Towards Understanding Knowledge Distillation

Mary Phuong [1] **Christoph H. Lampert** [1]

Abstract

Knowledge distillation, i.e. one classifier being trained on the outputs of another classifier, is an empirically very successful technique for knowledge transfer between classifiers. It has even been observed that classifiers learn much faster and more reliably if trained with the outputs of another classifier as soft labels, instead of from ground truth data. So far, however, there is no satisfactory theoretical explanation of this phenomenon. In this work, we provide the first insights into the working mechanisms of distillation by studying the special case of linear and deep linear classifiers. Specifically, we prove a generalization bound that establishes fast convergence of the expected risk of a distillation-trained linear classifier. From the bound and its proof we extract three key factors that determine the success of distillation: *data geometry* – geometric properties of the data distribution, in particular class separation, has an immediate influence on the convergence speed of the risk; *optimization bias* – gradient descent optimization finds a very favorable minimum of the distillation objective; and *strong monotonicity* – the expected risk of the student classifier always decreases when the size of the training set grows.

1. Introduction

In 2014, Hinton et al. (2014) made a surprising observation: they found it *easier* to train classifier using the real-valued outputs of another classifier as target values than using actual ground-truth labels. Calling the procedure *knowledge distillation*, or *distillation* for short, they noticed the positive effect to occur even when the existing classifier (called *teacher*) was trained on the same data as it used afterwards for the distillation-training of the new classifier (called *students*). Since that time, the positive properties of distillation-based training has been confirmed several times: the optimization step is generally more well-behaved than the optimization step in label-based training, and it needs less if any regularization or specific optimization tricks. Consequently, in several fields, distillation has become a standard technique for transfering the information between classifiers with different architectures, such as from deep to shallow neural networks or from ensembles of classifiers to individual ones.

While the practical benefits of distillation are beyond doubt, its theoretical justification remains almost completely unclear. Existing explanations rarely go beyond qualitative statements, e.g. claiming that learning from soft labels should be easier than learning from hard labels, or that in a multi-class setting the teacher's output provides information about how similar different classes are to each other.

In this work, we follow a different approach. Instead of studying distillation in full generality, we restrict our attention to a simplified, analytically tractable, setting: binary classification with linear teacher and linear student (either shallow or deep linear networks). For this situation, we achieve the first quantitative results about the effectiveness of distillation-based training. Specifically, our main results are: **1) We prove a generalization bound that establishes extremely fast convergence of the risk of distillation-trained classifiers**. In fact, it can reach zero risk from finite training sets. **2) We identify three key factors that explain the success of distillation**: *data geometry* – geometric properties of the data distribution, in particular class separation, directly influence the convergence speed of the student's risk; *optimization bias* – even though the distillation objective can have many optima, gradient descent optimization is guaranteed to find a particularly favorable one; and *strong monotonicity* – increasing the training set always decreases the risk of the student classifier.

2. Related Work

Ideas underpinning distillation have a long history dating back to the work of Ba & Caruana (2014); Bucilua et al. (2006); Craven & Shavlik (1996); Li et al. (2014); Liang et al. (2008). In its current and most widely known form, it was introduced by Hinton et al. (2014) in the context of

[1] IST Austria (Institute of Science and Technology Austria). Correspondence to: Mary Phuong <bphuong@ist.ac.at>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

neural network compression.

Since then, distillation has quickly gained popularity among practitioners and established its place in deep learning folklore. It has been found to work well across a wide range of applications, including e.g. transferring from one architecture to another (Geras et al., 2016), compression (Howard et al., 2017; Polino et al., 2018), integration with first-order logic (Hu et al., 2016) or other prior knowledge (Yu et al., 2017), learning from noisy labels (Li et al., 2017), defending against adversarial attacks (Papernot et al., 2016), training stabilization (Romero et al., 2015; Tang et al., 2016), distributed learning (Polino et al., 2018), reinforcement learning (Rusu et al., 2016) and data privacy (Celik et al., 2017).

In contrast to the empirical success, the mathematical principles underlying distillation's effectiveness have largely remained a mystery. To our knowledge, (Lopez-Paz et al., 2016) is the only work that examines distillation from a theoretical perspective. It casts distillation as a form of learning using privileged information (LUPI, Vapnik & Izmailov 2015), a learning setting in which additional per-instance information is available at training time but not at test time. However, even the LUPI view conceptually falls short of explaining the effectiveness of distillation. In particular, it concentrates on the aspect that the teacher's supervision to the student is noise-free. However, this argument does not suffice to explain, e.g., the success of distillation even when the original problem is noise-free to start with.

A more distantly related topic is *machine teaching* (Zhu, 2015). In machine teaching, a machine learning system is trained by a human teacher, whose goal is to hand-pick as small a training set as possible, while ensuring that the machine learns a desired hypothesis. Transferring knowledge via machine teaching techniques is extremely effective: perfect transfer is often possible from a small finite teaching set (Zhu, 2013; Liu & Zhu, 2016). However, the price for this radical reduction in sample complexity is the expensive training set construction. Our work shows that, at least in the linear setting, distillation achieves a similar effectiveness with a more practical form of supervision.

3. Background: Linear Distillation

We formally introduce distillation in the context of binary classification. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space, $\mathcal{Y} = \{0, 1\}$ the label space, and $P_{\mathbf{x}}$ the probability distribution of inputs. We assume $P_{\mathbf{x}}$ has a density.

The *teacher* $h^* : \mathcal{X} \to \mathcal{Y}$ is a fixed linear classifier, i.e. $h^*(\mathbf{x}) = \mathbb{1}\{\mathbf{w}_*^\intercal \mathbf{x} \geq 0\}$ for some $\mathbf{w}_* \in \mathbb{R}^d \setminus \{\mathbf{0}\}$, where $\mathbb{1}\{.\}$ returns 1 if the argument is true and 0 otherwise. The *student* also is a linear classifier, $h(\mathbf{x}) = \mathbb{1}\{\mathbf{w}^\intercal \mathbf{x} \geq 0\}$.

We allow the weight vector to be parameterised as a product of matrices, $\mathbf{w}^\intercal = \mathbf{W}_N \mathbf{W}_{N-1} \cdots \mathbf{W}_1$ for some $N \geq 1$. When $N \geq 2$, this parameterisation is known as a *deep linear network*. Although deep linear networks have no additional capacity compared to directly parameterised linear classifiers ($N = 1; \mathbf{w}^\intercal = \mathbf{W}_1$), they induce different gradient-descent dynamics, and are often studied as a first step towards understanding deep nonlinear networks (Saxe et al., 2014; Kawaguchi, 2016; Hardt & Ma, 2017).

Distillation proceeds as follows. First, we collect a *transfer set* $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ consisting of inputs $\mathbf{x}_i$ sampled i.i.d. from $P_{\mathbf{x}}$, and *soft labels* $y_i = \sigma(\mathbf{w}_*^\intercal \mathbf{x}_i)$ provided by the teacher, where σ is the sigmoid function, $\sigma(x) = 1/(1 + \exp(-x))$. The soft (real-valued) labels can be thought of as a more informative version of the hard (0/1-valued) labels of the standard classification setting. We write $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ for the data matrix. Second, the student is trained by minimizing the (normalized) cross-entropy loss,

$$L^1(\mathbf{w}) = -\frac{1}{n} \sum_{i=1}^{n} \Big[y_i \log \sigma(\mathbf{w}^\intercal \mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w}^\intercal \mathbf{x}_i)) \Big] - L^*, \quad (1)$$

where L^* is a normalization constant, such that the minimum of L^1 is 0. It only serves the purpose of simplifying notation and has no effect on the optimization.

The student observes the loss as a function of its parameters, i.e. the individual weight matrices,

$$L(\mathbf{W}_1, \dots, \mathbf{W}_N) := L^1((\mathbf{W}_N \mathbf{W}_{N-1} \cdots \mathbf{W}_1)^\intercal), \quad (2)$$

and optimizes it via gradient descent. For the theoretical analysis, we avoid the complications of stepsize selection and adopt the notion of *infinitesimal step size*[1], which turns the gradient descent procedure into a continuous *gradient flow*. We write $\mathbf{W}_i(\tau)$ for the value of the matrix $\mathbf{W}_i$ at time $\tau \in [0, \infty)$, with $\mathbf{W}_i(0)$ denoting the initial value, and $\mathbf{w}(\tau)^\intercal = \mathbf{W}_N(\tau) \cdots \mathbf{W}_1(\tau)$. Then, each $\mathbf{W}_i(\tau)$, for $i \in \{1, \dots, N\}$, evolves according to the following differential equation.

$$\frac{\partial \mathbf{W}_i(\tau)}{\partial \tau} = -\frac{\partial L}{\partial \mathbf{W}_i}(\mathbf{W}_1(\tau), \dots, \mathbf{W}_N(\tau)). \quad (3)$$

The student is trained until convergence, i.e. $\tau \to \infty$. We measure the *transfer risk* of the trained student, defined as the probability that its prediction differs from that of the teacher,

$$R(h) = \mathop{\mathbb{P}}_{\mathbf{x} \sim P_{\mathbf{x}}} [h(\mathbf{x}) \neq h^*(\mathbf{x})]. \quad (4)$$

In Section 4.2, we will derive a bound for the transfer risk and establish how rapidly it decreases as a function of n.

[1] For readers who are unfamiliar with gradient flows, it suffices to think of the stepsize as finite and ”sufficiently small”.

4. Generalization Properties of Linear Distillation

This section contains our main technical results. First, in Section 4.1, we provide an explicit characterization of the outcome of distillation-based training in the linear setting. In other words, we identify *what the student actually learns*. In particular, we prove that the student is able to perfectly identify the teacher's weight vector, if the number of training examples (n) is equal to the dimensionality of the data (d) or higher. If less data is available, under minor assumptions, the student finds the best approximation of the teacher's weight vector that is possible within the subspace spanned by the training data.

In Section 4.2 we use these results to study the generalization properties of the student classifier, i.e. we characerize *how fast the student learns*. Specifically, we prove a generalization bound with much more appealing properties than what is possible in the classic situation of learning from hard labels. As soon as enough training data is available ($n \geq d$), the student's risk is simply 0. Otherwise, the risk can be bounded explicitly in a distribution-dependent way that, in particular, allows us to identify three key factors that explain the success of distillation, and to understand when distillation-based transfer is most effective.

4.1. What Does the Student Learn?

In this section, we derive in closed form the asymptotic solution to the gradient flow (3) undergone by the student when trained by distillation. We state the results separately for directly parameterized linear classifiers ($N = 1$) and deep linear networks ($N \geq 2$), as the settings require slightly different ways of initializing parameters. Namely, in the former case, initializing $\mathbf{w}(0) = \mathbf{0}$ is valid, while in the latter case, this would lead to vanishing gradients, and we have to initialize with small (typically random) values.

Theorem 1. *Assume the student is a directly parameterised linear classifier ($N = 1$) with weight vector initialised at zero, $\mathbf{w}(0) = \mathbf{0}$. Then, the student's weight vector fulfills almost surely*

$$\mathbf{w}(t) \to \hat{\mathbf{w}}, \tag{5}$$

for $t \to \infty$, with

$$\hat{\mathbf{w}} = \left\{ \begin{array}{cc} \mathbf{w}_*, & n \geq d, \\ \mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{w}_*, & n < d. \end{array} \right. \tag{6}$$

Theorem 1 shows a remarkable property of distillation-based training for linear systems: if sufficiently many (at least d) data points are available, the student exactly recovers the teacher's weight vector, $\mathbf{w}_*$. This is a strong justification for distillation as a method of *knowledge transfer* between linear classifiers and the theorem establishes that the effect occurs not just in the infinite data limit ($n \to \infty$), as one might have expected, but already in the finite sample regime ($n \geq d$).

When few data points are available ($n < d$), the weight vector learned by the student is simply the *projection of the teacher's weight vector onto the data span* (the subspace spanned by the columns of $\mathbf{X}$). In a sense, this is the best the student can do: the gradient descent update direction $\frac{\partial \mathbf{w}(\tau)}{\partial \tau}$ always lies in the data span, so there is no way for the student to learn anything outside of it. The projection is the best subspace-constrained approximation of $\mathbf{w}_*$ with respect to the Euclidean norm. The extent to which Euclidean closeness implies closeness in predictions is a separate matter, and the subject of Section 4.2.

Proof sketch of Theorem 1. First, notice that $\hat{\mathbf{w}}$ is a global minimiser of L^1. Moreover, when $n \geq d$, it is (almost surely wrt. $\mathbf{X} \sim P_{\mathbf{x}}^n$) unique, and when $n < d$, it is (almost surely) the only one lying in the span of $\mathbf{X}$ and thus potentially reachable by gradient descent.

The proof consists of two parts. We prove that a) the gradient flow (3) drives the objective value towards the optimum, $L^1(\mathbf{w}(t)) \to 0$ as $t \to \infty$, and b) the distance between $\mathbf{w}(t)$ and the claimed asymptote $\hat{\mathbf{w}}$ is upper-bounded by the objective gap,

$$\|\mathbf{w}(t) - \hat{\mathbf{w}}\|^2 \leq cL^1(\mathbf{w}(t)) \tag{7}$$

for some constant $c > 0$ and all $t \in [0, \infty)$.

For part a), observe that L^1 is convex. For any $\tau \in [0, \infty)$, the time-derivative of $L^1(\mathbf{w}(\tau))$ is negative unless we are at a global minimum,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\tau} L^1(\mathbf{w}(\tau)) &= \nabla L^1(\mathbf{w}(\tau))^\intercal \left(\frac{\partial \mathbf{w}(\tau)}{\partial \tau} \right) \\ &= -\left\| \nabla L^1(\mathbf{w}(\tau)) \right\|^2, \end{aligned} \tag{8}$$

implying that the objective value $L^1(\mathbf{w}(\tau))$ decreases monotonically in τ. Hence, if we denote by $\mathcal{W} = \left\{ \mathbf{w} : L^1(\mathbf{w}) \leq L^1(\mathbf{0}) \right\}$ the $L^1(\mathbf{0})$-sublevel set of the objective, we know that $\mathbf{w}(\tau) \in \mathcal{W}$ for all $\tau \in [0, \infty)$. One can show that on this set, L^1 satisfies strong convexity, but only along certain directions: for some $\mu > 0$ and all $\mathbf{w}, \mathbf{v} \in \mathcal{W}$ such that $\mathbf{v} - \mathbf{w} \in \operatorname{span}(\mathbf{X})$,

$$L^1(\mathbf{v}) \geq L^1(\mathbf{w}) + \nabla L^1(\mathbf{w})^\intercal(\mathbf{v} - \mathbf{w}) + \frac{\mu}{2}\|\mathbf{v} - \mathbf{w}\|^2. \tag{9}$$

This allows us (via a technical derivation that we omit here) to relate the objective gap to the gradient norm: it can be shown that there exists $c' > 0$, such that

$$c' L^1(\mathbf{w}) \leq \frac{1}{2}\left\| \nabla L^1(\mathbf{w}) \right\|^2. \tag{10}$$

Applying the above to $\mathbf{w}(\tau)$ in (8), we are able to bound the amount of reduction in the objective in terms of the objective itself, ultimately proving linear convergence.

For part b), invoke (9) with $\mathbf{v} = \mathbf{w}(\tau)$ and $\mathbf{w} = \hat{\mathbf{w}}$; this gives $L^1(\mathbf{w}(\tau)) \geq \frac{\mu}{2}\|\mathbf{w}(\tau) - \hat{\mathbf{w}}\|^2$. □

The full proof is given in the Supplementary Material.

The next results is the analog of Theorem 1 for deep linear networks. Here, some technical conditions are needed because the parameters cannot all be initialized at 0.

Theorem 2. *Let $\hat{\mathbf{w}}$ be defined as in Theorem 1. Assume the student is a deep linear network, initialized such that for some $\epsilon > 0$,*

$$\|\mathbf{w}(0)\| < \min\left\{\|\hat{\mathbf{w}}\|, \epsilon^N\left(\epsilon^2\|\hat{\mathbf{w}}\|^{-\frac{2}{N}} + \|\hat{\mathbf{w}}\|^{2-\frac{2}{N}}\right)^{-\frac{N}{2}}\right\}, \tag{11}$$

$$L^1(\mathbf{w}(0)) < L^1(\mathbf{0}), \tag{12}$$

$$\mathbf{W}_{j+1}(0)^\intercal\mathbf{W}_{j+1}(0) = \mathbf{W}_j(0)\mathbf{W}_j(0)^\intercal \tag{13}$$

for $j = 1, \dots, N-1$. Then, for $n \geq d$, student's weight vector fulfills almost surely

$$\mathbf{w}(t) \to \hat{\mathbf{w}}, \tag{14}$$

and for $n < d$,

$$\|\mathbf{w}(t) - \hat{\mathbf{w}}\| \leq \epsilon, \tag{15}$$

for all t large enough.

The interpretation of the theorem is analogous to Theorem 1. Given enough data ($n \geq d$), the student learns to perfectly mimic the teacher. Otherwise, it learns an approximation at least ϵ-close to the projection of the teacher's weight vector onto the data span.

The conditions (11)–(13) appear for technical reasons and a closer look at them shows that they do not pose problems in practice. Condition (11) states that the network's weights should be initialised with sufficiently small values. Consequently, this assumption is easy to satisfy in practice. Condition (12) requires that the initial loss is smaller than the loss at $\mathbf{w} = \mathbf{0}$. This condition guarantees that the gradient flow does not hit the point $\mathbf{w} = \mathbf{0}$, where all gradient vanish and the optimization would stop prematurely. In practice, when the step size is finite, the condition is not needed. Nevertheless, it is also not hard to satisfy: for any near-zero initialisation, $\mathbf{w}(0) = \mathbf{w}_0$, either $\mathbf{w}_0$ or $-\mathbf{w}_0$ will satisfy (12), so at most one has to flip the sign on one of the $\mathbf{W}_i(0)$ matrices. Finally, condition (13) is called *balancedness* (Arora et al., 2018) and discussed in-depth in (Arora et al., 2019)). It simplifies the analysis of matrix products and makes it possible to explicitly analyze the evolution of $\mathbf{w}$ induced by gradient flow in the $\mathbf{W}_i$'s. Assuming near-zero initialization, the condition is automatically satisfied approximately and there is some evidence (Arora et al., 2019) suggesting that *approximate balancedness* may suffice for convergence results of the kind we are interested in. Otherwise, the condition can also simply be enforced numerically.

Proof sketch of Theorem 2. First, we establish convergence in the objective, $L^1(\mathbf{w}(t)) \to 0$ as $t \to \infty$, similarly to the case $N = 1$. Unlike that case, however, the evolution of the end-to-end weight vector $\mathbf{w}(\tau)$ is governed by complex mechanics induced by gradient flow in $\mathbf{W}_i$'s. A key tool for analyzing this induced flow was recently established in (Arora et al., 2018): the authors show that the induced flow behaves similarly to gradient flow with momentum applied directly to $\mathbf{w}$. Making use of this result, one can proceed analogously as in the case of $N = 1$ to show convergence in the objective.

Second, to show convergence in parameter space, we decompose $\mathbf{w}(t)$ into its projection onto the span of $\mathbf{X}$, and an orthogonal component. The $\mathbf{X}$-component converges to $\hat{\mathbf{w}}$, by strong convexity arguments as in the case $N = 1$. It remains to show that the orthogonal component is small. Now, recall that in the case $N = 1$, we initialise at $\mathbf{w}(0) = \mathbf{0}$ and move within the span, so the orthogonal component is always zero. When $N \geq 2$, the situation is different: a) we initialise with a potentially non-zero orthogonal component (because we need to avoid the spurious stationary point $\mathbf{w} = \mathbf{0}$), and b) the momentum term causes the orthogonal component to grow during optimisation. Luckily, the rate of growth can be precisely characterised and controlled by the initialisation norm $\|\mathbf{w}(0)\|$, so depending on how close to zero we initialise, we can upper-bound the size of the orthogonal component. This yields a bound on the distance $\|\mathbf{w}(t) - \hat{\mathbf{w}}\|$. □

For the formal proof, we refer the reader to the Supplemental Material.

4.2. How Fast Does the Student Learn?

In this section, we present our main quantitative result, a bound for the expected transfer risk in linear distillation.

We first introduce some geometric concepts. For any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$, denote by $\bar{\alpha}(\mathbf{u}, \mathbf{v}) \in [0, \pi/2]$ the unsigned angle between the vectors $\mathbf{u}$ and $\mathbf{v}$

$$\bar{\alpha}(\mathbf{u}, \mathbf{v}) = \cos^{-1}\left(\frac{|\mathbf{u}^\intercal\mathbf{v}|}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}\right). \tag{16}$$

A key quantity for us is the angle between $\mathbf{w}_*$ and a randomly chosen $\mathbf{x}$, for $\mathbf{x} \sim P_\mathbf{x}$. For a given transfer task $(P_\mathbf{x}, \mathbf{w}_*)$, we denote by p the reverse cdf of $\bar{\alpha}(\mathbf{w}_*, \mathbf{x})$,

$$p(\theta) = \mathop{\mathbb{P}}_{\mathbf{x}\sim P_\mathbf{x}}[\bar{\alpha}(\mathbf{w}_*, \mathbf{x}) \geq \theta] \qquad \text{for} \quad \theta \in [0, \pi/2]. \tag{17}$$

By construction, $p(\theta)$ is monotonically decreasing, starting with $p(0) = 1$ and approaches 0 for $\theta \to \pi/2$. Figure 1

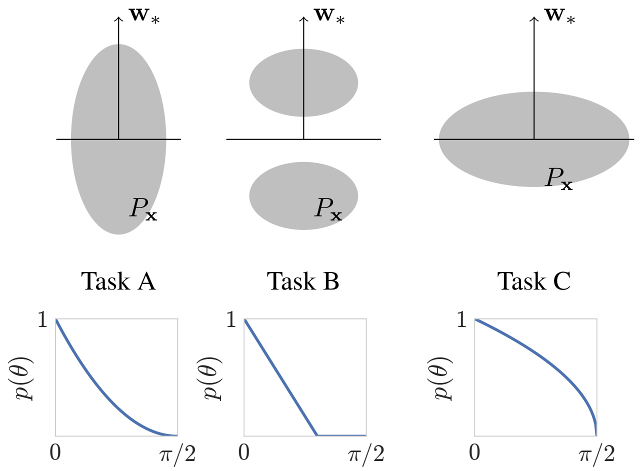

Figure 1. Schematic illustration of $p(\theta)$ for three different transfer tasks. In Task A, the angular alignment between the data and the teacher's weight vector is high, so $p(\theta)$ is fast descreasing. In Task B, it is also high, and in additional the classes are separated by a margin, so $p(\theta)$ reaches 0 before $\beta = \pi/2$. In Task C, the angular alignment is low, so $p(\theta)$ decreases rather slowly.

illustrates this behavior for three exemplary data distributions as *Tasks A,B and C*. In *Task A*, the probability mass is well aligned with the direction of the teacher's weight vector. The probability that a randomly chosen data point $x \sim P_{\mathbf{x}}$ has a large angle with $\mathbf{w}_*$ is small. Therefore, the value of $p(\theta)$ quickly drops with growing angle θ. In *Task B*, the data also aligns well with $\mathbf{w}_*$, but in addition, the data region remains bounded away from the decision boundary. Therefore, certain large angles can never occur, i.e. there exists a value $\theta_0 < \pi/2$, such that $p(\theta) = 0$ for $\theta \geq \theta_0$. In *Task C*, the situation is different: the data distribution is concentrated along the decision boundary and the probability of a angle between $\mathbf{w}_*$ and a randomly chosen data point $x \sim P_{\mathbf{x}}$ is large. As a consequence, $p(\theta)$ drops more slowly with growing angle than in the previous two settings.

We are now ready to state the main result. For improved readability, we phrase it for a student with infinitesimally small initialization, i.e. $\epsilon \to 0$. The general formulation can be found in the supplemental material.

Theorem 3 (Transfer risk bound for linear distillation)**.** *For any training set $\mathbf{X} \in \mathbb{R}^{d \times n}$, let $\hat{h}_{\mathbf{X}}(\mathbf{x}) = \mathbb{1}\{\hat{\mathbf{w}}^\intercal \mathbf{x} \geq 0\}$ be the linear classifier learned by distillation from a teacher with weight vector $\mathbf{w}_*$. Then, when $n \geq d$, it holds that*

$$\mathop{\mathbb{E}}_{\mathbf{X} \sim P_{\mathbf{x}}^{\otimes n}} \left[R\big(\hat{h}_{\mathbf{X}}\big) \right] = 0. \qquad (18)$$

For $n < d$, it holds for any $\beta \in [0, \pi/2]$ that

$$\mathop{\mathbb{E}}_{\mathbf{X} \sim P_{\mathbf{x}}^{\otimes n}} \left[R\big(\hat{h}_{\mathbf{X}}\big) \right] \leq p(\beta) + p(\pi/2 - \beta)^n \qquad (19)$$

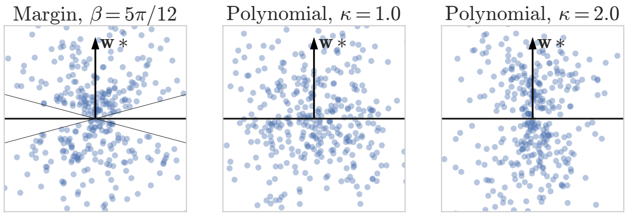

Figure 2. Examples of 2D distributions that fulfill the large-margin condition (left) and the polynomial condition with different parameters (center, right).

Equation (18) is unsurprising, of course, because in Section 4.1 we already established that for $n \geq d$ the student is able to perfectly mimic the teacher.

Inequality (19), however, is –to our knowledge– the first quantitative characterization how well a student can learn via distillation.

Before we provide the proof sketch, we present two instantiations of the bound for specific classes of tasks that provide insight how fast the right hand side of (19) actually decreases.

The margin case. The first class of tasks we consider are tasks in which the classes are separated by an *angular margin*, illustrated in Figure 2 (left). These tasks are characterized by a 'wedge' of zero probability mass near the boundary[2]. For these tasks, we obtain from Theorem 3 that the expected risk decays exponentially in n, up to $n = d - 1$.

Corollary 1 (Transfer risk of large-margin distributions)**.** *If there exists $\beta \in [0, \pi/2]$ such that $p(\beta) = 0$ and $\gamma := p(\pi/2 - \beta) < 1$, then*

$$\mathop{\mathbb{E}}_{\mathbf{X} \sim P_{\mathbf{x}}^{n}} \left[R\big(\hat{h}_{\mathbf{X}}\big) \right] \leq \gamma^n. \qquad (20)$$

The polynomial case. The second class are tasks for which we can upper-bound p by a κ-order polynomial. This can be done trivially for any task by setting $\kappa = 0.0$, but that choice would yield a vacuous bound. Higher values of κ correspond to stronger assumptions on the distribution but enable better rates. Figure 2 (center, right) shows examples of polynomial distributions for $\kappa \in \{1.0, 2.0\}$. The special case $\kappa = 1.0$ corresponds to a uniform angle distribution, while distribution with $\kappa = 2.0$ have low probability mass near the decision boundary, while not necessarily exhibiting a margin.

The following corollary establishes that for tasks with polynomial behavior of $p(\theta)$, the expected risk decays essentially at a rate of $(\log n/n)^\kappa$ or faster.

[2] In bounded domains this condition is, in particular, fulfilled in the *classical margin* situation (Schölkopf & Smola, 2002), when the classes are separated by a positive distance from each other.

Corollary 2 (Transfer risk of polynomial distributions)**.** *If there exists a* $\kappa \geq 0$ *be such that* $p(\theta) \leq c \cdot (1 - (2/\pi)\theta)^{\kappa}$ *for all* $\theta \in [0, \pi/2]$*, then*

$$\mathop{\mathbb{E}}_{\mathbf{X} \sim P_{\mathbf{x}}^n} \left[R(\hat{h}_{\mathbf{X}}) \right] \leq c \cdot \frac{1 + (\log n)^{\kappa}}{n^{\kappa}} \tag{21}$$

Proof. We apply Theorem 3 and insert the polynomial upper bound for p. For the case $n < d$, we get

$$\begin{aligned} &\mathop{\mathbb{E}}_{\mathbf{X} \sim P_{\mathbf{x}}^n} \left[R(\hat{h}_{\mathbf{X}}) \right] \\ &\leq (1 - (2/\pi)\beta)^{\kappa} + (1 - (2/\pi)(\pi/2 - \beta))^{n\kappa}. \end{aligned} \tag{22}$$

Setting $\beta = (\pi/2) \cdot n^{-1/n}$ and simplifying the resulting expressions yields

$$\leq \left(1 - e^{-\frac{\log n}{n}}\right)^{\kappa} + n^{-\kappa}. \tag{23}$$

Finally, we use the inequality $e^x \geq 1 + x$ and the claim follows. □

Note that, in contrast to many results in statistical learning theory, the bounds are far from vacuous, even when only little data is available. This can best be seen in Corollary 1, where $\gamma < 1$ and hence γ^n is an informative upper bound for the classification error. These observations suggest that distillation operates in a very different regime from classical hard-target learning. Standard bounds usually have little to say when $n < d$ and only start to be useful when $n \gg d$. In contrast, (linear) distillation ensures perfect transfer when $n \geq d$ and non-vacuous bounds are possible even when $n < d$.

4.3. Proof of Theorem 3

The case $n \geq d$ follows trivially from the result of Theorem 1 and 2. For the case $n < d$, the following property turns out to be crucial for obtaining a transfer rate of the form that we do.

Lemma 1 (Strong monotonicity)**.** *Let* $\hat{\mathbf{w}}(\mathbf{X})$ *denote the distillation solution* $\hat{\mathbf{w}}$ *as a function of the training data* $\mathbf{X}$*. Then, for any full-rank datasets* $\mathbf{X}_- \in \mathbb{R}^{d \times n_-}$ *and* $\mathbf{X}_+ \in \mathbb{R}^{d \times n_+}$ *such that* $\mathbf{X}_-$ *is contained in* $\mathbf{X}_+$*,*

$$\bar{\alpha}(\mathbf{w}_*, \hat{\mathbf{w}}(\mathbf{X}_+)) \leq \bar{\alpha}(\mathbf{w}_*, \hat{\mathbf{w}}(\mathbf{X}_-)). \tag{24}$$

Proof. If $n_+ \geq d$, then the left-hand side of (24) is zero and the claim follows. Otherwise, assume wlog that the first n_- columns of $\mathbf{X}_-$ and $\mathbf{X}_+$ coincide. Let $\mathbf{Q}_+\mathbf{R}_+ = \mathbf{X}_+$ be the QR factorisation of $\mathbf{X}_+$ with $\mathbf{Q}_+ \in \mathbb{R}^{d \times n_+}$ and $\mathbf{R}_+ \in \mathbb{R}^{n_+ \times n_+}$, and similarly for $\mathbf{X}_-$. Then $\hat{\mathbf{w}}(\mathbf{X}_+) = \mathbf{Q}_+\mathbf{Q}_+^{\intercal}\mathbf{w}_*$ and

$$\begin{aligned} \cos(\bar{\alpha}(\mathbf{w}_*, \hat{\mathbf{w}}(\mathbf{X}_+))) &= \frac{\mathbf{w}_*^{\intercal}\mathbf{Q}_+\mathbf{Q}_+^{\intercal}\mathbf{w}_*}{\|\mathbf{w}_*\| \cdot \left\|\mathbf{Q}_+\mathbf{Q}_+^{\intercal}\mathbf{w}_*\right\|} &(25)\\ &= \frac{\left\|\mathbf{Q}_+^{\intercal}\mathbf{w}_*\right\|}{\|\mathbf{w}_*\|}, &(26) \end{aligned}$$

and an analogous statement holds for $\mathbf{X}_-$. Now, because the first n_- columns of $\mathbf{Q}_+$ coincide with $\mathbf{Q}_-$, we have $\left\|\mathbf{Q}_+^{\intercal}\mathbf{w}_*\right\| \geq \left\|\mathbf{Q}_-^{\intercal}\mathbf{w}_*\right\|$ and

$$\cos(\bar{\alpha}(\mathbf{w}_*, \hat{\mathbf{w}}(\mathbf{X}_+))) \geq \cos(\bar{\alpha}(\mathbf{w}_*, \hat{\mathbf{w}}(\mathbf{X}_-))). \tag{27}$$

Taking $\cos^{-1}$ on both sides (and remembering that $\cos^{-1}$ is decreasing) yields the claim. □

For the moment, think of $\bar{\alpha}(\mathbf{w}_*, \hat{\mathbf{w}})$ as a proxy for the transfer risk, i.e. the closer the trained student $\hat{\mathbf{w}}$ is to the teacher $\mathbf{w}_*$ in terms of angles, the lower the transfer risk. A direct consequence of Lemma 1, and the reason we call it 'strong mononoticity', is that including additional data in the transfer set can never harm the transfer risk, only improve it. This property is specific to distillation; it does not hold in hard-target learning.

Proof of Theorem 3 ($n < d$)*.* For nonzero vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, we define $\alpha(\mathbf{u}, \mathbf{v}) \in [0, \pi]$ as a variant of $\bar{\alpha}$ (Equation 16) that takes the sign of $\mathbf{u}^{\intercal}\mathbf{v}$ into account,

$$\alpha(\mathbf{u}, \mathbf{v}) = \cos^{-1}\left(\frac{\mathbf{u}^{\intercal}\mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}\right). \tag{28}$$

We decompose the expected risk as follows:

$$\begin{aligned} \mathop{\mathbb{E}}_{\mathbf{X} \sim P_{\mathbf{x}}^n} \left[R(\hat{h}_{\mathbf{X}}) \right] &= \mathop{\mathbb{P}}_{\substack{\mathbf{X} \sim P_{\mathbf{x}}^n \\ \mathbf{x} \sim P_{\mathbf{x}}}} [\mathbf{w}_*^{\intercal}\mathbf{x} \cdot \hat{\mathbf{w}}^{\intercal}\mathbf{x} < 0] \\ &= \int_{\mathbf{x}: \bar{\alpha}(\mathbf{w}_*, \mathbf{x}) \geq \beta} \mathop{\mathbb{P}}_{\mathbf{X} \sim P_{\mathbf{x}}^n} [\mathbf{w}_*^{\intercal}\mathbf{x} \cdot \hat{\mathbf{w}}^{\intercal}\mathbf{x} < 0 | \mathbf{x}] \,\mathrm{d}P_{\mathbf{x}} \\ &+ \int_{\mathbf{x}: \bar{\alpha}(\mathbf{w}_*, \mathbf{x}) < \beta,\, \mathbf{w}_*^{\intercal}\mathbf{x} > 0} \mathop{\mathbb{P}}_{\mathbf{X} \sim P_{\mathbf{x}}^n} [\hat{\mathbf{w}}^{\intercal}\mathbf{x} < 0 | \mathbf{x}] \,\mathrm{d}P_{\mathbf{x}} \\ &+ \int_{\mathbf{x}: \bar{\alpha}(\mathbf{w}_*, \mathbf{x}) < \beta,\, \mathbf{w}_*^{\intercal}\mathbf{x} < 0} \mathop{\mathbb{P}}_{\mathbf{X} \sim P_{\mathbf{x}}^n} [\hat{\mathbf{w}}^{\intercal}\mathbf{x} > 0 | \mathbf{x}] \,\mathrm{d}P_{\mathbf{x}}. \end{aligned} \tag{29}$$

Let us fix some $\mathbf{x}$ for which $\bar{\alpha}(\mathbf{w}_*, \mathbf{x}) < \beta$ and $\mathbf{w}_*^{\intercal}\mathbf{x} > 0$ (i.e. an 'easy' positive test example); for this $\mathbf{x}$ we have $\alpha(\mathbf{w}_*, \mathbf{x}) = \bar{\alpha}(\mathbf{w}_*, \mathbf{x})$. Consider the situation where $\bar{\alpha}(\mathbf{w}_*, \mathbf{x}_i) < \pi/2 - \beta$ for some i (i.e. there is at least one good teaching point). Then, Lemma 1 with $\mathbf{X}_+ = \mathbf{X}$ and $\mathbf{X}_- = \mathbf{x}_i$ yields $\bar{\alpha}(\mathbf{w}_*, \hat{\mathbf{w}}) \leq \bar{\alpha}(\mathbf{w}_*, \mathbf{x}_i) < \pi/2 - \beta$. Combined with the triangle inequality, we obtain

$$\begin{aligned} \alpha(\hat{\mathbf{w}}, \mathbf{x}) &\leq \alpha(\mathbf{w}_*, \hat{\mathbf{w}}) + \alpha(\mathbf{w}_*, \mathbf{x}) &(30)\\ &\leq \bar{\alpha}(\mathbf{w}_*, \mathbf{x}_i) + \bar{\alpha}(\mathbf{w}_*, \mathbf{x}) < \pi/2, &(31) \end{aligned}$$

which implies $\hat{\mathbf{w}}^\intercal \mathbf{x} > 0$, i.e. a correct prediction (same as the teacher's). Conversely, an error can occur only if $\bar{\alpha}(\mathbf{w}_*, \mathbf{x}_i) \geq \pi/2 - \beta$ for all i. Because $\mathbf{x}_i$ are independent, we have

$$\begin{aligned}\underset{\mathbf{X}\sim P_{\mathbf{x}}^n}{\mathbb{P}}[\hat{\mathbf{w}}^\intercal \mathbf{x} < 0 | \, \mathbf{x} : \bar{\alpha}(\mathbf{w}_*, \mathbf{x}) < \beta, \, \mathbf{w}_*^\intercal \mathbf{x} > 0] \\ \leq \underset{\mathbf{X}\sim P_{\mathbf{x}}^n}{\mathbb{P}}[\forall_i : \bar{\alpha}(\mathbf{w}_*, \mathbf{x}_i) \geq \pi/2 - \beta] \quad (32) \\ = p(\pi/2 - \beta)^n.\end{aligned}$$

By a symmetric argument, one can show that

$$\begin{aligned}\underset{\mathbf{X}\sim P_{\mathbf{x}}^n}{\mathbb{P}}[\hat{\mathbf{w}}^\intercal \mathbf{x} > 0 | \, \mathbf{x} : \bar{\alpha}(\mathbf{w}_*, \mathbf{x}) < \beta, \, \mathbf{w}_*^\intercal \mathbf{x} < 0] \\ \leq p(\pi/2 - \beta)^n. \quad (33)\end{aligned}$$

Combining (29), (32) and (33) yields the result:

$$\begin{aligned}&\underset{\substack{\mathbf{X}\sim P_{\mathbf{x}}^n \\ \mathbf{x}\sim P_{\mathbf{x}}}}{\mathbb{P}}[\mathbf{w}_*^\intercal \mathbf{x} \cdot \hat{\mathbf{w}}^\intercal \mathbf{x} < 0] \leq \\ &\quad \leq \underset{\mathbf{x}}{\mathbb{P}}[\bar{\alpha}(\mathbf{w}_*, \mathbf{x}) \geq \beta] + \underset{\mathbf{x}}{\mathbb{P}}[\bar{\alpha}(\mathbf{w}_*, \mathbf{x}) < \beta] \cdot p(\pi/2 - \beta)^n \\ &\quad = p(\beta) + (1 - p(\beta)) \cdot p(\pi/2 - \beta)^n.\end{aligned}$$

□

5. Why Does Distillation Work?

From the formal analysis in the previous section, three concepts emerge as key factors for the success of distillation: *data geometry, optimization bias, and strong monotonicity*. In this section, we discuss these factors and provide some empirical confirmation how they affect or explain variations in the transfer risk.

5.1. Data Geometry

From Theorem 3 we know that the data geometry, in particular the *angular alignment* between the data distribution and the teacher, crucially impact how fast the student can learn. Formally, this is reflected in $p(\theta)$: the faster it decreases, the easier it should be for the student to learn the task.

To experimentally test the effect of data geometry on the effectiveness of distillation, we adopt the setting of Corollary 2. We consider a series of tasks of varying angular alignment, as measured by the degree, κ, of the polynomial by which $p(\theta)$ is upper bounded.

Specifically, for any κ, the task $(P_{\mathbf{x}}^\kappa, \mathbf{w}_*^\kappa)$ is defined by the following sampling procedure. First, an angle a is sampled from the κ-polynomial distribution, i.e. $\mathbb{P}\,[a \geq \theta] = (1 - (2/\pi)\theta)^\kappa$ for $\theta \in [0, \pi/2]$. Then, a direction $\mathbf{z}$ is uniformly sampled from all unit-length vectors that are at angle a with the teacher's weight vector, $\bar{\alpha}(\mathbf{w}_*, \mathbf{z}) = a$. Finally, $\mathbf{x} = \nu \mathbf{z}$ is returned for a random ν, distributed as a one-dimensional standard Gaussian.

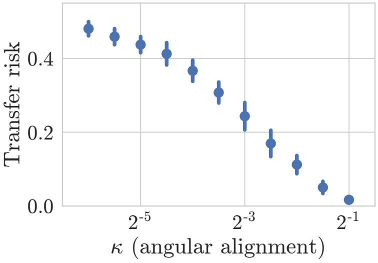

Figure 3. Transfer risk of linear distillation on tasks of varying angular alignment.

We use an input space dimension of $d = 1000$ and a transfer set size $n = 20$. Then, we train a linear student by distillation on each of the tasks and evaluate its transfer risk on held-out data. Figure 3 shows the results. The plot shows a clearly decreasing trend: on tasks with more favorable data geometry (higher κ), transfer via distillation is more effective and the student achieves lower risk.

5.2. Optimization Bias

A second key factor for the success of distillation is a specific optimization bias. For $n < d$, the distillation training objective (1) has many minima of identical function value but potentially different generalization properties. Therefore, the optimization method used could have a large impact on the transfer risk. As Theorems 1 and 2 show, gradient descent has a particularly favorable bias for distillation.

To verify this observation experimentally, we consider learners that are guided by an optimisation bias to different degrees: at one end of the spectrum is the gradient-descent learner we have studied in previous sections, while at the other end is a learner that treats all minimizers of the distillation training loss equally, i.e. that has no bias toward any of the solutions. Specifically, consider learners with weights of the form $\mathbf{w}_\delta = \hat{\mathbf{w}} + \delta \frac{\|\hat{\mathbf{w}}\|}{\|\mathbf{q}\|} \mathbf{q}$, where $\hat{\mathbf{w}}$ is the gradient-descent distillation solution and $\mathbf{q}$ is a Gaussian random vector in the subspace orthogonal to the data span, i.e. if $\mathbf{X}$ is the data matrix, then $\mathbf{X}^\intercal \mathbf{q} = \mathbf{0}$. All learners of this form globally minimize the distillation training loss, and depending on δ, they are more or less guided by the gradient-descent bias: $\delta = 0$ and $|\delta| \to \infty$ represent the two extremes mentioned above.

We train the learners $\mathbf{w}_\delta$ for $\delta \in \{0, 10, \ldots, 90\}$ on the digits 0 and 1 of the MNIST dataset, where inputs are treated as vectors in $\mathbb{R}^{784}$ and the teacher $\mathbf{w}_*$ is a logistic regression trained to classify 0s and 1s on an independent training set. We set the transfer set size to $n = 100$ and evaluate the risk on the test set.

Figure 4 shows the result. There is a clear trend in favor of learners that are more strongly guided by the gradient-

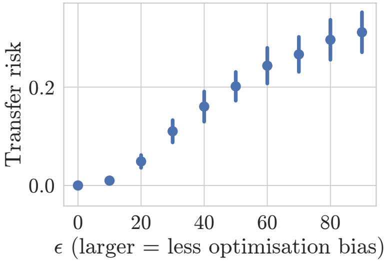

Figure 4. Transfer risk of linear distillation variants with different degrees of optimization bias, on the digits 0 and 1 of MNIST.

descent bias (small δ); these learners generally achieve lower transfer risk. This result supports the idea of optimization bias as a key component of distillation's success.

5.3. Strong Monotonicity

The third key factor we identify is *strong monotonicity*, as established in Lemma 1: training the student on more data always leads to a better approximation of the teacher's weight vector.

Compared to data geometry and optimisation bias, strong monotonicity is less amenable to experimental study because it is a downstream property that cannot directly be manipulated. We therefore take an indirect approach. We consider a set of learners including the gradient-descent distillation learner, the hard-target learner, and several learners with reduced optimisation bias (as in Section 5.2), and train them on the same task. For each learner, we note its expected risk calculated on a held-out set, and its *monotonicity index*, defined as the probability that an additional training example reduces the angle between the student's and the teacher's weight vectors rather than increasing it, i.e.

$$m(\mathbf{w}) = \mathop{\mathbb{P}}_{\substack{\mathbf{X}\sim P_{\mathbf{x}}^n \\ \mathbf{x}\sim P_{\mathbf{x}}}} [\bar{\alpha}(\mathbf{w}_*, \mathbf{w}([\mathbf{X}, \mathbf{x}])) < \bar{\alpha}(\mathbf{w}_*, \mathbf{w}(\mathbf{X}))], \tag{34}$$

where the student's weight vector $\mathbf{w}$ is now treated as a function of the training set. Thus, we can relate a learner's risk and its monotonicity.

We train the learners on the polynomial-angle task $(P_{\mathbf{x}}^{\kappa}, \mathbf{w}_*^{\kappa})$ from Section 5.1, with $\kappa = 1, d = 100$ and $n = 5$. The expected risk as well as the monotonicity index are estimated as averages over 1000 transfer sets.

The results are shown in Figure 5. There is a negative correlation between monotonicity and transfer risk, which supports the intuition of monotonicity as a desirable property and a possible explanation of distillation's success.

However, a few reservations are in order. First, as mentioned above, monotonicity cannot easily be manipulated, so its *effect* on transfer risk remains unknown. We can only measure correlation. Second, monotonicity is of binary nature; it only captures *whether* an extra data point helps or not. Yet for a quantitative characterization of risk, one would have to capture *by how much* an extra data point helps. We leave more refined definitions of monotonicity for future work.

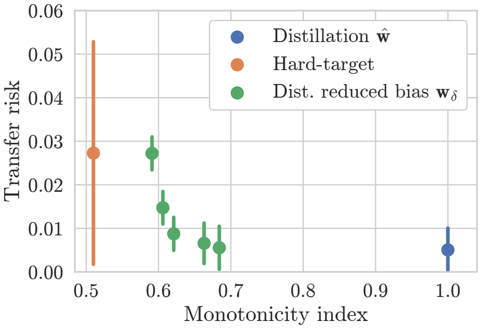

Figure 5. Expected transfer risk vs. monotonicity of different learners: gradient-descent based distillation (blue), hard-target learner (orange), and a series of distillation learners with reduced optimisation bias (green): $\mathbf{w}_\delta$ for $\delta \in \{1/16, 1/8, 1/4, 1/2, 1\}$, listed in order from left to right.

6. Conclusion

In this work, we have formulated and studied a linear model of knowledge distillation. Within this model, we have derived a) *a characterization of the solution learned by the student*, b) *a bound on the transfer risk, meaningful even in the low-data regime*, and c) *three key factors that explain the success of distillation*. In doing so, we hope to have enriched both the current intuitive and theoretical understanding of distillation, both of which have only been weakly developed.

Our work paints a picture of distillation as an extremely effective method for knowledge transfer that derives its power from an optimization bias of gradient-based methods initialized near the origin, which in particular has the effect that any additionally included training point can only improve the student's approximation of the teacher. Distillation further benefits strongly from a favorable data geometry, in particular a margin between classes.

While we have supported this picture by theory and empirical work only in the linear case, we hypothesize that similar properties also govern the behavior of distillation in the nonlinear setting. If this hypothesis turns out to be true, it would have implications for the design of transfer sets (a large teacher model being stored along with only the minimal dataset necessary for future transfer) or active learning (which samples are most informative to have labeled by the teacher). Potentially, strong monotonicity could serve as a leading design principle for new sample-efficient algorithms. We thus consider the extension to nonlinear models the main direction for future work.

References

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learing (ICML)*, 2018.

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Ba, J. and Caruana, R. Do deep nets really need to be deep? In *Conference on Neural Information Processing Systems (NIPS)*, 2014.

Bucilua, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

Celik, Z. B., Lopez-Paz, D., and McDaniel, P. Patient-driven privacy control through generalized distillation. In *IEEE Symposium on Privacy-Aware Computing (PAC)*, 2017.

Craven, M. and Shavlik, J. W. Extracting tree-structured representations of trained networks. In *Conference on Neural Information Processing Systems (NIPS)*, 1996.

Geras, K. J., Mohamed, A.-R., Caruana, R., Urban, G., Wang, S., Aslan, O., Philipose, M., Richardson, M., and Sutton, C. Blending LSTMs into CNNs. In *International Conference on Learning Representations (ICLR) Workshop*, 2016.

Hardt, M. and Ma, T. Identity matters in deep learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *Deep Learning Workshop at NIPS*, 2014.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. In *arXiv:1704.04861*, 2017.

Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. Harnessing deep neural networks with logic rules. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Kawaguchi, K. Deep learning without poor local minima. In *Conference on Neural Information Processing Systems (NIPS)*, 2016.

Li, J., Zhao, R., Huang, J.-T., and Gong, Y. Learning small-size DNN with output-distribution-based criteria. In *Conference of the International Speech Communication Association (Interspeech)*, 2014.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. Learning from noisy labels with distillation. In *International Conference on Computer Vision (ICCV)*, 2017.

Liang, P., Daumé III, H., and Klein, D. Structure compilation: trading structure for features. In *International Conference on Machine Learing (ICML)*, 2008.

Liu, J. and Zhu, X. The teaching dimension of linear learners. *Journal of Machine Learning Research (JMLR)*, 17 (1):5631–5655, 2016.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy (S&P)*, 2016.

Polino, A., Pascanu, R., and Alistarh, D. Model compression via distillation and quantization. In *International Conference on Learning Representations (ICLR)*, 2018.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. Policy distillation. In *International Conference on Learning Representations (ICLR)*, 2016.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Schölkopf, B. and Smola, A. J. *Learning With Kernels*. MIT Press, 2002.

Tang, Z., Wang, D., and Zhang, Z. Recurrent neural network training with dark knowledge transfer. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

Vapnik, V. and Izmailov, R. Learning using privileged information: similarity control and knowledge transfer. *Journal of Machine Learning Research (JMLR)*, 16(2): 2023–2049, 2015.

Yu, R., Li, A., Morariu, V. I., and Davis, L. S. Visual relationship detection with internal and external linguistic knowledge distillation. In *International Conference on Computer Vision (ICCV)*, 2017.

Zhu, J. Machine teaching for Bayesian learners in the exponential family. In *Conference on Neural Information Processing Systems (NIPS)*, 2013.

Zhu, X. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI Conference on Artificial Intelligence*, 2015.