Learning Backpropagation-Free Deep Architectures with Kernels

Shiyu Duan[1] **Shujian Yu**[1] **Yunmei Chen**[2] **Jose C. Principe**[1]

Abstract

We present a framework to "kernelize" (partly or completely) any neural network (NN) and show that after kernelization, feedforward networks may be learned without backpropagation (BP). When learning a multilayer feedforward NN in a supervised setting, the reason why BP is required is that there are no explicit targets for the hidden layers. We show that this need not be the case for its kernelized counterpart. To be specific, let a random sample S, a two-layer feedforward architecture $F_2 \circ F_1$ and an objective function $R(F_2 \circ F_1,\, S)$ be given, define $F_2^\star \circ F_1^\star := \arg\min_{F_2 \circ F_1} R(F_2 \circ F_1,\, S)$. In classification, we prove for some objective functions that if F_2 is kernelized and fully-connected with F_1, one can characterize the restriction of $F_1^\star$ on S to some degree using little information from F_2. And we show that this characterization can be used as a target for learning F_1, allowing one to train the model in a greedy, layer-wise fashion. The proposed training method extends to arbitrary feedforward NN architectures with the requirement that some layers have been kernelized. And it can be given an intuitive geometric interpretation, making the dynamics of learning in a deep network transparent. Empirical results are provided to complement our theory.

1. Introduction

One can "kernelize" any neural network (NN) by replacing each artificial neuron (McCulloch & Pitts, 1943), i.e., function approximator of the form $f(\mathbf{x}) = \sigma\big(\mathbf{w}^\top \mathbf{x} + b\big)$, with a kernel machine, i.e., function approximator of the form $f(\mathbf{x}) = \langle \mathbf{w},\, \phi(\mathbf{x})\rangle + b$ with kernel function $k(\mathbf{x},\, \mathbf{y}) = \langle \phi(\mathbf{x}),\, \phi(\mathbf{y})\rangle$. While the nonlinearities in deep NNs make it notoriously difficult to analyze these models, the simple interpretation of a kernel machine as a hyperplane in a reproducing kernel Hilbert space (RKHS) makes the kernelized networks more tractable mathematically. We shall refer to the kernelized NNs in general as kernel networks (KNs).

We then discuss learning feedforward KNs for classification. Evidently, these models can still be trained with backpropagation (BP) (Rumelhart et al., 1986). However, in the context of supervised learning, the need for BP in learning a deep architecture is caused by the fact that there is no explicit target information to tune the hidden layers (Rumelhart et al., 1986). Moreover, BP is usually computationally intensive and can suffer from vanishing gradient. And most importantly, BP forces the user to treat deep architectures as "black boxes" due to its end-to-end nature.

We propose a layer-by-layer learning framework by deriving explicit targets for the hidden layers. The targets are optimal for minimizing the objective function of the network according to the following result: let a random sample S, a two-layer feedforward architecture $F_2 \circ F_1$ and an objective function $R(F_2 \circ F_1,\, S)$ be given, where $\circ$ denotes function composition and $F_2,\, F_1$ are mappings between Euclidean spaces. Define $F_2^\star \circ F_1^\star := \arg\min_{F_2 \circ F_1} R(F_2 \circ F_1,\, S)$. In classification, we prove for some objective functions that if F_2 is kernelized and fully-connected with F_1, one can characterize the restriction of $F_1^\star$ on S to some degree using little information from F_2. And we show that this characterization can be used as a target for learning F_1. Consequently, one can greedily learn F_1 and F_2, in that order.

The layer-wise learning algorithm enjoys the same optimality guarantee as BP in the sense that they both effectively minimize the objective function. But the former is much faster and evidently less susceptible to vanishing gradient. It also greatly increases the transparency of deep models: the quality of learning in the hidden layers can be directly assessed during or after training, providing more information about training to the user. Also, new model selection paradigms are now available since the bad performance of the network can be traced to a certain layer, allowing the user to "debug" the layers individually. Moreover, the target for each hidden layer can be given an intuitive geometric interpretation, making the learning dynamics in a KN more interpretable than that in the original NN.

Empirical results are provided to complement our theory.

[1] Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida, USA [2] Department of Mathematics, University of Florida, Gainesville, Florida, USA. Correspondence to: Shiyu Duan <michaelshiyu@ufl.edu>.

We fully or partly kernelized both fully-connected and convolutional NNs and trained them layer-wise. The resulting KNs compare favorably with their NN equivalents trained with BP as well as some other commonly-used deep architectures trained with BP together with unsupervised greedy pre-training.

2. Assumptions and Notations

We consider the following setup and assumptions: let a realization of an i.i.d. random sample $\{(\mathbf{x}_n,\, y_n)\}_{n=1}^N$ be given, where $(\mathbf{x}_n,\, y_n) \in \mathbb{X}_1 \times \mathbb{Y} \subset \mathbb{R}^{d_0} \times \mathbb{R}$, denote $\{\mathbf{x}_n\}_{n=1}^N$ as $S_{\mathbf{X}}$ and $\{y_n\}_{n=1}^N$ as S_Y for convenience. For $i = 1, 2, \dots, l$, consider real, continuous, symmetric, positive definite (PD) (Schölkopf & Smola, 2001) kernel $k^{(i)} : \mathbb{X}_i \times \mathbb{X}_i \to \mathbb{R}$, $\mathbb{X}_i \subset \mathbb{R}^{d_{i-1}}$ (for $i > 1$, d_{i-1} is determined by the width of the $i-1^{\text{th}}$ layer). $k^{(i)}(\mathbf{x},\, \mathbf{y}) = \left\langle \phi^{(i)}(\mathbf{x}),\, \phi^{(i)}(\mathbf{y}) \right\rangle_{H_i}$, where $\phi^{(i)}$ is a mapping into RKHS H_i. Assume $k^{(i)}(\mathbf{x},\, \mathbf{x}) = c < +\infty$ for all $\mathbf{x} \in \mathbb{X}_i$ and $\inf_{\mathbf{x},\mathbf{y} \in \mathbb{X}_i} k^{(i)}(\mathbf{x},\, \mathbf{y}) = a > -\infty$. It is straightforward to check using Cauchy-Schwarz inequality that the first condition implies $\max_{\mathbf{x},\mathbf{y} \in \mathbb{X}_i} k^{(i)}(\mathbf{x},\, \mathbf{y}) = c$.

For Proposition 3.2 and Lemma 4.4, we impose the following smoothness assumption on all kernels considered: for each fixed $\mathbf{x} \in \mathbb{X}_i$, we assume that $k^{(i)}(\mathbf{x},\, \mathbf{y})$, as a function of $\mathbf{y}$, is $L_{\mathbf{x}}^{(i)}$-Lipschitz with respect to the Euclidean metric on $\mathbb{X}_i$. Let $\sup_{\mathbf{x} \in \mathbb{X}_i} L_{\mathbf{x}}^{(i)} = L^{(i)}$, which we assume to be finite. For Theorem 4.5, we assume that $k^{(3)}(\mathbf{x},\, \mathbf{y})$, as a function of $(\mathbf{x},\, \mathbf{y})$, depends only on and strictly decreases in $\|\mathbf{x} - \mathbf{y}\|_2$ for all $\mathbf{x},\, \mathbf{y} \in \mathbb{X}_3$ with $k^{(3)}(\mathbf{x},\, \mathbf{y}) > a$, and that the infimum $\inf_{\mathbf{x},\, \mathbf{y} \in \mathbb{X}_3} k^{(3)}(\mathbf{x},\, \mathbf{y}) = a$ is attained in $\mathbb{X}_3$ at all $\mathbf{x},\, \mathbf{y}$ with $\|\mathbf{x} - \mathbf{y}\|_2 \geq \eta$. Also assume that $\inf_{\mathbf{x},\, \mathbf{y} \in \mathbb{X}_3;\, \|\mathbf{x}-\mathbf{y}\|_2 < \eta} \left| \partial k^{(3)}(\mathbf{x},\, \mathbf{y}) / \partial \|\mathbf{x} - \mathbf{y}\|_2 \right| = \iota^{(3)}$ is defined and is positive.

For the rest of this paper, we shall use bold letters to denote vectors and matrices. For random elements, we use capital letter to denote the random element and lower-case letter a realization of it. The following notations will be used whenever convenient: for a general l-layer feedforward architecture $\mathbf{F}^{(l)} \circ \cdots \circ \mathbf{F}^{(1)}$ and for $i = 2, 3, \dots, l$, $\mathbf{F}^{(i)}(\mathbf{x}) := \mathbf{F}^{(i)} \circ \cdots \circ \mathbf{F}^{(1)}(\mathbf{x})$ and the shorthand $\mathbf{F}^{(i)}(S_{\mathbf{X}})$ represents $\left\{ \mathbf{F}^{(i)}(\mathbf{x}_n) \right\}_{n=1}^N$. When there is no confusion, we shall suppress the dependency of any loss function on the example for brevity, i.e., for a loss function ℓ, instead of writing $\ell(f,\, (x,\, y))$, we write $\ell(f)$.

3. Kernelizing a Neural Network

In this section, we discuss how to kernelize an NN. We first present the generic approach and then as an example, concretely define a fully-kernelized Multilayer Perceptron (MLP). To further shed light on the effect of kernelization on the expressive power of the original model, we give an analysis on the model complexity of a fully-kernelized MLP.

3.1. A Generic Approach to Kernelization

In an NN, any neuron, i.e., function approximator of the form $f(\mathbf{x}) = \sigma\big(\mathbf{w}^\top \mathbf{x} + b\big)$, can be directly replaced by a kernel machine, i.e., function approximator of the form $f(\mathbf{x}) = \langle \mathbf{w},\, \phi(\mathbf{x}) \rangle + b$ with kernel $k(\mathbf{x},\, \mathbf{y}) = \langle \phi(\mathbf{x}),\, \phi(\mathbf{y}) \rangle$, without altering the architecture and functionality of the network. In this way, one can kernelize an NN to any degree: a node, several nodes, a layer, several layers, or the entire network. KN inherits the expressive power of the original NN since a kernel machine is a universal function approximator under mild conditions (Park & Sandberg, 1991; Micchelli et al., 2006). Moreover, KN works in a more intuitive way since each kernelized node is a simple linear model in an RKHS. In comparison, it is notoriously difficult to analyze deep NNs due to the nonlinearities involved.

3.2. Kernelized MLP: The Architecture

As a more concrete example, we now define a fully-kernelized l-layer MLP, which we will specifically refer to as kernel MLP (kMLP).[1]

The l-layer kMLP is defined as follows. For $i \geq 1$, the i^{th} layer in a kMLP, denoted $\mathbf{F}^{(i)}$, is an array of d_i kernel machines: $\mathbf{F}^{(i)} : \mathbb{X}_i \to \mathbb{R}^{d_i}, \mathbf{F}^{(i)} = \left(f_1^{(i)},\, f_2^{(i)},\, \dots,\, f_{d_i}^{(i)} \right)$. Let $\mathbf{F}^{(0)}$ be the identity map on $\mathbb{R}^{d_0}$, each $f_j^{(i)} : \mathbb{X}_i \to \mathbb{R}$ is a hyperplane in H_i: $f_j^{(i)}(\mathbf{x}) = \left\langle \mathbf{w}_j^{(i)},\, \phi^{(i)}\left(\mathbf{F}^{(i-1)} \circ \cdots \circ \mathbf{F}^{(0)}(\mathbf{x}) \right) \right\rangle_{H_i} + b_j^{(i)}$, $\mathbf{w}_j^{(i)} \in H_i$, $b_j^{(i)} \in \mathbb{R}$. In practice, $\mathbf{w}_j^{(i)}$ is usually not accessible but can be approximated using $\sum_{n=1}^N \alpha_{nj}^{(i)} \phi^{(i)}\left(\mathbf{F}^{(i-1)} \circ \cdots \circ \mathbf{F}^{(0)}(\mathbf{x}_n) \right)$, where the $\alpha_{nj}^{(i)} \in \mathbb{R}$ are the learnable parameters.[2] The set of mappings $\left\{ \mathbf{F}^{(\ell)} \circ \cdots \circ \mathbf{F}^{(1)} :\, \alpha_{nj}^{(i)},\, b_j^{(i)} \in \mathbb{R} \text{ for all admissible } n,\, j,\, i \right\}$ defines an l-layer kMLP.

3.3. Gaussian Complexity of a Kernelized MLP

We give a bound on the model complexity of an l-layer kMLP using a well-known complexity measure called Gaussian complexity (Bartlett & Mendelson, 2002). In particular, the bound describes the relationship between the depth/width of the model and the complexity of its hypothesis class, providing useful information for model selection.

[1] A PyTorch-based (Paszke et al., 2017) library for implementing KN and the proposed layer-wise learning algorithm is available at: *https://github.com/michaelshiyu/kerNET*.

[2] The sufficiency of this expansion for minimizing the objective function will be later justified in Section 4 in the layer-wise setting using representer theorem (Schölkopf et al., 2001).

We first review the definition of Gaussian complexity.

Definition 3.1 (*Gaussian complexity*)**.** *Let $X_1, \dots, X_N$ be i.i.d. random elements defined on metric space $\mathbb{X}$ and let $\mathbb{F}$ be a set of functions mapping from $\mathbb{X}$ into $\mathbb{R}$. Define*

$$\hat{\mathcal{G}}_N(\mathbb{F}) = \mathbb{E}\left[\sup_{f\in\mathbb{F}}\left|\frac{2}{N}\sum_{n=1}^{N} Z_n f(X_i)\right| \middle| X_1, \dots, X_N\right],$$

where $Z_1, \dots, Z_N$ are independent standard normal random variables. The Gaussian complexity of $\mathbb{F}$ is defined as $\mathcal{G}_N(\mathbb{F}) = \mathbb{E}\,\hat{\mathcal{G}}_N(\mathbb{F})$

Intuitively, Gaussian complexity quantifies how well elements in a given function class can be correlated with a normally-distributed noise sequence of length N (Bartlett & Mendelson, 2002).

Proposition 3.2. *Given an l-layer kMLP, approximate $\mathbf{w}_j^{(i)}$ using $\sum_{\nu=1}^{m} \alpha_{\nu j}^{(i)}\phi^{(i)}\big(\mathbf{F}^{(i-1)}\circ\cdots\circ\mathbf{F}^{(0)}(\mathbf{x}_\nu)\big)$, where the $\mathbf{x}_\nu$ are an m-subset of $S_{\mathbf{X}}$, $\boldsymbol{\alpha}_j^{(i)} := \left(\alpha_{1j}^{(i)}, \dots, \alpha_{mj}^{(i)}\right) \in \mathbb{R}^m$ and $b_j^{(i)} \in \mathbb{R}$. Assume $\left\|\boldsymbol{\alpha}_j^{(i)}\right\|_1 \leq A_i$ and let $d_l = 1$. Consider $\mathbb{F}_1 = \big\{(f_1, \dots, f_{d_1}) : \mathbb{X}_1 \to \mathbb{R}^{d_1} \,|\, f_j \in \Omega,\, j = 1, \dots, d_1\big\}$, where Ω is a given hypothesis class that is closed under negation, i.e., if $f \in \Omega$, then $-f \in \Omega$. Denote the class of functions implemented by this kMLP as $\mathbb{F}_{l\text{-kMLP}}$, if $\mathbf{F}^{(1)} \in \mathbb{F}_1$, for $i \geq 2$, we have*

$$\mathcal{G}_N(\mathbb{F}_{l\text{-kMLP}}) \leq d_1 \prod_{i=2}^{l} A_i L^{(i)} d_i \mathcal{G}_N(\Omega).$$

It is worth noting that the model complexity kMLP grows in the depth and width of the network in a similar way as that of an MLP (Sun et al., 2016). In particular, the expressive power of the model increases linearly in the width of a given layer and roughly exponentially in the depth of the network.

4. A Layer-Wise Learning Framework

We now formally present a framework for learning feedforward KNs layer-wise for classification by explicitly characterizing targets for the hidden layers. The characterization may depend on on the objective function involved and hence using different objective functions may result in different realizations of the layer-wise framework.

We proceed by first describing the general framework and then provide a realization as an example. This realization is simple to implement and enjoys an intuitive geometric interpretation. Furthermore, despite that we only provide optimality guarantee for this realization under a specific choice of objective functions, we shall later empirically show that it works well with most popular objective functions. To simplify discussion, we shall restrict ourselves to binary classification ($\mathbb{Y} = \{+1, -1\}$) and directly give the result on classification with more than two classes in the end.

4.1. The Framework

We first describe the setup. Consider hypothesis $\mathcal{F} \in \mathbb{F} := \big\{\mathbf{F}^{(l)}\circ\cdots\circ\mathbf{F}^{(1)} \,|\, \mathbb{F}_i \ni \mathbf{F}^{(i)} \colon \mathbb{X}_i \to \mathbb{R}^{d_i},\, i = 1, \dots, l\big\}$, where $\mathbb{F}_i$ is some hypothesis class for all i. Let loss function $\ell_l(\mathcal{F}, (\mathbf{x}, y))$ be given, which induces a risk $R_l(\mathcal{F}) := \mathbb{E}_{(\mathbf{X}, Y)}\,\ell_l(\mathcal{F}, (\mathbf{X}, Y))$. Let $\tilde{R}_l(\mathcal{F}, (S_{\mathbf{X}}, S_Y))$ be an upper bound on R_l that we shall refer to as an objective function.[3]

It is important to note that for the rest of this section, we use the following definition of equivalence when we talk about two hypotheses of a given layer being equal : for $i = 1, \dots, l$, $\mathbf{F}^{(i)} = \mathbf{G}^{(i)}$ if and only if for any $(S_{\mathbf{X}}, S_Y)$, we have

$$\min_{\substack{\mathbf{F}^{(l)}, \dots, \mathbf{F}^{(i+1)},\\ \mathbf{F}^{(i-1)}, \dots, \mathbf{F}^{(1)}}} \tilde{R}_l\Big(\mathbf{F}^{(l)}\circ\cdots\circ\mathbf{F}^{(i)}\circ\cdots\circ\mathbf{F}^{(1)}\Big)$$
$$= \min_{\substack{\mathbf{G}^{(l)}, \dots, \mathbf{G}^{(i+1)},\\ \mathbf{G}^{(i-1)}, \dots, \mathbf{G}^{(1)}}} \tilde{R}_l\Big(\mathbf{G}^{(l)}\circ\cdots\circ\mathbf{G}^{(i)}\circ\cdots\circ\mathbf{G}^{(1)}\Big).$$

It is easy to check that this is indeed an equivalence relation. Intuitively, this means that we consider two hypotheses of a given layer to be equally good if the best networks one can build with these two hypotheses minimize the objective function equally well. It is easy to see that, in a strictly layer-wise setting where one may only learn one layer at a time, this notion of equivalence is a sufficient one for comparing hypotheses.

Denote the optimal hypothesis in the given hypothesis class $\mathbb{F}$ as $\mathbf{F}^{(l)\star}\circ\cdots\circ\mathbf{F}^{(1)\star} = \mathcal{F}^\star := \arg\min_{\mathcal{F}\in\mathbb{F}} \tilde{R}_l(\mathcal{F})$. We call $\mathbf{F}^{(i)\star}$ the "layer-wise optimality" of layer i. Assume $l = 2$ without loss of generality. The goal is to have the hidden layer $\mathbf{F}^{(1)}$ approximate $\mathbf{F}^{(1)\star}$, freeze it, and then learn the output layer $\mathbf{F}^{(2)}$.

4.1.1. Learning $\mathbf{F}^{(1)}$

The main theoretical difficulty in learning $\mathbf{F}^{(1)}$ is that there is no explicit supervision to learn from (Rumelhart et al., 1986). To this end, we approach by giving a sufficient condition on $\mathbf{F}^{(1)}$ such that $\mathbf{F}^{(1)} = \mathbf{F}^{(1)\star}$. This result provides a target for the hidden layer, making possible a greedy, layer-wise learning method. Note that compared to the original definition of $\mathbf{F}^{(1)\star}$, it is usually easier to work with the following more concrete characterization.

[3] We could of course work directly with the risk, but in a machine learning setting where no distributional assumption can be made, it is more practical to work with an objective function that is computable without such an assumption instead.

Lemma 4.1. *Suppose* $\mathbf{F}^{(1)\star} \in \mathbb{F}_1' \subseteq \mathbb{F}_1$ *and* $\mathbf{F}^{(2)\star} \in \mathbb{F}_2' \subseteq \mathbb{F}_2$*, we have*

$$\mathbf{F}^{(1)\star} = \operatorname*{arg\,min}_{\mathbf{F}^{(1)} \in \mathbb{F}_1'} \min_{\mathbf{F}^{(2)} \in \mathbb{F}_2'} \tilde{R}_2\Big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\Big),$$

where the minimum of $\tilde{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\big)$ *w.r.t.* $\mathbf{F}^{(2)}$ *is understood as that for each fixed* $\mathbf{F}^{(1)}$*.*

We now characterize $\mathbf{F}^{(1)\star}$ under two specific loss functions ℓ_2. Similar idea can be used to extend these results to more losses, which we leave as future work.

$\mathbf{F}^{(1)\star}$ under 0-1 loss

Let $d_2 = 1$, write $f^{(2)}$ in place of $\mathbf{F}^{(2)}$ accordingly and choose ℓ_2 to be the 0-1 loss defined as $\ell_2\big(f^{(2)} \circ \mathbf{F}^{(1)}\big) = \mathbb{1}_{\{yf^{(2)}(\mathbf{x}) \le 0\}}$. We first derive the objective function $\tilde{R}_2\big(f^{(2)} \circ \mathbf{F}^{(1)}\big)$ using the following bound on the risk R_2.

Lemma 4.2. *(Bartlett & Mendelson, 2002) Let* $\mathbb{F}_{2,A} = \Big\{f : \mathbf{x} \mapsto \big\langle \mathbf{w}, \phi^{(2)}(\mathbf{x})\big\rangle_{H_2} + b \,|\, \mathbf{x} \in \mathbb{X}_2, \, \|\mathbf{w}\|_{H_2} \le A, \, b \in \mathbb{R}\Big\}$. *Fix* $A, \gamma > 0$ *and* $\mathbf{F}^{(1)}$*, with probability at least* $1 - \delta$*, every function* $f^{(2)}$ *in* $\mathbb{F}_{2,A}$ *satisfies*

$$\Pr\Big(Yf^{(2)}(\mathbf{X}) \le 0\Big) \le \hat{R}_2\Big(f^{(2)} \circ \mathbf{F}^{(1)}\Big) + 2\mathcal{G}_N(\mathbb{F}_{2,A}) + \xi,$$

where $\hat{R}_2\big(f^{(2)} \circ \mathbf{F}^{(1)}\big)$ *is defined as* $\frac{1}{N}\sum_{n=1}^N \max\big(0, 1 - y_n f^{(2)}(\mathbf{x}_n)/\gamma\big)$*, the empirical hinge loss, and* $\xi = (8/\gamma + 1)\sqrt{\log(4/\delta)/2N}$*. Given the assumptions on* $k^{(2)}$*, we have* $\mathcal{G}_N(\mathbb{F}_{2,A}) \le 2A\sqrt{c/N}$*.*

Without loss of generality, set hyperparameter $\gamma = 1$. Note that for a given $f^{(2)}$, $A = \big\|\mathbf{w}_{f^{(2)}}\big\|_{H_2}$ is the smallest nonnegative real number such that $f^{(2)} \in \mathbb{F}_{2,A}$ and it is immediate that this gives the tightest bound in Lemma 4.2. Hence we choose the objective function $\tilde{R}_2\big(f^{(2)} \circ \mathbf{F}^{(1)}\big)$ to be $\hat{R}_2\big(f^{(2)} \circ \mathbf{F}^{(1)}\big) + \tau\big\|\mathbf{w}_{f^{(2)}}\big\|_{H_2}$, where $\tau > 0$ is a hyperparameter. Let $\kappa = \frac{1}{N}\sum_{n=1}^N \mathbb{1}_{\{y_n = +\}}$. We now characterize $\mathbf{F}^{(1)\star}$.

Theorem 4.3. *Suppose* $\tau < \sqrt{2(c-a)}\min(\kappa, 1-\kappa)$ *and that* $f^{(2)\star}$ *achieves zero hinge loss on at least one example from each class. If* $\mathbf{F}^{(1)}$ *satisfies*

$$\begin{aligned} &k^{(2)}\Big(\mathbf{F}^{(1)}(\mathbf{x}_+), \, \mathbf{F}^{(1)}(\mathbf{x}_-)\Big) = a \quad \text{and} \\ &k^{(2)}\Big(\mathbf{F}^{(1)}(\mathbf{x}), \, \mathbf{F}^{(1)}(\mathbf{x}')\Big) = c \end{aligned} \tag{1}$$

for all pairs of $\mathbf{x}_+, \mathbf{x}_-$ *from distinct classes in* $S_\mathbf{X}$ *and all pairs of* $\mathbf{x}, \mathbf{x}'$ *from the same class, then* $\mathbf{F}^{(1)} = \mathbf{F}^{(1)\star}$*.*

$\mathbf{F}^{(1)\star}$ under an L^1-type loss

We now consider a special L^1-type loss that will become relevant when we later present a realization of the layer-wise framework. Let a third kernel $k^{(3)}$ be given and define an $N \times N$ matrix $\mathbf{G}^\star$ as $(\mathbf{G}^\star)_{mn} = a$, if $y_m \ne y_n$ and $(\mathbf{G}^\star)_{mn} = c$, if $y_m = y_n$. and another $N \times N$ matrix $\mathbf{G}_2$ as $(\mathbf{G}_2)_{mn} = k^{(3)}\big(\mathbf{F}^{(2)}(\mathbf{x}_m), \, \mathbf{F}^{(2)}(\mathbf{x}_n)\big)$. Consider the loss function defined as $\ell_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}, \, (\mathbf{x}_m, \, y_m), \, (\mathbf{x}_n, \, y_n)\big) = \big|(\mathbf{G}^\star)_{mn} - (\mathbf{G}_2)_{mn}\big|$. This specifies $R_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}, \, (\mathbf{X}, \, Y)\big)$ as $\mathbb{E}_{(\mathbf{X}, \, Y)}\, \ell_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}, \, (\mathbf{X}, \, Y), \, (\mathbf{X}, \, Y)\big)$. Note that due to the boundedness assumption on $k^{(3)}$, we have $\ell_2 \le 2\max(|a|, \, |c|)$. Let $\hat{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\big)$ be the sample mean of $\big\{\ell_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}, \, (\mathbf{x}_m, \, y_m), \, (\mathbf{x}_n, \, y_n)\big)\big\}_{m,n=1}^N$, we first find an objective function $\tilde{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\big)$.

Lemma 4.4. *For* $j = 1, 2, \ldots, d_2$*, let* $f_j^{(2)} \in \mathbb{F}_2$*, where* $\mathbb{F}_2$ *is a given hypothesis class. There exists an absolute constant* $C > 0$ *such that for each* $\mathbf{F}^{(1)}$*, with probability at least* $1 - \delta$*,*

$$R_2\Big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\Big) \le \hat{R}_2\Big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\Big) + \frac{4L^{(3)}Cd_2}{\max\left(|c|, \, |a|\right)}\mathcal{G}_N(\mathbb{F}_2) + \xi,$$

where $\xi = \sqrt{8\log(2/\delta)/N}$*.*

We are now in a position to characterize $\mathbf{F}^{(1)\star}$. Let $\mathbb{F}_2 = \mathbb{F}_{2,A}$ be as defined in Lemma 4.2. For a given $\mathbf{F}^{(2)} = \Big(f_1^{(2)}, \, \ldots, \, f_{d_2}^{(2)}\Big)$, it is immediate that $A = \max_j \Big\|\mathbf{w}_{f_j^{(2)}}\Big\|_{H_2}$ is the smallest nonnegative real number such that $f_j^{(2)} \in \mathbb{F}_{2,A}$ for all j, giving the tightest bound in Lemma 4.4 (recall the bound on $\mathcal{G}_N(\mathbb{F}_{2,A})$ in Lemma 4.2). Hence we choose objective function $\tilde{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\big)$ to be $\hat{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\big) + \tau\max_j \Big\|\mathbf{w}_{f_j^{(2)}}\Big\|_{H_2}$, where $\tau > 0$ is a hyperparameter. Let $\psi = \sum_{m,n=1}^N \mathbb{1}_{\{y_m \ne y_n\}}/N^2$.

Theorem 4.5. *Suppose* $\tau < \sqrt{2d_2(c-a)}\psi\iota^{(3)}$ *and that* $\mathbf{F}^{(2)\star}$ *achieves zero loss on at least one pair of examples from distinct classes. If* $\mathbf{F}^{(1)}$ *satisfies Eq. 1 for all pairs of* $\mathbf{x}_+, \, \mathbf{x}_-$ *from distinct classes in* $S_\mathbf{X}$ *and all pairs of* $\mathbf{x}, \, \mathbf{x}'$ *from the same class, then* $\mathbf{F}^{(1)} = \mathbf{F}^{(1)\star}$*.*

4.1.2. LEARNING $\mathbf{F}^{(2)}$

Despite that there is no need to manually derive supervision, the strictly layer-wise setting poses yet another difficulty for the learning of the output layer $\mathbf{F}^{(2)}$: when the hidden layer creates error, layer-wise optimality of the output layer, i.e., $\mathbf{F}^{(2)\star}$, may cease being optimal. To be specific, for $\mathbf{F}^{(1)\circ} \ne \mathbf{F}^{(1)\star}$, there may exist $\mathbf{F}^{(2)\circ} \ne \mathbf{F}^{(2)\star}$ such that $\tilde{R}_2\big(\mathbf{F}^{(2)\circ} \circ \mathbf{F}^{(1)\circ}\big) < \tilde{R}_2\big(\mathbf{F}^{(2)\star} \circ \mathbf{F}^{(1)\circ}\big)$. Using this notation, we define the network-wise optimality as follows: let $\mathbf{F}^{(1)\circ} \in \mathbb{F}_1$ be given, network-wise optimality of the output layer refers to $\mathbf{F}^{(2)\circ\star} := \arg\min_{\mathbf{F}^{(2)} \in \mathbb{F}_2} \tilde{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)\circ}\big)$. Note that $\mathbf{F}^{(2)\circ\star}$ is uniformly optimal and this notion of optimality is what we should have the output layer learn in practice. However, network-wise optimality is not locally defined for the output layer as it requires knowledge on $\mathbf{F}^{(1)\circ}$. Consequently, it is not obvious that a layer-wise learning algorithm, i.e., an algorithm that learns from the hidden layer to the output one layer one a time, freezing each layer after learning it, can learn this notion of optimality at the output layer.

We now show that $\mathbf{F}^{(2)\circ\star}$ is learnable even in this purely layer-wise setting using a "relaxed" objective function. To this end, we first need the following bound on the total error of a two-layer feedforward architecture whose output layer is kernelized and fully-connected with the hidden layer.

Proposition 4.6. *For $i = 1, 2$, let $\epsilon_i = \left\|\mathbf{F}^{(i)} - \mathbf{F}^{(i)\star}\right\|_s := \sup_{\mathbf{x}\in\mathbb{X}_i}\left\|\mathbf{F}^{(i)}(\mathbf{x}) - \mathbf{F}^{(i)\star}(\mathbf{x})\right\|_2$. Define $\mathbb{F}_2 = \left\{f : \mathbf{x} \mapsto \left\langle \mathbf{w}, \phi^{(2)}(\mathbf{x})\right\rangle_{H_2} + b \,|\, \mathbf{x} \in \mathbb{X}_2, b \in \mathbb{R}\right\}$. For $\mathbf{F}^{(2)} = \left(f_1^{(2)}, \ldots, f_{d_2}^{(2)}\right)$ with $f_j^{(2)} \in \mathbb{F}_2$ for all j, we have*

$$\left\|\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)} - \mathbf{F}^{(2)\star} \circ \mathbf{F}^{(1)\star}\right\|_s \leq \epsilon_2 + \sqrt{\epsilon_1}\sqrt{2L^{(2)}\sum_{j=1}^{d_2}\left\|\mathbf{w}_{f_j^{(2)}}\right\|_{H_2}^2}.$$

By Proposition 4.6, finding network-wise optimality is equivalent to finding layer-wise optimality under layer-wise regularization on norm of weights. In other words, suppose the objective function of the output layer is $\tilde{R}_2\left(\mathbf{F}^{(2)}\right)$[4] then to find $\mathbf{F}^{(2)\circ\star}$, it suffices to switch to the new relaxed objective function $\tilde{R}_2^\circ\left(\mathbf{F}^{(2)}\right) := \tilde{R}_2\left(\mathbf{F}^{(2)}\right) + \tau^\circ \max_j \left\|\mathbf{w}_{f_j^{(2)}}\right\|_{H_2}$, where $\tau^\circ > 0$ depends on the upstream error ϵ_1, hence is unknown. However, by treating it as a hyperparameter, the relaxed objective becomes a function only of $\mathbf{F}^{(2)}$. Thus, using this relaxed objective $\tilde{R}_2^\circ\left(\mathbf{F}^{(2)}\right)$ produces a layer-wise algorithm that learns the network-wise optimality in the output layer.

4.2. A Realization of the Layer-Wise Framework

The results we have so far enable us to give a realization of the proposed layer-wise learning framework. In this section, we first describe this realization and then show that it enjoys a geometric interpretation that makes the learning dynamics transparent. Moreover, we show that this realization enables a simple acceleration method for the kernelized non-input layers, making the architecture more practical.

4.2.1. The Realization

Consider an l-layer feedforward architecture and assume without loss of generality that layers $2, \ldots, l$ are kernelized and fully-connected. To derive supervision for the hidden layers, we first work with $\mathbf{F}^{(1)} \circ \cdots \circ \mathbf{F}^{(l-1)}$ and $\mathbf{F}^{(l)}$ being the $\mathbf{F}^{(1)}$ and $\mathbf{F}^{(2)}$ in the previous section, respectively. Choosing the objective function defined before Theorem 4.3 as $\tilde{R}_l$, we see from Theorem 4.3 that the $\mathbf{G}^\star$ defined before Lemma 4.4 characterizes $\mathbf{F}^{(l-1)\star}(S_\mathbf{X})$. And to use this characterization as a target for learning $\mathbf{F}^{(1)} \circ \cdots \circ \mathbf{F}^{(l-1)}$, we can choose the objective function defined before Theorem 4.5 as $\tilde{R}_{l-1}$. Then we work with $\mathbf{F}^{(1)} \circ \cdots \circ \mathbf{F}^{(l-2)}$ and $\mathbf{F}^{(l-1)}$ being the $\mathbf{F}^{(1)}$ and $\mathbf{F}^{(2)}$, respectively. By Theorem 4.5, $\mathbf{G}^\star$ again characterizes $\mathbf{F}^{(l-2)\star}(S_\mathbf{X})$. Continuing in this fashion would give targets for the rest of the hidden layers, which turn out to be always $\mathbf{G}^\star$. Now, sequentially learn $\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \ldots, \mathbf{F}^{(l)}$, in that order, by minimizing the corresponding objective functions using some optimization algorithm and freeze each layer after learning.

Note that this realization gives great flexibility to the architecture used: $\mathbf{F}^{(1)}$ need not be kernelized or a single layer. This allows one to apply this realization to partially kernelized NNs. The algorithm also directly applies to classification with more than two classes since the definition of $\mathbf{G}^\star$ is agnostic to the number of classes involved. As a side note, the sufficiency of expanding the kernel machines of each layer on the training sample (see Section 3) for minimizing the objective functions in Theorem 4.3 and Theorem 4.5 is trivially justified since the generalized representer theorem directly applies (Schölkopf et al., 2001).

4.2.2. Geometric Interpretation of Learning Dynamics

The hidden targets $\mathbf{F}^{(i)\star}(S_\mathbf{X})$ in this realization enjoy a straightforward geometric interpretation: examples from distinct classes are as distant as possible in the RKHS whereas examples from the same class are as concentrated as possible (see proof (B) of Theorem 4.3). Intuitively, such a representation is the "easiest" for the classification task. Further, since the hidden target is consistent for all $i < l$, the learning dynamics of this realization in a deep architecture is clear: the network maps the data sequentially through layers, with each layer trying to push apart examples from different classes while squeeze together those within the same class. In other words, each layer learns a more separable representation of the data. Eventually, the output layer works as a classifier on the final representation and since it would be "simple" after the mappings of the lower layers, the learned decision boundary would generalize better to unseen data, as suggested by the bounds above.

[4] The hidden layer has been learned and fixed when we learn the output layer, so we do not explicitly write $\mathbf{F}^{(1)}$ as an argument to $\tilde{R}_2$.

4.2.3. Accelerating the Kernelized Layers

There is a natural method to accelerate the kernelized non-input layers: the hidden targets are sparse in the sense that $\phi^{(i+1)}\big(\mathbf{F}^{(i)\star}(\mathbf{x}_m)\big) = \phi^{(i+1)}\big(\mathbf{F}^{(i)\star}(\mathbf{x}_n)\big)$ if $y_m = y_n$ and $\phi^{(i+1)}\big(\mathbf{F}^{(i)\star}(\mathbf{x}_m)\big) \neq \phi^{(i+1)}\big(\mathbf{F}^{(i)\star}(\mathbf{x}_n)\big)$ if $y_m \neq y_n$ (see the proof (B) of Theorem 4.3). Since we approximate $\mathbf{w}_j^{(i+1)}$ using $\sum_{n=1}^{N} \alpha_{nj}^{(i+1)} \phi^{(i+1)}\big(\mathbf{F}^{(i)\star}(\mathbf{x}_n)\big)$, retaining only one example from each class would result in exactly the same hypothesis class because $\Big\{\sum_{n=1}^{N} \alpha_{nj}^{(i+1)} \phi^{(i+1)}\big(\mathbf{F}^{(i)\star}(\mathbf{x}_n)\big) \,|\, \alpha_{nj}^{(i+1)} \in \mathbb{R}\Big\} = \Big\{\sum_{n=+,-} \alpha_{nj}^{(i+1)} \phi^{(i+1)}\big(\mathbf{F}^{(i)\star}(\mathbf{x}_n)\big) \,|\, \alpha_{nj}^{(i+1)} \in \mathbb{R}\Big\}$ for arbitrary $\mathbf{x}_+$, $\mathbf{x}_-$ in $S_{\mathbf{X}}$. Thus, after training a given layer, depending on how close $\mathbf{G}_i$ is to $\mathbf{G}^\star$, one may discard some of the centers for kernel machines of the next layer to speed up its training without sacrificing performance. This trick also has a regularization effect on the kernel machines since the number of trainable parameters of a kernel machine grows linearly in the number of its centers.

5. Related Works

The idea of combining connectionism with kernel method was initiated by Cho & Saul (2009). In their work, an "arc cosine" kernel was so defined as to imitate the computations performed by a one-layer MLP. Zhuang et al. (2011) extended the idea to arbitrary kernels with a focus on MKL, using an architecture similar to a two-layer kMLP. As a further generalization, Zhang et al. (2017) proposed kMLP and fully-kernelized CNN. However, they did not extend the idea to more network architectures. Scardapane et al. (2017) proposed to reparameterize each nonlinearity in an NN with a kernel expansion, resulting in a network similar to a KN but is trained with BP. Mairal et al. (2014) proposed to learn hierarchical representations by learning mappings of kernels that are invariant to irrelevant variations in images. Wilson et al. (2016) proposed to learn the covariance matrix of a Gaussian process using an NN in order to make the kernel "adaptive". It is evident that such an interpretation of "adaptive" kernels can be given to KNs as well. This idea also underlies the now standard approach of combining a deep NN with SVM for classification, which was first explored by Huang & LeCun (2006) and Tang (2013) and can be viewed as a special case of the proposed kernelization framework. In terms of the training of such hybrid systems, there are mainly two methods. The first is to apply BP to the entire model (Tang, 2013), which enjoys an optimality guarantee from BP but forces the SVM to be trained with gradient descent instead of the more efficient optimization algorithms that are usually used for SVMs. The alternative is to feed the hidden representations from a trained NN to the SVM and train the latter in the usual way (Huang & LeCun, 2006), but this practice is not theoretically solid. The proposed layer-wise learning framework serves as another alternative that combines the best of both worlds: one can train the NN and SVM separately with an optimality guarantee as that given by BP. Much works have been done to improve or substitute BP in learning a deep architecture. Most aim at improving the classical method, working as add-ons for BP. The most notable ones are perhaps the unsupervised greedy pre-training techniques proposed by Hinton et al. (2006) and Bengio et al. (2007). Among works that try to completely substitute BP, none provided a comparable optimality guarantee in theory as that given by BP. Fahlman & Lebiere (1990) pioneered the idea of greedily learn the architecture of an NN. In their work, each new node is added to maximize the correlation between its output and the residual error signal. Several authors explored the idea of approximating error signals propagated by BP locally at each layer or each node (Bengio, 2014; Carreira-Perpinan & Wang, 2014; Lee et al., 2015; Balduzzi et al., 2015; Jaderberg et al., 2016). Zhou & Feng (2017) proposed a BP-free deep architecture based on decision trees. Raghu et al. (2017) attempted to quantify the quality of hidden representations toward learning more interpretable deep architectures, sharing a motivation similar to ours.

6. Experiments

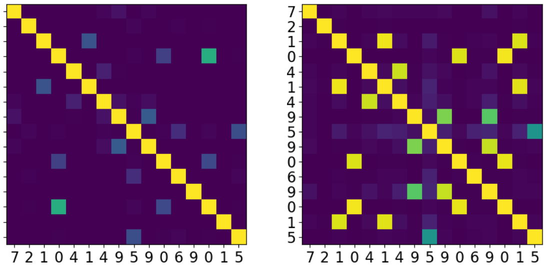
In this section, we provide empirical results to validate our theory. In the first part, we demonstrate the competence of kernelized NNs and the effectiveness of the layer-wise learning method using kMLPs. We use the proposed realization of our layer-wise framework and Adam (Kingma & Ba, 2014) as the optimization algorithm. First, we show that this realization works well with most popular loss functions. We then compare kMLPs trained with BP and the layer-wise method to show the effectiveness of the latter. Finally, to further showcase the competence of the greedily-trained kernelized models, we compare kMLPs learned layer-wise with other popular deep architectures including MLPs, Deep Belief Networks (DBNs) (Hinton & Salakhutdinov, 2006) and Stacked Autoencoders (SAEs) (Vincent et al., 2010), with the last two trained using a combination of unsupervised greedy pre-training and standard BP (Hinton et al., 2006; Bengio et al., 2007). We also visualize the learning dynamics of kMLPs and show that it is intuitive and simple to interpret. In the second part of the experiments, we partially kernelize the classic LeNet-5 (LeCun et al., 1998) and compare it with the original to validate our claim that the proposed kernelization and training framework is flexible in the sense that it works well with any given feedforward NN architecture and one can freely decide the degree of kernelization. The hidden representations learned from the two models are visualized. We show that the hidden representations learned by the kernelized model are much more discriminative than that from the original.

6.1. Part 1: Kernelizing MLPs

In terms of the datasets used. *rectangles, rectangles-image* and *convex* are binary classification datasets, *mnist (50k test)* and *mnist (50k test) rotated* are variants of MNIST. *fashion-mnist* is the Fashion-MNIST dataset (Xiao et al., 2017). These datasets all contain 28×28 grayscale images. In *rectangles, rectangles-image*, the model needs to learn if the height of the rectangle is longer than the width, and in *convex*, if the white region is convex. Examples from these datasets are shown in Fig 3 in the Appendix. In actual training, no preprocessing method was used. As for the specific kernels used, we used Gaussian kernels ($k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|_2/\sigma^2}$) for the kernelized models for all our experiments. Note that the Gaussian kernel does not strictly satisfy the condition that the infimum a is attained (see the extra assumptions before Theorem 4.5), but for practical purposes, this is not an issue due to the exponential decay of the tails of the kernel. We set $(\mathbf{G}^\star)_{mn} = 1$ if $y_m = y_n$ and 0 otherwise. To ensure that the comparisons with other models are fair, we used the cross entropy loss as the loss function for the output layer for all models. More details can be found in Appendix A.

We first test the effect of using different hidden loss functions using a two-hidden-layer kMLP. The three hidden layer loss functions tested include the proposed L^1-distance, the L^2-distance and the empirical alignment (Cristianini et al., 2002) between $\mathbf{G}_i$ and $\mathbf{G}^\star$, where i is the hidden layer being optimized. The definition of these two matrices can be found in our characterization of $\mathbf{F}^{(1)\star}$ under an L^1-type loss. All three losses quantify the dissimilarity between matrices and are commonly used in practice. On *convex*, this kMLP achieved a test error rate of 19.36%, 18.53% and 21.70% using alignment, L^2 and L^1 as the hidden losses, respectively. As a baseline, our best two-hidden-layer MLP achieved an error rate of 23.28% on this dataset. For the rest of our experiments, we use the best result from using these three hidden losses for our greedily-trained models.

We now test the layer-wise learning method against BP using the standard MNIST dataset (LeCun et al., 2010). Results from several MLPs were added as baselines. These models were trained with Adam or RMSProp (Tieleman & Hinton, 2012) and extra training techniques such as dropout (Srivastava et al., 2014) and batch normalization (BN) (Ioffe & Szegedy, 2015) were applied to boost performance. kMLPs accelerated using the proposed method (kMLP$^{\text{FAST}}$) were also tested, for which we randomly discarded some centers of each non-input layer before its training. Two popular acceleration methods for kernel machines were compared, including using a parametric representation (kMLP$^{\text{PARAM}}$), i.e., for each node in a kMLP, $f(\mathbf{x}) = \sum_{n=1}^{m} k(\mathbf{w}_n, \mathbf{x})$, $\mathbf{w}_n$ learnable, and using random Fourier features (kMLP$^{\text{RFF}}$) (Rahimi & Recht, 2008).



(a) Kernel matrix of the first hidden layer (epoch 25).

(b) Kernel matrix of the second hidden layer (epoch 15).

Figure 1. Visualizing the learning dynamics in a two-hidden-layer kMLP. Each entry in the kernel matrices corresponds to the inner product between the learned representations of two examples in the RKHS. The labels are marked on the two axes. The examples used to produce this figure are provided in the Appendix (Fig. 4a) in the order of the labels plotted here. The darker the entry, the more distant the learned representations are in the RKHS.

Results in Table 1 validate the optimality guarantee of our layer-wise framework. For both the single-hidden-layer and the two-hidden-layer kMLPs, the layer-wise algorithm consistently outperformed BP. The layer-wise method is also much faster than BP. In fact, it is practically impossible to use BP to train kMLP with more than two hidden layers without any acceleration method due to the computational complexity involved. Moreover, it is worth noting that the proposed acceleration trick is clearly very effective despite its simplicity and even produced models outperforming the original ones, which may be due to its regularization effect. This shows that kMLP together with the greedy learning scheme can be of practical interest even when dealing with the massive data sets in today's machine learning.

From Table 2, we see that the performance of kMLP is on par with some of the most popular and most mature deep architectures. In particular, the greedily-trained kMLPs compared favorably with their direct NN equivalents, i.e., the MLPs, even though neither batch normalization nor dropout was used for the former.

In Fig. 1, we visualize the learning dynamics within a two-hidden-layer kMLP learned layer-wise. Since by construction of the Gaussian kernel, the image vectors are all of unit norm in the RKHS, we can visualize the distance between two vectors by visualizing the value of their inner product. In Fig. 1b, we can see that while the image vectors are distributed randomly prior to training (not shown here, see Fig. 4c in the Appendix), there is a clear pattern in their distribution after training that reflects the dynamics of training: the layer-wise method squeezes examples from the same class closer together while pushes examples from different class farther apart. And it is intuitive that such a representation would be simple to classify. Fig. 1a and

Table 1. Testing the proposed layer-wise learning and acceleration method on MNIST. If applicable, the numbers following the model names indicate the number of hidden layers used. For kMLP$^{\text{FAST}}$, we also include in parentheses the ratio between the number of training examples randomly chosen as centers for the kernel machines in the given layer and the size of the training set. Apart from kMLP-2 (BP), the BP kMLP results are from (Zhang et al., 2017). For all tables in this paper, the entries correspond to test errors (%) and 95% confidence intervals (%). Results with overlapping confidence intervals are considered equally good. Best results are marked in bold.

MLP-1 (RMSProp+BN)	MLP-1 (RMSProp+dropout)	MLP-2 (RMSProp+BN)	MLP-2 (RMSProp+dropout)	kMLP-1 (BP)	kMLP-1 (greedy)	kMLP-1$^{\text{RFF}}$ (BP)
2.05 ± 0.28	1.77 ± 0.26	**1.58 ± 0.24**	1.67 ± 0.25	3.44 ± 0.36	1.77 ± 0.26	2.01 ± 0.28

kMLP-1$^{\text{param}}$ (BP)	kMLP-1$^{\text{FAST}}$ (greedy)	kMLP-2 (BP)	kMLP-2 (greedy)	kMLP-2$^{\text{RFF}}$ (BP)	kMLP-2$^{\text{param}}$ (BP)	kMLP-2$^{\text{FAST}}$ (greedy)
1.88 ± 0.27	1.75 ± 0.26 (0.54)	3.66 ± 0.37	**1.56 ± 0.24**	1.92 ± 0.27	2.45 ± 0.30	**1.47 ± 0.24 (1/0.19)**

Table 2. Comparing kMLPs (trained fully layer-wise) with MLPs and other popular deep architectures trained with BP and BP enhanced by unsupervised greedy pre-training. The MLP-1 (SGD), DBN and SAE results are from (Larochelle et al., 2007).

	rectangles	rectangles-image	convex	mnist (50k test)	mnist (50k test) rotated	fashion-mnist
MLP-1 (SGD)	7.16 ± 0.23	33.20 ± 0.41	32.25 ± 0.41	4.69 ± 0.19	18.11 ± 0.34	15.47 ± 0.71
MLP-1 (Adam)	5.37 ± 0.20	28.82 ± 0.40	30.07 ± 0.40	4.71 ± 0.19	18.64 ± 0.34	12.98 ± 0.66
MLP-1 (RMSProp+BN)	5.37 ± 0.20	23.81 ± 0.37	28.60 ± 0.40	4.57 ± 0.18	18.75 ± 0.34	14.55 ± 0.69
MLP-1 (RMSProp+dropout)	5.50 ± 0.20	23.67 ± 0.37	36.28 ± 0.42	4.31 ± 0.18	14.96 ± 0.31	12.86 ± 0.66
MLP-2 (SGD)	5.05 ± 0.19	**22.77 ± 0.37**	25.93 ± 0.38	5.17 ± 0.19	18.08 ± 0.34	12.94 ± 0.66
MLP-2 (Adam)	4.36 ± 0.18	25.69 ± 0.38	25.68 ± 0.38	4.42 ± 0.18	17.22 ± 0.33	**11.48 ± 0.62**
MLP-2 (RMSProp+BN)	4.22 ± 0.18	23.12 ± 0.37	23.28 ± 0.37	3.57 ± 0.16	13.73 ± 0.30	11.51 ± 0.63
MLP-2 (RMSProp+dropout)	4.75 ± 0.19	23.24 ± 0.37	34.73 ± 0.42	3.95 ± 0.17	13.57 ± 0.30	**11.05 ± 0.61**
DBN-1	4.71 ± 0.19	23.69 ± 0.37	19.92 ± 0.35	3.94 ± 0.17	14.69 ± 0.31	N/A
DBN-3	2.60 ± 0.14	**22.50 ± 0.37**	**18.63 ± 0.34**	3.11 ± 0.15	**10.30 ± 0.27**	N/A
SAE-3	2.41 ± 0.13	24.05 ± 0.37	**18.41 ± 0.34**	3.46 ± 0.16	**10.30 ± 0.27**	N/A
kMLP-1	**2.24 ± 0.13**	23.29 ± 0.37	19.15 ± 0.34	**3.10 ± 0.15**	11.09 ± 0.28	11.72 ± 0.63
kMLP-1$^{\text{FAST}}$	**2.36 ± 0.13 (0.05)**	23.86 ± 0.37 (0.01)	20.34 ± 0.35 (0.17)	**2.95 ± 0.15 (0.1)**	12.61 ± 0.29 (0.1)	**11.45 ± 0.62 (0.28)**
kMLP-2	**2.24 ± 0.13**	23.30 ± 0.37	**18.53 ± 0.34**	3.16 ± 0.15	**10.53 ± 0.27**	**11.23 ± 0.62**
kMLP-2$^{\text{FAST}}$	**2.21 ± 0.13 (0.3/0.3)**	23.24 ± 0.37 (0.01/0.3)	19.32 ± 0.35 (0.005/0.03)	3.18 ± 0.15 (0.3/0.3)	10.94 ± 0.27 (0.1/0.7)	**10.85 ± 0.61 (1/0.28)**

1b suggest that this greedy, layer-wise approach still learns "deep" representations: the higher-level representations are more distinctive for different digits than the lower-level ones. Moreover, since learning becomes increasingly simple for the upper layers as the representations become more and more well-behaved, these layers are usually easy to set up and converge very fast during training.

6.2. Part 2: Kernelizing the Classic LeNet-5

We kernelize the output layer of the classic LeNet-5 (LeCun et al., 1998) architecture and train it layer-wise. Since we are interested in evaluating the layer-wise method on partially-kernelized NNs instead of achieving state-of-the-art performance, we use the original LeNet-5 without increasing the size of any layer or the number of layers. ReLU (Glorot et al., 2011) and max pooling were used as activations and pooling layers, respectively. Both models were optimized using Adam. The two networks were trained and tested on the unpreprocessed MNIST, Fashion-MNIST and CIFAR-10 (Krizhevsky & Hinton, 2009) datasets.

In Table 3, the results suggest that kernelization and the layer-wise method resulted in performance increase in all datasets. Fig. 2 provides more insights into the difference of kLeNet-5 and LeNet-5, in which we plotted the activations of the last hidden layer of the two models after PCA dimension reduction using the MNIST test set. In particular, we see that the representations in the last hidden layer of kLetNet-5 are much more discriminative for different digits than that in LeNet-5. This observation potentially suggests that the layer-wise method turns deep architectures into more efficient representation learners.

Table 3. Kernelizing the output layer of the classic LeNet-5. The kernelized model (kLeNet-5) was trained layer-wise.

	mnist	fashion-mnist	cifar-10
LeNet-5	**0.76 ± 0.17**	9.34 ± 0.57	**36.42 ± 0.94**
kLeNet-5	**0.75 ± 0.17**	**8.67 ± 0.55**	**35.87 ± 0.94**

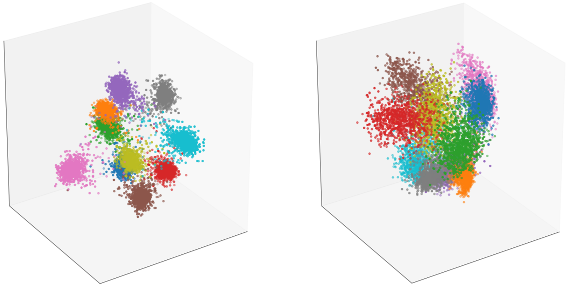


Figure 2. Visualizing the data representation of the MNIST test set in the last hidden layer of kLeNet-5 (left) and LeNet-5 (right). Each color corresponds to a digit. Representations learned by kLeNet-5 are more discriminative for different digits.

References

Balduzzi, D., Vanchinathan, H., and Buhmann, J. M. Kickback cuts backprop's red-tape: Biologically plausible credit assignment in neural networks. In *AAAI*, pp. 485–491, 2015.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bengio, Y. How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv preprint arXiv:1407.7906*, 2014.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pp. 153–160, 2007.

Carreira-Perpinan, M. and Wang, W. Distributed optimization of deeply nested systems. In *Artificial Intelligence and Statistics*, pp. 10–19, 2014.

Cho, Y. and Saul, L. K. Kernel methods for deep learning. In *Advances in neural information processing systems*, pp. 342–350, 2009.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. S. On kernel-target alignment. In *Advances in neural information processing systems*, pp. 367–373, 2002.

Fahlman, S. E. and Lebiere, C. The cascade-correlation learning architecture. In *Advances in neural information processing systems*, pp. 524–532, 1990.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science*, 313 (5786):504–507, 2006.

Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural computation*, 18 (7):1527–1554, 2006.

Huang, F. J. and LeCun, Y. Large-scale learning with svm and convolutional for generic object categorization. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pp. 284–291. IEEE, 2006.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., Silver, D., and Kavukcuoglu, K. Decoupled neural interfaces using synthetic gradients. *arXiv preprint arXiv:1608.05343*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480. ACM, 2007.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 498–515. Springer, 2015.

Mairal, J., Koniusz, P., Harchaoui, Z., and Schmid, C. Convolutional kernel networks. In *Advances in neural information processing systems*, pp. 2627–2635, 2014.

McCulloch, W. S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Micchelli, C. A., Xu, Y., and Zhang, H. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

Park, J. and Sandberg, I. W. Universal approximation using radial-basis-function networks. *Neural computation*, 3 (2):246–257, 1991.

Paszke, A., Gross, S., Chintala, S., and Chanan, G. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, 2017.

Pisier, G. *The volume of convex bodies and Banach space geometry*, volume 94. Cambridge University Press, 1999.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pp. 6076–6085, 2017.

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

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–538, 1986.

Scardapane, S., Van Vaerenbergh, S., Totaro, S., and Uncini, A. Kafnets: kernel-based non-parametric activation functions for neural networks. *arXiv preprint arXiv:1707.04035*, 2017.

Schölkopf, B. and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Schölkopf, B., Herbrich, R., and Smola, A. J. A generalized representer theorem. In *Computational learning theory*, pp. 416–426. Springer, 2001.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Sun, S., Chen, W., Wang, L., Liu, X., and Liu, T.-Y. On the depth of deep neural networks: A theoretical view. In *AAAI*, pp. 2066–2072, 2016.

Tang, Y. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *Artificial Intelligence and Statistics*, pp. 370–378, 2016.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Zhang, S., Li, J., Xie, P., Zhang, Y., Shao, M., Zhou, H., and Yan, M. Stacked kernel network. *arXiv preprint arXiv:1711.09219*, 2017.

Zhou, Z.-H. and Feng, J. Deep forest: Towards an alternative to deep neural networks. *arXiv preprint arXiv:1702.08835*, 2017.

Zhuang, J., Tsang, I. W., and Hoi, S. C. Two-layer multiple kernel learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 909–917, 2011.

Supplementary Material for Learning Backpropagation-Free Deep Architectures with Kernels

A. Experimental Setup and Additional Figures

The data set *rectangles* has 1000 training images, 200 validation images and 50000 test images. The model is required to tell if a rectangle contained in an image has a larger width or length. The location of the rectangle is random. The border of the rectangle has pixel value 255 and pixels in the rest of an image all have value 0. *rectangles-image* is the same as *rectangles* except that the inside and outside of the rectangle are replaced by an image patch, respectively. *rectangles-image* has 10000 training images, 2000 validation images and 50000 test images. *convex* consists of images in which there are white regions (pixel value 255) on black (pixel value 0) background. The model needs to tell if the region is convex. This data set has 6000 training images, 2000 validation images and 50000 test images. *mnist (50k test)* contains 10000 training images, 2000 validation images and 50000 test images taken from MNIST. *mnist (50k test) rotated* is the same as the fourth except that the digits have been randomly rotated. For detailed descriptions of the data sets, see (Larochelle et al., 2007).

The experimental setup for the greedily-trained kMLPs is as follows, kMLP-1 corresponds to a one-hidden-layer kMLP with the first layer consisting of 15 to 150 kernel machines using the same Gaussian kernel and the second layer being a single or ten (depending on the number of classes) kernel machines using another Gaussian kernel. Hyperparameters were selected using the validation set. The validation set was then used in final training only for early-stopping based on validation error. For the standard MNIST and Fashion-MNIST, the last 5000 training examples were held out as validation set. kMLP-1^{FAST} is the same kMLP for which we accelerated by randomly choosing a subset of the training set as centers for the second layer after the first had been trained. The kMLP-2 and kMLP-2^{FAST} are the two-hidden-layer kMLPs, the second hidden layers of which contained 15 to 150 kernel machines. Settings of all the kMLPs trained with BP can be found in (Zhang et al., 2017). Note that because it is extremely time/memory-consuming to train kMLP-2 with BP without any acceleration method, to make training possible, we could only randomly use 10000 examples from the entire training set of 55000 examples as centers for the kMLP-2 (BP) from Table 1.

In Table 2, we compared kMLP with a one/two-hidden-layer MLP (MLP-1/MLP-2), a one/three-hidden-layer DBN (DBN-1/DBN-3) and a three-hidden-layer SAE (SAE-3). For these models, hyperparameters were also selected using the validation set. For the MLPs, the sizes of the hidden layers were chosen from the interval [25, 700]. All hyperparameters involved in Adam, RMSProp and BN were set to the suggested default values in the corresponding papers. If used, dropout or BN was added to the hidden layers and the best probability for dropout was found using the validation set. For DBN-3 and SAE-3, the sizes of the three hidden layers varied in intervals [500, 3000], [500, 4000] and [1000, 6000], respectively. DBN-1 used a much larger hidden layer than DBN-3 to obtain comparable performance. A simple calculation shows that the total numbers of parameters in the kMLPs were fewer than those in the corresponding DBNs and SAEs by orders of magnitude in all experiments. Like in the training for the kMLPs, the validation set were also reserved for early-stopping in final training. The DBNs and SAEs had been pre-trained unsupervisedly before the supervised training phase, following the algorithms described in (Hinton et al., 2006; Bengio et al., 2007). More detailed settings for these models were reported in (Larochelle et al., 2007).

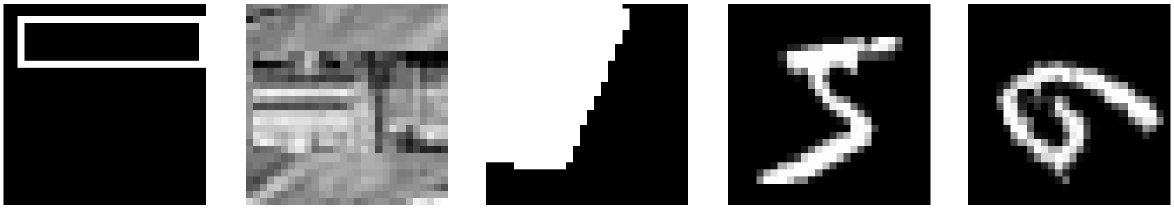
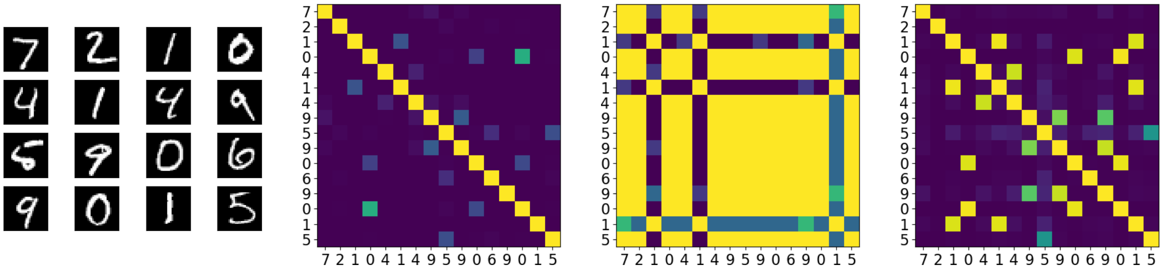


Figure 3. From left to right: example from *rectangles*, *rectangles-image*, *convex*, *mnist (50k test)* and *mnist (50k test) rotated*.



(a) Examples from test set.

(b) Kernel matrix of the first hidden layer (epoch 25).

(c) Kernel matrix of the second hidden layer (epoch 0).

(d) Kernel matrix of the second hidden layer (epoch 15).

Figure 4. Complete version of Fig. 1, in which we also include the test images and visualize the initial distribution of the representation vectors in the RKHS.

B. Proofs

Lemma B.1. *Given kernel $k : \mathbb{X}_2 \times \mathbb{X}_2 \to \mathbb{R}$, where $\mathbb{X}_2 \subset \mathbb{R}^{d_1}$. Let $\mathbb{F}_2 = \{f : \mathbb{X}_2 \to \mathbb{R},\ f(\mathbf{x}) = \sum_{\nu=1}^{m} \alpha_\nu k(\mathbf{x}_\nu,\, \mathbf{x}) + b \,|\, \boldsymbol{\alpha} = (\alpha_1,\, \dots,\, \alpha_m) \in \mathbb{R}^m,\ \|\boldsymbol{\alpha}\|_1 \leq A, b \in \mathbb{R}\}$, where the $\mathbf{x}_\nu$ are an m-subset of $S_{\mathbf{X}}$.*

Define $\mathbb{F}_1$ as in Proposition 3.2 and define $\mathbb{F}_2 \circ \mathbb{F}_1 = \{h : \mathbf{x} \mapsto \sum_{\nu=1}^{m} \alpha_\nu k(\mathbf{F}(\mathbf{x}_\nu),\, \mathbf{F}(\mathbf{x})) + b \,|\, \|\boldsymbol{\alpha}\|_1 \leq A,\, b \in \mathbb{R},\, \mathbf{F} \in \mathbb{F}_1\}$.

If the range of some element in Ω contains 0 and that Ω is closed under taking absolute value, i.e., if $f \in \Omega$, then $|f| \in \Omega$, we have

$$\mathcal{G}_N(\mathbb{F}_2 \circ \mathbb{F}_1) \leq A L d_1 \mathcal{G}_N(\Omega).$$

Proof.

$$\begin{aligned}
\hat{\mathcal{G}}_N(\mathbb{F}_2 \circ \mathbb{F}_1) &= \mathbb{E} \sup_{\boldsymbol{\alpha},\, \mathbf{F}} \left| \frac{2}{N} \sum_{n=1}^{N} \sum_{\nu=1}^{m} \alpha_\nu k(\mathbf{F}(\mathbf{x}_\nu),\, \mathbf{F}(\mathbf{x}_n)) Z_n \right| \\
&= \mathbb{E} \sup_{\boldsymbol{\alpha},\, \mathbf{F}} \frac{2}{N} \sum_{n=1}^{N} \sum_{\nu=1}^{m} \alpha_\nu k(\mathbf{F}(\mathbf{x}_\nu),\, \mathbf{F}(\mathbf{x}_n)) Z_n && \text{($\mathbb{F}_2$ is closed under negation)} \\
&\leq \mathbb{E} \sup_{\boldsymbol{\alpha},\, \mathbf{F}} \frac{2}{N} \sum_{\nu=1}^{m} |\alpha_\nu| \sum_{n=1}^{N} k(\mathbf{F}(\mathbf{x}_\nu),\, \mathbf{F}(\mathbf{x}_n)) Z_n \\
&\leq \frac{2}{N} A \mathbb{E} \sup_{\mathbf{F}} \max_{\nu} \sum_{n=1}^{N} k(\mathbf{F}(\mathbf{x}_\nu),\, \mathbf{F}(\mathbf{x}_n)) Z_n && (\|\boldsymbol{\alpha}\|_1 \leq A) \\
&= \frac{2}{N} A \mathbb{E} \sup_{\mathbf{F}} \max_{\nu} \sum_{n=1}^{N} (k(\mathbf{F}(\mathbf{x}_\nu),\, \mathbf{F}(\mathbf{x}_n)) - k(\mathbf{F}(\mathbf{x}_\nu),\, \mathbf{0})) Z_n \\
&\leq \frac{2}{N} A \mathbb{E} \sup_{\mathbf{F}} \max_{\nu} \sum_{n=1}^{N} L_{\mathbf{F}(\mathbf{x}_\nu))} \|\mathbf{F}(\mathbf{x}_n)\|_2 Z_n && \text{(Lipschitz condition on k)} \\
&\leq \frac{2}{N} A L \mathbb{E} \sup_{\mathbf{F}} \sum_{n=1}^{N} \|\mathbf{F}(\mathbf{x}_n)\|_2 Z_n && \text{(definition of L)} \\
&\leq \frac{2}{N} A L \mathbb{E} \sup_{\mathbf{F}} \sum_{n=1}^{N} \|\mathbf{F}(\mathbf{x}_n)\|_1 Z_n \\
&\leq \frac{2}{N} A L d_1 \mathbb{E} \sup_{f} \sum_{n=1}^{N} |f(\mathbf{x}_n)| Z_n \\
&\leq \frac{2}{N} A L d_1 \mathbb{E} \sup_{f} \sum_{n=1}^{N} f(\mathbf{x}_n) Z_n && \text{(Ω is closed under taking absolute value)} \\
&\leq A L d_1 \hat{\mathcal{G}}_N(\Omega).
\end{aligned}$$

Taking expectation with respect to the $\mathbf{X}_n$ finishes the proof. □

Proof of Proposition 3.2. It is trivial to check that the hypothesis class of each layer satisfies the conditions on Ω in Lemma B.1. Then the result follows from repeatedly applying Lemma B.1. □

Proof of Lemma 4.1. Let $\mathbf{G}^{(1)} = \arg\min_{\mathbf{F}^{(1)} \in \mathbb{F}_1'} \min_{\mathbf{F}^{(2)} \in \mathbb{F}_2'} \tilde{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}\big)$, $\mathbf{G}^{(2)} = \arg\min_{\mathbf{F}^{(2)} \in \mathbb{F}_2'} \tilde{R}_2\big(\mathbf{F}^{(2)} \circ \mathbf{G}^{(1)}\big)$.

Suppose $\mathbf{F}^{(1)\star} \neq \mathbf{G}^{(1)}$,

$$\begin{aligned}
\tilde{R}_2\Big(\mathbf{G}^{(2)} \circ \mathbf{G}^{(1)}\Big) &= \min_{\mathbf{F}^{(2)} \in \mathbb{F}_2'} \tilde{R}_2\Big(\mathbf{F}^{(2)} \circ \mathbf{G}^{(1)}\Big) && \text{(definition of } \mathbf{G}^{(2)}\text{)} \\
&< \min_{\mathbf{F}^{(2)} \in \mathbb{F}_2'} \tilde{R}_2\Big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)\star}\Big) && \text{(definition of } \mathbf{G}^{(1)} \text{ and } \mathbf{F}^{(1)\star} \neq \mathbf{G}^{(1)}\text{)} \\
&= \tilde{R}_2\Big(\mathbf{F}^{(2)\star} \circ \mathbf{F}^{(1)\star}\Big). && \text{(definition of } \mathbf{F}^{(2)\star}\text{)}
\end{aligned}$$

However, this contradicts the optimality of $\mathbf{F}^{(2)\star} \circ \mathbf{F}^{(1)\star}$. □

***Proof of Theorem 4.3**.* Throughout this proof we shall drop the layer indices 1 and 2 for brevity, which will cause no confusion since the hidden layer, being a vector-valued function, will be denoted by an upper-case letter and the output layer, being real-valued, a lower-case letter.

Let $\mathbb{F}'$ denote the set of all f that achieves zero hinge loss on at least one example from each class, given that $\mathbf{F}$ satisfies Eq. 1, the idea is to first collect enough information about $\arg\min_{f \in \mathbb{F}} \tilde{R}(f \circ \mathbf{F}) =: f^\circ$ with $(\mathbf{w}^\circ, b^\circ)$ such that we can compute $\tilde{R}(f^\circ \circ \mathbf{F}) = \hat{R}(f^\circ \circ \mathbf{F}) + \tau \|\mathbf{w}^\circ\|_H$. We then show that $f^\circ \in \mathbb{F}'$. Now, for any other $\mathbf{F}'$, let $\arg\min_{f \in \mathbb{F}'} \tilde{R}(f \circ \mathbf{F}') =: f'$ with $(\mathbf{w}', b')$, we show that $\min_{f \in \mathbb{F}'} \tilde{R}(f \circ \mathbf{F}') = \hat{R}(f' \circ \mathbf{F}') + \tau \|\mathbf{w}'\|_H \geq \hat{R}(f^\circ \circ \mathbf{F}) + \tau \|\mathbf{w}^\circ\|_H = \min_{f \in \mathbb{F}'} \tilde{R}(f \circ \mathbf{F})$. The desired result then follows from Lemma 4.1. We now start the formal proof. From now on, we assume that Eq. 1 holds for $\mathbf{F}$.

First, it is easy to see that $\|\phi(\mathbf{F}(\mathbf{x}_-) - \mathbf{F}(\mathbf{x}_+))\|_H$ is maximized over all representations for all $\mathbf{x}_-$, $\mathbf{x}_+$. Moreover, we have $\phi(\mathbf{F}(\mathbf{x})) = \phi(\mathbf{F}(\mathbf{x}'))$ if $y = y'$ and $\phi(\mathbf{F}(\mathbf{x})) \neq \phi(\mathbf{F}(\mathbf{x}'))$ if $y \neq y'$: Indeed, by Cauchy-Schwarz inequality, for all $\mathbf{x}$, $\mathbf{x}' \in S_{\mathbf{X}}$, $k(\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{x}')) = \langle \phi(\mathbf{F}(\mathbf{x})), \phi(\mathbf{F}(\mathbf{x}')) \rangle_H \leq \|\phi(\mathbf{F}(\mathbf{x}))\|_H \|\phi(\mathbf{F}(\mathbf{x}'))\|_H$ and the equality holds if and only if $\phi(\mathbf{F}(\mathbf{x})) = p\phi(\mathbf{F}(\mathbf{x}'))$ for some real constant p. Using the assumption on k, namely, that $\|\phi(\mathbf{F}(\mathbf{x}))\|_H = \sqrt{c}$ for all $\mathbf{F}(\mathbf{x})$, we further conclude that the equality holds if and only if $p = 1$. And the second half of the claim follows simply from $c > a$. Thus, all examples from the $+$ and $-$ class can be viewed as one vector $\phi(\mathbf{F}(\mathbf{x}_+))$ and $\phi(\mathbf{F}(\mathbf{x}_-))$, respectively.

The hypothesis f° cannot pass both $\mathbf{F}(\mathbf{x}_+)$ and $\mathbf{F}(\mathbf{x}_-)$, i.e., $f^\circ(\mathbf{F}(\mathbf{x}_+)) = 0$ and $f^\circ(\mathbf{F}(\mathbf{x}_-)) = 0$ cannot happen simultaneously since if so, first subtract b°, rotate while keeping $\|\mathbf{w}^\circ\|_H$ unchanged and add some suitable b' to get a new f' such that $f'(\mathbf{F}(\mathbf{x}_-)) < 0$ and $f'(\mathbf{F}(\mathbf{x}_+)) > 0$, then it is easy to see that $\hat{R}(f' \circ \mathbf{F}) + \tau \|\mathbf{w}'\|_H < \hat{R}(f^\circ \circ \mathbf{F}) + \tau \|\mathbf{w}^\circ\|_H$. But by assumption on f°, this is not possible.

Now write

$$\begin{aligned}
y_+ f^\circ(\mathbf{F}(\mathbf{x}_+)) + y_- f^\circ(\mathbf{F}(\mathbf{x}_-)) &= \langle \phi(\mathbf{F}(\mathbf{x}_+)) - \phi(\mathbf{F}(\mathbf{x}_-)),\, \mathbf{w}^\circ \rangle_H \\
&= \|\phi(\mathbf{F}(\mathbf{x}_+)) - \phi(\mathbf{F}(\mathbf{x}_-))\|_H \|\mathbf{w}^\circ\|_H \cos\theta_{\mathbf{F},\,\mathbf{w}^\circ} =: \zeta.
\end{aligned} \tag{2}$$

First note that for an arbitrary $\theta_{\mathbf{F},\,\mathbf{w}^\circ}$, ζ is less than or equal to 2 since one can always adjust b° such that $y_+ f^\circ(\mathbf{F}(\mathbf{x}_+)) = y_- f^\circ(\mathbf{F}(\mathbf{x}_-))$ without changing ζ and hence having a larger ζ will not further reduce $\hat{R}(f^\circ \circ \mathbf{F})$, which is 0 when $\zeta = 2$, but will result in a larger $\|\mathbf{w}^\circ\|_H$ according to Eq. 2. On the other hand, $\theta_{\mathbf{F},\,\mathbf{w}^\circ}$ must be 0 since this gives the largest ζ with the smallest $\|\mathbf{w}^\circ\|_H$. Indeed, if f° does not satisfy $\theta_{\mathbf{F},\,\mathbf{w}^\circ} = 0$, one could always shift, rotate while keeping $\|\mathbf{w}^\circ\|_H$ fixed and then shift the hyperplane back to produce another f' with $\theta_{\mathbf{F},\,\mathbf{w}'} = 0$ and this f' results in a larger ζ if $\zeta < 2$ or the same ζ if $\zeta = 2$ but a smaller $\|\mathbf{w}'\|_H$ by rescaling. Hence $\hat{R}(f' \circ \mathbf{F}) + \tau \|\mathbf{w}'\|_H < \hat{R}(f^\circ \circ \mathbf{F}) + \tau \|\mathbf{w}^\circ\|_H$ but again, this is impossible.

Together with what we have shown earlier, we conclude that $2 \geq \zeta > 0$. Then for some $t \in \mathbb{R}$, we have

$$\hat{R}(f^\circ \circ \mathbf{F}) = \kappa \max(0,\, 1 - t) + (1 - \kappa) \max(0,\, 1 - (\zeta - t)).$$

First note that we can choose t freely while keeping $\mathbf{w}^\circ$ fixed by changing b°. If $\kappa = 1/2$, we have

$$\hat{R}(f^\circ \circ \mathbf{F}) = \begin{cases} 1 - \zeta/2, & \text{if } 1 \geq t \geq \zeta - 1 \\ \frac{1}{2}(1 + t - \zeta), & \text{if } t > 1 \\ \frac{1}{2}(1 - t), & \text{if } t < \zeta - 1. \end{cases}$$

Evidently, the last two cases both result in $\hat{R}(f^\circ \circ \mathbf{F}) > 1 - \zeta/2$ hence f° must produce a t in $[\zeta - 1,\, 1]$ and $\hat{R}(f^\circ \circ \mathbf{F}) = 1 - \zeta/2$.

Now, when $\kappa \neq 1/2$, first observe that if $1 \geq t \geq \zeta - 1$,

$$\begin{aligned}\hat{R}(f^\circ \circ \mathbf{F}) &= \kappa(1 - t) + (1 - \kappa)(t - (\zeta - 1)) \\ &= (1 - 2\kappa)t + \kappa - (1 - \kappa)(\zeta - 1).\end{aligned}$$

If $\kappa > 1/2$, $\hat{R}(f^\circ \circ \mathbf{F})$ decreases in t hence t must be 1 for f°, which implies $\hat{R}(f^\circ \circ \mathbf{F}) = (1 - \kappa)(2 - \zeta)$. Similarly, if $\kappa < 1/2$, $t = \zeta - 1$ and hence $\hat{R}(f^\circ \circ \mathbf{F}) = \kappa(2 - \zeta)$.

Now suppose $t \geq 1$, $\hat{R}(f^\circ \circ \mathbf{F}) = (1 - \kappa)(1 + t - \zeta)$, which increases in t and hence $t = 1$ and $\hat{R}(f^\circ \circ \mathbf{F}) = (1 - \kappa)(2 - \zeta)$. If $\kappa < 1/2$, since $(1 - \kappa)(2 - \zeta) > \kappa(2 - \zeta)$, this combination of κ and t contradicts the optimality assumption of f°.

If $t \leq \zeta - 1$, $\hat{R}(f^\circ \circ \mathbf{F}) = \kappa(1 - t) = \kappa(2 - \zeta)$, where the second equality is because $\hat{R}(f^\circ \circ \mathbf{F})$ decreases in t. Again, $\kappa > 1/2$ leads to a contradiction.

Combining all cases, we have

$$\begin{aligned}\hat{R}(f^\circ \circ \mathbf{F}) + \tau\|\mathbf{w}^\circ\|_H &= \min(\kappa,\, 1 - \kappa)(2 - \zeta) + \frac{\tau\zeta}{\|\phi(\mathbf{F}(\mathbf{x}_+)) - \phi(\mathbf{F}(\mathbf{x}_-))\|_H} \\ &= \min(\kappa,\, 1 - \kappa)(2 - \zeta) + \frac{\tau\zeta}{\sqrt{2(c - a)}} \\ &= 2\min(\kappa,\, 1 - \kappa) + \left(\frac{\tau}{\sqrt{2(c - a)}} - \min(\kappa,\, 1 - \kappa)\right)\zeta,\end{aligned}$$

which, by the assumption on τ, strictly decreases in ζ over $(0,\, 2]$. Hence f° must satisfy $\zeta = 2$, which implies $\hat{R}(f^\circ \circ \mathbf{F}) = 0$ and this proves $f^\circ \in \mathbb{F}'$ and we have

$$\hat{R}(f^\circ \circ \mathbf{F}) + \tau\|\mathbf{w}^\circ\|_H = \frac{\sqrt{2}\tau}{\sqrt{c - a}}.$$

Now, for any other $\mathbf{F}'$ and f'. Let $\mathbf{x}_+^{\mathbf{w}'}$, $\mathbf{x}_-^{\mathbf{w}'}$ be the pair of examples with the largest $f'(\mathbf{F}'(\mathbf{x}_+)) - f'(\mathbf{F}'(\mathbf{x}_-))$. We have

$$y_+^{\mathbf{w}'} f'\Big(\mathbf{F}'\Big(\mathbf{x}_+^{\mathbf{w}'}\Big)\Big) + y_-^{\mathbf{w}'} f'\Big(\mathbf{F}'\Big(\mathbf{x}_-^{\mathbf{w}'}\Big)\Big) = \Big\|\phi\Big(\mathbf{F}'\Big(\mathbf{x}_+^{\mathbf{w}'}\Big)\Big) - \phi\Big(\mathbf{F}'\Big(\mathbf{x}_-^{\mathbf{w}'}\Big)\Big)\Big\|_H \|\mathbf{w}'\|_H \cos\theta_{\mathbf{F}',\,\mathbf{w}'} =: \zeta'.$$

Then

$$\begin{aligned}\hat{R}(f' \circ \mathbf{F}') + \tau\|\mathbf{w}'\|_H &\geq \tau\|\mathbf{w}'\|_H \\ &= \frac{\tau\zeta'}{\big\|\phi\big(\mathbf{F}'\big(\mathbf{x}_+^{\mathbf{w}'}\big)\big) - \phi\big(\mathbf{F}'\big(\mathbf{x}_-^{\mathbf{w}'}\big)\big)\big\|_H \cos\theta_{\mathbf{F}',\,\mathbf{w}'}} \\ &\geq \frac{\tau|\zeta'|}{\big\|\phi\big(\mathbf{F}'\big(\mathbf{x}_+^{\mathbf{w}'}\big)\big) - \phi\big(\mathbf{F}'\big(\mathbf{x}_-^{\mathbf{w}'}\big)\big)\big\|_H} \\ &\geq \frac{2\tau}{\sqrt{2(c - a)}} \qquad\qquad\qquad\qquad (\text{since } f' \in \mathbb{F}') \\ &\geq \hat{R}(f^\circ \circ \mathbf{F}) + \tau\|\mathbf{w}^\circ\|_H,\end{aligned}$$

This proves the desired result. □

Lemma B.2. *Suppose $f_1 \in \mathbb{F}_1, \ldots, f_d \in \mathbb{F}_d$ are elements from sets of real-valued functions defined on all of $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_m$, where $\mathbb{X}_j \subset \mathbb{R}^d$ for all j, $\mathbb{F} \subset \mathbb{F}_1 \times \cdots \times \mathbb{F}_d$. For $\mathbf{f} \in \mathbb{F}$, define $\omega \circ \mathbf{f} : \mathbb{X}_1 \times \cdots \times \mathbb{X}_m \times \mathbb{Y} \to \mathbb{R}$ as $(\mathbf{x}_1, \ldots, \mathbf{x}_m, y) \mapsto \omega(f_1(\mathbf{x}_1), \ldots, f_d(\mathbf{x}_1), f_1(\mathbf{x}_2), \ldots, f_d(\mathbf{x}_m), y)$, where $\omega : \mathbb{R}^{md} \times \mathbb{Y} \to \mathbb{R}^+ \cup \{0\}$ is bounded and*

L-Lipschitz for each $y \in \mathbb{Y}$ with respect to the Euclidean metric on $\mathbb{R}^{md}$. Let $\omega \circ \mathbb{F} = \{\omega \circ \mathbf{f} : \mathbf{f} \in \mathbb{F}\}$. Denote the Gaussian complexity of $\mathbb{F}_i$ on $\mathbb{X}_j$ as $\mathcal{G}_N^j(\mathbb{F}_i)$, if the $\mathbb{F}_i$ are closed under negation, i.e., for all i, if $f \in \mathbb{F}_i$, then $-f \in \mathbb{F}_i$, we have

$$\mathcal{G}_N(\omega \circ \mathbb{F}) \leq 2L \sum_{i=1}^{d} \sum_{j=1}^{m} \mathcal{G}_N^j(\mathbb{F}_i). \tag{3}$$

In particular, for all j, if the $\mathbf{X}_n^j$ upon which the Gaussian complexities of the $\mathbb{F}_i$ are evaluated are sets of i.i.d. random elements with the same distribution, we have $\mathcal{G}_N^1(\mathbb{F}_i) = \cdots = \mathcal{G}_N^m(\mathbb{F}_i) \coloneqq \mathcal{G}_N(\mathbb{F}_i)$ for all i and Eq. 3 becomes

$$\mathcal{G}_N(\omega \circ \mathbb{F}) \leq 2mL \sum_{i=1}^{d} \mathcal{G}_N(\mathbb{F}_i).$$

This lemma is a generalization of a result on the Gaussian complexity of Lipschitz functions on $\mathbb{R}^k$ from (Bartlett & Mendelson, 2002). And the technique used in the following proof is also adapted from there.

Proof. For the sake of brevity, we prove the case where $m = 2$. The general case uses exactly the same technique except that the notations would be more cumbersome.

Let $\mathbb{F}$ be indexed by $\mathcal{A}$. Without loss of generality, assume $|\mathcal{A}| < \infty$. Define

$$T_\alpha = \sum_{n=1}^{N} \omega(f_{\alpha,1}(\mathbf{X}_n), \ldots, f_{\alpha,d}(\mathbf{X}'_n), y_n) Z_n;$$
$$V_\alpha = L \sum_{n=1}^{N} \sum_{i=1}^{d} (f_{\alpha,i}(\mathbf{X}_n) Z_{n,i} + f_{\alpha,i}(\mathbf{X}'_n) Z_{N+n,i}),$$

where $\alpha \in \mathcal{A}$, the $(\mathbf{X}_n, \mathbf{X}'_n)$ are a random sample of size N on $\mathbb{X}_1 \times \mathbb{X}_2$ and $Z_1, \ldots, Z_N, Z_{1,1}, \ldots, Z_{2N,d}$ are i.i.d. standard normal random variables.

Let arbitrary $\alpha, \beta \in \mathcal{A}$ be given, define $\|T_\alpha - T_\beta\|_2^2 = \mathbb{E}\,(T_\alpha - T_\beta)^2$, where the expectation is taken over the Z_n. Define $\|V_\alpha - V_\beta\|_2^2$ similarly and we have

$$\begin{aligned}\|T_\alpha - T_\beta\|_2^2 &= \sum_{n=1}^{N} (\omega(f_{\alpha,1}(\mathbf{X}_n), \ldots, f_{\alpha,d}(\mathbf{X}'_n), y_n) - \omega(f_{\beta,1}(\mathbf{X}_n), \ldots, f_{\beta,d}(\mathbf{X}'_n), y_n))^2 \\ &\leq L^2 \sum_{n=1}^{N} \sum_{i=1}^{d} \Big((f_{\alpha,i}(\mathbf{X}_n) - f_{\beta,i}(\mathbf{X}_n))^2 + (f_{\alpha,i}(\mathbf{X}'_n) - f_{\beta,i}(\mathbf{X}'_n))^2 \Big) \\ &= \|V_\alpha - V_\beta\|_2^2.\end{aligned}$$

By Slepian's lemma (Pisier, 1999) and since the $\mathbb{F}_i$ are closed under negation,

$$\begin{aligned}\frac{N}{2} \hat{\mathcal{G}}_N(\omega \circ \mathbb{F}) &= \mathbb{E}_{Z_n} \sup_{\alpha \in \mathcal{A}} T_\alpha \\ &\leq 2\mathbb{E}_{Z_{n,i},\, Z_{N+n,i}} \sup_{\alpha \in \mathcal{A}} V_\alpha \\ &\leq \frac{N}{2} 2L \sum_{i=1}^{d} \Big(\hat{\mathcal{G}}_N^1(\mathbb{F}_i) + \hat{\mathcal{G}}_N^2(\mathbb{F}_i) \Big).\end{aligned}$$

Taking the expectation of the $\mathbf{X}_n, \mathbf{X}'_n$ on both sides proves the result. □

Proof of Lemma 4.4. Normalize ℓ_2 to $[0, 1]$ by dividing $2\max(|c|, |a|)$. Then the loss function becomes

$$\ell_2\Big(\mathbf{F}^{(2)} \circ \mathbf{F}^{(1)}, (\mathbf{x}_m, y_m), (\mathbf{x}_n, y_n)\Big) = \frac{1}{2\max(|c|, |a|)} \Big| k^{(3)}\Big(\mathbf{F}^{(2)}(\mathbf{x}_m), \mathbf{F}^{(2)}(\mathbf{x}_n)\Big) - (\mathbf{G}^\star)_{mn} \Big|.$$

For each fixed $(\mathbf{G}^\star)_{mn}$,

$$\begin{aligned}
&\left|\ell_2\Big(\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)},\,(\mathbf{x}_m,\,y_m),\,(\mathbf{x}_n,\,y_n)\Big)-\ell_2\Big(\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)},\,(\mathbf{x}'_m,\,y'_m),\,(\mathbf{x}'_n,\,y'_n)\Big)\right|\\
&\quad\le\frac{1}{2\max(|c|,\,|a|)}\left|k^{(3)}\Big(\mathbf{F}^{(2)}(\mathbf{x}_m),\,\mathbf{F}^{(2)}(\mathbf{x}_n)\Big)-k^{(3)}\Big(\mathbf{F}^{(2)}(\mathbf{x}'_m),\,\mathbf{F}^{(2)}(\mathbf{x}'_n)\Big)\right| \qquad\qquad \text{(triangle inequality)}\\
&\quad\le\frac{1}{2\max(|c|,\,|a|)}\bigg(\left|k^{(3)}\Big(\mathbf{F}^{(2)}(\mathbf{x}_m),\,\mathbf{F}^{(2)}(\mathbf{x}_n)\Big)-k^{(3)}\Big(\mathbf{F}^{(2)}(\mathbf{x}_m),\,\mathbf{F}^{(2)}(\mathbf{x}'_n)\Big)\right|\\
&\qquad+\left|k^{(3)}\Big(\mathbf{F}^{(2)}(\mathbf{x}_m),\,\mathbf{F}^{(2)}(\mathbf{x}'_n)\Big)-k^{(3)}\Big(\mathbf{F}^{(2)}(\mathbf{x}'_m),\,\mathbf{F}^{(2)}(\mathbf{x}'_n)\Big)\right|\bigg)\\
&\quad\le\frac{L^{(3)}}{2\max(|c|,\,|a|)}\left(\left\|\mathbf{F}^{(2)}(\mathbf{x}_n)-\mathbf{F}^{(2)}(\mathbf{x}'_n)\right\|_2+\left\|\mathbf{F}^{(2)}(\mathbf{x}_m)-\mathbf{F}^{(2)}(\mathbf{x}'_m)\right\|_2\right)\\
&\quad\le\frac{L^{(3)}}{\max(|c|,\,|a|)}\left\|\Big(\mathbf{F}^{(2)}(\mathbf{x}_n)-\mathbf{F}^{(2)}(\mathbf{x}'_n),\,\mathbf{F}^{(2)}(\mathbf{x}_m)-\mathbf{F}^{(2)}(\mathbf{x}'_m)\Big)\right\|_2.
\end{aligned}$$

Hence ℓ_2 is $L^{(3)}/\max(|c|,\,|a|)$-Lipschitz in $\big(\mathbf{F}^{(2)}(\mathbf{x}_m),\,\mathbf{F}^{(2)}(\mathbf{x}_n)\big)$ with respect to the Euclidean metric on $\mathbb{R}^{2d_2}$ for each $(\mathbf{G}^\star)_{mn}$.

The result follows from Lemma B.2 with $m=2,\,d=d_2$ and Corollary 15 in (Bartlett & Mendelson, 2002). □

Proof of Theorem 4.5. This proof uses essentially the same idea as that of Theorem 4.3. First, assume that $\mathbf{F}^{(1)}$ satisfies Eq. 1 and let $\mathbb{F}'_2$ be the set of all $\mathbf{F}^{(2)}$ that achieves zero loss on at least one pair of examples from different classes. Define $\left(f_1^{(2)\circ},\,\dots,\,f_{d_2}^{(2)\circ}\right)=\mathbf{F}^{(2)\circ}:=\arg\min_{\mathbf{F}^{(2)}\in\mathbb{F}'_2}\tilde{R}_2\big(\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)}\big)$. Due to the complete dependence of $k^{(3)}(\mathbf{x},\,\mathbf{y})$ on $\|\mathbf{x}-\mathbf{y}\|_2$, we can rewrite $k^{(3)}\big(\mathbf{F}^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_m)\big),\,\mathbf{F}^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_n)\big)\big)$ as $h^{(3)}\big(\big\|\mathbf{F}^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_m)\big)-\mathbf{F}^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_n)\big)\big\|_2\big)$ for some $h^{(3)}$. Define $\mu_{mn}^{(2)}=\big\|\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_m)\big)-\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_n)\big)\big\|_{H_2}$, we have

$$\begin{aligned}
\hat{R}_2\Big(\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)}\Big)&=\frac{1}{N^2}\sum_{m,n=1}^{N}\left|h^{(3)}\left(\left\|\mathbf{F}^{(2)}\Big(\mathbf{F}^{(1)}(\mathbf{x}_m)\Big)-\mathbf{F}^{(2)}\Big(\mathbf{F}^{(1)}(\mathbf{x}_n)\Big)\right\|_2\right)-(\mathbf{G}^\star)_{mn}\right|\\
&=\frac{1}{N^2}\sum_{m,n=1}^{N}\left|h^{(3)}\left(\sqrt{\sum_{j=1}^{d_2}\Big(f_j^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_m)\big)-f_j^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_n)\big)\Big)^2}\right)-(\mathbf{G}^\star)_{mn}\right|\\
&=\frac{1}{N^2}\sum_{m,n=1}^{N}\left|h^{(3)}\left(\sqrt{\sum_{j=1}^{d_2}\Big(\mu_{mn}^{(2)}\Big\|\mathbf{w}_{f_j^{(2)}}\Big\|_{H_2}\cos\theta_{mn,\,\mathbf{F}^{(1)},\,\mathbf{w}_{f_j^{(2)}}}\Big)^2}\right)-(\mathbf{G}^\star)_{mn}\right|.
\end{aligned}$$

Given that Eq. 1 holds, we have shown in the proof of Theorem 4.3 that $\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x})\big)=\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}')\big)$ if $y=y'$ and $\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x})\big)\neq\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}')\big)$ if $y\neq y'$. Then we have

$$\begin{aligned}
&\hat{R}_2\Big(\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)}\Big)+\tau\max_{1\le j\le d_2}\Big\|\mathbf{w}_{f_j^{(2)}}\Big\|_{H_2}\\
&=\frac{1}{N^2}\sum_{y_m\neq y_n}\left(h^{(3)}\left(\sqrt{\sum_{j=1}^{d_2}\Big(\mu_{mn}^{(2)}\Big\|\mathbf{w}_{f_j^{(2)}}\Big\|_{H_2}\cos\theta_{\mathbf{F}^{(1)},\,\mathbf{w}_{f_j^{(2)}}}\Big)^2}\right)-a\right)+\tau\max_{1\le j\le d_2}\Big\|\mathbf{w}_{f_j^{(2)}}\Big\|_{H_2}.
\end{aligned}$$

The hypothesis $\mathbf{F}^{(2)\circ}$ must satisfy

$$\begin{aligned}
&\sqrt{\sum_{j=1}^{d_2}\Big(\mu_{mn}^{(2)}\Big\|\mathbf{w}_{f_j^{(2)\circ}}\Big\|_{H_2}\cos\theta_{\mathbf{F}^{(1)},\,\mathbf{w}_{f_j^{(2)\circ}}}\Big)^2}\le\eta;\\
&\Big\|\mathbf{w}_{f_1^{(2)\circ}}\Big\|_{H_2}=\cdots=\Big\|\mathbf{w}_{f_{d_2}^{(2)\circ}}\Big\|_{H_2}=:\|\mathbf{w}^\circ\|_{H_2};\\
&\cos\theta_{\mathbf{F}^{(1)},\,\mathbf{w}_{f_i^{(2)\circ}}}=1,\,\forall i.
\end{aligned}$$

The first observation is trivial since if otherwise, one can always reduce the largest $\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$ to obtain equality to η, this gives the same $\hat{R}_2$ with a smaller $\tau\max_j\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$, which contradicts the definition of $\mathbf{F}^{(2)\circ}$. Note that if during shrinking the largest $\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$, this element ceases to be the largest among all j, we shall continue the process with the new (two or more) largest instead. To see the rest of the claim, note that for the largest of the $\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$, we must have $\theta_{\mathbf{F}^{(1)},\,\mathbf{w}_{f_j^{(2)\circ}}} = 0$ since if not, one could shift and rotate the hyperplane and again obtain the same $\hat{R}_2$ with a smaller $\tau\max_j\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$. Reducing the largest $\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$ and increasing the second largest by scaling, one would get a smaller $\tilde{R}_2$. It is immediate that the minimal $\tilde{R}_2$ (w.r.t. only the first and second largest $\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$) is attained when the largest and the second largest $\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$ are equal. Then a similar argument as before gives $\theta_{\mathbf{F}^{(1)},\,\mathbf{w}_{f_j^{(2)\circ}}} = 0$ for both of them. The rest of the claim follows via repeatedly applying this argument to all the $\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}$.

Define

$$\left|f^{\circ}\left(\mathbf{F}^{(1)}(\mathbf{x}_+)\right) - f^{\circ}\left(\mathbf{F}^{(1)}(\mathbf{x}_-)\right)\right| = \left\|\phi^{(2)}\left(\mathbf{F}^{(1)}(\mathbf{x}_+)\right) - \phi^{(2)}\left(\mathbf{F}^{(1)}(\mathbf{x}_-)\right)\right\|_{H_2}\|\mathbf{w}^{\circ}\|_{H_2}.$$

Then we have

$$\begin{aligned}
&\hat{R}_2\left(\mathbf{F}^{(2)\circ}\circ\mathbf{F}^{(1)}\right) + \tau\max_{1\le j\le d_2}\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}\\
&\quad= \left(h^{(3)}\left(\sqrt{d_2}\left|f^{\circ}\left(\mathbf{F}^{(1)}(\mathbf{x}_+)\right) - f^{\circ}\left(\mathbf{F}^{(1)}(\mathbf{x}_-)\right)\right|\right) - a\right)\psi + \frac{\tau\left|f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_+)\big) - f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_-)\big)\right|}{\left\|\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_+)\big) - \phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x}_-)\big)\right\|_{H_2}}\\
&\quad= \left(h^{(3)}\left(\sqrt{d_2}\left|f^{\circ}\left(\mathbf{F}^{(1)}(\mathbf{x}_+)\right) - f^{\circ}\left(\mathbf{F}^{(1)}(\mathbf{x}_-)\right)\right|\right) - a\right)\psi + \frac{\tau\left|f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_+)\big) - f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_-)\big)\right|}{\sqrt{2(c-a)}}.
\end{aligned}$$

As we have shown, $\sqrt{d_2}\left|f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_+)\big) - f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_-)\big)\right| \in [0,\,\eta]$. Let $\lambda = \left|f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_+)\big) - f^{\circ}\big(\mathbf{F}^{(1)}(\mathbf{x}_-)\big)\right|$ and differentiate the r.h.s. of the above equation w.r.t. λ and using the assumption on τ, we have

$$\sqrt{d_2}\psi\frac{dh^{(3)}(t)}{dt} + \frac{\tau}{\sqrt{2(c-a)}} < 0.$$

Hence the overall risk decreases in λ over $\left[0,\,\eta/\sqrt{d_2}\right]$, which implies that $\mathbf{F}^{(2)\circ}$ must have $\lambda = \eta/\sqrt{d_2}$ and that $\mathbf{F}^{(2)\circ}\in\mathbb{F}_2'$ and we have

$$\hat{R}_2\left(\mathbf{F}^{(2)\circ}\circ\mathbf{F}^{(1)}\right) + \tau\max_{1\le j\le d_2}\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2} = \frac{\tau\eta}{\sqrt{2d_2(c-a)}}.$$

Now for any other $\mathbf{F}^{(1)\prime}$, define $\left(f_1^{(2)\prime},\,\ldots,\,f_{d_2}^{(2)\prime}\right) = \mathbf{F}^{(2)\prime} := \arg\min_{\mathbf{F}^{(2)}\in\mathbb{F}_2'}\tilde{R}_2\big(\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)\prime}\big)$. Assume without loss of generality that the largest $f_j^{(2)\prime}\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_+)\big) - f_j^{(2)\prime}\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_-)\big)$ over all j, $\mathbf{x}_+$ and $\mathbf{x}_-$ is attained at $j=1$ and write $\mathbf{w}' = \mathbf{w}_{f_1^{(2)\prime}}$, $f' = f_1^{(2)\prime}$ for convenience. Let $\mathbf{x}_+^{\mathbf{w}'}$, $\mathbf{x}_-^{\mathbf{w}'}$ be the pair with the largest $f'\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_+)\big) - f'\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_-)\big)$. Note that the assumption that $\mathbf{F}^{(2)\prime}\in\mathbb{F}_2'$ implies that $f'\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_+)\big) - f'\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_-)\big) \ge \eta/\sqrt{d_2}$. Then we have

$$\begin{aligned}
\hat{R}_2\left(\mathbf{F}^{(2)\prime}\circ\mathbf{F}^{(1)}\right) + \tau\max_{1\le j\le d_2}\left\|\mathbf{w}_{f_j^{(2)\prime}}\right\|_{H_2} &\ge \tau\|\mathbf{w}'\|_{H_2}\\
&= \frac{\tau\left|f'\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_+)\big) - f'\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_-)\big)\right|}{\left\|\phi^{(2)}\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_+)\big) - \phi^{(2)}\big(\mathbf{F}^{(1)\prime}(\mathbf{x}_-)\big)\right\|_{H_2}\left|\cos\theta_{\mathbf{F}^{(1)\prime},\,\mathbf{w}'}\right|}\\
&\ge \frac{\tau\eta}{\sqrt{2d_2(c-a)}}\\
&= \hat{R}_2\left(\mathbf{F}^{(2)\circ}\circ\mathbf{F}^{(1)}\right) + \tau\max_{1\le j\le d_2}\left\|\mathbf{w}_{f_j^{(2)\circ}}\right\|_{H_2}.
\end{aligned}$$

And the desired result follows from Lemma 4.1. □

Proof of Lemma 4.6. First, it is trivial to check that the so-defined s metric is indeed a metric. In particular, it satisfies the triangle inequality. For $i = 2, \ldots, l$,

$$\begin{aligned}
\left\|\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)} - \mathbf{F}^{(2)\star}\circ\mathbf{F}^{(1)\star}\right\|_s &\le \left\|\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)} - \mathbf{F}^{(2)}\circ\mathbf{F}^{(1)\star}\right\|_s + \left\|\mathbf{F}^{(2)}\circ\mathbf{F}^{(1)\star} - \mathbf{F}^{(2)\star}\circ\mathbf{F}^{(1)\star}\right\|_s \\
&\le \sup_{\mathbf{x}\in\mathbb{X}_2}\sqrt{\sum_{j=1}^{d_2}\left(f_j^{(2)}\circ\mathbf{F}^{(1)}(\mathbf{x}) - f_j^{(2)}\circ\mathbf{F}^{(1)\star}(\mathbf{x})\right)^2} + \epsilon_2 \\
&= \sup_{\mathbf{x}\in\mathbb{X}_2}\sqrt{\sum_{j=1}^{d_2}\left(\left\langle\mathbf{w}_j^{(2)},\, \phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x})\big) - \phi^{(2)}\big(\mathbf{F}^{(1)\star}(\mathbf{x})\big)\right\rangle_{H_2}\right)^2} + \epsilon_2 \\
&\le \sup_{\mathbf{x}\in\mathbb{X}_2}\sqrt{\sum_{j=1}^{d_2}\left(\left\|\mathbf{w}_{f_j^{(2)}}\right\|_{H_2}\left\|\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x})\big) - \phi^{(2)}\big(\mathbf{F}^{(1)\star}(\mathbf{x})\big)\right\|_{H_2}\right)^2} + \epsilon_2 \\
&= \sup_{\mathbf{x}\in\mathbb{X}_2}\left\|\phi^{(2)}\left(\mathbf{F}^{(1)}(\mathbf{x})\right) - \phi^{(2)}\left(\mathbf{F}^{(1)\star}(\mathbf{x})\right)\right\|_{H_2}\sqrt{\sum_{j=1}^{d_2}\left\|\mathbf{w}_{f_j^{(2)}}\right\|_{H_2}^2} + \epsilon_2 \\
&\le \sup_{\mathbf{x}\in\mathbb{X}_2}\sqrt{\left(L^{(2)}_{\mathbf{F}^{(1)}(\mathbf{x})} + L^{(2)}_{\mathbf{F}^{(1)\star}(\mathbf{x})}\right)\left\|\mathbf{F}^{(1)}(\mathbf{x}) - \mathbf{F}^{(1)\star}(\mathbf{x})\right\|_2}\sqrt{\sum_{j=1}^{d_2}\left\|\mathbf{w}_{f_j^{(2)}}\right\|_{H_2}^2} + \epsilon_2 \\
&\le \sqrt{2L^{(2)}\sum_{j=1}^{d_2}\left\|\mathbf{w}_{f_j^{(2)}}\right\|_{H_2}^2}\sqrt{\left\|\mathbf{F}^{(1)} - \mathbf{F}^{(1)\star}\right\|_s} + \epsilon_2,
\end{aligned}$$

where for the second to last inequality, we have written $\left\|\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x})\big) - \phi^{(2)}\big(\mathbf{F}^{(1)\star}(\mathbf{x})\big)\right\|_{H_2}$ as $\sqrt{\left\langle\phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x})\big) - \phi^{(2)}\big(\mathbf{F}^{(1)\star}(\mathbf{x})\big),\, \phi^{(2)}\big(\mathbf{F}^{(1)}(\mathbf{x})\big) - \phi^{(2)}\big(\mathbf{F}^{(1)\star}(\mathbf{x})\big)\right\rangle_{H_2}}$ and used the Lipschitz condition on $k^{(2)}$. This proves the lemma. □