高次元データ学習における特徴学習の優位性

Taiji Suzuki[1,2]
Joint work with Sho Okumoto[1], Jimmy Ba[3], Murat A. Erdogdu[3], Zhichao Wang[4], Denny Wu[3], Greg Yang[5]
[1]Graduate School of Information Science and Technology, the University of Tokyo,
[2]RIKEN Center for Advanced Intelligence Project
[3]University of Toronto and Vector Institute, [4]University of California, San Diego,
[5]Microsoft Research AI

1 Learnability of convolutional neural networks for infinite dimensional input

Here, we show the learning ability of convolutional neural networks for infinite dimensional input investigated in Okumoto & Suzuki (2022). First, we prepare the notations and introduce the problem setting. Throughout this section, we use the following notations. Let $\mathbb{R}_{>0} := \{s \in \mathbb{R} : s > 0\}$, and for a set $\mathbb{D}$, let $\mathbb{D}^\infty := \{(s_1, \dots, s_i, \dots) : s_i \in \mathbb{D}\}$ (for example, $\mathbb{R}^\infty := \{(s_i)_{i=1}^\infty : s_i \in \mathbb{R}\ (\forall i = 1, 2, \dots)\}$). For $s \in \mathbb{R}^\infty$, let $\mathrm{supp}((s) = \{i \in \mathbb{N} : s_i \neq 0\}$. Let $\mathbb{N}_0^\infty := \{l \in (\mathbb{N} \cup \{0\})^\infty : \mathrm{supp}(l) < \infty\}$ and define $\mathbb{Z}_0^\infty$ and $\mathbb{R}_0^\infty$ in the same way. Furthermore, for $s \in \mathbb{R}_0^\infty$, let $2^s := 2^{\sum_{i=1}^\infty s_i}$. For $L \in \mathbb{N}$, let $[L] = \{1, \dots, L\}$. For $a \in \mathbb{R}$, let $\lfloor a \rfloor$ be the largest integer less than or equal to a.

We consider a regression problem where the predictor (input) is infinite dimensional. Let λ be the uniform probability measure on $([0,1], \mathcal{B}([0,1]))$ where $\mathcal{B}([0,1])$ is the Borel σ-field on $[0,1]$, and let λ^∞ be the product measure of λ on $([0,1]^\infty, \mathcal{B}([0,1]^\infty))$ where $\mathcal{B}([0,1]^\infty)$ is the product σ-algebra generated by the cylindric sets $\cap_{j \le d}\{x \in [0,1]^\infty : x_j \in B_j\}$ for $d = 1, 2, \dots$ and $B_j \in \mathcal{B}([0,1])$. Let P_X be a probability measure defined on the measurable space $([0,1]^\infty, \mathcal{B}([0,1]^\infty))$ that is absolutely continuous to λ^∞ and its Radon-Nikodym derivative satisfies $\|\frac{\mathrm{d}P_X}{\mathrm{d}\lambda^\infty}\|_{L^\infty([0,1]^\infty)} < \infty$. Then, suppose that there exists a true function $f^\circ : [0,1]^\infty \to \mathbb{R}$, and consider the following nonparametric regression problem with an infinite dimensional input:

$$Y = f^\circ(X) + \xi, \tag{1.1}$$

where X is a random variable taking its value on $[0,1]^\infty$ and obeys the distribution P_X introduced above, and ξ is a observation noise generated from $N(0, \sigma^2)$ (a normal distribution with mean 0 and variance $\sigma^2 > 0$). Let P be the joint distribution of X and Y obeying the regression model.

What we investigate in the following is (i) how efficiently we can approximate the true function f° by a neural network, and (ii) how accurately deep learning can estimate the true function f° from n observations $D_n = (X_i, y_i)_{i=1}^n$ where $(X_i, y_i)_{i=1}^n$ are i.i.d. observations from the model. As a performance measure, we employ the mean squared error $\|f - f^\circ\|_{P_X}^2 := \mathrm{E}_P[(f(X) - f^\circ(X))^2]$, which can be seen as the excess risk of the predictive error $\mathrm{E}_{(X,Y)\sim P}[(f(X) - Y)^2]$ associated with the squared loss (i.e., $\|f - f^\circ\|_{P_X}^2 = \mathrm{E}_{(X,Y)\sim P}[(f(X) - Y)^2] - \mathrm{E}_{(X,Y)\sim P}[(f^\circ(X) - Y)^2] = \mathrm{E}_{(X,Y)\sim P}[(f(X) - Y)^2] - \inf_{f:\text{measurable}} \mathrm{E}_{(X,Y)\sim P}[(f(X) - Y)^2]$).

1.1 Mixed and anisotropic smoothness on infinite dimensional variables

Here, we introduce a function class in which we suppose the true function f° is included. For a given $l \in \mathbb{Z}_0^\infty$, define $\psi_{l_i} : [0,1] \to \mathbb{R}$ as $\psi_{l_i}(x) = \begin{cases} \sqrt{2}\cos(2\pi|l_i|x) & (l_i < 0), \\ \sqrt{2}\sin(2\pi|l_i|x) & (l_i > 0), \\ 1 & (l_i = 0), \end{cases}$ for $x \in [0,1]$, and define $\psi_l(X) := \prod_{i=1}^\infty \psi_{l_i}(x_i)$ for $X = (x_i)_{i=1}^\infty \in [0,1]^\infty$. Let $L^2([0,1]^\infty) := \{f : [0,1]^\infty \to \mathbb{R} :$

$\int_{[0,1]^\infty} f^2(x)\mathrm{d}\lambda^\infty(x) < \infty\}$ equipped with the inner product $\langle f, g\rangle := \int_{[0,1]^\infty} f(x)g(x)\mathrm{d}\lambda^\infty(x)$ for $f, g \in L^2([0,1]^\infty)$. Then, $(\psi_l)_{l\in\mathbb{Z}_0^\infty}$ forms a complete orthonormal system of $L^2([0,1]^\infty)$, that is, $f \in L^2([0,1]^\infty)$ can be expanded as $f(X) = \sum_{l\in\mathbb{Z}_0^\infty}\langle f, \psi_l\rangle\psi_l(X)$ (see Ingster & Stepanova (2011) for example). For $s \in \mathbb{N}_0^\infty$, let $\delta_s(f) : \mathbb{R}^\infty \to \mathbb{R}$ be

$$\delta_s(f)(\cdot) = \sum_{l\in\mathbb{Z}_0^\infty : \lfloor 2^{s_i-1}\rfloor \le |l_i| < 2^{s_i}} \langle f, \psi_l\rangle\psi_l(\cdot),$$

which can be seen as the frequency component of f of frequency $|l_i| \simeq 2^{s_i}$ toward each coordinate. We also define $\|f\|_p := \left(\int_{[0,1]^\infty} |f|^p \mathrm{d}\lambda^\infty\right)^{1/p}$ for $p \ge 1$. Then, we define a function space with a general smoothness configuration as follows.

Definition 1 (Function class with γ-smoothness)**.** *For a given $\gamma : \mathbb{N}_0^\infty \to \mathbb{R}_{>0}$ which is monotonically non-decreasing with respect to each coordinate. For $p \ge 1$, $\theta \ge 1$, we define the* γ-smooth space *as*

$$\mathcal{F}^\gamma_{p,\theta}([0,1]^\infty) := \left\{ f = \sum_{l\in\mathbb{Z}_0^\infty} \langle f, \psi_l\rangle\psi_l : \Big(\sum_{s\in\mathbb{N}_0^\infty} 2^{\theta\gamma(s)} \|\delta_s(f)\|_p^\theta \Big)^{1/\theta} < \infty \right\},$$

equipped with the norm $\|f\|_{\mathcal{F}^\gamma_{p,\theta}} := \left(\sum_{s\in\mathbb{N}_0^\infty} 2^{\theta\gamma(s)} \|\delta_s(f)\|_p^\theta\right)^{1/\theta}$.

In the following, $\mathcal{F}^\gamma_{p,\theta}([0,1]^\infty)$ is abbreviated to $\mathcal{F}^\gamma_{p,\theta}$, and its unit ball is denoted by $U(\mathcal{F}^\gamma_{p,\theta})$. Remind that $\delta_s(f)$ represents the frequency component associated with the frequency $(2^{s_i})_{i=1}^\infty$, and then the norm of the γ-smooth space imposes weight $2^{\theta\gamma(s)}$ on each frequency component associated with s. In that sense, $\gamma(s)$ controls the weight of each frequency component and accordingly a function in the space can have different smoothness toward different coordinates. As a special case of $\gamma(s)$, we investigate the following ones. We can see that a finite dimensional analysis can be easily reduced to a special case of the infinite dimensional analysis. In that sense, our analysis generalizes existing finite dimensional analyses.

Definition 2 (Mixed smoothness and anisotropic smoothness)**.** *Given a monotonically non-decreasing sequence $a = (a_i)_{i=1}^\infty \in \mathbb{R}_{>0}^\infty$, we define the mixed smoothness as*

(mixed smoothness) $\qquad \gamma(s) = \langle a, s\rangle,$

where $\langle a, s\rangle := \sum_{i=1}^\infty a_i s_i$[1]*, and define the anisotropic smoothness as*

(anisotropic smoothness) $\qquad \gamma(s) = \max\{a_i s_i : i \in \mathbb{N}\}.$

Each component a_i of $a = (a_i)_{i=1}^\infty$ represents the smoothness of the function with respect to the variable x_i. Since we assumed $(a_i)_{i=1}^\infty$ is monotonically non-decreasing, a function in the space has higher smoothness toward the coordinate x_i with higher index i. In other words, the function f in the space is less sensitive to the variable x_i with a larger index i. For example, in computer vision tasks, we may suppose x_i with a large index i corresponds to a higher frequency component of the input image, and then the function is less sensitive to such high frequency components and more sensitive to a low-frequency "global" information. This can be seen as an infinite dimensional variant of the mixed smooth Besov space (Schmeisser, 1987; Sickel & Ullrich, 2009) and the anisotropic Besov space (Nikol'skii, 1975; Vybiral, 2006; Triebel, 2011). In our theoretical analysis, we will assume that the true target function f^o is included in the γ-smooth function space.

Assumption 3. *The target function satisfies $f^\mathrm{o} \in U(\mathcal{F}^\gamma_{p,\theta})$ with $p \ge 1$ and $\theta \ge 1$, and $\|f^\mathrm{o}\|_\infty \le B_f$ for a fixed constant $B_f > 0$, where the smoothness γ is either the mixed smoothness or the anisotropic smoothness.*

[1]Note that, since the number of nonzero components of $s \in \mathbb{N}_0^\infty$ is finite, the summation always converges. For the same reason, the maximum in the anisotropic smoothness is also attained by some finite index i.

1.2 Definition of a dilated convolutional neural network

Here, we introduce the neural network model that we investigate. Let $L \in \mathbb{N}$ be the depth of the network and d_i $(i = 1, \dots, L+1)$ be the width of the i-th layer in the network where we set $d_{L+1} = 1$. Then, the fully connected neural network (FNN) can be given by $(A_L\eta(\cdot) + b_L) \circ \cdots \circ (A_i\eta(\cdot) + b_i) \circ \cdots \circ (A_1 x + b_1)$ where $A_i \in \mathbb{R}^{d_{i+1} \times d_i}$, $b_i \in \mathbb{R}^{d_{i+1}}$ and $\eta(x) = \max\{x, 0\}$ is the ReLU activation function that is applied element-wise. The set of FNN with depth $L \in \mathbb{N}$, maximum width $W \in \mathbb{N}$, norm bound $B > 0$, and sparsity level $S \in \mathbb{N}$ is defined by

$$\Phi(L, W, S, B) := \Big\{ f(x) = (A_L\eta(\cdot) + b_L) \circ \cdots \circ (A_i\eta(\cdot) + b_i) \circ \cdots \circ (A_1 x + b_1) \ : \\ \max_{i=1,\dots,L} \|A_i\|_\infty \vee \|b_i\|_\infty \le B, \ \sum_{i=1}^{L} \|A_i\|_0 + \|b_i\|_0 \le S, \ \max_{i=1,\dots L} d_i \le W \Big\},$$

where $\|\cdot\|_\infty$ is the maximum absolute value among the elements of a vector or matrix[2], and $\|\cdot\|_0$ is the number of non-zero elements of a vector or matrix.

Next, we define the (dilated) CNNs. Let $C \in \mathbb{N}$ be the number of channels and $\mathbb{R}^{C \times \infty} := \big\{(x_1, \dots x_i, \dots) : x_i \in \mathbb{R}^C\big\}$. Suppose that $w \in \mathbb{R}^{C \times W'}$ is a filter with a width $W' \in \mathbb{N}$, channel size $C \in \mathbb{N}$ and an interval $h \in \mathbb{N}$, then define the *dilated convolution* $w \star_h X' \in \mathbb{R}^\infty$ for an infinite-sequence of vectors $X' = (x'_{i,j})_{i=1,j=1}^{C,\infty} \in \mathbb{R}^{C \times \infty}$ as $(w \star_h X')_k = \sum_{i=1}^{C} \sum_{j=1}^{W'} w_{i,j} x'_{i,h(j-1)+k}$. When $h = 1$, it is called a normal convolution. Moreover, given a filter $F \in \mathbb{R}^{C' \times C \times W'}$ with (C')-multiple channel outputs, we define its corresponding convlution $\mathrm{Conv}_{h,F} : \mathbb{R}^{C \times \infty} \to \mathbb{R}^{C' \times \infty}$ as

$$\mathrm{Conv}_{h,F}(X') = \begin{pmatrix} F_{1,:,:} \star_h X' \\ \vdots \\ F_{C',:,:} \star_h X' \end{pmatrix}.$$

Then, the *dilated CNN* can be defined as follows.

Definition 4 (Dilated CNN)**.** *For a given L', $W' \in \mathbb{N}$, suppose that we are given filters $F_l \in \mathbb{R}^{C_{l+1} \times C_l \times W'}$ with the number of channels $C_l \in \mathbb{N}$ $(l \in [L'])$ with $C_1 = 1$ and an FNN $g_{\mathrm{FNN}} \in \Phi(L, W, B, S)$, then a neural network given by $f(X) = \Big(g_{\mathrm{FNN}} \circ \mathrm{Conv}_{W'^{L'-1}, F_{L'}} \circ \cdots \circ \mathrm{Conv}_{W'^{l-1}, F_l} \circ \cdots \circ \mathrm{Conv}_{1, F_1} \circ X\Big)_1$ is called a* dilated CNN[3]*, where g_{FNN} is assumed to be applied in an element-wise manner to the infinite sequence. The set of dilated CNNs with the same number of channels $C_l = C'$ $(2 \le \forall l \le L')$ in all layers but $C_1 = 1$ is denoted by*

$$\mathcal{P}(L', B', W', C', L, W, S, B) = \Big\{ \Big(g_{\mathrm{FNN}} \circ \mathrm{Conv}_{W'^{L'-1}, F_{L'}} \circ \cdots \circ \mathrm{Conv}_{1, F_1} \circ X\Big)_1 : \\ F_l \in \mathbb{R}^{C' \times C' \times W'} \ (l \ge 2), \ F_1 \in \mathbb{R}^{C' \times 1 \times W'}, \ \|F_l\|_\infty \le B', \ g_{\mathrm{FNN}} \in \Phi(L, W, B, S) \Big\}.$$

For simplicity, the set of dilated CNNs is abbreviated to $\mathcal{P}$ when there is no ambiguity about the parameter configuration. When $L' = 1$, it coincides with a set of regular CNNs. In our analysis, it is sufficient to consider an dilated CNN with a constant number of channels throughout all layers ($C_l = C$ $(\forall l \in [L'])$). To evaluate the estimation accuracy, it is important to assume the functions in the set is bounded in terms of the L_∞-norm. For that purpose, we consider an dilated CNN clipped by a bound $B_f > 0$ defined as $\bar{\mathcal{P}}(B_f, L', B', W', C, L, W, S, B) := \big\{\bar{f}(X) = (-B_f \vee (B_f \wedge f(X))) : f \in \mathcal{P}(L', B', W', C, L, W, S, B)\big\}$.

1.3 Approximation and estimation errors of deep learning

In this section, we give our main result about the approximation and estimation errors of FNNs and dilated CNNs when the true function f^{o} is in the γ-smooth function class. For a given $T > 0$ and the smoothness $\gamma : \mathbb{N}_0^\infty \to \mathbb{R}_{>0}$, define

$$I(T, \gamma) := \{i \in \mathbb{N} : \exists s \in \mathbb{N}_0^\infty, \ s_i \ne 0, \ \gamma(s) < T\},$$

and then the following quantities play an important role in our approximation error analysis.

[2] We define $a \vee b := \max\{a, b\}$ and $a \wedge b := \min\{a, b\}$ for $a, b \in \mathbb{R}$.

[3] Here, we employ $h = W'^{k-1}$ for the k-th layer convolution.

Definition 5 (Axial complexity and frequency direction complexity)**.** *The* axial complexity *is defined by* $d_{\max}(T,\gamma) := |I(T,\gamma)|$. *Moreover, the* frequency direction complexity *is defined by* $f_{\max}(T,\gamma) := \max_{s\in\mathbb{N}_0^\infty:\ \gamma(s)\le T} \max_{i\in\mathbb{N}} s_i$.

The axial complexity is used to evaluate how many components need to be extracted from a given infinite-dimensional sequence $X \in \mathbb{R}^\infty$ to achieve a particular approximation error, and the frequency complexity characterizes up to which frequency we require to approximate a target function with a particular error. Let

$$v := \left(\frac{1}{p} - \frac{1}{2}\right)_+,\ \alpha(\gamma) := \sup_{s\in\mathbb{N}_0^\infty} \frac{\sum_{i=1}^\infty s_i}{\gamma(s)},\ G(T,\gamma) := \sum_{s\in\mathbb{N}_0^\infty:\ \gamma(s)<T} 2^s,$$

where $(x)_+ := \max\{x, 0\}$. Then, a general approximation error theory for FNNs can be obtained as follows.

Theorem 6 (Approximation error for the γ-smooth space by FNNs)**.** *Assume that* $\gamma,\ \gamma' : \mathbb{N}_0^\infty \to \mathbb{R}_{>0}$ *satisfy*

$$\gamma'(s) < \gamma(s),\ v\alpha(\gamma) < 1,\ v\alpha(\gamma') < 1,$$

and the target function $f \in \mathcal{F}_{p,\theta}^\gamma$ $(p \ge 1,\ \theta \ge 1)$ *to be approximated satisfies* $\|f\|_\infty \le B_f$ *for a constant* $B_f \in \mathbb{R}_{>0}$. *For arbitrary* $T > 0$, *we let a tuple* $(d_{\max}, f_{\max}, G)$ *be*

$$(d_{\max}, f_{\max}, G) = \begin{cases} (d_{\max}(\gamma), f_{\max}(\gamma), G(T,\gamma)) & (1 \le \theta \le 2), \\ (d_{\max}(\gamma'), f_{\max}(\gamma'), G(T,\gamma')) & (2 < \theta), \end{cases}$$

and with some positive constants $K,\ K'$ *depending only on* B_f, *we let*

$$L = 2K \max\{d_{\max}^2, T^2, (\log G)^2, \log f_{\max}\}, \qquad W = 21 d_{\max} G,$$
$$S = 1764 K d_{\max}^2 \max\{d_{\max}^2, T^2, (\log G)^2, \log f_{\max}\} G, \qquad B = (\sqrt{2})^{d_{\max}} K'.$$

Then, there exists an FNN $\hat{R}_T \in \Phi(L, W, S, B)$ *with* $d_{\max}$*-dimensional input that takes* $(x_i)_{i\in I(T,\gamma)} \in [0,1]^{d_{\max}}$ *as an input such that* $f' : [0,1]^\infty \to \mathbb{R}$ *given by* $f'(X) := \hat{R}_T\big((x_i)_{i\in I(T,\gamma)}\big)$ *for* $X = (x_i)_{i=1}^\infty \in [0,1]^\infty$ *satisfies*

$$\|f - f'\|_2 \lesssim \begin{cases} 2^{-(1-v\alpha(\gamma))T} \|f\|_{\mathcal{F}_{p,\theta}^\gamma} & (1 \le \theta \le 2), \\ 2^{-(1-v\alpha(\gamma'))T} \left(\sum_{T\le\gamma'(s)} 2^{\frac{2\theta}{\theta-2}(\gamma'(s)-\gamma(s))}\right)^{1/2-1/\theta} \|f\|_{\mathcal{F}_{p,\theta}^\gamma} & (2 < \theta). \end{cases}$$

According to this theorem, the derived approximation error can be achieved by FNNs if the required $d_{\max}$ components of the input X is extracted. This theorem clarifies how the decay rate of the frequency components of the target function affects the approximation accuracy. Since the approximation accuracy is determined by $(d_{\max}, f_{\max}, G)$, it is not directly affected by the dimensionality but is characterized merely by the smoothness parameter γ. Intuitively, $T > 0$ controls the approximation accuracy and simultaneously controls up to which frequency is used for the approximation. Specifically, the difficulty of the approximation is determined by the number of bases required that is characterized by the number of $s \in \mathbb{N}_0^\infty$ with $\gamma(s) < T$, and the maximum frequency required for the approximation is also important for the analysis. The bound is proven by evaluating an approximation error of a trigonometric polynomial approximation of $f \in \mathcal{F}_{p,\theta}^\gamma$ and showing that we can construct a neural network that approximates a trigonometric polynomial with a certain accuracy.

Here, we derive a concrete convergence rate for CNNs in a setting where γ is mixed or anisotropic smoothness and the smoothness parameter $a = (a_i)_{i=1}^\infty$ is polynomially increasing. In this setting, we just need to use only one layer CNN.

Assumption 7. *There exists* $0 < q < \infty$ *such that the smoothness parameter* $a = (a_i)_{i=1}^\infty$ *satisfies* $a_i = \Omega(i^q)$. *We also assume* $a_1 < a_2$ *for the mixed smoothness setting.*

This assumption impose that the target function should be sufficiently smooth with respect to higher order indices. Under this setting, we show the approximation and estimation errors as follows. First, the approximation error by the CNNs can be evaluated as follows.

Theorem 8 (Approximation error bound under smoothness with polynomial order increase)**.** *Suppose that Assumptions 3 and 7 hold, then we have the following approximation error bounds:*

1. Mixed smoothness ($\gamma(s) = \langle a, s\rangle$): *Suppose that $v/a_1 < 1$. Then, for arbitrary $T > 0$, there exists a configuration of the network structure, $L' = 1,\ B' = 1,\ W' \sim T^{\frac{1}{q}},\ C' \sim T^{\frac{1}{q}}$ and*

$$L_1(T) \sim \max\left\{T^{\frac{2}{q}}, T^2\right\},\ W_1(T) \sim \left(\prod_{i=2}^{\infty}\left(1 - 2^{\frac{-(a_i - a_1)}{a_1}}\right)^{-1}\right) T^{\frac{1}{q}} 2^{\frac{T}{a_1}},$$
$$S_1(T) \sim \left(\prod_{i=2}^{\infty}\left(1 - 2^{\frac{-(a_i - a_1)}{a_1}}\right)^{-1}\right) T^{\frac{2}{q}} \max\left\{T^{\frac{2}{q}}, T^2\right\} 2^{\frac{T}{a_1}},\ B_1(T) \sim (\sqrt{2})^{T^{\frac{1}{q}}},$$

such that there exists an dilated CNN $f' \in \mathcal{P}(L', B', W', C', L_1(T), W_1(T), S_1(T), B_1(T))$ satisfying the following approximation error:

$$\|f' - f^{\circ}\|_2 \lesssim 2^{-\left(1 - \frac{v}{a_1}\right)T}.$$

2. Anisotropic smoothness ($\gamma(s) = \max_i\{a_i s_i\}$): *Let $\tilde{a} := (\sum_{i=1}^{\infty} a_i^{-1})^{-1}$ and suppose $0 < \tilde{a}$ and $v < \tilde{a}$, then there exists a network structure setting $L' = 1,\ B' = 1,\ W' \sim T^{\frac{1}{q}},\ C' \sim T^{\frac{1}{q}}$ and*

$$L_2(T) \sim \max\left\{T^{\frac{2}{q}}, T^2\right\},\ W_2(T) \sim T^{\frac{1}{q}} 2^{T/\tilde{a}},\ S_2(T) \sim T^{\frac{2}{q}} \max\left\{T^{\frac{2}{q}}, T^2\right\} 2^{T/\tilde{a}},\ B_2(T) \sim (\sqrt{2})^{T^{\frac{1}{q}}},$$

such that there exists an dilated CNN $f' \in \mathcal{P}(L', B', W', C', L_2(T), W_2(T), S_2(T), B_2(T))$ satisfying the following approximation error: $\|f' - f^{\circ}\|_2 \lesssim 2^{-(1 - v/\tilde{a})T}$.

From this theorem, we can see that the number of layers, the width, the number of parameters, and the size of the parameters are both determined by T and the smoothness parameter a. Moreover, in Theorem 6, the approximation error was derived assuming that the appropriate index set $I(T, \gamma)$ was provided. On the other hand, in Theorem 8, we do not make such an assumption because the CNNs can automatically extract the required index $I(T, \gamma)$.

Next, we consider the estimation error of these models in the regression problem (Eq. (1.1)). Suppose that we are given n observations $D_n = (X_i, y_i)_{i=1}^n$ following the model (1.1). We consider the empirical risk minimization estimator (ERM estimator) in the model $\bar{\mathcal{P}}$ that is given by any minimizer of the empirical risk:

$$\hat{f} \in \operatorname*{argmin}_{f \in \bar{\mathcal{P}}} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - y_i)^2.$$

As we have stated above, we employ the mean squared error $\|\hat{f} - f^{\circ}\|_{P_X}^2$ as a performance measure. Since $\hat{f}$ depends on the training data D_n, we take expectation with respect to D_n: $\mathrm{E}_{P^n}[\|\hat{f} - f^{\circ}\|_{P_X}^2] := \mathrm{E}_{(X_i, y_i)_{i=1}^n \sim P^n}[\|\hat{f} - f^{\circ}\|_{P_X}^2]$. Then, the following theorem holds.

Theorem 9 (Estimation error under smoothness with polynomial order increase)**.** *Suppose that Assumptions 3 and 7 hold, then we have the following estimation error bounds:*

1. Mixed smoothness ($\gamma(s) = \langle a, s\rangle$): *If $v/a_1 < 1$, then by setting the network structure as $L' = 1,\ B' = 1,\ W' \sim (\log n)^{\frac{1}{q}},\ C' \sim (\log n)^{\frac{1}{q}}$ and $(L, W, S, B) = (L_1(T), W_1(T), S_1(T), B_1(T))$ for $T = \frac{a_1}{2(a_1 - v) + 1} \log_2(n)$, the ERM estimator $\hat{f}$ in $\bar{\mathcal{P}}(B_f, L', B', W', C', L, W, S, B)$ achieves*

$$\mathrm{E}_{P^n}[\|\hat{f} - f^{\circ}\|_{P_X}^2] \lesssim \left(\prod_{i=2}^{\infty}(1 - 2^{\frac{-(a_i - a_1)}{a_1}})^{-1}\right) n^{-\frac{2(a_1 - v)}{2(a_1 - v) + 1}} (\log n)^{\frac{2}{q} + 2} \max\{(\log n)^{\frac{4}{q}}, (\log n)^4\}.$$

2. Anisotropic smoothness ($\gamma(s) = \max_i\{a_i s_i\}$): *Under the same setting, if $v < \tilde{a}$, by setting the network structure as $L' = 1,\ B' = 1,\ W' \sim (\log n)^{\frac{1}{q}},\ C' \sim (\log n)^{\frac{1}{q}}$ and $(L, W, S, B) = (L_2(T), W_2(T), S_2(T), B_2(T))$ for $T = \frac{\tilde{a}}{2(\tilde{a} - v) + 1} \log_2(n)$, the ERM estimator $\hat{f}$ achieves*

$$\mathrm{E}_{P^n}[\|\hat{f} - f^{\circ}\|_{P_X}^2] \lesssim n^{-\frac{2(\tilde{a} - v)}{2(\tilde{a} - v) + 1}} (\log n)^{\frac{2}{q} + 2} \max\{(\log n)^{\frac{4}{q}}, (\log n)^4\}.$$

This theorem shows that even if the dimension of the input data is infinite, for a function with a particular smoothness, CNNs can achieve a dimension-independent convergence rate which is a polynomial order, that is, it can avoid the curse of dimensionality by utilizing the increasing

smoothness. We can see that the derived convergence rate is a direct extension of finite dimensional one. Actually, if $v = 0$, the rate for the anisotropic smoothness matches that of the finite dimensional one (Suzuki & Nitanda, 2021) up to poly-log order which is known as minimax optimal. Therefore, CNNs can achieve the optimal rate up to poly-log order at least when $v = 0$. As for the mixed smoothness, a finite dimensional version was analyzed (Suzuki, 2019) and a similar rate was derived. However, our analysis assumes $a_1 < a_2$ and $a_i = \Omega(i^q)$ and thus obtained completely dimensionality independent bound while the bound by Suzuki (2019) depends on d in the exponent of the poly-log order.

2 High-dimensional asymptotics of feature learning

Here, we show the asymptotic analysis of predictive accuracy of two layer neural networks with feature learning developed in Ba et al. (2022). We consider the training of a fully-connected two-layer neural network (NN) with N neurons,

$$f_{\mathrm{NN}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{x}, \boldsymbol{w}_i \rangle) = \frac{1}{\sqrt{N}} \boldsymbol{a}^\top \sigma(\boldsymbol{W}^\top \boldsymbol{x}), \tag{2.1}$$

where $\boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{W} \in \mathbb{R}^{d \times N}, \boldsymbol{a} \in \mathbb{R}^N$, σ is the nonlinear activation function applied entry-wise, and the training objective is to minimize the empirical risk. Our analysis will be made in the *proportional asymptotic limit*, i.e., the number of training data n, the input dimensionality d, and the number of neurons N jointly tend to infinity. Intuitively, this regime reflects the setting where the network width and data size are comparable, which is consistent with practical choices of model scaling.

In this section, $\|\cdot\|$ denotes the ℓ_2-norm for vectors and the $\ell_2 \to \ell_2$ operator norm for matrices, and $\|\cdot\|_F$ is the Frobenius norm. For matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$, $\mathrm{tr}(\boldsymbol{M}) = \frac{1}{n}\mathrm{Tr}(\boldsymbol{M})$ is the normalized trace. $\mathcal{O}_d(\cdot)$ and $o_d(\cdot)$ stand for the standard big-O and little-o notations, where the subscript highlights the asymptotic variable; we write $\tilde{\mathcal{O}}(\cdot)$ when the (poly-)logarithmic factors are ignored. $\mathcal{O}_{d,\mathbb{P}}(\cdot)$ (resp. $o_{d,\mathbb{P}}(\cdot)$) represents big-O (resp. little-o) in probability as $d \to \infty$. $\Omega(\cdot), \Theta(\cdot)$ are defined analogously. Γ is the standard Gaussian distribution in $\mathbb{R}^d$. Given $f : \mathbb{R}^d \to \mathbb{R}$, we denote its L^p-norm w.r.t. Γ as $\|f\|_{L^p(\mathbb{R}^d,\Gamma)}$, which we abbreviate as $\|f\|_{L^p}$ when the context is clear.

2.1 Training procedure

Gradient descent on the 1st layer. Given training examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, we learn the two-layer NN (2.1) by minimizing the empirical risk: $\mathcal{L}(f) = \frac{1}{n}\sum_{i=1}^n \ell(f(\boldsymbol{x}_i), y_i)$, where ℓ is the squared loss $\ell(x, y) = \frac{1}{2}(x - y)^2$. As previously remarked, fixing the first layer $\boldsymbol{W}$ at random initialization and learning the second layer $\boldsymbol{a}$ yields an RF model, which is a convex problem with closed-form solution. In contrast, we are interested in *learning the feature map (representation)*; hence we first fix $\boldsymbol{a}$ (at initialization) and perform gradient descent on $\boldsymbol{W}$. We write the initialized first-layer as $\boldsymbol{W}_0$, and the weights after one gradient step as $\boldsymbol{W}_1$. The gradient update, which we refer to as the *feature learning step*, with learning rate η is given as: $\boldsymbol{W}_1 = \boldsymbol{W}_0 + \eta\sqrt{N} \cdot \boldsymbol{G}_0$ where

$$\boldsymbol{G}_0 := \frac{1}{n}\boldsymbol{X}^\top \left[\left(\frac{1}{\sqrt{N}} \left(\boldsymbol{y} - \frac{1}{\sqrt{N}} \sigma(\boldsymbol{X}\boldsymbol{W}_0)\boldsymbol{a} \right) \boldsymbol{a}^\top \right) \odot \sigma'(\boldsymbol{X}\boldsymbol{W}_0) \right], \tag{2.2}$$

in which $\odot$ is the Hadamard product, σ' is the derivative of σ (acting entry-wise), and we denoted the input feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, and the corresponding label vector $\boldsymbol{y} \in \mathbb{R}^n$. We remark that the $\sqrt{N}$-scaling in front of η accounts for the $\frac{1}{\sqrt{N}}$-prefactor in our definition of two-layer NN (2.1).

Ridge regression for the 2nd layer. After obtaining the updated weights $\boldsymbol{W}_1$, we evaluate the quality of the new CK features by computing the prediction risk of the *kernel ridge regression* estimator on top of the first-layer representation. Note that if ridge regression is performed on the same data $\boldsymbol{X}$, then after one feature learning step, $\boldsymbol{W}_1$ is no longer independent of $\boldsymbol{X}$, which significantly complicates the analysis. To circumvent this difficulty, we estimate the regression coefficients $\hat{\boldsymbol{a}}$ using *a new set of training data* $\{\tilde{\boldsymbol{x}}_i, \tilde{y}_i\}_{i=1}^n$, which for simplicity we assume to

have the same size as the original dataset. This can be interpreted as the representation being "pretrained" on separate data before the ridge regression estimator is learned.

Denoting the feature matrix on the fresh training set $\{\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{y}}\}$ as $\boldsymbol{\Phi} := \frac{1}{\sqrt{N}}\sigma(\tilde{\boldsymbol{X}}\boldsymbol{W}_1) \in \mathbb{R}^{n\times N}$, the CK ridge regression estimator can be obtained by solving $\hat{\boldsymbol{a}} = \operatorname{argmin}_{\boldsymbol{a}} \left\{ \frac{1}{n}\|\tilde{\boldsymbol{y}} - \boldsymbol{\Phi}\boldsymbol{a}\|^2 + \frac{\lambda}{N}\|\boldsymbol{a}\|^2 \right\}$.

2.2 Student-teacher setting and main assumptions

Given a target function (teacher model) f^* and a learned model $\hat{f}$, we evaluate the model performance using the prediction risk: $\mathcal{R}(\hat{f}) = \mathbb{E}_{\boldsymbol{x}}(\hat{f}(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 = \|\hat{f} - f^*\|_{L^2}^2$, where the expectation is taken over the test data from the same training distribution.

We utilize the orthogonal decomposition of the activation function σ. Define the coefficients

$$\mu_0 = \mathbb{E}[\sigma(z)], \quad \mu_1 = \mathbb{E}[z\sigma(z)], \quad \mu_2 = \sqrt{\mathbb{E}[\sigma(z)^2] - \mu_0^2 - \mu_1^2}, \quad \text{where } z \sim \mathcal{N}(0,1). \tag{2.3}$$

This implies $\sigma(z) = \mu_0 + \mu_1 z + \sigma_\perp(z)$, where $\mathbb{E}[\sigma_\perp(z)] = \mathbb{E}[z\sigma_\perp(z)] = 0$, and $\mathbb{E}[\sigma_\perp(z)^2] = \mu_2^2$.

Similarly, for square integrable target function f^*, we have the orthogonal decomposition

$$f^*(\boldsymbol{x}) = \mu_0^* + \mu_1^*\langle \boldsymbol{x}, \boldsymbol{\beta}_*\rangle + \mathsf{P}_{>1}f^*(\boldsymbol{x}), \quad \mu_1^*\boldsymbol{\beta}_* = \mathbb{E}[\boldsymbol{x}f^*(\boldsymbol{x})], \tag{2.4}$$

where $\mathsf{P}_{>1}$ is the projector orthogonal to constant and linear functions in $L^2(\mathbb{R}^d, \Gamma)$, which implies that $\mathbb{E}[\mathsf{P}_{>1}f^*(\boldsymbol{x})] = 0, \mathbb{E}[\boldsymbol{x}\mathsf{P}_{>1}f^*(\boldsymbol{x})] = \boldsymbol{0}$. As $d \to \infty$, quantities defined in (2.4) satisfy $\|\boldsymbol{\beta}_*\| = 1$, $\|\mathsf{P}_{>1}f^*\|_{L^2} \to \mu_2^*$, where $\mu_0^*, \mu_1^*, \mu_2^*$ are bounded constants. Intuitively, μ_0^*, μ_1^*, and μ_2^* can be interpreted as the "magnitude" of the constant, linear, and nonlinear components of f^*, respectively.

Assumption 10.

1. ***Proportional limit.*** *$n, d, N \to \infty$, $n/d \to \psi_1$, $N/d \to \psi_2$, where $\psi_1, \psi_2 \in (0, \infty)$.*
2. ***Gaussian initialization.*** *$\sqrt{d}\cdot[\boldsymbol{W}_0]_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $\sqrt{N}\cdot[\boldsymbol{a}]_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, for $i \in [d], j \in [N]$.*
3. ***Normalized activation.*** *The activation function σ has λ_σ-bounded first three derivatives almost surely. In addition, σ satisfies $\mu_0 = 0$ and $\mu_1, \mu_2 \neq 0$ defined in (2.3).*
4. ***Single-index teacher.*** *Labels are generated as $y_i = f^*(\boldsymbol{x}_i) + \varepsilon_i$, where $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \boldsymbol{I})$, and ε_i is i.i.d. sub-Gaussian noise with mean 0 and variance σ_ε^2. The teacher $f^*(\boldsymbol{x}) = \sigma^*(\langle \boldsymbol{x}, \boldsymbol{\beta}_*\rangle)$, where $\boldsymbol{\beta}_* \in \mathbb{R}^d$ with $\|\boldsymbol{\beta}_*\| = 1$, and σ^* is Lipschitz with $\mu_0^* = 0$, $\mu_1^* \neq 0$ as defined in (2.4).*

2.3 $\eta = \Theta(1)$: improvement over the initial CK

In this section, we precisely characterize the CK prediction risk under the small learning rate $\eta = \Theta(1)$. We first introduce the Gaussian equivalence property which will be useful in the risk computation. The Gaussian Equivalence Theorem (GET) states that the performance of a nonlinear kernel model is the same as that of a noisy linear model. Specifically, for the ridge regression estimator, define

$$\begin{aligned} \mathcal{R}_{\mathrm{F}}(\lambda) &= \mathbb{E}_{\boldsymbol{x}}\big(\langle \boldsymbol{\phi}_{\mathrm{F}}(\boldsymbol{x}), \hat{\boldsymbol{a}}_\lambda\rangle - f^*(\boldsymbol{x})\big)^2, \\ \hat{\boldsymbol{a}}_\lambda &= \operatorname{argmin}_{\boldsymbol{a}}\Big\{ \frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{\phi}_{\mathrm{F}}(\boldsymbol{x}_i), \boldsymbol{a}\rangle)^2 + \frac{\lambda}{N}\|\boldsymbol{a}\|^2 \Big\}, \end{aligned} \tag{2.5}$$

where $\mathrm{F} \in \{\mathrm{CK}, \mathrm{GE}\}$ indicates the choice of feature map, which can be either the nonlinear CK feature $\boldsymbol{\phi}_{\mathrm{CK}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sigma(\boldsymbol{W}^\top\boldsymbol{x})$, or the linear Gaussian equivalent (GE) feature $\boldsymbol{\phi}_{\mathrm{GE}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\big(\mu_1\boldsymbol{W}^\top\boldsymbol{x} + \mu_2\boldsymbol{z}\big)$ where $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ is independent of $\boldsymbol{x}$, $\boldsymbol{W}$. In the following, for both $\boldsymbol{\phi}_{\mathrm{CK}}$ and $\boldsymbol{\phi}_{\mathrm{GE}}$, we take $\boldsymbol{W}$ to be the updated weight matrix $\boldsymbol{W}_1$ after one GD step.

The Gaussian equivalence refers to the universality phenomenon $\mathcal{R}_{\mathrm{CK}}(\lambda) \approx \mathcal{R}_{\mathrm{GE}}(\lambda)$. For RF models, the GET has been rigorously proved in Hu & Lu (2020); Montanari & Saeed (2022); Mei & Montanari (2022). Furthermore, Goldt et al. (2021); Loureiro et al. (2021) provided empirical evidence that such equivalence holds for more general feature maps, including the representation

of certain pretrained NNs (e.g., see Loureiro et al. (2021, Figure 4)). Since our setting goes beyond RF models and cannot be covered by the prior results, we establish the GET for our *trained* feature map under small learning rate.

Theorem 1. *Suppose that Assumption 10 holds and the activation σ is an odd function. If the learning of $\boldsymbol{W}_1$ in (2.2) and estimation of $\hat{\boldsymbol{a}}_\lambda$ in (2.5) are performed on independent training data $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$, respectively, then the GET holds after the first-layer weight is trained for one gradient step with learning rate $\eta = \Theta(1)$; that is, for the CK feature $\boldsymbol{\phi}_{\mathrm{CK}}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\sigma(\boldsymbol{W}_1^\top \boldsymbol{x})$, and $\lambda > 0$,*

$$|\mathcal{R}_{\mathrm{CK}}(\lambda) - \mathcal{R}_{\mathrm{GE}}(\lambda)| = o_{d,\mathbb{P}}(1).$$

Implications of Gaussian equivalence. Under the GET, we can alternatively compute $\mathcal{R}_{\mathrm{GE}}(\lambda)$, the prediction risk of ridge regression on noisy Gaussian features $\boldsymbol{\phi}_{\mathrm{GE}}$, which is much easier to analyze. The GET also implies that the kernel estimator is essentially "linear" in high dimensions. For the squared loss, it is straightforward to verify that the Gaussian equivalent model cannot learn the nonlinear component of the target function $\mathsf{P}_{>1} f^*$ as follows.

Fact 2. *Under the same assumptions as Theorem 1, $\mathcal{R}_{\mathrm{GE}}(\lambda) \geq \|\mathsf{P}_{>1} f^*\|_{L^2}^2$ for any $\psi_1, \psi_2, \lambda > 0$.*

Hence when $\eta = \Theta(1)$, even though training the first-layer $\boldsymbol{W}$ for one step can lead to non-trivial improvement over the initial RF model (which we precisely quantify in Section 2.3), the learned CK cannot outperform the best linear model on the input features. In other words, to (possibly) learn a nonlinear f^*, the trained feature map needs to violate the GET. In the case of one gradient step on $\boldsymbol{W}$, this amounts to using a sufficiently large step size, which we analyze in Section 2.4.

Precise asymptotics of CK ridge regression Having established the Gaussian equivalence property for the CK ridge estimator after one gradient step with $\eta = \Theta(1)$, we can now compute the asymptotic prediction risk for the trained kernel and compare with the initialized RF. To quantify the discrepancy in the prediction risk (2.5), we write $\mathcal{R}_0(\lambda)$ as the prediction risk of the initialized RF ridge regression estimator (on the feature map $\boldsymbol{x} \mapsto \sigma(\boldsymbol{W}_0^\top \boldsymbol{x})$), and $\mathcal{R}_1(\lambda)$ as the prediction risk of the ridge estimator on the trained feature map after one feature learning step $\boldsymbol{x} \mapsto \sigma(\boldsymbol{W}_1^\top \boldsymbol{x})$.

Theorem 3. *Under the same assumptions as Theorem 1 and $\eta = \Theta(1)$, we have*

$$\mathcal{R}_0(\lambda) - \mathcal{R}_1(\lambda) \xrightarrow{\mathbb{P}} \delta(\eta, \lambda, \psi_1, \psi_2) \geq 0,$$

where $\delta(\eta, \lambda, \psi_1, \psi_2)$ is a non-negative constant. Here, δ is a non-negative function of $\eta, \lambda, \psi_1, \psi_2 \in (0, +\infty)$ with parameters μ_1^*, μ_1, μ_2, and it vanishes if and only if (at least) one of μ_1^*, μ_1 and η is equal to zero.

Remarkably, this improvement (when $\delta > 0$) holds for any $\psi_1, \psi_2 \in (0, \infty)$, that is, taking one gradient step (with learning rate $\eta = \Theta(1)$) is *always* beneficial, even when the training set size n is small.

2.4 $\eta = \Theta(\sqrt{N})$: improvement over the kernel lower bound

In this section, we consider a gradient step with large learning rate $\eta = \Theta(\sqrt{N})$, which matches the asymptotic order of the Frobenius norm of the gradient $\boldsymbol{G}_0$ and that of the initialized weight matrix $\boldsymbol{W}_0$. Note that after absorbing the prefactors, this learning rate scaling is analogous to the maximal update parameterization (Yang & Hu, 2020), which admits a feature learning limit. More specifically, the change in each coordinate of the feature vector $[\sigma(\boldsymbol{W}^\top \boldsymbol{x})]_i$ is $\tilde{\Theta}_{d,\mathbb{P}}(1)$, which has roughly the same order of magnitude as its value at initialization.

Due to the large step size, columns of the updated weight matrix $\boldsymbol{W}_1$ are no longer near-orthogonal, which is an important property in existing analyses of the Gaussian equivalence. Indeed, we will see that in this regime, the ridge regression estimator on the trained CK features is no longer "linear" and can potentially outperform the kernel lower bound in the proportional

limit. However, in the absence of GET, it is difficult to derive the precise asymptotics of the CK model. As an alternative, we establish an *upper bound* on the prediction risk $\mathcal{R}_1(\lambda)$, which we then compare against the kernel ridge lower bound.

Existence of a "good" solution. Given the trained first-layer weights $\boldsymbol{W}_1$, we first construct a second-layer $\tilde{\boldsymbol{a}}$ for which the prediction risk can be upper-bounded. For a pair of nonlinearities (σ, σ^*), we introduce a scalar τ^* which is the optimum of the following minimization problem:

$$\tau^* := \inf_{\kappa \in \mathbb{R}} \mathbb{E}_{\xi_1}\Big[\big(\sigma^*(\xi_1) - \mathbb{E}_{\xi_2}\sigma(\kappa\xi_1 + \xi_2)\big)^2\Big], \tag{2.6}$$

where $\xi_1, \xi_2 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. We write κ^* as an optimal value at which τ^* is attained (when τ^* is not achieved by a finite κ, the same argument holds by introducing a small tolerance factor $\epsilon > 0$ in τ^*). Roughly speaking, τ^* approximates the prediction risk of a specific student model which takes the form of an average over a *subset* of neurons (after one feature learning step). In particular, the first term on the RHS of (2.6) containing σ^* corresponds to the teacher f^*, and the second term $\mathbb{E}_{\xi_2}$ represents the constructed student model. The following lemma shows that we can find some $\tilde{\boldsymbol{a}}$ on the trained CK features whose prediction risk is approximately τ^*, under the additional assumption that the activation function σ is bounded.

Lemma 4 (Informal)**.** *Suppose that Assumption 10 holds and σ is bounded. Then, after one gradient step on $\boldsymbol{W}$ with $\eta = \Theta(\sqrt{N})$, there exist some second-layer coefficients $\tilde{\boldsymbol{a}}$ such that the constructed student model $\tilde{f}(\boldsymbol{x}) = \frac{1}{\sqrt{N}}\tilde{\boldsymbol{a}}^\top\sigma(\boldsymbol{W}_1^\top\boldsymbol{x})$ achieves a prediction risk which is "close" to τ^*.*

It is worth noting that the definition of τ^* does not involve the specific value of the learning rate η. This is because for any choice of $\eta = \Theta(\sqrt{N})$, due to the Gaussian initialization of a_i, we can find a subset of weights that receive a "good" learning rate (with high probability) such that the corresponding neurons are useful for learning the teacher model. In addition, observe that τ^* is a simple Gaussian integral which can be numerically or analytically computed. For instance, when $\sigma = \sigma^* = \mathrm{erf}$, one can easily verify that $\kappa^* = \sqrt{3}$ and $\tau^* = 0$.

Prediction risk of ridge regression. Since we have established the existence of a "good" student model $\tilde{f}$ that can achieve a prediction risk close to τ^* (as defined in (2.6)), in what follows, we prove an upper bound for the prediction risk of the ridge regression estimator on the trained CK features $\mathcal{R}_1(\lambda)$ in terms of the scalar τ^*.

Theorem 5. *Under the same assumptions as Lemma 4, after one gradient step on $\boldsymbol{W}$ with $\eta = \Theta(\sqrt{N})$, there exist constants $C, \psi_1^* > 0$ such that for any $n/d > \psi_1^*$, the ridge regression estimator (2.5) with regularization parameter $n^{\varepsilon-1} < N^{-1}\lambda < n^{-\varepsilon}$ for some small $\varepsilon > 0$ satisfies*

$$\mathcal{R}_1(\lambda) \le 10\tau^* + C\Big(\sqrt{\tau^*}\cdot\sqrt{\tfrac{d}{n}} + \tfrac{d}{n}\Big),$$

with probability 1 as $n, d, N \to \infty$ proportionally.

While Theorem 5 does not provide exact expression of the prediction risk, the upper bound still allows us to compare the prediction risk of the CK ridge regression before and after one large gradient step. In particular, if $\|\mathsf{P}_{>1}f^*\|_{L^2}^2 \ge 10\tau^*$ (the constant 10 is not optimized), we know that the trained CK can outperform the kernel lower bound (and also the initialized CK) in the proportional limit, when the ratio $\psi_1 = n/d$ is sufficiently large.

Corollary 6. *Under the same conditions as Theorem 5, there exists a constant ψ_1^* such that for any $\psi_1 > \psi_1^*$, the following holds with probability 1 when $n, d, N \to \infty$ proportionally:*

- *For $\sigma = \sigma^* = \mathrm{erf}$, we have $\mathcal{R}_1(\lambda) = \mathcal{O}(d/n)$.*
- *For $\sigma = \sigma^* = \tanh$, we have $\mathcal{R}_1(\lambda) < \|\mathsf{P}_{>1}f^*\|_{L^2}^2$.*

In the two examples outlined above, training the features by taking one large gradient step on the first-layer parameters can lead to substantial improvement in the performance of the CK model. In fact, the new ridge regression estimator may outperform a wide range of kernel models.

However, we emphasize that this separation is only present in specific pairs of (σ, σ^*) for which the scalar τ^* is sufficiently small. In general settings, learning a good representation would likely require a training procedure that takes more than one gradient step (even if f^* is as simple as a single-index model).

References

J. Ba, M. A. Erdogdu, T. Suzuki, Z. Wang, D. Wu, and G. Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. to appear.

S. Goldt, B. Loureiro, G. Reeves, F. Krzakala, M. Mézard, and L. Zdeborová. The gaussian equivalence of generative models for learning with shallow neural networks. *Proceedings of Machine Learning Research vol*, 145:1–46, 2021.

H. Hu and Y. M. Lu. Universality laws for high-dimensional learning with random features. *arXiv preprint arXiv:2009.07669*, 2020.

Y. Ingster and N. Stepanova. Estimation and detection of functions from anisotropic sobolev classes. *Electronic Journal of Statistics*, 5:484–506, 2011.

B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, and L. Zdeborová. Learning curves of generic features maps for realistic datasets with a teacher-student model. *Advances in Neural Information Processing Systems*, 34, 2021.

S. Mei and A. Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4): 667–766, 2022.

A. Montanari and B. N. Saeed. Universality of empirical risk minimization. In *Conference on Learning Theory*, pp. 4310–4312. PMLR, 2022.

S. M. Nikol'skii. *Approximation of functions of several variables and imbedding theorems*, volume 205. Springer-Verlag Berlin Heidelberg, 1975.

S. Okumoto and T. Suzuki. Learnability of convolutional neural networks for infinite dimensional input via mixed and anisotropic smoothness. In *International Conference on Learning Representations*, 2022.

H.-J. Schmeisser. An unconditional basis in periodic spaces with dominating mixed smoothness properties. *Analysis Mathematica*, 13(2):153–168, 1987.

W. Sickel and T. Ullrich. Tensor products of Sobolev–Besov spaces and applications to approximation from the hyperbolic cross. *Journal of Approximation Theory*, 161(2):748–786, 2009.

T. Suzuki. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

T. Suzuki and A. Nitanda. Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic Besov space. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021. to appear.

H. Triebel. Entropy numbers in function spaces with mixed integrability. *Revista matemática complutense*, 24(1):169–188, 2011.

J. Vybiral. Function spaces with dominating mixed smoothness. *Dissertationes Math. (Rozprawy Mat.)*, 436:3–73, 2006.

G. Yang and E. J. Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.