Regression as Classification: Influence of Task Formulation on Neural Network Features

Lawrence Stewart
INRIA & ENS
PSL Research University
Paris

Francis Bach
INRIA & ENS
PSL Research University
Paris

Quentin Berthet
Google Brain
Paris

Jean-Philippe Vert
Google Brain
Paris

Abstract

Neural networks can be trained to solve regression problems by using gradient-based methods to minimize the square loss. However, practitioners often prefer to reformulate regression as a classification problem, observing that training on the cross entropy loss results in better performance. By focusing on two-layer ReLU networks, which can be fully characterized by measures over their feature space, we explore how the implicit bias induced by gradient-based optimization could partly explain the above phenomenon.

We provide theoretical evidence that the regression formulation yields a measure whose support can differ greatly from that for classification, in the case of one-dimensional data. Our proposed optimal supports correspond directly to the features learned by the input layer of the network. The different nature of these supports sheds light on possible optimization difficulties the square loss could encounter during training, and we present empirical results illustrating this phenomenon.

1 INTRODUCTION

Two of the most commonplace supervised learning tasks are regression and classification. The goal of the former is to predict real-valued labels for data, whilst the goal of the latter is to predict discrete labels. Regression models are conventionally trained using the squared error loss, whilst classification models are typically trained using the cross-entropy loss.

Over the past years, neural networks have notably advanced scientific capabilities for both classification and regression problems (Goodfellow et al., 2016). In addition, neural networks have desirable attributes, such as their ability to learn complex non-linear functions, as well as exhibiting adaptivity to low-dimensional supports, smoothness and latent linear sub-spaces (see Bach, 2017).

Some examples of advances in classification can be found in computer vision (Krizhevsky et al., 2012; He et al., 2016; Szegedy et al., 2016; Tan and Le, 2019) and natural language processing (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). Similarly, neural networks have achieved the state of the art on regression problems, such as pose estimation (Sun et al., 2013; Toshev and Szegedy, 2014; Belagiannis et al., 2015; Liu et al., 2016). Interestingly, it can be remarked that the amount of scientific work applying neural networks to classification tasks significantly outweighs that for regression problems.

The predictive power of neural networks does not come without drawbacks. Unlike kernel methods (Schölkopf and Smola, 2002; Berg et al., 1984), to which neural networks are closely related, there are no optimization guarantees for finite neural networks, which may become stuck in local minima of the loss function. The existence of such local minima is a consequence of the non-convexity of loss functions with respect to the weights of deep neural networks that have non-linearities between layers.

The undesirable convergence to a local minimum of a loss function typically leads to under-fitting. Local minima are often encountered in training even when the data are generated directly from a teacher neural network (Safran and Shamir, 2018). Over-parametrization can sometimes help to alleviate this problem (Neyshabur et al., 2015; Goodfellow et al., 2015), but this is not guaranteed (Nakkiran et al., 2021).

A commonly seen practise within the machine learning community is the transformation of regression problems into classification problems. Instead of training a neural network using the square loss function on the original re-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

gression problem, one instead trains the model using the cross entropy loss on a new discretized classification task. Such a reformulation can often yield better performance, despite the cross entropy loss having no notion of distance between classes.

There are several synonymous names referring to the above practise: discretizing, binning, quantizing or digitizing a regression problem. Throughout this paper, we will refer to this practise as the *binning phenomenon*. We provide some examples of literature utilizing this technique, but our list is certainly not exhaustive.

Zhang et al. (2016) found discretizing the *"ab" color-space* yielded better predictions for image colorization. Similarly, by binning the pixel space, Van Oord et al. (2016) improved upon previous regression-based approaches (Theis and Bethge, 2015; Uria et al., 2014) for generative image modelling. Reformulation of regression as classification has also led to state-of-the-art performance in the fields of age estimation (Rothe et al., 2015), pose estimation (Rogez et al., 2017), and reinforcement learning (Akkaya et al., 2019; Schrittwieser et al., 2020). The practise is also seen outside of academic research, for example in the winning solution of the NOAA Right Whale Recognition Kaggle challenge[1].

1.1 Contributions

The goal of this paper is to examine how the implicit bias obtained when training neural networks with gradient-based methods could provide one possible explanation to the binning phenomenon. In order to utilize recent results on optimization (Chizat and Bach, 2018) and implicit bias (Chizat and Bach, 2020; Boursier et al., 2022), we restrict ourselves to the case of two layer neural networks with the ReLU non-linearity (Nair and Hinton, 2010). Our contributions are the following:

- We study two simplified problems which closely relate to the implicit biases induced when training over-parameterized models on the square and cross entropy losses, in the case of one-dimensional data. In particular, we provide supports of optimal measures for both of these problems. These supports correspond directly to the features learnt by finite networks.
- We postulate that a sparse optimal support for the regression implicit bias could result in optimization difficulties, shedding light on one possible explanation for the binning phenomenon. We provide synthetic experiments which exhibit this behaviour.

The code to reproduce our experiments can be found at https://github.com/LawrenceMMStewart/Regression-as-Classification.

1.2 Limitations

Our analysis and empirical results only demonstrate the link between implicit biases and the binning phenomenon for two-layer neural networks. Experimentation showed that deeper models did not suffer under-fitting on our synthetic problem when trained on the square loss (see Appendix G). Secondly, the optimal supports we propose are for problems that closely resemble the implicit biases of Boursier et al. (2022); Chizat and Bach (2020). The reparameterization we invoke to simplify analysis of the feature space introduces a factor into the total variation, which for simplicity we ignore. Finally, the link between our proposed supports and optimization is only seen empirically. Producing theory to describe whether or not a regression problem will encounter optimization difficulties as a consequence of implicit biases remains a difficult open problem.

1.3 Notation

For any $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. For a vector $x \in \mathbb{R}^d$ and $l \in [d]$, let $x_{[l]} \in \mathbb{R}^l$ denote the vector consisting of the first l indices of x. Let e_j denote the j^{th} canonical basis vector of $\mathbb{R}^k$. Let $S^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$. Let $(\cdot)_+ = \max(\cdot, 0)$ denote the ReLU non-linearity, where the maximum is taken element-wise. Let Ω_* denote the dual norm of Ω, a norm on $\mathbb{R}^k$. Let $\mathbb{1}(x = v)$ denote the indicator function, taking the value of 1 if $x = v$, otherwise 0 for $x \neq v$. Let $I_S : \mathbb{R}^k \to \{0, \infty\}$ denote the characteristic function of convex set $S \subseteq \mathbb{R}^k$, where $I_S(y) = 0$ if $y \in S$, otherwise $I_S(y) = \infty$. Let σ_S denote the support function of convex set $S \subseteq \mathbb{R}^k$, defined as $\sigma_S(y) = \sup_{w \in S} w^T y$. Let $s : \mathbb{R}^k \to \mathbb{R}^k$ denote the softmax function, where $(s(v))_j = e^{v_j} / \sum_{l=1}^k e^{v_l}$.

2 FORMULATING REGRESSION AS CLASSIFICATION

Let $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times [0, 1]$ denote the train data set for a regression problem, where we have assumed without loss of generality that the labels $y_1, \ldots, y_n$ have been normalized to the unit interval. To discretize the regression data, divide the interval $[0, 1]$ into k bins with midpoints given by $\lambda \in \mathbb{R}^k$, where $0 = \lambda_1 < \cdots < \lambda_k = 1$. The new discrete labels $\tilde{y}_i \in \arg\min_{j \in [k]} |y_i - \lambda_j|$ correspond to which of the k bins each of the y_i falls into, taking the left-most bin in case of ties. Figure 1 visually depicts this process.

The newly discretized data $\{(x_i, \tilde{y}_i)\}_{i=1}^n$ can then be used to train a classifier $f : \mathbb{R}^d \to \mathbb{R}^k$. If obtaining a real-valued prediction is imperative, one can take the expected value

[1] https://deepsense.ai/deep-learning-right-whale-recognition-kaggle/

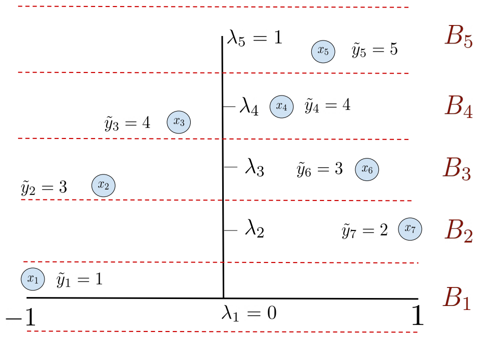


Figure 1: Depiction of binning / discretizing regression data $\{(x_i, y_i)\}_{i=1}^{6}$ using $k = 5$ bins $B_1, \ldots, B_k$, each of uniform size with midpoints $0 = \lambda_1 < \cdots < \lambda_k = 1$. Here $x_i \in [-1, 1]$ and $y_i \in [0, 1]$. The new labels $\tilde{y}_i \in [k]$ correspond to which of the k bins $B_1, \ldots, B_k$ the labels y_i fall into.

over the bins $s\,(f(x))^T\,\lambda \in \mathbb{R}$.

3 NEURAL NETWORKS

3.1 Finite Sized Neural Networks

Let $x \in \mathbb{R}^d$ be a vector whose final entry is one[2], i.e., $x_d = 1$ and $x_{[d-1]} \in \mathbb{R}^{d-1}$. Let $a = (a_1, \ldots, a_m) \in \mathbb{R}^{m\times d}$ and $b = (b_1, \ldots, b_m) \in \mathbb{R}^{m\times k}$ denote matrices which we refer to as the input layer and output layer respectively. A two-layer ReLU neural network $F_{a,b} : \mathbb{R}^d \to \mathbb{R}^k$ is defined as:

$$\forall x \in \mathbb{R}^d\,, \quad F_{a,b}(x) = \sum_{j=1}^{m} b_j (a_j^T x)_+\,. \tag{1}$$

The above equation is equivalent to the common convention of writing the linear and constant terms of the model separately:

$$F_{a,b}(x) = \sum_{j=1}^{m} b_j \big(\underbrace{a_{j,[d-1]}}_{\text{linear}}{}^T x_{[d-1]} + \underbrace{a_{j,d}}_{\text{constant}} \big)_+. \tag{2}$$

A two-layer neural network can be thought of as a model that jointly learns a set of features $\left\{(\,a_j^T\cdot)_+\right\}_{j=1}^{m}$ and a linear weighting $\{b_j\}_{j=1}^{m}$ over these features.

Since the ReLU is positively homogeneous, one can renormalize the weights $a_j \leftarrow \frac{a_j}{\|a_j\|}$ and $b_j \leftarrow b_j\,\|a_j\|$ so that $a_j \in S^{d-1}$, without affecting $F_{a,b}$. Without loss of generality, we will assume throughout that $F_{a,b}$ has layers re-normalized in such fashion.

3.2 Infinite Width Neural Networks

An extension of the above is to consider models that learn a linear weighting over the set of all features $\{(a^T\cdot)_+ : a \in S^{d-1}\}$. Such models are called infinite-width neural networks and are expressed via measures, which now take the place of the output layer b.

Let $\mathcal{M}(S^{d-1}, \mathbb{R}^k)$ be the set of signed Radon measures (Rudin, 1970; Evans and Garzepy, 1991) over S^{d-1} taking values in $\mathbb{R}^k$. An infinite width network characterized by $\nu \in \mathcal{M}(S^{d-1}, \mathbb{R}^k)$ is defined as:

$$F_\nu(x) = \int_{S^{d-1}} (a^T x)_+ d\nu(a) \quad \in \mathbb{R}^k. \tag{3}$$

The finite models described by equation (1) can also be expressed in the infinite-width form by taking $\nu^{(a,b)} = \sum_{j=1}^{m} b_j \delta_{a_j}$. With a slight abuse of notation we can write $F_{a,b} = F_{\nu^{(a,b)}}$ to represent this.

4 IMPLICIT BIAS

Gradient-based optimization methods can result in a preference for certain solutions to a problem, known as an implicit bias. Possibly the simplest example of this is logistic regression (with no regularization), where training a linear predictor on a linearly separable dataset via (stochastic) gradient descent yields a solution that converges to the max-margin solution (Soudry et al., 2018, Theorem 3). Similar results hold for least-squares linear regression (Gunasekar et al., 2018a).

The implicit bias of both linear neural networks (Gunasekar et al., 2018b; Ji and Telgarsky, 2019; Nacson et al., 2019) and homogeneous neural networks (Lyu and Li, 2020; Chizat and Bach, 2020) has been studied for models trained to minimize a classification loss function with exponential tails, such as the cross entropy and exponential loss. Similar results exist for finite width two-layer networks trained with the square loss on regression problems (Boursier et al., 2022).

4.1 Regression

Let $x_1, \ldots, x_n \in \mathbb{R}^d$ be data with labels $y_1, \ldots, y_n \in \mathbb{R}$. With assumptions on the data[3], Boursier et al. (2022, Section 3.2) show that the gradient flow for a two-layer ReLU network trained on the square loss converges to a measure solving the following problem:

$$\begin{aligned} \inf_{\nu \in \mathcal{M}(S^{d-1}, \mathbb{R})} \quad & \int_{S^{d-1}} |d\nu(a)| \\ \text{subject to} \quad & F_\nu(x_i) = y_i \quad \forall i \in [n]. \end{aligned} \tag{4}$$

[2] This notation combines the *constant terms* of neural networks with the parameters (instead of treating them separately) by appending one to the data vector.

[3] Whilst the implicit bias for models trained on the square loss is observed empirically in experiments, the proof is restricted only to the case of orthonormal data.

For finite sized neural networks, this implicit bias selects networks which have minimum ℓ_1-norm on their output layer from the set of all networks achieving zero square loss on the train set.

4.2 Classification

Let $x_1, \dots, x_n \in \mathbb{R}^d$ be data with discrete labels $y_1, \dots, y_n \in [k]$. Extending Chizat and Bach (2020, Theorems 3 and 5) from the logistic to soft-max loss (which corresponds to using Theorem 7 from Soudry et al. (2018) instead of Theorem 3, see also Appendix A), the gradient flow for an infinitely sized neural network trained on the cross entropy loss (multi-class classification) converges to a solution of:

$$\begin{aligned} &\inf_{\nu \in \mathcal{M}(S^{d-1}, \mathbb{R}^k)} && \int_{S^{d-1}} \|d\nu(u)\| \\ &\text{subject to} && (e_{y_i} - e_l)^T F_\nu(x_i) \geq \mathbb{1}(y_i \neq l), \\ & && \forall i \in [n], \quad \forall l \in [k]. \end{aligned} \tag{5}$$

From the viewpoint of finite networks, the above implicit bias selects models whose output layer weight matrix is of minimum ℓ_1/ℓ_2 group norm (Bach et al., 2012, Section 1.3) from the set of all networks who satisfy a hard-margin constraint on class predictions for the train set.

5 RE-PARAMETERIZATION

5.1 Change of Variable

In this section, we re-parameterize the feature space S^1 of the infinite-width networks described in equation (3), in the case of one-dimensional data. This allows us to study simplified problems that are closely related to problems (4) and (5).

For the case of real-valued data $x \in \mathbb{R}$, we modify the notation of equation (3) to write:

$$F_\nu(x) = \int_{S^1} (a_1 x + a_2)_+ \, d\nu(a_1, a_2). \tag{6}$$

Each input weight $(a_1, a_2) \in S^1$ corresponds to a feature $\psi_a(x) = (a_1 x + a_2)_+$, which is piece-wise linear with slope a_1 at the 'active part' of the ReLU. We note that the two poles $(0, 1)$ and $(0, -1)$ correspond to the constant features $\psi_{(0,1)}(x) = 1$ and $\psi_{(0,-1)}(x) = 0$. Defining $\tilde{S^1} = S^1 \backslash \{(0, 1), \, (0, -1)\}$, we can hence rewrite equation (6) as:

$$F_\nu(x) = \int_{\tilde{S^1}} (a_1 x + a_2)_+ \, d\nu(a_1, a_2) \; + \; \nu\left((0, 1)\right). \tag{7}$$

For the sake of simplicity, we restrict our analysis to the set of measures $\mathcal{M}(\tilde{S^1}, \mathbb{R}^k)$, which corresponds to the same set of neural networks as $\mathcal{M}(S^1, \mathbb{R}^k)$, up to a constant. We will later see through the proofs of Section 6 that such a simplification is indeed permitted; for the implicit bias problems we will study, any missing constants $\nu((0, 1))$ only lead to changes in the weightings of boundary features of the re-parameterized feature space.

The rough idea behind our re-parameterization is to utilise the positive-homogeneity of the ReLU to normalize the input-layer weights by the slope magnitude of their corresponding features. After re-parameterization, all features will have slopes of unit magnitude. This simplifies analysis, as the slopes of piece-wise linear segments of finite neural networks will now be controlled entirely by the network's output layer.

More formally, let $\mathbb{W} = \{-1, 1\} \times \mathbb{R}$, and consider the Borel measurable function:

$$\begin{aligned} G: \quad & \tilde{S^1} && \longrightarrow && \mathbb{W} \\ & (a_1, a_2) && \longmapsto && \left(\tfrac{a_1}{|a_1|}, -\tfrac{a_2}{|a_1|} \right). \end{aligned} \tag{8}$$

We denote the re-parameterized input-layer weights as $(s, c) := G((a_1, a_2))$, and perform the following change of variable using G:

$$\begin{aligned} F_\nu(x) &= \int_{\tilde{S^1}} (a_1 x + a_2)_+ d\nu(a_1, a_2) \\ &= \int_{\tilde{S^1}} \left(\frac{1}{|a_1|}(a_1 x + a_2) \right)_+ |a_1| d\nu(a_1, a_2) \\ &= \int_{\mathbb{W}} \left(s(x - c) \right)_+ \, d\mu(s, c) \; := \; f_\mu(x), \end{aligned} \tag{9}$$

where μ is the push-forward measure of $|a_1| d\nu(a_1, a_2)$ by G. The above change of variable defines a natural mapping:

$$\begin{aligned} T: \mathcal{M}(\tilde{S^1}, \mathbb{R}^k) && \longrightarrow && \mathcal{M}(\mathbb{W}, \mathbb{R}^k) \\ \nu && \longmapsto && \mu, \end{aligned} \tag{10}$$

where $\mu = T(\nu) \implies F_\nu = f_\mu$. One can think of c as the critical point or 'kink' of an input weight, that is the point of discontinuity in the ReLU of the corresponding feature. s can be thought of as the sign of the feature; when $s = 1$ / -1 the feature ramps rightwards / leftwards.

For short-hand, we write $u = (s, c)$ and abbreviate the re-parameterized features as:

$$\phi_u(x) = \left(s(x - c) \right)_+ . \tag{11}$$

Figure 2 depicts an example of features in the re-parameterized space $\mathbb{W}$. Using the above notation, we can write:

$$f_\mu(x) = \int_{\mathbb{W}} \phi_u(x) \, d\mu(u). \tag{12}$$

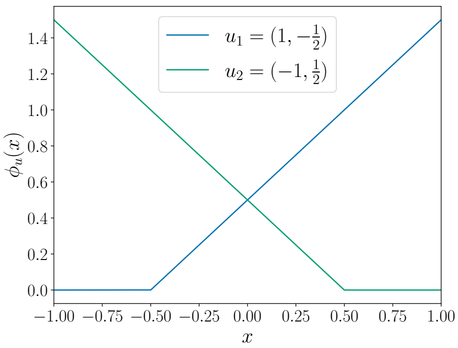


Figure 2: A depiction of two features ϕ_{u_1} and ϕ_{u_2}, where $u_1 = (1, -\frac{1}{2})$ is a right-ramping feature with kink at $\frac{-1}{2}$ and $u_2 = (-1, \frac{1}{2})$ is a left-ramping feature with kink at $\frac{1}{2}$.

5.2 Simplified Implicit Biases

Without loss of generality[4], we restrict our analysis to $\mathcal{M}(\mathbb{U}, \mathbb{R}^k)$, where $\mathbb{U} = \{-1, 1\} \times [-1, 1]$. Let $-1 = x_1 < \cdots < x_n = 1$ be ordered, real-valued data; such data can be obtained by max-min re-scaling. We define the following two problems:

Regression:

$$\begin{aligned} \inf_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R})} \quad & \int_{\mathbb{U}} |d\mu(u)| \\ \text{subject to} \quad & f_\mu(x_i) = y_i \quad \forall i \in [n]. \end{aligned} \tag{13}$$

Classification:

$$\begin{aligned} \inf_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R}^k)} \quad & \int_{\mathbb{U}} \|d\mu(u)\| \\ \text{subject to} \quad & (e_{y_i} - e_l)^T f_\mu(x_i) \geq \mathbb{1}(y_i \neq l), \\ & \forall i \in [n], \quad \forall l \in [k]. \end{aligned} \tag{14}$$

The above problems correspond to the implicit biases of equations (4) (regression) and (5) (classification). Despite the equivalence $f_{T(\nu)} = F_\nu$, problems (13) and (14) are different due to the factor of $|a_1|$ introduced into the total-variation when performing the change of variable. However, problems (13) and (14) are easier to work with, as discussed in Section 5.

6 OPTIMAL SUPPORTS

Regression Support. Let $-1 = x_1 < \cdots < x_n = 1$ be ordered data with corresponding real labels $y_1, \ldots, y_n \in \mathbb{R}$. We define:

$$R_{reg} = \{x_1, x_n\} \cup \left\{ x_i \; : \; \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \neq \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \right\}.$$

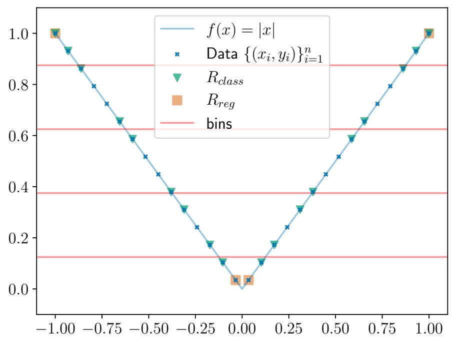


Figure 3: R_{reg} and R_{class} for regression data taken from the function $x \mapsto |x|$.

In words, R_{reg} contains $\{x_1, x_n\}$ and any points which lie at the meeting of two line segments of the piece-wise interpolant for the data $\{(x_i, y_i)\}_{i=1}^n$. A visual example of R_{reg} can be found in Figure 3. We further define $F_{reg} = \{-1, 1\} \times R_{reg}$ as the set of input weights whose features have kinks located at points appearing in R_{reg}.

Classification Support. Let $-1 = x_1 < \cdots < x_n = 1$ be ordered data with corresponding discrete labels $y_1, \ldots, y_n \in [k]$. We define the set R_{class} as follows:

$$R_{class} = \{x_1, x_n\} \cup \{x_i : y_{i-1} \neq y_i \text{ or } y_{i+1} \neq y_i\}.$$

In words, R_{class} contains $\{x_1, x_n\}$ and all other x_i which have a differing label from either of its two adjacent neighbours in the sequence $(x_i)_{i=1}^n$. An example of R_{class} is depicted in Figure 3. Similarly, we define $F_{class} = \{-1, 1\} \times R_{class}$ as the set of input weights whose features have kinks located at points appearing in R_{class}.

We are now ready to state our main theoretical result, which shows how the implicit biases of regression (13) and classification (14) differ in support.

Theorem 6.1. *For real-valued, ordered data $-1 = x_1 < \cdots < x_n = 1$:*

1. *There exists $\mu \in \mathcal{M}(\mathbb{U}, \mathbb{R})$ with supp$(\mu) \subseteq F_{reg}$ which is optimal for problem (13) with labels $y_1, \ldots, y_n \in \mathbb{R}$.*
2. *There exists $\nu \in \mathcal{M}(\mathbb{U}, \mathbb{R}^k)$ with supp$(\nu) \subseteq F_{class}$ which is optimal for problem (14) with labels $y_1, \ldots, y_n \in [k]$.*

Remark: The optimal support R_{reg} depends completely on the data set $\{(x_i, y_i)\}_{i=1}^n$, whilst R_{class} depends both on the data and the number of bins k used for discretization. In general, by increasing k, one can increase the size of the R_{class}[5]. This additional dependence on k gives R_{class} the

[4] Considering $\mathbb{U}$ over $\mathbb{W}$ is purely a syntactic preference in order to keep the kinks of features within the unit interval; all results and proofs generalize to $\mathbb{W}$.

[5] Excluding trivial problems, for example, when the target regression function is very close to being constant.

potential to include more points than R_{reg}. It is not hard to think of simple regression problems for which R_{reg} is sparse amongst $\{x_1, \dots, x_n\}$, but where R_{class} is not (for a suitable choice of k). We will explore this idea further in Section 7, and its relationship to the binning phenomenon.

6.1 An Outline for the Proof of Theorem 6.1

1. We begin in Section 6.2 by introducing a general optimization problem (15) which encompasses both problems (13) and (14). We derive the dual of this problem in Lemma 6.2.
2. Let $U_X = \{-1, 1\} \times \{x_1, \dots, x_n\}$ be the set of features having kinks at position of the data. We aim to show there exists optimal measures for problems (13) and (14), whose supports are subsets of U_X. The proof of this Proposition is broken into smaller results:
   (a) In Lemma 6.3 we derive a sufficient condition for dual feasibility to problem (15).
   (b) We show Corollary 6.3.1, which states that the existence of a feasible measure with support in U_X is a sufficient condition for the existence of an optimal measure for problem (15), having support in U_X.
   (c) We construct feasible measures for both both problems (13) and (14) in order to apply Corollary 6.3.1 and conclude the proof of Proposition 1.
3. We apply Proposition 1 to problems (13) and (14), but for data sets consisting only of points in R_{reg} and R_{class}. We then extend these solutions to the complete train data set $\{x_1, \dots, x_n\}$, and show that strong duality is indeed attained, which concludes the proof of Theorem 6.1.

6.2 A Generalized Implicit Bias Problem

Let Ω be any norm on $\mathbb{R}^k$. For a family of non-empty closed convex sets $S_1, \dots, S_n \subseteq \mathbb{R}^k$ we define the following optimization problem:

$$\inf_{\mu \in \mathcal{M}(\mathbb{U}, \mathbb{R}^k)} \int_{\mathbb{U}} \Omega\left(d\mu(u)\right) + \sum_{i=1}^{n} I_{S_i}(f_\mu(x_i)). \tag{15}$$

By setting Ω to be the Euclidean norm on $\mathbb{R}^k$ and choosing k and S_i, one can recover both problems (13) and (14). More precisely, by setting $k = 1$ and $S_i = \{y_i\}$, we obtain problem (13). On the other hand, taking $k > 1$ and

$$\begin{aligned} S_i = \{v \in \mathbb{R}^k : (e_{y_i} - e_l)^T v \geq \mathbb{1}(y_i \neq l) \\ \forall i \in [n], \quad \forall l \in [k] \quad \}, \end{aligned}$$

we recover problem (14) where the data have discrete labels $y_1, \dots, y_n \in [k]$.

Lemma 6.2. *The dual of problem* (15) *is:*

$$\begin{aligned} &\sup_{\alpha_1 \dots, \alpha_n \in \mathbb{R}^k} && -\sum_{i=1}^{n} \sigma_{S_i}(\alpha_i) \\ &\textit{subject to} && \Omega_* \left(\sum_{i=1}^{n} \alpha_i \phi_u(x_i) \right) \leq 1 \quad \forall u \in \mathbb{U}. \end{aligned} \tag{16}$$

Proof. The full proof is given in Appendix B. We provide a brief outline. By Fenchel duality (Moreau, 1966), we have:

$$\sum_{i=1}^{n} I_{S_i}(f_\mu(x_i)) = \sum_{i=1}^{n} \sup_{\alpha_i \in \mathbb{R}^k} \left\{ \langle \alpha_i, f_\mu(x_i) \rangle - \sigma_{S_i}(\alpha_i) \right\},$$

The dual problem can be obtained by substituting this into problem (15) and exchanging the order of the supremum and infinum. In order to resolve the infinum, we use properties of the dual norm. □

6.2.1 Restricting the Position of Kinks to the Data

Let $U_X = \{-1, 1\} \times \{x_1, \dots, x_n\}$ denote the set of input weights whose features have kinks at $\{x_1, \dots, x_n\}$.

Proposition 1. *For real-valued, ordered data $-1 = x_1 < \dots < x_n = 1$:*

1. *There exists $\mu \in \mathcal{M}(U_X, \mathbb{R})$ which is optimal for problem* (13) *with labels $y_1, \dots, y_n \in \mathbb{R}$.*
2. *There exists $\nu \in \mathcal{M}(U_X, \mathbb{R}^k)$ which is optimal for problem* (14) *with labels $y_1, \dots, y_n \in [k]$.*

To prove Proposition 1, we show a series of lemmas that combine to give the desired result.

Lemma 6.3. *Suppose $\alpha_1, \dots, \alpha_n \in \mathbb{R}^k$ satisfy:*

$$\Omega_* \left(\sum_{i=1}^{n} \alpha_i \phi_u(x_i) \right) \leq 1 \quad \forall u \in U_X.$$

Then $\alpha_1 \dots, \alpha_n$ are feasible for problem (16).

Proof. The full proof is detailed in Appendix C, and relies on the convexity of Ω_* combined with the fact that $\sum_{i=1}^{n} \alpha_i \phi_u(x_i)$ is piece-wise affine in c for both left and right-wards ramping features. □

Corollary 6.3.1. *Suppose there exists μ feasible for the following problem:*

$$\inf_{\mu \in \mathcal{M}(U_X, \mathbb{R}^k)} \int_{\mathbb{U}} \Omega\left(d\mu(u)\right) + \sum_{i=1}^{n} I_{S_i}\left(f_\mu(x_i)\right). \tag{17}$$

Then there exists $\mu^* \in \mathcal{M}(U_X, \mathbb{R}^k)$ which is optimal for problem (15).

Proof. Let P_1, D_1 denote respectively the primal and dual values for problem (15). Similarly let P_2, D_2 denote the primal and dual values for problem (17). Since $\mathcal{M}(U_X, \mathbb{R}^k) \subset \mathcal{M}(\mathbb{U}, \mathbb{R}^k)$, it follows that $P_2 \geq P_1$ and $D_2 \geq D_1$.

Problem (17) is a norm minimization problem with convex constraints which has a feasible point μ, so it attains strong duality (Boyd et al., 2004, Chapter 5). Let (μ^*, α^*) denote any primal-dual pair which attains strong duality. By Lemma 6.3, α^* is also dual-feasible for problem (16) so $D_2 = D_1$. We conclude that:

$$P_2 \geq P_1 \geq D_1 = D_2 = P_2 \quad \implies P_2 = P_1,$$

so (μ^*, α^*) are optimal for problem (15). □

Lemma 6.4. *For real-valued, ordered data $-1 = x_1 < \cdots < x_n = 1$:*

1. *There exists $\mu \in \mathcal{M}(U_X, \mathbb{R})$ which is feasible for problem (13) with labels $y_1, \ldots, y_n \in \mathbb{R}$.*
2. *There exists $\nu \in \mathcal{M}(U_X, \mathbb{R}^k)$ which is feasible for problem (14) with labels $y_1, \ldots, y_n \in [k]$.*

Proof. The proof is constructive and detailed in Appendix D. □

Proof of Proposition 1: Follows directly from combining (6.4) and Corollary (6.3.1).

Proof of Theorem 6.1: The proof is detailed in Appendix E. We will briefly provide an outline. Consider problem (13) but for a new data set $\{(x_i, y_i)\}_{i \in R_{reg}}$. By Proposition 1, there exists a primal-dual optimal pair (μ^*, α^*) with $\text{supp}(\mu^*) \subseteq F_{reg}$ which solves problem (13). We remark that μ^* is feasible for problem (13) with the full data set $\{(x_i, y_i)\}_{i=1}^n$. It remains to show $\exists \alpha \in \mathbb{R}^n$ which is feasible and attains strong duality with μ^*. For this, we extend $\alpha^* \in \mathbb{R}^m$ to $\tilde{\alpha} \in \mathbb{R}^n$ by appending zeroes to any new entries, where $m = |R_{reg}|$. It is clear that $\tilde{\alpha}$ is dual-feasible, and by verifying that the pair $(\mu^*, \tilde{\alpha})$ attains strong duality we conclude. A similar reasoning applies to classification.

7 SYNTHETIC REGRESSION TASK

We present a simple one-dimensional toy regression task, illustrating how the optimal supports of Theorem 6.1 can induce the binning phenomenon. The regression data set is generated from a finite teacher neural network μ_T, which has 9 neurons in the hidden layer. The resulting target function $f_{\mu_T} : [-1, 1] \to [0, 1]$ is the sum of two large-scale triangles and two small-scale triangles, depicted in Figure 4. As usual with supervised learning problems, the task is to fit a model's parameters using a train set to obtain minimum square error on a separate validation set.

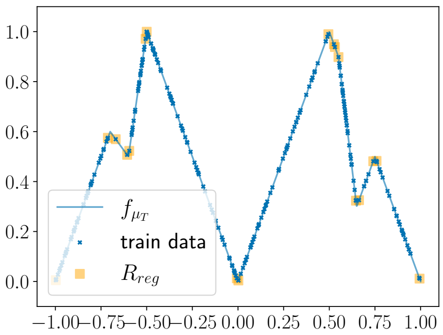


Figure 4: The train data set, consisting of 250 data points $\{(x_i, f_{\mu_T}(x_i))\}_{i=1}^{250}$, is depicted by the blue crosses. R_{reg} is depicted in orange, and is notably sparse, consisting of just 18 points. On the other-hand, discretizing the data with $k = 50$ bins results in the set R_{class} containing 230 of the data points.

7.1 Experiment Setup

Data: To generate discrete labels, we divided the y-axis into $k = 50$ bins of uniform size, so that the midpoint of the first/last bin was 0/1. For both the train and validation sets, we sampled x_i uniformly so that each bin contained the same number of points $(x_i, f_{\mu_T}(x_i))$. The train and validation data sets both consisted of 250 data points.

Models: We trained two over-parameterized models:

1. **Regression Model:** $10,000$ neurons in the hidden layer with scalar output, totalling $30,000$ weights. Trained using the square loss.
2. **Classification Model:** 500 neurons in the hidden layer with vector output of dimension $k = 50$, totalling $26,000$ weights. Trained using the cross-entropy loss.

Training: Both models were trained using gradient descent for thirty random initializations of their weights, following the scheme given by Glorot and Bengio (2010). A hyperparameter sweep was used to find the optimal learning rate for each of the models. The stopping criterion was when neither the train nor validation losses decreased from their best observed values over a duration of 1000 epochs. The final model parameters were taken from the epoch that obtained lowest square validation error. To obtain real-valued predictions from the classification model, we took the expected value over the bins as described in Section 2.

7.2 Results

The validation RMSE for the thirty random intializations is displayed in Figure 5 and Table 1. Our regression task

	RMSE $\times 10^2$			
	Best	**Worst**	**Mean**	**Std. Dev**
Regression	3.70	6.85	4.55	1.38
Classification	0.86	1.54	1.21	0.19

Table 1: Population statistics for the RMSE over 30 random initializations of model weights.

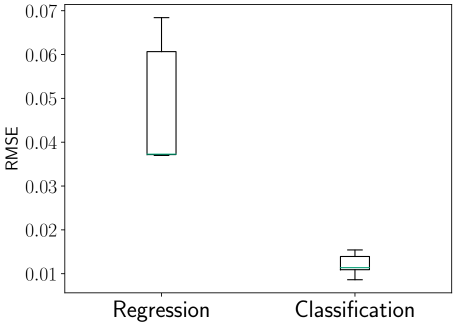


Figure 5: Population RMSE over 30 random initializations.

clearly exhibits the binning phenomenon; every classification models attained lower validation error than the best performing regression model. Moreover, the classification models were more stable to train, exhibiting less variance in performance over the thirty random initializations.

The predictions of the worst performing regression and classification models are depicted in Figures 6a and 6c respectively. It can be seen that the regression model was unable to fit the smaller-scale triangles from the train data, converging to a local minima of the square loss.

Figures 6b and 6d depict the kinks c_j corresponding to the model's input weights a_j, for both the regression and classification model respectively. The x-axis depicts the position of a feature's kink, and the y-axis expresses the norm of the corresponding output-layer weight. The support of the regression model is notably sparse, with the kinks gathering at points corresponding to to the peaks of the larger-scale triangles. The model has struggled during optimization to recover all of R_{reg}, lacking the features whose kinks are located at peaks of the smaller-scale triangles, and as a consequence suffers under-fitting.

On the other-hand, the classification model recovers a support which has features more evenly distributed across the unit interval, aligning with the optimal support R_{class} described in Theorem 6.1. As a consequence, the classification model does not suffer the same optimization problem as the regression model.

7.3 Discussion

Chizat and Bach (2018) show that in the infinite width limit, the gradient flow of a two-layer neural network converges to the global minimizer of the problem. Our experiment indicates that even simple problems can result in global convergence only being guaranteed at extreme widths.

For regression data generated from a teacher network with m_0 hidden-neurons and Gaussian weights, Safran and Shamir (2018) show that training a model with $m = m_0+1$ neurons helps alleviate under-fitting, postulating that increasing m further aids optimization. This is a clear example where regression does not suffer under-fitting, and over-parameterization aids training. Our results indicate somewhat surprisingly that the implicit bias can play a fundamental role in gradient based optimization, even for over-parameterized models and when $m >> m_0$.

Goodfellow et al. (2015) demonstrate that on a straight line between the optimal parameters and a random initialization, various over-parameterized state of the art vision models encounter no local minima. We provide evidence in Appendix F demonstrating that for even simple problems, both the regression and classification models can deviate from a linear path during optimization.

8 IMPLICIT BIAS FOR HIGHER DIMENSIONS

We provide an experiment that indicates that the properties of the supports provided in Section 6 likely apply for higher dimensions. We generated regression data $(x_i, f_{\mu_T}(x_i)) \in [-1,1]^3 \times \mathbb{R}$ from a teacher network μ_T with three neurons and random weights, where $x_3 = 1$. Similar to section 7.1, we trained over-parameterized regression and classification models on the regression data and binned data (using $k = 25$ uniform sized bins) respectively. The precise details of the experiment can be found in Appendix H.

Each feature a_j is now characterized by the line where it ramps. That is to say, the points $x \in \mathbb{R}^3$ satisfying:

$$a_{j,1}x_1 + a_{j,2}x_2 + a_{j,3} = 0,$$

where $x_3 = 1$. The critical lines characterizing the features of the regression and classification models are depicted in Figure 7a and 7b, respectively. We see that the regression model recovers a sparse support, whilst the classification model's features are more evenly distributed over unit square corresponding to (x_1, x_2). These observations are similar to R_{reg} and R_{class} in the one-dimensional case, suggesting that the difference in implicit bias between regession and classification support we identified in one-dimensional problems is likely to hold in higher dimensions.

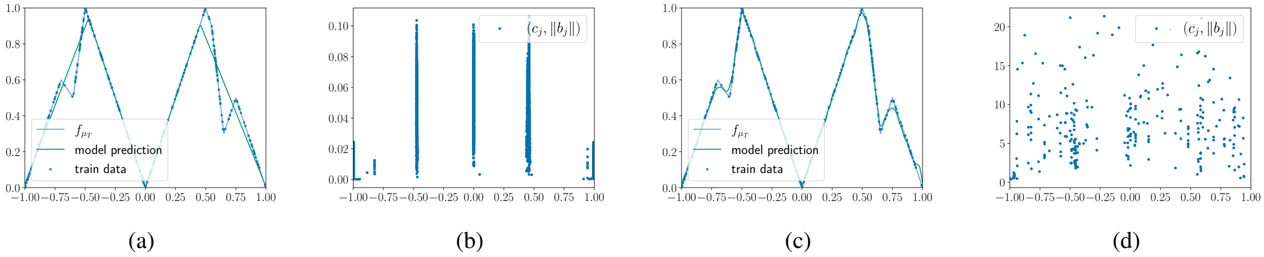


Figure 6: Predictions for the worst performing regression / classification models are depicted in Figures 6a / 6c. Supports for the worst performing regression / classification models are depicted in Figures 6b / 6d.

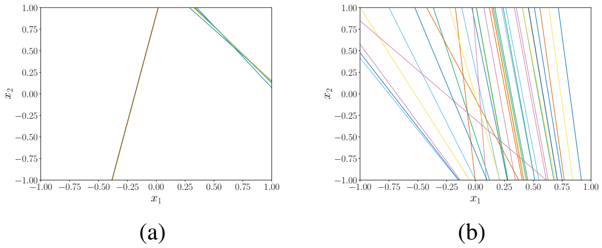


Figure 7: Critical lines of the regression model's features (left) and classification model's features (right), for two-dimensional input data.

9 CONCLUSION

We have presented supports R_{reg}, R_{class} characterizing finite neural networks which are solutions to problems relating to known implicit biases for regression and classification, in the case of one-dimensional data. We postulated that the differences between these two supports provided one explanation for the binning phenomenon. This claim was supported by numerical experiments, demonstrating that over-parameterized models learn features which notably coincide with the supports we proposed. Moreover, our synthetic problem clearly exhibited the binning phenomenon, resulting from the inability of the regression model to recover all of the sparse optimal support during training. Finally, we provided empirical evidence that the characteristics of our proposed supports hold in higher dimensions.

As far as we are aware, the implicit biases of arbitrarily deep neural networks is still an active research topic and is not currently known. If theorems similar to those of Chizat and Bach (2020) and Boursier et al. (2022) are obtained for deeper models, it may indeed be possible to extend the reasoning presented in our paper to deep neural nets by inspecting layer-wise features.

Our results raise many questions, both from a practical perspective and from a theoretical stand-point.

Practice: For some problems, the cross-entropy loss outperforms the square loss on regression tasks, despite it having no information about relationship between classes. Future works could investigate how to best incorporate the notion of adjacency between the bins, building on existing works such as Evgeniou et al. (2005). Other directions could include exploring different ways to discretize the data (e.g., jointly learning bins of differing sizes), or how best to choose the number of bins k for discretization.

Theory: A natural progression would be to prove a Theorem similar to that of 6.1, but for the implicit biases described in Boursier et al. (2022); Chizat and Bach (2020). Another option would be to extend the results of Theorem 6.1 to the case of multi-dimensional data. Further works on the implicit bias of deep models could help to explain the binning phenomenon reported in the literature mentioned in Section 1.

Acknowledgements

Thanks to Wilson Jallet and Nat McAleese for discussions relating to the binning phenomenon. We would also like to extend our thanks and gratitude to all of the reviewers for their constructive feedback. We also acknowledge support from the French government under the management of the Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), as well as from the European Research Council (grant SEQUOIA 724063).

References

I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, and R. Ribas. Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab. Robust optimization for deep regression. In *Proceedings of the International Conference on Computer Vision*, pages 2830–2838, 2015.

C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*, volume 100. Springer, 1984.

E. Boursier, L. Pillaud-Vivien, and N. Flammarion. Gradient flow dynamics of shallow ReLU networks for square loss and orthogonal inputs. In *Advances in Neural Information Processing Systems*, 2022.

S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in Neural Information Processing Systems*, 31, 2018.

L. Chizat and F. Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338, 2020.

L. C. Evans and R. F. Garzepy. *Measure Theory and Fine Properties of Functions*. Routledge, 1991.

T. Evgeniou, C. A. Micchelli, M. Pontil, and J. Shawe-Taylor. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(4), 2005.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth International Conference on Artificial Intelligence and Atatistics*, pages 249–256, 2010.

I. Goodfellow, O. Vinyals, and A. Saxe. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations*, 2015.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841, 2018a.

S. Gunasekar, J. D. Lee, D. Soudry, and N. Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in Neural Information Processing Systems*, 31, 2018b.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Z. Ji and M. Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

X. Liu, W. Liang, Y. Wang, S. Li, and M. Pei. 3D head pose estimation with convolutional neural network trained on synthetic images. In *International Conference on Image Processing*, pages 1289–1293, 2016.

K. Lyu and J. Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020.

J.-J. Moreau. Fonctionnelles convexes. *Séminaire Jean Leray*, pages 1–108, 1966.

M. S. Nacson, S. Gunasekar, J. Lee, N. Srebro, and D. Soudry. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In *International Conference on Machine Learning*, pages 4683–4692, 2019.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning*, 2010.

P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *ICLR (Workshop)*, 2015.

G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3433–3441, 2017.

R. Rothe, R. Timofte, and L. Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the International Conference on Computer Vision workshops*, pages 10–15, 2015.

W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1970.

I. Safran and O. Shamir. Spurious local minima are common in two-layer relu neural networks. In *International Conference on Machine Learning*, pages 4433–4441, 2018.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond*. MIT Press, 2002.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19 (1):2822–2878, 2018.

Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Advances in Neural information Processing Systems*, 27, 2014.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the Conference on Computer vision and Pattern Recognition*, pages 2818–2826, 2016.

M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.

L. Theis and M. Bethge. Generative image modeling using spatial lstms. *Advances in Neural Information Processing Systems*, 28, 2015.

A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

B. Uria, I. Murray, and H. Larochelle. A deep and tractable density estimator. In *International Conference on Machine Learning*, pages 467–475, 2014.

A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756, 2016.

A. Vaswani et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666, 2016.

Appendix

A PROOF OF IMPLICIT BIAS FOR MULTI-CLASS SOFTMAX REGRESSION

We consider the set-up of Chizat and Bach (2020, Theorem 3), and can follow the exact same proof, except that now the feature map is k-dimensional rather than 1-dimensional. Assumption $(A1)$ is unchanged, while Assumption $(A3)$ is considered component-wise.

For Assumption $(A2)$, we use the framework of Theorem 7 from Soudry et al. (2018) instead of the one in Theorem 3.

We can then extend the informal argument from Chizat and Bach (2020) that when using a predictor $\sum_{i=1}^m b_j(a_j^\top x)_+$, we converge to the minimum ℓ_2-norms $\sum_{i=1}^m \|b_j\|_2^2 + \|a_j\|_2^2$, which is minimized by scaling invariance as $2\sum_{i=1}^m \|b_j\|_2 \|a_j\|_2$, and thus, writing the predictor as:

$$\sum_{i=1}^m b_j(a_j^\top x)_+ = \sum_{i=1}^m b_j\|a_j\|_2((a_j/\|a_j\|_2)^\top x)_+ = \int_{S^{d-1}} (a^T x)_+ d\nu(a) \qquad \text{for} \quad \nu = \sum_{j=1}^m b_j\|a_j\|\delta_{\frac{a_j}{\|a_j\|}},$$

the penalty is exactly proportional to the total variation norm with ℓ_2-penalties.

B MINIMIZING TOTAL VARIATION OF A MEASURE WITH CONVEX LOWER SEMI-CONTINUOUS CONSTRAINTS

Lemma B.1. *Let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner-product defined on $\mathbb{R}^k$, and let Ω denote any norm on $\mathbb{R}^k$. Then for any measurable function $g : \mathbb{U} \to \mathbb{R}^k$:*

$$\sup_{\mu \in \mathcal{M}(\mathbb{U}, \mathbb{R}^k)} \int_{\mathbb{U}} \Big(\langle g(u), d\mu(u) \rangle - \Omega\left(d\mu(u)\right) \Big) \quad = \quad \begin{cases} 0 & \textit{if} \quad \Omega_*\left(g(u)\right) \leq 1 \quad \forall u \in \mathbb{U} \\ \infty & \textit{otherwise.} \end{cases} \tag{18}$$

Proof. Suppose that $\exists u_0 \in \mathbb{U}$ such that $\Omega_*\left(g(u_0)\right) > 1$. By definition of the dual norm, this implies:

$$\langle v_0, g(u_0) \rangle = \sup_{\Omega(v)=1} \langle v, g(u_0) \rangle > 1,$$

where we have used $v_0 \in \mathbb{R}^k$ to denote the vector that attains the supremum. For $t > 0$, consider the measure $\mu = tv_0\delta_{u_0} \in \mathcal{M}(\mathbb{U}, \mathbb{R}^k)$. Then:

$$\begin{aligned} \int_{\mathbb{U}} \Big(\langle g(u), d\mu(u) \rangle - \Omega\left(d\mu(u)\right) \Big) &= \langle g(u_0), tv_0 \rangle - \Omega\left(tv_0\right) \\ &= t \left(\underbrace{\langle g(u_0), v_0 \rangle - \Omega\left(v_0\right)}_{>1} \right). \end{aligned}$$

Hence taking the limit as $t \to \infty$ leads to an unbounded supremum. Conversely, suppose that $\Omega_*\left(g(u)\right) \leq 1 \quad \forall u \in \mathbb{U}$. Then for any $u \in \mathbb{U}$ and $v \in \mathbb{R}^k$ one has:

$$\begin{aligned} \langle g(u), v \rangle &= \Omega(v) \left\langle g(u), \frac{v}{\Omega(v)} \right\rangle \\ &\leq \Omega(v) \sup_{\Omega(w)=1} \langle g(u), w \rangle \\ &= \Omega(v)\, \Omega_*\left(g(u)\right) \leq \Omega(v). \end{aligned}$$

We conclude that $\int_{\mathbb{U}} \Big(\langle g(u), d\mu(u)\rangle - \Omega\left(d\mu(u)\right) \Big) \leq 0$, and hence:

$$\Omega\left(g(u)\right)_* \leq 1 \quad \forall a \in A \Longrightarrow \sup_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R}^k)} \int_{\mathbb{U}} \Big(\langle g(u), d\mu(u)\rangle - \Omega\left(d\mu(u)\right) \Big) \quad = \quad 0.$$

□

Lemma B.2. *The dual of problem* (15) *is:*

$$\begin{aligned} &\sup_{\alpha_1 \ldots, \alpha_n \in \mathbb{R}^k} && -\sum_{i=1}^{n} \sigma_{S_i}(\alpha_i) \\ &\textit{subject to} && \Omega_*\left(\sum_{i=1}^{n} \alpha_i \phi_u(x_i)\right) \leq 1 \quad \forall u \in \mathbb{U}. \end{aligned}$$

Proof. By Fenchel duality (Moreau, 1966) one has:

$$I^{**}_{S_i}\left(f_\mu(x_i)\right) = I_{S_i}\left(f_\mu(x_i)\right) = \sup_{\alpha_i \in \mathbb{R}^k} \left\{ \langle \alpha_i, f_\mu(x_i)\rangle - \sigma_{S_i}(\alpha_i) \right\}.$$

Plugging this into the Lagrangian we obtain:

$$\begin{aligned} L(\mu) &= \int_{\mathbb{U}} \Omega\left(d\mu(u)\right) \quad + \quad \sum_{i=1}^{n} \sup_{\alpha_i \in \mathbb{R}^k} \left\{ \langle \alpha_i, f_\mu(x_i)\rangle - \sigma_{S_i}(\alpha_i) \right\} \\ &= \int_{\mathbb{U}} \Omega\left(d\mu(u)\right) \quad + \quad \sup_{\alpha_1, \ldots, \alpha_n \in \mathbb{R}^k} \sum_{i=1}^{n} \left\{ \langle \alpha_i, f_\mu(x_i)\rangle - \sigma_{S_i}(\alpha_i) \right\} \\ &= \int_{\mathbb{U}} \Omega\left(d\mu(u)\right) \quad + \quad \sup_{\alpha_1, \ldots, \alpha_n \in \mathbb{R}^k} \left\{ \int_{\mathbb{U}} \Big\langle \sum_{i=1}^{n} \alpha_i \phi_u(x_i), d\mu(u) \Big\rangle \quad - \sum_{i=1}^{n} \sigma_{S_i}(\alpha_i) \right\} \\ &= \sup_{\alpha_1, \ldots, \alpha_n \in \mathbb{R}^k} \left\{ - \int_{\mathbb{U}} \Big(\Big\langle \underbrace{- \sum_{i=1}^{n} \alpha_i \phi_u(x_i)}_{g(u)}, d\mu(u) \Big\rangle \; - \; \Omega\left(d\mu(u)\right) \Big) \quad - \sum_{i=1}^{n} \sigma_{S_i}(\alpha_i) \right\}. \end{aligned}$$

The primal value $\inf\limits_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R}^k)} L(\mu)$ is hence:

$$\inf_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R}^k)} \; \sup_{\alpha_1, \ldots, \alpha_n \in \mathbb{R}^k} \left\{ - \int_{\mathbb{U}} \Big(\big\langle g(u), d\mu(u) \big\rangle \; - \; \Omega\left(d\mu(u)\right) \Big) \quad - \sum_{i=1}^{n} \sigma_{S_i}(\alpha_i) \right\}.$$

The dual problem is obtained by switching the order of the supremum and infinum:

$$\begin{aligned} &\sup_{\alpha_1, \ldots, \alpha_n \in \mathbb{R}^k} \; \inf_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R}^k)} \left\{ - \int_{\mathbb{U}} \Big(\big\langle g(u), d\mu(u) \big\rangle \; - \; \Omega\left(d\mu(u)\right) \Big) \quad - \sum_{i=1}^{n} \sigma_{S_i}(\alpha_i) \right\} \\ = &\sup_{\alpha_1, \ldots, \alpha_n \in \mathbb{R}^k} \left\{ \quad \inf_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R}^k)} \left\{ - \int_{\mathbb{U}} \Big(\big\langle g(u), d\mu(u) \big\rangle \; - \; \Omega\left(d\mu(u)\right) \Big) \right\} \quad - \sum_{i=1}^{n} \sigma_{S_i}(\alpha_i) \right\}. \end{aligned}$$

Applying Lemma B.1 we conclude the dual problem is:

$$\begin{aligned}\sup_{\alpha_1 \dots, \alpha_n \in \mathbb{R}^k} \quad & -\sum_{i=1}^n \sigma_{S_i}(\alpha_i) \\ \text{subject to} \quad & \Omega_* \left(\sum_{i=1}^n \alpha_i \phi_u(x_i) \right) \leq 1 \quad \forall u \in \mathbb{U}.\end{aligned}$$

□

C PROOF OF LEMMA 6.3

Define:

$$\begin{aligned}\mathbb{U}^+ &= \{1\} \times [-1, 1] \\ \mathbb{U}^- &= \{-1\} \times [-1, 1],\end{aligned}$$

and note that $\mathbb{U} = \mathbb{U}^+ \cup \mathbb{U}^-$. Similarly we define:

$$\begin{aligned}{U_X}^+ &= \{1\} \times \{x_1, \dots, x_n\} \\ {U_X}^- &= \{-1\} \times \{x_1, \dots, x_n\}.\end{aligned}$$

Firstly, let us show that:

$$\Omega_* \left(\sum_{i=1}^n \alpha_i \phi_u(x_i) \right) \leq 1 \quad \forall u \in \mathbb{U}^+. \tag{19}$$

This is equivalent to showing that:

$$\Omega_* \left(\sum_{i=1}^n \alpha_i (x_i - c) \right) \leq 1 \quad \forall c \in [-1, 1]. \tag{20}$$

By assumption, we know that:

$$\begin{aligned} & \Omega_* \left(\sum_{i=1}^n \alpha_i \phi_u(x_i) \right) \leq 1 \quad \forall u \in {U_X}^+ \\ \iff \quad & \Omega_* \left(\sum_{i=1}^n \alpha_i (x_i - c)_+ \right) \leq 1 \quad \forall c \in \{x_1, \dots, x_n\}.\end{aligned}$$

Let $g(c) = \sum_{i=1}^n \alpha_i (x_i - c)_+$ and remark that g is a piece-wise affine function with line segments meeting at $\{x_1, \dots, x_n\}$. As a consequence, $\forall c \in [-1, 1]$, $\exists \theta \in [0, 1]$ and $i \in [n - 1]$ such that:

$$g(c) = \theta \, g(x_i) + (1 - \theta) \, g(x_{i+1}).$$

By the convexity of the dual norm, we have:

$$\Omega_* \left(g(c) \right) \leq \theta \, \underbrace{\Omega_* \left(g(x_i) \right)}_{\leq 1} + (1 - \theta) \, \underbrace{\Omega_* \left(g(x_{i+1}) \right)}_{\leq 1} \leq 1.$$

We conclude that (20) (and hence (19)) hold. We conclude by repeating the same argument for $\mathbb{U}^-$, but replacing $g(c)$ with $h(c) = \sum_{i=1}^n \alpha_i (c - x_i)_+$.

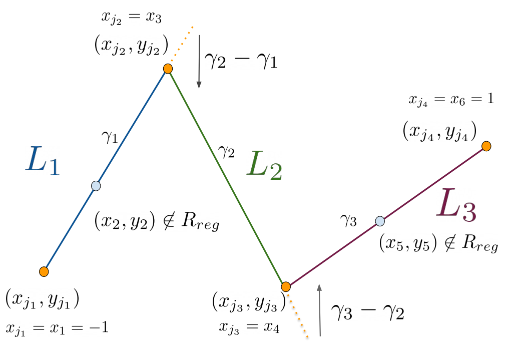


Figure 8: Graphical depiction of R_{reg} for six arbitrary data points $(x_1, y_1), \dots, (x_6, y_6)$. In this example $J_{reg} = \{1, 3, 4, 6\}$, resulting in a piece-wise interpolant with three line components L_1, L_2 and L_3.

D CONSTRUCTION OF FEASIBLE MEASURES

D.1 REGRESSION

Proof. As in the statement let $-1 = x_1 < \cdots < x_n = 1$ and $y_1, \cdots, y_n \in \mathbb{R}$ denote the regression data. We define:

$$J_{reg} = \{i \in [n] \,:\, x_i \in R_{reg}\}, \tag{21}$$

as the set of indices i which correspond to data points $x_i \in R_{reg}$. Let $m = |J_{reg}|$, and note that $\{1, n\} \subseteq J_{reg} \implies m \geq 2$. Without loss of generality we assume that the elements of J_{reg} are sorted in increasing order $1 = j_1 < \cdots < j_m = n$.

For $l \in [m-1]$, let L_l denote the line passing through (x_{j_l}, y_{j_l}) and $\left(x_{j_{l+1}}, y_{j_{l+1}}\right)$. By definition, the equation of each of the $m-1$ lines will be:

$$L_l(x) = \gamma_l(x - x_{j_l}) + y_{j_l}, \tag{22}$$

where:

$$\gamma_l = \frac{y_{j_{l+1}} - y_{j_l}}{x_{j_{l+1}} - x_{j_l}}. \tag{23}$$

A graphical depiction of this above notation can be found in Figure 8. Finally, we write P to denote the piece-wise linear interpolant of the data $\{(x_i, y_i)\}_{i=1}^n$, where:

$$x \in [x_{j_l}, x_{j_{l+1}}] \quad \implies \quad P(x) = L_l(x) \qquad \forall l \in [m-1]. \tag{24}$$

It is sufficient to construct a measure $\mu \in \mathcal{M}(U_X, \mathbb{R})$ such that $f_\mu = P$, since then $f_\mu(x_i) = P(x_i) = y_i \quad \forall i \in [n]$.

We begin by first constructing a measure μ_1 such that $f_{\mu_1}(x) = L_1(x)$, which we will later build upon to construct μ. From the definition of R_{reg}, the measure μ_1 will satisfy $f_{\mu_1}(x_i) = y_i$ for all $i \in [j_2]$.

Let $u_1, u_n \in U_X$ be defined as $u_1 = (1, x_1)$ and $u_n = (-1, x_n)$. We claim that μ_1 can be written in the following form:

$$\mu_1 = w_l \delta_{u_1} + w_r \delta_{u_n}. \tag{25}$$

To show this is true, we search for weights $w_r, w_l \in \mathbb{R}$ that satisfy:

$$\begin{aligned} L_1(x) = \gamma_1(x - x_1) + y_1 &= f_{\mu_1}(x) \\ &= \int_{\mathbb{U}} \phi_u(x)\, d\mu_1(u) \\ &= w_l \phi_{u_1}(x) + w_r \phi_{u_n}(x) \\ &= w_l (x - x_1)_+ + w_r (x_n - x)_+ \\ &= w_l (x - (-1)) + w_r (1 - x)_+ \\ &= w_l (x + 1)_+ + w_r (1 - x)_+. \end{aligned}$$

Substituting $x = 1$ gives:

$$\begin{aligned} w_l (2)_+ + w_r (0)_+ &= 2\gamma_1 + y_1 \\ \implies w_l &= \gamma_1 + \frac{y_1}{2}. \end{aligned}$$

Similarly, substituting $x = -1$ gives:

$$\begin{aligned} w_l (0)_+ + w_r (2)_+ &= 0\gamma_1 + y_1 \\ \implies w_r &= \frac{y_1}{2}. \end{aligned}$$

In the case that $m = 2$ we are done, since $f_{\mu_1} = P$. Otherwise, consider the measure:

$$\mu = \underbrace{\frac{y_1}{2}\delta_{u_n} + \left(\frac{y_1}{2} + \gamma_1\right)\delta_{u_1}}_{\mu_1} + \sum_{l=2}^{m-1} (\gamma_{l+1} - \gamma_l)\, \delta_{u_{j_l}} \quad \in \mathcal{M}(U_X, \mathbb{R}), \tag{26}$$

where $u_{j_l} = (1, x_{j_l})$. We claim that $f_\mu = P$, which is equivalent to saying that:

$$\forall l \in [m-1], \quad f_\mu(x) = L_l(x) \quad \forall x \in [x_{j_l}, x_{j_{l+1}}]. \tag{27}$$

We will show by induction that the above statement holds. The base case $l = 1$ is immediately verified, as $f_\mu(x) = L_1(x) \quad \forall x \in [x_{j_1}, x_{j_2}]$. To prove the inductive step, suppose for some $q \in [m-1]$ that the following holds:

$$\forall l \in [q-1], \quad f_\mu(x) = L_l(x) \quad \forall x \in [x_{j_l}, x_{j_{l+1}}]. \tag{28}$$

We need to show that:

$$\forall l \in [q], \quad f_\mu(x) = L_l(x) \quad \forall x \in [x_{j_l}, x_{j_{l+1}}]. \tag{29}$$

To do this, we note that:

$$\begin{aligned} x \in [x_{j_{q-1}}, x_{j_q}] \quad \implies \quad f_\mu(x) &= L_{q-1}(x) + (\gamma_q - \gamma_{q-1})\, \left(x - x_{j_{q-1}}\right)_+ \\ &= L_q(x). \end{aligned}$$

□

D.2 CLASSIFICATION

Proof. As in the statement let $-1 = x_1 < \cdots < x_n = 1$ and $y_1, \ldots, y_n \in [k]$ denote the classification data. We define the one-hot labels $Z \in \{0,1\}^{n \times k}$ as:

$$Z_{i,l} = \begin{cases} 1 & \text{if} \quad y_i = l \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

We define k regression data-sets $D_1, \ldots, D_k$, where:

$$D_l = \{(x_i, Z_{i,l})\}_{i=1}^n \quad \forall l \in [k]. \tag{31}$$

Applying the result obtained from Appendix D.1, $\exists \mu_1, \ldots, \mu_k \in \mathcal{M}(U_X, \mathbb{R})$ such that:

$$f_{\mu_l}(x_i) = Z_{i,l} \quad \forall i \in [n] \quad \forall l \in [k]. \tag{32}$$

By definition of Z, we conclude that $\mu = (\mu_1, \ldots, \mu_k) \in \mathcal{M}(U_X, \mathbb{R}^k)$ is feasible for problem (14).

□

E PROOF OF THEOREM 6.1

E.1 REGRESSION

Let $J_{reg} = \{i \in [n] \, : \, x_i \in R_{reg}\}$ and let $m = |J_{reg}|$.

Consider the following problem:

$$\begin{aligned} \inf_{\mu \in \mathcal{M}(\mathbb{U}, \mathbb{R})} \quad & \int_{\mathbb{U}} |d\mu(u)| \\ \text{subject to} \quad & f_\mu(x_j) = y_j \quad \forall j \in J_{reg}. \end{aligned} \tag{33}$$

Let P_1 and P_2 be the primal values for problems (13) and (33) respectively. As problem (33) has less constraints than problem (13), we can remark that $P_2 \leq P_1$. By Proposition 1, $\exists \mu^* \in \mathcal{M}(U_X, \mathbb{R})$ and ${\alpha_1}^*, \ldots, {\alpha_m}^* \in \mathbb{R}$ optimal for problem (33) satisfying supp$(\mu^*) \subseteq F_{reg}$.

Assume without loss of generality that J_{reg} is ordered, with $1 = j_1 < \cdots < j_m = n$. For $i \in J_{reg}$, let $\psi(i) \in [m]$ denote the position of i in the ordered list $j_1, \ldots, j_m$. We construct $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n \in \mathbb{R}$ as follows:

$$\tilde{\alpha}_i = \begin{cases} 0 & \text{if} \quad i \notin R_{reg} \\ \alpha^*_{\psi(i)} & \text{if} \quad i \in R_{reg}. \end{cases} \tag{34}$$

${\alpha^*}_1, \ldots, {\alpha^*}_m$ correspond to the m data points $(x_{j_1}, y_{j_1}), \ldots, (x_{j_m}, y_{j_m})$. Our constructed $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n$ is the extension of the above to all of the train data $(x_1, y_1), \ldots, (x_n, y_n)$, where $i \notin R_{reg} \implies \tilde{\alpha}_i = 0$.

By construction, μ^* and $\tilde{\alpha}_1, \ldots, \tilde{\alpha_n}$ attain strong duality for (13). However, it remains to verify that they are indeed prime and dual feasible for problem (13) respectively. For this, it is enough to verify that:

$$-\sigma_{S_i}(\tilde{\alpha}_i) = I_{S_i}\left(f_{\mu^*}(x_i)\right) \quad \forall i \in [n] \setminus J_{reg},$$

where the sets S_i are those corresponding to regression, described in Section 6.2. We remark that f_{μ^*} is a piece-wise affine function with line segments meeting at $\{(x_j, y_j)\}_{j \in J_{reg}}$. By definition, if $i \notin J_{reg}$ then $f_{\mu^*}(x_i) = y_i \implies I_{S_i}\left(f_{\mu^*}(x_i)\right) = 0$. Finally, $\sigma_{S_i}(\tilde{\alpha}_i) = \sigma_{S_i}(0) = 0 \quad \forall i \in [n] \setminus J_{reg}$.

E.2 CLASSIFICATION

Let $J_{class} = \{i \in [n] \,:\, x_i \in R_{class}\}$ and let $m = |J_{class}|$.

Consider the following problem:

$$\begin{aligned} \inf_{\mu \in \mathcal{M}(\mathbb{U},\mathbb{R}^k)} \quad & \int_{\mathbb{U}} \|d\mu(u)\| \\ \text{subject to} \quad & (e_{y_j} - e_l)^T f_\mu(x_j) \geq \mathbb{1}(y_j \neq l), \\ & \forall j \in J_{class}, \quad \forall l \in [k]. \end{aligned} \tag{35}$$

Let P_1 and P_2 be the primal values for problems (14) and (35) respectively. As problem (35) has less constraints than problem (14), we can remark that $P_2 \leq P_1$. By Proposition 1, $\exists \mu^* \in \mathcal{M}(U_X, \mathbb{R}^k)$ and $\alpha_1{}^*, \ldots, \alpha_m{}^* \in \mathbb{R}^k$ optimal for problem (35) satisfying supp$(\mu^*) \subseteq F_{class}$.

Assume without loss of generality that J_{class} is ordered, with $1 = j_1 < \ldots < j_m = n$. For $i \in J_{class}$, let $\psi(i) \in [m]$ denote the position of i in the ordered list $j_1, \ldots, j_m$. We construct $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n \in \mathbb{R}^k$ as follows:

$$\tilde{\alpha}_i = \begin{cases} \mathbf{0} & \text{if} \quad i \notin R_{class} \\ \alpha^*_{\psi(i)} & \text{if} \quad i \in R_{class}. \end{cases} \tag{36}$$

$\alpha^*_1, \ldots, \alpha^*_m$ correspond to the m data points $(x_{j_1}, y_{j_1}), \ldots, (x_{j_m}, y_{j_m})$. Our constructed $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n$ is the extension of the above to all of the train data $(x_1, y_1), \ldots, (x_n, y_n)$, where $i \notin R_{class} \implies \tilde{\alpha}_i = \mathbf{0}$.

By construction, μ^* and $\tilde{\alpha}_1, \ldots, \tilde{\alpha_n}$ attain strong duality for (13). However, it remains to verify that they are indeed prime and dual feasible for problem (13) respectively. For this, it is enough to verify that:

$$-\sigma_{S_i}(\tilde{\alpha}_i) = I_{S_i}\left(f_{\mu^*}(x_i)\right) \quad \forall i \in [n] \setminus J_{class},$$

where the sets S_i are those corresponding to classification, described in Section 6.2.

We begin by noting that $\forall l \in [k]$, $f_{\mu^*}(\cdot)_k$ is a piece-wise affine function with line segments meeting at points contained in some subset of $\{(x_j, y_j)\}_{j \in J_{class}}$. By definition:

$$\begin{aligned} i \notin J_{class} \quad & \implies \quad f_{\mu^*}(x_i)^T(e_{y_i} - e_l) \geq \mathbb{1}(y_i \neq 1) \qquad \forall l \in [k] \\ & \implies I_{S_i}\left(f_\mu(x_i)\right) = 0 \qquad \forall i \in [n] \setminus J_{class} \end{aligned}$$

Finally, $\sigma_{S_i}(\tilde{\alpha}_i) = \sigma_{S_i}(\mathbf{0}) = 0 \quad \forall i \in [n] \setminus J_{class}$.

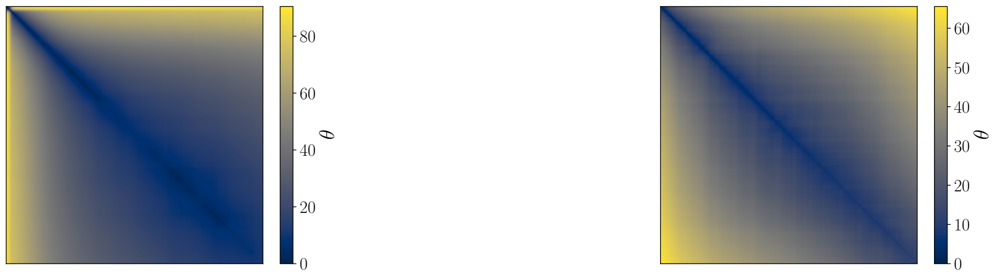
F TRAJECTORY OF GRADIENT DESCENT

Figures 9a / 9b depict the angles formed between the first 1000 gradients obtained during training for the worst performing regression / classification models. Despite both models being over-parameterized for the problem, it is clear that the optimization route for the models was not a straight line. Similar results were seen over all of the thirty random intializations.

G EXPERIMENTS WITH THREE-LAYER NEURAL NETWORKS

We provide results showing that the binning phenomenon observed in Section 7 only applies to two-layer neural networks. In other words, three-layer networks did not suffer the under-fitting we observed in Section 7.2.

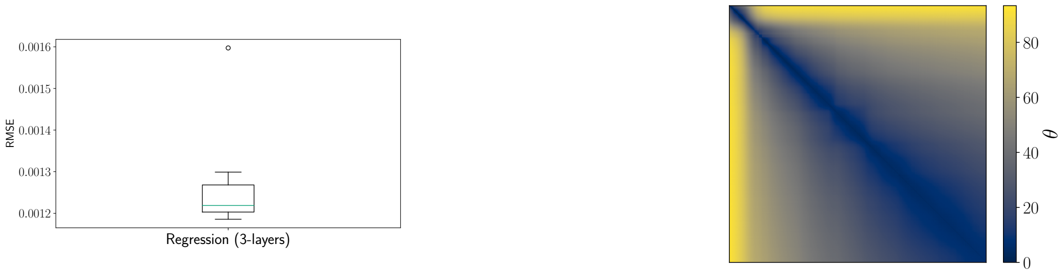
We trained a regression model with two hidden layers consisting of 1000 and 250 neurons (totalling $252,250$ parameters) using the square loss. The model was trained for ten random initializations of its weights, in the same manner as detailed in Section 7.1. The RMSE for the ten random initializations is depicted by Figure 10a.



(a)

(b)

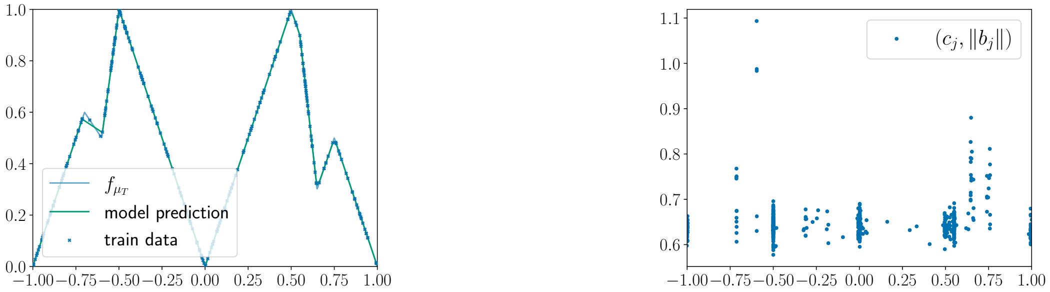
Figure 9: Angle θ between the first 1000 gradients obtained during training for the regression model (left) and the classification model (right).



(a)

(b)

Figure 10: RMSE (left) and angle θ between the first 1000 gradients obtained during training (right) for the three-layer neural network.



(a)

(b)

Figure 11: Worst performing model predictions (left) and support (right), over the ten random initializations.

Figure 10b shows that the gradient descent route during the first 1000 epochs of training was not linear. The predictions of the worst performing model over the 10 runs are depicted in Figure 11a. It can be seen that the network did not suffer the under-fitting observed for two-layer networks, detailed in Section 7.2. The support of the model is depicted in Figure 11b.

H TWO-DIMENSIONAL OPTIMAL SUPPORTS FOR SYNTHETIC DATA

We generated a regression problem from a random teacher model μ_T with three neurons, with weights being initialized as by Glorot and Bengio (2010). Our train data set consisted of 625 data points $\{(x_i, f_{\mu_T}(x_i))\}$, where the $x_i \in [-1,1]^2 \times \{1\}$ are spaced evenly over the 25×25 unit grid. To generate discretized labels we used $k = 25$ bins.

We trained two over-parameterized models:

1. **Regression Model:** 500 neurons in the hidden layer with scalar output. Trained using the square loss.
2. **Classification Model:** 500 neurons in the hidden layer with vector output of dimension $k = 25$. Trained using the cross-entropy loss.

As mentioned in Section 8, each feature a_j is now characterized by the line where it ramps, which we will refer to as the feature's "critical line". That is to say, the points $x \in \mathbb{R}^3$ satisfying:

$$a_{j,1}x_1 + a_{j,2}x_2 + a_{j,3} = 0,$$

where $x_3 = 1$. These can be thought of as the equivalent of c_j defined in Section 5, but for the two-dimensional case.

The critical lines characterizing the features of the regression and classification models after training are depicted in Figure 7a and 7b, respectively. Features with critical lines which do not cross the unit square only correspond to affine transformations of the resulting prediction, and for this reason can be ignored. Similarly, features killed by the output layer[6] since their contributions to the model's prediction are irrelevant.

We see that the regression model recovers a sparse support, whilst the classification model's features are more evenly distributed over unit square corresponding to (x_1, x_2). These observations are similar to R_{reg} and R_{class} in the one-dimensional case, suggesting that the difference in implicit bias between regression and classification support we identified in one-dimensional problems may hold in more general situations.

[6] That is to say the features a_j such that $\|a_j\|\|b_j\|$ is very small relative to other features.