PAPER

Statistical mechanics of deep learning

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: Les Houches 2022

ÉCOLE DE PHYSIQUE DES HOUCHES

Statistical mechanics of deep learning

Freya Behrens[1], **Nischal Mainali**[2], **Chiara Marullo**[3], **Sebastian Lee**[4], **Ben Sorscher**[5] **and Haim Sompolinsky**[2,6,*]

[1] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[2] The Hebrew University, Jerusalem, Israel
[3] Sapienza Università di Roma, Rome, Italy
[4] Imperial College London, London, United Kingdom
[5] Stanford University, Stanford, CA, United States of America
[6] Center for Brain Science, Harvard University, Cambridge, MA, United States of America
E-mail: haim@fiz.huji.ac.il

Received 30 November 2023
Accepted for publication 3 April 2024
Published 30 October 2024

Online at stacks.iop.org/JSTAT/2024/104007
https://doi.org/10.1088/1742-5468/ad3a60

Keywords: computational neuroscience, deep learning

J. Stat. Mech. (2024) 104007

Contents

*Author to whom any correspondence should be addressed.

J. Stat. Mech. (2024) 104007

1. Introduction

Goals of deep learning theory: what should it seek to explain?

1. Memory capacity: how much random data can a network store?
2. Expressivity: what classes of functions can a given architecture express or approximate?
3. Generalization, regularization, inductive biases and overparameterization.
4. Learning dynamics.
5. Representations: understanding latent or hidden representations that emerge during or after learning.
6. Relation to the brain.

The first three lectures will fit mostly into goal 3 with some elements of goal 4. The final two lectures will cover work in goals 1, 5, and 6.

Statistical mechanics of deep learning

What can the particular angle of statistical mechanics contribute to the above goals of deep learning theory? There are several themes of statistical mechanics that are relevant:

1. Thermodynamic limits: what parameters or data need to be taken to infinity, and at what rate, in order for statistical mechanics to say something interesting about the system?
2. Order parameters: macroscopic properties or global functions of the entire degrees of freedom of the network that control or explain the performance.
3. Typicality, universality, fluctuations.
4. Scales, scaling relations, power laws.
5. Critical points/phase transitions.
6. Tools: coarse graining, renormalization group, mean-field theories, perturbation theory, numerics, etc.

J. Stat. Mech. (2024) 104007

Challenges: why is the problem so hard?

1. Complex data structures.
2. Complex network architectures (heterogeneity, strong long-range interactions).
3. There is little widely agreed upon and reproducible phenomenology; empirical data are fine-tuned and can be artifacts of engineered solutions.

Early work on statistical mechanics of learning focused on simple data structures (Gaussian inputs, random labels, etc) and simple architectures (homogeneous recurrent networks or single-layer networks). Modern statistical mechanics aims to be more relevant to machine learning and neuroscience by addressing real world datasets and architectures (long short-term memory (LSTMs), transformers, etc).

2. Lecture 1: statistical mechanics of deep learning and the infinite width limit

2.1. Bayesian learning and statistical mechanics of deep learning

This chapter is largely based on work in *Statistical Mechanics of Deep Linear Neural Networks: The Backpropagating Kernel Renormalization* by Li and Sompolinsky [1].

2.1.1. Bayesian learning and inference. **Bayesian Learning:** we consider a multi-layer architecture with a single linear output:

$$f(x,\Theta) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i x_i^L(x,W), \tag{1}$$

where $a \in R^N$is the readout weight vector, N denotes the size of the Lth layer, $x \in R^{N_0}$ is the input vector, and x_i^L is the activation of the $i = 1,\ldots,N$ neuron in the top (Lth) layer in response to the input vector x, which depends of course on all the hidden-layer weights W. Specifically, the activation of neuron i in layer L is

$$x_i^L(x,W) = \phi\left(N^{-1/2} w^{iL} \cdot x^{L-1}(x)\right), \tag{2}$$

where ϕ is the activation function. Finally, we denote by $\Theta = (a,W)$, the vector of all weights.

Random network scaling: (also known as the 'lazy' regime). By normalizing the pre-activations $w^{iL} \cdot x^{L-1}(x)$ with $N^{-1/2}$ we anticipate a learning process such that even after learning, the correlations between w^{iL} and $x^{L-1}(x)$ are weak and sum up to a dot product of magnitude only $O(\sqrt{N})$ (the same order of magnitude as w, completely random).

Mean-field scaling: (also known as 'rich' or 'feature-learning' regime). For instance, an alternative normalization would be N^{-1}, reminiscent of the standard mean field in statistical mechanics. In statistical mechanics, the choice of scaling is typically dictated by the **pattern** of the interactions, in our case w. Since w is being learned, our situation

J. Stat. Mech. (2024) 104007

is more complex; the choice of normalization itself has an effect on the statistics of w. To generate a pre-activation of $O(1)$, the learning process would need to modify the statistics of w more drastically in the mean-field scaling than in the random network scaling. One effect of this scaling is that it permits non-trivial representations to be learned in the weights.

Learning: consider supervised learning that minimizes the following mean-squared error loss function, also known as the likelihood,

$$L\left(\Theta|\mathcal{D}\right)=\frac{1}{2}\sum_{\mu=1}^{P}\left(f\left(x^{\mu},\Theta\right)-y^{\mu}\right)^{2} \tag{3}$$

where the training data $\mathcal{D}=\{x^{\mu},y^{\mu}\}_{\mu=1}^{P}$ are the P pairs of input and target labels. We assume that the learning process is Bayesian and, given the data distribution $\mathcal{D}$, results in a probability distribution on the weights of the form

$$P\left(\Theta|\mathcal{D}\right)=Z^{-1}\exp\left(-\frac{\beta}{2}\sum_{\mu=1}^{P}\left[\frac{1}{\sqrt{N}}\sum_{i=1}^{N}a_{i}x_{i}^{L}\left(x^{\mu},W\right)-y^{\mu}\right]^{2}-\frac{1}{2\sigma^{2}}\|\Theta\|^{2}\right), \tag{4}$$

with the normalizing constant

$$Z\left(\mathcal{D}\right)=\int\mathrm{d}\Theta\exp\left(-\frac{\beta}{2}\sum_{\mu=1}^{P}\left[\frac{1}{\sqrt{N}}\sum_{i=1}^{N}a_{i}x_{i}^{L}\left(x^{\mu},W\right)-y^{\mu}\right]^{2}-\frac{1}{2\sigma^{2}}\|\Theta\|^{2}\right). \tag{5}$$

The last term in the exponent is an L_2 regularizer biasing the probability in weight space to small norm weights. We will call σ the **regularizer strength**. We can thus consider learning as modifying the probability in weight space,

$$P_{\text{posterior}}\left(\Theta\right)\propto P_{\text{prior}}\left(\Theta\right)\exp\left\{-\beta L\left(\Theta|\mathcal{D}\right)\right\}, \tag{6}$$

where the regularization term acts like a prior, i.e.

$$P_{\text{prior}}\left(\Theta\right)\propto\exp\left\{-\frac{1}{2\sigma^{2}}\|\Theta\|^{2}\right\}. \tag{7}$$

This treatment of the learning process as shaping the posterior distribution in weight space of deep networks with the training data (and accompanying loss) is called **Bayesian Learning or Bayesian Neural Networks.** In general, we are interested in using the trained network to make an inference on a 'test' input vector (the training input is a special case). Within the Bayesian framework, this amounts to evaluating the posterior probability of the predictor. In most cases, we are content with the following first two moments, the **predictor mean**:

$$\left\langle f\left(x\right)\right\rangle=\int\mathrm{d}\Theta P\left(\Theta\right)f\left(x,\Theta\right), \tag{8}$$

J. Stat. Mech. (2024) 104007

and the **predictor variance**:

$$\langle(\delta f(x))^2\rangle = \int \mathrm{d}\Theta P(\Theta)(\delta f(x,\Theta))^2, \tag{9}$$

where the averages are w.r.t. the parameters Θ. The expression $\delta f(x) = f(x) - \langle f(x)\rangle$ is the centered f. The variance is a measure of the 'certainty' in the network prediction.

Generalization error: the per-sample generalization error $\varepsilon_g(x)$ is defined as

$$\varepsilon_g(x) = \langle(f(x,\Theta) - y(x))^2\rangle, \tag{10}$$

$$= \text{bias} + \text{variance}; \tag{11}$$

where

$$\text{bias} = (\langle f(x,\Theta)\rangle - y(x))^2, \tag{12}$$

$$\text{variance} = \langle(\delta f(x,\Theta))^2\rangle, \tag{13}$$

and $y(x)$ is the target (unknown) rule for $\mathcal{D}$. Thus, the generalization error can be decomposed into a bias and variance contributions, like in many other models of inference. However, it is important to note that $\varepsilon_g(x)$ is both x-dependent and training-data dependent, and that the variance originates solely from the variability in the estimates of the weights. Thus, in conventional deterministic supervised learning this variance is absent. To compute the global generalization error one needs to calculate

$$\varepsilon_g = \int \mathrm{d}x P(x) \int \Pi_\mu \mathrm{d}x_\mu P(x_\mu)\varepsilon_g(x|\mathcal{D}). \tag{14}$$

This averaging process is typically a source of additional variance, but this will not be discussed in detail in these lectures. See [2] for a more detailed related discussion.

2.1.2. Generating functional. It is convenient to evaluate the statistics of the predictor by turning the partition function into a generating functional. To do this, we formally add an external source term (coupled to the predictor variable $-il$) to the action (the exponent of the posterior):

$$-il\frac{1}{\sqrt{N}}\sum_{i=1}^{N} a_i x_i^L(W,x). \tag{15}$$

This makes the partition function a function of this source:

$$Z(l) = \int \mathrm{d}\Theta \exp\left(-\beta L(\Theta|D) - \frac{1}{2\sigma^2}\|\Theta\|^2 - il\frac{1}{\sqrt{N}}\sum_{i=1}^{N} a_i x_i^L(W,x)\right), \tag{16}$$

such that the statistics of f is given by differentiating the log of this generating functional. In particular, this generates

$$\langle f(x)\rangle = i\frac{\partial}{\partial l}\log Z|_{l=0}, \tag{17}$$

J. Stat. Mech. (2024) 104007

and

$$\langle (\delta f(x))^2 \rangle = -\frac{\partial^2}{\partial l^2} \log Z|_{l=0}. \tag{18}$$

2.1.3. Bayesian learning and statistical mechanics. There is a direct parallel between the above Bayesian framework and the statistical mechanics of learning. In the latter, we consider the loss as forming an energy function on the weights,

$$E(\Theta) = L(\Theta|D) + \frac{T}{2\sigma^2}\|\Theta\|^2, \tag{19}$$

with parameterization $T = \beta^{-1}$. Here, learning is viewed as a stochastic process, the result of which is an equilibrium Gibbs distribution at temperature T, i.e.

$$P(\Theta) = Z^{-1} \exp -\beta E(\Theta). \tag{20}$$

Note that for notational convenience, the L_2 regularization term has been absorbed by the energy function. To be consistent with the convention above that the prior contribution is independent of β, the regularization term in E is scaled by a factor of T.

2.1.4. Langevin learning. The simplest stochastic learning process that leads over a long time to the above Gibbs equilibrium distribution is the continuous time stochastic differential equation known as the Langevin equation, defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}\Theta = -\nabla_\Theta E(\Theta) + \sqrt{2T}\eta(t), \tag{21}$$

where η is a white noise process with the same dimension as Θ. Averaging over trajectories of $\eta(t)$, this process induces a time-dependent probability on Θ, which converges to

$$\lim_{t\to\infty} P_t(\Theta) = Z^{-1} \exp -\beta E(\Theta). \tag{22}$$

Langevin dynamics connects the statistical mechanics framework to conventional gradient descent learning.

2.1.5. Over-parametrization and the zero-temperature limit. The above framework holds for arbitrary temperatures $T > 0$. A particularly interesting limit is $T \to 0$. In this limit, the posterior probability is centered at the parameters Θ that minimize the loss function. Note that because the L_2 regularizer term in E was multiplied by T, the posterior is dominated by the loss term at low temperature (as is clear from (4)).

In many areas of both physics and learning in traditional neural networks, the vector Θ that minimizes the loss is unique or consists of a small number of degenerate points. In contrast, deep networks are typically much larger than that which is minimally required to satisfy all the constraints of the training data. This regime is known as

J. Stat. Mech. (2024) 104007

the over-parameterized regime, where there is a huge subspace of the parameters for which

$$L(\Theta|\mathcal{D}) = 0. \tag{23}$$

Consequently, the posterior distribution is concentrated in this subspace, i.e. the probability of Θ for which $L(\Theta|\mathcal{D}) > 0$ is zero. Within this subspace of zero loss, the distribution is **not** uniform. In fact, the solution subspace is typically unbounded, such that a uniform sampling is undefined. In our setting, this problem is resolved by the addition of the Gaussian regularizer, which biases the distribution to weights with small norms. Thus, while the regularizer term does not compete with the loss term in the low-temperature T limit, it does bias the distribution in the zero-loss subspace. This is different from the usual ridge regression, where the regularizer term is of the same order as the loss term, and there is a competition between the two.

Relation to gradient descent: given a specific choice of initial Θ, gradient descent dynamics converge to a unique solution. It is interesting to ask if there is a relation between these dynamics and the statistical mechanics predictions. In general, since the Langevin dynamics is ergodic, the Gibbs distribution has no memory of the initial Θ (assuming a sufficiently small T to ensure exploration of the solution space). Thus, the Gaussian prior is not strictly speaking a prior on the initial weights but rather a regularizer for the final weight distribution. However, the statistical mechanics results are at least qualitatively similar to standard gradient-based learning if the initial weights are sampled from a Gaussian distribution with width σ. Thus, one can think of σ as analogous to the strength of the initial weights in gradient descent dynamics. However, it is important to keep in mind that conceptually, the statistical mechanics results are not about the dynamics or the initialization but about the equilibrium statistics of the weights.

2.1.6. Hyperparameters and the thermodynamic limit. Before moving forward with calculations it is worth highlighting the important hyperparameters aside from the training data: the size of the layers N, the number of layers L, the regularizer strength σ, the temperature T, and the number of data samples P. Solving the problem analytically for arbitrary values of these parameters is generally impossible, even in much simpler statistical mechanical systems. Hence, we will proceed in a similar manner to many other statistical mechanics calculations and focus on the infinite size limit in an appropriately defined thermodynamic limit. For deep networks we define the thermodynamic limit as

$$P, N \to \infty, \tag{24}$$

with a constant ratio

$$\alpha = \frac{P}{N} = O(1). \tag{25}$$

The rest of the hyperparameters, such as the regularizer strength σ, and the number of layers L remain finite. In general, to explore the dependence on the ratio α we will vary

J. Stat. Mech. (2024) 104007

$N = \alpha P$, keeping P large and fixed. We choose this procedure as we want to explore the dependence on variation in the network width N for fixed training data (including input dimension and data size). Details of these issues will be elucidated later.

2.2. W-dependent statistics of the readout weights

Because we defined the network output as linear in the readout weights a and used the mean-squared error (MSE) loss, the statistics of a conditioned on the hidden weight W is Gaussian. One obtains straightforwardly that

$$\langle a|W\rangle = \sigma^2\beta\left[I + \sigma^2\beta N^{-1}X^L X^{LT}\right]^{-1} N^{-1/2}X^L Y, \tag{26}$$

where X^L is the $N \times P$ training input matrix with

$$\left(X^L\right)_{i,\mu} = x_i^L\left(x^\mu\right). \tag{27}$$

The conditional variance is

$$\langle \delta a \delta a^T|W\rangle = \sigma^2\left[I + \sigma^2\beta N^{-1}X^L X^{LT}\right]^{-1}. \tag{28}$$

We can write

$$\left[I + \sigma^2\beta N^{-1}X^L X^{LT}\right]^{-1} = I - \sigma^2 N^{-1}X^L\tilde{K}^{-1}X^{LT}, \tag{29}$$

where we have introduced the $P \times P$ **kernel matrices**

$$K = \frac{\sigma^2}{N}\left(X^L\right)^T X^L. \tag{30}$$

The elements of K are normalized dot products of pairs of top layer activations corresponding to two training input vectors, scaled by σ^2. We further define

$$\tilde{K} = K + TI. \tag{31}$$

Thus, we can rewrite the moments of a as

$$\langle a|W\rangle = \sigma^2 N^{-1/2}X^L\tilde{K}^{-1}Y, \tag{32}$$

and

$$\langle \delta a \delta a^T|W\rangle = \sigma^2\left[I - \sigma^2 N^{-1}X^L\tilde{K}^{-1}X^{LT}\right]. \tag{33}$$

In particular

$$\langle \delta a^T \delta a\rangle = \sigma^2\left(N - \mathrm{Tr}\tilde{K}^{-1}K\right), \tag{34}$$

$$\langle a^T a\rangle = \sigma^2\left(N - \mathrm{Tr}\tilde{K}^{-1}K\right) + \sigma^2 Y^T\tilde{K}_L^{-2}KY. \tag{35}$$

J. Stat. Mech. (2024) 104007

Using the statistics of a we can substitute to compute the statistics of the predictor:

$$\langle f(x)\rangle = k_L(x)^T \tilde{K}_L^{-1} Y, \tag{36}$$

and

$$\langle (\delta f(x))^2\rangle = K_L(x,x) - k_L(x)^T \tilde{K}_L^{-1} k_L(x), \tag{37}$$

where $k_L(x)$ is a P-dimensional vector whose components are $K_L(x,x^\mu)$, i.e.

$$k_L(x) = K_L(x,x^\mu) = \sigma^2 N^{-1} X^{LT} x^L(x), \tag{38}$$

and

$$K_L(x,x) = \sigma^2 N^{-1} x^{LT}(x)\, x^L(x). \tag{39}$$

So far, our equations hold for finite temperature T. We will now take the zero-temperature limit.

2.2.1. Zero-temperature limit. **1. Wide regime:** $\alpha < 1$, i.e. the network width L is larger than the number of parameters P. In this case we have

$$\langle a^T a\rangle = \sigma^2 N(1-\alpha) + \sigma^2 Y^T K_L^{-1} Y, \tag{40}$$

$$\langle f(x)\rangle = k_L(x)^T K_L^{-1} Y, \tag{41}$$

and

$$\langle (\delta f(x))^2\rangle = K_L(x,x) - k_L(x)^T K_L^{-1} k_L(x). \tag{42}$$

Note that evaluating the predictor on the training data, yields for equation (41)

$$\langle f(X)\rangle = K_L K_L^{-1} Y = Y, \tag{43}$$

$$\langle (\delta f(X))^2\rangle = K_L - K_L K_L^{-1} K_L = 0 \tag{44}$$

as expected.

2. Narrow Regime: $\alpha > 1$, i.e. few parameters but many data samples.

There are no fluctuations in a because the solution (which, for generic untrained W, will have non-zero loss), is unique:

$$\langle a^T a\rangle = \langle a^T\rangle\langle a\rangle = \sigma^2 Y^T K_L^+ Y, \tag{45}$$

where K_L^+ is the pseudoinverse of K. Note that in the expression for the predictor, the zero-temperature limit is smooth since K^{-1} is evaluated in the subspace spanned by the N, $P-$dimensional training vectors. It is worth emphasizing that all the results above

J. Stat. Mech. (2024) 104007

are conditioned on the value of W. We now turn our attention to W. We do this first by integrating out a and obtaining the marginal posterior on W.

2.3. Integrating out *a*

Since the energy function is quadratic in a it is straightforward to integrate it out, yielding

$$P_L(W) = Z^{-1} \exp\left(-\frac{\|W\|^2}{2\sigma^2} - \frac{1}{2} Y^T \tilde{K}_L^{-1} Y - \frac{1}{2} \log\det \tilde{K}_L\right). \tag{46}$$

Recall that $\tilde{K} = K + TI$. Note that K_L depends on W. We can interpret this as inducing a Hamiltonian on w, i.e. $P_L(W) \propto \exp^{-H_L(W)}$, where

$$H_L(W) = \frac{1}{2} Y^T \tilde{K}_L^{-1} Y + \frac{1}{2} \log\det \tilde{K}_L + \frac{\|W\|^2}{2\sigma^2}. \tag{47}$$

Only the first term depends on Y and can be seen as the norm of a. The second term is the entropic term, while the last term is simply the prior. To evaluate the relevant quantities, we need to average the above results over W. For instance,

$$\langle a^T a \rangle = \sigma^2 \left[N - Tr\left\langle K_L \tilde{K}_L^{-1} \right\rangle + Y^T \left\langle \tilde{K}_L^{-1} K_L \tilde{K}_L^{-1} \right\rangle Y\right]. \tag{48}$$

The predictor statistics on an arbitrary input x are given by

$$\langle f(x) \rangle = \left\langle k_L(x)^T \tilde{K}_L^{-1} \right\rangle Y, \tag{49}$$

and

$$\begin{aligned} \langle (\delta f(x))^2 \rangle = {} & \langle K_L(x,x) \rangle - \left\langle k_L(x)^T \tilde{K}_L^{-1} k_L(x) \right\rangle \\ & + Y^T \left\langle \tilde{K}_L^{-1} k_L(x) k_L^T(x) \tilde{K}_L^{-1} \right\rangle Y - \langle f(x) \rangle^2, \end{aligned} \tag{50}$$

where the last two terms reflect the contribution from the W-dependent mean of f induced by fluctuations in W. Computing these averages using $H_L(W)$ is intractable, except for special limits; one such limit is described below.

2.4. The infinite width limit

A special limit is when we take $N \to \infty$ keeping P fixed; this is known as the Gaussian process limit. This will make some computations feasible that are difficult in the fixed α regime. Since the data dependence contribution to $H_L(W)$ (47) is of order P and the

J. Stat. Mech. (2024) 104007

prior term is of order N^2, the training terms can be treated as negligible perturbations on the Hamiltonian. Hence,

$$\langle F(W)\rangle = Z^{-1}\int \mathrm{d}W F(W)\exp\{-H_L(W)\} \approx Z^{-1}\int \mathrm{d}W F(W)\exp -\frac{\|W\|^2}{2\sigma^2}. \tag{51}$$

Thus, averaging over W is done simply using the Gaussian prior. Quantities $F(W) = f(K(W))$ that depend on W through the kernel $K_L(W)$ obey the properties of self-averaging. Consider

$$K_L(x,y) = \sigma^2 N^{-1}\sum_{i=1}^{N}\phi\left(N^{-1/2}w^{Li}\cdot x^{L-1}(x)\right)\phi\left(N^{-1/2}w^{Li}\cdot x^{L-1}(y)\right). \tag{52}$$

Since w^i's are uncorrelated vectors in the large N limit, we have

$$K_L(x,y) = \sigma^2\left\langle\phi\left(N^{-1/2}w^{L}\cdot x^{L-1}(x)\right)\phi\left(N^{-1/2}w^{L}\cdot x^{L-1}(y)\right)\right\rangle_{w^L} \tag{53}$$

$$= \sigma^2\langle\phi(z)\,\phi(z')\rangle_{z,z'}; \tag{54}$$

where, conditioned on lower layer-weights, z and z' are two correlated Gaussian distributions, with zero averages and variances:

$$\langle z^2\rangle = \sigma^2 N^{-1}x^{L-1}(x)\cdot x^{L-1}(x) = K_{L-1}(x,x), \tag{55}$$

$$\langle z'^2\rangle = \sigma^2 N^{-1}x^{L-1}(y)\cdot x^{L-1}(y) = K_{L-1}(y,y), \tag{56}$$

$$\langle zz'\rangle = \sigma^2 N^{-1}x^{L-1}(x)\cdot x^{L-1}(y) = K_{L-1}(x,y). \tag{57}$$

Thus, we can write

$$\lim_{N\to\infty} K_L(x,y) = \sigma^2 F\left(K_{L-1}(x,x), K_{L-1}(y,y), K_{L-1}(x,y)\right). \tag{58}$$

We can iterate this procedure to derive the fully averaged kernel, which is called the Gaussian process (GP) kernel, K_{GP}. This kernel depends on the nonlinearity ϕ, the layers L, the regularization strength σ^2, the input kernels

$$K_L^{\mathrm{GP}}(x,y) = F_L\left(K_0(x,x), K_0(y,y), K_0(x,y)\right), \tag{59}$$

and the first layer kernel

$$K_0(x,y) = \sigma^2 N_0^{-1}x^T y. \tag{60}$$

In particular

$$\langle f(x)\rangle = \left(k_L^{GP}(x)\right)^T \tilde{K}_L^{GP-1}Y, \tag{61}$$

and

$$\langle(\delta f(x))^2\rangle = K_L^{\mathrm{GP}}(x,x) - \left(k_L^{\mathrm{GP}}(x)\right)^T \tilde{K}_L^{\mathrm{GP}-1}k_L^{\mathrm{GP}}(x). \tag{62}$$

2.5. Limitations of the Gaussian process limit

While the Gaussian process limit is an important building block to a theory of deep neural networks, it has a few key limitations,

1. Unrealistic limit
2. Cannot explore overparametrization effects, width dependence
3. No representation learning
4. Pathological behavior in large L.

Discussion: some of the above limitations can be alleviated if one considers the limit of infinite width (and finite P) under different scaling (e.g. one can observe representation learning in the mean-field scaling discussed above). However, this regime still does not capture the more realistic limit of a large number of samples. Despite the limitations, it is still worth emphasizing that the infinite width limit is a very useful starting point. In many aspects, it can provide important insights into deep learning, especially in the absence of other analytical tools, to probe the finite α limit for more complex architectures. Finally, when considering the finite α limit, there are two types of departures from the GP limit. First, even if W are random Gaussian and only the readout weights a are learned, there is still an effect on the statistical properties of the functions of the kernels, i.e. the kernels are not self-averaging. The second departure is when W does not remain Gaussian but is affected by learning. We will see these effects when we get concrete results.

3. Lecture 2: backpropagating kernel renormalization (BPKR) in deep linear neural networks

3.1. Integrating the top-layer weights

In the previous chapter we derived the marginal over the weights W at zero temperature T as

$$P_L(W) = Z^{-1} \exp\left(-\frac{\|W\|^2}{2\sigma^2} - \frac{1}{2} Y^T \tilde{K}_L^{-1} Y - \frac{1}{2}\log\det \tilde{K}_L\right). \tag{63}$$

It is extremely useful to introduce P auxiliary variables t_μ and write

$$P(W) = Z^{-1} \int \mathrm{d}^P t \exp\left(-\frac{1}{2} t^T \tilde{K}_L t + i t^T Y - \frac{\|W\|^2}{2\sigma^2}\right), \tag{64}$$

where the normalization becomes

$$Z = \int \mathrm{d}W \int \mathrm{d}^P t \exp\left(-\frac{1}{2} t^T \tilde{K}_L t + i t^T Y - \frac{\|W\|^2}{2\sigma^2}\right). \tag{65}$$

https://doi.org/10.1088/1742-5468/ad3a60

J. Stat. Mech. (2024) 104007

Note that by Gaussian integration of t this is equivalent to equation (63). Furthermore, by writing the generating functional as

$$Z(l) = \int \mathrm{d}\Theta \int \Pi_{\mu}^{P} \mathrm{d}t_{\mu} \exp\left(-\frac{1}{2\sigma^2}\Theta^T\Theta + \sum_{\mu=1}^{P} it_{\mu}\left[\frac{1}{\sqrt{N}}\sum_{i=1}^{N} a_i x_i^L(W,x^{\mu}) - y^{\mu}\right] + il\left[\frac{1}{\sqrt{N}}\sum_{i=1}^{N} a_i x_i^L(W,x)\right] + \frac{T}{2}t^T t\right), \tag{66}$$

we observe that the source l conjugate to the predictor can be defined as $l = t_{P+1}$, where $x_{P+1} = x$, and $y^{P+1} = 0$. The temperature term however, does not involve t_{P+1}. Thus, we can write in the above Hamiltonian of W and t:

$$t^T \tilde{K}_L t = \sum_{\mu,\nu=1}^{P+1} t_{\mu} t_{\nu} \tilde{K}_L^{\mu\nu}, \quad x_{P+1} = x, \quad t_{P+1} = l, \quad \tilde{K}_L^{P+1,P+1} = K_L^{P+1,P+1}; \tag{67}$$

and compute the predictor moments

$$\langle f(x_{P+1})\rangle = i\partial_{t_{P+1}} \log Z|_{t_{P+1}=0}, \tag{68}$$

and

$$\langle \delta f^2(x_{P+1})\rangle = -\partial_{t_{P+1}}^2 \log Z|_{t_{P+1}=0}. \tag{69}$$

Integrating over the full set of W to evaluate $Z = \int \mathrm{d}W Z_L(W)$ is complicated. Instead, we integrate first over the top layer of weights and keep the rest of the W fixed. As before, we assume $x_i^L(x,W) = \phi(N^{-1/2} w^{Li} \cdot x^{L-1})$, and evaluate

$$Z_{L-1} = \int \Pi_{i=1}^{N} \mathrm{d}w^i \int \mathrm{d}^P t \exp\left(-\frac{1}{2}t^T \tilde{K}_L t + it^T Y - \frac{\|w^{iL}\|^2}{2\sigma^2}\right) \tag{70}$$

$$= \int \Pi_{i=1}^{N} \mathrm{d}w^i \int \mathrm{d}^P t \exp\left(-\frac{1}{2N}t^T \sum_{i=1}^{N} K_{w^i} t + it^T Y - \frac{T}{2}t^T t - \frac{\|w^{iL}\|^2}{2\sigma^2}\right), \tag{71}$$

where the PxP matrix K_{w^i} is defined via

$$K_w(x,x') = \sigma^2 \phi\left(N^{-1/2} w \cdot x^{L-1}(x)\right) \phi\left(N^{-1/2} w \cdot x^{L-1}(x')\right). \tag{72}$$

Switching around the order of integration and averaging over w:

$$Z_{L-1} = \int \mathrm{d}t \Pi_{i=1}^{N} \left\langle \exp -\frac{1}{2N} t^T K_{w^i} t \right\rangle_{w^i} \exp\left(it^T Y - \frac{T}{2}t^T t\right) \tag{73}$$

$$= \int \mathrm{d}t \exp\left[it^T Y + NG(t) - \frac{T}{2}t^T t\right], \tag{74}$$

J. Stat. Mech. (2024) 104007

where we have introduced the G function

$$G(t) = \log \left\langle \exp -\frac{1}{2N} t^T K_w t \right\rangle_w , \tag{75}$$

and the averages are w.r.t. to a single vector w^i with i.i.d. $\mathcal{N}(0,\sigma)$ components. The average w.r.t. w can also be written as

$$G(t) = \log \left\langle \exp -\frac{\sigma^2}{2N} t^T \phi(Z)\, \phi\left(Z^T\right) t \right\rangle_z , \tag{76}$$

where Z is a $P-$dim Gaussian vector with zero mean and correlations $\langle ZZ^T \rangle = K_{L-1}$. Finally, the average predictor can be calculated from the statistics of t; in particular its mean is given by

$$\langle f(x) \rangle = -i \langle t^T k(x,t) \rangle_t, \tag{77}$$

where

$$k(x,t) = \frac{\left\langle k_w(x) \exp -\frac{1}{2N} t^T K_w t \right\rangle_w}{\left\langle \exp -\frac{1}{2N} t^T K_w t \right\rangle_w}. \tag{78}$$

We see that in the end the problem boils down to evaluating the statistics of t. In the following section, we will focus on a class of deep networks where this problem is tractable: those with linear activations.

3.2. Deep linear networks

Linear networks are characterized by $\phi(z) = z$, such that

$$x_i^l(x,W) = N^{-1/2} w^{li} \cdot x^{l-1}. \tag{79}$$

Note a couple of obvious but important implications of such a class of functions. Firstly, these functions are linear mappings from input to output, regardless of depth. Basically, we do not gain expressive power with greater depth. Accordingly, the interpolation threshold is determined effectively by the ratio of the input dimension N_0 and number of patterns, i.e.

$$\alpha_0 = \frac{P}{N_0}. \tag{80}$$

Hence, we work in the regime $\alpha_0 < 1$. The overparametrization regime and interpolation thresholds are independent of N (there is a solution even for $N = 1$). Thus, in the linear case we must consider the thermodynamic limit as $P, N, N_0 \to \infty$ with both α and α_0 of $O(1)$. Despite the limitations on expressivity, the loss function can still be highly nonlinear due to the products of W with depth. As a result, statistical mechanics is highly nonlinear and the solution can still be rather complex and rich.

J. Stat. Mech. (2024) 104007

3.2.1. Evaluation of G for a linear network. To simplify things further we will focus on the zero-temperature limit. The term inside the exponent of our G function for linear activations can be written as

$$\frac{1}{2N}t^T K_w t = \frac{\sigma^2}{2N^2}\sum_{\mu\nu} t^\mu \left(w^T x^\mu_{L-1}\right)\left(w^T x^\nu_{L-1}\right) t^\nu \tag{81}$$

$$= \frac{\sigma^2}{2N} w^T M w, \tag{82}$$

where we define

$$M = \frac{1}{N} X^{L-1} t t^T X^{L-1T}. \tag{83}$$

This expression is now quadratic in w, and so the whole Gaussian integral is now quadratic. Hence, adding the quadratic L_2 term and integrating w yields

$$G = -\frac{1}{2}\log\det\left(I + \frac{\sigma^4}{N}M\right). \tag{84}$$

But M is a rank 1 matrix, so this just collapses to a scalar, which we can evaluate using:

$$\log\det\left(I + \frac{\sigma^4}{N}M\right) = \log\left(1 + \frac{\sigma^4}{N}\mathrm{Tr}M\right) \tag{85}$$

$$= \log\left(1 + \frac{\sigma^2}{N}t^T K_{L-1} t\right). \tag{86}$$

Therefore, we arrive at

$$G = -\frac{1}{2}\log\left[1 + \frac{\sigma^2}{N}t^T K_{L-1} t\right]. \tag{87}$$

We can substitute this expression for G back into equation (74) to get:

$$Z_{L-1} = \int \mathrm{d}t \exp\left[it^T Y - \frac{N}{2}\log\left(1 + \frac{\sigma^2}{N}t^T K_{L-1} t\right)\right]. \tag{88}$$

For simplicity, in the following we will consider the zero-temperature limit $T = 0$. It is a straightforward extension to consider the finite temperature case.

3.2.2. Integrating t. Now we would like to focus on integrating out the dependence on t. This constitutes a high-dimensional integral over t, but G is now a scalar function of the t's, making this a candidate for an order parameter, which we can compute using saddle point estimation.

$$Z_{L-1} = \int \mathrm{d}u_{L-1} \int_{-1} \mathrm{d}h_{L-1} \int \Pi \frac{\mathrm{d}t_\mu}{\sqrt{2\pi}} \tag{89}$$
$$\times \exp\left[it^T Y - \frac{N}{2}\log\left(1+h_{L-1}\right) + \frac{N}{2\sigma^2} h_{L-1}u_{L-1} - u_{L-1}t^T K_{L-1} t \right]$$
$$= \int \mathrm{d}u \int_{-1} \mathrm{d}h_{L-1}$$
$$\times \exp\left[-\frac{1}{2} Y^T \left(u_{L-1}K_{L-1}\right)^{-1} Y - \frac{1}{2}\log\det K_{L-1} + \frac{P}{2}\log u_{L-1} \right.$$
$$\left. -\frac{N}{2}\log\left(1+h_{L-1}\right) + \frac{1}{2\sigma^2} N u_{L-1} h_{L-1} \right]. \tag{90}$$

Let us compare this term with

$$Z_L = \exp\left[-\frac{1}{2} Y^T K_L^{-1} Y - \frac{1}{2}\log\det K_L \right]. \tag{91}$$

Note that in the linear case,

$$\langle K_L \rangle_{W_L} = \frac{1}{N}\sum_i \langle \left(w^{Li T} x^\mu_{L-1}\right)\left(w^{Li T} x^\nu_{L-1}\right) t^\nu \rangle_{w^i}, \tag{92}$$

where the average is w.r.t. a Gaussian measure of w^i, is simply

$$\langle K_L \rangle_{W_L} = \sigma^2 K_{L-1}. \tag{93}$$

Thus, we observe that the deviation from the naive GP limit, is in the scalar kernel renormalization factor u_{L-1}.

To summarize, averaging W_L yields a Hamiltonian for W_{L-1} with similar form to that of W_L, except for the appearance of two new global degrees of freedom, u_{L-1} and h_{L-1}, which represent the non-trivial contribution from marginalizing over W_L.

3.2.3. Saddle point. We now use the large P limit to evaluate the integral over these new global variables via a saddle point method:

$$u_L = \frac{1}{1+h_{L-1}}, \tag{94}$$

$$u_{L-1} = \sigma^2 \left[(1-\alpha) + N^{-1} Y^T \left(u_{L-1}K_{L-1}\right)^{-1} Y \right]. \tag{95}$$

Recall that before averaging over W_L, we had

$$\langle a^T a \rangle_a = \sigma^2 \left[N(1-\alpha) + Y^T {K_L}^{-1} Y \right]. \tag{96}$$

In fact, one can show that the RHS equals the norm squared of a averaged over both a and W_L. Hence, the order parameter has a nice interpretation, relating it to the norm squared of the readout weights, which is to be solved self-consistently via

J. Stat. Mech. (2024) 104007

$$Nu_{L-1} = \langle a^T a \rangle_{a,W_L}. \tag{97}$$

Summary: zero temperature

$$Z_{L-1} = \exp\left[N f\left(u_{L-1}, h_{L-1}\right) - \frac{1}{2} Y^T \left(\sigma^2 u_{L-1} K_{L-1}\right)^{-1} Y - \frac{1}{2} \log\det K_{L-1} \right] \tag{98}$$

$$\sigma^2 u_{L-1} = \sigma^2 \left[(1-\alpha) + N^{-1} Y^T \left(\sigma^2 u_{L-1} K_{L-1}\right)^{-1} Y \right] = N^{-1} \langle a^T a \rangle \tag{99}$$

$$K_{L-1} = \frac{\sigma^2}{N} X_{L-1}^T X_{L-1} \tag{100}$$

3.2.4. Evaluation of predictor statistics. Using the results above one can evaluate the predictor statistics (averaged over both a and W_L):

$$\langle f(x) \rangle = k_{L-1}^T(x) K_{L-1}^{-1} Y; \tag{101}$$

$$\langle (\delta f(x))^2 \rangle = u_{L-1} \left(K_{L-1}(x,x) - k_{L-1}^T(x) K_{L-1}^{-1} k_{L-1}(x) \right). \tag{102}$$

Note that the equation for the mean is identical to the GP limit (the renormalization factors cancel out). However, the predictor variance is scaled relative to the GP expression by the renormalization factor u_{L-1}.

3.3. Integrating all the weights-full solution

As we have seen, integrating W_L results in a Hamiltonian for the rest of the weights, which is similar in form to H_L, except for the kernel renormalization factor. Thus, we can continue integrating out layer by layer as depicted in figure 1(A). Each integration results in a similar H_l but with additional scalar integration variables. At the saddle point, all of them are identical (assuming the widths of each layer are equal). The process is complete when the first layer W_1 is integrated, yielding

$$Z_0 = \exp\left[N f\left(u_0, h_0\right) - \frac{1}{2} Y^T \left(u_0^L K_0\right)^{-1} Y - \frac{1}{2} \log\det K_0 \right], \tag{103}$$

which can be thought of as the entire volume of the solution space. The self-consistent equation for the Kernel renormalization (KR) factor is:

$$u_0 = \sigma^2 \left[(1-\alpha) + N^{-1} Y^T \left(u_0^L K_0\right)^{-1} Y \right] = \langle a^T a \rangle, \tag{104}$$

which is smooth for all $0 \leqslant \alpha \leqslant \frac{N_0}{N}$. The upper bound corresponds to the interpolation threshold where K_0 diverges. The fully averaged predictor statistics become

$$\langle f(x) \rangle = k_0^T(x) K_0^{-1} Y, \tag{105}$$

J. Stat. Mech. (2024) 104007

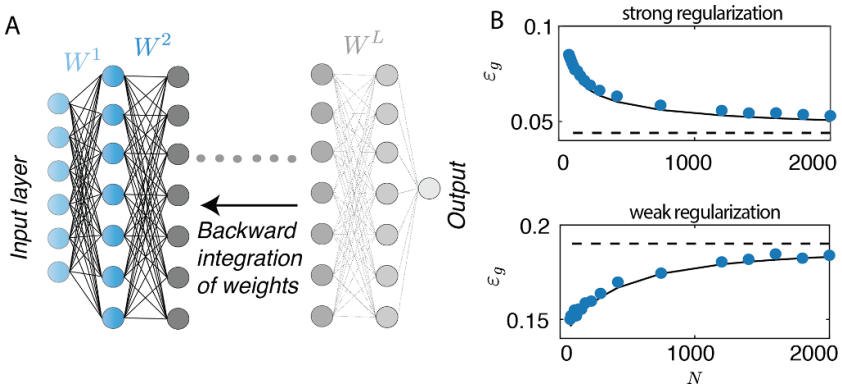

Figure 1. Finite width kernel renormalization. (A) A schematic of the BPKR approach. A renormalization factor is introduced at each step during backward integration until all the network weights are averaged out. (B) Generalization error on binary MNIST classification in fully connected rectified linear unit (ReLU) networks, for small (top) and large (bottom) regularization strength, σ: theory in black and numerical experiments in blue. Reprinted figure with permission from [1], Copyright (2021) by the American Physical Society.

and

$$\langle (\delta f(x))^2 \rangle = u_0^L \left(K_0(x,x) - k_0^T(x) K_0^{-1} k_0(x) \right). \tag{106}$$

This can be compared to the GP limit, where $u_0 = \sigma^2$.

3.4. Kernel statistics

We have shown how successive integration of the weights results in successive Hamiltonians, which depend on the kernels of the remaining 'top layers' but are renormalized by scalar KR factors. These factors renormalize the statistics of the predictor (at zero temperature, only its variance). However, the effect of finite α on the statistics of the kernels is subtle. We show a few results below.

1. Mean kernel:

$$\langle K_l \rangle = \sigma^{2l} \left[e^{-l/N} K_0 + \frac{m_l}{N} Y Y^T \right] \tag{107}$$

where m_l depends on $l, \sigma^2$and u_0

$$m_l = u_0^{-L} \left(\frac{\sigma^{2l} u_0^{-l} - 1}{\sigma^2 u_0^{-1} - 1} \right). \tag{108}$$

Thus, the leading correction to the GP limit is of only order $1/N$. Note that these corrections reflect the effect of learning on the posterior on W, since for all N

$$\langle K_l \rangle_{\text{gauss}} = \sigma^{2l} K_0. \tag{109}$$

2. Mean inverse kernel:
Here, we need to discuss two regimes:

Wide networks: $\alpha < 1$

$$\langle K_l^{-1} \rangle = \frac{1}{\sigma^{2l}(1-\alpha)^l} K_0^{-1} + O(1/N) \tag{110}$$

where the $1/N$ correction depends on the target labels. The leading corrections are simply the effect of finite α on the inverse kernel averaged over Gaussian weights.

Narrow networks: $\alpha > 1$

Here, the inverse kernel is divergent but can be regularized by temperature, T. Adding the temperature term, one obtains that the mean inverse kernels are proportional to $1/T$. For instance,

$$\frac{1}{N}\text{Tr}\langle (K_L + TI)^{-1} \rangle = \frac{(\alpha - 1)}{T}. \tag{111}$$

3.5. Approximate kernel renormalization for nonlinear deep neural networks

$$f(x, \Theta) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i x_i^L(x, W) \tag{112}$$

$$Z(\mathcal{D}) = \int \mathrm{d}\Theta \exp\left(-\frac{\beta}{2} \sum_{\mu=1}^{P} \left[\frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i x_i^L(x^\mu, W) - y^\mu \right]^2 - \frac{1}{2\sigma^2} \|\Theta\|^2 \right) \tag{113}$$

$$P_L(W) = Z^{-1} \exp\left[-\frac{\|W\|^2}{2\sigma^2} - \frac{1}{2} Y^T K_L^{-1} Y - \frac{1}{2} \log\det K_L \right]. \tag{114}$$

Note that K_L depends on W

$$K_L(x, y) = \sigma^2 N^{-1} X^{LT} X^L \tag{115}$$

$$K_L(x^\mu, y^\nu) = \sigma^2 N^{-1} X^{LT} X^L = \frac{\sigma^2}{N} \sum_i \phi\left(w^{iL} \cdot x_{L-1}(x^\mu)\right) \phi\left(w^{iL} \cdot x_{L-1}(x^\nu)\right). \tag{116}$$

By integrating out the top layer we have introduced auxiliary Pt variables and arrived at

$$Z_{L-1} = \int \mathrm{d}t \exp\left[it^T Y + NG(t) - \frac{T}{2} t^T t \right] \tag{117}$$

J. Stat. Mech. (2024) 104007

$$G(t) = \log \left\langle \exp -\frac{1}{2N} t^T K_w t \right\rangle_w \tag{118}$$

where the averages are w.r.t. to a single vector w^i with i.i.d. $\mathcal{N}(0,\sigma)$ components. The average w.r.t. w can also be written as

$$G(t) = \log \left\langle \exp -\frac{\sigma^2}{2N} \left(t^T \phi(Z)\right)^2 \right\rangle_z = \log \left\langle \exp -\frac{\sigma^2}{2N} \left(t^T \phi(Z)\right)^2 \right\rangle_z \tag{119}$$

where Z is a $P-$dim Gaussian vector with zero mean and correlations $\langle ZZ^T \rangle = K_{L-1}$.

The problem is that $G(t)$ is a complicated function of t for a nonlinear network.

Here, we make a naive approximation where we approximate the distribution of the random scalar variable v

$$v = \sigma t^T \phi(Z) \tag{120}$$

by a Gaussian. Depending on the nonlinearity, this variable may have a non-zero mean. For simplicity, I ignore the mean for now and pretend it has zero mean, so that the variance is simply

$$\langle v^2 \rangle = \sigma^2 t^T \langle \phi(Z)\, \phi\left(Z^T\right) \rangle t = t^T K_L^{\mathrm{GP}} t \tag{121}$$

so that

$$G(t) \approx \frac{1}{2} \log \left(1 + \frac{\sigma^2}{N} t^T K_L^{\mathrm{GP}} t\right). \tag{122}$$

Compare with the linear case. There we obtained

$$G(t) = \frac{1}{2} \log \left(1 + \frac{\sigma^4}{N} t^T K_{L-1} t\right) \tag{123}$$

but recall that in the linear case,

$$K_L^{\mathrm{GP}} = \sigma^2 K_{L-1} \tag{124}$$

so it is consistent with the above.

Finally, if we iterate this approximate kernel renormalization we obtain,

$$\langle f(x) \rangle = k_L^{\mathrm{GP}T}(x)\, K_L^{\mathrm{GP}-1} Y \tag{125}$$

$$\langle (\delta f(x))^2 \rangle = \sigma^{-2L} u_0^L \left(K_L^{\mathrm{GP}}(x,x) - k_L^{\mathrm{GP}T}(x)\, K_L^{\mathrm{GP}-1} k_L^{\mathrm{GP}}(x)\right) \tag{126}$$

$$u_0 = \sigma^2 \left[(1-\alpha) + N^{-1} Y^T \left(u_0^L \sigma^{-2L} K_L^{\mathrm{GP}}\right)^{-1} Y\right] \tag{127}$$

https://doi.org/10.1088/1742-5468/ad3a60

J. Stat. Mech. (2024) 104007

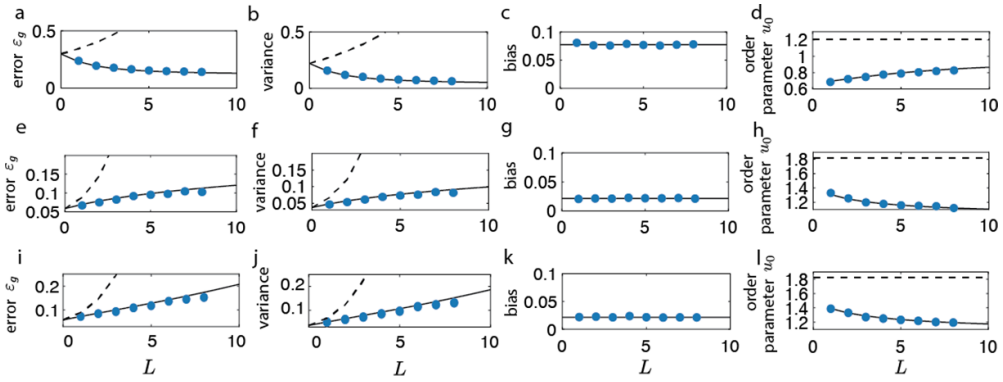

Figure 2. Dependence of the generalization error on the network depth L. The generalization error (a), (e), (i), variance (b), (f), (j), bias of the predictor (c), (g), (k), and the order parameter u_0 (d), (h), (l) as a function of L. Black lines: theory. Blue dots: simulation. Black dashed lines: the GP limit ($N \to \infty$). (a)–(d) The sub-regime, where the generalization error decreases with L. (e)–(h) The sub-regime, where the generalization error increases with L approaching a finite limit. (i)–(l) The high noise regime, where the generalization error increases with L and diverges as $L \to \infty$. Reprinted figure with permission from [1], Copyright (2021) by the American Physical Society.

or absorbing σ^2into u_0

$$\langle (\delta f(x))^2 \rangle \approx u_0^L \left(K_L^{\mathrm{GP}}(x,x) - k_L^{\mathrm{GP}T}(x) K_L^{\mathrm{GP}-1} k_L^{\mathrm{GP}}(x) \right) \tag{128}$$

$$u_0 = \left[(1-\alpha) + N^{-1} Y^T \left(u_0^L K_L^{\mathrm{GP}} \right)^{-1} Y \right]. \tag{129}$$

This result allows us to study the dependence of generalization error on network depth (figure 2), to study the evolution of kernels throughout the layers of a trained network (figure 3), and yields accurate predictions even for ReLU networks trained on MNIST (figures 4 and 5).

4. Lecture 3: globally gated deep linear neural networks

4.1. Introduction: architectures

Neural networks can include various types of gating units, including LSTM cells [3] or grated recurrent unit (GRU)'s [4]. These include multiplicative interactions between two vectors, which can be interpreted as one vector attending to specific subsets of the other. A minimal example of gatings acting on the local scale on the pre-activation of a single neuron $w \cdot x$ is the ReLU (rectified linear unit) activation function ϕ. It is defined as

$$\phi(w \cdot x) = (w \cdot x)\, \theta(w \cdot x) \tag{130}$$

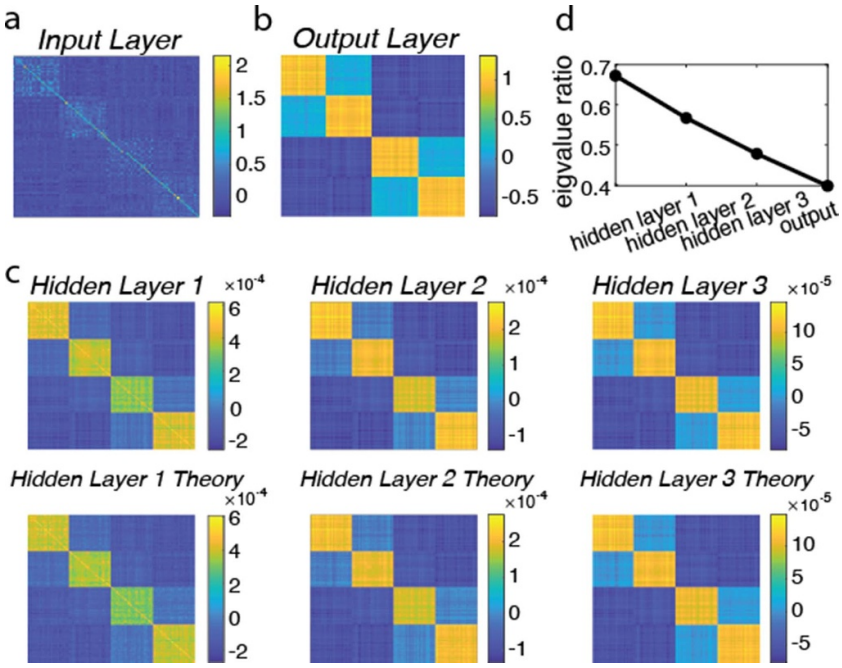

Figure 3. Simulation and theory for the mean kernel for the binary classification task on MNIST. The network is trained on four different MNIST digits, which are grouped into two higher-order categories. The output of the network is six-dimensional: four of the output units are 'one-hot' representations of the four digits, the other two outputs label the inputs according to their high-order category. (a) The input similarity matrix. (b) The output similarity matrix. (c) The average kernel of the hidden layer for $l = 1, 2, 3$. Top: simulation. Bottom: theory. (d) The ratio between the mean of the second and third largest eigenvalues (corresponding to the magnitude of the four smaller blocks) and the largest eigenvalue (corresponding to the magnitude of the two larger blocks) of the non-GP correction terms in the mean layer kernels is monotonically decreasing with l. Reprinted figure with permission from [1], Copyright (2021) by the American Physical Society.

where $\theta(a) = 1_{a \geqslant 0}$ is the thresholding Heaviside function. This idea can be generalized to gated linear units (GaLU) [5], where a new parameter v adjusts the thresholding values via

$$\phi(w \cdot x) = (w \cdot x)\,\theta(v \cdot x)\,. \tag{131}$$

Note that v is not learned, but randomized and fixed during training, slightly akin to random features (for more variations of this architecture refer to [6–8]).

In the cases above, the gating mechanism is applied locally, to a single hidden neuron. Another local structure that can be gated is multiple synapses in conjunction.

Another alternative version of gating is **global gating** where, at each layer, the gating mechanism depends on the original input x directly (figure 6) [9]. Biologically motivated, dendrites work as inhibitory or stimulating gates to a subset of neurons, as depicted in figure 7. For each of the M dendrites, the inputs x are gated via the gating

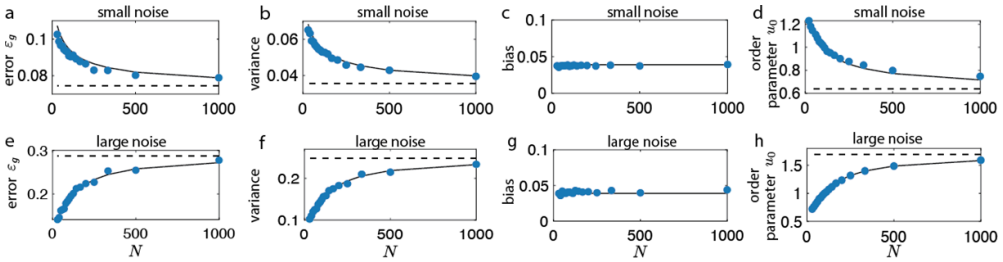

Figure 4. A single hidden layer ($L=1$) ReLU network trained on MNIST binary classification of two digits (0 and 1). The generalization error (a), (e), variance (b), (f), and squared bias (c), (g) of the predictor, and the order parameter u_0 (d), (h) as a function of N. Black lines: theory. Blue dots: simulation. Black dashed lines: GP limit ($N \to \infty$). (a)–(d) Results in the small noise regime, where the generalization error decreases with N. (e)–(h) Results in the large noise regime, where the generalization error increases with N. Reprinted figure with permission from [1], Copyright (2021) by the American Physical Society.

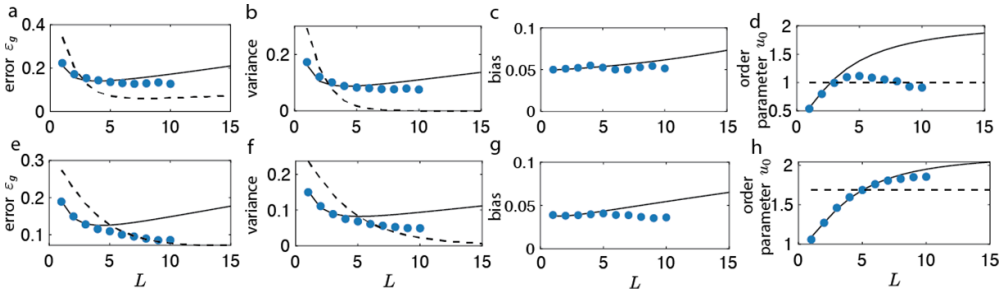

Figure 5. Generalization errors of deep ReLU networks as a function of depth L. Blue dots: simulation. Black lines: theoretical approximation. Black dashed lines: GP limit. The generalization error (a), (e), variance (b), (f), and bias (c), (g) of the predictor, and the order parameter u_0 (d), (h). (a)–(d) Results for the 'template' model with noisy linear teacher labels. (e)–(h) Results for a binary MNIST classification task. Reprinted figure with permission from [1], Copyright (2021) by the American Physical Society.

functions g_m. These functions are shared across several neurons per layer. This setup can be formalized as

$$f(x) = \frac{1}{\sqrt{NM}} \sum_{i=1}^{N} \sum_{m=1}^{M} a_i^m x_i^L g_m(x) \tag{132}$$

where the g_m are gating functions, which are not specified further for now. Each individual layer $l > 1$ is gated again in terms of

$$x_i^l = \frac{1}{\sqrt{NM}} \sum_{j=1}^{N} \sum_{m=1}^{M} W_{ij}^{l,m} x_j^{l-1} g_m(x) \,. \tag{133}$$

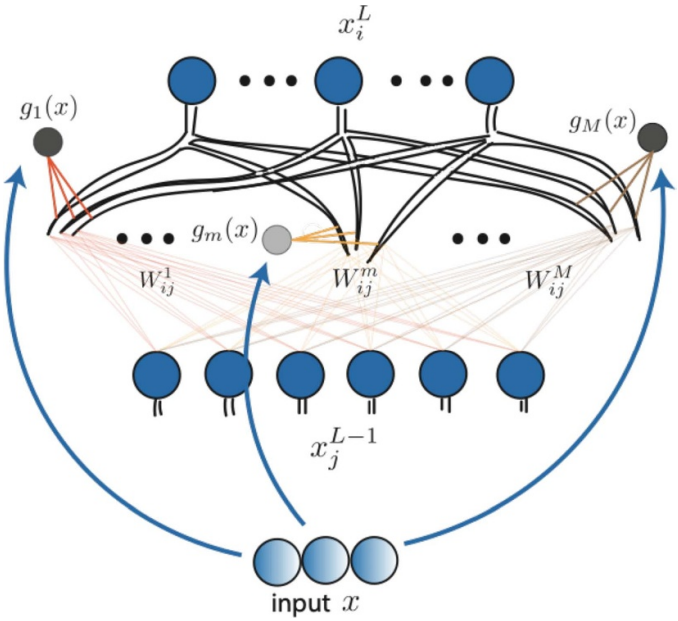

Figure 6. An example of the configurations of a neural network with global gating. The gating functions g_m directly depend on the input x and regulate the hidden layers x_i^L. See [9] for details. Reprinted figure with permission from [1], Copyright (2021) by the American Physical Society.

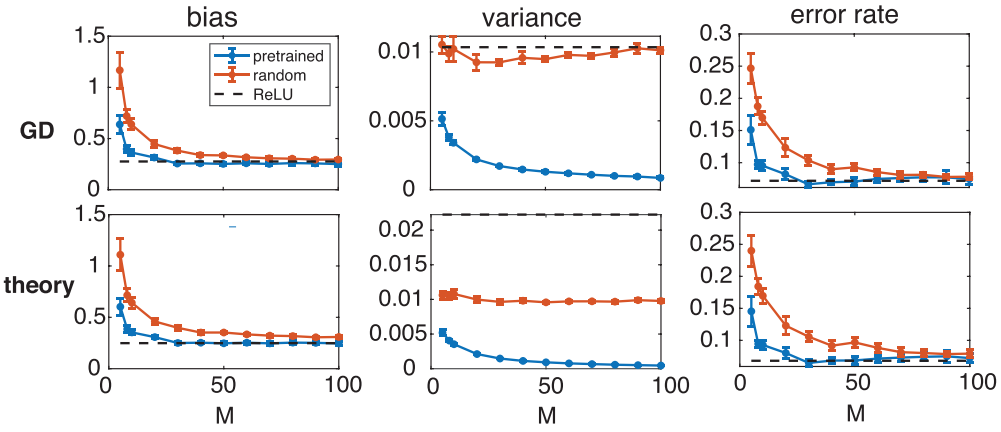

Figure 7. Dependence of generalization on M for globally-gated deep linear network (GGDLN)s trained on the MNIST dataset. The bias, variance, and error rate as a function of M for random and pretrained gatings; the theory agrees well with gradient descent (GD) dynamics. The performance of GGDLNs improves and approaches the ReLU network for sufficiently large M, and improves faster for pretrained gatings compared to random gatings. See [9] for further details. Reprinted figure with permission from [1], Copyright (2021) by the American Physical Society.

Since adding gatings to the input layer is equivalent to expanding the input dimension and replacing x_j by $x_j g_m(x)$, and learning does not affect how the gatings interact with the input, we do not add gatings to the input layer for simplicity. Then, the first layer takes on the form

J. Stat. Mech. (2024) 104007

$$x_i^1 = \frac{1}{\sqrt{N_0}} \sum_{j=1}^{N_0} W_{ij}^1 x_j \,. \tag{134}$$

Introducing gating functions at hidden layers that act on the original input $g_m(x)$ can be viewed as skip connections. Note that generally, the number and parameters of gating units may vary between layers. However, for simplicity, we will fix both as constant over all layers. Since no learning is involved in the gating functions g_m, the biological interpretation/motivation for this model can be viewed as giving contextual information about the task into the layer via the gating function. However, they might be adapted externally in a top-down approach, allowing a network to achieve a different task with the same neurons but different gating functions.

4.2. Memory capacity

We want to inspect the memory capacity, or the expressivity of such a globally gated network, i.e. how many data samples P it can distinguish. The input–output relation, equation (132), can be written as a linear function w.r.t. the expanded input x

$$f(x) = \sum_{m_1,\ldots,m_L,j} W^{\text{eff}}_{m_1,\ldots,m_L,j} x^{\text{eff}}_{m_1,\ldots,m_L,j} \tag{135}$$

where the effective input $x_{\text{eff}}(x)$ is of $N_0 M^L$-dimension with all possible gating orders applied to every original input

$$x^{\text{eff}}_{m_1,\ldots,m_L,j} = g_{m_1}(x)\, g_{m_2}(x) \cdots g_{m_L}(x)\, x_j \,, \ (m_l = 1,\ldots,M; j = 1,\ldots,N_0) \tag{136}$$

and the effective weights are

$$W^{\text{eff}}_{m_1,\ldots,m_L,j} \propto \sum_{\{i_l\}} a_{i_L}^{m_L} w_{i_L i_{L-1}}^{L m_{L-1}} \ldots w_{i_2 i_1}^{2 m_1} w_{i_1 j}^{1}. \tag{137}$$

As the gating units are shared across layers, the effective input has $N_0 \binom{M+L-1}{L}$ **independent dimensions**. Here, the combinatorial term comes from the number of possible combinations of choosing L gatings from M total number of gatings with repetitions. Assuming $N \gg M^L$, i.e. the number of neurons per layer is much larger than the number of gatings, which we will assume throughout, the capacity is

$$P \leqslant N_0 \binom{M+L-1}{L} . \tag{138}$$

The case is different for small N since, in this case, $W^{\text{eff}}_{m_1,\ldots,m_L,j}$ is lower rank. In the more general case of different sets of gatings in each layer, a similar argument yields

$$P \leqslant N_0 M^L \tag{139}$$

again for $N \gg M^L$.

J. Stat. Mech. (2024) 104007

To conclude, the expressivity increases with more gatings M, which is something that cannot be observed in purely linear networks, but is qualitatively similar to the nonlinearities in deep neural nets.

4.3. Statistical mechanics: the GP limit

We begin with

$$f(x) = \frac{1}{\sqrt{NM}} \sum_{i=1}^{N} \sum_{m=1}^{M} a_{i,m} x_{L,i} g_m(x) \tag{140}$$

and introduce the MSE loss with P samples and Gaussian prior as before,

$$Z = \int \mathrm{d}\Theta \exp\left[-\frac{1}{2T} \sum_{\mu=1}^{P} \left(\frac{1}{\sqrt{NM}} \sum_{i=1}^{N} \sum_{m=1}^{M} a_{i,m} x_{L,i}^{\mu} g_m(x^{\mu}) - y^{\mu} \right)^2 - \frac{1}{2\sigma^2} \Theta^{\top} \Theta \right]. \tag{141}$$

Integrating over $\mathbf{a}$ in the $T \to 0$ limit, we have

$$Z = \int \mathrm{d}\boldsymbol{W} \int \Pi_{\mu=1}^{P} \mathrm{d}t_{\mu} \exp\left[-\frac{1}{2} t^{\top} K_L(\boldsymbol{W}) t + i t^{\top} Y - \frac{1}{2\sigma^2} \mathrm{Tr}\left(\boldsymbol{W}^{\top} \boldsymbol{W} \right) \right] \tag{142}$$

and

$$\langle f(x) \rangle = k_L(x)^{\top} K_L^{-1} Y \tag{143}$$

$$\langle \delta f(x)^2 \rangle = K_L(x,x) - k_L(x)^{\top} K_L^{-1} k_L(x) \tag{144}$$

where the average is only w.r.t. the readout weights, so the kernels depend on all W's. In our case, the kernels are the product of the dot product of the input pairs and the dot product of the pair of vectors of the output gating units as

$$K_L^{\mu\nu}(\mathbf{W}) = \left(\frac{\sigma^2}{M} g(x^{\mu})^{\top} g(x^{\nu}) \right) \left(\frac{1}{N} x_L^{\mu}(\mathbf{W})^{\top} x_L^{\nu}(\mathbf{W}) \right) \tag{145}$$

where μ, ν are indices of data inputs.

To obtain the GP limit as before, we fix P (and M) and let N go to infinity, resulting in self-averaged kernels with Gaussian weights,

$$K_L(x, y|\mathbf{W}) \to K_{\mathrm{GP}}(x,y) = \left(\frac{\sigma^2}{M} g(x)^{\top} g(y) \right)^{L} K_0(x,y) \tag{146}$$

$$K_0(x,y) = \frac{\sigma^2}{N_0} x^{\top} y. \tag{147}$$

J. Stat. Mech. (2024) 104007

Note that the GP kernel matrix becomes singular as P approaches the above-mentioned network capacity, also known as the interpolation threshold, which results in a diverging bias and vanishing variance.

4.4. Back-propagated kernel renormalization

4.4.1. A single hidden layer. Let us start with a single hidden layer. Integrating W we obtain,

$$Z = \int \Pi_{\mu=1}^{P} \mathrm{d}t_\mu \exp\left[it^\top Y + NG(t)\right] \tag{148}$$

$$G(t) = \log\left\langle \exp -\frac{1}{2N} t^\top K_1^w t \right\rangle_w \tag{149}$$

$$t^T K_1^w t = \sum_{\mu\nu} t^\nu t^\mu \frac{\sigma^2}{M} g\left(x^\mu\right)^\top g\left(x^\nu\right) \frac{1}{N_0} x^{\mu\top} w_{1,i} w_{1,i}^\top x^\nu. \tag{150}$$

Adding the Gaussian measure on w, we obtain,

$$\frac{1}{2} w_{1,i}^\top A w_{1,i} \tag{151}$$

where A is the NxN matrix

$$A = \sigma^{-2} I + \frac{\sigma^2}{M} \frac{1}{N} \sum_{\mu\nu} t_\mu t_\nu g\left(x^\mu\right)^\top g\left(x^\nu\right) \frac{1}{N_0} x^\mu x^{\nu T}. \tag{152}$$

Now, A has rank M (recall that without the weights it was a scalar), hence the integral yields

$$G(t) = -\frac{1}{2} \log\det\left(I + \frac{1}{N}\mathcal{H}\right) \tag{153}$$

$$\mathcal{H}^{mn} = \frac{\sigma^2}{M} \sum_{\mu\nu} t_\mu t_\nu g_m\left(x^\mu\right) g_n\left(x^\nu\right) K_0^{\mu\nu} \tag{154}$$

and $\mathcal{H}$ is our new order parameter in matrix form. Reinserting this into our original normalization constant we obtain

$$Z = \int \mathrm{d}U \int \mathrm{d}\mathcal{H} \int \mathrm{d}t \exp\left[it^\top Y - \frac{1}{2} t^\top \tilde{K} t - \frac{N}{2} \log\det\left(I + \mathcal{H}\right) + \frac{N}{2\sigma^2} \mathrm{Tr}\left(U\mathcal{H}\right)\right] \tag{155}$$

with the renormalized kernel and gates

$$\tilde{K}^{\mu\nu} = \frac{1}{M} g(x^\mu)^\top U g(x^\nu)) K_0^{\mu\nu}, \tag{156}$$

J. Stat. Mech. (2024) 104007

$$\tilde{K} = \frac{1}{M} K_0 \circ \tilde{g}^T \tilde{g}\,, \tag{157}$$

$$\tilde{g} = \sqrt{U} g\,, \tag{158}$$

and the U is acting as a delta function on $\mathcal{H}$ when we integrate over it:

$$U = I - N^{-1} M^{-1} \tilde{g}^\top \left[\tilde{K}^{-1} \circ K_0\right] \tilde{g} + N^{-1} M^{-1} \tilde{g}^\top \left[\tilde{K}^{-1} Y Y^\top \tilde{K}^{-1} \circ K_0\right] \tilde{g}\,, \tag{159}$$

$$U_{mn} = \left\langle \frac{1}{N} \sum_{i=1}^{N} a_{i,m} a_{i,n} \right\rangle . \tag{160}$$

Finally, the mean and variance become

$$\langle f(x) \rangle_\Theta = \tilde{k}(x)^\top \tilde{K}^{-1} Y \tag{161}$$

$$\langle \delta f(x)^2 \rangle_\Theta = \tilde{K}(x,x) - \tilde{k}(x)^\top \tilde{K}^{-1} \tilde{k}(x)\,. \tag{162}$$

Ergo, renormalization happens via a matrix order parameter and not a scalar. For the global gated units, the renormalization not only modifies the amplitude but also the shape of the kernel. Hence, both the variance and the mean of the predictor are affected.

5. Lecture 4: manifold representations in deep neural network (DNN)s 1: separability and geometry

5.1. Biological motivation

Both humans and animals have the innate ability to succeed in discriminating different objects despite substantial variability in physical characteristics. For example, mammals are able to recognize objects despite variations in orientation, position, pose, illumination, and background, or can even discriminate different smells in the presence of different odor concentrations.

5.2. Model of manifolds

To conceptualize perceptual manifolds, let us consider a set of N neurons responding to a specific sensory signal associated with an object. The neural response to the provided stimulus is a vector in $\mathbb{R}^N$. Changes in the physical parameters of the input stimulus do not change the object identity modulating the neural state vector. The set of all state vectors corresponding to responses to all possible stimuli associated with the same object can be viewed as a manifold in the neural state space. From this geometrical perspective, object recognition is equivalent to the task of discriminating the manifolds of different objects from each other (figure 8).

Definition 1. A manifold M is a compact subset of an affine subspace of $\mathbb{R}^N$ with affine dimension $D < N$. Let $\mathcal{S}$ be the set that defines the shape of the manifolds, a point $\mathbf{x} \in M$ s parameterized as

J. Stat. Mech. (2024) 104007

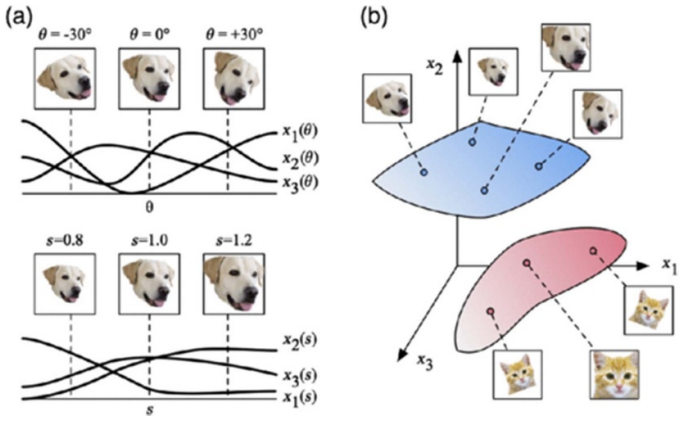

Figure 8. Population activity of N neurons in response to images of dogs of different sizes and rotations lies along a manifold. Reprinted figure with permission from [11], Copyright (2018) by the American Physical Society.

$$\mathbf{x}(\vec{s}) = \mathbf{x}_0 + \sum_{i=1}^{D} s_i \mathbf{u}_i, \tag{163}$$

where $\{\mathbf{u}_i\}_{i=1}^{D}$ is the set of orthonormal bases of the D-dimensional linear subspace containing M, and $\vec{s} \in \mathcal{S}$ is a D-dimensional vector whose components s_i represent the coordinates of the manifold point within this subspace constrained to be in the set $\mathcal{S}$. This definition is schematized in figure 9.

For now on we will consider P perceptual manifolds $\{M^\mu\}_{\mu=1,\ldots,P}$ corresponding to P perceptual objects and, for simplicity, we will make the following assumptions:

1. **Centered manifolds:** D-dimensional subspace, each perpendicular to the centers $\mathbf{x}_0^\mu \in M^\mu \forall \mu = 1, \ldots, P$.
2. **Same geometry:** the coordinate set $\mathcal{S}$ is the same for all the manifolds (extensions that consider heterogeneous geometries will be discussed later).

Examples of manifolds and the different notions of dimension:

- Linear—varying intensity
- 2D cosine tuning to angle, varying angle and amplitude
- 2D higher Fourier transforms
- Object manifolds
- Data manifolds
- Dynamic manifolds

5.3. Linear separability of manifolds

We start our analysis by defining the separability of objects on the basis of their representations in a given neuronal population. We denote a collection of objects as linearly

J. Stat. Mech. (2024) 104007

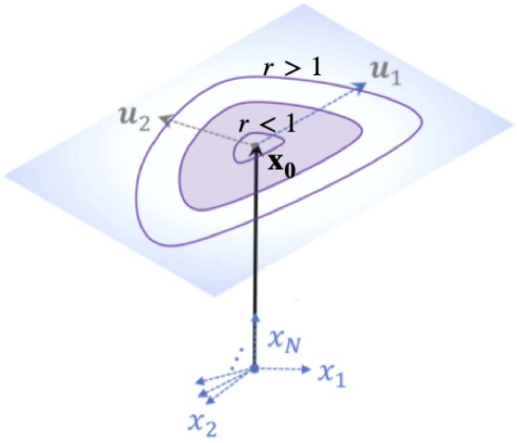

Figure 9. A $D=2$ manifold embedded in $\mathbb{R}^N$. Reprinted figure with permission from [11], Copyright (2018) by the American Physical Society.

separable (or simply separable) if they can be classified into two desired classes by a hyperplane in the state space of the population.

For this reason, to investigate the separability properties of manifolds, it is helpful to consider scaling a manifold M^μ by an overall scale factor $r \in \mathbb{R}^+$ without changing its shape.

Definition 2. We define the scaling relative to the center by a scalar $r \in \mathbb{R}^+$, by

$$rM^\mu := \left\{ r \sum_{i=1}^{D} s_i \mathbf{u}_i^\mu \mid \vec{s} \in \mathcal{S} \right\}. \tag{164}$$

Let us observe that when $r \to 0$, the manifold rM^μ converges to a point $\mathbf{x}_0^\mu$. However, when $r \to \infty$, the manifold rM^μ spans the entire affine subspace. If the manifold is symmetric (such as for an ellipsoid), there is a natural choice for a center. We will later provide an appropriate definition for the center point for general asymmetric manifolds.

Task: We study the separability of P manifolds into two classes, denoted by binary labels $y^\mu = \pm 1$, by a linear hyperplane passing through the origin. A hyperplane is described by a weight vector $\mathbf{w} \in \mathbb{R}^N$, normalized so $\|\mathbf{w}\|^2 = N$ and the hyperplane correctly separates the manifolds with a margin $\kappa \geqslant 0$ if it satisfies,

$$y^\mu \mathbf{w} \cdot \mathbf{x}^\mu \geqslant \kappa, \quad \forall \mathbf{x}^\mu \in M^\mu. \tag{165}$$

Now, to do some work, we have to make some **statistical assumptions**:

1. Random positions and orientations: all components of $\mathbf{u}_i^\mu$ and $\mathbf{x}_0^\mu$ are drawn i.i.d. from Gaussian distributions with zero mean and variance $\frac{1}{N}$.
2. Random labeling: the binary labels $y^\mu = \pm 1$ are randomly assigned to each manifold with equal probabilities.

J. Stat. Mech. (2024) 104007

3. Thermodynamic limit: we will study the thermodynamic limit where $N, P \to \infty$, but with a finite load $\alpha := \frac{P}{N}$. In addition, the manifold geometries as specified by the set $\mathcal{S} \in \mathbb{R}^D$ and, in particular, by their affine dimension, D, are held fixed in the thermodynamic limit.

5.3.1. Bounds on capacity. By varying the scale factor r it is easy to derive the following bounds on the linear separability of general manifolds with finite margin κ. We will call the capacity the maximal value of α, denoted by α_M, for which there is a weight vector separating the manifolds correctly with probability 1 in the thermodynamic limit.

First, in the limit of $r \to 0$, the problem is reduced to linear separation of P random points, the centers; hence, the capacity is given by Gardner's formula,

$$\alpha_M = \alpha_0(\kappa), \tag{166}$$

which is the maximum load for separation of random i.i.d. points with a margin κ given by the Gardner theory [10],

$$\alpha_0^{-1}(\kappa) = \int_{-\infty}^{\kappa} \mathrm{D}t\,(t-\kappa)^2 \tag{167}$$

with Gaussian measure $Dt = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$.

When $r \to \infty$, the weight vector must be orthogonal to the total subspace spanned by the manifolds. Hence, the effective dimension of the problem is $N - PD$, and in this null space it is again reduced to separating the centers (or rather their projection onto the null space). Hence, $P < \alpha_0(\kappa)(N - PD)$ yielding $P < \alpha_0(\kappa)N/(1 + D\alpha_0(\kappa))$. Combining the two limits we obtain,

$$\alpha_0(\kappa) \geqslant \alpha_{\mathrm{M}}(\kappa) \geqslant \frac{\alpha_0(\kappa)}{1 + \alpha_0(\kappa) D} \tag{168}$$

and for zero margin,

$$2 \geqslant \alpha_{\mathrm{M}} \geqslant \frac{2}{1+2D}. \tag{169}$$

For many interesting cases, the affine dimension D is large and the gap in equation (168) is overly loose. Hence, it is important to derive an estimate of the capacity for manifolds with finite sizes and evaluate the dependence of the capacity and the nature of the solution on the geometrical properties of the manifolds, as shown below.

For random point clouds with m points per manifold for zero margin the capacity is

$$\alpha_{\mathrm{Random}} = \frac{2}{m}. \tag{170}$$

Therefore, in this case, equation (169) will be

$$2 \geqslant \alpha_{\mathrm{M}} \geqslant \frac{2}{m}. \tag{171}$$

Now, for large D and m we want to understand the conditions under which

$$\alpha_M \gg \frac{1}{D}, \frac{1}{m}. \tag{172}$$

This is the unentangled regime.

Convex hulls:
Finally, we note that the linear separation of manifolds is a convex operation. Hence, a solution separates the manifolds if it separates their convex hulls.

5.4. Support manifolds and manifold anchor points

Consider a maximum margin solution to the separation of our manifolds. This solution can be written in terms of a subset of the examples, the support vectors which lie on the margin hyperplanes. In general, a fraction of the manifolds will be interior; they do not overlap with the margin planes, so will not contribute to the solution vector. Other manifolds will have non-zero overlap with the margin planes; these manifolds are the *support manifolds* of the system. The nature of their support varies and can be characterized by the dimension, k, of the span of the intersecting set of conv(S) with the margin planes. Some support manifolds, which we call *touching* manifolds, intersect with the margin hyperplane only at a single point. They have a support dimension $k = 1$, and this touching point is on the boundary of $\mathcal{S}$.

The other extreme is *fully supporting* manifolds, which completely reside in the margin hyperplane. They are characterized by $k = D + 1$. In this case, w is parallel to the translation vector $\vec{c}$ of S. Hence, all the points in S are support vectors, and all have the same overlap, κ. For smooth convex hulls (i.e. when S is strongly convex), no other manifold support configurations exist. For other types of manifolds, there are also *partially supporting* manifolds, whose convex hull intersections with the margin hyperplanes consist of k dimensional faces with $1 < k < D + 1$. For instance, $k = 2$ implies that the intersecting set is an edge whereas, in the case of $k = 3$, it is a planar 2-face of the convex hull.

From the above discussion, it follows that, in general, support manifolds may have many support vectors, i.e. vectors that are on the margin planes. However, one can always combine their contribution to $\mathbf{w}$ and represent them as a single vector; therefore, one can write,

$$\mathbf{w} = \sum_{\mu=1}^{P} \lambda_\mu y^\mu \tilde{\mathbf{x}}^\mu, \lambda_\mu \geqslant 0 \tag{173}$$

J. Stat. Mech. (2024) 104007

where each manifold contributes at most a single point, $\tilde{\mathbf{x}}^{\mu}$. For manifolds with non-zero contribution to $\mathbf{w}$, this vector is unique and is called the *manifold's anchor point*. Note that the anchor point is a convex combination of points on the manifold; hence, it is a vector in the convex hull of the μth manifold, $\tilde{\mathbf{x}}^{\mu} \in \mathrm{conv}\left(M^{\mu}\right)$, but not necessarily on the original manifold itself. More specifically, when the manifold is touching, the anchor point is just the touching point. However, when $k > 1$, the anchor points are *unique points* residing in the k dimensional faces of the intersection between $\mathrm{conv}\left(M^{\mu}\right)$ and the margin planes.

5.5. Mean-field theory for manifold separation

Following Gardner's framework, using the mean-field theory, it is possible to derive an expression for the capacity of the linear separation of manifolds, $\alpha_M(\kappa)$, in the thermodynamic limit.

We can write the volume of the space of the solution as

$$V = \int \mathrm{d}^N \mathbf{w} \delta\left(\|\mathbf{w}\|^2 - N\right) \Pi_{\mu, \mathbf{x}^{\mu} \in M^{\mu}} \Theta\left(y^{\mu} \mathbf{w} \cdot \mathbf{x}^{\mu} - \kappa\right). \tag{174}$$

Here, $\Theta(\cdot)$ is the Heaviside function to enforce the margin constraint in equation (165) and the delta function ensures that $\|\mathbf{w}\|^2 = N$.

Defining

$$h^{\mu}_{\min} := \min_{\vec{s} \in S} y^{\mu} \mathbf{w} \cdot \mathbf{x}^{\mu}\left(\vec{s}\right) = v^{\mu}_0 + \min_{\vec{s} \in S} v^{\mu} \cdot s, \tag{175}$$

we can rewrite equation (173) as

$$V = \int \mathrm{d}^N \mathbf{w} \delta\left(\|\mathbf{w}\|^2 - N\right) \Pi_{\mu} \Theta\left(h^{\mu}_{\min} - \kappa\right). \tag{176}$$

Using the Replica method, it is possible to find that, in the thermodynamic limit, the general form of the inverse capacity is

$$\alpha^{-1}_{\mathrm{M}}(\kappa) = \left\langle F\left(t_0, t\right)\right\rangle_{t_0, \vec{t}} \tag{177}$$

$$F\left(t_0, t\right) = \min_{v_0, v} \left\{ \left(v_0 - t_0\right)^2 + \|v - t\|^2 \mid \min_{\vec{s}} \left\{ v \cdot s \mid \vec{s} \in \mathcal{S} \right\} + v_0 - \kappa \geqslant 0 \right\} \tag{178}$$

where $t_i \sim \mathcal{N}(0,1)$, and the components of the vector V represent the signed fields induced by the solution vector $\mathbf{w}$ on the basis vectors of the manifold. The Gaussian vector $\vec{t}$ represents the part of the variability in v due to quenched variability in the manifold's basis vectors and the labels.

5.5.1. Interpretation: anchor points. To gain insight into the nature of the maximum margin solution, it is useful to consider the Karush–Kuhn–Tucker (KKT) conditions of the convex optimization in equation (177). Solving the above optimization problem can be done using KKT formalism.

For each sampled t, we add the Lagrange multiplier

$$(v_0 - t_0)^2 + \|v - t\|^2 - \lambda (v \cdot s + v_0 - \kappa) = 0 \tag{179}$$

where

$$\begin{aligned} &\lambda \geqslant 0 \\ &v \cdot s + v_0 - \kappa \geqslant 0 \\ &\lambda (v \cdot s + v_0 - \kappa) = 0. \end{aligned} \tag{180}$$

Differentiating,

$$v_0 = t_0 + \lambda \tag{181}$$

$$v = t + \lambda s \tag{182}$$

$$s = \arg\min (v \cdot s). \tag{183}$$

Let us now define $\vec{T} := (t_0, t), \vec{V} := (v_0, v)$, and $\vec{S} := (1, s)$. The KKT conditions that characterize the unique solution of $\vec{V}$ for $F(\vec{T})$ are given by

$$\vec{V} = \vec{T} + \lambda \tilde{S}\left(\vec{T}\right) \tag{184}$$

where $\tilde{S}(\vec{T})$ is a point on the convex hull of S with minimal overlap with $\vec{V}$.

The scale factor λ is either zero or positive, corresponding to whether $v \cdot s + v_0 - \kappa$ is positive or zero. If $v \cdot s + v_0 - \kappa > 0$, then $\lambda = 0$, meaning that $\vec{V} = \vec{T}$. If $v \cdot s + v_0 - \kappa = 0$ then $\lambda > 0$ and $\vec{V} \neq \vec{T}$. Then, we get the self-consistent equation for the parameter λ

$$\lambda = \frac{\left[-\vec{T} \cdot \tilde{S}\left(\vec{T}\right) + \kappa\right]_+}{\left\|\tilde{S}\left(\vec{T}\right)\right\|^2}. \tag{185}$$

Remark 1. Here, v represents the projection of the weight vector on a given manifold, S is a support vector of this manifold (anchor point), and λ is the support vector coefficient. Thus, λS is the contribution to V from the support vector of this manifold, and T is the random contribution from other manifolds.

The average of T reflects the variation in the position of the anchor point of a given manifold when the weight vector changes due to changes in the environment via resampled orientation and position or labeling.

5.6. Balls

We can study the capacity of a simple setting where the object manifolds are Euclidean balls of radius R in dimension D,

J. Stat. Mech. (2024) 104007

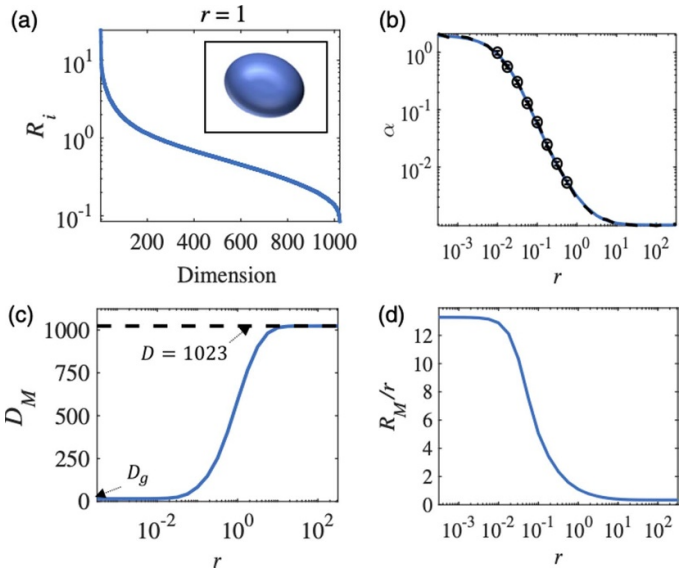

Figure 10. Linear classification of ellipsoids, with radii computed from realistic image data (see [11] for details). Reprinted figure with permission from [11], Copyright (2018) by the American Physical Society.

$$\alpha_{\text{Ball}}^{-1}(R,D) = \int_0^\infty \mathrm{d}t \chi_D(t) \int_{\kappa - tR^{-1}}^{\kappa + tR} \mathrm{D}t_0 \frac{(-t_0 + tR + \kappa)^2}{(1+R^2)} + \int_0^\infty \mathrm{d}t \chi_D(t) \int_{-\infty}^{\kappa - tR^{-1}} \mathrm{D}t_0 \left[(t_0 - \kappa)^2 + t^2\right] \tag{186}$$

where,

$$\chi_D(t) = \frac{2^{1-\frac{D}{2}}}{\Gamma\left(\frac{D}{2}\right)} t^{D-1} e^{-\frac{1}{2}t^2},\ t \geqslant 0 \tag{187}$$

is the D-dimensional Chi probability density function.

For large D and R,

$$\alpha_{\text{Ball}}(R_M, D_M) \approx \frac{1 + R_M^{-2}}{D_M}. \tag{188}$$

This calculation is generalized to ellipsoids in figure 10.

5.7. Manifold anchor geometry

Calculating the contributions of the different regimes to the capacity of general manifolds is hard. However, it turns out that when the dimensionality of the manifold is large, their capacity is similar to that of balls with radius and dimensions equivalent to an appropriately defined manifold's effective radius and dimension. To find out these

J. Stat. Mech. (2024) 104007

effective geometric measures, we note that the mean-field theory defines a new measure on the manifolds or, more precisely, on their convex hull.

For a fixed manifold, the theory specifies the position of the anchor point, $s(t_0,t)$, as a function of the $D+1$ dimensional Gaussian vector $T=(t_0,t)$. Thus, as the position/orientation/labels of the 'environment manifolds' vary, the anchor point varies, yielding a measure on the manifold, from which we can calculate several statistics. In particular, we can define

$$R_M^2 = \langle ||s(t_0,t)||^2 \rangle_{t_0,t} \tag{189}$$

where

$$s(t_0,t) = \arg\min_{\vec{s}\in S} v(t_0,t)\cdot s \tag{190}$$

where the average is over (t_0,t). We can of course generalize it to calculate the set of D manifold radii through the principal components of the manifold covariance matrix

$$C_M = \langle ss^T \rangle_{t_0,t}. \tag{191}$$

The dimensionality measures the angular spread of the anchor points in different directions. This is captured by

$$D_M = \langle (\hat{s}\cdot t)^2 \rangle_{t_0,t}. \tag{192}$$

5.8. Separating of general manifolds in high dimensions

As hinted above, we have shown that the capacity of manifolds with large D_M is well approximated by

$$\alpha_M \approx \alpha_{\text{Ball}}(R_M, D_M) \approx \frac{1+R_M^{-2}}{D_M}. \tag{193}$$

Furthermore, in high dimensions, there are two main regimes that behave qualitatively differently in both geometry and capacity.

The scaling regime

$$R_M\sqrt{D}_M < 1. \tag{194}$$

In this regime, the capacity is of order 1 . We call it a scaling regime because it behaves linearly when we scale the manifolds by a global factor r . Naively, we would expect that the radius will scale linearly with r and that the dimensionality will be invariant to it. In our case, this is true only in the scaling regime. Here, most of the time the manifolds are either interior or touching; therefore, the measures only explore the boundaries of the manifolds and the anchor points scale linearly with r, D_M is independent of r and, quite often, $D_M \ll D$ in this regime.

J. Stat. Mech. (2024) 104007

Nonlinear regime:

$$R_M\sqrt{D_M} > 1. \tag{195}$$

Here, R_M increases sublinearly with r and D_M increases, sometimes dramatically. And the capacity is

$$\alpha_M \approx \frac{1}{D_M}. \tag{196}$$

The reason for this counterintuitive behavior is that our geometry *is a collective property* of an ensemble of manifolds, which together specify the separating (or margin planes). Thus, when the sizes of the manifolds are increased by a large factor r, they become harder to separate. Hence, more of them become support manifolds with large intersections with the margin planes. This implies that anchor points tend to be in the interior of the manifolds and increasingly more dimensions are 'explored'. In the limit of $r \to \infty$, $D_M \to D$ and $\alpha \to D^{-1}$.

The evolution of these quantities along the layers of pre-trained deep neural networks is shown in figure 11.

6. Lecture 5: manifold representations in DNNs 2: generalization and few-shot learning

In the final lecture we turn to *generalization*, and we study the learning of novel concepts. Within our framework, we will assume that each new concept is represented by an underlying manifold. Our task will be to learn to classify this novel concept on the basis of only a few training examples drawn randomly from the manifold. In this final lecture we will:

1. Calculate the generalization error as a function of the geometry of the manifold, and the number of samples we are given.
2. Address the question of *few-shot* learning: under what conditions are a few training examples enough?

Unlike classification, few-shot learning depends on the density, or measure, of examples on the manifold. We will assume a uniform distribution of examples on the manifold.

6.1. Separating a manifold from the origin by a single example

6.1.1. D-dimensional sphere. Before proceeding to the full theory, we begin by studying a simplified setting, which highlights some of the interesting behavior of few-shot learning in high dimensions. We model learning a novel concept as learning to separate a single manifold from the origin. We will further assume this manifold to be a sphere living in a subspace $\{u_i\}_{i=1,\ldots,D}$ orthogonal to its centroid, x_0. We now draw a single

J. Stat. Mech. (2024) 104007

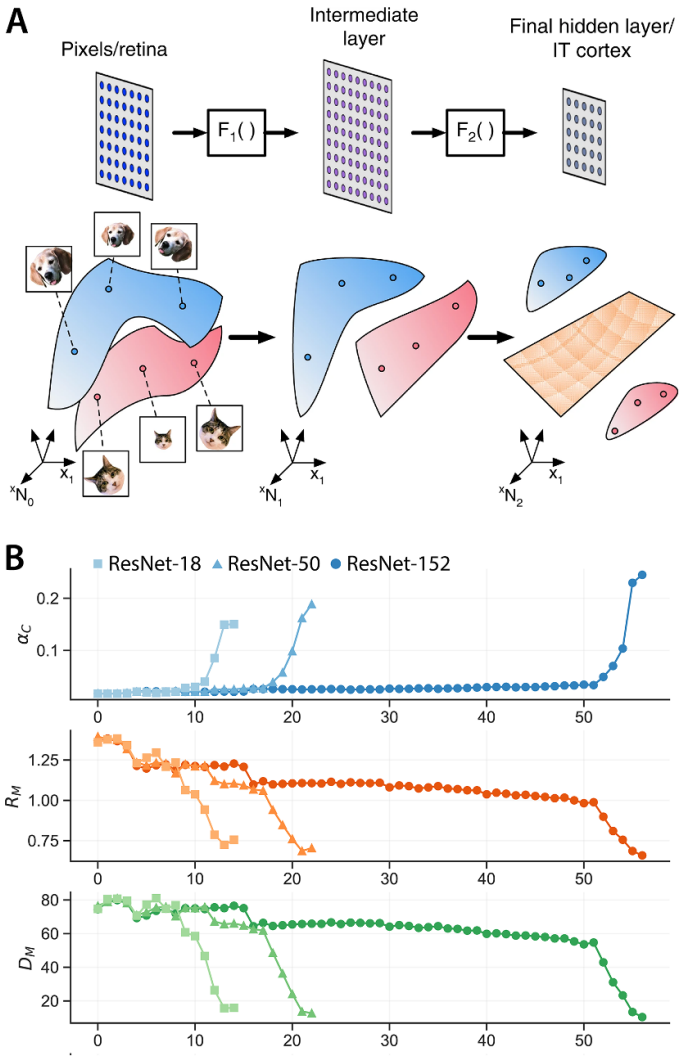

Figure 11. Geometry and separability of object manifolds in deep neural networks. (A) Illustration of three layers in a visual hierarchy, where the population response of the first layer is mapped into the intermediate layer by F1 and into the last layer by F2 (top). The transformation of per-stimuli responses is associated with changes in the geometry of the object manifold: the collection of responses to stimuli of the same object (colored blue for a 'dog' manifold and pink for a 'cat' manifold). Changes in geometry may result in transforming object manifolds that are not linearly separable (in the first and intermediate layers) into separable ones in the last layer (separating hyperplane, colored orange). (B) Changes in classification capacity α_C, manifold radius R_M, and manifold dimension D_M across the layers of pretrained DNNs (ResNets). See [12] for details. Reproduced from [12], with permission from Springer Nature.

training example x from this manifold. We can always define u_1 to be the direction of the example,

$$x = x_0 + Ru_1. \tag{197}$$

We then learn a linear classifier w on the basis of this single training example,

$$w = x. \tag{198}$$

To evaluate the generalization error, we draw a test example,

$$y = x_0 + R\sum_{i=1}^{D} s_i\, u_i. \tag{199}$$

The projection of this test example onto the linear classifier w is given by,

$$w \cdot y = (x_0 + Ru_1) \cdot \left(x_0 + R\sum_{i=1}^{D} s_i u_i \right) \tag{200}$$

$$\approx 1 + R^2 s_1. \tag{201}$$

In high dimensions D, the s_i are Gaussian with variance $1/D$ so

$$\varepsilon = H\left(\sqrt{D/R^4}\right). \tag{202}$$

In fact, R is the radius relative to the distance from the center, so we can write

$$\varepsilon = H\left(SNR\right) \tag{203}$$

where,

$$SNR = \frac{\|\Delta x_0\|^2}{D^{-1}} \tag{204}$$

where the distance Δx_0 is normalized by R. This expression reveals that the SNR *increases* with the dimension of the manifold. Hence, *high-dimensional* manifolds are better suited for generalization, unlike for classification (memorization), where low-dimensional manifolds are preferred. The reason for this arises from the fact that in high dimensions the alignment between the training examples and the non-signal directions of the manifold is suppressed, reducing variability, which can lead to increased generalization error.

Note: in fact, s is not Gaussian but bounded. The worst case would be

$$y = x_0 - Ru_1 \tag{205}$$

for which the overlap is

$$1 - R^2. \tag{206}$$

Hence, for $R < 1$ the error is zero. So the correct expression is

$$\varepsilon \approx H(\sqrt{D/(R^4 - 1)}). \tag{207}$$

However, when $R > 1$ this non-Gaussianity can be neglected.

J. Stat. Mech. (2024) 104007

6.1.2. D-dimensional ellipsoid. We now extend our analysis to the slightly more complex setting of classifying an ellipsoid with radii $\{R_i\}_{i=1,\ldots,D}$

$$X = x_0 + \sum_{i=1}^{D} s_i R_i u_i. \tag{208}$$

We find a very similar result for the generalization error to the case of spheres (see [13] for details), except that the dimension D is replaced with an 'effective' dimension D_{eff},

$$\varepsilon \approx H\left(SNR\right) \tag{209}$$

$$SNR = \frac{\|\Delta x_0\|^2}{\sqrt{D_{\text{eff}}^{-1}}} \tag{210}$$

where,

$$D_{\text{eff}} = \frac{\left(\sum_{i=1}^{D} R_i^2\right)^2}{\sum_{i=1}^{D} R_i^4} = \text{participation ratio}. \tag{211}$$

The quantity D_{eff} which emerges from the calculation is a well-known measure of dimensionality called the participation ratio, which captures the 'effective' number of dimensions the manifold explores. Hence, the generalization error is governed by simple properties of the geometry of the manifold: its radius and dimension. And, as we saw for spheres, high-dimensional ellipsoids are better for one-shot generalization.

6.2. M-shot learning of pairs of general manifolds

We now consider the problem of classifying *two* novel concepts. We further extend our analysis from the one-shot learning problem to the few-shot learning problem, where we are given a few training examples from each manifold, $\{x_\mu^a\}_{\mu=1,\ldots,m}$, $\{x_\mu^b\}_{\mu=1,\ldots,m}$.

Prototype learning:
Many decision rules are possible to classify the training examples (e.g. SVMs or k-nearest-neighbors), but in the high-dimensional few-shot setting a particularly simple decision rule, called 'prototype learning', tends to work best. Prototype learning works by constructing an estimate of each manifold's centroid (a 'prototype'),

$$\hat{x}_a = \frac{1}{m}\sum_{\mu=1}^{m} x_\mu^a \tag{212}$$

$$\hat{x}_b = \frac{1}{m}\sum_{\mu=1}^{m} x_\mu^b. \tag{213}$$

J. Stat. Mech. (2024) 104007

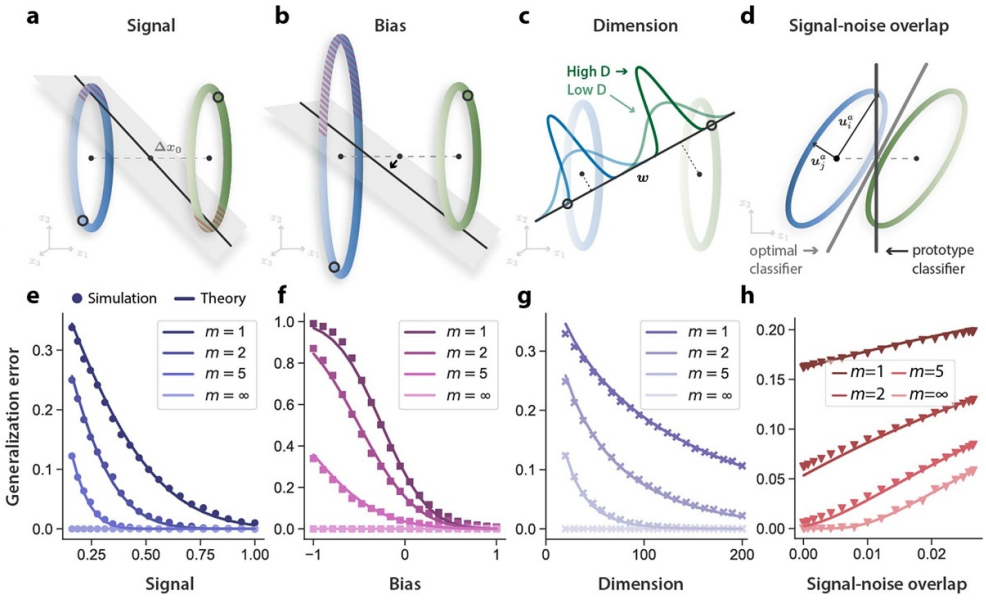

Figure 12. Geometric theory of few-shot learning (see [13] for details). Reproduced with permission from [13].

And positioning a linear decision boundary halfway between the centroids,

$$w = \hat{x}_a - \hat{x}_a. \tag{214}$$

Results:
The calculation for the generalization error for pairs of ellipsoids is considerably more involved, so we leave it to [13]. The result depends on each manifold's centroid $\boldsymbol{x_0}$, and radii R_i along a set of orthonormal basis directions $\boldsymbol{u}_i, i = 1, \ldots, N$, capturing the extent of natural variation of examples belonging to the same object. A useful measure of the overall size of these variations is the mean-squared radius $R^2 \equiv \frac{1}{N}\sum_{i=1}^{N} R_i^2$.

The probability of error on a test point on manifold a is given by,

$$\varepsilon_a \approx H\left(SNR\right) \tag{215}$$

$$SNR = \frac{1}{2}\frac{\|\Delta x_0\|^2 + \left(R_b^2 R_a^{-2} - 1\right)/m}{\sqrt{D_a^{-1}/m} + \|\Delta x_0 \cdot U_b\|^2/m + \|\Delta x_0 \cdot U_a\|^2}. \tag{216}$$

The SNR depends on four interpretable geometric properties:

(1) **Signal**. $\|\boldsymbol{\Delta x_0}\|^2 \equiv \|\boldsymbol{x_0^a} - \boldsymbol{x_0^b}\|^2/R_a^2$ represents the pairwise distance between the manifolds' centroids, $\boldsymbol{x_0}^a$ and $\boldsymbol{x_0}^b$, normalized by R_a^2. As the pairwise distance between manifolds increases, they become easier to separate, leading to higher SNR and lower error. We denote $\|\boldsymbol{\Delta x_0}\|^2$ as the signal and $\boldsymbol{\Delta x_0}$ as the signal direction. The **signal** is the analog of the inverse squared manifold radius R_M studied above.

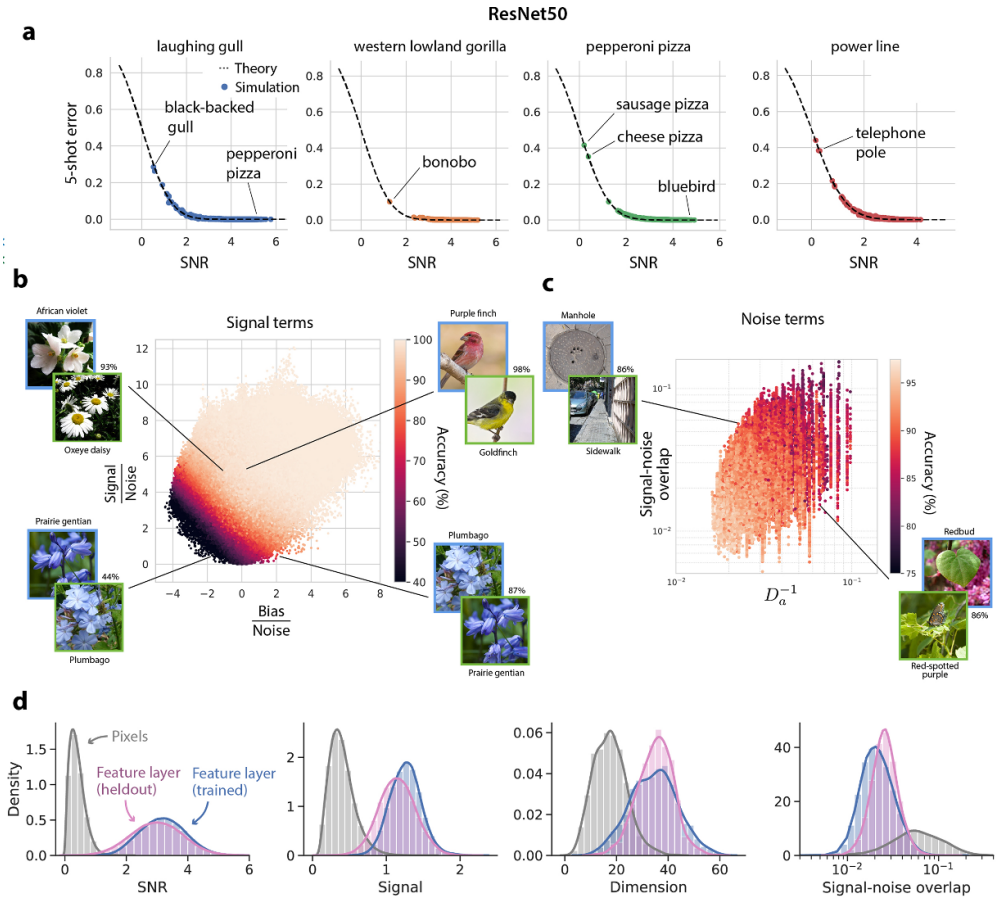

Figure 13. Theory predicts few-shot learning performance in deep convolutional neural network (DCNN)s (see [13] for details). Reproduced with permission from [13].

(2) **Bias**. $R_b^2 R_a^{-2} - 1$ represents the average bias of the linear classifier. Importantly, this bias is asymmetric: when manifold a is larger than manifold b, the bias term is negative, predicting a lower SNR for manifold a.

(3) **Dimension**. In our theory, $D_a \equiv (R_a^2)^2 / \sum_{i=1}^{N} (R_i^a)^4$, known as the *participation ratio*, quantifies the number of dimensions along which the object manifold varies significantly, which is often much smaller than the number of neurons N. This is the analog of the manifold dimension D_M studied above. Intriguingly, in contrast to the role of dimensionality for capacity discussed above, for few-shot learning high-dimensional manifolds are preferred. This enhanced performance is due to the fact that few-shot learning involves comparing a test example to the training examples of each novel object. In low dimensions, the distance from the test example to each of the training examples varies significantly, contributing a noise term to the SNR. However, in high dimensions these distances concentrate around their typical value, and hence the noise term is suppressed as D_a^{-1}. Note that this benefit of high-dimensional representations is unique to few-shot learning, since the noise term can

J. Stat. Mech. (2024) 104007

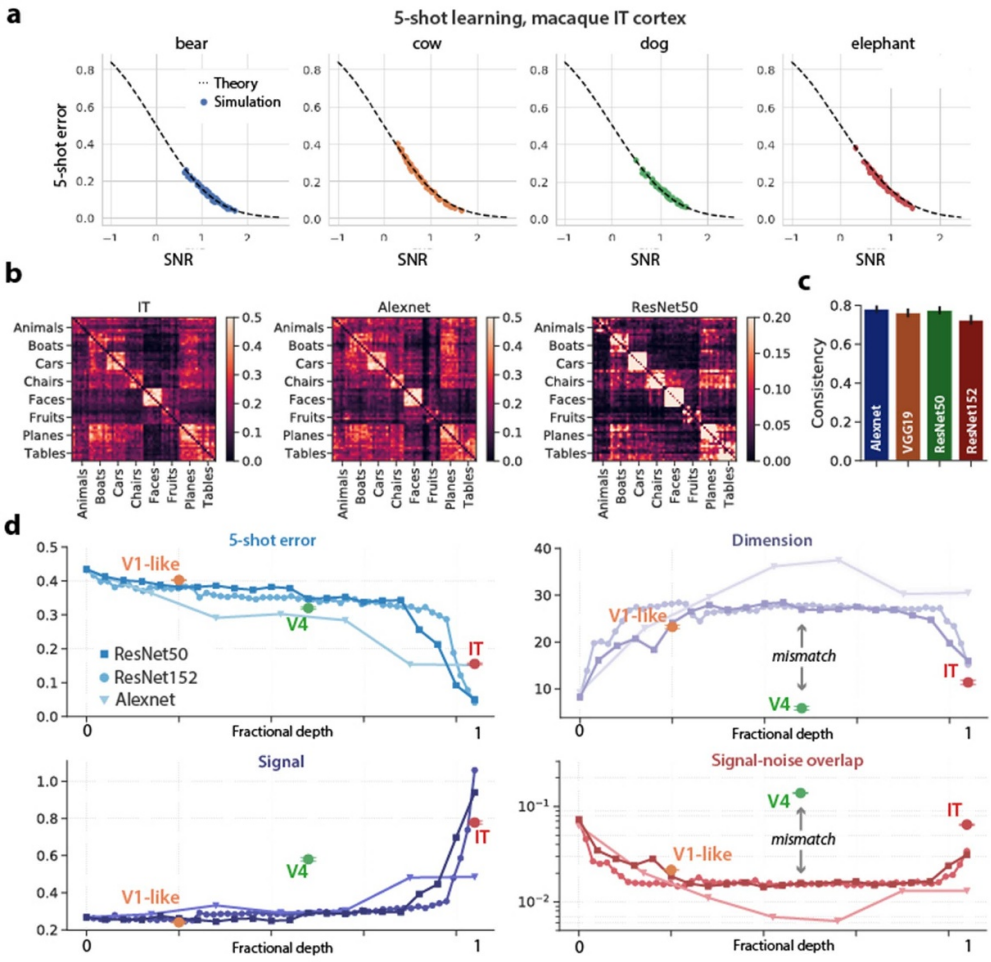

Figure 14. Theory predicts few-shot learning performance in macaque IT cortex (see [13] for details). Reproduced with permission from [13].

also be suppressed by averaging over many training examples (hence, the factor of $1/m$ multiplying inverse dimension).

(4) **Signal–noise overlap**. $\|\Delta \boldsymbol{x_0} \cdot \boldsymbol{U}_a\|^2$ and $\|\Delta \boldsymbol{x_0} \cdot \boldsymbol{U}_b\|^2$ quantify the overlap between the signal direction $\Delta \boldsymbol{x_0}$ and the manifold axes of variation $\boldsymbol{U}_a \equiv [\boldsymbol{u_1^a} R_1^a, \ldots, \boldsymbol{u_N^a} R_N^a]/\sqrt{R_a^2}$ and $\boldsymbol{U}_b \equiv [\boldsymbol{u_1^b} R_1^b, \ldots, \boldsymbol{u_N^b} R_N^b]/\sqrt{R_b^2}$. The generalization error increases as the overlap between the signal and noise directions increases. We note that signal–noise overlap is bounded above by $1/D_a$, and hence is small in high dimensions.

To validate our theory, we conducted experiments on visual object manifolds from pretrained DNNs (ResNet50) and primate inferior temporal (IT) cortex neural activity, finding agreement across visual categories (figures 13 and 14).

J. Stat. Mech. (2024) 104007

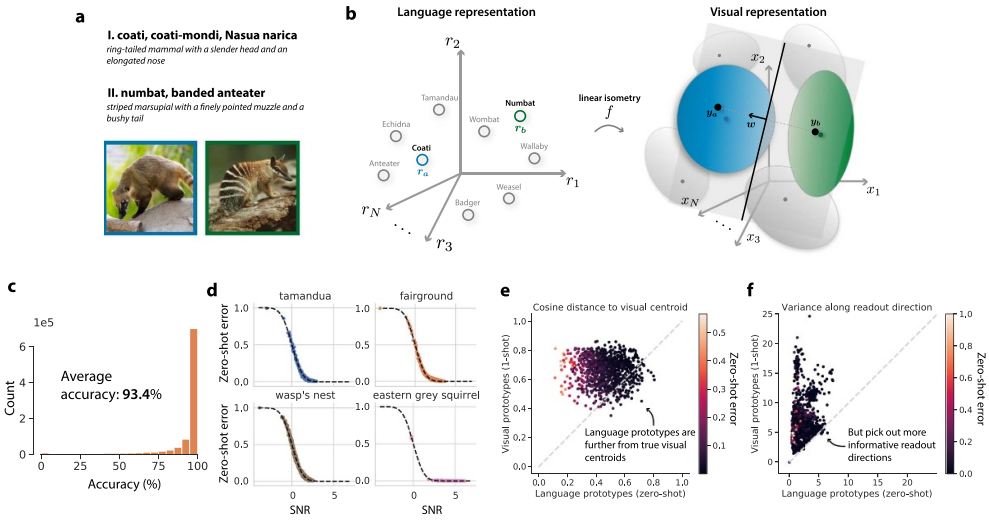

Figure 15. Alignment of visual and language representations for zero-shot learning (see [13] for details). Reproduced with permission from [13].

6.3. Applications to DCNNs and cortical data

This theory can be used to predict the generalization error of few-shot learning experiments on synthetic manifolds (figure 12), deep neural networks trained on visual object recognition tasks (e.g. a ResNet50 on ImageNet, figure 13), and even neural activity patterns recorded from the IT cortex of macaques performing object recognition tasks (figure 14). Intriguingly, the geometry of manifolds in DNNs and in the primate cortex share some universal properties but also exhibit striking differences: in particular, manifolds in intermediate layers of DNNs are over an order of magnitude higher-dimensional than manifolds in intermediate layers of the primate visual cortex.

6.4. Alignment of visual and language representations for zero-shot learning

Humans are also capable of learning new visual concepts based only on language descriptors, without any explicit visual training examples (zero-shot learning). By studying representations of the language descriptors of the visual concepts considered above, using a simple word2vec language encoding model, we find a remarkable degree of alignment between visual representations and language representations (figure 15). Indeed, these representations can be closely aligned via a simple learned linear isometry. Moreover, this alignment generalizes to novel concepts, affording good zero-shot learning performance without any further updates to the representations. We extend our theory to predict the generalization error of few-shot learning based on the geometry of these representations, and their alignment, in [13].

J. Stat. Mech. (2024) 104007

7. Discussion

Because of the staggering complexity of neural circuits in the brain across many scales, a natural and fundamental question faced by neuroscience is: what are the relevant coarse-grained order parameters to study in order to build a theory of neural computation? Deep learning faces the same question, where neural networks today are trained with billions or trillions of parameters to perform extraordinarily intricate tasks. Which order parameters can we measure to begin to understand the computations performed by these neural circuits? The *geometry* of neural representations offers one promising such set of order parameters. As we have seen, both DNNs and sensory hierarchies reformat manifold geometry in such a way that is useful for downstream computations by simple neural circuits. The compact object manifolds which emerge as a result of this formatting enable rapid learning of novel objects and concepts by simple decision rules, such as prototype learning. Furthermore, manifolds in deep neural networks and in the brain share intriguing similarities, while also exhibiting striking differences. Understanding which microscopic features of neural circuits give rise to these macroscopic geometric quantities, why they are sometimes similar and sometimes different in the brain and in deep networks, and what computational role they play, are central questions for neuroscience and deep learning in the future.

Acknowledgments

These are notes from the lectures of Haim Sompolinsky given at the summer school 'Statistical Physics and Machine Learning', that took place in Les Houches School of Physics in France from 4th to 29th July 2022. The school was organized by Florent Krzakala and Lenka Zdeborová from EPFL.

References

[1] Li Q and Sompolinsky H 2021 Statistical mechanics of deep linear neural networks: the backpropagating kernel renormalization *Phys. Rev.* X **11** 031059
[2] Seung H S, Sompolinsky H and Tishby N 1992 Statistical mechanics of learning from examples *Phys. Rev.* A **45** 6056
[3] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735
[4] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 Learning phrase representations using RNN encoder-decoder for statistical machine translation (arXiv:1406.1078)
[5] Fiat J, Malach E and Shalev-Shwartz S 2019 Decoupling gating from linearity *CoRR* (arXiv:1906.05032)
[6] Budden D, Marblestone A, Sezener E, Lattimore T, Wayne G and Veness J 2020 Gaussian gated linear networks *Advances in Neural Information Processing Systems* vol 33, ed H Larochelle, M Ranzato, R Hadsell, M Balcan and H Lin (Curran Associates, Inc) pp 16508–19
[7] Sezener E, Grabska-Barwińska A, Kostadinov D, Beau M, Krishnagopal S, Budden D, Hutter M, Veness J, Botvinick M, Clopath C, Häusser M and Latham P E 2021 A rapid and efficient learning rule for biological neural circuits *bioRxiv Preprint* (https://doi.org/10.1101/2021.03.10.434756) (posted online 12 March 2021, accessed 25 June 2022)
[8] Saxe A, Sodhani S and Lewallen S J 2022 The neural race reduction: dynamics of abstraction in gated networks *Proc. 39th Int. Conf. on Machine Learning* (*Proc. of Machine Learning Research*) vol 162, ed K Chaudhuri, S Jegelka, L Song, C Szepesvari, G Niu and S Sabato PMLR pp 19287–309

J. Stat. Mech. (2024) 104007

[9] Li Q and Sompolinsky H 2022 Globally gated deep linear networks *Advances in Neural Information Processing Systems* ***35*** p 34789
[10] Gardner E 1988 The space of interactions in neural network models *J. Phys. A: Math. Gen.* **21** 257
[11] Chung S, Lee D D and Sompolinsky H 2018 Classification and geometry of general perceptual manifolds *Phys. Rev.* X **8** 031003
[12] Cohen U, Chung S, Lee D D and Sompolinsky H 2020 Separability and geometry of object manifolds in deep neural networks *Nat. Commun.* **11** 746
[13] Sorscher B, Ganguli S and Sompolinsky H 2022 Neural representational geometry underlies few-shot concept learning *Proc. Natl Acad. Sci.* **119** e2200800119

J. Stat. Mech. (2024) 104007