Eq. (2) is traditionally called the *student*. Notice that, if $K \geq M$, there exist a student

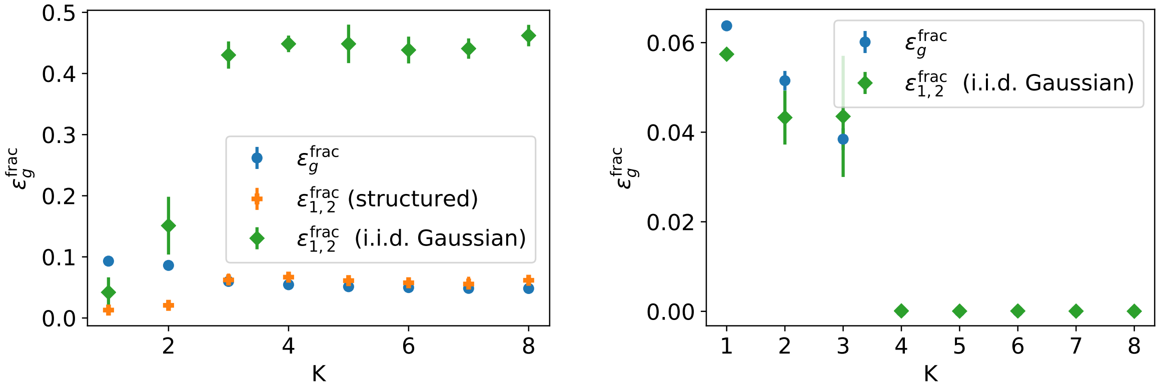


Figure 1: *(Left)* **Networks trained independently on MNIST achieve similar performance, but learn different functions.** For two networks trained independently on the MNIST odd-even classification task, we show the averaged final fractional test error, $\epsilon_g^{\mathrm{frac}}$ (blue dots). We also plot $\epsilon_{1,2}^{\mathrm{frac}}$ (5), the fraction of Gaussian i.i.d. inputs and MNIST test images the networks classify differently after training (green diamonds and orange crosses, resp.). *(Right)* **Training independent networks on a teacher task with i.i.d. inputs does not reproduce this behaviour.** We plot the results of the same experiment, but for Gaussian i.i.d. inputs with teacher labels y_μ^* (Eq. 4, $M = 4$). For both plots, $g(x) = \mathrm{erf}\left(x/\sqrt{2}\right)$, $\eta = 0.2$, $P^* = 76N$, $N = 784$.

network that has zero generalisation error, the one with the same architecture and parameters as the teacher.

We now proceed to demonstrate experimentally two significant differences in the dynamics and the performance of neural networks trained on realistic data sets and networks trained within the vanilla teacher-student setup.

2.3. Independent networks achieve similar performance, but learn different functions when trained on structured tasks

We trained two sigmoidal networks with K hidden units, starting from two independent draws of initial conditions to discriminate odd from even digits in the MNIST database. We trained both networks using SGD with constant learning rate η, Eq. (2), until the generalisation error had converged to a stationary value. We plot this asymptotic fractional test error $\epsilon_g^{\mathrm{frac}}$ as blue circles on the left in Fig. 1 (the averages are taken over both networks and over several realisations of the initial conditions). We observed the same qualitative behaviour when we employed the early-stopping error to evaluate the networks, where we take the minimum of the generalisation error during training (see Appendix D.3).

First, we note that increasing the number of hidden units in the network decreases the test error on this task. We also compared the networks to one another by counting the fraction of inputs which the two networks classify differently,

$$\epsilon_{1,2}^{\mathrm{frac}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \equiv \frac{1}{2P^*} \sum_{\mu}^{P^*} \Theta\left[-\phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}_1)\phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}_2)\right]. \tag{5}$$

This is a measure of the degree to which both networks have learned the same function $\phi(\boldsymbol{x}, \boldsymbol{\theta})$. Independent networks disagree on the classification of MNIST test images at a rate that roughly corresponds to their test error for $K \geq 3$ (orange crosses). However, even though the additional parameters of bigger networks are helpful in the discrimination task (decreasing ϵ_g), both networks learn increasingly different functions when evaluated over the whole of $\mathbb{R}^N$ using Gaussian inputs

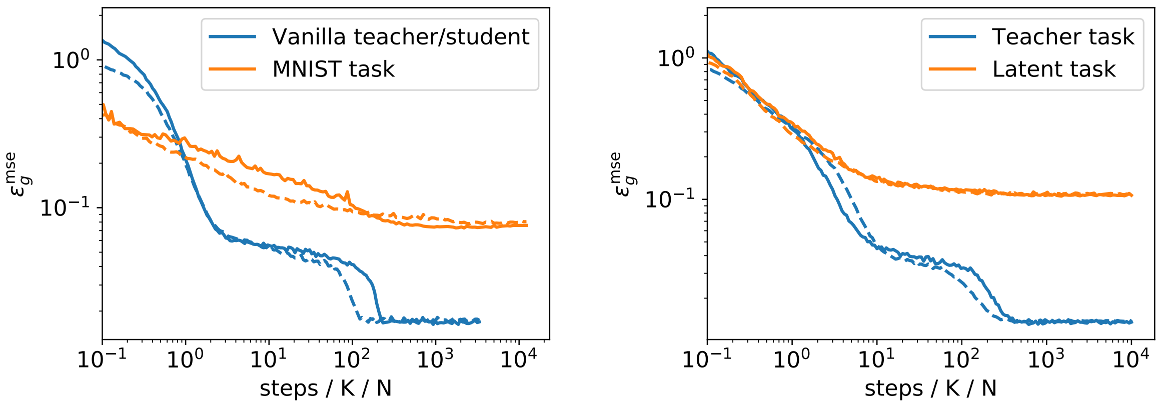


Figure 2: *(Left)* **Extended periods with stationary test error during training ("plateaus") appear in the vanilla teacher-student setup, not on MNIST.** We plot the generalisation error $\epsilon_g^{\mathrm{mse}}$ (3) of a network trained on Gaussian i.i.d. inputs with teacher labels (Eq. 4, blue) and when learning to discriminate odd from even digits in MNIST (orange). We trained either the first layer only (dashed) or both layers (solid). Notice the log scale on the x-axes. *(Right)* **Both structured inputs and latent labels are required to diminish the plateau for synthetic data.** Same experiment, but now the network is trained on structured inputs (Eq. 6) ($f(x) = \mathrm{sgn}(x), D = 10$), with teacher labels y_μ^* (Eq. 4, blue) and with latent labels $\widetilde{y}_\mu^*$ (Eq. 7, orange). In both plots, $g(x) = \mathrm{erf}\left(x/\sqrt{2}\right), N = 784, P = 76N, M = 4, K = 3, \eta = 0.2$.

as the network size K increases (green diamonds). The network learned the right function on the lower-dimensional manifold on which MNIST inputs concentrate, but not outside of it.

This behaviour is not reproduced if we substitute the MNIST data set with a data set of the same size drawn from the vanilla teacher-student setup from Sec. 2.2 with $M = 4$, leaving everything else the same (right of Fig. 1). The final test error decreases with K, and as soon as the expressive power of the network is at least equal to that of the teacher, *i.e.* $K \geq M$, the asymptotic test error goes to zero, since the data set is large enough for the network to recover the teacher's weights to within a very small error, leading to a small generalisation error. We also computed the $\epsilon_{1,2}^{\mathrm{frac}}$ evaluated using Gaussian i.i.d. inputs (green diamonds). Networks with fewer parameters than the teacher find different approximations to that function, yielding finite values of $\epsilon_{1,2}$. If they have just enough parameters ($K = M$), they learn the same function. Remarkably, they also learn the same function when they have significantly *more* parameters than the teacher. The vanilla teacher-student setup is thus unable to reproduce the behaviour observed when training on MNIST.

2.4. The generalisation error exhibits plateaus during training on i.i.d. inputs

We plot the generalisation dynamics, *i.e.* the test error as a function of training time, for neural networks of the form (1) in Fig. 2. For a data set drawn from the vanilla teacher-student setup with $M = 4$, (blue lines in the left-hand plot of Fig. 2), we observe that there is an extended period of training during which the test error ϵ_g remains constant before a sudden drop. These "plateaus" are well-known in the literature for both SGD, where they appear as a function of time [12, 47, 48], and in batch learning, where they appear as a function of the training set size [4, 49]. Their appearance is related to different stages of learning: After a brief exponential decay of the test error at the start of training, the network "believes" that data are linearly separable and all her hidden units have roughly the same overlap with all the teacher nodes. Only after a longer time, the network picks up the additional structure of the teacher and "specialises": each of its hidden units ideally becomes strongly correlated with one and only one hidden unit of the teacher before the generalisation error decreases exponentially to its final value.

In contrast, the generalisation dynamics of the same network trained on the MNIST task (orange trajectories on the left of Fig. 2) shows no plateau. In fact, plateaus are rarely seen during the training of neural networks (note that during training, we do not change any of the hyper-parameters, *e.g.* the learning rate η.)

It has been an open question how to eliminate the plateaus from the dynamics of neural networks trained in the teacher-student setup. The use of second-order gradient descent methods such as natural gradient descent [50] can shorten the plateau [51], but we would like to focus on the more practically relevant case of first-order SGD. Yoshida et al. [43] recently showed that the length and the existence of the plateau depend on the dimensionality of the output of the network, but our aim is to build a model where the plateau disappears independently of the output dimension.

3. The hidden manifold model

3.1. Definition

We now introduce a new generative probabilistic model for structured data sets with the aim of reproducing the behaviour observed during training on MNIST, but with a synthetic data set. The main motivation for introducing such a model is that it is possible to derive a closed-form solution of the learning dynamics, allowing for a detailed analytical study shown in Section 5.

To generate a data set containing P inputs in N dimensions, we first choose D feature vectors $\boldsymbol{f}_r$, $r = 1, \ldots, D$. These are vectors in N dimensions and we collect them in a feature matrix $\boldsymbol{F} \in \mathbb{R}^{D\times N}$. Next we draw P vectors $\boldsymbol{c}_\mu$ with random i.i.d. components and collect them in the matrix $\boldsymbol{C} \in \mathbb{R}^{P\times D}$. The vector $\boldsymbol{c}_\mu$ gives the coordinates of the μth input on the lower-dimensional manifold spanned by the feature vectors in $\boldsymbol{F}$. We will call $\boldsymbol{c}_\mu$ the *latent representation* of the input $\boldsymbol{x}_\mu$, which is given by the μth row of

$$\boldsymbol{X} = f\left(\boldsymbol{C}\boldsymbol{F}/\sqrt{D}\right) \in \mathbb{R}^{P\times N}, \tag{6}$$

where f is a non-linear function acting component-wise. In this model, the “world” of the data on which the true label can depend is a D-dimensional manifold, which is obtained from the linear subspace of $\mathbb{R}^N$ generated by the D lines of matrix $\boldsymbol{F}$, through a folding process induced by the nonlinear function f. We note that the structure of data of the same type arises in a learned variational autoencoder network [52] with single layer, or in a learned GAN network [53] with a single layer generator network, the matrix C then corresponds to the random input, the F to the learned features, f is the corresponding output activation. The exact form of f is not important. We shall exemplify this statement in Sect. 4 where we work out the exact solutions of the online learning dynamics in the “thermodynamic limit” where $N, D \to \infty$ with a fixed ratio: we will show explicitly that the whole learning dynamics depend on the folding function f only through three parameters (see Eqs. (22)).

The latent labels are obtained by applying a two-layer neural network with weights $\widetilde{\boldsymbol{\theta}} = (\widetilde{\boldsymbol{W}} \in \mathbb{R}^{M\times D}, \widetilde{\boldsymbol{v}} \in \mathbb{R}^M)$ within the unfolded hidden manifold according to

$$\widetilde{y}_\mu^* = \phi(\boldsymbol{c}_\mu, \widetilde{\boldsymbol{\theta}}) = \sum_m^M \tilde{v}^m \tilde{g}\left(\widetilde{\boldsymbol{w}}^m \boldsymbol{c}_\mu/\sqrt{D}\right). \tag{7}$$

We draw the weights in both layers component-wise i.i.d. from the normal distribution with unity variance, unless we note it otherwise. The key point here is the dependency of labels $\widetilde{y}_\mu^*$ on the coordinates of the lower-dimensional manifold $\boldsymbol{c}_\mu$ rather than on the high-dimensional data $\boldsymbol{x}_\mu$. The exact functional form of this dependence is not crucial for the empirical part of this work, there are other forms that would present the same behaviour, notably ones where the latent representation is conditioned to the labels as in conditional GANs [54] or the manifold model of [32]. For the analytical solution of the model in Section 5, the fact that the matrix C is

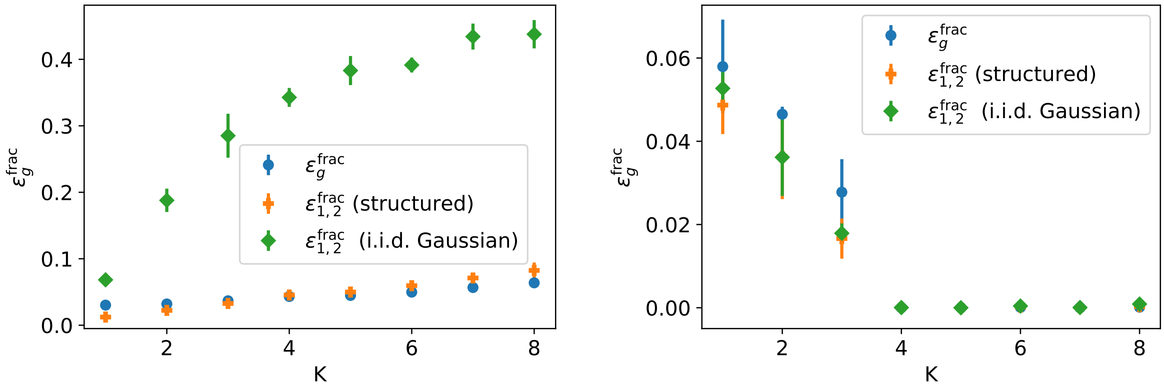


Figure 3: A latent task on structured inputs makes independent networks behave like networks trained on MNIST. *(Left)* For two networks trained independently on a binary classification task with structured inputs (6) and latent labels $\widetilde{y}_\mu^*$ (Eq. 7, $M = 1$), we plot the final fractional test error, $\epsilon_g^{\mathrm{frac}}$ (blue dots). We also plot $\epsilon_{1,2}^{\mathrm{frac}}$ (5), the fraction of Gaussian i.i.d. inputs and structured inputs the networks classify differently after training (green diamonds and orange crosses, resp.). *(Right)* In the same experiment, structured inputs with *teacher labels* y_μ^* (4) ($M = 4$) fail to reproduce the behaviour observed on MNIST (cf. Fig. 1). In both plots, $f(x) = \mathrm{sgn}(x), g(x) = \mathrm{erf}\left(x/\sqrt{2}\right), D = 10, \eta = 0.2.$

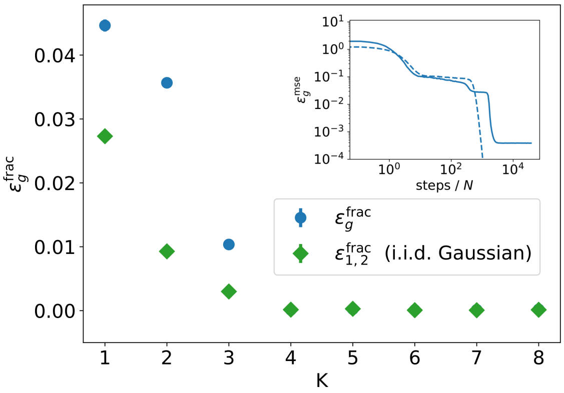
random i.i.d. and the labels are conditioned to it rather than the other way around simplifies the analysis.

In the numerical simulations of this section, we choose the entries of both $\boldsymbol{C}$ and $\boldsymbol{F}$ to be i.i.d. draws from the normal distribution with mean zero and unit variance. To ensure comparability of the data sets for different data-generating function $f(x)$, we always center the input matrix $\boldsymbol{X}$ by subtracting the mean value of the entire matrix from all components and we rescale inputs by dividing all entries by the covariance of the entire matrix before training. We stress at this point that our results including the analysis of Section 5 holds for deterministic (learned if needed) matrices F, we only require some balanced conditions stated in Eqs. (14-16)

3.2. Learning in the hidden manifold model

We repeated the experiments reported in Sec. 2.3 using data sets generated from the hidden manifold model with $D = 10$ latent dimensions (see Appendix D.4 for experiments with large D). On the right of Fig. 3, we plot the asymptotic performance of a network trained on structured inputs which lie on a manifold (6) with a "teacher-task" as in (4): the labels are a function of the high-dimensional inputs, $y_\mu^* = \phi(\boldsymbol{x}_\mu, \boldsymbol{\theta}^*)$, and they do not depend explicitly on the latent representation $\boldsymbol{c}_\mu$. In this case, the final results are similar to those of networks trained on data from the vanilla teacher-student setup (*cf.* left of Fig. 1): given enough data, the network recovers the teacher function if the network has at least as many parameters as the teacher. Once the teacher weights are recovered by both networks, they achieve zero test error (blue circles) and they agree on the classification of random Gaussian inputs because they do implement the same function.

The left plot of Fig. 3 shows network performance when trained on the same inputs, but this time with a "latent-task" where the labels are a function of the latent representation of the inputs: $\widetilde{y}_\mu^* = \phi(\boldsymbol{c}_\mu, \widetilde{\boldsymbol{\theta}}^*)$. The asymptotic performance of the networks then resembles that of networks trained on MNIST: after convergence, the two networks disagree on structured inputs at a rate that is roughly their generalisation error, but as K increases, they also learn increasingly different functions, up to the point where they will agree on their classification of a random Gaussian input in just half the cases. The hidden manifold model thus reproduces the behaviour



#	*inputs* $\boldsymbol{x}_i$	*task* y_i^*
(1)	i.i.d. Gaussian	teacher task, Eq. (4)
(2)	i.i.d. Gaussian	teacher trained on MNIST, Eq. (4)
(3)	structured, Eq. (6)	teacher task, Eq. (4)
(4)	structured, Eq. (6)	latent task, Eq. (7)

Figure 4: *(Left)* Same plot as the right plot of Fig. 1 with Gaussian i.i.d. inputs $\boldsymbol{x}_\mu$ and labels y_μ^* (4) provided by a teacher network with $M = 4$ hidden units that was pre-trained on the MNIST task, reaching $\sim 5\%$ on the task. *Inset:* Typical generalisation dynamics of networks where we train the first or both layers (dashed and solid, resp.). $g(x) = \operatorname{erf}\left(x/\sqrt{2}\right), \eta = 0.2, N = 784, M = K = 4, P^* = 76N$. *(Right)* Four different setups for synthetic data sets in supervised learning problems.

of independent networks trained on MNIST.

We now look at the learning dynamics. Again, we repeat the experiment of Sec. 2.4, but we train networks on structured inputs $\boldsymbol{X} = \operatorname{sgn}(\boldsymbol{CF})$ with teacher-task (y_μ^*) and latent-task ($\widetilde{y}_\mu^*$), respectively. It is clear from Fig. 2 that the plateaus that are present in the teacher-task are no longer seen when going to a latent-task. In Appendix D.2, we demonstrate that the lack of plateaus for latent-tasks in Fig. 2 is *not* due to the fact that the network in the latent-task asymptotes at a higher generalisation error than the teacher task. We will come back to the plateau phenomenon in greater detail once we have derived the ODEs for online learning in Sec. 5.

3.3. Latent-tasks and hidden-manifold inputs model real data sets

Our quest to reproduce the behaviour of networks trained on MNIST has led us to consider three different setups so far: the vanilla teacher-student setup, *i.e.* a teacher-task on unstructured inputs; and teacher- and latent- tasks on structured inputs lying in a hidden manifold. While it is not strictly possible to test the case of a latent-task with unstructured inputs, we can approximate this setup by training a network on the MNIST task and then using the resulting network as a teacher to generate labels y_μ^* (4) for inputs drawn i.i.d. component-wise from the standard normal distribution. To test this idea, we trained both layers sigmoidal networks with $M = 4$ hidden units using vanilla SGD on the MNIST task, where they reach a generalisation error of about 5%. They have thus clearly learnt some of the structure of the MNIST task. However, as we show on the left of Fig. 4, independent students trained on a data set with i.i.d. Gaussian inputs $\boldsymbol{x}_\mu$ and true labels y_μ^* given by the pre-trained teacher network behave similarly to students trained in the vanilla teacher-student setup of Sec. 2.3. Furthermore, the learning dynamics of a network trained in this setup display the plateaus that we observed in the vanilla teacher-student setup (inset of Fig. 4).

On the right of Fig. 4, we summarise the four different setups for synthetic data sets in supervised learning problems that we have analysed in this paper. Out of these four, only the hidden manifold model, consisting of a latent task on structured inputs, reproduced the behaviour of neural networks trained on the MNIST task. We anticipate that other models and label generative processes would reproduce the empirical behaviours observed in this paper, the main advantage of the specific model defined here is its analytic tractability.

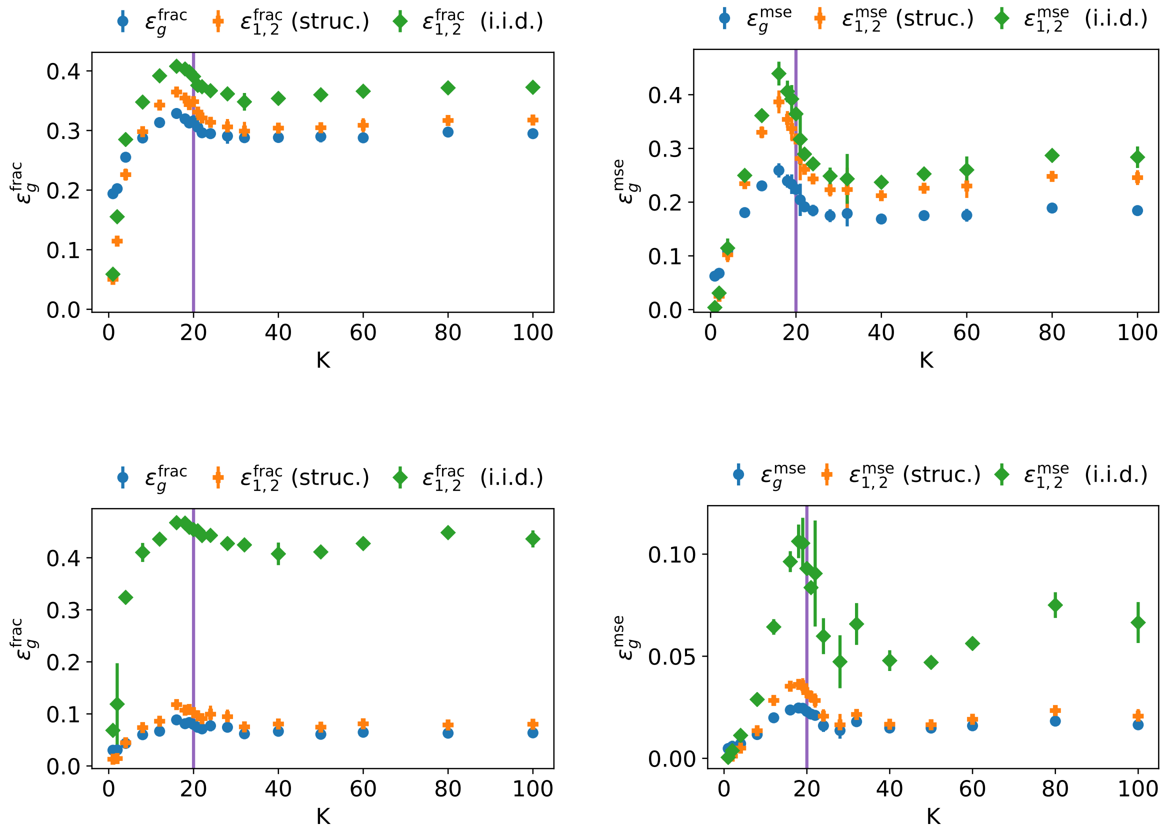


Figure 5: The double descent phenomenology in the hidden manifold model with ($D = 250$, top) and ($D = 10$, bottom) latent dimensions. Fractional and mean-squared error for students (left and right, resp.). For both plots, $g(x) = \tilde{g}(x) = \operatorname{erf}\left(x/\sqrt{2}\right), \eta = 0.2, P = 20N, N = 500, M = 1$, Gaussian initial weights with std. dev. 10^{-3}.

3.4. Double descent phenomenology in the hidden manifold model

We also point out that training two-layer neural network on the data from the hidden manifold model presents the double descent phenomenon: when more and more parameters are added to the network the generalisation error first goes up at around a point where the number of parameters corresponds to the number of samples (vertical lines in Fig. 5) and then goes steadily down without a sign of overfitting. This phenomenon has been discussed widely in the recent deep learning literature [40, 55, 56], and dates back to early works in statistical mechanics of learning [4, 57].

In order to reproduce the double descent picture we repeated our experiments with two independent students by training two-layer fully connected networks with K hidden units on structured inputs with a latent task. In all experiments, we take the teacher to have one hidden unit ($M = 1$) and set its second-layer weight to unity. We train both layer of the student until convergence of the generalisation error, starting from initial weights which are drawn i.i.d. from the normal distribution with variance 10^{-3}. We chose a fixed training with a number of samples $P = 20N$. In Fig. 5), we mark the network that has the same number of parameters as the training samples with a vertical line located at $K = P/N$. In Fig. 5, we see a clear peak behaviour with all three measured errors peaking just before this line. This is consistent with the observations of [55, 56] where this behaviour was demonstrated from MNIST and other data sets and more generic neural networks. This behaviour is very consistently observed in deep learning [58].

4. Analytical study of the Hidden Manifold Model: the Gaussian Equivalence Theorem

4.1. The asymptotic limit of the hidden manifold model

As we have seen the learning phenomenology of the HMM shows interesting similarities to learning from a "real" database. The interest of the model is that, at the same time, it is amenable to analytic studies.

In the following, we shall be interested in a thermodynamic limit where the size of the input space N goes to ∞, together with the number P of patterns that are presented for learning, keeping the ratio $\alpha = P/N$ fixed. The problem can be studied analytically in this case if one assumes that the latent dimension D, i.e. the dimension of the feature space, also scales with N, meaning that it goes to ∞ with a fixed ratio $\delta = D/N$ which is of order 1 with respect to N, so that we have

$$N, P, D \to \infty, \quad \text{with fixed } \alpha = \frac{P}{N} \text{ and } \delta = \frac{D}{N}. \tag{8}$$

The difficulty in analysing HMM comes from the fact that the various components of one given input pattern, say $x_{\mu i}$ and $x_{\nu j}$, have correlations. However, it turns out that the relevant variables which are the "local fields" acting on the neurons in the hidden layer can be shown to follow a Gaussian distribution in the thermodynamic limit (8). We shall make this statement precise in Sec 4.2 in the form of the "Gaussian Equivalence Theorem" (GET). Then we shall use this theorem in order to derive the exact analytical equations for the online learning in Sec. 5.

A special case of the Gaussian Equivalence Theorem was in fact known in random matrix theory [17, 20–24] and the mapping was explicitly used in [18, 19]. We stress that the GET does not require the matrix $\boldsymbol{F}$) to be a random one, and is valid as well for deterministic matrices. This allows to generalise these mappings to the case of deterministic features using Hadamard and Fourier matrices, such as the one used in Fastfood [59] or ACDC [60] layers. These orthogonal projections are actually known to be more effective than the purely random ones [61]. It also allows generalisation of the analysis in this paper for data coming from a learned GAN, along the lines of [24, 25]. We shall illustrate this point below by analysing the dynamics of online learning when the feature matrix $\boldsymbol{F}$ is a deterministic Hadamard matrix (cf. Sec. 5.2.2).

4.2. Gaussian Equivalence Theorem

Let $\{C_r\}_{r=1}^{D}$ be D i.i.d. Gaussian random variables distributed as $\mathcal{N}(0,1)$. In the following we shall denote by $\mathbb{E}$ the expectation value with respect to this distribution. Define N variables u_i, $i = 1, \ldots, N$ as linear superpositions of the C_r variables,

$$u_i \equiv \frac{1}{\sqrt{D}} \sum_{r=1}^{D} C_r F_{ir} \ , \tag{9}$$

and M variables ν^m, $m = 1, \ldots, M$ as other linear superpositions,

$$\nu^m \equiv \frac{1}{\sqrt{D}} \sum_{r=1}^{D} C_r \tilde{w}_r^m \ , \tag{10}$$

where $\tilde{w}_r^m$ are the teacher weights Eq. (7). Define K variables λ_k as linear superpositions of $f(u_i)$ where f is an arbitrary function:

$$\lambda^k \equiv \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^k f(u_i) \ , \tag{11}$$

where $\tilde{w}_i^k$ are the student weights Eq. (1). Denoting by $\langle g(u) \rangle$ the expectation of a function $g(u)$ when u is a normal variable with distribution $u \sim \mathcal{N}(0,1)$, we also introduce for convenience the "centered" variables

$$\tilde{\lambda}^k \equiv \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^k (f(u_i) - \langle f(u) \rangle) \,. \tag{12}$$

We shall define the "thermodynamic" or "asymptotic" limit' as the limit $N \to \infty$, $D \to \infty$, keeping K, M and the ratio D/N finite. Notice that our notations keeps upper indices for indices which take values in a finite range ($k, \ell \in \{1, \dots, K\}$, $m, n \in \{1, \dots, M\}$), and lower indices for those which have a range of order N ($i, j \in \{1, \dots, N\}$; $r, s \in \{1, \dots, D\}$).

As the C_r are Gaussian, the u_i variables are also Gaussian variables, with mean zero and a matrix of covariance

$$U_{ij} = \mathbb{E}\,[u_i u_j] = \frac{1}{D} \sum_{r=1}^{D} F_{ir} F_{jr} \ . \tag{13}$$

We assume that, in the thermodynamic limit, the $\boldsymbol{W}$, $\tilde{\boldsymbol{W}}$ and $\mathbf{F}$ matrices have elements of $\mathcal{O}(1)$ and that they are "balanced" in the sense that:

$$\forall p, q\, \forall k_1, \dots, k_p, r_1, \dots r_q : \ S^{k_1 k_2 \dots k_p}_{r_1 r_2 \dots r_q} = \frac{1}{\sqrt{N}} \sum_i w_i^{k_1} w_i^{k_2} \dots w_i^{k_p} F_{ir_1} F_{ir_2} \dots F_{ir_q} = \mathcal{O}(1), \tag{14}$$

with a similar scaling for the combinations involving the teacher weights $\tilde{w}_r^m$. We also assume that

$$\frac{1}{\sqrt{D}} \sum_{r=1}^{D} F_{ir} F_{jr} = \mathcal{O}\left(\frac{1}{N}\right) \tag{15}$$

for $i \neq j$, and we normalise $\mathbf{F}$ and $\mathbf{W}$ to

$$\sum_{r=1}^{D} (F_{ir})^2 = D; \quad \sum_{i=1}^{N} (w_i^k)^2 = N. \tag{16}$$

Notice that the only variables which are drawn i.i.d. from a Gaussian distribution are the coefficients C_r. Most importantly, the matrices $\mathbf{F}$ and $\mathbf{W}$ can be arbitrary (and deterministic) as long as they are balanced.

Note that the covariances of the u_i variables scale in the thermodynamic limit as

$$\mathbb{E}\,[u_i^2] = 1; \quad \mathbb{E}\,[u_i u_j] = \mathcal{O}(1/\sqrt{D}),\ i \neq j. \tag{17}$$

Under these conditions:

Theorem 4.1. ***Gaussian Equivalence Theorem (GET)*** *In the asymptotic limit when $N \to \infty$, $D \to \infty$, keeping K, M and the ratio D/N finite, $\{\lambda^k\}$ and $\{\nu^m\}$ are $K+M$ jointly Gaussian variables, with mean*

$$\mathbb{E}\,[\lambda_k] = a \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^k; \quad \mathbb{E}\,[u^m] = 0 \,, \tag{18}$$

and covariance

$$Q^{k\ell} \ \equiv \mathbb{E}\,[\tilde{\lambda}^k \tilde{\lambda}^\ell] \ = (c - a^2 - b^2) W^{k\ell} + b^2 \Sigma^{k\ell} \,, \tag{19}$$

$$R^{km} \equiv \mathbb{E}\,[\tilde{\lambda}^k \nu^m] = b \frac{1}{D} \sum_{r=1}^{D} S_r^k \tilde{w}_r^m \,, \tag{20}$$

$$T^{mn} \equiv \mathbb{E}\,[\nu^m \nu^n] = \frac{1}{D} \sum_{r=1}^{D} \tilde{w}_r^m \tilde{w}_r^n \,, \tag{21}$$

The "folding function" $f(\cdot)$ appears through the three coefficients a, b, c, which are defined as

$$a = \langle f(u) \rangle, \quad b = \langle u f(u) \rangle, \quad c = \langle f(u)^2 \rangle \tag{22}$$

where $\langle\psi(u)\rangle$ denotes the expectation value of the function ψ when $u \sim \mathcal{N}(0,1)$ is a Gaussian variable.

The covariances are defined in terms of the three matrices

$$S_r^k \equiv \frac{1}{\sqrt{N}} \sum_{i=1}^N w_i^k F_{ir} \,, \tag{23}$$

$$W^{k\ell} \equiv \frac{1}{N} \sum_{i=1}^N w_i^k w_i^\ell \,, \tag{24}$$

$$\Sigma^{k\ell} \equiv \frac{1}{D} \sum_{r=1}^D S_r^k S_r^\ell \,, \tag{25}$$

whose elements are assumed to be of order $\mathcal{O}(1)$ in the asymptotic limit.

The proof of the theorem is given in Appendix A. This Gaussian theorem shows that there is a whole family of activation functions $f(x)$ (those that have the same values for a, b and c) that will lead to equivalent analytical results for the learning curves studied in this paper. Furthermore, it forms a basis from which we can develop an analytical understanding of learning with the hidden manifold model, as we show in the next section.

5. The dynamics of stochastic gradient descent for the HMM in the teacher-student setup

We now analyse the dynamics of stochastic gradient descent in the case of *online learning*, where at each step of the algorithm $\mu = 1, 2, \ldots$, the student's weights are updated according to Eq. (2) using a previously unseen sample $(\boldsymbol{x}_\mu, y_\mu)$. This case is also known as one-shot or single-pass SGD, and it has the advantage that it can be exactly described analytically, as we will show in this section.

Before diving into the details of the analysis, we checked numerically that online learning in the thermodynamic limit of Eq. (8) preserves the effects we observed in our experiments of Sec. 2. On the left of Fig. 6, we show the behaviour of independent students for the same experiment as in Fig. 1, but here the networks are trained on a data set drawn from the hidden manifold model with $\delta = 0.05$ that is large enough that every sample is used only once during training. We checked numerically that the results shown in this plot remain unchanged, apart from finite-size effects and sample-to-sample variations, as we increase N and D while keeping $\delta = 0.05$, going up to $N = 1000$. We were able to reproduce the results from MNIST and Fashion MNIST (Figs. 1, 10) using the online learning for the hidden manifold model. On the right of Fig. 6, we plot the generalisation dynamics of a two-layer network trained on the HMM with $\delta = 0.5$ during online learning. When training only the first layer of weights (dashed line), or when training both layers (solid line), we see that there are no distinguishable plateaus in the dynamics, in stark contrast to the vanilla teacher-student setup shown in Fig. 2. Online learning of the hidden manifold model is thus a sensible case to study the effects we observed on MNIST at the outset of the paper.

The goal of our analysis is to track the mean-squared generalisation error of the student with respect to the teacher at all times,

$$\epsilon_g^{\text{mse}}(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) \equiv \frac{1}{2}\mathbb{E}\,\left([\phi(\boldsymbol{x}, \boldsymbol{\theta}) - \tilde{y}^*]^2\right)\,, \tag{26}$$

where the expectation $\mathbb{E}$ denotes an average over an input drawn from the hidden manifold model, Eq. (6), with latent label $\widetilde{y}_\mu^* = \phi(\mathbf{c}_\mu, \widetilde{\boldsymbol{\theta}}^*)$ given by a teacher network with fixed weights $\widetilde{\boldsymbol{\theta}}^*$ acting on the latent representation (7). Note that the weights of both the student and the teacher,

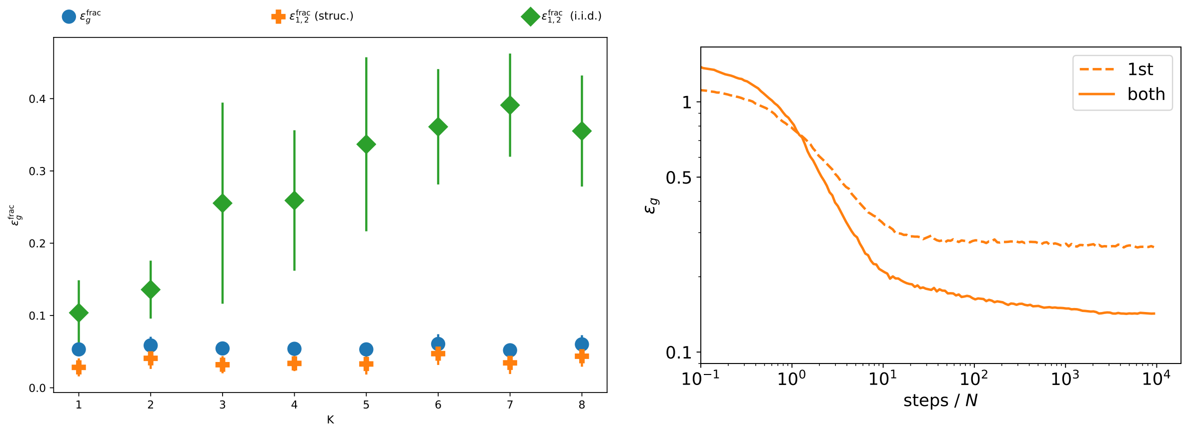


Figure 6: The online learning in the hidden manifold model (8) **reproduces the behaviour of independent students seen on MNIST.** *(Left)* Same plot as Fig. 1, but this time we train the two independent students on a data set drawn from the hidden manifold (8) with $\delta = 0.05$, training only the first layer of the student network. We also checked numerically that this behaviour is unchanged as we increase N and D while keeping δ constant, going up to $N = 1000$. Here, $f(x) = \mathrm{sgn}(x), g(x) = \tilde{g}(x) = \mathrm{erf}(x/\sqrt{2}), N = 300, D = 15, M = 1, \eta = 0.2, \tilde{v}_m = 1, v_k = 1$. *(Right)* Generalisation dynamics of online learning using a data set drawn from the HMM with $\delta = 0.5$. The plateau seen in the vanilla-teacher-student setup (Fig. 2) are greatly reduced; see Sec. 5.2.1 for a detailed discussion. $f(x) = \mathrm{sgn}(x), g(x) = \tilde{g}(x) = \mathrm{erf}(x/\sqrt{2}), N = 1000, D = 50, M = 4, K = 3, \eta = 0.2$.

as well as the feature matrix F_{ir}, are held fixed when taking the average, which is an average only over the coefficients $c_{\mu r}$.

The analysis of online learning has been performed previously for the vanilla teacher-student model (4) with i.i.d. Gaussian inputs [12, 47, 62–64], and has recently been put on a rigorous foundation [13].

Here, we generalise this type of analysis to two-layer neural networks trained on the Hidden Manifold Model. Since we saw in the numerical part of our paper that the phenomenology described in Sec. 3.2 does not depend on whether we train both layers or only the first layer of the student, here we study training the first layer only, while the second layer of both the teacher and the student is fixed at $v_k = \tilde{v}^*_m = 1$ for all $k = 1, \ldots, K;\ m = 1, \ldots, M$, similar to previous work [12, 63, 64]. Generalising our equations to the case where both layers are trained is a straightforward exercise that we leave for future work. To keep notation compact, we focus on cases where $a = \mathbb{E}\, f(u) = 0$ in (22), which leads to $\tilde{\lambda}^k = \lambda^k$ in (12). A generalisation to the case where $a \neq 0$ is straightforward.

We can make progress with the high-dimensional average over $\boldsymbol{x}$ in Eq. (26) by noticing that the input $\boldsymbol{x}$ and its latent representation $\boldsymbol{c}$ only enter the expression via the local fields ν^m and λ^k, Eqs. (10, 11):

$$\epsilon_g^{\mathrm{mse}}(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = \frac{1}{2}\sum_{k,\ell}^{K} \mathbb{E}\, g(\lambda^k) g(\lambda^\ell) + \frac{1}{2}\sum_{n,m}^{M} \mathbb{E}\, \tilde{g}(\nu^n)\tilde{g}(\nu^m) - \sum_{k}^{K}\sum_{n}^{M} \mathbb{E}\, g(\lambda^k)\tilde{g}(\nu^m) \tag{27}$$

and the average is now taken over the joint distribution of local fields $\{\lambda^{k=1,\ldots,K}, \nu^{m=1,\ldots,M}\}$. The key step is now to invoke the Gaussian Equivalence Theorem 4.1, which guarantees that this distribution is a multivariate normal distribution with covariances $Q^{k\ell}, R^{km}$, and T^{nm} (19–21). We stress at this point that in the online learning each new presented sample is independent of what has been learned so far and this is why the GET can be used at every step. This is in contrast with the full-batch learning where the corresponding results were only conjectured to be hold in [19]. Depending on the choice of $g(x)$ and $\tilde{g}(x)$, this makes it possible to compute the average analytically; in any case, the GET guarantees that we can express $\epsilon_g(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}})$ as a function of only $Q^{k\ell}, R^{km}$, and T^{nm}, which are called *order parameters* in statistical physics [12, 47, 62]:

$$\lim_{N,D\to\infty} \epsilon_g^{\mathrm{mse}}(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = \epsilon_g^{\mathrm{mse}}(Q^{k\ell}, R^{kn}, T^{nm}) \tag{28}$$

where in taking the limit, we keep the ratio $\delta = {}^{D}\!/\!{}_{N}$ finite (see Eq. (8)). For example, for a student with $g(\lambda^k) = \mathrm{erf}(\lambda^k/\sqrt{2})$ and a teacher with $\tilde{g}(\nu^m) = \max(0, \nu^m)$, we find that

$$\begin{aligned}\epsilon_g^{\mathrm{mse}}(Q^{k\ell}, R^{kn}, T^{nm}) = \frac{1}{\pi}\sum_{i,k} \arcsin\left(\frac{Q^{ik}}{\sqrt{1+Q^{ii}}\sqrt{1+Q^{kk}}}\right) - \sum_{k,n} \frac{R^{kn}}{\sqrt{2\pi}\sqrt{1+Q^{kk}}} \\ + \sum_{n,m} \frac{2\sqrt{T^{mm}T^{nn} - (T^{nm})^2} + T^{nm}\left[\pi + 2\arctan\left(\frac{T^{nm}}{\sqrt{T^{mm}T^{nn}-(T^{nm})^2}}\right)\right]}{8\pi}.\end{aligned} \tag{29}$$

Intuitively, the order parameter R^{kn} measures the similarity between the action of the ith student node on an input $\boldsymbol{x}_\mu$ and the nth teacher node acting on the corresponding latent representation c_μ. The matrix $Q^{k\ell} = \left[c - b^2\right] W^{k\ell} + b^2 \Sigma^{k\ell}$ quantifies the similarity between two student nodes k and ℓ, and has two parts: the latent student-student overlap $\Sigma^{k\ell}$, which measures the overlap of the weights of two students nodes after they have been projected to the hidden manifold, and the ambient student-student overlap $W^{k\ell}$, which measures the overlap between the vectors $\boldsymbol{w}^k, \boldsymbol{w}^\ell \in \mathbb{R}^N$. The overlaps of the teacher nodes are collected in the matrix T^{nm}, which is *not* time-dependent, as it is a function of the teacher weights only. Our aim is then to obtain a closed set of differential equations that describe the dynamics of the order parameters, which we will call their equations of motion.

5.1. Derivation of the equations of motion

When we make a step of SGD, we update the weight w_i^k using a new sample, generated using a previously unused sample according to

$$\left(w_i^k\right)_{\mu+1} - \left(w_i^k\right)_\mu = -\frac{\eta}{\sqrt{N}} \Delta g'(\lambda^k) f(u_i)\,, \tag{30}$$

where $\Delta = \sum_{j=1}^K g(\lambda^j) - \sum_{m=1}^M \tilde{g}(\nu^m)$. From here on out, we shall drop the index μ on the right-hand side as we work at a fixed iteration time. We will keep the convention of Sec. 4.2 where extensive indices (taking values up to N or D) are below the line, while we'll use upper indices when they take finite values up to M or K. The challenge of controlling the learning in the thermodynamic limit will be to write closed equations using matrices with only "upper" indices left. Furthermore, we will adopt the convention that the indices $j, k, \ell, \iota = 1, \ldots, K$ always denote *student* nodes, while $n, m = 1, \ldots, M$ are reserved for teacher hidden nodes.

5.1.1. First steps

When we study the evolution of quantities that are linear in the weights, like S_r^k and the order parameters constructed from it, *e.g.* $\Sigma^{k\ell}$, we need to study

$$\left[\sum_{j=1}^K g(\lambda^j) - \sum_{m=1}^M \tilde{g}(\nu^m)\right] g'(\lambda^k) f(u_i) = \sum_{j\neq k}^K \mathpzc{a}_i^{jk} + \mathpzc{b}_i^k - \sum_{n=1}^M \mathpzc{c}_i^{nk}\,, \tag{31}$$

where

$$\mathpzc{a}_i^{jk} = g(\lambda^j) g'(\lambda^k) f(u_i)\,, \tag{32}$$

$$\mathpzc{b}_i^k = g(\lambda^k) g'(\lambda^k) f(u_i)\,, \tag{33}$$

$$\mathpzc{c}_i^{nk} = \tilde{g}(\nu^n) g'(\lambda^k) f(u_i)\,. \tag{34}$$

We can thus follow the dynamics of S_r^k (23), which is linear in the weights and enters the definition of the order parameters R^{km} (20) and Σ^{kl} (25):

$$\left(S_r^k\right)_{\mu+1} - \left(S_r^k\right)_{\mu} = -\frac{\eta}{N}\sum_i F_{ir}\left[\sum_{j\neq k}^{K} a_i^{jk} + b_i^k - \sum_n^M c_i^{nk}\right]. \tag{35}$$

We want to average this update equation over a new incoming sample, i.e. over the $\boldsymbol{c}_r$ variables. Upon contraction with F_{ir} in Eq. (35), we are thus led to computing the averages

$$\mathcal{A}_r^{jk} \equiv \frac{1}{\sqrt{N}}\sum_i \mathbb{E}\left[F_{ir} a_i^{jk}\right] = \mathbb{E}\left[g(\lambda^j)g'(\lambda^k)\beta_r\right], \tag{36}$$

$$\mathcal{B}_r^k \equiv \mathbb{E}\left[g(\lambda^k)g'(\lambda^k)\beta_r\right], \tag{37}$$

and

$$\mathcal{C}_r^{nk} = \mathbb{E}\left[\tilde{g}(\nu^n)g'(\lambda^k)\beta_r\right] , \tag{38}$$

where

$$\beta_r = \frac{1}{\sqrt{N}}\sum_i F_{ir} f(u_i). \tag{39}$$

The crucial fact that allows for an analytic study of online learning is that, at each step μ of SGD, a previously unseen input $\boldsymbol{x}_\mu$ is used to evaluate the gradient. The latent representation $\boldsymbol{c}_\mu$ of this input is given by a new set of i.i.d. Gaussian random variables $c_{\mu r}$, which are thus independent of the current weights of the student at that time. In the thermodynamic limit, the GET of the previous section shows that, for one given value of r, the $K+M+1$ variables $\{\lambda^k\}$, $\{\nu^m\}$ and β_r have a joint Gaussian distribution, making it possible to express the averages over $\{\lambda^k, \nu^m, \beta_r\}$ in terms of only their covariances.

Looking closer, we see that the average of (36,37,38) over this Gaussian distribution involves two sets of random variables: on the one hand, the $M+K$ local fields $\{\nu^m, \lambda^k\}$, which have correlations of order 1, and on the other hand the variable β_r (for one given value of r). It turns out that β_r is only weakly correlated with the local fields $\{\nu^m, \lambda^k\}$ (the correlation is $\mathcal{O}(1/\sqrt{N})$). In Appendix A.1, we discuss how to compute this type of average and prove Lemma A.1, which for the averages (36–38) yields

$$\begin{aligned}\mathcal{A}_r^{jk} = \frac{1}{Q^{kk}Q^{jj} - (Q^{kj})^2}\Big[&Q^{jj}\mathbb{E}\left[g'(\lambda^k)\lambda^k g(\lambda^j)\right]\,\mathbb{E}\left[\lambda^k\beta_r\right] - Q^{kj}\mathbb{E}\left[g'(\lambda^k)\lambda^j g(\lambda^j)\right]\,\mathbb{E}\left[\lambda^k\beta_r\right]\\ &-Q^{kj}\mathbb{E}\left[g'(\lambda^k)\lambda^k g(\lambda^j)\right]\,\mathbb{E}\left[\lambda^j\beta_r\right] + Q^{kk}\mathbb{E}\left[g'(\lambda^k)\lambda^j g(\lambda^j)\right]\,\mathbb{E}\left[\lambda^j\beta_r\right]\Big],\end{aligned} \tag{40}$$

$$\mathcal{B}_r^k = \frac{1}{Q^{kk}}\mathbb{E}\left[g'(\lambda^k)\lambda^k g(\lambda^k)\right]\,\mathbb{E}\left[\lambda^k\beta_r\right], \tag{41}$$

$$\begin{aligned}\mathcal{C}_r^{nk} = \frac{1}{Q^{kk}T^{nn} - (R^{kn})^2}\Big[&T^{nn}\mathbb{E}\left[g'(\lambda^k)\lambda^k\tilde{g}(\nu^n)\right]\,\mathbb{E}\left[\lambda^k\beta_r\right] - R^{kn}\mathbb{E}\left[g'(\lambda^k)\nu^n\tilde{g}(\nu^n)\right]\,\mathbb{E}\left[\lambda^k\beta_r\right]\\ &-R^{kn}\mathbb{E}\left[g'(\lambda^k)\lambda^k\tilde{g}(\nu^n)\right]\,\mathbb{E}\left[\nu^n\beta_r\right] + Q^{kk}\mathbb{E}\left[g'(\lambda^k)\nu^n\tilde{g}(\nu^n)\right]\,\mathbb{E}\left[\nu^n\beta_r\right]\Big].\end{aligned} \tag{42}$$

This yields

$$\left(S_r^k\right)_{\mu+1} - \left(S_r^k\right)_{\mu} = -\frac{\eta}{\sqrt{N}}\left[\sum_{j\neq k}^{K}\mathcal{A}_r^{jk} + \mathcal{B}_r^k - \sum_n^M \mathcal{C}_r^{nk}\right], \tag{43}$$

with only the single intensive index r left. While this equation would appear to open up a way to write down the equation of motion for the "teacher-student" overlap R^{km} by contracting (43) with $\tilde{w}_r^m$, we show in Appendix B that such a program will lead to an infinite hierarchy of equations. To avoid this problem, we rotate the problem to a different basis, as we explain in the next section.

5.1.2. Changing the basis to close the equations

We can close the equations for the order parameters by studying their dynamics in the basis given by the eigenvectors of the operator

$$\Omega_{rs} \equiv \frac{1}{N} \sum_i F_{ir} F_{is}, \tag{44}$$

which is a $D \times D$ symmetric matrix, with diagonal elements $\Omega_{rr} = 1$, and off-diagonal elements of order $1/\sqrt{N}$. Consider the orthogonal basis of eigenvectors $\psi_{\tau=1,\dots,D}$ of this matrix, with corresponding eigenvalues ρ_τ, such that

$$\sum_s \Omega_{rs} \psi_{\tau s} = \rho_\tau \psi_{\tau r}. \tag{45}$$

We will suppose that the components of the eigenvectors $\psi_{\tau r}$ are of order 1 and we impose the following normalisation:

$$\sum_s \psi_{\tau s} \psi_{\tau' s} = D \delta_{\tau\tau'}, \qquad \sum_\tau \psi_{\tau r} \psi_{\tau s} = D \delta_{rs}. \tag{46}$$

In this basis, the teacher-student overlap R^{km} (20) is given by

$$R^{km} = \frac{b}{D} \sum_\tau \Gamma_\tau^k \tilde{\omega}_\tau^m, \tag{47}$$

where we have introduced the projections

$$\Gamma_\tau^k = \frac{1}{\sqrt{D}} \sum_r S_r^k \psi_{\tau r} \tag{48}$$

and

$$\tilde{\omega}_\tau^m = \frac{1}{\sqrt{D}} \sum_r \tilde{w}_r^m \psi_{\tau r}. \tag{49}$$

Since $\tilde{\omega}_\tau^m$ is a static variable, the time evolution of Γ_τ^k is given by

$$\left(\Gamma_\tau^k\right)_{\mu+1} - \left(\Gamma_\tau^k\right)_\mu = -\frac{\eta}{\sqrt{\delta} N} \sum_r \psi_{\tau r} \left[\sum_{j \neq k}^K \mathcal{A}_r^{jk} + \mathcal{B}_r^k - \sum_n^M \mathcal{C}_r^{nk} \right] \tag{50}$$

As we aim to compute the remaining sum over r, two types of terms appear:

$$\sum_r \psi_{\tau r} \mathbb{E}\left[\lambda^k \beta_r\right] = \frac{1}{\sqrt{\delta}} \left((c - b^2)\delta + b^2 \rho_\tau\right) \Gamma_\tau^k = \frac{d_\tau}{\sqrt{\delta}} \Gamma_\tau^k, \tag{51}$$

where we have defined $d_\tau = (c - b^2)\delta + b^2 \rho_\tau$, and

$$\sum_r \psi_{\tau r} \mathbb{E}\left[\nu^n \beta_r\right] = \frac{b}{\sqrt{\delta}} \rho_\tau \tilde{\omega}_\tau^n. \tag{52}$$

Putting everything together, the final evolution of Γ_τ^k is

$$\begin{aligned}\left(\Gamma_\tau^k\right)_{\mu+1} - \left(\Gamma_\tau^k\right)_\mu = -\frac{\eta}{\delta N}\Bigg[& d_\tau \Gamma_\tau^k \sum_{j\neq k} \frac{Q^{jj}\mathbb{E}\left[g'(\lambda^k)\lambda^k g(\lambda^j)\right] - Q^{kj}\mathbb{E}\left[g'(\lambda^k)\lambda^j g(\lambda^j)\right]}{Q^{kk}Q^{jj} - (Q^{kj})^2} \\ & + \sum_{j\neq k} d_\tau \Gamma_\tau^j \frac{Q^{kk}\mathbb{E}\left[g'(\lambda^k)\lambda^j g(\lambda^j)\right] - Q^{kj}\mathbb{E}\left[g'(\lambda^k)\lambda^k g(\lambda^j)\right]}{Q^{kk}Q^{jj} - (Q^{kj})^2} \\ & + d_\tau \Gamma_\tau^k \frac{1}{Q^{kk}} \mathbb{E}\left[g'(\lambda^k)\lambda^k g(\lambda^k)\right] \\ & - d_\tau \Gamma_\tau^k \sum_n \frac{T^{nn}\mathbb{E}\left[g'(\lambda^k)\lambda^k \tilde{g}(\nu^n)\right] - R^{kn}\mathbb{E}\left[g'(\lambda^k)\nu^n \tilde{g}(\nu^n)\right]}{Q^{kk}T^{nn} - (R^{kn})^2} \\ & - b\rho_\tau \sum_n \tilde{\omega}_\tau^n \frac{Q^{kk}\mathbb{E}\left[g'(\lambda^k)\nu^n \tilde{g}(\nu^n)\right] - R^{kn}\mathbb{E}\left[g'(\lambda^k)\lambda^k \tilde{g}(\nu^n)\right]}{Q^{kk}T^{nn} - (R^{kn})^2}\Bigg].\end{aligned} \tag{53}$$

At this point, we note that the remaining averages appearing in this expression, such as $\mathbb{E}\left[\lambda^k g'(\lambda^k)\tilde{g}(\nu^m)\right]$, depend only on subsets of the local fields $\{\lambda^{k=1,\dots,K}, \nu^{m=1,\dots,M}\}$. As discussed above, the GET guarantees that these random variables follow a multi-dimensional Gaussian distribution, so these averages depend only on the covariances of the local fields R^{km}, $Q^{k\ell}$, and T^{mn}. To simplify the subsequent equations, we will use the following shorthand for the three-dimensional Gaussian averages

$$I_3(k,j,n) \equiv \mathbb{E}\left[g'(\lambda^k)\lambda^j \tilde{g}(\nu^n)\right], \tag{54}$$

which was introduced by Saad & Solla [12]. Arguments passed to I_3 should be translated into local fields on the right-hand side by using the convention where the indices j, k, ℓ, ι always refer to student local fields λ^j, etc., while the indices n, m always refer to teacher local fields ν^n, ν^m. Similarly,

$$I_3(k,j,j) \equiv \mathbb{E}\left[g'(\lambda^k)\lambda^j g(\lambda^j)\right], \tag{55}$$

where having the index j as the third argument means that the third factor is $g(\lambda^j)$, rather than $\tilde{g}(\nu^m)$ in Eq. (54). The average in Eq. (54) is taken over a three-dimensional normal distribution with mean zero and covariance matrix

$$\Phi^{(3)}(k,j,n) = \begin{pmatrix} \mathbb{E}\left[\lambda^k\lambda^k\right] & \mathbb{E}\left[\lambda^k\lambda^j\right] & \mathbb{E}\left[\lambda^k\nu^n\right] \\ \mathbb{E}\left[\lambda^k\lambda^j\right] & \mathbb{E}\left[\lambda^j\lambda^j\right] & \mathbb{E}\left[\lambda^j\nu^n\right] \\ \mathbb{E}\left[\lambda^k\nu^n\right] & \mathbb{E}\left[\lambda^j\nu^n\right] & \mathbb{E}\left[\nu^n\nu^n\right] \end{pmatrix} = \begin{pmatrix} Q^{kk} & Q^{kj} & R^{kn} \\ Q^{kj} & Q^{jj} & R^{jn} \\ R^{kn} & R^{jn} & T^{nn} \end{pmatrix}. \tag{56}$$

The explicit forms of the integrals I_3 and I_4 that appear in the equations of motion for the order parameters and the generalisation error for networks with $g(x) = \operatorname{erf}\left(x/\sqrt{2}\right)$. They were first given by [12, 47]. Denoting the elements of the covariance matrix such as Φ^3 (56) as ϕ_{ij}, we have

$$I_3(\cdot,\cdot,\cdot) = \frac{2}{\pi}\frac{1}{\sqrt{\Lambda_3}}\frac{\phi_{23}(1+\phi_{11}) - \phi_{12}\phi_{13}}{1+\phi_{11}} \tag{57}$$

with

$$\Lambda_3 = (1+\phi_{11})(1+\phi_{33}) - \phi_{13}^2. \tag{58}$$

For the average I_4, we have a covariance matrix $\Phi^{(4)}$ that is populated in analogy to $\Phi^{(3)}$ (56), we have

$$I_4(\cdot,\cdot,\cdot,\cdot) = \frac{4}{\pi^2}\frac{1}{\sqrt{\Lambda_4}}\arcsin\left(\frac{\Lambda_0}{\sqrt{\Lambda_1 \Lambda 2}}\right) \tag{59}$$

where

$$\Lambda_4 = (1+\phi_{11})(1+\phi_{22}) - \phi_{12}^2, \tag{60}$$

$$\Lambda_0 = \Lambda_4\phi_{34} - \phi_{23}\phi_{24}(1+\phi_{11}) - \phi_{13}\phi_{14}(1+\phi_{22}) + \phi_{12}\phi_{13}\phi_{24} + \phi_{12}\phi_{14}\phi_{23}, \tag{61}$$

$$\Lambda_1 = \Lambda_4(1+\phi_{33}) - \phi_{23}^2(1+\phi_{11}) - \phi_{13}^2(1+\phi_{22}) + 2\phi_{12}\phi_{13}\phi_{23}, \tag{62}$$

$$\Lambda_2 = \Lambda_4(1+\phi_{44}) - \phi_{24}^2(1+\phi_{11}) - \phi_{14}^2(1+\phi_{22}) + 2\phi_{12}\phi_{14}\phi_{24} \tag{63}$$

5.1.3. Dynamics of the teacher-student overlap R^{km}

We are now in a position to write the update equation for

$$\left(R^{km}\right)_{\mu+1} - \left(R^{km}\right)_{\mu} = \frac{b}{D}\sum_{\tau}\left[\left(\Gamma_{\tau}^{k}\right)_{\mu+1} - \left(\Gamma_{\tau}^{k}\right)_{\mu}\right]\tilde{\omega}_{\tau}^{m}, \tag{64}$$

where we have used that the $\tilde{\omega}_{\tau}^{m}$ are static. When performing the last remaining sum over τ, two types of terms appear. First, there is

$$\tilde{T}^{mn} \equiv \frac{1}{D}\sum_{\tau}\rho_{\tau}\tilde{\omega}_{\tau}^{m}\tilde{\omega}_{\tau}^{n}. \tag{65}$$

which depends only on the choice of the feature matrix F_{ir} and the teacher weights w_{mr}^{*} and is thus a constant of the motion. The second type of term is of the form

$$\frac{1}{D}\sum_{\tau}\rho_{\tau}\Gamma_{\tau}^{\ell}\tilde{\omega}_{\tau}^{n}. \tag{66}$$

This sum cannot be reduced to a simple expression in terms of other order parameters. Instead, we are led to introduce the density

$$r^{km}(\rho) = \frac{1}{\varepsilon_{\rho}}\frac{1}{D}\sum_{\tau}\Gamma_{\tau}^{k}\tilde{\omega}_{\tau}^{m}\ \mathbb{1}\left(\rho_{\tau}\in[\rho,\rho+\varepsilon_{\rho}[\right), \tag{67}$$

where $\mathbb{1}(\cdot)$ is the indicator function which evaluates to 1 if the condition given to it as an argument is true, and which otherwise evaluates to 0. We take the limit $\varepsilon_{\rho}\to 0$ after the thermodynamic limit. Then we can rewrite the order parameter R^{km} as an integral over the density r^{km}, weighted by the distribution of eigenvalues of the operator Ω_{rs}, $P_{\Omega}(\rho)$:

$$R^{km} = b\int \mathrm{d}\rho\ P_{\Omega}(\rho)\ r^{km}(\rho)\,. \tag{68}$$

If, for example, we take the elements of the feature matrix F_{ir} to be element-wise i.i.d. from the normal distribution with mean zero and unit variance, then the limiting density of eigenvalues of Ω is given by the Marchenko-Pastur law [65]:

$$P_{\mathrm{MP}}(\rho) = \frac{1}{2\pi\delta}\frac{\sqrt{(\rho_{\max}-\rho)(\rho-\rho_{\min})}}{\rho}, \tag{69}$$

where $\rho_{\min} = \left(1-\sqrt{\delta}\right)^2$ and $\rho_{\max} = \left(1+\sqrt{\delta}\right)^2$.

The update equation of $r^{km}(\rho)$ can be obtained immediately by substituting the update equation for Γ_{τ}^{k} (53) into its definition (67). Finally, in the thermodynamic limit, the normalised number of steps $t = \mu/N$ can be interpreted as a continuous time-like variable, and so we have

$$R^{km}(t) = b\int \mathrm{d}\rho\ P_{\Omega}(\rho)\ r^{km}(\rho, t) \tag{70}$$

with

$$\begin{aligned}\frac{\partial r^{km}(\rho,t)}{\partial t} = -\frac{\eta}{\delta}\Bigg(& d(\rho) r^{km}(\rho) \sum_{j\neq k} \frac{Q^{jj}\, I_3(k,k,j) - Q^{kj} I_3(k,j,j)}{Q^{jj}Q^{kk} - (Q^{kj})^2} \\ & + d(\rho) \sum_{j\neq k} r^{jm}(\rho) \frac{Q^{kk} I_3(k,j,j) - Q^{kj}\, I_3(k,k,j)}{Q^{jj}Q^{kk} - (Q^{kj})^2} \\ & + d(\rho) r^{km}(\rho) \frac{1}{Q^{kk}} I_3(k,k,k) \\ & - d(\rho) r^{km}(\rho) \sum_n \frac{T^{nn} I_3(k,k,n) - R^{kn} I_3(k,n,n)}{Q^{kk}T^{nn} - (R^{kn})^2} \\ & - b\rho \sum_n \tilde{T}^{nm} \frac{Q^{kk} I_3(k,n,n) - R^{kn} I_3(k,k,n)}{Q^{kk}T^{nn} - (R^{kn})^2} \Bigg),\end{aligned} \tag{71}$$

where $d(\rho) = (c - b^2)\delta + b^2\rho$. Note that while we have dropped the explicit time dependence from the right-hand side to keep the equation readable, all the order parameters on the right-hand side are explicitly time-dependent, i.e. $Q^{jj} = Q^{jj}(t)$, $r^{km}(\rho) = r^{km}(\rho, t)$, and the averages $I_3(\cdot)$ are also time-dependent via their dependence on the order parameters (see Eq. (54) and the subsequent discussion). In order to close the equations of motion, we now need to find the equations for the order parameters that are quadratic in the weights.

5.1.4. Order parameters that are quadratic in the weights

There are two order parameters that appear when evaluating the covariance of the λ variables:

$$Q^{k\ell} \equiv \mathbb{E}\left[\lambda^k \lambda^\ell\right] = \left[c - b^2\right] W^{k\ell} + b^2 \Sigma^{k\ell}. \tag{72}$$

We will look at both $W^{k\ell}$ and $\Sigma^{k\ell}$ in turn now.

Equation of motion for $W^{k\ell}$ For the student-student overlap matrix

$$W^{k\ell} = \frac{1}{N} \sum_i^N w_i^k w_i^\ell, \tag{73}$$

we find, after some algebra, that updates read

$$\begin{aligned}\left(W^{k\ell}\right)^{\mu+1} - \left(W^{k\ell}\right)_\mu = & -\frac{\eta}{N^{3/2}} \sum_i^N w_i^\ell \left[\sum_{j\neq k}^K \mathscr{a}_i^{jk} + \mathscr{b}_i^k - \sum_n^M \mathscr{c}_i^{nk}\right] \\ & - \frac{\eta}{N^{3/2}} \sum_i^N w_i^k \left[\sum_{j\neq \ell}^K \mathscr{a}_i^{j\ell} + \mathscr{b}_i^\ell - \sum_n^M \mathscr{c}_i^{n\ell}\right] \\ & + \frac{\eta^2}{N^2} \sum_i^N f(u_i)^2 g'(\lambda^k) g'(\lambda^\ell) \Bigg[\sum_{j,\iota}^K g(\lambda^j) g(\lambda^\iota) + \sum_{n,m}^M \tilde{g}(\nu^n)\tilde{g}(\nu^m) \\ & \qquad -2 \sum_j^K \sum_m^M g(\lambda^j)\tilde{g}(\nu^m)\Bigg]\end{aligned} \tag{74}$$

For the terms linear in the learning rate η, we can immediately carry out the sum over i, which yields terms of the type

$$\frac{1}{\sqrt{N}} \sum_i w_i^\ell \mathbb{E}\left[g(\lambda^j) g'(\lambda^k) f(u_i)\right] = \mathbb{E}\left[g'(\lambda^k) \lambda^\ell g(\lambda^j)\right] = I_3(k,\ell,j) \quad \text{etc.} \tag{75}$$

The term quadratic in the learning rate η requires the evaluation of terms of the type

$$\frac{1}{N}\sum_i \mathbb{E}\left[f(u_i)^2 g'(\lambda^k)g'(\lambda^\ell)g(\lambda^j)g(\lambda^\iota)\right] = c\mathbb{E}\left[g'(\lambda^k)g'(\lambda^\ell)g(\lambda^j)g(\lambda^\iota)\right]. \tag{76}$$

The sum over i thus makes this second-order term contribute to the total variation of $W^{k\ell}$ at leading order, and we're left with an average over four local fields, for which we introduce the short-hand

$$I_4(k,\ell,j,\iota) \equiv \mathbb{E}\left[g'(\lambda^k)g'(\lambda^\ell)g(\lambda^j)g(\lambda^\iota)\right], \tag{77}$$

where we use the same notation as we did for $I_3(\cdot)$ (54). The full equation of motion for $W^{k\ell}$ thus reads

$$\begin{aligned}\frac{\mathrm{d}W^{k\ell}(t)}{\mathrm{d}t} =& -\eta\left(\sum_j^K I_3(k,\ell,j) - \sum_n I_3(k,\ell,n)\right) - \eta\left(\sum_j^K I_3(\ell,k,j) - \sum_n I_3(\ell,k,n)\right)\\ &+ c\eta^2\left(\sum_{j,a}^K I_4(k,\ell,j,a) - 2\sum_j^K\sum_m^M I_4(k,\ell,j,m) + \sum_{n,m}^M I_4(k,\ell,n,m)\right).\end{aligned} \tag{78}$$

Equation of motion for $\Sigma^{k\ell}$ After rotating to the basis ψ_τ, we have

$$\Sigma^{k\ell} \equiv \frac{1}{D}\sum_r S_r^k S_r^\ell = \frac{1}{D}\sum_\tau \Gamma_\tau^k \Gamma_\tau^\ell. \tag{79}$$

It is then immediate that

$$\begin{aligned}(\Sigma^{k\ell})^{\mu+1} - (\Sigma^{k\ell})_\mu =& \frac{1}{D}\sum_\tau \left(\Gamma_\tau^\ell\right)_\mu\left[(\Gamma_\tau^k)^{\mu+1} - (\Gamma_\tau^k)_\mu\right] + \frac{1}{D}\sum_\tau\left(\Gamma_\tau^k\right)_\mu\left[(\Gamma_\tau^\ell)^{\mu+1} - (\Gamma_\tau^\ell)_\mu\right]\\ &+ \frac{\eta^2}{D^2N}\sum_\tau\sum_{r,s}^R \psi_{\tau r}\psi_{\tau s}\mathbb{E}\left[\Delta^2 g'(\lambda^k)g'(\lambda^\ell)\beta_r\beta_s\right].\end{aligned} \tag{80}$$

The terms linear in η can be obtained directly by substituting in the update equation for Γ_τ^k (53) and are similar to the update equation for $r^{km}(\rho)$. As for the term quadratic in η, we have to leading order

$$\begin{aligned}\frac{\eta^2}{DN}\sum_{r,s}^R \psi_{\tau r}\psi_{\tau s}\mathbb{E}\left[\Delta^2 g'(\lambda^k)g'(\lambda^\ell)\beta_r\beta_s\right] &= \frac{\eta^2}{DN}\sum_r^R (\psi_{\tau r})^2\mathbb{E}\left[\Delta^2 g'(\lambda^k)g'(\lambda^\ell)\right]\mathbb{E}\left[\beta_r^2\right]\\ &= \frac{\eta^2}{N}\mathbb{E}\left[\Delta^2 g'(\lambda^k)g'(\lambda^\ell)\right]\left[(c-b^2)\rho_\tau + \frac{b^2}{\delta}\rho_\tau^2\right],\end{aligned} \tag{81}$$

where we have used that covariance of β_r is given by

$$\mathbb{E}\left[\beta_r^2\right] = c - b^2 + \frac{b^2}{\delta}\sum_t \Omega_{rt}^2. \tag{82}$$

To deal with the remaining sum over τ, we again make use of the fact that the equation of motion for $\Sigma^{k\ell}$ depends on the eigenvector index τ only through the eigenvalue ρ_τ. Introducing the density

$$\sigma^{k\ell}(\rho) = \frac{1}{\varepsilon_\rho}\frac{1}{D}\sum_\tau \Gamma_\tau^k\Gamma_\tau^\ell \mathbb{1}\left(\rho_\tau \in [\rho, \rho+\varepsilon_\rho[\right), \tag{83}$$

as we did for $r^{km}(\rho)$ (67), we have

$$\Sigma^{k\ell}(t) = \int \mathrm{d}\rho\, P_\Omega(\rho)\,\sigma^{k\ell}(\rho,t) \tag{84}$$

with

$$
\begin{aligned}
\frac{\partial \sigma^{k\ell}(\rho,t)}{\partial t} = -\frac{\eta}{\delta} \Bigg(& d(\rho)\sigma^{k\ell}(\rho) \sum_{j\neq k} \frac{Q^{jj} I_3(k,k,j) - Q^{kj} I_3(k,j,j)}{Q^{jj}Q^{kk} - (Q^{kj})^2} \\
& + \sum_{j\neq k} d(\rho)\sigma^{j\ell}(\rho) \frac{Q^{kk} I_3(k,j,j) - Q^{kj} I_3(k,k,j)}{Q^{jj}Q^{kk} - (Q^{kj})^2} \\
& + d(\rho)\sigma^{k\ell}(\rho) \frac{1}{Q^{kk}} I_3(k,k,k) \\
& - d(\rho)\sigma^{k\ell}(\rho) \sum_{n} \frac{T^{nn} I_3(k,k,n) - R^{kn} I_3(k,n,n)}{Q^{kk}T^{nn} - (R^{kn})^2} \\
& - b\rho \sum_{n} r^{\ell n}(\rho) \frac{Q^{kk} I_3(k,n,n) - R^{kn} I_3(k,k,n)}{Q^{kk}T^{nn} - (R^{kn})^2} \\
& + \text{all of the above with } \ell \to k, k \to \ell \Bigg). \\
+ \eta^2 \left[(c-b^2)\rho + \frac{b^2}{\delta}\rho^2 \right] \Bigg(& \sum_{j,a}^{K} I_4(k,\ell,j,a) \\
& -2 \sum_{j}^{K} \sum_{m}^{M} I_4(k,\ell,j,m) + \sum_{n,m}^{M} I_4(k,\ell,n,m) \Bigg)
\end{aligned}
\tag{85}
$$

This last result completes the programme that we embarked upon at the beginning of this Section: we have derived a closed set of equations of motion for the teacher-student overlap R^{km} (68,71) and the student-student overlap $Q^{k\ell} = \left[c - b^2\right] W^{k\ell} + b^2 \Sigma^{k\ell}$ (78,84,85). These equations give us complete access to the dynamics of a neural network performing one-shot stochastic gradient descent on a data set generated by the hidden manifold model. We can now integrate these equations and substitute the values of the order parameters at any time into the expression for the generalisation error (28), thereby tracking the dynamics of the generalisation error at all times. We describe this procedure in more detail next.

5.2. Solving the equations of motion

The equations describing online learning that we have derived using the GET are valid for any choice of functions $f(x)$, $g(x)$ and $\tilde{g}(x)$. To solve the equations for a particular setup, one first needs to compute the three constants a, b, c (22) and to evaluate the averages yielding the functions I_3 and I_4. Choosing $g(x) = \tilde{g}(x) = \mathrm{erf}(x/\sqrt{2})$, these averages can be computed analytically [47]. Second, one needs to determine the spectral density of the feature matrix F_{ir}. In our experiments in the previous parts of this paper, we drew the entries of the feature matrix F_{ir} i.i.d. from the normal distribution with mean zero and unit variance, so the limiting distribution of the eigenvalues ρ in the integrals (68) and (84) is the Marchenko-Pastur distribution (69) (but see also Sec. 5.2.2 for a non-random matrix $\boldsymbol{F}$).

We illustrate the use of the equations of motion in Fig. 7, where we plot the dynamics of the generalisation error (top left) and of the order parameters R^{km}, $\Sigma^{k\ell}$ and $W^{k\ell}$ computed during a single experiment (crosses) and as obtained from integration of the equations of motion (line). We provide a complete implementation of the equations of motion together with the code for our simulations, which can be found on GitHub at `https://github.com/sgoldt/hidden-manifold-model`.

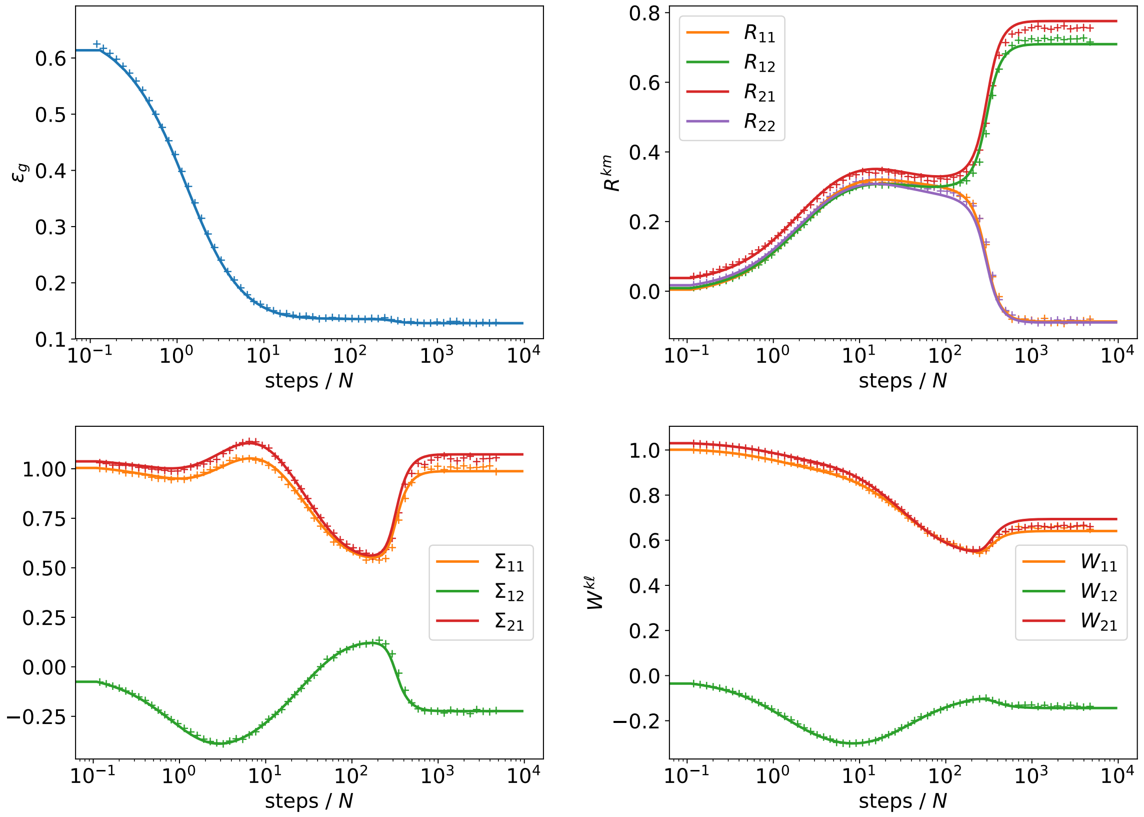


Figure 7: The analytical description of the hidden manifold generalisation dynamics matches experiments. We plot the time evolution of the generalisation error $\epsilon_g(\alpha)$ and the three order-parameters R^{km}, $\Sigma^{k\ell}$ and $W^{k\ell}$ obtained by integration of the ODEs (solid) and from a single run of SGD (2) (crosses). $f(x) = \operatorname{sgn}(x), g(x) = \tilde{g}(x) = \operatorname{erf}(x/\sqrt{2}), N = 8000, D = 4000, M = 2, K = 2, \eta = 1$.

5.2.1. Specialisation of hidden units in the hidden manifold model

Closer inspection of the time evolution of the order parameters in Fig. 7 reveals the mechanism of learning by the neural network. If we look at the order parameter R^{km} (top right), we see that during the initial decay of the generalisation error up to a time $t = {}^{\mu}\!/\!_{N} \sim 10$, all elements of the matrix R^{km} increase from roughly 0 to a constant value, $R^{km} \simeq 0.35$. In other words, the correlations between the pre-activation λ^k of any student node and the pre-activation ν^m of any teacher node is roughly the same. As training continues, the student nodes “specialise”: the pre-activation of one student node becomes strongly correlated with the pre-activation of only a single teacher node. In the example shown in Fig. 7, we have strong correlations between the pre-activation of the first student and the second teacher node (R^{12}), and between the second student and first teacher node (R^{21}). The specialisation of the teacher-student correlations is actually preceded by a de-correlation of the student units, as can be seen from the plots of the latent and ambient student-student overlaps $\Sigma^{k\ell}$ and $W^{k\ell}$, respectively (bottom of Fig. 7). Similar specialisation transitions have been observed in the vanilla teacher-student setup for both online and batch learning [4]. In these setups, there is a large drop in the generalisation error as student nodes specialise, leading to the appearance of the plateaus in the learning dynamics, whereas here, we see but a tiny change in generalisation error.

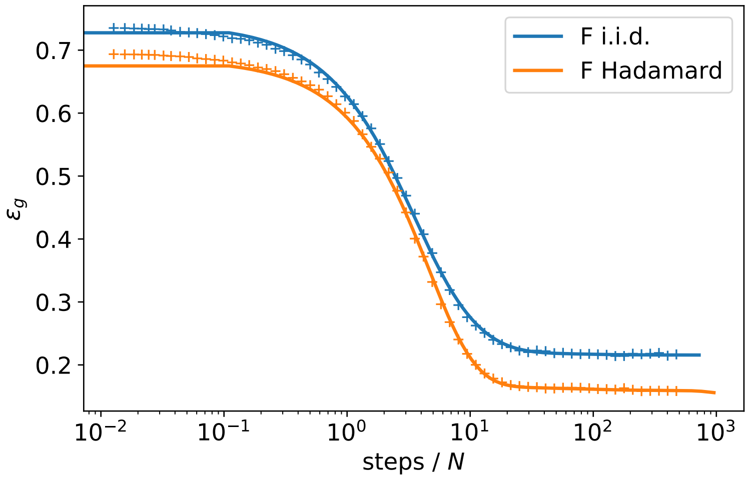


Figure 8: The ODE analysis is asymptotically correct for non-random feature matrices $\boldsymbol{F}$. We plot the time evolution of the generalisation error ϵ_g obtained by integration of the ODEs (solid) and from a single run of SGD (2) (crosses) for two different matrices $\boldsymbol{F}$: (i) elements F_{ir} are drawn i.i.d. from the standard normal distribution (blue); (ii) $\boldsymbol{F}$ is a Hadamard matrix [66] $f(x) = \operatorname{sgn}(x), g(x) = \tilde{g}(x) = \operatorname{erf}(x/\sqrt{2}), N = 1023, D = 1023, M = 2, K = 2, \eta = 0.2.$

5.2.2. Using non-random feature matrices $\boldsymbol{F}$

Our derivation of the ODEs for online learning did not assume that the feature matrix $\boldsymbol{F}$ be random; instead, we only require the balance condition stated in Eq. (14) as well as the normalisation conditions (15, 16). To illustrate this point, we plot the examples of online learning dynamics with $M = K = 2$ in Fig. 8, with the prediction from the ODE as solid lines and the result of a single simulation with crosses. In blue, we show results where the elements of F_{ir} were drawn i.i.d. from the standard normal distribution. For the experiment in orange, $\boldsymbol{F} = \boldsymbol{H}_N$, where $\boldsymbol{H}_N$ is a Hadamard matrix [66]. Hadamard matrices are $N \times N$ matrices, so $\delta = 1$ and are popular in error-correcting codes such as the Reed-Muller code [67,68]. They can be defined via the relation

$$\boldsymbol{H}_N \boldsymbol{H}_N^\top = N \mathbb{I}_N \,, \tag{86}$$

where $\mathbb{I}_N$ is the $N \times N$ identity matrix. As we can see from Fig. 8, the ODEs capture the generalisation dynamics of the Hadamard-case just as well. Another application explored in [17,25] is to consider data coming from a trained GAN.

6. The relation of the HMM to learning with random features

As we alluded to in the introduction, there exists a deep relation between the hidden manifold model (HMM) and random feature learning with unstructured i.i.d. input data. The idea of learning with random features goes back to the mechanical perceptron of the 1960s [14] and was popularised by the random kitchen sinks of Recht and Rahimi [15,16]. This connection is interesting beyond the formal analogy since random features are studied in detail, and can approximate kernel methods to arbitrary precision [15,16].

Let us introduce random feature learning using notation that will make the relation to the hidden manifold model explicit (see also Fig. 9). The P samples of the input data live in dimension D and are collected in the matrix $\boldsymbol{C} \in \mathbb{R}^{P \times D}$. The neural network used for random kitchen-sink or random feature learning consists of a first layer projecting the D-dimensional

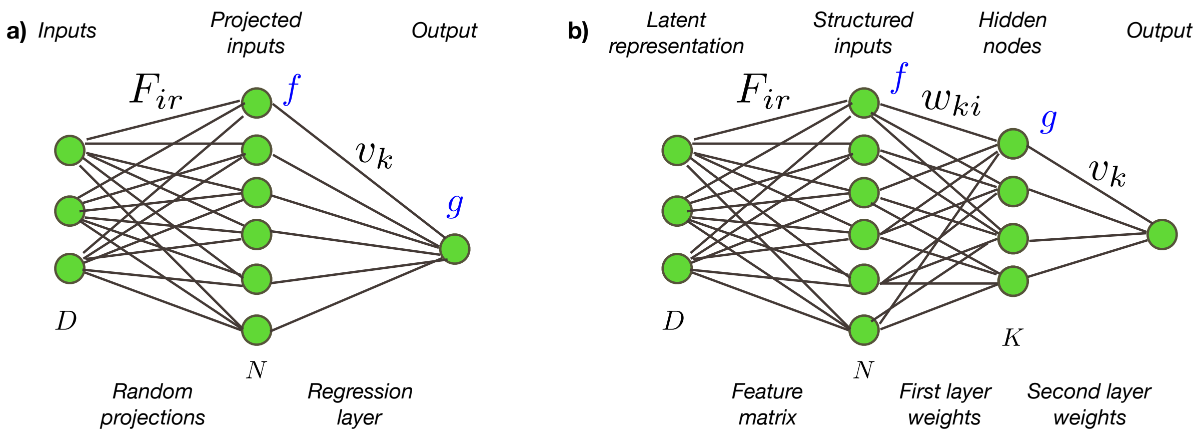


Figure 9: The relation between the hidden manifold model and learning with random features. *(Left)* The key idea of random feature learning is to first project an input $\boldsymbol{c}_\mu$ to a higher-dimensional space using a random projection matrix $\boldsymbol{F}$. Then, a non-linearity f is applied, and conventionally, a layer of weights is found via linear ridge regression. *(Right)* In this paper, we study a setup where structured inputs are trained by projecting i.i.d. vectors $\boldsymbol{c}_\mu$ (the latent representation) to the higher-dimensional ambient space before applying a non-linear function $f(x)$. Then, we train two-layer neural networks with K hidden units and first- and second-layer weights $\boldsymbol{W}$ and $\boldsymbol{v}$, respectively, on this data set.

input into an N-dimensional space via a set of random projections (a.k.a. features) collected in a matrix $\boldsymbol{F} \in \mathbb{R}^{D\times N}$, and a component-wise non-linear function $f(\cdot)$, so that the output of the first layer of the neural network is given by $\boldsymbol{X} = f(\boldsymbol{CF}/\sqrt{D}) \in \mathbb{R}^{P\times N}$. Finally, a second layer of weights is obtained via linear ridge regression on the projections $\boldsymbol{X}$ and the labels $\boldsymbol{y} \in \mathbb{R}^P$. The setting of the present paper corresponds to using a committee machine with K hidden units to perform regression on the projections $\boldsymbol{X}$ with labels $\boldsymbol{y}$ (7) via stochastic gradient descent.

The analytical solution of the dynamics of (one-pass) stochastic gradient descent algorithm (2) for committee machines that we derive in Sec. 5 applies to the random projections $\boldsymbol{X}$ when the input data $\boldsymbol{C}$ is an element-wise i.i.d. random matrix, and the output labels are generated by a teacher committee machine with random weights (7) that acts on the input data $\boldsymbol{C}$. Crucially, we do not require the feature matrix $\boldsymbol{F}$ to be random, and instead only require a certain "balance" condition that we state precisely in Eq. (14). The thermodynamic limit we consider takes the dimension D, the number of random features N and the number of samples P to infinity with fixed ratios $\alpha = P/N = \mathcal{O}(1)$, $\delta = D/N = \mathcal{O}(1)$. This limit is quite remarkable comparing to existing literature on learning with random features. On the one hand, considering a number of samples that is proportional to the input dimension is very interesting, as most existing approaches require the number of samples to be much larger, which in turn makes the learning problem simpler. On the other hand, having a number of features that is of the same order as the input dimension limits the performance of these methods, as it is known that for random feature learning to be powerful, the number of features should be much larger than the input dimension, see e.g. [18]. For instance, one can reinterpret the ODE analysis in Sec. 5 as the analysis of the performance of SGD applied to random features followed by a committee machine.

It is remarkable that concurrently, the same setting and scaling limit as in the present paper was considered and analysed for random feature learning with linear teacher, and ridge regression learning in [18], and for max-margin linear classifiers in [19]. Both these papers consider full batch learning, i.e. all samples are used at the same time, which makes them different from our online (one-pass SGD) analysis. The principles underlying the analytic solution of [18, 19] boil down to the Gaussian Equivalence Principle, which is stated and used independently in those papers. More precisely in [18] the full-batch formulas are proven from random matrix theory, while in [19] the formulas are only conjectured. Note also that one of the main points of [18] was the double descent behaviours that we also reproduced in Sec. 3.4.

We note that thanks to the Gaussian Equivalence Theorem, the full batch learning in the present model – with committee teacher and committee student and deterministic matrix $\boldsymbol{F}$ – is also amenable to analysis with the replica method, thus generalising the results of [18, 19]. We give a rough sketch of how the formulas of [18, 19] can be obtained as an application of the GET in connection with the replica method from statistical physics in Appendix C. Since this is a rather general approach to studying the full batch learning problem, we leave its full investigation for our future work.

Acknowledgements

We would like to thank the Kavli Institute For Theoretical Physics for its hospitality during an extended stay, during which parts of this work were conceived and carried out. We acknowledge funding from the ERC under the European Union's Horizon 2020 Research and Innovation Programme Grant Agreement 714608-SMiLe, from "Chaire de recherche sur les modèles et sciences des données", Fondation CFM pour la Recherche-ENS, and from the French National Research Agency (ANR) grant PAIL. This research was supported in part by the National Science Foundation under Grant No. NSF PHY-1748958

References

[1] V. Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.

[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning.* MIT Press, 2012.

[3] H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056–6091, 1992.

[4] A. Engel and C. Van den Broeck. *Statistical Mechanics of Learning.* Cambridge University Press, 2001.

[5] L. Zdeborová and F. Krzakala. Statistical physics of inference: thresholds and algorithms. *Adv. Phys.*, 65(5):453–552, 2016.

[6] Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1998.

[7] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189–208, 1983.

[8] J.A. Costa and A.O. Hero. Learning intrinsic dimension and intrinsic entropy of high-dimensional datasets. In *2004 12th European Signal Processing Conference*, pages 369–372, 2004.

[9] E. Levina and P.J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems 17*, 2004.

[10] S. Spigler, M. Geiger, and M. Wyart. Asymptotic learning curves of kernel methods: empirical data v.s. Teacher-Student paradigm. *arXiv:1905.10843*, 2019.

[11] E. Gardner and B. Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983–1994, 1989.

[12] D. Saad and S.A. Solla. Exact Solution for On-Line Learning in Multilayer Neural Networks. *Phys. Rev. Lett.*, 74(21):4337–4340, 1995.

[13] S. Goldt, M.S. Advani, A.M. Saxe, F. Krzakala, and L. Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In *Advances in Neural Information Processing Systems 33*, 2019.

[14] F. Rosenblatt. *Principles of Neurodynamics.* Spartan, 1962.

[15] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[16] A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.

[17] C. Louart, Z. Liao, and Romain Couillet. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248, 2018.

[18] S. Mei and A. Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv:1908.05355*, 2019.

[19] A. Montanari, F. Ruan, Y. Sohn, and J. Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv:1911.01544*, 2019.

[20] W. Hachem, P. Loubaton, and J. Najim. Deterministic equivalents for certain functionals of large random matrices. *Ann. Appl. Probab.*, 17(3):875–930, 06 2007.

[21] X. Cheng and A. Singer. The spectrum of random inner-product kernel matrices. *Random Matrices: Theory and Applications*, 2(04):1350010, 2013.

[22] Z. Fan and A. Montanari. The spectral norm of random inner-product kernel matrices. *Probability Theory and Related Fields*, 173(1-2):27–85, 2019.

[23] J. Pennington and P. Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pages 2637–2646, 2017.

[24] M.E.A. Seddik, M. Tamaazousti, and R. Couillet. Kernel random matrices of large concentrated data: the example of gan-generated images. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7480–7484. IEEE, 2019.

[25] Anonymous. Random matrix theory proves that deep learning representations of GAN-data behave as Gaussian mixtures. In *Submitted to International Conference on Learning Representations*, 2020. under review.

[26] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[27] A.B. Patel, M.T. Nguyen, and R. Baraniuk. A probabilistic framework for deep learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2558–2566. Curran Associates, Inc., 2016.

[28] Marc M. Mean-field message-passing equations in the hopfield model and its generalizations. *Physical Review E*, 95(2):022117, 2017.

[29] M. Gabrié, A. Manoel, C. Luneau, J. Barbier, N. Macris, F. Krzakala, and L. Zdeborová. Entropy and mutual information in models of deep neural networks. In *Advances in Neural Information Processing Systems 31*, pages 1826–1836, 2018.

[30] E. Mossel. Deep learning and hierarchical generative models. *arXiv:1612.09057*, 2018.

[31] S. Chung, Daniel D. Lee, and H. Sompolinsky. Classification and Geometry of General Perceptual Manifolds. *Physical Review X*, 8(3):31003, 2018.

[32] Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *bioRxiv*, page 644658, 2019.

[33] P. Rotondo, M. Cosentino Lagomarsino, and M. Gherardi. Counting the learnable functions of structured data. *arXiv:1903.12021*, 2019.

[34] T.M. Cover. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965.

[35] Y. Yoshida and M. Okada. Data-dependence of plateau phenomenon in learning with neural network — statistical mechanical analysis. In *Advances in Neural Information Processing Systems 32*, pages 1720–1728. 2019.

[36] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft. Convergent Learning: Do different neural networks learn the same representations? In D. Storcheus, A. Rostamizadeh, and S. Kumar, editors, *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pages 196–212. PMLR, 2015.

[37] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability. In *Advances in Neural Information Processing Systems 30*, pages 6076–6085. Curran Associates, Inc., 2017.

[38] A.S. Morcos, M. Raghu, and S. Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems 31*, pages 5727–5736, 2018.

[39] T.L.H. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65(2):499–556, 1993.

[40] M.S. Advani and A.M. Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv:1710.03667*, 2017.

[41] B. Aubin, A. Maillard, J. Barbier, F. Krzakala, N. Macris, and L. Zdeborová. The committee machine: Computational to statistical gaps in learning a two-layers neural network. In *Advances in Neural Information Processing Systems 31*, pages 3227–3238, 2018.

[42] J. Barbier, F. Krzakala, N. Macris, L. Miolane, and L. Zdeborová. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences*, 116(12):5451–5460, 2019.

[43] Y. Yoshida, R. Karakida, M. Okada, and S.-I. Amari. Statistical mechanical analysis of learning dynamics of two-layer perceptron with multiple output units. *Journal of Physics A: Mathematical and Theoretical*, 52(18), 2019.

[44] R. Ge, J.D. Lee, and T. Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv:1711.00501*, 2017.

[45] Y. Li and Y. Y. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

[46] S. Arora, N. Cohen, W. Hu, and Y. Luo. Implicit Regularization in Deep Matrix Factorization. In *Advances in Neural Information Processing Systems 33*, 2019.

[47] M. Biehl and H. Schwarze. Learning by on-line gradient descent. *J. Phys. A. Math. Gen.*, 28(3):643–656, 1995.

[48] M. Biehl, P. Riegler, and C. Wöhler. Transient dynamics of on-line learning in two-layered neural networks. *Journal of Physics A: Mathematical and General*, 29(16), 1996.

[49] H. Schwarze. Learning a rule in a multilayer neural network. *Journal of Physics A: Mathematical and General*, 26(21):5781–5794, 1993.

[50] H.H. Yang and S.-I. Amari. The Efficiency and the Robustness of Natural Gradient Descent Learning Rule. In M I Jordan, M J Kearns, and S A Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 385–391, 1998.

[51] M. Rattray, D. Saad, and S.-I. Amari. Natural Gradient Descent for On-Line Learning. *Physical Review Letters*, 81(24):5461–5464, 1998.

[52] D.P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[54] D. Michelsanti and Z. Tan. Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification. *arXiv:1709.01703*, 2017.

[55] S Spigler, M Geiger, S d'Ascoli, L Sagun, G Biroli, and M Wyart. A jamming transition from under-to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, 2019.

[56] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[57] M. Opper. Statistical mechanics of learning: Generalization. *The Handbook of Brain Theory and Neural Networks,*, pages 922–925, 1995.

[58] P. Nakkiran, G. Kaplun, T. Bansal, Y. amd Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv:1912.02292*, 2019.

[59] Q. Le, T. Sarlós, and A. Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013.

[60] M. Moczulski, M. Denil, J. Appleyard, and N. de Freitas. Acdc: A structured efficient linear layer. *arXiv:1511.05946*, 2015.

[61] F.X. Yu, A.T. Suresh, K.M. Choromanski, D.N. Holtmann-Rice, and S. Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016.

[62] W. Kinzel and P. Ruján. Improving a Network Generalization Ability by Selecting Examples. *EPL (Europhysics Letters)*, 13(5):473–477, 1990.

[63] D. Saad and S.A. Solla. On-line learning in soft committee machines. *Phys. Rev. E*, 52(4):4225–4243, 1995.

[64] D. Saad. *On-line learning in neural networks*, volume 17. Cambridge University Press, 2009.

[65] V.A. Marchenko and L.A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.

[66] J. Hadamard. Resolution d'une question relative aux determinants. *Bull. des sciences math.*, 2:240–246, 1893.

[67] D.E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Transactions of the IRE professional group on electronic computers*, 3:6–12, 1954.

[68] I. S Reed. A class of multiple-error-correcting codes and the decoding scheme. Technical report, Massachusetts Inst of Tech Lexington Lincoln Lab, 1953.

[69] E. Gardner. Maximum storage capacity in neural networks. *EPL (Europhysics Letters)*, 4(4):481, 1987.

[70] M. Mézard, G. Parisi, and M. Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

[71] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

A. Proof of the Gaussian Equivalence Theorem

A.1. Nonlinear functions of weekly correlated Gaussian random variables

In order to prove the GET theorem 4.1 we first formulate and establish some lemmas concerning the correlations between nonlinear functions of weakly correlated random variables.

A.1.1. Correlations of two functions

Lemma A.1. *Given $n+p$ random variables organised in two vectors,*

$$x = \begin{pmatrix} x^1 \\ \cdot \\ \cdot \\ x^n \end{pmatrix} \quad , \quad y = \begin{pmatrix} y^1 \\ \cdot \\ \cdot \\ y^p \end{pmatrix} \quad , \tag{87}$$

with a joint Gaussian distribution, denote by $\mathbb{E}$ the expectation with respect to this distribution. The first moments are supposed to vanish,

$$\mathbb{E}\, x_i = 0 \quad , \quad \mathbb{E}\, y_j = 0 \quad , \tag{88}$$

and we denote by $Q, R, \varepsilon S$ the covariances :

$$\mathbb{E}\left[x_i x_j\right] = Q_{ij} \quad , \quad \mathbb{E}\left[y_i y_j\right] = R_{ij} \quad , \quad \mathbb{E}\left[x_i y_j\right] = \varepsilon S_{ij} \ . \tag{89}$$

Let $f(x)$ and $g(y)$ be two functions of x and y respectively regular enough so that $\mathbb{E}_x[x_i f(x)]$, $\mathbb{E}_x[x_i x_j f(x)]$, $\mathbb{E}_y[y_i f(y)]$ and $\mathbb{E}_y[y_i y_j f(y)]$ exist, where $\mathbb{E}_x$ denotes the expectation with respect to the distribution $\mathcal{N}(a,Q)$ of x and $\mathbb{E}_y$ denotes the expectation with respect to the distribution $\mathcal{N}(b,R)$ of x.

Then, in the $\varepsilon \to 0$ limit:

$$\begin{aligned} \mathbb{E}\left[f(x)g(y)\right] &= \mathbb{E}_x[f(x)]\,\mathbb{E}_y[g(y)] \\ &+ \varepsilon \sum_{i=1}^{n}\sum_{j=1}^{p} \mathbb{E}_x[x_i f(x)] \left(Q^{-1}SR^{-1}\right)_{ij} \mathbb{E}_y[y_j g(y)] + \mathcal{O}(\varepsilon^2) \ . \end{aligned} \tag{90}$$

Proof. The result is obtained by a straightforward expansion in ε.

The joint distribution of x and y is

$$P(x,y) = \frac{1}{Z} \exp\left[-\frac{1}{2}\begin{pmatrix} x & y \end{pmatrix} M^{-1} \begin{pmatrix} x \\ y \end{pmatrix}\right] \tag{91}$$

where

$$M = \begin{pmatrix} Q & \varepsilon S \\ \varepsilon S^T & R \end{pmatrix} . \tag{92}$$

One can expand the inverse matrix M^{-1} to first order in ε:

$$M^{-1} = \begin{pmatrix} Q^{-1} & 0 \\ 0 & R^{-1} \end{pmatrix} - \varepsilon \begin{pmatrix} 0 & Q^{-1}SR^{-1} \\ R^{-1}S^TQ^{-1} & 0 \end{pmatrix} \tag{93}$$

and substitute this into the joint distribution (91). This gives:

$$\begin{aligned} P(x,y) &= \frac{1}{Z}\exp\left[-\frac{1}{2}\begin{pmatrix} x & y \end{pmatrix}\begin{pmatrix} Q^{-1} & 0 \\ 0 & R^{-1} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}\right] \\ & \left[1 + \varepsilon \sum_{i=1}^{n}\sum_{j=1}^{p} x_i \left(Q^{-1}SR^{-1}\right)_{ij} y_j + \mathcal{O}(\varepsilon^2)\right] \ . \end{aligned} \tag{94}$$

Using this expression, the result (90) follows immediately. □

An immediate application of the lemma to the case when $n = p = 1$ is the following. Consider two Gaussian random variables u_1, u_2 with mean zero and covariance

$$\mathbb{E}\,[u_1^2] = 1 \quad ; \quad \mathbb{E}\,[u_2^2] = 1 \quad ; \quad \mathbb{E}\,[u_1 u_2] = \varepsilon m_{12} \ , \tag{95}$$

and two functions f_1 and f_2. Define, for $\in \{1, 2\}$:

$$a_i = \langle f_i(u) \rangle \quad ; \quad b_i = \langle u f_i(u) \rangle \tag{96}$$

where $\langle . \rangle$ denotes the average over the distribution of the random Gaussian variable u distributed as $\mathcal{N}(0, 1)$.

Then, in the $\varepsilon \to 0$ limit, the correlation between $f(u_1)$ and $g(u_2)$ is given by

$$\mathbb{E}\,[f_1(u_1) f_2(u_2)] = a_1 a_2 + \varepsilon m_{12} b_1 b_2 + \mathcal{O}(\varepsilon^2)\,. \tag{97}$$

This means that, if we consider centered functions $\tilde{f}_i(u_i) = f_i(u_i) - a_i$, their covariance is

$$\mathbb{E}\,[\tilde{f}_1(u_1) \tilde{f}_2(u_2)] = +\varepsilon m_{12} b_1 b_2 + \mathcal{O}(\varepsilon^2)\,. \tag{98}$$

This result generalises to correlation functions of higher order, as stated in the following lemma.

A.1.2. Higher-order correlations

Lemma A.2. *Consider m Gaussian random variables $u_1, \ldots, u_m$ with mean zero and covariance*

$$\forall i : \ \mathbb{E}\,[u_i^2] = 1 \quad ; \quad \forall i \neq j : \ \mathbb{E}\,[u_i u_j] = \varepsilon m_{ij} \ , \tag{99}$$

and m functions f_1,..., f_m. Define as before :

$$a_i = \langle f_i(u) \rangle \quad ; \quad b_i = \langle u f_i(u) \rangle, \quad i \in \{1, \ldots, m\} \tag{100}$$

and define the centered functions as

$$\tilde{f}_i(u) = f_i(u) - a_i \ , \tag{101}$$

then

$$\begin{aligned} \lim_{\varepsilon \to 0} \frac{1}{\varepsilon^{p/2}} \mathbb{E}\, \tilde{f}_1(u_1) \ldots \tilde{f}_m(u_m) \ &= \ b_1 \ldots b_m \sum_{\sigma \in \Pi} m_{\sigma_1 \sigma_2} m_{\sigma_{p-1} \sigma_p} \quad \textit{if } m \textit{ is even} \\ &= \ 0 \quad \textit{if } m \textit{ is odd} \end{aligned} \tag{102}$$

where Π denotes all the $m!/(2^{m/2}(m/2)!)$ partitions of $\{1, \ldots, m\}$ into $m/2$ disjoint pairs. This result means that, for the moments involving only different indices, the random variables $\tilde{f}_1(u_1)/\sqrt{\varepsilon}, \ldots, \tilde{f}_m(u_m)/\sqrt{\varepsilon}$ behave, in the $\varepsilon \to 0$ limit, like Gaussian variables with a covariance matrix $b_i b_j m_{ij}$.

Proof. The covariance matrix U of the variables $u_1, \ldots, u_m$ has elements 1 on the diagonal, and elements of order ε out of the diagonal: $U = \mathbb{I} + \varepsilon m$. One can expand U^{-1} in powers of ε:

$$U^{-1} = \sum_{p=0}^{\infty} (-\varepsilon)^p m^p\,. \tag{103}$$

The integration measure over the variables $u_1, \ldots, u_m$ can be expanded as:

$$\sqrt{(2\pi)^m \text{det} M}\ e^{-\frac{1}{2} \sum_i u_i^2} \prod_{p=1}^{\infty} G_p(u_1, \ldots, u_m) \tag{104}$$

where

$$G_p(u_1,\dots,u_m) = 1 + \left(-\frac{\varepsilon}{2}\right)^p \sum_{ij} (m^p)_{ij} u_i u_j + \frac{1}{2!}\left(-\frac{\varepsilon}{2}\right)^{2p} \sum_{ijk\ell} (m^p)_{ij}(m^p)_{k\ell} u_i u_j u_k u_\ell + \dots \quad (105)$$

When we compute the integral of $\tilde{f}_1(u_1)\dots\tilde{f}_m(u_m)$ with the measure (104, because of the fact that $\langle\tilde{f}_i(u_i)\rangle = 0$, we need to include terms coming from $\prod_p G_p(u_1,\dots,u_p)$ that involve at least one power of each of the variables $u_1,\dots,u_m$.

When m is even, say $m = 2r$, for $\varepsilon \to 0$, the term of this kind with the smallest power of ε is the monomial $u_1 \dots u_{2r}$ that comes from the rth order term in G_1. This gives:

$$\mathbb{E}\, f_1(u_1)\dots f_{2r}(u_{2r}) = \frac{1}{r!}\left(\frac{\varepsilon}{2}\right)^r \hat{\sum}_{i_1 j_1 \dots i_r j_r} m_{i_1 j_1} m_{i_r j_r} + \mathcal{O}(\varepsilon^{r+1}) \;, \quad (106)$$

where the sum $\hat{\sum}_{i_1 j_1 \dots i_r j_r}$ runs over all permutations of the indices $1,\dots,2r$. This proves (102) for m even.

When m is odd, $m = 2r+1$, for $\varepsilon \to 0$, the leading terms coming from $\prod G_p$ that give a non-zero result are monomials of the type $u_1^1 u_2 \dots u_{2r+1}$. They are of order $\mathcal{O}(\varepsilon^{r+1})$. This proves (102) for m odd. □

Corollary A.3. *In the special case $m = 3$, we get*

$$\mathbb{E}\,[f_1(u_1)f_2(u_2)f_3(u_3)] = a_1a_2a_3 + \varepsilon(a_1 m_{23} b_2 b_3 + a_2 m_{13} b_1 b_3 + a_3 m_{12} b_1 b_2)\,. \quad (107)$$

A.2. Proof of the Theorem

The proof is based on the computation of moments of the variables λ^k and ν^m, showing that, in the thermodynamic limit, all the moments are those of Gaussian random variables. Here we shall explicit the proof up to fourth order moments, and leave to the reader the generalisation to higher order moments.

A.2.1. Covariances

We first compute the covariance matrix $G^{k\ell} = \mathbb{E}[\tilde{\lambda}^k\tilde{\lambda}^\ell]$:

$$\begin{aligned} G^{k\ell} &= \frac{1}{N}\sum_{i,j} w_i^k w_j^\ell \mathbb{E}\,(f(u_i)-a)(f(u_j)-a) && (108)\\ &= (c-a^2)W^{k\ell} + \frac{1}{N}\sum_{i\neq j} w_i^k w_j^\ell \mathbb{E}\,(f(u_i)-a)(f(u_j)-a)\;. && (109)\end{aligned}$$

In the last piece, we need to compute $\mathbb{E}\,[(f(u_i)-a)(f(u_j)-a)]$ for two Gaussian random variables u_i and u_j which are weakly correlated in the large N limit. In fact, as $i \neq j$:

$$\mathbb{E}\, u_i u_j = U_{ij} \quad (110)$$

is of order $1/\sqrt{D}$. In the thermodynamic limit, we can apply the lemma (A.1) which gives:

$$\mathbb{E}\, f(u_i)f(u_j) = a^2 + b^2 \frac{1}{D}\sum_{r=1}^{D} F_{ir}F_{jr}\,. \quad (111)$$

From(109, 111) we get the covariance of λ variables as written in (19). The covariance $\mathbb{E}\,[\nu^m\nu^n]$ is analogous.

We now compute the covariance $\mathbb{E}\,[\tilde{\lambda}^k\nu^m]$. This is equal to

$$\frac{1}{\sqrt{N}}\sum_{i=1}^{N} w_i^k \frac{1}{\sqrt{D}}\sum_{r=1}^{D} \tilde{w}_r^m\; \mathbb{E}\,[f(u_i)C_r]\,. \quad (112)$$

The two variables u_i and c_r are Gaussian random variables with a correlation

$$\mathbb{E}\left[u_i C_r\right] = \frac{1}{\sqrt{D}} F_{ir} \tag{113}$$

which goes to zero as $\mathcal{O}(1/\sqrt{N})$ in the thermodynamic limit. We can thus use Lemma (A.1), and more precisely Eq. (98), to get

$$\mathbb{E}\left[f(u_i) C_r\right] = \frac{1}{\sqrt{D}} F_{ir} \langle u f(u) \rangle \langle C_r^2 \rangle = \frac{b}{\sqrt{D}} F_{ir}\,. \tag{114}$$

Using this result in (112) gives Eq. (20).

A.2.2. Fourth moments of $\tilde{\lambda}^k$ variables

We study the fourth moment defined as:

$$G^{k_1 k_2 k_3 k_4} = \langle \tilde{\lambda}^{k_1} \tilde{\lambda}^{k_2} \tilde{\lambda}^{k_3} \tilde{\lambda}^{k_4} \rangle = \frac{1}{N^2} \sum_{i_1, i_2, i_3, i_4} w_{i_1}^k w_{i_2}^\ell w_{i_3}^{k'} w_{i_4}^{\ell'} \langle \tilde{f}(u_{i_1}) \tilde{f}(u_{i_2}) \tilde{f}(u_{i_3}) \tilde{f}(u_{i_4}) \rangle \tag{115}$$

where $\tilde{f}(u) = f(u) - a$ is the centered function.

We shall decompose the sum over i_1, i_2, i_3, i_4 depending on the number of distinct indices there are.

Distinct indices Let us study the first piece of the fourth moment $\langle \lambda^{k_1} \lambda^{k2} \lambda^{k_3} \lambda^{k_4} \rangle$:

$$G_4^{k_1 k_2 k_3 k_4} = \frac{1}{N^2} \sum_{i_1, i_2, i_3, i_4}' w_{i_1}^{k_1} w_{i_2}^{k_2} w_{i_3}^{k_3} w_{i_4}^{k_4} \langle \tilde{f}(u_{i_1}) \tilde{f}(u_{i_2}) \tilde{f}(u_{i_3}) \tilde{f}(u_{i_4}) \rangle \tag{116}$$

where the sum runs over four indices i_1, i_2, i_3, i_4 which are distinct from each other. We can use the factorisation property of the 4th moments of $f(u)$ of lemma (A.2). This gives

$$\begin{aligned} G_4^{k_1 k_2 k_3 k_4} &= \frac{1}{N^2} \sum_{i_1, i_2, i_3, i_4}' w_{i_1}^{k_1} w_{i_2}^{k_2} w_{i_3}^{k_3} w_{i_4}^{k_4} \left[\langle \tilde{f}(u_{i_1}) \tilde{f}(u_{i_2}) \rangle \langle \tilde{f}(u_{i_3}) \tilde{f}(u_{i_4}) \rangle + 2 \text{ perm.} \right] \\ &= \left(\left[\frac{1}{N} \sum_{i_1, i_2}' w_{i_1}^{k_1} w_{i_2}^{k_2} \langle f(u_{i_1}) f(u_{i_2}) \rangle \right] \left[\frac{1}{N} \sum_{i_3, i_4}' w_{i_3}^{k_3} w_{i_4}^{k_4} \langle f(u_{i_3}) f(u_{i_4}) \rangle \right] - \text{Corr.} \right) \\ &\quad +2 \text{ perm.}. \end{aligned} \tag{117}$$

The correction terms come from pieces where the intersection between $\{i_1, i_2\}$ and $\{i_3, i_4\}$ is non-empty. If we first neglect this correction, we find

$$G_4^{k_1 k_2 k_3 k_4} = b^4 \left[\left(\Sigma^{k_1 k_2} - W^{k_1 k_2} \right) \left(\Sigma^{k_3 k_4} - W^{k_3 k_4} \right) + 2 \text{ perm.} \right]. \tag{118}$$

Now we shall show that the corrections are negligible. Consider the term $i_1 = i_3$, $i_2 \neq i_4$. This gives a correction

$$- \frac{1}{N^2} \sum_{i_1, i_2, i_4}' w_{i_1}^{k_1} w_{i_2}^{k_2} w_{i_1}^{k_3} w_{i_4}^{k_4} \left[\langle \tilde{f}(u_{i_1}) \tilde{f}(u_{i_2}) \rangle \langle \tilde{f}(u_{i_1}) \tilde{f}(u_{i_4}) \rangle \right]. \tag{119}$$

Using (98)

$$\langle \tilde{f}(u_{i_1}) \tilde{f}(u_{i_2}) \rangle = b^2 U_{i_1 i_2} = b^2 \frac{1}{D} F_{i_1 r} F_{i_2 r}\ , \tag{120}$$

we get the expression for the correction

$$-\frac{1}{N^2R^2}\langle uf(u)\rangle^4 \sideset{}{'}\sum_{i_1,i_2,i_4} w_{i_1}^{k_1}w_{i_2}^{k_2}w_{i_1}^{k_3}w_{i_4}^{k_4}F_{i_1r}F_{i_2r}F_{i_1s}F_{i_4s} = -\frac{1}{\sqrt{N}R^2}\sum_{r,s} S_{rs}^{k_1k_3}S_r^{k_2}S_s^{k_4}\ . \tag{121}$$

Using our hypothesis on the fact that the quantities S are of order one, this correction is clearly at most of order $\mathcal{O}(1/\sqrt{N})$, and therefore negligible.

The last correction that we need to consider is the term where $i_1 = i_3 = i$, and $i_2 = i_4 = j$. This gives

$$-\frac{1}{N^2}\sideset{}{'}\sum_{i,j} w_i^{k_1}w_j^{k_2}w_i^{k_3}w_j^{k_4}\langle\tilde{f}(u_i)\tilde{f}(u_j)\rangle^2 = -\frac{1}{NR^2}\langle uf(u)\rangle^4\sum_{r,s}\left[S_{rs}^{k_1k_3}S_{rs}^{k_2k_4} - S_{rrss}^{k_1k_3k_2k_4}\right]\ , \tag{122}$$

which is again negligible in the large N limit.

Three distinct indices Let us study the contributions to the fourth moment of λ coming from three distinct indices. We study the case where $i_1 = i_4$:

$$E^{k_1k_2k_3k_4} = \frac{1}{N^2}\sideset{}{'}\sum_{i_1,i_2,i_3} w_{i_1}^{k_1}w_{i_2}^{k_2}w_{i_3}^{k_3}w_{i_1}^{k_4}\langle\tilde{f}(u_{i_1})^2\tilde{f}(u_{i_2})\tilde{f}(u_{i_3})\rangle\,. \tag{123}$$

Using the expression for the third moment of functions of u_1, u_2, u_3 found in (107), we get:

$$\begin{aligned} E^{k_1k_2k_3k_4} &= cb^2\frac{1}{N^2}\sideset{}{'}\sum_{i_1,i_2,i_3} w_{i_1}^{k_1}w_{i_2}^{k_2}w_{i_3}^{k_3}w_{i_1}^{k_4} - \text{Corr.} \\ &= cb^2W^{k_1k_4}\left[\Sigma^{k_2k_3} - W^{k_2k_3}\right] - \text{Corr.} \end{aligned} \tag{124}$$

The corrections come from cases when $i_1 = i_2$ or $i_1 = i_3$. For instance the piece with $i_1 = i_2$ gives

$$-cb^2\frac{1}{NR}\sum_r S_r^{k_1k_2k_4}S_r^{k_3} \tag{125}$$

which is $\mathcal{O}(1/N)$ at most.

The only pieces that do not vanish in the large N limit are thus the pieces similar to the one computed in (124). Putting all of them together we find that the contribution to $\langle\tilde{\lambda}^{k_1}\tilde{\lambda}^{k_2}\tilde{\lambda}^{k_3}\tilde{\lambda}^{k_4}\rangle$ coming form pieces with exactly three distinct indices in i_1, i_2, i_3, i_4 is equal to:

$$G_3^{k_1k_2k_3k_4} = cb^2\big(X^{k_1k_2;k_3k_4} + X^{k_1k_3;k_2k_4} + X^{k_1k_4;k_2k_3} + X^{k_2k_3;k_1k_4} + X^{k_2k_4;k_1k_3} + X^{k_3k_4;k_1k_2}\big)$$

where

$$X^{k_1k_2;k_3k_4} = W^{k_1k_2}\left[\Sigma^{k_3k_4} - W^{k_3k_4}\right]\,. \tag{126}$$

Two distinct indices Let us now study the contribution to the fourth moment of λ coming from two distinct indices. We study first one piece of this contribution to the fourth moment, corresponding to $i_1 = i_2 = i$, $i_3 = i_4 = j$:

$$F^{k_1k_2k_3k_4} = \frac{1}{N^2}\sideset{}{'}\sum_{i,j} w_i^{k_1}w_i^{k_2}w_j^{k_3}w_j^{k_4}\langle\tilde{f}(u_i)^2\tilde{f}(u_j)^2\rangle\,. \tag{127}$$

To leading order in the thermodynamic limit, we can write

$$\langle\tilde{f}(u_i)^2\tilde{f}(u_j)^2\rangle = c^2 \tag{128}$$

and therefore

$$F^{k_1k_2k_3k_4} = c^2 W^{k_1k_2} W^{k_3k_4} \tag{129}$$

(the correction coming from $i = j$ being obviously at most $\mathcal{O}(1/N)$).

We study now the second piece of this contribution to the fourth moment, corresponding to $i_1 = i_2 = i_3 = i$, $i_4 = j$. This is equal to

$$\frac{1}{N^2} \sum_{i,j}{}' w_i^{k_1} w_i^{k_2} w_i^{k_3} w_j^{k_4} \langle \tilde{f}(u_i)^3 \tilde{f}(u_j) \rangle \, . \tag{130}$$

Using

$$\langle \tilde{f}(u_i)^3 \tilde{f}(u_j) \rangle = b \langle u \tilde{f}(u)^3 \rangle \frac{1}{D} \sum_r F_{ir} F_{jr} \ , \tag{131}$$

this gives

$$b \langle u \tilde{f}(u)^3 \rangle \frac{1}{NR} \sum_r S_r^{k_1k_2k_3} S_r^{k_4} \tag{132}$$

and it is therefore negligible.

Therefore all the contributions to the fourth moment of λ coming from exactly two distinct indices are of the type (129). They give a total contribution:

$$G_2^{k_1k_2k_3k_4} = c^2 \left[W^{k_1k_2} W^{k_3k_4} + W^{k_1k_3} W^{k_2k_4} + W^{k_1k_4} W^{k_2k_4} \right] . \tag{133}$$

One distinct index The contribution to the fourth moment $\langle \lambda^{k_1} \lambda^{k_2} \lambda^{k_3} \lambda^{k_4} \rangle$ coming from $i_1 = i_2 = i_3 = i_4$ is clearly of $\mathcal{O}(1/N)$ and can be neglected.

Final result for the four-point correlation function of λ variables We can now put together all the contributions to the fourth moment $\langle \tilde{\lambda}^{k_1} \tilde{\lambda}^{k_2} \tilde{\lambda}^{k_3} \tilde{\lambda}^{k_4} \rangle$ coming form pieces with four distinct indices found in (118), those with three distinct indices found in (126), and those with two distinct indices found in (133). Defining

$$Y^{k_1k_2} = \Sigma^{k_1k_2} - W^{k_1k_2} \, , \tag{134}$$

and recalling the definition (126) of the X variables, we obtain:

$$\begin{aligned} \langle \tilde{\lambda}^{k_1} \tilde{\lambda}^{k_2} \tilde{\lambda}^{k_3} \tilde{\lambda}^{k_4} \rangle &= b^4 \big(Y^{k_1k_2} Y^{k_3k_4} + Y^{k_1k_3} Y^{k_2k_4} + Y^{k_1k_4} Y^{k_2k_3} \big) \\ &+ b^2 c \big(X^{k_1k_2;k_3k_4} + X^{k_1k_3;k_2k_4} \\ &+ X^{k_1k_4;k_2k_3} + X^{k_2k_3;k_1k_4} + X^{k_2k_4;k_1k_3} + X^{k_3k_4;k_1k_2} \big) \\ &+ c^2 \left[W^{k_1k_2} W^{k_3k_4} + W^{k_1k_3} W^{k_2k_4} + W^{k_1k_4} W^{k_2k_4} \right] . \end{aligned} \tag{135}$$

We can see that this is equal to

$$\left[b^2 Y^{k_1k_2} + c W^{k_1k_2} \right] \left[b^2 \rangle^2 Y^{k_3k_4} + c W^{k_3k_4} \right] + 2 \text{ perm.} \tag{136}$$

which proves that

$$\langle \tilde{\lambda}^{k_1} \tilde{\lambda}^{k_2} \tilde{\lambda}^{k_3} \tilde{\lambda}^{k_4} \rangle = \langle \tilde{\lambda}^{k_1} \tilde{\lambda}^{k_2} \rangle \langle \tilde{\lambda}^{k_3} \tilde{\lambda}^{k_4} \rangle + 2 \text{ permutations} \, . \tag{137}$$

With this, it is clear how to proceed with the calculation of the fourth moments involving λ and ν variables. We first need to study the moments with three λ and one ν, then moments with two λ and two ν, and finally the moments with one λ and three ν variables. In the interest of conciseness, we do not spell out the full details of this calculations here, which proceeds very similarly to the calculations performed hitherto.

The generalisation to higher moments of λ variables uses the same strategy, together with repeated use of Lemma A.2, and careful decomposition in subsets of distinct indices. In result, the set of λ variables has a Gaussian distribution in the thermodynamic limit.

B. The equations of motion do not close in the trivial basis

Here we give a short demonstration that it is not possible to close the equations for order parameters if we do not rotate their dynamics to the basis given by the eigenvectors of Ω, which is what we do in our derivation in Sec. 5.

B.1. Order parameters that are linear in the weights

To start with a variable that is linear in the weights, take the time-evolution of S_r^k. It is clear that the tensor structure of the result (43) will be of the form

$$(S_r^k)^{\mu+1} - (S_r^k)^{\mu} = -\frac{\eta}{N}\left[\sum_{\ell} D^{k\ell}\sum_s \Omega_{rs}S_s^{\ell} + \sum_m E^{km}\sum_s \Omega_{rs}\tilde{w}_s^m\right] \tag{138}$$

where $D^{k\ell}$ and E^{km} are known quantities, expressed in terms of the matrices Q, T, R, and we have introduced the operator

$$\Omega_{rs} = \frac{1}{N}\sum_i F_{ir}F_{is} \tag{139}$$

which has diagonal elements equal 1, and off diagonal elements of order $1/\sqrt{N}$.

In particular we can use this evolution to study the evolution of R:

$$(R^{km})^{\mu+1} - (R^{km})^{\mu} = -\langle uf(u)\rangle\frac{\eta}{N}\left[\sum_{\ell} D^{k\ell}\frac{1}{D}\sum_{rs}\tilde{w}_r^m\Omega_{rs}S_s^{\ell} + \sum_m E^{km}\frac{1}{D}\sum_{rs}\tilde{w}_s^r\Omega_{rs}\tilde{w}_s^m\right] \tag{140}$$

The point of this analysis is to show that the time evolution of S_r^k involves $(\Omega S)_r^{\ell}$. Therefore to know the evolution of S we need the one of ΩS. This is not inocuous because, in order to have dynamical evolution equations with only "up" indices, we need to contract it. The evolution of R^{km}, which is proportional to the scalar product (in the R-dimensional manifold space) of S^k and $\tilde{w}^m$, is thus given by the scalar product of ΩS^k and $\tilde{w}^m$.

It is not difficult to see that the evolution of ΩS will require knowing $\Omega^2 S$ etc. So we have an infinite hierarchy of coupled equations. However these can be closed by changing basis for S.

C. Replica analysis for full-batch learning for the hidden manifold mode: a sketch

We sketch here how the results in [18, 19] can be re-derived and generalized as an application of the GET used in connection with the replica method from statistical physics. This is actually a general approach to study the full batch learning problem. We shall not develop the full approach here, leaving it for future work, but we show the general strategy.

Consider the HMM trained on P patterns

$$X_{\mu i} = f\left(\sum_{r=1}^{D} C_{\mu r}F_{ri}/\sqrt{D}\right) \tag{141}$$

with output labels given by

$$\tilde{y}_{\mu} = \sum_{m=1}^{M}\tilde{v}_m\tilde{g}\left(\sum_{r=1}^{D} C_{\mu r}\tilde{w}_r^m/\sqrt{D}\right). \tag{142}$$

Our neural network is a two-layer committee machine, the output corresponding to input pattern μ is given by:

$$\tilde{y}_{\mu} = \sum_{k=1}^{K}\tilde{g}\left(\sum_{i=1}^{N} X_{\mu i}w_i^k/\sqrt{D}\right). \tag{143}$$

The weights w_i^k are learnt during the training, while the coefficients $C_{\mu r}$, describing the weight of pattern μ on feature r, are i.i.d. Gaussian random variables of mean 0 and variance 1. The other matrices, $F, \widetilde{w}$ are fixed a priori, and they are supposed to be balanced in the sense that they satisfy the hypothesis of the GET. We introduce a loss function, or in physical terms a learning energy, given by

$$\sum_{\mu=1}^{P} \mathcal{E}(\widetilde{y}_\mu - y_\mu). \tag{144}$$

In our numerical studies for instance, we used a least square function $\mathcal{E}(y) = y^2$.

Using Gardner's approach [69], we suppose that the weights w_i^k have a prior distribution $\prod_{i,k} P_w(w_i^k)$ which is i.i.d. for each i and k, and we estimate the "partition function":

$$Z = \prod_{i=1}^{N} \prod_{k=1}^{K} \left[dw_i^k P_w(w_i^k) \right] e^{-\beta \sum_\mu \mathcal{E}(\widetilde{y}_\mu - y_\mu)} \ . \tag{145}$$

Sending $\beta \to \infty$ will allow to minimise the energy $\sum_\mu \mathcal{E}(\widetilde{y}_\mu, y_\mu)$. The optimal training energy is thus found as $-d \log Z / d\beta$ estimated in the $\beta \to \infty$ limit.

We would like to evaluate $\mathbb{E}_C \log Z$, which is the average of $\log Z$ over the Gaussian distribution of the coefficients $C_{\mu r}$. We use the replica methods which consists in computing $\mathbb{E}_C Z^n$, in the limit $n \to 0$. We can write Z^n as the partition function of replicated weights w_i^{ka}, where $a = 1, \dot{,} n$ is a replica index.

$$Z^n = \prod_{i=1}^{N} \prod_{k=1}^{K} \prod_{a=1}^{N} \left[dw_i^{ka} P_w(w_i^{ka}) \right] e^{-\beta \sum_{\mu,a} \mathcal{E}(\widetilde{y}_\mu - y_\mu^a)}. \tag{146}$$

where

$$y_\mu^a = \sum_{k=1}^{K} \tilde{g} \left(\sum_{i=1}^{N} X_{\mu i} w_i^{ka} / \sqrt{D} \right). \tag{147}$$

When we compute the average of Z^n with respect to the distribution of $C_{\mu r}$, we notice first that the average for different values of μ factorises. Furthermore, one can introduce the auxiliary variables

$$u_{i\mu} = \frac{1}{\sqrt{D}} \sum_{r=1}^{D} F_{ir} C_{\mu r} \tag{148}$$

$$\lambda_\mu^{ka} = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^{ka} f[u_{i\mu}] \tag{149}$$

$$\nu_\mu^m = \frac{1}{\sqrt{D}} \sum_{r=1}^{D} \widetilde{w}_r^m C_{r\mu} \tag{150}$$

so that the Boltzmann weight is given by

$$\exp \left[-\beta \sum_{\mu,a} \mathcal{E}(\sum_m \widetilde{g}(\nu_\mu^m) - \sum_k g(\lambda_\mu^{ka})) \right] \tag{151}$$

Therefore this weight depends on w_i^a only through the quantities λ_μ^{ka} and ν_μ^m. This makes it easy to do the average over $C_{r\mu}$: the GET tells us that, for a given μ the Kn variables λ_μ^{ka} and the Mn variables ν_μ^m are joint Gaussian variables, with mean 0 (assuming for simplicity that $\langle f(u) \rangle = 0$) and covariance given by the GET. Denoting by $\mathbb{E}_{\lambda;\nu}$ the expectation with respect to this joint Gaussian distribution, we get:

$$\mathbb{E}_C Z^n = \prod_{ika} \left[dw_i^{ka} P_w(w_i^{ka}) \right] \prod_\mu \mathbb{E}_{\lambda;\nu} \exp \left[-\beta \sum_{\mu,a} \mathbb{E}(\sum_m \widetilde{g}(\nu_\mu^m) - \sum_k g(\lambda_\mu^{ka})) \right]. \tag{152}$$

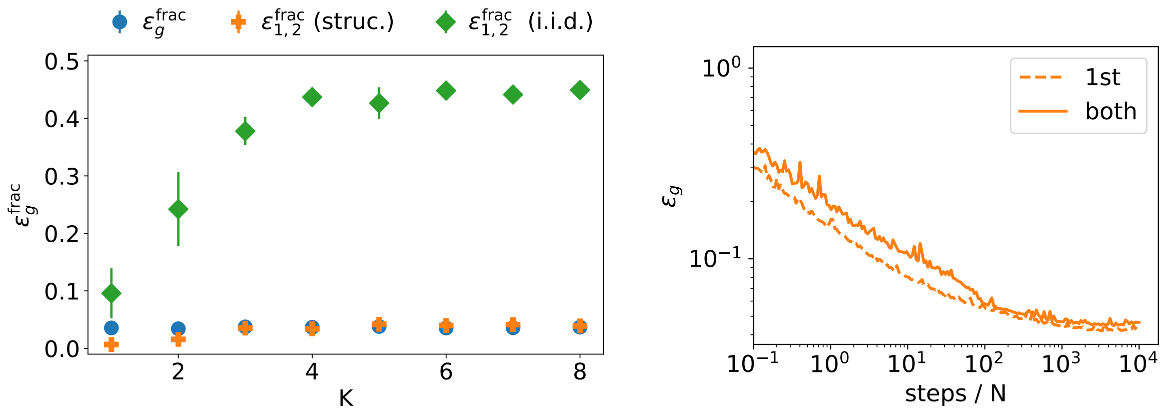


Figure 10: Experimental results for neural networks trained on Fashion MNIST. *(Left)* Same plot as Fig. 1, but this time we train the networks on the Fashion MNIST data set. *(Right)* Generalisation dynamics of a two-layer network with $K = 3$ hidden units trained on the FMNIST odd-even discrimination task described in Sec. D.1. Similar to Fig. 2, we see that the plateaus characteristic of the vanilla teacher-student setup are missing here. In both plots, $g(x) = \operatorname{erf}(x/\sqrt{2}), N = 784, P^* = 76N, K = 3, \eta = 0.2$

From this expression, using standard methods from replica theory [70], one can obtain the quenched average $\mathbb{E}_C \log Z$, and therefore compute the training error and test error. This whole study will be the subject of an upcoming paper.

D. Additional numerical experiments

D.1. Experiments on Fashion MNIST

To verify that our experimental results are not specific to the MNIST data set, we replicated our experiments using the Fashion MNIST data set [71], which consists of 60 000 images of fashion products, divided into ten classes. We split these classes into two groups: (T-shirt/top, Pullover, Coat, Shirt, Bag) vs (Trouser, Dress, Sandal, Sneaker, Ankle boot) and trained networks to discriminate between these two classes. As we show in Fig. 10, the behaviour of independent students shows the same behaviour as it did on MNIST: While the generalisation error improves slightly as we increase the number of parameters in the network, two independent students disagree on Fahsion MNIST images only at a rate that is comparable to their generalisation error. However, they learn globally different functions, as is evidenced by their large disagreement on the classification of i.i.d. Gaussian inputs.

D.2. The existence of plateaus is not explained by the asymptotic generalisation error

We have demonstrated on the right of Fig. 2 that neural networks trained on data drawn from the hidden manifold model (HMF) introduced here do not show the plateau phenomenon, where the generalisation error stays stationary after an initial exponential decay, before dropping again. Upon closer inspection, one might think that this is due to the fact that the student trained on data from the HMF asymptotes at a higher generalisation error than the student trained in the vanilla teacher-student setup. This is not the case, as we demonstrate in Fig. 11: we observe no plateau in a sigmoidal network trained on data from the HMF even that network asymptotes at a generalisation error that is, within fluctuations, the same as the generalisation error of a network of the same sized trained in the vanilla teacher-student setup and which shows a plateau.

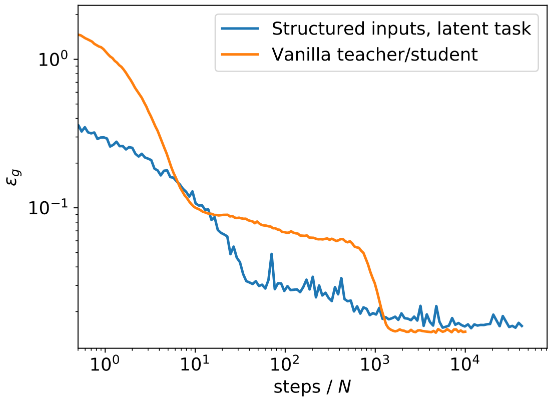


Figure 11: The plateau in the vanilla teacher-student setup can have larger generalisation error than the asymptotic error in a latent task on structured inputs. Generalisation dynamics of a sigmoidal network where we train only the first layer on (i) structured inputs $\boldsymbol{X} = \max(0, \mathbf{CF})$ with latent labels $\tilde{y}_i$ (7) (*blue*, $D = 10$) and (ii) the vanilla teacher-student setup (Sec. 2.1, *orange*). In both cases, $M = 5, K = 6, \eta = 0.2, P = 76N, v^*_m = 1$.

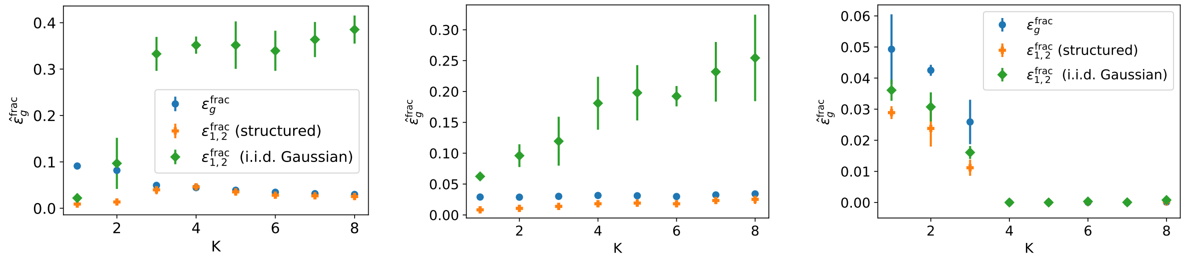


Figure 12: Measuring early stopping errors does not affect the phenomenology of latent and teacher tasks. *(Left)* Performance of independent sigmoidal students on the MNIST task as evaluated by the early-stopping generalisation error. (*Center* and *Right*) We reproduce Fig. 3 of the main text, but this time we plot the early-stopping generalisation error $\hat{\epsilon}_g^{\text{frac}}$ for two networks trained independently on a binary classification task with structured inputs (6) and latent labels $\widetilde{y}_i^*$ (Eq. 7, $M = 1$, *Center*) and teacher labels y_i^* (4) ($M = 4$) *(Left)*. In both plots, $f(x) = \text{sgn}(x), g(x) = \text{erf}\left(x/\sqrt{2}\right), D = 10, \eta = 0.2$.

D.3. Early-stopping yields qualitatively similar results

In Fig. 12, we reproduce Fig. 3, where we compare the performance of independent neural networks trained on the MNIST task *(Left)*, or trained on structured inputs with a latent task *(Center)* and a teacher task *(Right)*, respectively. This time, we the early-stopping generalisation error $\hat{\epsilon}_g^{\text{frac}}$ rather than the asymptotic value at the end of training. We define $\hat{\epsilon}_g^{\text{frac}}$ as the minimum of ϵ_g^{frac} during the whole of training. Clearly, the qualitative result of Sec. 3.2 is unchanged: although we use structured inputs (6) in both cases, independent students will learn different functions which agree on those inputs only when they are trained on a latent task (7) *(Center)*, but not when trained on a vanilla teacher task (4) *(Right)*. Thus structured inputs and latent tasks are sufficient to reproduce the behaviour observed when training on the MNIST task.

D.4. Dynamics with a large number of features $D \sim N$

Here we investigate the behaviour of networks trained on data from the hidden manifold model when the number of feature vectors D is on the same order as the input dimension N. We call

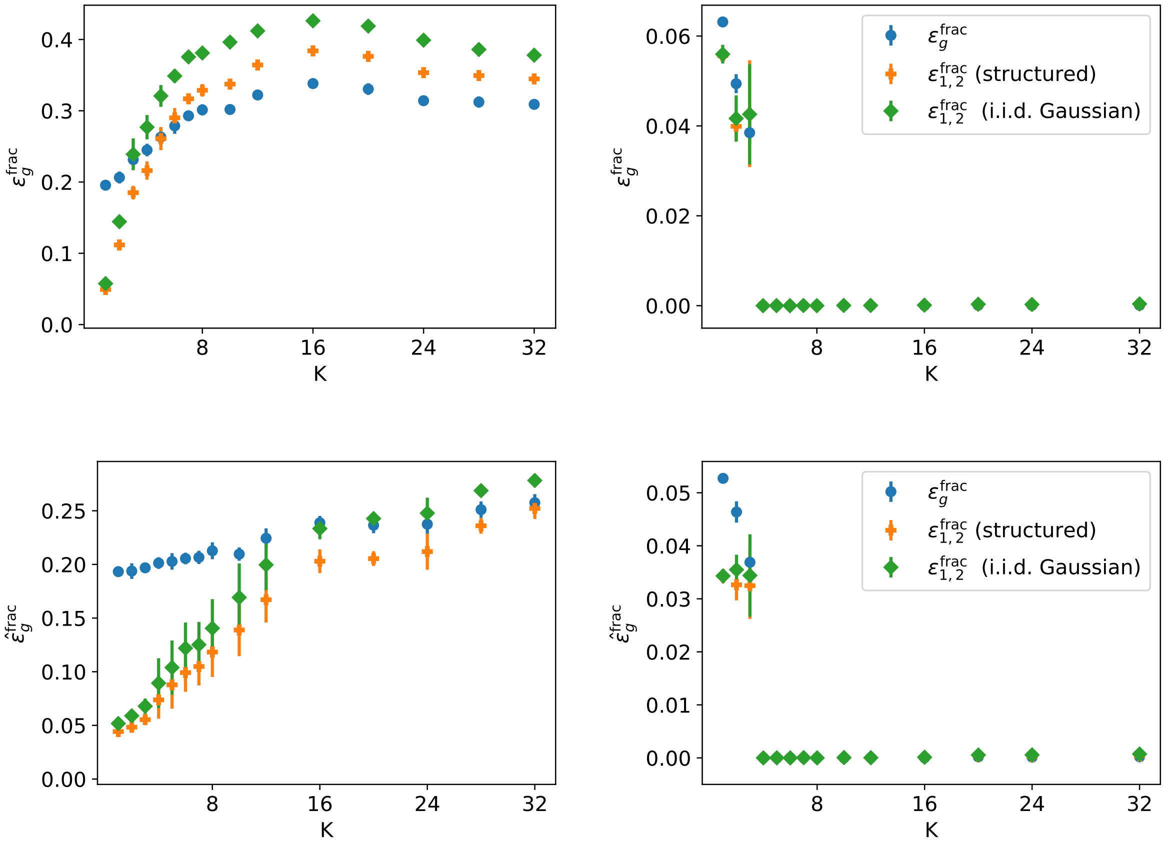


Figure 13: Performance of independent networks trained on a latent task with inputs in many latent directions $D = N/2$. *(Top Left)* For two networks trained independently on a binary classification task with structured inputs (6) and latent labels $\tilde{y}_i^*$ (Eq. 7, $M = 1$), we plot the final fractional test error, $\epsilon_g^{\mathrm{frac}}$ (blue dots). We also plot $\epsilon_{1,2}^{\mathrm{frac}}$ (5), the fraction of Gaussian i.i.d. inputs and structured inputs the networks classify differently after training (green diamonds and orange crosses, resp.). *(Top Right)* Same experiment, but with structured inputs and *teacher labels* y_i^* (4) ($M = 4$). *(Bottom row)* Same plots as in the top row, but this time for the early-stopping error $\hat{\epsilon}^{\mathrm{frac}}$ (see Sec. D.3). In all plots, $f(x) = \mathrm{sgn}(x), g(x) = \mathrm{erf}\left(x/\sqrt{2}\right), N = 500, D = 250, \eta = 0.2$.

this the regime of extensive D. It is a different regime from MNIST, where experimental studies consistently find that inputs lie on a low-dimensional manifold of dimension $D \sim 14$, which is much smaller than the input dimension $N = 784$ [8–10].

We show the results of our numerical experiments with $N = 500, D = 250$ in Fig. 13, where we reproduce Fig. 3 for the asymptotic (top row) and the early-stopping (bottow row) generalisation error. The behaviour of networks trained on a teacher task with structured inputs (right column) is unchanged w.r.t. to the case with $D = 10$. For the latent task, increasing the number of hidden units, however, *increases* the generalisation error, indicating severe over-fitting, which is only partly mitigated by early stopping. The generalisation error on this task is generally much higher than in the low-D regime and clearly, increasing the width of the network is not the right way to learn a latent task; instead, it would be intriguing to analyse the performance of deeper networks on this task where finding a good intermediate representation for inputs is key. This is an intriguing avenue for future research.

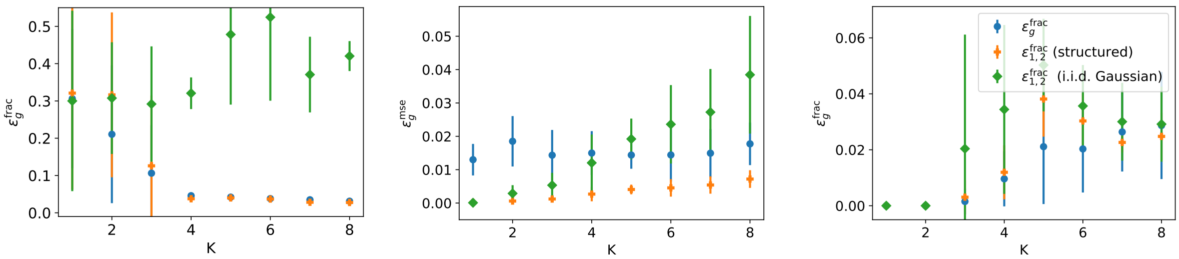


Figure 14: Behaviour of independent students with ReLU activation functions. *(Left)* Asymptotic generalisation error of independent students with ReLU activation function $g(x) = \max(0, x)$ on the MNIST task. (*Center* and *Right*) We reproduce Fig. 3 of the main text for two networks with ReLU activation trained independently on a binary classification task with structured inputs (6) and latent labels $\widetilde{y}_i^*$ (Eq. 7, $M = 1$) *(Center)* and teacher labels y_i^* (4) ($M = 4$ *Right*). In both plots, $f(x) = \text{sgn}(x), g(x) = \max(0, x), D = 10, \eta = 0.1$.

D.5. Independent students with ReLU activation function

We also verified that the behaviour of independent networks we observed on MNIST with sigmoidal students persists when training networks with ReLU activation function and that the hidden manifold model is able to reproduce it for these networks. We show the results of our numerical experiments in Fig. 14. To that end, we trained both layers of a network $\phi(\boldsymbol{x}, \boldsymbol{\theta})$ with $g(x) = \max(x, 0)$ starting from small initial conditions, where we draw the weights component-wise i.i.d. from a normal distribution with variance 10^{-6}.

We see that the generalisation error of ReLU networks on the MNIST task (*Left* of Fig. 14) decreases with increasing number of hidden units, while the generalisation error on MNIST inputs of the two independent students with respect to each other is comparable or less than the generalisation error of each individual network on the MNIST task.

On structured inputs with a teacher task (*Right* of Fig. 14), where labels were generated by a teacher with $M = 4$ hidden units, the student recovers the teacher such that its generalisation error is less than 10^{-3} for $K > 4$, and both independent students learn the same function, as evidenced by their generalisation errors with respect to each other. This is the same behaviour that we see in Fig. 3 for sigmoidal networks. The finite value of the generalisation error for $K = M = 4$ is due to two out of ten runs taking a very long time to converge, longer than our simulation lasted for. Finally, we see that for a latent task on structured inputs, the generalisation error of the two networks with respect to each other increases beyond the generalisation error on structured inputs of each of them, as we observed on MNIST. Thus we have recovered the phenomenology that we described for sigmoidal networks in ReLU networks, too.