Proceedings of Machine Learning Research vol 283:1–12, 2025 7th Annual Conference on Learning for Dynamics and Control

# A Theoretical Analysis of Soft-Label vs Hard-Label Training in Neural Networks

**Saptarshi Mandal** SMANDAL4@ILLINOIS.EDU
*UIUC*

**Xiaojun Lin** XJLIN@IE.CUHK.EDU.HK
*CUHK*

**R. Srikant** RSRIKANT@ILLINOIS.EDU
*UIUC*

**Editors:** N. Ozay, L. Balzano, D. Panagou, A. Abate

## Abstract

Knowledge distillation, where a small student model learns from a pre-trained large teacher model, has achieved substantial empirical success since the seminal work of (Hinton et al., 2015). Despite prior theoretical studies exploring the benefits of knowledge distillation, an important question remains unanswered: why does soft-label training from the teacher require significantly fewer neurons than directly training a small neural network with hard labels? To address this, we first present motivating experimental results using simple neural network models on a binary classification problem. These results demonstrate that soft-label training consistently outperforms hard-label training in accuracy, with the performance gap becoming more pronounced as the dataset becomes increasingly difficult to classify. We then substantiate these observations with a theoretical contribution based on two-layer neural network models. Specifically, we show that soft-label training using gradient descent requires only $O\left(\frac{1}{\gamma^2\epsilon}\right)$ neurons to achieve a classification loss averaged over epochs smaller than some $\epsilon > 0$, where $\gamma$ is the separation margin of the limiting kernel. In contrast, hard-label training requires $O\left(\frac{1}{\gamma^4} \cdot \ln\left(\frac{1}{\epsilon}\right)\right)$ neurons, as derived from an adapted version of the gradient descent analysis in (Ji and Telgarsky, 2020). This implies that when $\gamma \leq \epsilon$, i.e., when the dataset is challenging to classify, the neuron requirement for soft-label training can be significantly lower than that for hard-label training. Finally, we present experimental results on deep neural networks, further validating these theoretical findings.

**Keywords:** Knowledge Distillation, Projected Gradient Descent, Model Compression

## 1. Introduction

Knowledge distillation is a popular technique for training a smaller 'student' machine learning model by transferring knowledge from a large pre-trained 'teacher' model. A lightweight machine learning model is useful in many resource-constrained application scenarios, such as cyber-physical systems, mobile devices, edge computing, AR/VR, etc. due to several reasons such as limitations in memory, inference speed, and training data availability. Knowledge distillation has been proven to be a powerful solution for these challenges through model compression (Hinton et al., 2015; Buciluundefined et al., 2006; Gou et al., 2021b). Among the various methods of knowledge distillation (Gou et al., 2021a), one prevalent approach involves using the teacher model's output logits as soft targets for training the student model. Specifically, smaller models trained with well-designed soft labels demonstrate competitive performance compared to more complex models trained with original (hard) labels. This method and its variants have been shown to be effective in many settings such as object detection (Chen et al., 2017), reinforcement learning (Xu et al., 2020) and recommendation systems (Pan et al., 2019).

The empirical success of knowledge distillation has motivated many theoretical studies that aim to understand why training with soft-labels from the teacher is more effective than training with hard-labels directly. However, to the best of our knowledge, prior works (reviewed in the Related Work subsection later) do not explain why, when training a neural network, soft-label training can succeed with much fewer number of neurons than hard-label training. To the best of our knowledge, this work is the first to provide theoretical insight into this phenomenon.

Our contributions in this paper can be summarized as follows:

1. As the motivation to understand why knowledge distillation is effective, we first present (in section 2) a few experimental observations using a simple model: a 2-layer fully connected neural network with ReLU activation trained for binary classification on a dataset derived from MNIST, trained via projected gradient descent. We observed that for small neural networks, soft-label training reaches higher accuracy than hard-label training. While this is a well-known phenomenon, to understand the conditions under which soft-label training is really effective, we considered two versions of the dataset, one in which there were more digits and the other where there were fewer digits. We observed that the the performance gain due to soft-label training becomes more significant when the dataset is difficult to classify (i.e., the one with more digits) red(can we add more experiments on the two-layer neural network). Our theoretical models and results are motivated by these experiments and explain these experimental observations.

2. We theoretically analyze the training dynamics of a 2-layer neural network student model. Assuming access to the soft labels provided by the infinite-width teacher (which can be modeled by the limiting kernel of the neural network), we train the student to minimize the cross-entropy loss to the soft labels. We show that soft-label training using projected gradient descent requires $O\left(\frac{1}{\gamma^2\epsilon}\right)$ neurons to achieve a classification loss averaged over epochs smaller than $\epsilon$, where $\gamma$ is the separation margin of the limiting kernel. For the hard label setting, we adapt the gradient descent results of (Ji and Telgarsky, 2020) to our projected gradient descent framework, which requires $O(1/\gamma^4)$ neurons. This result highlights the superiority of soft-label training in terms of neuron efficiency, particularly in challenging classification scenarios when $\gamma$ is very small.

3. Finally, we perform experiments on deep learning models with real-world datasets. Our experiments confirm that the above insight is not only valid for shallow 2-layer networks, but also deeper networks like VGG and ResNet.

### 1.1. Related Work

There are many papers that aim to explain different aspects of knowledge distillation. In this section, we focus on reviewing the works that contributes to the theoretical understanding of knowledge distillation.

A few prior works focus on linear models using the cross-entropy loss for the distillation objective. (Phuong and Lampert, 2021) assumes that both the student and teacher models are of the form $f(x) = w^\top x$, where $w$ is the weight vector of the student network, $x$ is the input vector, and

$f(x)$ is the output. They show that, under certain conditions, the student model converges to the same weight vector as the teacher model. (Ji and Zhu, 2020) extends this result to NTK-linearized deep networks, where the width of the student network approaches infinity. Similarly, (Panahi et al., 2022) demonstrates that the predictions from a neural network trained with soft labels from a teacher network match those from the teacher when the student is an extremely wide two-layer neural network.

(Das and Sanghavi, 2023) study a similar setup but highlight that self-distillation can mitigate the impact of noisy data. On the other hand, (Mobahi et al., 2020) explain the benefits of self-distillation through improved regularization, focusing on fitting a nonlinear function to training data within a Hilbert space.

However, to the best of our knowledge, none of these studies address the central question of whether soft-label training enables the use of significantly fewer neurons compared to hard-label training, which is the primary focus of our paper. A significant limitation of these linear models is that they require the student and teacher to share the same feature representations. While this assumption may be reasonable for self-distillation, it fails to capture the fundamental principle of model compression, where the student model is often much smaller and has a different feature space than the teacher.

Another line of work explores the benefits of knowledge distillation by arguing that distillation primarily acts to reduce the variance of the empirical risk relative to training with hard labels, thereby improving generalization performance. (Menon et al., 2020) argue that using Bayes class probabilities as targets, instead of hard labels, reduces the excess risk associated with the function set learned by the student, leading to better generalization. Similarly, (Zhou et al., 2021) propose that soft labels provide a bias-variance trade-off in the cross-entropy loss for the student.

However, despite the simplicity of their approaches, both (Menon et al., 2020) and (Zhou et al., 2021) focus only on the excess risk component of the generalization error and do not provide insights into the training dynamics or behavior of the student during learning.

In summary, prior works do not address the following question: why does distillation using soft labels from a teacher lead to good performance using a smaller neural network compared to training with hard labels? Answering this question is the main focus of our paper.

## 2. Preliminary Experimental Observations

In this section, we present preliminary experimental observations on the training of a two-layer neural network student model using both ground truth hard labels and soft labels generated by a larger teacher network. For the experiments summarized in Table 2, we perform binary classification on the MNIST dataset, where digits greater than 4 are labeled as class 1 and others as class 0.

Two configurations of the MNIST data are considered:

1. **Full dataset:** Includes images of all digits (0–9).
2. **Reduced dataset:** Excludes images corresponding to the digits $\{1, 7, 4, 9\}$.

The first configuration is more challenging to classify because of the difficulty in distinguishing between visually similar digits such as 1 and 7 or 4 and 9. Our observations indicate that the performance of the student model trained with soft labels remains relatively stable when transitioning from the easier dataset to the harder one. In contrast, the performance of the student model trained with hard labels shows a more significant decline. For instance, in the case of a student network with 4 neurons, the accuracy drops significantly when moving from the reduced dataset to the full dataset under hard-label training. However, with soft-label training, the performance experiences only a minimal drop in accuracy.

| Dataset | Teacher Net | Student Net | Teacher Acc | St_Hard Acc | St_Soft Acc |
|---|---|---|---|---|---|
| all except 1,7,4,9 | 512 | 8 | 98.75 | 97.19 | 97.67 |
| | 512 | 6 | 98.75 | 96.84 | 97.26 |
| | 512 | 4 | 98.75 | 93.04 | 95.83 |
| all digits | 512 | 8 | 98.34 | 95.77 | 96.58 |
| | 512 | 6 | 98.34 | 95.33 | 96.00 |
| | 512 | 4 | 98.34 | 88.84 | 95.62 |

Table 1: Performance Comparison on Different Datasets for the 2-layer Model. The second and third columns denote the number of neurons in the hidden layer for the teacher and student networks, respectively.

## 3. Theoretical Results

In the previous section, we presented a few experimental observations on student-teacher training using a simple model of two-layer neural network. In this section, we provide theoretical insights into these observations, summarized in Theorem 2 and the corresponding Corollary 4. We begin by introducing the student-teacher model used for training, outlining a few assumptions about the training data, and specifying the choice of distillation loss for the student to be trained using soft labels. This is followed by a description of the Projected Gradient Descent (PGD) method used for training and a theoretical analysis of the training dynamics.

### 3.1. Model and Assumptions

We consider the following fully-connected two-layer neural network,with $m$ neurons in its hidden layer and with ReLU activation, as the student model:

$$f(x; W, a) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \sigma(W_j^\top x) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \mathbf{1}(W_j^\top x \geq 0) W_j^\top x, \tag{1}$$

where $\sigma(z) = \max(0, z)$ is the ReLU activation function, $a_j \in \mathbb{R}$ and $W_j \in \mathbb{R}^d$ for $j \in [1, m]$ are respectively the final layer weight and the hidden layer weight vector corresponding to each neuron $j$ in the hidden layer. Let the vector $a$ and the matrix $W$ denote the collection of $a_j$ and $W_j$ as their $j^{\text{th}}$ element and $j^{\text{th}}$ row, respectively. Let the neural network output for an input sample $x_i$ be further denoted as $f_i(W)$ for any $i \in [n]$.

We initialized the neural network using the symmetric random initialization which was proposed in (Bai and Lee, 2020) and later used in (Cayci et al., 2023). The initial parameters are given as follows: $a_j = -a_{j+\frac{m}{2}} \overset{\text{i.i.d.}}{\sim} \text{Unif}\{-1, 1\}$ and $W(0)_j = W(0)_{j+\frac{m}{2}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$ independent and identically distributed over $j = 1, 2, .., \frac{m}{2}$ and are independent from each other. For this symmetric initialization, we assume $m$ is an even number. The symmetric initialization ensures that $f_i(W(0)) = 0, \forall i \in [n]$.

The above model of neural network is studied in detail for hard-label training in the works by (Ji and Telgarsky, 2020), (Du et al., 2019), (Arora et al., 2019), etc. Similar to the setting in these prior works, we fix the final layer weight vector $a$ and only train the hidden layer weight matrix $W$.

The dataset with the ground truth values under consideration is denoted by $\mathcal{D} \coloneqq \{(x_i, y_i)\}_{i=1}^n$ for some finite integer $n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. For simplicity, assume $\|x_i\|_2 \leq 1, \forall i \in [1, n]$, which is standard in prior works. Let $x_i, \forall i \in [n]$ also satisfy: $\|x_i\| \geq c$ for some $c > 0$.

We make the following assumption on the dataset $\mathcal{D}$ characterizing the separability by the corresponding infinite-width NTK as in Ji and Telgarsky (2020): Let $\mu_{\mathcal{N}}$ be the Gaussian measure on $\mathcal{R}^d$, given by the Gaussian density with respect to the Lebesgue measure on $\mathbb{R}^d$. We consider the following Hilbert space

$$\mathcal{H} \coloneqq \left\{ w : \mathbb{R}^d \to \mathbb{R}^d \,\middle|\, \int \|w(z)\|_2^2 \, d\mu_{\mathcal{N}}(z) < \infty \right\}. \tag{2}$$

For any $x \in \mathbb{R}^d$, define $\phi_x \in \mathcal{H}$ by $\phi_x(z) \coloneqq \mathbf{1}\left(\langle z, x \rangle > 0\right) x$.

**Assumption 1** *There exists $\overline{v} \in \mathcal{H}$ and $\gamma > 0$, such that $\|\overline{v}(z)\|_2 \leq 1$ for any $z \in \mathcal{R}^d$, and for any $1 \leq i \leq n$,*

$$y_i \langle \overline{v}, \phi_i \rangle_{\mathcal{H}} \coloneqq y_i \int \langle \overline{v}(z), \phi_i(z) \rangle \, d\mu_{\mathcal{N}}(z) \geq \gamma. \tag{3}$$

Using standard notations, $\phi_x(W_j(0)) = \mathbf{1}(W_j(0)^\top x > 0)x$, for all $j \in [m]$ are called the NTK-features for input data $x$. Let us denote $\overline{\phi}_x^0 \in \mathbb{R}^{md \times 1}$ as a concatenation of all $\phi_x(W_j(0))$ for input data $x$. The above assumption ensures the separability of the induced set $\left\{\overline{\phi}_{x_i}^0, y_i\right\}_{i=1}^n$ when $m$ is sufficiently large (see Lemma 2.3 in (Ji and Telgarsky, 2020)). As shown in (Ji and Telgarsky, 2020), there is always a $\gamma > 0$ satisfying assumption 1 as long as no two inputs $x_i$ and $x_j$ with $i, j \in [1, n], i \neq j$ are parallel in the dataset $\mathcal{D}$.

For our theoretical results, we define the soft labels for the student model based on the Hilbert space $\mathcal{H}$, as described below.

Let $v : \mathbb{R}^d \to \mathbb{R}^d$ be a function in the RKHS $\mathcal{H}$ such that $\|v(z)\|_2 \leq 1$ for all $z \in \mathbb{R}^d$, and $y_i \mathbb{E}_{z \sim \mathcal{N}(0, I_d)}(\phi_i(z)^\top v(z)) \geq \gamma$. Assumption 1 ensures the existence of such a $v(z)$. Define for each $i \in [n]$

$$z_i \coloneqq \mathbb{E}_{z \sim \mathcal{N}(0, I_d)}(\phi_i(z)^\top v(z)).$$

The soft labels for each input $x_i$ are then given by

$$p_i = \mu(z_i) \coloneqq \frac{1}{1 + \exp(-z_i)},$$

where $\mu(z)$ denotes the softmax function applied to the target logit $z_i$. In other words, we assume that we have access to the soft labels from a teacher which is the kernel limit of an inifnite-width neural network. We fix the function $v$ for the remainder of the paper. The choice of a particular $v$ does not affect our analysis, provided it satisfies the conditions described above.

Our training objective is to minimize the following empirical risk over the dataset $\mathcal{D}$:

$$R^{KL}(W) := \frac{1}{n}\sum_{i=1}^{n} \ell^{KL}(p_i, f_i(W)), \tag{4}$$

where $\ell^{KL}(p_i, f_i(W))$ is the Kullback-Leibler (KL) divergence between the soft-label $p_i$ and the softmax version of the network output $f_i(W)$:

$$\ell^{KL}(p_i, f_i(W)) := p_i \ln\left(\mu(f_i(W))/p_i\right) + (1-p_i)\ln\left((1-\mu(f_i(W)))/(1-p_i)\right).$$

We now discuss the student training procedure. The student is trained on soft labels using a projected gradient descent (PGD) algorithm. First, we define $\nabla_{W_j} R^{KL}(W)$ which serves as the gradient of the risk with respect to the weight for the $j^{\text{th}}$ neuron:

$$\nabla_{W_j} R^{KL}(W) = \frac{1}{n}\sum_{i=1}^{n} \nabla_f \ell^{KL}(p_i, f_i(W)) \nabla_{W_j} f_i(W) \tag{5}$$

where,

$$\nabla_{W_j} f_i(W) = \frac{a_j}{\sqrt{m}}\phi_i(W_j) = \frac{a_j}{\sqrt{m}}\mathbf{1}(W_j^\top x_i > 0).$$

Note that the gradient of the loss is defined for the whole domain of $W_j$ in $\mathcal{R}^d$ even though the ReLU activation function is not differentiable at 0. Let $W(t)$ denotes the weight matrix of the neural network after $t^{\text{th}}$ iteration of training. Let the feasible set of weights be defined as $S_B := \{W : \|W_j - W_j(0)\|_2 \leq \frac{B}{\sqrt{m}}\}$, where $B$ is a hyperparameter in the training. The PGD algorithm updates the weights per iteration using the following steps:

1. **Descent step:** $\hat{W}_j(t+1) = W_j(t) - \eta \nabla_{W_j} R^{KL}(W_j(t)), \quad \forall t \geq 0.$
2. **Projection step:** $W_j(t+1) =$ Projection of $\hat{W}_j(t+1)$ into $S_B$.

In the next subsection we provide the key theoretical result of this paper. In Theorem 2, we characterize the neuron requirement to achieve an arbitrarily small training loss averaged over epochs. We then compare this with the neuron requirement for hard-label training, as established for Gradient Descent in (Ji and Telgarsky, 2020), by adapting their result to Projected Gradient Descent in this paper.

### 3.2. Neuron Requirement for Soft-label Training

We are now ready to present the first result of this paper. The following Theorem provides a characterization of the number of neurons required for soft-label training to achieve a small empirical risk, $R^{KL}(W(t))$, averaged over iterations $t < T$.

**Theorem 2** *Let $\beta \in (0,1)$, $\delta \in \left(0, \frac{1}{3}\right)$ be fixed real numbers. If the number of neurons $m$ satisfies*

$$m \geq \frac{C_1}{\beta}\left(\sqrt{\frac{2}{\pi}}\frac{1}{c} + 3\sqrt{\ln\left(\frac{2n}{\delta}\right)}\right)^2, \tag{6}$$

*and the PGD algorithm is run with a projection radius $B = 1$ for $T$ iterations such that $T \geq \frac{9}{\beta^2}$, using a constant step size $\eta$ satisfying $\eta \leq \frac{\beta}{3}$. Then, the following bound on the averaged empirical risk holds:*

$$\frac{1}{T}\sum_{\tau<T} R^{KL}(W(\tau)) \leq \beta, \tag{7}$$

*with probability at least $1 - 3\delta$ over the random initialization. Here, $C_1 = \frac{96}{(1+e^2)}$ is an absolute constant.*

In light of the above Theorem, we next analyze the performance of the student network in classifying the training data. We define the classification error as:

$$R(W) \coloneqq R(W; \mathcal{D}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(y_i f_i(W) > 0). \tag{8}$$

Using a reverse Pinsker inequality (Lemma 4.1 of (Götze et al., 2019)), we provide an upper bound on $R(W)$ in terms of $R^{KL}(W)$:

**Lemma 3** *If $0 < p_i < 1$, where $p_i$ is the soft label in the above construction for the data input $x_i$ for $i \in [n]$, then the classification loss $R(W)$ can be related to the surrogate loss $R^{KL}(W)$ as follows:*

$$R(W(t)) \leq \frac{32}{\gamma^2} R^{KL}(W(t)). \tag{9}$$

Combining the above Lemma with Theorem 2, we arrive at the following Corollary:

**Corollary 4** *Let $\epsilon \in (0, 1)$ and $\delta \in \left(0, \frac{1}{3}\right)$ be fixed real numbers. If the number of neurons satisfies:*

$$m \geq \frac{C_2}{\gamma^2 \epsilon}\left(\sqrt{\frac{2}{\pi}}\frac{1}{c} + 3\sqrt{\ln\left(\frac{2n}{\delta}\right)}\right)^2, \tag{10}$$

*then the PGD algorithm with $B = 1$, step size $\eta \leq \frac{\gamma^2\epsilon}{3}$, and iteration count $T \geq \frac{9}{\gamma^4\epsilon^2}$ ensures the following guarantee on the classification loss with probability at least $1 - 3\delta$:*

$$\frac{1}{T}\sum_{\tau<T} R(W(\tau)) \leq \epsilon, \tag{11}$$

*where $C_2 = 32 \cdot C_1$.*

#### 3.2.1. COMPARISON WITH HARD-LABEL TRAINING

Now we are ready to compare the neuron requirement based on Corollary 4 with that of the requirement for hard label training as established in (Ji and Telgarsky, 2020). The empirical risk for hard-label training on the dataset $\mathcal{D} \coloneqq \{x_i, y_i\}_{i=1}^n$ is defined as:

$$R^h(W) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \ln(1 + \exp(-y_i f_i(W))). \tag{12}$$

The following proposition for hard-label training is adapted from the result on gradient descent with hard labels by (Ji and Telgarsky, 2020), modified to fit our projected gradient descent (PGD) setting. The modification introduces a projection step after each weight update, where the weight of each neuron is constrained to the set $W_j - W_j(0) \leq \frac{B}{\sqrt{m}}$, for some hyperparameter $B > 0$. The proof of Proposition 5 is provided in the Appendix of the full version of the paper (link).

**Proposition 5** *Fix $\beta \in (0, 1)$ and $\delta \in \left(0, \frac{1}{3}\right)$. With a choice of the projection ball radius $B = \frac{2}{\gamma}\ln\left(\frac{2}{\ln(2)\beta}\right)$, if the number of training iterations satisfies $T \geq \frac{8}{\gamma^2\eta}\frac{\ln^2\left(\frac{2}{\ln(2)\beta}\right)}{\ln(2)\beta}$, $\eta \leq 1$, and the number of neurons satisfies:*

$$m \geq \frac{16}{\gamma^4}\left(\frac{2\sqrt{2}}{c\sqrt{\pi}}\ln\left(\frac{2}{\ln(2)\beta}\right) + 3\sqrt{\ln\left(\frac{2n}{\delta}\right)}\right)^2, \tag{13}$$

*then the following holds with probability at least $1 - 3\delta$*

$$\frac{1}{T}\sum_{\tau<T} R^h(W(\tau)) \leq \beta. \tag{14}$$

The hard-label surrogate loss guarantee in equation (14) implies the following classification loss guarantee:

$$\frac{1}{T}\sum_{\tau<T} R(W(\tau)) \leq \frac{1}{\ln(2)}\beta.$$

It is worth mentioning that, without the projection step, the neuron requirement for hard-label training, as established by (Ji and Telgarsky, 2020), is $O\left(\frac{1}{\gamma^8}, \ln\left(\frac{n}{\delta}\right), \ln\left(\frac{1}{\epsilon}\right)\right)$.

**Comparing the neuron requirements:** Based on Corollary 4 and Proposition 5, the requirements for hard-label and soft-label training to ensure the classification loss averaged over iterations is less than some $\epsilon > 0$ can be expressed as:

$$O\left(\frac{1}{\gamma^4}, \ln\left(\frac{n}{\delta}\right), \ln\left(\frac{1}{\epsilon}\right)\right) \quad \text{(hard labels)},$$

and

$$O\left(\frac{1}{\gamma^2\epsilon}\ln\left(\frac{n}{\delta}\right), \frac{1}{\beta}\right) \quad \text{(soft labels)}.$$

Suppose that there is a target classification error of epsilon. The above results suggest that, when $\gamma < \epsilon$, i.e., when the data-set is more difficult the separate, soft-label training reduces the neuron requirement by a factor of $\gamma$ in the regime of PGD training. In other words, when the separation margin $\gamma$ associated with the classification task is sufficiently small, soft-label training requires significantly fewer neurons to achieve similar performance compared to hard-label training.

### 3.3. Proof Sketch of Theorem 2

While the detailed proofs of all the results presented in the paper is provided in the Appendix of the full version of the paper (link), we present a high-level proof sketch for Theorems 2 here. Before presenting the proof sketch, we first introduce some additional notations and quantities.

Similar to the definition of $\overline{\phi}_i^0$, define the feature map at iteration $t$, $\overline{\phi}_i^t$, based on the weight $W(t)$. Specifically, the feature corresponding to the $j^{\text{th}}$ neuron is given by $\phi_i(W_j(t))$. Now, define $f_i^t(W)$ for each data sample $x_i$ as:

$$f_i^t(W) \coloneqq \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \mathbf{1}(W_j(t)^\top x_i \geq 0) W_j^\top x_i = \frac{1}{\sqrt{m}} \langle \overline{\phi}_i^t, W \rangle,$$

where $W_j(t)$ is the weight of the $j^{\text{th}}$ neuron at the $t^{\text{th}}$ iteration of PGD.

Using these definitions, we define the following expression for each $t \leq T$:

$$R^{t,KL}(W) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \ell^{KL}(p_i, f_i^t(W)).$$

**Initial Feature Map and Separability:** The following Lemma 6 implies the existence of a weight matrix $U$ satisfying $\|U_j - W_j(0)\|_2 \leq \frac{1}{\sqrt{m}}$ such that $|\langle \overline{\phi}_i^0, U \rangle - z_i|$ is small for all $i \in [n]$, with high probability, when $m$ is sufficiently large.

**Lemma 6** *Let $U \in \mathbb{R}^{m \times d}$ be defined as $U_j = \frac{a_j}{\sqrt{m}} v(W_j(0)), \forall j \in [m]$. Then, under Assumption 1, for the symmetric random initialization of the neural network weights, the following holds with probability at least $1 - \delta$:*

$$\left| f_i^0(U) - z_i \right| \leq \frac{1}{\sqrt{m}} \sqrt{2 \ln(2n/\delta)}, \quad \forall i \in [1, n], \tag{15}$$

This observation suggests that a linear function (linear in the weight $W$) of the form $\langle \overline{\phi}_i^0, W \rangle$ can approximate the soft label $z_i$ for each $i$ with high probability. This intuition is crucial for the subsequent steps of the proof.

**Convergence of Soft Label Surrogate Loss:** Next we show that under sufficient conditions on $T$ and $\eta$, the soft-label risk averaged over all iterations converges to the quantity $\frac{1}{T} \sum_{t<T} R^{t,KL}(\overline{W})$ for any $\overline{W}$ in the feasible set $S_B$.

The remainder of the proof is devoted to showing that $R^{t,KL}(\overline{W})$ is small for all $t \leq T$ with high probability, for an appropriately chosen $\overline{W}$.

**Bounding $R^{t,KL}(\overline{W})$:** The idea is to bound $\ell^{KL}(p_i, f_i^t(\overline{W}))$ for each $i$ with high probability. For this purpose, we use a reverse Pinsker's inequality, as stated in Lemma 4.1 of (Götze et al., 2019). This inequality allows us to upper-bound the KL divergence loss $\ell^{KL}(p_i, f_i^t(\overline{W}))$ for each $i$ by the distance between the corresponding logits, $|z_i - f_i^t(\overline{W})|$.

Now, choosing $\overline{W} = W(0) + U$ and applying the triangle inequality, we obtain:

$$|z_i - f_i^t(\overline{W})| \leq |z_i - f_i^0(U)| + |f_i^t(U) - f_i^0(U)| + |f_i^t(W(0)) - f_i^0(W(0))|.$$

From Lemma 6, we know that $\|z_i - f_i^0(U)\|_2$ is small under the sufficient conditions. Therefore, it remains to show that the terms $|f_i^t(U) - f_i^0(U)|$ and $|f_i^t(W(0)) - f_i^0(W(0))|$ are both small for each $i$ with high probability. We use the analysis in the neural tangent kernel literature ((Jacot et al., 2020; Chen et al., 2020)) for bounding this expression.

## 4. More Experimental Results

In this section, we validate our results using deep neural networks, with VGG 8+3 as the teacher and VGG 2+3 as the student; see (Simonyan and Zisserman, 2014) for a detailed description of the VGG architecture. The dataset chosen for our experiments is the CIFAR-10 cat/dog dataset.

Since our theory suggests that harder-to-classify datasets benefit more from soft-label training, we added Gaussian noise to the CIFAR-10 cat/dog dataset to make it more challenging to classify. We compare the performance of soft-label vs. hard-label training across different datasets: the original CIFAR-10 dataset and noise-added CIFAR-10 datasets. The results are summarized in Figure 4. Consistent with our theoretical predictions, the experiments demonstrate that harder-to-classify datasets benefit significantly more from distillation.

All experimental points are averaged over 10 independent runs. For each run, we employed early stopping with a patience of 20 epochs and a maximum of 100 iterations. The dataset was split into training, validation, and test sets with proportions of 80%, 10%, and 10%, respectively. We trained the models using gradient descent with the Adam optimizer and applied $L_2$ regularization.

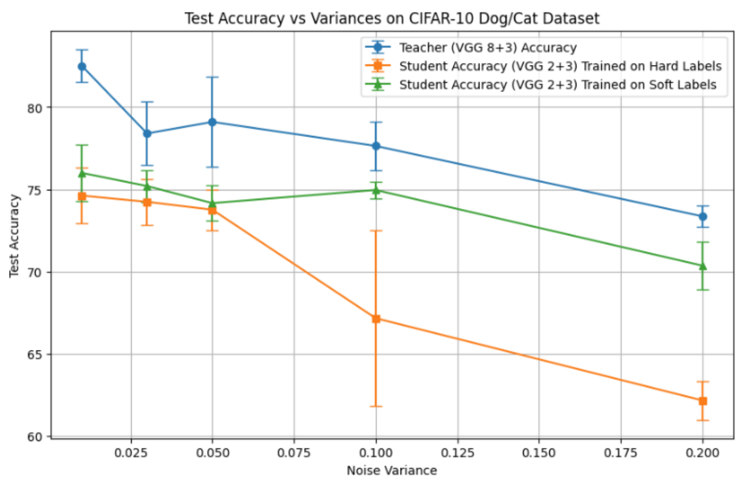


Figure 1: Classification accuracy under Gaussian noise on CIFAR-10 cat/dog with VGG 8+3 as the teacher and VGG 2+3 as the student.

## 5. Conclusions

In this paper, we provide theoretical results which show that soft-label training leads to fewer number of neurons than hard–label training for the same training accuracy. Our proofs provide some intuition for this phenomenon, as stated at the end of the longer version of the paper, which we summarize here. The parameters of a neural network have a dual role: one is to identify good features and the other is to assign weights to these features. With good initialization, one can start the training with good features. In contrast to hard-label training, soft-label training ensures that the network parameters do not deviate too much from initialization, thus approximately maintaining good features throughout the training process, but they deviate just enough to assign the right weights to various features.

## 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, 2019.

Yu Bai and Jason D. Lee. Beyond linearization: On quadratic and higher-order approximation of wide neural networks, 2020.

Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150464. URL https://doi.org/10.1145/1150402.1150464.

Semih Cayci, Siddhartha Satpathi, Niao He, and R. Srikant. Sample complexity and overparameterization bounds for temporal-difference learning with neural network approximation. *IEEE Transactions on Automatic Control*, 68(5):2891–2905, 2023. doi: 10.1109/TAC.2023.3234234.

Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/e1e32e235eee1f970470a3a6658dfdd5-Paper.pdf.

Zixiang Chen, Yuan Cao, Quanquan Gu, and Tong Zhang. A generalized neural tangent kernel analysis for two-layer neural networks. *Advances in Neural Information Processing Systems*, 33: 13363–13373, 2020.

Rudrajit Das and Sujay Sanghavi. Understanding self-distillation in the presence of label noise, 2023.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks, 2019.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, March 2021a. ISSN 1573-1405. doi: 10.1007/s11263-021-01453-z. URL http://dx.doi.org/10.1007/s11263-021-01453-z.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021b.

Friedrich Götze, Holger Sambale, and Arthur Sinulis. Higher order concentration for functions of weakly dependent random variables. *Electronic Journal of Probability*, 24(none), January 2019. ISSN 1083-6489. doi: 10.1214/19-ejp338. URL http://dx.doi.org/10.1214/19-EJP338.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks, 2020.

Guangda Ji and Zhanxing Zhu. Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20823–20833. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ef0d3930a7b6c95bd2b32ed45989c61f-Paper.pdf.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks, 2020.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, Seungyeon Kim, and Sanjiv Kumar. Why distillation helps: a statistical perspective, 2020.

Hossein Mobahi, Mehrdad Farajtabar, and Peter L. Bartlett. Self-distillation amplifies regularization in hilbert space. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Yiteng Pan, Fazhi He, and Haiping Yu. A novel enhanced collaborative autoencoder with knowledge distillation for top-n recommender systems. *Neurocomputing*, 332:137–148, 2019. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.12.025. URL https://www.sciencedirect.com/science/article/pii/S0925231218314796.

Ashkan Panahi, Arman Rahbar, Chiranjib Bhattacharyya, Devdatt Dubhashi, and Morteza Haghir Chehreghani. Analysis of knowledge transfer in kernel regime. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 1615–1624, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557237. URL https://doi.org/10.1145/3511808.3557237.

Mary Phuong and Christoph H. Lampert. Towards understanding knowledge distillation, 2021.

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

Zhiyuan Xu, Kun Wu, Zhengping Che, Jian Tang, and Jieping Ye. Knowledge transfer in multi-task deep reinforcement learning for continuous control. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15146–15155. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/acab0116c354964a558e65bdd07ff047-Paper.pdf.

Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective, 2021.