Implicit Regularization via Neural Feature Alignment

Aristide Baratin[1*] **Thomas George**[1*] **César Laurent**[1] **R Devon Hjelm**[2,1]
Guillaume Lajoie[1] **Pascal Vincent**[1,3] **Simon Lacoste-Julien**[1,3]
[1] Mila, Université de Montréal [2] Microsoft Research [3] Canada CIFAR AI chair

Abstract

We approach the problem of implicit regularization in deep learning from a geometrical viewpoint. We highlight a regularization effect induced by a dynamical alignment of the neural tangent features introduced by Jacot et al. (2018), along a small number of task-relevant directions. This can be interpreted as a combined mechanism of feature selection and compression. By extrapolating a new analysis of Rademacher complexity bounds for linear models, we motivate and study a heuristic complexity measure that captures this phenomenon, in terms of sequences of tangent kernel classes along optimization paths. The code for our experiments is available as https://github.com/tfjgeorge/ntk_alignment.

1 Introduction

One important property of deep neural networks is their ability to generalize well on real data. Surprisingly, this is even true with very high-capacity networks *without explicit regularization* (Neyshabur et al., 2015; Zhang et al., 2017; Hoffer et al., 2017). This seems at odds with the usual understanding of the bias-variance trade-off (Geman et al., 1992; Neal et al., 2018; Belkin et al., 2019): highly complex models are expected to overfit the training data and perform poorly on test data (Hastie et al., 2009). Solving this apparent paradox requires understanding the various learning biases induced by the training procedure, which can act as implicit regularizers (Neyshabur et al., 2015, 2017b).

In this paper, we help clarify one such implicit regularization mechanism, by examining the evolution of the *neural tangent features* (Jacot et al., 2018) learned by the network along the optimization paths. Our results can be understood from two complementary perspectives: a *geometric* perspective – the (uncentered) covariance of the tangent features defines a metric on the function class, akin to the Fisher information metric (e.g., Amari, 2016); and a *functional* perspective – through the tangent kernel and its RKHS. In standard supervised classification settings, our main observation is a dynamical alignment of the tangent features along a small number of task-relevant directions during training. We interpret this phenomenon as a combined mechanism of *feature selection* and *compression*. The intuition motivating this work is that such a mechanism allows large models to adapt their capacity to the task, which in turn underpins their generalization abilities.

Specifically, our main contributions are as follows:

1. Through experiments with various architectures on MNIST and CIFAR10, we give empirical insights on how the tangent features and their kernel adapt to the task during training (Section 3). We observe in particular a sharp increase of the anisotropy of their spectrum early in training, as well as an increasing similarity with the class labels, as measured by *centered kernel alignment* (Cortes et al., 2012).

2. Drawing upon intuitions from linear models (Section 4.1), we argue that such a dynamical alignment acts as *implicit regularizer*. We motivate a new heuristic complexity measure which captures this phenomenon, and empirically show better correlation with generalization compared to various measures proposed in the recent literature (Section 4).

2 Preliminaries

Let $\mathcal{F}$ be a class of functions (e.g a neural network) parametrized by $\mathbf{w} \in \mathbb{R}^P$. We restrict here to *scalar*

---

*Equal contribution. Proceedings of the 24^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

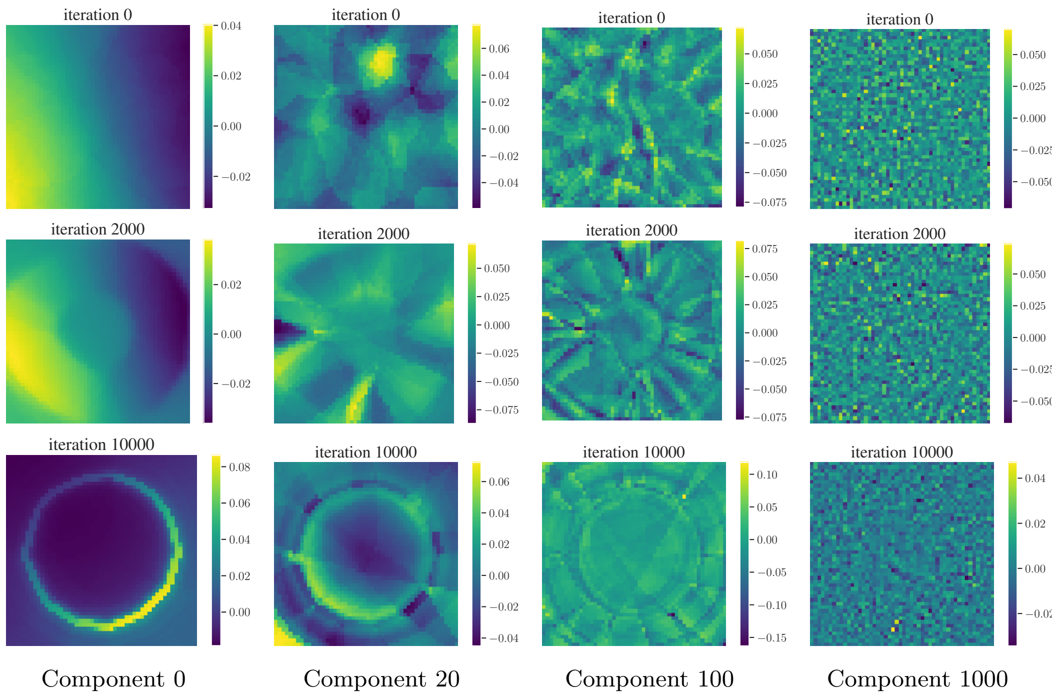

Figure 1: Evolution of eigenfunctions of the tangent kernel, ranked in nonincreasing order of the eigenvalues (**in columns**), at various iterations during training (**in rows**), for the $2d$ Disk dataset. After a number of iterations, we observe modes corresponding to the class structure (e.g. boundary circle) in the top eigenfunctions. Combined with an increasing anistropy of the spectrum (e.g $\lambda_{20}/\lambda_1 = 1.5\%$ at iteration 0, 0.2% at iteration 2000), this illustrates a stretch of the tangent kernel, hence a (soft) compression of the model, along a small number of features that are highly correlated with the classes.

functions $f_{\mathbf{w}}\colon \mathcal{X} \to \mathbb{R}$ to keep notation light.[1]

Tangent Features. We define the **tangent features** as the function gradients w.r.t the parameters,

$$\Phi_{\mathbf{w}}(\mathbf{x}) := \nabla_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x}) \in \mathbb{R}^P. \tag{1}$$

The corresponding kernel $k_{\mathbf{w}}(\mathbf{x}, \tilde{\mathbf{x}}) = \langle \Phi_{\mathbf{w}}(\mathbf{x}), \Phi_{\mathbf{w}}(\tilde{\mathbf{x}}) \rangle$ is the **tangent kernel** (Jacot et al., 2018). Intuitively, the tangent features govern how small changes in parameter affect the function's outputs,

$$\delta f_{\mathbf{w}}(\mathbf{x}) = \langle \delta \mathbf{w}, \Phi_{\mathbf{w}}(\mathbf{x}) \rangle + O(\|\delta \mathbf{w}\|^2). \tag{2}$$

More formally, the (uncentered) covariance matrix $g_{\mathbf{w}} = \mathbb{E}_{\mathbf{x}\sim\rho}\left[\Phi_{\mathbf{w}}(\mathbf{x})\Phi_{\mathbf{w}}(\mathbf{x})^\top\right]$ w.r.t the input distribution ρ acts as a **metric tensor** on $\mathcal{F}$: assuming $\mathcal{F} \subset L^2(\rho)$, this is the metric induced on $\mathcal{F}$ by pullback of the L^2 scalar product. It characterizes the geometry of the function class $\mathcal{F}$. Metric (as symmetric $P \times P$ matrices) and tangent kernels (as rank P integral operators) share the same spectrum (see Prop 4 in Appendix A.3).

Spectral Bias. The structure of the tangent features impacts the evolution of the function during training. To formalize this, we introduce the covariance eigenvalue decomposition $g_{\mathbf{w}} = \sum_{j=1}^P \lambda_{\mathbf{w}j} \boldsymbol{v}_{\mathbf{w}j} \boldsymbol{v}_{\mathbf{w}j}^\top$, which summarizes the predominant directions in parameter space. Given n input samples $(\mathbf{x}_i)$ and $\mathbf{f}_{\mathbf{w}} \in \mathbb{R}^n$ the vector of outputs $f_{\mathbf{w}}(\mathbf{x}_i)$, consider gradient descent updates $\delta \mathbf{w}_{\mathrm{GD}} = -\eta \nabla_{\mathbf{w}} L$ for some cost function $L := L(\mathbf{f}_{\mathbf{w}})$. The following elementary result (see Appendix A.5) shows how the corresponding function updates in the linear approximation (2), $\delta f_{\mathrm{GD}}(\mathbf{x}) := \langle \delta \mathbf{w}_{\mathrm{GD}}, \Phi_{\mathbf{w}}(\mathbf{x}) \rangle$, decompose in the **eigenbasis**[2] of the tangent kernel:

$$u_{\mathbf{w}j}(\mathbf{x}) = \frac{1}{\sqrt{\lambda_{\mathbf{w}j}}} \langle \boldsymbol{v}_{\mathbf{w}j}, \Phi_{\mathbf{w}}(\mathbf{x}) \rangle \tag{3}$$

Lemma 1 (Local Spectral Bias)**.** *The function updates decompose as* $\delta f_{GD}(\mathbf{x}) = \sum_{j=1}^P \delta f_j u_{\mathbf{w}j}(\mathbf{x})$ *with*

$$\delta f_j = -\eta \lambda_{\mathbf{w}j} (\boldsymbol{u}_{\mathbf{w}j}^\top \nabla_{\mathbf{f}_{\mathbf{w}}} L), \tag{4}$$

where $\boldsymbol{u}_{\mathbf{w}j} = [u_{\mathbf{w}j}(\mathbf{x}_1), \cdots u_{\mathbf{w}j}(\mathbf{x}_n)]^\top \in \mathbb{R}^n$ *and* $\nabla_{\mathbf{f}_{\mathbf{w}}}$ *denotes the gradient w.r.t the sample outputs.*

This illustrates how, from the point of view of function space, the metric/tangent kernel eigenvalues act as a mode-specific rescaling $\eta \lambda_{\mathbf{w}j}$ of the learning rate.[3] This

[1] The extension to vector-valued functions, relevant for the multiclass classification setting, is presented in Appendix A, along with more mathematical details.

[2] The functions $(u_{\mathbf{w}j})_{j=1}^P$ form an orthonormal family in $L^2(\rho)$, i.e. $\mathbb{E}_{\mathbf{x}\sim\rho}[u_{\mathbf{w}j} u_{\mathbf{w}j'}] = \delta_{jj'}$, and yield the spectral decomposition $k_{\mathbf{w}}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{j=1}^P \lambda_{\mathbf{w}j} u_{\mathbf{w}j}(\mathbf{x}) u_{\mathbf{w}j}(\tilde{\mathbf{x}})$ of the tangent kernel as an integral operator (see Appendix A.3).

[3] Intuitively, the eigenvalue $\lambda_{\mathbf{w}j}$ can be thought of as defining a local 'learning speed' for the mode j.

is a local version of a well-known bias for linear models trained by gradient descent (e.g in linear regression, see Appendix A.5.2), which prioritizes learning functions within the top eigenspaces of the kernel. Several recent works (Bietti & Mairal, 2019; Basri et al., 2019; Yang & Salman, 2019) investigated such bias for neural networks, in *linearized* regimes where the tangent kernel remains constant during training (Jacot et al., 2018; Du et al., 2019; Allen-Zhu et al., 2019). As a simple example, for a randomly initialized MLP on 1D uniform data, Fig. 8 in Appendix A.5 shows an alignment of the tangent kernel eigenfunctions with Fourier modes of increasing frequency, in line with prior empirical observations (Rahaman et al., 2019; Xu et al., 2019) of a 'spectral bias' towards low-frequency functions.

Tangent Features Adapt to the Task. By contrast, our aim in this paper is to highlight and discuss *non-linear* effects, in the (standard) regime where the tangent features and their kernel evolve during training (e.g., Geiger et al., 2019; Woodworth et al., 2020).

As a first illustration of such effects, Fig. 1 shows visualizations of eigenfunctions of the tangent kernel (ranked in nonincreasing order of the eigenvalues), during training of a 6-layer deep 256-unit wide MLP by gradient descent of the binary cross entropy loss, on a simple classification task: $y(\mathbf{x}) = \pm 1$ depending on whether $\mathbf{x} \sim \text{Unif}[-1, 1]^2$ is in the centered disk of radius $\sqrt{2/\pi}$ (details in Appendix C.1). After a number of iterations, we observe (rotation invariant) modes corresponding to the class structure (e.g. boundary circle) showing up in the *top* eigenfunctions of the learned kernel. We also note an increasing spectrum anisotropy – for example, the ratio λ_{20}/λ_1, which is 1.5% at iteration 0, has dropped to 0.2% at iteration 2000. The interpretation is that the tangent kernel (and the metric) *stretch* along a relatively small number of directions that are highly correlated with the classes during training. We quantify and investigate this effect in more detail below.

3 Neural Feature Alignment

In this section, we study in more detail the evolution of the tangent features during training. Our main results are to highlight (i) a sharp increase of the anisotropy of their spectrum early in training; (ii) an increasing similarity with the class labels, as measured by **centered kernel alignment** (CKA) (Cristianini et al., 2002; Cortes et al., 2012). We interpret this as a combined mechanism of feature selection and model compression.

3.1 Setup

We run experiments on MNIST (LeCun et al., 2010) and CIFAR10 (Krizhevsky & Hinton, 2009) with standard MLPs, VGG (Simonyan & Zisserman, 2014) and Resnet (He et al., 2016) architectures, trained by stochastic gradient descent (SGD) with momentum, using cross-entropy loss. We use PyTorch (Paszke et al., 2019) and NNGeometry (George, 2021) for efficient evaluation of tangent kernels.

In multiclass settings, tangent kernels evaluated on n samples carry additional class indices $y \in \{1 \cdots c\}$ and thus are $nc \times nc$ matrices, $(\boldsymbol{K}_{\mathbf{w}})_{ij}^{yy'} := k_{\mathbf{w}}(\mathbf{x}_i, y; \mathbf{x}_j, y')$ (details in Appendix A.4). In all our experiments, we evaluate tangent kernels on mini-batches of size $n = 100$ from both the training set and the test set; for $c = 10$ classes, this yields kernel matrices of size 1000×1000. We report results obtained from *centered* tangent features $\Phi_{\mathbf{w}}(\mathbf{x}) \to \Phi_{\mathbf{w}}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}}\Phi_{\mathbf{w}}(\mathbf{x})$, though we obtain qualitatively similar results for uncentered features (see plots in Appendix C.2).

3.2 Spectrum Evolution

We first investigate the evolution of the tangent kernel *spectrum* for a VGG19 on CIFAR 10, trained with and without label noise (Fig. 2). The take away is an anisotropic increase of the spectrum during training. We report results for kernels evaluated on training examples (solid line) and test examples (dashed line).[4]

The first observation is a significant *increase* of the spectrum, early in training (note the log scale for the x-axis). By the time the model reaches 100% training accuracy, the maximum and average eigenvalues (Fig. 2, 2nd row) have gained more than 2 orders of magnitude.

The second observation is that this evolution is highly *anisotropic*, i.e larger eigenvalues increase faster than lower ones. This results in a (sharp) increase of spectrum anisotropy, early in training. We quantify this using a notion of **effective rank** based on spectral entropy (Roy & Vetterli, 2007). Given a kernel matrix $\boldsymbol{K}$ in $\mathbb{R}^{r \times r}$ with (strictly) positive eigenvalues $\lambda_1, \cdots, \lambda_r$, let $\mu_j = \lambda_j / \sum_{i=1}^r \lambda_j$ be the trace-normalized eigenvalues. The effective rank is defined as erank $= \exp(H(\boldsymbol{\mu}))$ where $H(\boldsymbol{\mu})$ is the Shannon entropy,

$$\text{erank} = \exp(H(\boldsymbol{\mu})), \; H(\boldsymbol{\mu}) = -\sum_{j=1}^{r} \mu_j \log(\mu_j). \quad (5)$$

This effective rank is a real number between 1 and r, upper bounded by rank$(\boldsymbol{K})$, which measures the 'uniformity' of the spectrum through the entropy. We also track the various **trace ratios**

$$T_k = \sum_{j<k} \lambda_j / \sum_{j} \lambda_j, \quad (6)$$

[4]The striking similarity of the plots for train and test kernels suggests that the spectrum of empirical tangent kernels is robust to sampling variations in our setting.

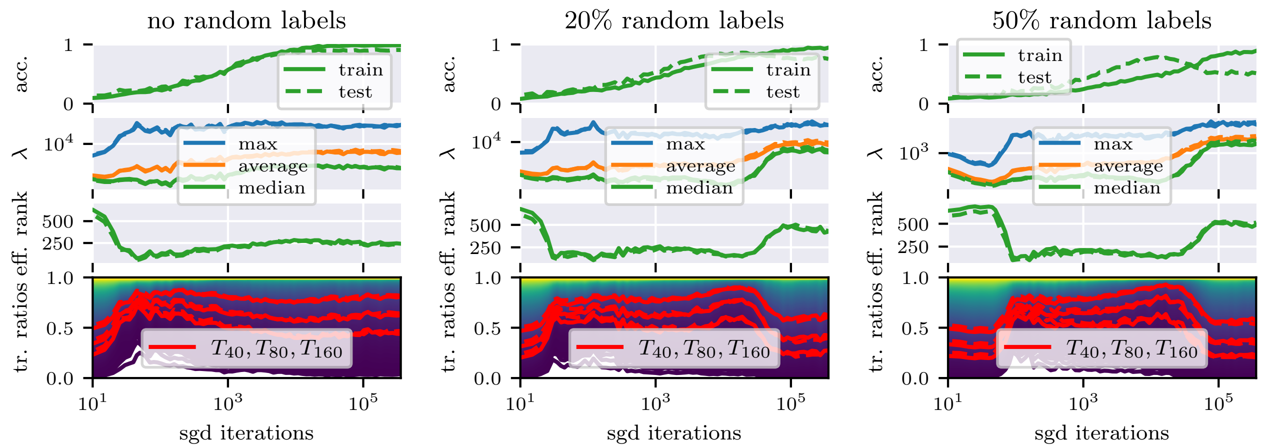

Figure 2: Evolution of the tangent kernel **spectrum** (max, average and median eigenvalues), **effective rank** (5) and **trace ratios** (6) during training of a VGG19 on CIFAR10 with various ratio of random labels, using cross-entropy and SGD with batch size 100, learning rate 0.01 and momentum 0.9. Tangent kernels are evaluated on batches of size 100 from both the training set (solid lines) and the test set (dashed lines). The plots in the top row show train/test accuracy.

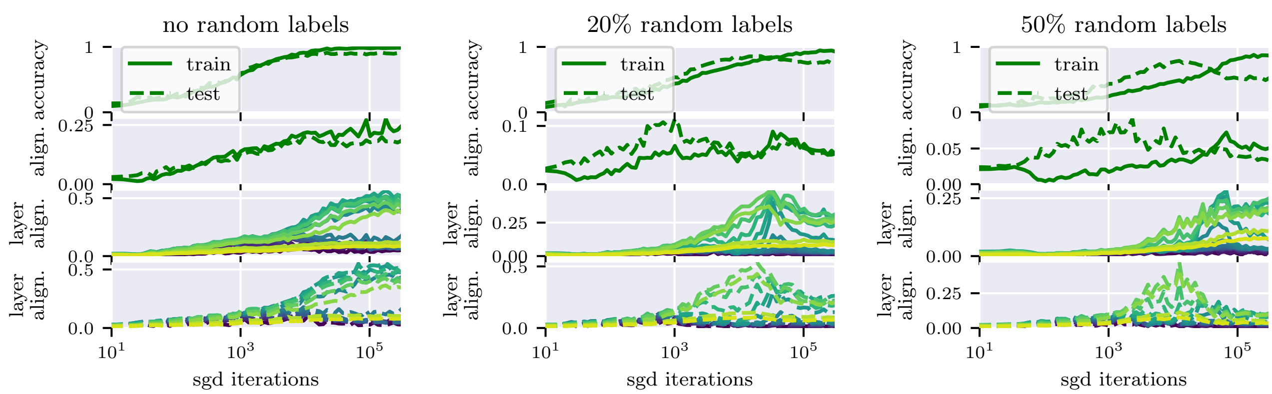

Figure 3: Evolution of the (tangent) **feature alignment with class labels** as measured by CKA (7), during training of a VGG19 on CIFAR10 (same setup as in Fig. 2). Tangent kernels and label vectors are evaluated on batches of size 100 from both the training set (solid lines) and the test set (dashed lines). The plots in the last two rows show the alignment of tangent features associated to *each layer*. Layers are mapped to colors sequentially from input layer (-), through intermediate layers (-), to output layer (-). See Fig. 11 and 13 in Appendix C for additional architectures and datasets.

which quantify the relative importance of the top k eigenvalues.

We note (Fig. 2, third row) a drop of the effective rank early in training (e.g. to less than 10% of its initial value in our experiments with no random labels; less than 20% when half of the labels are randomized). This can also be observed from the highlighted (in red) trace ratios T_{40}, T_{80} and T_{160} (Fig. 2, fourth row), e.g. the first top 40 eigenvalues (T_{40}), over 1000 in total, accounting for more than 70% of the total trace.

Remarkably, in the presence of high label noise, the effective rank of the tangent kernel (and hence that of the metric) evaluated on *training* examples (anti)-correlates nicely with the *test* accuracy: while decreasing and remaining relatively low during the learning phase (increase of test accuracy), it begins to rise again when overfitting starts (decrease of test accuracy). This suggests that this effective rank already provides a good proxy for the effective capacity of the network.

3.3 Alignment to class labels

We now include the evolution of the eigenvectors in our study. We investigate the similarity of the learned tangent features with the class label through centered kernel alignment. Given two kernel matrices $\boldsymbol{K}$ and $\boldsymbol{K}'$ in $\mathbb{R}^{r\times r}$, it is defined as (Cortes et al., 2012)

$$\mathrm{CKA}(\boldsymbol{K},\boldsymbol{K}') = \frac{\mathrm{Tr}[\boldsymbol{K}_c\boldsymbol{K}'_c]}{\|\boldsymbol{K}_c\|_F\|\boldsymbol{K}'_c\|_F} \in [0,1] \qquad (7)$$

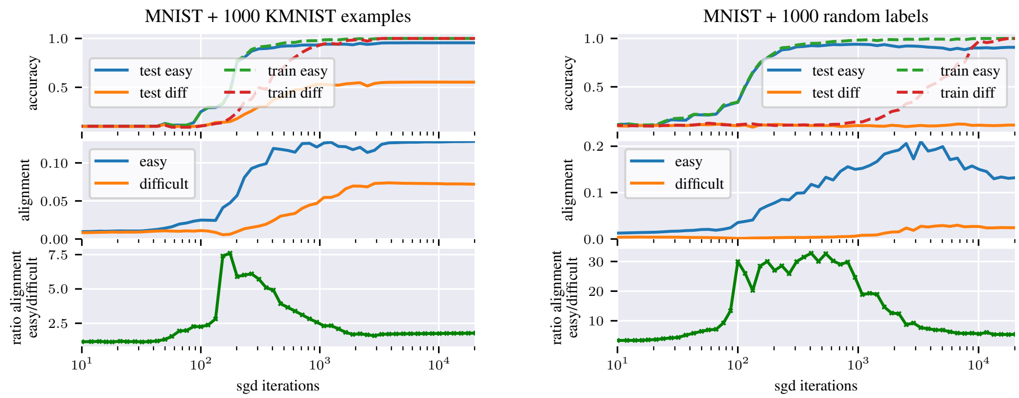

Figure 4: Alignment *easy* versus *difficult*: We augment a dataset composed of 10.000 *easy* MNIST examples with 1000 *difficult* examples from 2 different setups: **(left)** 1000 MNIST examples with random label **(right)** 1000 KMNIST examples. We train a MLP with 6 layers of 80 hidden units using SGD with learning rate=0.02, momentum=0.9 and batch size=100. We observe that the alignment to (train) labels increases faster and to a higher value for the easy examples.

where the subscript c denotes the feature centering operation, i.e. $\boldsymbol{K}_c = C\boldsymbol{K}C$ where $C = I_r - \frac{1}{r}\mathbf{1}\mathbf{1}^T$ is the centering matrix, and $\|\cdot\|_F$ is the Froebenius norm. CKA is a normalized version of the Hilbert-Schmidt Independence Criterion (Gretton et al., 2005) designed as a dependence measure for two sets of features. The normalization makes CKA invariant under isotropic rescaling.

Let $\boldsymbol{Y} \in \mathbb{R}^{nc}$ be the vector resulting from the concatenation of the one-hot label representations $\boldsymbol{Y}_i \in \mathbb{R}^c$ of the n samples. Similarity with the labels is measured through CKA with the rank-one kernel $\boldsymbol{K_Y} := \boldsymbol{Y}\boldsymbol{Y}^\top$. Intuitively, $\text{CKA}(\boldsymbol{K}, \boldsymbol{K_Y})$ is high when $\boldsymbol{K}$ has low (effective) rank and such that the angle between $\boldsymbol{Y}$ and its top eigenspaces is small.[5] Maximizing such an index has been used as a criterion for kernel selection in the literature on learning kernels (Cortes et al., 2012).

With the same setup as in Section 3.2, we observe (Fig. 3, 2nd row) an increasingly high CKA between the tangent kernel and the labels as training progresses. The trend is similar for other architectures and datasets (e.g., Fig. 11 in Appendix C shows CKA plots for MLP on MNIST and Resnets 18 on CIFAR10).

Interestingly, in the presence of high level noise, the CKA reaches a much higher value during the learning phase (increase of test accuracy) for tangent kernels and labels evaluated for *test* than for *train* inputs (note test labels are not randomized). Together with Equ. 4, this suggests a stronger learning bias towards features predictive of the *clean* labels. This is line with empirical observations that, in the presence of noise, deep networks 'learn patterns faster than noise' (Arpit et al., 2017) (see Section 3.4 below for additional insights).

[5]In the limiting case $\text{CKA}(\boldsymbol{K}, \boldsymbol{K_Y}) = 1$, the features are all aligned with each other and parallel to $\boldsymbol{Y}$.

We also report the alignments of the *layer-wise* tangent kernels. By construction, the tangent kernel, obtained by pairing features $\Phi_{w_p}(\mathbf{x})\Phi_{w_p}(\tilde{\mathbf{x}})$ and summing over all parameters w_p of the network, can also be expressed as the sum of layer-wise tangent kernels, $\boldsymbol{K}_\mathbf{w} = \sum_{\ell=1}^{L} \boldsymbol{K}_\mathbf{w}^\ell$, where $\boldsymbol{K}_\mathbf{w}^\ell$ results from summing only over parameters of the layer ℓ. We observe a high CKA, reaching more than 0.5 for a number of intermediate layers.[6] In the presence of high label noise, we note that CKAs tend to peak when the test accuracy does.

3.4 Hierarchical Alignment

A key aspect of the generalization question concerns the articulation between learning and memorization, in the presence of noise (Zhang et al., 2017) or difficult examples (e.g., Sagawa et al., 2020). Motivated by this, we would like to probe the evolution of the tangent features *separately* in the directions of both types of examples in such settings. To do so, our strategy is to measure CKA for tangent kernels and label vectors evaluated on examples from two subsets of the same size in the training dataset – one with 'easy' examples, the other with 'difficult' ones. Our setup is to augment 10.000 MNIST training examples with 1000 difficult examples of 2 types: (i) examples with random labels and (ii) examples from the dataset KMNIST (Clanuwat et al., 2018). KMNIST images present features similar to MNIST digits (grayscale handwritten characters) but represent Japanese characters.

The results are shown in Fig. 4. As training progresses, we observe that the CKA on the easy examples increases faster (and to a higher value) than that on the

[6]We were expecting to see a gradually increasing CKA with ℓ; we do not have any intuitive explanation for the relatively low alignment observed for the very top layers.

difficult ones; in the case of the (structured) difficult examples from KMNIST, we also note an increase of the CKA later in training. This demonstrates a hierarchy in the adaptation of the kernel, measured by the ratio between both alignments. From the intuition developed in the paper (see spectral bias in Equ.(4)), we interpret this aspect of the non-linear dynamics as favoring a sequentialization of learning across patterns of different complexity ('easy patterns first'), a phenomenon analogous to one pointed out in the context of deep linear networks (Saxe et al., 2014; Lampinen et al., 2018; Gidel et al., 2019).

3.5 Ablation

Effect of depth. In order to study the influence of depth on alignment and test the robustness to the choice of seeds, we reproduce the experiment of the previous section for MLP with different depths, while varying parameter initialization and minibatch sampling. Our results, shown in Fig 13 (Appendix C), suggest that the alignment effect is magnified as depth increases. We also observe that the ratio of the maximum alignment between easy and difficult examples is increased with depth, but stays high for a smaller number of iterations.

Effect of the learning rate. We observed in our experiments that increasing the learning rate tend to enhance alignment effects.[7] As an illustration, we reproduce in Fig. 14 the same plots as in Fig. 2, for a learning rate reduced to 0.003. We observe a similar drop of the effective rank as in Fig. 2 at the beginning of training, but to a much (about 3 times) higher value.

4 Measuring Complexity

In this section, drawing upon intuitions from linear models, we illustrate in a simple setting how the alignment of tangent features can act as implicit regularization. By extrapolating Rademacher complexity bounds for linear models, we also motivate a new complexity measure for neural networks and compare its correlation to generalization against various measures proposed in the literature. We refer to Appendix B for a review of classical results, further technical details, and proofs.

[7]Note that for wide enough networks and small enough learning rate, we expect to recover the linear regime where the tangent features are constant during training (Jacot et al., 2018; Du et al., 2019; Allen-Zhu et al., 2019).

4.1 Insights from Linear Models

4.1.1 Setup

We restrict here to scalar functions $f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x})\rangle$ linearly parametrized by $\mathbf{w} \in \mathbb{R}^P$. Such a function class defines a *constant* (tangent) kernel and geometry, as defined in Section 2. Given n input samples, the n features $\Phi(\mathbf{x}_i) \in \mathbb{R}^P$ yield an $n \times P$ feature matrix $\mathbf{\Phi}$.

Our discussion will be based on the (empirical) **Rademacher complexity**, which shows up in generalization bounds (Bartlett & Mendelson, 2002); see Appendix B.2 for a review. It measures how well $\mathcal{F}$ correlates with random noise on the sample set $\mathcal{S}$:

$$\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}\in\{\pm 1\}^n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(\mathbf{x}_i)\right]. \qquad (8)$$

The Rademacher complexity depends on the size (or **capacity**) of the class $\mathcal{F}$. Constraints on the capacity, such as those induced by the implicit bias of the training algorithm, can reduce the Rademacher complexity and lead to sharper generalization bounds.

A standard approach for controlling capacity is in terms of the *norm* of the weight vector – usually the ℓ_2-norm. In general, given any invertible matrix $A \in \mathbb{R}^{P\times P}$, we may consider the norm $\|\mathbf{w}\|_A := \sqrt{\mathbf{w}^\top g_A \mathbf{w}}$ induced by the metric $g_A = AA^\top$. Consider the (sub)classes of functions induced by balls of given radius:

$$\mathcal{F}^A_{M_A} = \{f_{\mathbf{w}} : \mathbf{x} \mapsto \langle \mathbf{w}, \Phi(\mathbf{x})\rangle \mid \|\mathbf{w}\|_A \le M_A\}. \qquad (9)$$

A direct extension of standard bounds for the Rademacher complexity (see Appendix B.3) yields,

$$\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{F}^A_{M_A}) \le (M_A/n)\|A^{-1}\mathbf{\Phi}^\top\|_{\mathrm{F}} \qquad (10)$$

where $\|A^{-1}\mathbf{\Phi}^\top\|_{\mathrm{F}}$ is the Froebenius norm of the *rescaled* feature matrix.[8]

This freedom in the choice of rescaling matrix A raises the question of which of the norms $\|\cdot\|_A$ provide meaningful measures of the model's capacity. Recent works (Belkin et al., 2018; Muthukumar et al., 2020) pointed out that using ℓ_2 norm is not coherently linked with generalization in practice. We discuss this issue in Appendix B.5, illustrating how meaningful norms critically depend on the geometry defined by the features.

4.1.2 Feature Alignment as Implicit Regularization

Here we describe a simple procedure making the geometry *adaptive* along optimization paths. The goal is to

[8]We also have $\|A^{-1}\mathbf{\Phi}^\top\|_{\mathrm{F}} = \sqrt{\mathrm{Tr}\boldsymbol{K}_A}$ in terms of the (rescaled) kernel matrix $\boldsymbol{K}_A = \mathbf{\Phi} g_A^{-1}\mathbf{\Phi}^\top$.

SuperNat update ($\tilde{A}_0 = \boldsymbol{I}$, $\Phi_0 = \Phi$, $\boldsymbol{K}_0 = \boldsymbol{K}$):

1. Perform gradient step $\widetilde{\mathbf{w}}_{t+1} \leftarrow \mathbf{w}_t + \delta\mathbf{w}_{\text{GD}}$
2. Find minimizer $\tilde{A}_{t+1}$ of $\|\delta\mathbf{w}_{\text{GD}}\|_{\tilde{A}} \|\tilde{A}^{-1}\boldsymbol{\Phi}_t^\top\|_{\text{F}}$
3. Reparametrize:

$$\mathbf{w}_{t+1} \leftarrow \tilde{A}_{t+1}^\top \widetilde{\mathbf{w}}_{t+1}, \Phi_{t+1} \leftarrow \tilde{A}_{t+1}^{-1}\Phi_t$$

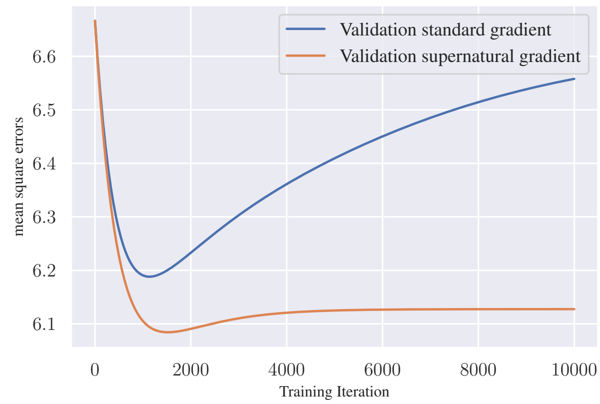

Figure 5: **(left) SuperNat** algorithm and **(right)** validation curves obtained with standard and **SuperNat** gradient descent, on the noisy linear regression problem. At each iteration, **SuperNat** identifies dominant features and stretches the kernel along them, thereby slowing down and eventually freezing the learning dynamics in the noise direction. This naturally yields better generalization than standard gradient descent on this problem.

illustrate in a simple setting how feature alignment can impact complexity and generalization, in a way that mimics the behaviour of a non-linear dynamics. The idea is to *learn* a rescaling metric at each iteration of our algorithm, using a local version of the bounds (10).

Complexity of Learning Flows. Since we are interested in functions $f_{\mathbf{w}}$ that result from an iterative algorithm, we consider functions $f_{\mathbf{w}} = \sum_t \delta f_{\mathbf{w}_t}$ written in terms of a sequence of updates[9] $\delta f_{\mathbf{w}_t}(\mathbf{x}) = \langle \delta\mathbf{w}_t, \Phi(\mathbf{x})\rangle$ (we set $f_0 = 0$ to keep the notation simple), with *local* constraints on the parameter updates:

$$\mathcal{F}_{\boldsymbol{m}}^{\boldsymbol{A}} = \{f_{\mathbf{w}} : \mathbf{x} \mapsto \textstyle\sum_t \langle \delta\mathbf{w}_t, \Phi(\mathbf{x})\rangle \mid \|\delta\mathbf{w}_t\|_{A_t} \leq m_t\} \tag{11}$$

The result (10) extends as follows.

Theorem 2 (Complexity of Learning Flows)**.** *Given any sequences* $\boldsymbol{A}$ *and* $\boldsymbol{m}$ *of invertible matrices* $A_t \in \mathbb{R}^{P\times P}$ *and positive numbers* $m_t > 0$*, we have the bound*

$$\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{F}_{\boldsymbol{m}}^{\boldsymbol{A}}) \leq \textstyle\sum_t (m_t/n)\|A_t^{-1}\boldsymbol{\Phi}^\top\|_{\text{F}}. \tag{12}$$

Note that, by linear reparametrization invariance $\mathbf{w} \mapsto A^\top\mathbf{w}$, $\Phi \mapsto A^{-1}\Phi$, the *same* result can be formulated in terms of the sequence $\boldsymbol{\Phi} = \{\Phi_t\}_t$ of feature maps $\Phi_t = A_t^{-1}\Phi$. The function class (11) can equivalently be written as

$$\mathcal{F}_{\boldsymbol{m}}^{\boldsymbol{\Phi}} = \{f_{\mathbf{w}} : \mathbf{x} \mapsto \textstyle\sum_t \langle \tilde{\delta}\mathbf{w}_t, \Phi_t(\mathbf{x})\rangle \mid \|\tilde{\delta}\mathbf{w}_t\|_2 \leq m_t\} \tag{13}$$

In this formulation, the result (12) reads:

$$\widehat{\mathcal{R}}_{\mathcal{S}}(\mathcal{F}_{\boldsymbol{m}}^{\boldsymbol{\Phi}}) \leq \textstyle\sum_t (m_t/n)\|\boldsymbol{\Phi}_t\|_{\text{F}}. \tag{14}$$

Optimizing the Feature Scaling. To obtain learning flows with lower complexity, Thm. 2 suggests modification of the algorithm to include, at each iteration t, a reparametrization step with a suitable matrix $\tilde{A}_t$ giving a low contribution to the bound (12). Applied to gradient descent (GD), this leads to a new update rule sketched in Fig. 5 (left), where the optimization in Step 2 is over a given class of reparametrization matrices. The successive reparametrizations yield a varying feature map $\Phi_t = A_t^{-1}\Phi$ where $A_t = \tilde{A}_0 \cdots \tilde{A}_t$.[10]

In the original representation Φ, **SuperNat** amounts to natural gradient descent (Amari, 1998) with respect to the local metric $g_{A_t} = A_t A_t^\top$. By construction, we also have $\delta f_{\mathbf{w}_t}(\mathbf{x}) = \langle \delta\mathbf{w}_{\text{GD}}, \Phi_t(\mathbf{x})\rangle$ where $\delta\mathbf{w}_{\text{GD}}$ are standard gradient descent updates in the linear model with feature map Φ_t.

As an example, let $\boldsymbol{\Phi} = \sum_{j=1}^n \sqrt{\lambda_j}\boldsymbol{u}_j\boldsymbol{v}_j^\top$ be the SVD of the feature matrix. We restrict to the class of matrices

$$\tilde{A}_{\boldsymbol{\nu}} = \sum_{j=1}^n \sqrt{\nu_j}\boldsymbol{v}_j\boldsymbol{v}_j^\top + \text{Id}_{\text{span}\{\boldsymbol{v}\}^\perp} \tag{15}$$

labelled by weights $\nu_j > 0, j = 1, \cdots, n$. With such a class, the action $\boldsymbol{\Phi}_t^\top \to A_{\boldsymbol{\nu}}^{-1}\boldsymbol{\Phi}_t^\top$ merely rescales the

[9] In order to not assume a specific upper bound on the number of iterations, we can think of the updates from an iterative algorithm as an infinite sequence $\{\delta\mathbf{w}_0, \cdots \delta\mathbf{w}_t, \cdots\}$ such that for some T, $\delta\mathbf{w}_t = 0$ for all $t > T$.

[10] Note that upon training a non-linear model, the updates of the tangent features take the same form $\Phi_t = \tilde{A}_t^{-1}\Phi_{t-1}$ as in Step 3 of **SuperNat**, the difference being that $\tilde{A}_t$ is now a differential operator, e.g. at first order $\tilde{A}_t = \text{Id} - \delta\mathbf{w}_t^\top \frac{\partial}{\partial \mathbf{w}_t}$.

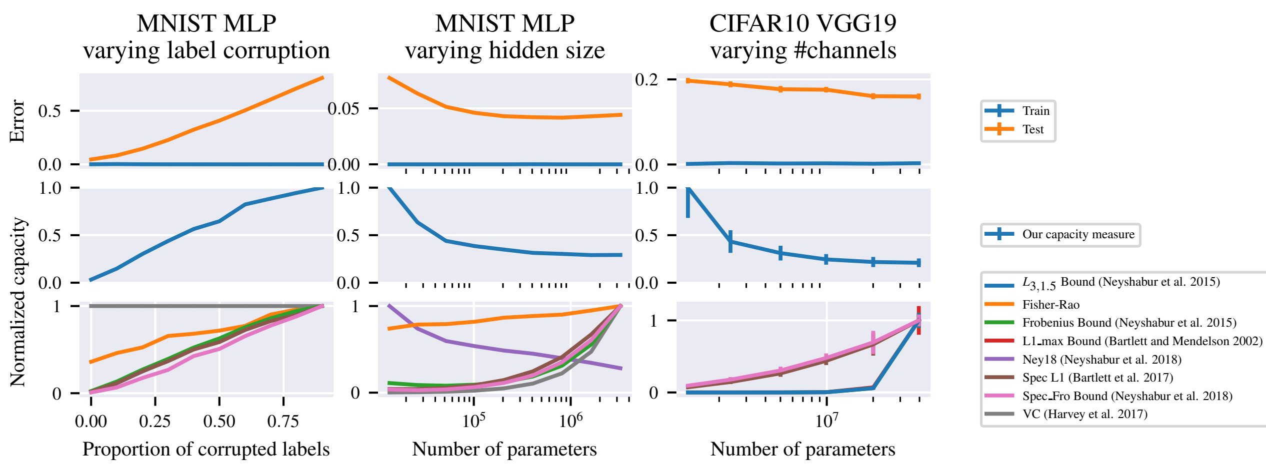

Figure 6: Complexity measures on MNIST with a 1 hidden layer MLP **(left)** as we increase the hidden layer size, **(center)** for a fixed hidden layer of 256 units as we increase label corruption and **(right)** for a VGG19 on CIFAR10 as we vary the number of channels. All networks are trained until cross-entropy reaches 0.01. Our proposed complexity measure and the one by Neyshabur et al. 2018 are the only ones to correctly reflect the shape of the generalization gap in these settings.

singular values $\lambda_{jt} \to \lambda_{jt}/\nu_j$, leaving the singular vectors unchanged. We work with gradient descent w.r.t a cost function L, so that $\delta\mathbf{w}_{\mathrm{GD}} = -\eta\nabla_{\mathbf{w}}L$.

Proposition 3. *Any minimizer in Step 2 of* **SuperNat** *over matrices $\tilde{A}_{\boldsymbol{\nu}}$ in the class (15), takes the form*

$$\nu^*_{jt} = \kappa\frac{1}{|\boldsymbol{u}_j^\top \nabla_{\mathbf{f_w}} L|} \tag{16}$$

where $\nabla_{\mathbf{f_w}}$ denotes the gradient w.r.t the sample outputs $f_{\mathbf{w}} := [f_{\mathbf{w}}(\mathbf{x}_1), \cdots f_{\mathbf{w}}(\mathbf{x}_n)]^\top$, for some constant $\kappa > 0$.

In this context, this yields the following update rule, up to isotropic rescaling, for the singular values of $\mathbf{\Phi}_t$:

$$\lambda_{j(t+1)} = |\boldsymbol{u}_j^\top \nabla_{\mathbf{f_w}} L|\lambda_{jt}. \tag{17}$$

In this illustrative setting, we see how the feature map (or kernel) adapts to the task, by stretching (resp. contracting) its geometry in directions $\boldsymbol{u}_j$ along which the residual $\nabla_{\mathbf{f_w}} L$ has large (resp. small) components. Intuitively, if a large component $|\boldsymbol{u}_j^\top \nabla_{\mathbf{f_w}} L|$ corresponds to signal and a small one $|\boldsymbol{u}_k^\top \nabla_{\mathbf{f_w}} L|$ corresponds to noise, then the ratio $\lambda_{jt}/\lambda_{kt}$ of singular values gets rescaled by the signal-to-noise ratio, thereby increasing the alignment of the learned features to the signal.

As a proof of concept, we consider the following regression setup. We consider a linear model with Gaussian features $\Phi = [\varphi, \varphi_{\mathrm{noise}}] \in \mathbb{R}^{d+1}$ where $\varphi \sim \mathcal{N}(0,1)$ and $\varphi_{\mathrm{noise}} \sim \mathcal{N}(0, \frac{1}{d}I_d)$. Given n input samples, the n features $\Phi(\mathbf{x}_i)$ yield $\boldsymbol{\varphi} \in \mathbb{R}^n$ and $\boldsymbol{\varphi}_{\mathrm{noise}} \in \mathbb{R}^{n\times d}$. We assume the label vector takes the form $\boldsymbol{y} = \boldsymbol{\varphi} + P_{\mathrm{noise}}(\boldsymbol{\epsilon})$, where Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I_n)$ is projected onto the noise features through $P_{\mathrm{noise}} = \boldsymbol{\varphi}_{\mathrm{noise}}\boldsymbol{\varphi}_{\mathrm{noise}}^\top$. The model is trained by gradient descent of the mean squared loss and its **SuperNat** variant, where Step 2 uses the analytical solution of Proposition 3. We set $d = 10, \sigma^2 = 0.1$ and use $n = 50$ training points.

Fig 5 (right) shows test error obtained with standard and **SuperNat** gradient descent on this problem. At each iteration, **SuperNat** identifies dominant features (feature selection, here φ) and stretches the metric along them, thereby slowing down and eventually freezing the dynamics in the orthogonal (noise) directions (compression). The working hypothesis in this paper, supported by the observations of Section 3, is that for neural networks, such a (tangent) feature alignment is dynamically induced as an effect of non-linearity.

4.2 A New Complexity Measure for Neural Networks

Equ. (14) provides a bound of the Rademacher complexity for the function classes (11) specified by a *fixed* sequence of feature maps (see Appendix B.4 for a generalization to the multiclass setting). By extrapolation to the case of non-deterministic sequences of feature maps, we propose using

$$\mathcal{C}(f_{\mathbf{w}}) = \sum_t \|\delta\mathbf{w}_t\|_2 \|\mathbf{\Phi}_t\|_{\mathrm{F}} \tag{18}$$

as a heuristic measure of complexity for neural networks, where $\mathbf{\Phi}_t$ is the learned tangent feature matrix[11] at training iteration t, and $\|\delta\mathbf{w}_t\|_2$ is the norm of the SGD update. Following a standard protocol for studying complexity measures, (e.g., Neyshabur et al.,

[11] In terms of tangent kernels, $\|\mathbf{\Phi}_t\|_{\mathrm{F}} = \sqrt{\mathrm{Tr}\boldsymbol{K}_t}$ where $\boldsymbol{K}_t$ is the tangent kernel (Gram) matrix.

2017a), Fig. 6 shows its behaviour for MLP on MNIST and VGG19 on CIFAR10 trained with cross entropy loss, with **(left)** fixed architecture and varying level of corruption in the labels and **(right)** varying hidden layer size/number of channels up to 4 millions parameters, against other capacity measures proposed in the recent literature. We observe that it correctly reflects the shape of the generalization gap.

5 Related Work

Role of Feature Geometry in Linear Models. Analysis of the relation between capacity and feature geometry can be traced back to early work on kernel methods (Schölkopf et al., 1999a), which lead to data-dependent error bounds in terms of the eigenvalues of the kernel Gram matrix (Schölkopf et al., 1999b).

Recently, new analysis of minimum norm interpolators and max margin solutions for overparametrized linear models emphasize the key role of feature geometry, and specifically feature anisotropy, in the generalization performance (Bartlett et al., 2019; Muthukumar et al., 2019, 2020; Xie et al., 2020). Feature anisotropy combined to a high predictive power of the dominant features is the condition for a high centered alignment between kernel and class labels. In the context of neural networks, our results highlight the role of the non linear training dynamics in favouring such conditions.

Generalization Measures. There has been a large body of work on complexity/generalization measures for neural networks (see, Jiang et al., 2020, and references therein), some of which theoretically motivated by norm or margin based bounds (e.g., Neyshabur et al., 2019; Bartlett et al., 2017). Liang et al. (2019) proposed using the Fisher-Rao norm of the solution as a geometrically invariant complexity measure. By contrast, our approach to measuring complexity takes into account the geometry along the whole optimization trajectories. Since the geometry we consider is defined through the gradient second moments, our perspectice is closely related to the notions of stiffness (Fort et al., 2019) and coherent gradients (Chatterjee, 2020).

Dynamics of Tangent Kernels. Several recent works investigated the 'feature learning' regime where neural tangent kernels evolve during training (Geiger et al., 2019; Woodworth et al., 2020). Independent concurrent works highlight alignment and compression phenomena similar to the one we study here (Kopitkov & Indelman, 2020; Paccolat et al., 2020). We offer various complementary empirical insights, and frame the alignment mechanism from the point of view of implicit regularization.

6 Conclusion

Through experiments with modern architectures, we highlighted an effect of dynamical alignment of the neural tangent features and their kernel along a small number of task-dependent directions during training, reflected by an early drop of the effective rank and an increasing similarity with the class labels, as measured by centered kernel alignment. We interpret this effect as a combined mechanism of feature selection and model compression of around dominant features.

Drawing upon intuitions from linear models, we argued that such a dynamical alignment acts as implicit regularizer. By extrapolating a new analysis of Rademacher complexity bounds for linear models, we also proposed a complexity measure that captures this phenomenon, and showed that it correlates with the generalization gap when varying the number of parameters, and when increasing the proportion of corrupted labels.

The results of this paper open several avenues for further investigation. The type of complexity measure we propose suggests new principled ways to design algorithms that learn the geometry in which to perform gradient descent (Srebro et al., 2011; Neyshabur et al., 2017b). Whether a procedure such as **SuperNat** can produce meaningful practical results for neural networks remains to be seen.

One of the consequences one can expect from the alignment effects highlighted here is to bias learning towards explaining most of the data with a small number of highly predictive features. While this feature selection ability might explain in part the performance of neural networks on a range of supervised tasks, it may also make them brittle under spurious correlation (e.g., Sagawa et al., 2020) and underpin their notorious weakness to generalize out-of-distribution (e.g., Geirhos et al., 2020). Resolving this tension is an important challenge towards building more robust models.

Acknowledgments

We thank X. Y Lu and V. Thomas for collaboration at an early stage of this project, A. Sordoni for insightful discussions, G. Gidel, A. Mitra and M. Pezeshki for helpful feedback. This research was partially supported by the Canada CIFAR AI Chair Program (held at Mila); by NSERC through the Discovery Grants RGPIN-2017-06936 (S.LJ) and RGPIN-2018-04821 (G.L) and an Alexander Graham Bell Canada Graduate Scholarship (CGS D) award (A.B); by FRQNT Young Investigator Startup Program 2019- NC-253251 (G.L); and by a Google Focused Research award (S.LJ). S.LJ and P.V are CIFAR Associate Fellows in the Learning in Machines & Brains program.

References

Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. volume 97 of *Proceedings of Machine Learning Research*, pp. 242–252, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Shun-Ichi Amari. *Information Geometry and Its Applications*, volume 194. Springer, 2016.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. *arXiv preprint arXiv:1706.05394*, 2017.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 2002.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *NIPS*, 2017.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300[stat.ML]*, 2019.

Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems 32*, pp. 4761–4771. 2019.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *ICML*, 2018.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116 (32):15849–15854, 2019.

Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, and Marie Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16 (10):2197–2219, 2004.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems 32*, pp. 12893–12904. 2019.

Mikio L Braun. *Spectral properties of the kernel matrix and their relation to kernel methods in machine learning*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2005.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2020.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. 2018.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *JMLR*, 13(1):795–828, 2012. ISSN 1532-4435.

Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz S. Kandola. On kernel-target alignment. In *NIPS*. 2002.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Stanislav Fort, Pawel Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

Mario Geiger, Stephano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *arXiv:1906.08034 [cs.LG]*, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arxiv preprint arXiv:2004.07780 [cs.CV]*, 2020.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992. doi: 10.1162/neco.1992.4.1.1.

Thomas George. NNGeometry: Easy and Fast Fisher Information Matrices and Neural Tangent Kernels in PyTorch, February 2021.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 3202–3211. Curran Associates, Inc., 2019.

Arthur Gretton, Olivier Bousquet, Alexander Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms, 2005.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause (eds.), *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1832–1841, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: Closing the generalization gap in large batch training of neural networks. In *NIPS*, 2017.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NIPS*, pp. 8571–8580. 2018.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *ICLR*, 2020.

Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Pathological spectra of the fisher information metric and its variants in deep neural networks. *arXiv:1910.05992 [stat.ML]*, 2019a.

Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. *AISTATS 2019*, 2019b.

D. Kopitkov and V. Indelman. Neural spectrum alignment: Empirical study. In *International Conference on Artificial Neural Networks (ICANN)*, September 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Andrew K Lampinen, Andrew K Lampinen, and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv.org*, 2018.

S. Lang. *Fundamentals of Differential Geometry*. Graduate Texts in Mathematics. Springer New York, 2012. ISBN 9781461205418.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer Science & Business, New York, 2013.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *Proceedings of Machine Learning Research*, volume 89, pp. 888–896, 2019.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X, 9780262018258.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *arXiv preprint arXiv:1903.09139[cs.LG]*, 2019.

Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Sahai, Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *arXiv preprint arXiv:2005.08054 [cs.LG]*, 2020.

Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, and Ioannis Mitliagkas. A modern take on the bias-variance tradeoff in neural networks. *arXiv:1810.08591 [cs.LG]*, 2018.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *ICLR workshop track*, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pp. 5949–5958, 2017a.

Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv:1705.03071 [cs.LG]*, 2017b.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *International Conference on Learning Representations (ICLR)*, 2019.

Jonas Paccolat, Leonardo Petrini, Mario Geiger, Kevin Tyloo, and Matthieu Wyart. Geometric compression of invariant manifolds in neural nets. *arXiv preprint arXiv:2007.11471*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor

Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610. IEEE, 2007.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. *arXiv:2005.04345 [cs.LG]*, 2020.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural network. In *In International Conference on Learning Representations*, 2014.

B. Schölkopf, S. Mika, C. J.C. Burges, P. Knirsch, K. R. Muller, G. Ratsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *Trans. Neur. Netw.*, 10(5):1000–1017, September 1999a. ISSN 1045-9227.

B. Schölkopf, J. Shawe-Taylor, AJ. Smola, and RC. Williamson. Kernel-dependent support vector error bounds. In *Artificial Neural Networks, 1999. ICANN 99*, volume 470 of *Conference Publications*, pp. 103–108. Max-Planck-Gesellschaft, IEEE, 1999b.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Nati Srebro, Karthik Sridharan, and Ambuj Tewari. On the universality of online mirror descent. In *Advances in Neural Information Processing Systems 24*. 2011.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. *arXiv:2002.09277 [cs.LG]*, 2020.

Yuege Xie, Rachel Ward, Holger Rauhut, and Chou Hung-Hsu. Weighted optimization: better generalization by smoother interpolation. *arXiv preprint arXiv:2006.08495*, 2020.

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In Tom Gedeon, Kok Wai Wong, and Minho Lee (eds.), *Neural Information Processing*, pp. 264–274, Cham, 2019. Springer International Publishing. ISBN 978-3-030-36708-4.

Greg Yang and Hadi Salman. A fine grained spectral perspective on neural networks. *arxiv preprint arXiv:1907.10599[cs.LG]*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.