UNDERSTANDING SQUARE LOSS IN TRAINING OVERPARAMETRIZED NEURAL NETWORK CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning has achieved many breakthroughs in modern classification tasks. Numerous architectures have been proposed for different data structures but when it comes to the loss function, the cross-entropy loss is the predominant choice. Recently, several alternative losses have seen revived interests for deep classifiers. In particular, empirical evidence seems to promote square loss but a theoretical justification is still lacking. In this work, we contribute to the theoretical understanding of square loss in classification by systematically investigating how it performs for overparametrized neural networks in the neural tangent kernel (NTK) regime. Interesting properties regarding the generalization error, robustness, and calibration error are revealed. We consider two cases, according to whether classes are separable or not. In the general non-separable case, fast convergence rate is established for both misclassification rate and calibration error. When classes are separable, the misclassification rate improves to be exponentially fast. Further, the resulting margin is proven to be lower bounded away from zero, providing theoretical guarantees for robustness. We expect our findings to hold beyond the NTK regime and translate to practical settings. To this end, we conduct extensive empirical studies on practical neural networks, demonstrating the effectiveness of square loss in both synthetic low-dimensional data and real image data. Comparing to cross-entropy, square loss has comparable generalization error but noticeable advantages in robustness and model calibration.

1 INTRODUCTION

The pursuit of better classifiers has fueled the progress of machine learning and deep learning research. The abundance of benchmark image datasets, e.g., MNIST, CIFAR, ImageNet, etc., provides test fields for all kinds of new classification models, especially those based on deep neural networks (DNN). With the introduction of CNN, ResNets, and transformers, DNN classifiers are constantly improving and catching up to the human-level performance. In contrast to the active innovations in model architecture, the training objective remains largely stagnant, with cross-entropy loss being the default choice. Despite its popularity, cross-entropy has been shown to be problematic in some applications. Among others, Yu et al. (2020) argued that features learned from cross-entropy lack interpretability and proposed a new loss aiming for maximum coding rate reduction. Pang et al. (2019) linked the use of cross-entropy to adversarial vulnerability and proposed a new classification loss based on latent space matching. Guo et al. (2017) discovered that the confidence of most DNN classifiers trained with cross-entropy is not well-calibrated.

Recently, several alternative losses have seen revived interests for deep classifiers. In particular, many existing works have presented empirical evidence promoting the use of square loss over cross-entropy. Hui & Belkin (2020) conducted large-scale experiments comparing the two and found that square loss tends to perform better in natural language processing related tasks while cross-entropy usually yields slightly better accuracy in image classification. Similar comparisons are also made in Demirkaya et al. (2020). Kornblith et al. (2020) compared a variety of loss functions and output layer regularization strategies on the accuracy and out-of-distribution robustness, and found that square loss has greater class separation and better out-of-distribution robustness.

In comparison to the empirical investigation, theoretical understanding of square loss in training deep learning classifiers is still lacking. Through our lens, square loss has its uniqueness among

classic classification losses, and we argue that it has great potentials for modern classification tasks. Below we list our motivations and reasons why.

Explicit feature modeling Deep learning's success can be largely attributed to its superior ability as feature extractors. For classification, the ideal features should be separated between classes and concentrated within classes. However, when optimizing cross-entropy loss, it's not clear what the learned features should look like (Yu et al., 2020). In comparison, square loss uses the label codings (one-hot, simplex etc.) as features, which can be modeled explicitly to control class separations.

Model Calibration An ideal classifier should not only give the correct class prediction, but also with the correct confidence. Calibration error measures the closeness of the predicted confidence to the underlying conditional probability η. Using square loss in classification can be essentially viewed as regression where it treats discrete labels as continuous code vectors. It can be shown that the optimal classifier under square loss is $2\eta - 1$, linear with the ground truth. This distinguishing property allows it to easily recover η. In comparison, the optimal classifiers under the hinge loss and cross-entropy are $\text{sign}(2\eta - 1)$ and $\log(\frac{\eta}{1-\eta})$, respectively. Therefore, hinge loss doesn't provide reliable information on the prediction confidence, and cross-entropy can be problematic when η is close to 0 or 1 (Zhang, 2004). Hence, in terms of model calibration, square loss is a natural choice.

Connections to popular approaches Mixup (Zhang et al., 2017) is a popular data augmentation technique where augmented data are constructed via convex combinations of inputs and their labels. Like in square loss, mixup treats labels as continuous and is shown to improve the generalization of DNN classifiers. In knowledge distillation (Hinton et al., 2015), where a student classifier is trying to learn from a trained teacher, Menon et al. (2021) proved that the "optimal" teacher with the ground truth conditional probabilities provides the lowest variance in student learning. Since classifiers trained using square loss is a natural consistent estimator of η, one can argue that it is a better teacher. In contrastive learning (Arora et al., 2019b), the features are learned by pulling together positive pairs and pushing away negative ones. With the label information available, Khosla et al. (2020) proposed supervised contrastive learning. By closely examining the training objective, we find that the optimal features from supervised contrastive learning can be obtained by square loss, with simplex label coding (more details in Section 4).

Despite its lack of popularity in practice, square loss has many advantages that can be easily overlooked. In this work, we systematically investigate from a statistical estimation perspective, the properties of deep learning classifiers trained using square loss. The neural networks in our analysis are required to be sufficiently overparametrized in the neural tangent kernel (NTK) regime. Even though this restricts the implication of our results, it is a necessary first step towards a deeper understanding. In summary, our main contributions are:

- Generalization error bound: We consider two cases, according to whether classes are separable or not. In the general non-separable case, we adopt the classical binary classification setting with smooth conditional probability. Fast rate of convergence is established for overparametrized neural network classifiers with Tsybakov's noise condition. If two classes are separable with positive margins, we show that overparametrized neural network classifiers can provably reach zero misclassification error with probability *exponentially* tending to one. To the best of our knowledge, this is the *first* such result for separable but not linear separable classes. Furthermore, we bridge these two cases and offer a *unified* view by considering auxiliary random noise injection.
- Robustness (margin property): When two classes are separable, the decision boundary is not unique and large-margin classifiers are preferred. In the separable case, we further show that the decision boundary of overparametrized neural network classifiers trained by square loss cannot be too close to the data support and the resulting margin is lower bounded away from zero, providing theoretical guarantees for robustness.
- Calibration error: We show that classifiers trained using square loss are inherently well-calibrated, i.e., the trained classifier provides consistent estimation of the ground-truth conditional probability in L_∞ norm. Such property doesn't hold for cross-entropy.
- Multi-class classification: The standard square loss usually struggles when the number of classes is large. To address this issue, we adopt the simplex label coding (Mroueh et al., 2012) and reveal nontrivial connections to supervised contrastive learning (Khosla et al., 2020) that they share the same optimal solution.

- Empirical evaluation: We corroborate our theoretical findings with empirical experiments in both synthetic low-dimensional data and real image data. Comparing to cross-entropy, square loss has comparable generalization error but noticeable advantages in robustness and model calibration.

This work contributes towards the theoretical understanding of deep classifiers, from an estimation point of view, which has been a classic topic in statistics literature. Among others, Mammen & Tsybakov (1999) established the optimal convergence rate for 0-1 loss excess risk when the decision boundary is smooth. Zhang (2004); Bartlett et al. (2006) extended the analysis to various surrogate losses. Audibert & Tsybakov (2007); Kohler & Krzyzak (2007) studied the convergence rates for plug-in classifiers from local averaging estimators. Steinwart et al. (2007) investigated the convergence rate for support vector machine using Gaussian kernels. We build on and extend classic results to neural networks in the NTK regime. Comparing to existing works on deep learning classification, e.g., Kim et al. (2018) derived fast convergence rates of ReLU DNN classifiers that minimize the empirical hinge loss, our results incorporate the training algorithm and apply to trained classifiers.

We require the neural network to be overparametrized, which has been extensively studied recently, under the umbrella term NTK. Most such results are in the regression setting with a handful of exceptions. Ji & Telgarsky (2019) showed that only polylogarithmic width is sufficient for gradient descent to overfit the training data using logistic loss. Hu et al. (2020) proved generalization error bound for regularized NTK in classification. Cao & Gu (2019) provided optimization and generalization guarantees for overparametrized network trained with cross-entropy. In comparison, our results are sharper in the sense that we take the ground truth data assumptions into consideration. This allows a faster convergence rate, especially when the classes are separable, where the exponential convergence rate is attainable. The NTK framework greatly reduces the technical difficulty for our theoretical analysis. However, our results are mainly due to properties of the square loss itself and we expect them to hold for a wide range of classifiers.

There are other works investigating the use of square loss for training (deep) classifiers. Han et al. (2021) uncovered that the "neural collapse" phenomenon also occurs under square loss where the last-layer features eventually collapse to their simplex-style class-means. Muthukumar et al. (2020) compared classification and regression tasks in the overparameterized linear model with Gaussian features, illustrating different roles and properties of loss functions used at the training and testing phases. Poggio & Liao (2019) made interesting observations on effects of popular regularization techniques such as batch normalization and weight decay on the gradient flow dynamics under square loss. These findings support our theoretical results' implication, which further strengthens our beliefs that the essence comes from the square loss and our analysis can go beyond NTK regime.

The rest of this paper is arranged as follows. Section 2 presents some preliminaries. Main theoretical results are in Section 3. The simplex label coding is discussed in Section 4 followed by numerical studies in Section 5 and conclusions in Section 6. Technical proofs and details of the numerical studies can be found in the Appendix.

2 PRELIMINARIES

Notation For a function $f : \Omega \to \mathbb{R}$, let $\|f\|_\infty = \sup_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})|$ and $\|f\|_p = (\int_\Omega |f(\boldsymbol{x})|^p d\boldsymbol{x})^{1/p}$. For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_p$ denotes its p-norm, for $1 \le p \le \infty$. L_p and l_p are used to distinguish function norms and vector norms. For two positive sequences $\{a_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$, we write $a_n \lesssim b_n$ if there exists a constant $C > 0$ such that $a_n \le C b_n$ for all sufficiently large n. We write $a_n \asymp b_n$ if $a_n \lesssim b_n$ and $b_n \lesssim a_n$. Let $[N] = \{1, \dots, N\}$ for $N \in \mathbb{N}$, $\mathbb{I}$ be the indicator function, and $\boldsymbol{I}_d$ be the $d \times d$ identity matrix. $N(\mu, \boldsymbol{\Sigma})$ represents Gaussian distribution with mean μ and covariance $\boldsymbol{\Sigma}$.

Classification problem settings Let P be an underlying probability measure on $\Omega \times \boldsymbol{Y}$, where $\Omega \subset \mathbb{R}^d$ is compact and $\boldsymbol{Y} = \{1, -1\}$. Let (X, Y) be a random variable with respect to P. Suppose we have observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset (\Omega \times Y)^n$ i.i.d. sampled according to P. The classification task is to predict the unobserved label y given a new input $\boldsymbol{x} \in \Omega$. Let η defined on Ω denote the conditional probability, i.e., $\eta(\boldsymbol{x}) = \mathbb{P}(y = 1|\boldsymbol{x})$. The key quantity of interest is the misclassification error, i.e., 0-1 loss, In the population level, the 0-1 loss can be written as

$$L(f) = \mathbb{E}_X \mathbb{I}\{\text{sign}(f(X)) \neq Y\} = \mathbb{E}_{f(X)\ge 0}(1 - \eta(X)) + \mathbb{E}_{f(X)<0}\eta(X), \tag{2.1}$$

where the expectation is taken with respect to the probability measure P. Clearly, an optimal classifier with the minimal 0-1 loss is $2\eta - 1$.

According to Ω and the support of η, there are two scenarios of interest. If η only takes value from $\{0,1\}$, and the sets $\Omega_1 = \{\boldsymbol{x}|\eta(\boldsymbol{x}) = 1\}$ and $\Omega_2 = \{\boldsymbol{x}|\eta(\boldsymbol{x}) = 0\}$ are separated, then we call this case the *separable case*[1]. If the probability measure of $\{\boldsymbol{x}|\eta(\boldsymbol{x}) \in (0,1)\}$ is non-zero, then we call this case the *non-separable case*. In the separable case, we assume that there exists a positive margin, i.e., $\text{dist}(\Omega_1, \Omega_2) \geq 2\gamma > 0$, where γ is a constant, and $\text{dist}(\Omega_1, \Omega_2) = \inf_{\boldsymbol{x}\in\Omega_1, \boldsymbol{x}'\in\Omega_2} \|\boldsymbol{x} - \boldsymbol{x}'\|_2$. In the non-separable case, to quantify the difficulty of classification, we adopt the well-established Tsybakov's noise condition (Audibert & Tsybakov, 2007), which assumes the measure of the "difficult region" where $\eta(\boldsymbol{x}) \approx 1/2$ is relatively small.

Definition 2.1 (Tsybakov's noise condition)**.** Let $0 < \kappa \leq \infty$ and P_X be the marginal distribution of P on X. We say that P has Tsybakov noise exponent κ if there exists a constant $C > 0$ such that for all sufficiently small $t > 0$, $P_X(|2\eta(X) - 1| < t) \leq C \cdot t^{\kappa}$.

A large value of κ implies the difficult region to be small. It is expected that a larger κ leads to a faster convergence rate of a neural network classifier. This intuition is verified for the overparametrized neural network classifier trained by square loss and ℓ_2 regularization. See Section 3 for more details.

Neural network setup We mainly focus on the one-hidden-layer ReLU neural network family $\mathcal{F}$ with m nodes in the hidden layer, denoted by

$$f_{\boldsymbol{W},\boldsymbol{A}}(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(\boldsymbol{W}_r^\top \boldsymbol{x}),$$

where $\boldsymbol{x} \in \Omega$, $\boldsymbol{W} = (\boldsymbol{W}_1, \cdots, \boldsymbol{W}_m) \in \mathbb{R}^{d\times m}$ is the weight matrix in the hidden layer, $\boldsymbol{A} = (a_1, \cdots, a_m)^\top \in \mathbb{R}^m$ is the weight vector in the output layer, $\sigma(z) = \max\{0, z\}$ is the rectified linear unit (ReLU). The initial values of the weights are independently generated from

$$\boldsymbol{W}_r(0) \sim N(\boldsymbol{0}, \xi^2 \boldsymbol{I}_m), \quad a_r \sim \text{unif}\{-1, 1\}, \quad \forall r \in [m].$$

Based on the observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the goal of training a neural network is to find a solution to

$$\min_{\boldsymbol{W},\boldsymbol{A}} \sum_{i=1}^{n} l(f_{\boldsymbol{W},\boldsymbol{A}}(\boldsymbol{x}_i), y_i) + \mu \mathcal{R}(\boldsymbol{W}, \boldsymbol{A}), \tag{2.2}$$

where l is the loss function, $\mathcal{R}$ is the regularization, and $\mu \geq 0$ is the regularization parameter. In this work, we consider square loss associated with ℓ_2 regularization, i.e., $l(f_{\boldsymbol{W},\boldsymbol{A}}(\boldsymbol{x}_i), y_i) = (f_{\boldsymbol{W},\boldsymbol{A}}(\boldsymbol{x}_i) - y_i)^2$ and $\mathcal{R}(\boldsymbol{W}, \boldsymbol{A}) = \|\boldsymbol{W}\|_2^2$. Since $a \cdot \sigma(z) = \text{sign}(a) \cdot \sigma(|a|z)$, we can always reparametrize the network to have all a_i's to be either 1 or -1. Therefore, we consider fixing $\boldsymbol{A}$ and only training the weights $\boldsymbol{W}$.

A popular way to train the neural network is via gradient based methods. It has been shown that the training process of DNNs can be characterized by the neural tangent kernel (NTK) (Jacot et al., 2018). As is usually assumed in the NTK literature (Arora et al., 2019a; Hu et al., 2020; Bietti & Mairal, 2019; Hu et al., 2021), we consider data on the unit sphere $\mathbb{S}^{d-1}$, i.e., $\|\boldsymbol{x}_i\|_2 = 1, \forall i \in [n]$, and the neural network is highly overparametrized ($m \gg n$) and trained by gradient descent (GD). For details about NTK and GD in one-hidden-layer ReLU neural networks, we refer to Appendix A.

3 Theoretical Results

In this section, we present our main theoretical results. Throughout the analysis, we assume that the overparametrized neural network $f_{\boldsymbol{W},\boldsymbol{A}}$ and the training process via GD satisfy Assumption C.1 (see Appendix C), which essentially requires the neural network to be sufficiently overparametrized (with a finite width), and imposes conditions on the learning rate and iteration number. Our theoretical results consist of three parts: generalization error, robustness, and calibration error.

3.1 Generalization error bound

In classification, the generalization error is typically referred to as the misclassification error, which can be quantified by $L(f)$ defined in Equation 2.1. In the non-separable case, the excess risk,

[1] In the separable case we consider, the classes are not limited to linearly separable but can be arbitrarily complicated.

defined by $L(f) - L^*$, is used to evaluate the quality of a classifier f, where $L^* = L(2\eta - 1)$, which minimizes the 0-1 loss. The following theorem states that the overparametrized neural network with GD and ℓ_2 regularization can achieve a small excess risk in the non-separable case.

Theorem 3.1 (Excess risk in the non-separable case)**.** Suppose Assumptions C.1, C.2, and C.4 hold. Assume the conditional probability $\eta(\boldsymbol{x})$ satisfies Tsybakov's noise condition with component κ. Let $\mu \asymp n^{\frac{d-1}{2d-1}}$. Then

$$L(\widehat{f}) = L^* + O_{\mathbb{P}}(n^{-\frac{d(\kappa+1)}{(2d-1)(\kappa+2)}}). \tag{3.1}$$

From Theorem 3.1, we can see that as κ becomes larger, the convergence rate becomes faster, which is intuitively true. It is known that the optimal convergence rate under Assumptions C.2 and C.4 is $O_{\mathbb{P}}(n^{-\frac{d(\kappa+1)}{d\kappa+4d-2}})$ (Audibert & Tsybakov, 2007). The differences between Equation 3.1 and the optimal convergence rate is that there is an extra $(d-1)\kappa$ in the denominator of the convergence rate in Equation 3.1. If the conditional probability η has a bounded Lipschitz constant, then Kohler & Krzyzak (2007) showed that the convergence rate based on the plug-in kernel estimate is $O_{\mathbb{P}}(n^{-(\kappa+1)/(\kappa+3+d)})$, which is slower than the rate in Equation 3.1 if d is large.

Now we turn to the separable case. Since η only takes value from $\{0, 1\}$ in the separable case, η is bounded away from 1/2. Therefore, one can trivially take $\kappa \to \infty$ in Equation 3.1 and obtain the convergence rate $O_{\mathbb{P}}(n^{-d/(2d-1)})$. However, this rate can be significantly improved in the separable case, as stated in the following theorem.

Theorem 3.2 (Generalization error in the separable case)**.** Suppose Assumptions C.1, C.3, and C.5 hold. Let $\mu = o(1)$. There exist constants C_1, C_2 such that the misclassification rate is 0% with probability at least $1 - \delta - C_1 \exp(-C_2 n)$, and δ can be arbitrarily small by enlarging the neural network's width.

Note that in Theorem 3.2, the regularization parameter can take any rate that converges to zero. In particular, μ can be zero, and the corresponding classifier overfits the training data. Theorem 3.2 states that the convergence rate in the separable case is exponential, if a sufficiently wide neural network is applied. This is because the observed labels are not corrupted by noise, i.e., $\mathbb{P}(y = 1|\boldsymbol{x})$ is either one or zero. Therefore, it is easier to classify separable data, which is intuitively true.

3.2 Robustness and Calibration Error

If two classes are separable with positive margin, the decision boundary is not unique. Practitioners often prefer the decision boundary with large margins, which are robust against possible perturbation on input points (Elsayed et al., 2018; Ding et al., 2018). The following theorem states that the square loss trained margin can be lower bounded by a positive constant.

Theorem 3.3 (Robustness in the separable case)**.** Suppose the assumptions of Theorem 3.2 are satisfied. Let $\mu = o(1)$. Then there exist constants C, C_1, C_2 such that for all n,

$$\min_{\boldsymbol{x} \in \mathcal{D}_T, \boldsymbol{x}' \in \Omega_1 \cup \Omega_2} \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \geq C,$$

and the misclassification rate is 0% with probability at least $1 - \delta - C_1 \exp(-C_2 n)$, where $\mathcal{D}_T$ is the decision boundary, and δ is as in Theorem 3.2.

The optimal decision boundary is unique if two classes are non-separable. In this case, practitioners may not only want a classifier with a small excess risk, but also want to recover the underlying conditional probability η. Therefore, square loss is naturally preferred since it treats the classification problem as a regression problem. The following theorem states that, one can recover the conditional probability η by using an overparametrized neural network with ℓ_2 regularization and GD training.

Theorem 3.4 (Calibration error)**.** Suppose the conditions in Theorem 3.1, Assumption C.3 and C.4 are fulfilled. Let $\mu \asymp n^{\frac{d-1}{2d-1}}$. Then

$$\|(f_{\boldsymbol{W},\boldsymbol{A}} + 1)/2 - \eta\|_{L_\infty} = O_{\mathbb{P}}(n^{-\frac{1}{4d-2}}). \tag{3.2}$$

Theorem 3.4 states that the underlying conditional probability in the non-separable case can be recovered by $(f_{\boldsymbol{W},\boldsymbol{A}} + 1)/2$. The form $(f_{\boldsymbol{W},\boldsymbol{A}} + 1)/2$ is to account for the $\{-1, 1\}$ label coding.

Under {0,1} coding, the estimator would be $f_{\boldsymbol{W},\boldsymbol{A}}$ itself. The L_∞ consistency doesn't hold for cross-entropy trained neural networks, due to the form of the optimal solution $\log(\frac{\eta}{1-\eta})$. With limited capacity, the network's confidence prediction is bounded away from 0 and 1 (Zhang, 2004).

Remark 1 (Technical challenge)**.** Despite the similar forms of regression and classification using square loss, most of the regression analysis techniques cannot be directly applied to the classification problem, even if the supports of two classes are non-separable. Moreover, it is clear that classification problems in the separable case are completely different with regression problems.

Remark 2 (Extension on NTK)**.** Although our analysis only concerns overparametrized one-hidden-layer ReLU neural networks, it can potentially apply to other types of neural networks in the NTK regime. Recently, it has been shown that overparametrized multi-layer networks correspond to the Laplace kernel (Geifman et al., 2020; Chen & Xu, 2020). As long as the trained neural networks can approximate the classifier induced by the NTK, our results can be naturally extended.

3.3 Transition from separable to non-separable

The general non-separable case and the special separable case can be connected via Gaussian noise injection. In practice, data augmentation is an effective way to improve robustness and the simplest way is Gaussian noise injection (He et al., 2019). In this section, we only consider it as an auxiliary tool for theoretical analysis purpose and not for actual robust training. Injecting Gaussian noise amounts to convoluting a Gaussian distribution $N(0, \upsilon^2 \boldsymbol{I}_d)$ to the marginal distribution P_X, which enlarges both Ω_1 and Ω_2 to $\mathbb{R}^d$ and a unique decision boundary $\mathcal{D}_\upsilon$ can be induced. Correspondingly, the "noisy" conditional probability, denoted as $\widetilde{\eta}_\upsilon$, is also smoothed to be continuous on $\mathbb{R}^d$. As $\upsilon \to 0$, $\|\widetilde{\eta}_\upsilon - \eta\|_\infty \to 0$ on Ω_1 and Ω_2 and the limiting $\widetilde{\eta}_0$ is a piecewise constant function with discontinuity at the induced decision boundary.

Lemma 3.5 (Tsybakov's noise condition under Gaussian noises)**.** Let the margin be $2\gamma > 0$, the noise be $N(0, \upsilon^2 \boldsymbol{I}_d)$. Then there exist some constants $T, C > 0$ such that for any $0 < t < T$,

$$P_X(|2\widetilde{\eta}_\upsilon(X) - 1| < t) \leq \frac{C\upsilon^2}{\gamma} \exp\left(-\frac{\gamma^2}{2\upsilon^2}\right) \cdot t.$$

Theorem 3.6 (Exponential convergence rate)**.** Suppose the classes are separable with margin $2\gamma > 0$. No matter how complicated $\Omega_1 \cup \Omega_2$ are, the excess risk of the over parameterized neural network classifier satisfying Assumptions C.1 and C.4 has the rate $O_{\mathbb{P}}(e^{-n\gamma/7})$.

The proof of Theorem 3.6 involves taking the auxiliary noise to zero, e.g., $\upsilon = \upsilon_n \asymp 1/\sqrt{n}$. The exponential convergence rate is a direct outcome of Lemma 3.5 and Theorem 3.1.

Remark 3. Theorems 3.4 and 3.6 share the same gist that the over parameterized neural network classifiers can have exponential convergence rate when data are separable with positive margin, while the result of Theorem 3.6 is weaker than that of Theorem 3.4, but with milder conditions. Nevertheless, Theorem 3.6 bridges the non-separable case and separable case.

4 Simplex label Coding

In binary classification, the labels are usually encoded as -1 and 1. When there are $K > 2$ classes, the default label coding is one-hot, i.e., the code for class k is $\boldsymbol{e}_k := (0, \ldots, 0, 1, 0, \ldots, 0)^\top \in \mathbb{R}^K$ with only the k-th element being one. One can generalize the theoretical development without too much pain in Section 3 by employing the following objective function

$$\min_{\boldsymbol{W},\boldsymbol{A}} \sum_{j=1}^{K} \sum_{i=1}^{n} (f_{j,\boldsymbol{W},\boldsymbol{A}}(\boldsymbol{x}_i) - y_{i,j})^2 + \mu \|\boldsymbol{W}\|_2^2,$$

where $f_{\boldsymbol{W},\boldsymbol{A}} : \Omega \mapsto \mathbb{R}^K$, and $\boldsymbol{y}_i = (y_{i,1}, ..., y_{i,K})^\top$ is the label of i-th observation.

However, it is empirically observed that vanilla square loss struggles when the number of classes are large. For a single training data point $\boldsymbol{x}$ and label k, Hui & Belkin (2020) proposed to modify the training objective from the typical $(f_k(\boldsymbol{x}) - 1)^2 + \sum_{i \neq k} f_i(\boldsymbol{x})^2$ to $J \cdot (f_k(\boldsymbol{x}) - M)^2 + \sum_{i \neq k} f_i(\boldsymbol{x})^2$, where J, M are hyperparameters to make f_k more prominent in the loss. Similar modification is also proposed in Demirkaya et al. (2020).

Existing modifications to square loss in multi-class classification are mainly for optimization while in this section, we provide another perspective from feature representation and promote the simplex coding scheme (Mroueh et al., 2012). The one-hot coding forces the features for each class to be orthogonal in the unit sphere, which is problematic for large K since the representations are all cramped into the corner (positive direction in each dimension) taking only 2^{-K} of the entire sphere. In comparison, maximally separated K points on the unit sphere is the better choice, which coincides with the vertices of a $(K-1)$-simplex in $K-1$ dimensions. When $K=2$, this reduces to the typical $-1, 1$ coding. The following proposition states a relationship between the simplex coding scheme and the conditional probability.

Proposition 4.1 (Conditional probability)**.** Let $f^* : \Omega \to \mathbb{R}^K$ minimize the mean square error $\mathbb{E}_X(f^*(X) - \boldsymbol{v}_y)^2$, where $\boldsymbol{v}_y$ is the simplex coding vector of label y. Then we have

$$\eta_k(\boldsymbol{x}) := \mathbb{P}(y = k|\boldsymbol{x}) = \left((K-1)f^*(\boldsymbol{x})^\top \boldsymbol{v}_k + 1\right)/K. \tag{4.1}$$

An interesting observation is that for K-class classification, feature spaces with more than $K-1$ dimensions are redundant as K class centers can only determine a $(K-1)$-dim subspace. Take $K=3$ as an example, where the maximally separated points in a 100-dim sphere are the same as those in a unit circle with pairwise angle $120°$ (vertices of an equilateral triangle).

In the feature space, the maximum separation property is shared by contrastive learning where features of positive pairs are pulled together while features of negative pairs are pushed away on high dimensional sphere. The contrastive style learning can also be adapted to supervised learning (Khosla et al., 2020) where positive and negative pairs are determined by the class labels. The following proposition establishes a connection between supervised contrastive learning and simplex coding, implying the resemblance of square loss with simplex coding and supervised contrastive learning.

Proposition 4.2. In supervised contrastive learning (Khosla et al., 2020), features corresponding to the simplex coding minimize the supervised contrastive loss.

5 NUMERICAL STUDIES

Although our theoretical results are for overparametrized neural network in the NTK regime, we expect our conclusions to generalize to practical network architectures. The focus of this section is not on improving the state-of-the-art performance for deep classifiers, but to illustrate the difference between cross-entropy and square loss. We provide experiment results on both synthetic and real data, to support our theoretical findings and illustrate the practical benefits of square loss in training overparametrized DNN classifiers. Compared with cross-entropy, the square loss has comparable generalization performance, but with stronger robustness and smaller calibration error.

5.1 SYNTHETIC DATA

We consider the square loss based and cross-entropy based overparametrized neural networks (ONN) with ℓ_2 regularization, denoted as SL-ONN + ℓ_2 and CE-ONN + ℓ_2, respectively. The chosen ONNs are two-hidden-layer ReLU neural networks with 500 neurons for each layer, and the parameter μ is selected via a validation set. More implementation details are in Appendix F.1.

Separable case We consider two separated classes with spiral curve like supports. We also present the performance of the cross-entropy based ONN without ℓ_2 regularization (CE-ONN). Figure 1 shows one instance of the test misclassification rate and decision boundaries attained by SL-ONN + ℓ_2 (Left), CE-ONN + ℓ_2 (Center), and CE-ONN (Right). From this example and other examples in Appendix F.1, it can be seen that SL-ONN + ℓ_2 has a smaller test misclassification rate and a much smoother decision boundary. In particular, in the yellow region, where the training data are sparse, SL-ONN + ℓ_2 fits the correct data distribution best.

Non-separable case We consider the conditional probability $\eta(\boldsymbol{x}) = \sin(\sqrt{2}\pi||\boldsymbol{x}||_2), \boldsymbol{x} \in [-1,1]^2$, and the calibration performance of SL-ONN + ℓ_2 and CE-ONN + ℓ_2, where the classifiers are denoted by $\widehat{f}_{l2}$ and $\widehat{f}_{ce}$, respectively. The results are presented in Figure 2. The error bar plot[2] of the test calibration error shows that $\widehat{f}_{l2}$ has the smaller mean and standard deviation than $\widehat{f}_{ce}$.

[2]In an error bar plot, the center of each plot is the mean, and the upper and lower red dashes denote (mean+one standard deviation) and (mean − one standard deviation), respectively.

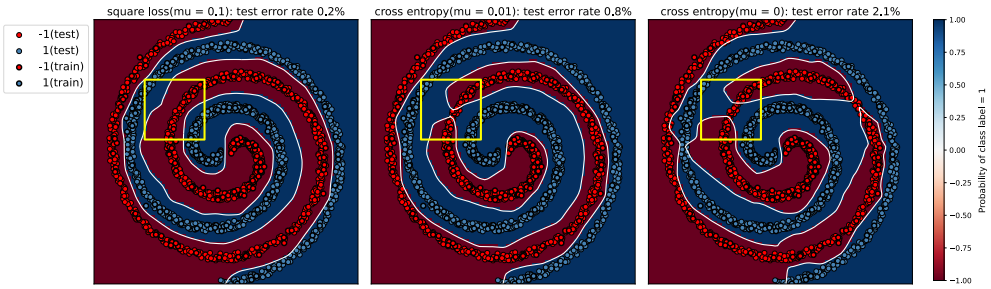

Figure 1: Test misclassification rates and decision boundaries predicted by: SL-ONN + ℓ_2 (Left); CE-ONN + ℓ_2 (Center); CE-ONN (Right) for the separable case. The test error rate hereby in the title of each subfigure refers test misclassification rate.

It suggests that square loss generally outperforms cross entropy in calibration. The histogram and kernel density estimation of the test calibration errors for one case show that the pointwise calibration errors on the test points of $\widehat{f}_{l2}$ are more concentrated around zero than those of $\widehat{f}_{ce}$. Moreover, despite a comparable misclassification rate with $\widehat{f}_{ce}$, $\widehat{f}_{l2}$ has a smaller calibration error. Figure 2 demonstrates that SL-ONN + ℓ_2 recovers η much better than CE-ONN + ℓ_2.

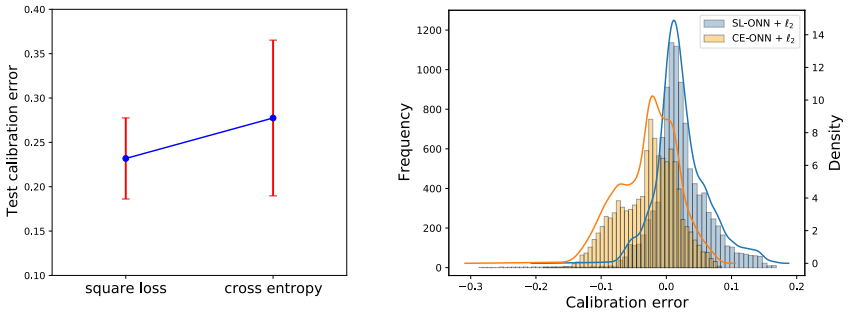

Figure 2: (Left) The error bar plot of test calibration errors for the 40 replicated runs. (Right) The histogram and kernel density estimation of test calibration errors in one instance from 40 replications. In this instance, $||(\widehat{f}_{l2}+1)/2-\eta||_{L_\infty}$ = 0.188, $||(\widehat{f}_{ce}+1)/2-\eta||_{L_\infty}$ = 0.287, and the test misclassification rate for $\widehat{f}_{l2}$ and $\widehat{f}_{ce}$ are 0.167 and 0.164, respectively.

5.2 Real Data

To make a fair comparison, we adopt popular architectures, ResNet (He et al., 2016) and Wide ResNet (Zagoruyko & Komodakis, 2016) and evaluate them on the CIFAR image classification datasets, with only the training loss function changed, from cross-entropy (CE) to square loss with simplex coding (SL). Further, we don't employ any large scale hyper-parameter tuning and all the parameters are kept as default except for the learning rate (lr) and batch size (bs), where we are choosing from the better of (lr=0.01, bs=32) and (lr=0.1, bs=128). Each experiment setting is replicated 5 times and we report the average performance followed by its standard deviation in the parenthesis. (lr=0.01, bs=32) works better for the most cases except for square loss trained WRN-16-10 on CIFAR-100. More experiment details and additional results can be found in Appendix F.2.

Generalization In both CIFAR-10 and CIFAR-100, the performance of cross-entropy and square loss with simplex coding are quite comparable, as observed in Hui & Belkin (2020). Cross-entropy tends to perform slightly better for ResNet, especially on CIFAR-100 with an advantage of less than 1%. There is a more significant gap with Wide ResNet where square loss outperforms cross-entropy by more than 1% on both CIFAR-10 and CIFAR-100. The details can be found in Table 1.

Adversarial robustness Normally trained deep classifiers are found to be adversarially vulnerable and adversarial attacks provide a powerful tool to evaluate classification robustness. For our exper-

Table 1: Test accuracy on CIFAR datasets. Average accuracy larger than 0 but less than 0.1 is denoted as 0^* without standard deviation.

Dataset	Network	Loss	Clean acc %	*PGD-100*			*AutoAttack*		
				$\alpha = 2/225$	$\alpha = 4/225$	$\alpha = 8/225$	$\alpha = 2/225$	$\alpha = 4/225$	$\alpha = 8/225$
CIFAR-10	ResNet-18	CE	**95.15 (0.11)**	8.81 (1.61)	0.65 (0.24)	0	2.74 (0.09)	0	0
		SL	95.04 (0.07)	**30.53 (0.92)**	**6.64 (0.67)**	**0.86 (0.24)**	**4.10 (0.50)**	$\mathbf{0}^*$	0
	WRN-16-10	CE	93.94 (0.16)	1.04 (0.10)	0	0	0.33 (0.06)	0	0
		SL	**95.02 (0.11)**	**37.47 (0.61)**	**23.16 (1.28)**	**7.88 (0.72)**	**5.37 (0.50)**	$\mathbf{0}^*$	0
CIFAR-100	ResNet-50	CE	**79.82 (0.14)**	2.31 (0.07)	0^*	0	0.99 (0.10)	0^*	0
		SL	78.91 (0.14)	**13.76 (1.30)**	**4.63 (1.20)**	**1.21 (0.80)**	**3.67 (0.60)**	**0.16 (0.05)**	0
	WRN-16-10	CE	77.89 (0.21)	0.83 (0.07)	0^*	0	0.42 (0.07)	0	0
		SL	**79.65 (0.15)**	**6.48 (0.40)**	**0.42 (0.04)**	$\mathbf{0}^*$	**2.73 (0.20)**	$\mathbf{0}^*$	0

iment, we consider the classic PGD attack (Madry et al., 2017) and the state-of-the-art AutoAttack (Croce & Hein, 2020), with attack strength level 2/255, 4/255, 8/255. AutoAttack contains both white-box and black-box attacks and offers a more comprehensive evaluation of adversarial robustness. The results are reported in Table 1. Even though classifiers trained with square loss is far away from adversarially robust, but it consistently gives higher adversarial accuracy.

One thing to notice is that when constructing white-box attacks, square loss will not work well since it doesn't directly reflect the classification accuracy. More specifically, for a correctly classified image $(\boldsymbol{x}, y)$, maximizing the square loss may result in linear scaling of the classifier $f(\boldsymbol{x})$, which doesn't change the predicted class (see Appendix F.2 for more discussion). To this end, we consider a special attack for classifiers trained by square loss by maximizing the cosine similarity between $f(\boldsymbol{x})$ and $\boldsymbol{v}_y$. In our experiments, this special attack rarely outperforms the standard PGD with cross-entropy and the reported PGD accuracy are from the latter settings. This property of square loss may be an advantage in defending adversarial attacks.

Model calibration The predicted class probabilities for square loss can be obtained from Equation 4.1. Expected calibration error (ECE) measures the absolute difference between predicted confidence and the actual accuracy. Deep classifiers are usually found to be over-confident (Vaicenavicius et al., 2019). Using ResNet as an example, we report the typical reliability diagram in Figure 3. On CIFAR-10 with ResNet-18, the average ECE for cross-entropy is 0.028 (0.002) while that for square loss is 0.0097 (0.001). On CIFAR-100 with ResNet-50, the average ECE for cross-entropy is 0.094 (0.005) while that for square loss is 0.068 (0.005). Square loss results are much more calibrated with significantly smaller ECE.

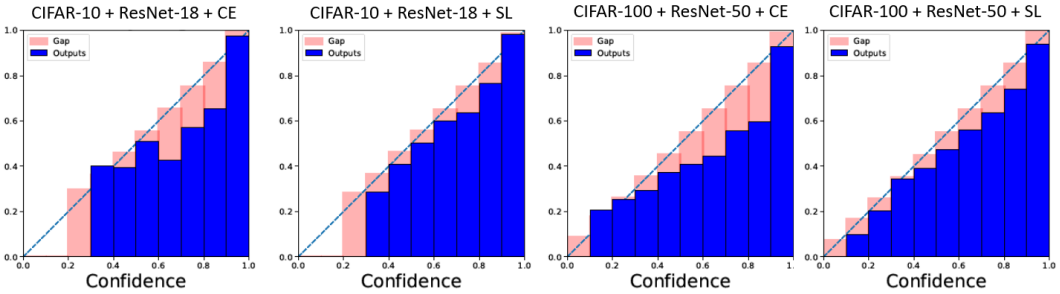

Figure 3: Reliability diagrams of ResNet-18 on CIFAR-10 and ResNet-50 on CIFAR-100. Square loss trained models behave more well-calibrated while cross-entropy trained ones tend to be visibly more over-confident.

6 Conclusions

Classification problems are ubiquitous in deep learning. As a fundamental problem, any progress in classification can potentially benefit numerous relevant tasks. Despite its lack of popularity in practice, square loss has many advantages that can be easily overlooked. Through both theoretical analysis and empirical studies, we identify several ideal properties of using square loss in training neural network classifiers, including provable fast convergence rates, strong robustness, and small calibration error. We encourage readers to try square loss in your own application scenarios.

Ethnics Statement We acknowledge the ICLR Code of Ethics. This submission is mostly theoretical and the authors could not think of any potential violations of them in this submission.

Reproducibility Statement For our theoretical results, explanations of assumptions can be found in Appendix C and a complete proof of the claims can be found in the Appendix D and Appendix E. Our experiment details can be found in Appendix F, with both data and training descriptions.

REFERENCES

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019a.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019b.

Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 35(2):608–633, 2007.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Colin Bennett and Robert C Sharpley. *Interpolation of Operators*. Academic press, 1988.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019.

Yuan Cao and Quanquan Gu. Generalization error bounds of gradient descent for learning overparameterized deep ReLU networks. *arXiv preprint arXiv:1902.01384*, 2019.

Lin Chen and Sheng Xu. Deep neural tangent kernel and Laplace kernel have the same RKHS. *arXiv preprint arXiv:2009.10683*, 2020.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pp. 2206–2216. PMLR, 2020.

Ahmet Demirkaya, Jiasi Chen, and Samet Oymak. Exploring the role of loss functions in multiclass classification. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–5. IEEE, 2020.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

David Eric Edmunds and Hans Triebel. *Function Spaces, Entropy Numbers, Differential Operators*, volume 120. Cambridge University Press, 2008.

Gamaleldin F Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. *arXiv preprint arXiv:1803.05598*, 2018.

Amnon Geifman, Abhay Yadav, Yoni Kasten, Meirav Galun, David Jacobs, and Ronen Basri. On the similarity between the Laplace and neural tangent kernels. *arXiv preprint arXiv:2007.01580*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 588–597, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Tianyang Hu, Wenjia Wang, Cong Lin, and Guang Cheng. Regularization matters: A nonparametric perspective on overparametrized neural network. In *International Conference on Artificial Intelligence and Statistics*, pp. 829–837. PMLR, 2021.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations*, 2020.

Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pp. 8571–8580, 2018.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks. *arXiv preprint arXiv:1909.12292*, 2019.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Yongdai Kim, Ilsang Ohn, and Dongha Kim. Fast convergence rates of deep neural networks for classification. *arXiv preprint arXiv:1812.03599*, 2018.

Michael Kohler and Adam Krzyzak. On the rate of convergence of local averaging plug-in classification rules under a margin condition. *IEEE Transactions on Information Theory*, 53(5):1735–1742, 2007.

Simon Kornblith, Honglak Lee, Ting Chen, and Mohammad Norouzi. Demystifying loss functions for classification. 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Enno Mammen and Alexandre B Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *International Conference on Machine Learning*, pp. 7632–7642. PMLR, 2021.

Youssef Mroueh, Tomaso Poggio, Lorenzo Rosasco, and Jean-Jacques Slotine. Multiclass learning with simplex coding. *arXiv preprint arXiv:1209.1360*, 2012.

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

Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv preprint arXiv:1905.10626*, 2019.

Tomaso Poggio and Qianli Liao. Generalization in deep network classifiers trained with the square loss. Technical report, CBMM Memo No, 2019.

Ingo Steinwart, Clint Scovel, et al. Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics*, 35(2):575–607, 2007.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3459–3467. PMLR, 2019.

Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004.

Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems*, 33, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, pp. 56–85, 2004.