Extracurricular Learning: Knowledge Transfer Beyond Empirical Distribution

Abstract

Knowledge distillation has been used to transfer knowledge learned by a sophisticated model (teacher) to a simpler model (student). This technique is widely used to compress model complexity. However, in most applications the compressed student model suffers from an accuracy gap with its teacher. We propose extracurricular learning, a novel knowledge distillation method, that bridges this gap by (1) modeling student and teacher uncertainties; (2) sampling training examples from underlying data distribution; and (3) matching student and teacher output distributions. We conduct extensive evaluations on regression and classification tasks and show that compared to the original knowledge distillation, extracurricular learning reduces the gap by 46% to 68%. This leads to major accuracy improvements compared to the empirical risk minimization-based training for various recent neural network architectures: 7.9% regression error reduction on the MPIIGaze dataset, +3.4% to +9.1% for top-1 image classification accuracy on the CIFAR100 dataset, and +2.9% for top-1 image classification accuracy on the ImageNet dataset.

1 Introduction

Data-driven models based on Deep Neural Networks (DNN) have shown state-of-the-art accuracies for many tasks such as computer vision [31], speech recognition [24], and reinforcement learning [51]. Training a generalizable model in supervised learning setup requires a large model capacity (to avoid underfitting) and a large labeled dataset (to avoid overfitting). In practice, both requirements cannot be perfectly satisfied: we have limited labeled data, and model size is bounded with the computational budget determined by the hardware that runs the model. Knowledge transfer/distillation and data augmentation methods have been developed to address the challenges with computational cost and data scarcity. We briefly discuss these methods, which are also the building blocks of this work.

Knowledge Distillation: Overparameterized neural networks learn better representations that lead to better generalization accuracy [1]. For example, both the PyramidNet-110 model [20] and the larger PyramidNet-200 model achieve a perfect accuracy on the training set of the CIFAR100 dataset [30], while the latter has 3% higher generalization accuracy. This motivated transferring the “knowledge” encoded in the more accurate larger model to the smaller one. Knowledge Distillation [6, 25] (KD) established an important mechanism through which one model (typically of higher capacity, called teacher) can train another model (typically a smaller model that satisfies the computational budget, called student). KD has been implemented in many machine learning tasks, for example image classification [25], object detection [10, 61], video labeling [69], natural language processing [57, 38, 54, 34, 58], and speech recognition [9, 56, 35].

The idea of KD is to encourage the student to imitate teacher’s output over various examples. For example, in classification, the teacher’s output includes not only the correct class index (the argmax of softmax generated probabilities), but also the additional information regarding the similarities to other classes (the probabilities of other classes). The amount of additional information can be
quantified by the entropy of the softmax probabilities produced by the teacher. A teacher with small training loss would produce low entropy outputs over the dataset, making knowledge transfer less effective. Some proposed remedies for this issue in previous works include matching the logits of student and teacher [6], increasing the entropy by smoothing teacher’s output [25], encouraging the student to match its intermediate feature maps to that of the teacher [48], or explicitly training a teacher with high entropy outputs [43]. Despite success of recent KD methods, in many applications, there is still a large gap between the teacher and student generalization accuracies. In this work, we propose a new KD algorithm to bridge this gap.

Data Augmentation: Lack of sufficient labeled data is another challenge in supervised learning. There are several data augmentation approaches to tackle this challenge. These methods exploit domain knowledge to transform training examples to generate more data [52, 51, 14], learn a data generation policy [13, 12, 26, 33, 70], augment the intermediate features of the model [17, 64], or find difficult examples using adversarial training [62]. Some of the recent methods [66, 60, 65, 23, 18] mix two or more data points from the empirical distribution to generate new data points. Alternatively, instead of manually designed transformations, generative models [42, 59, 29, 19] could be utilized to sample new training examples. Here, we also use samples from approximate data distribution to construct an improved KD algorithm. Note that unlike the classical data augmentation methods that need to acquire labels for the augmented data, in our KD framework we only need unlabeled samples.

As mentioned above, in general, there is a gap between generalization accuracies of student and teacher. In this work, we present a novel KD method, Extracurricular Learning (XCL), to bridge this gap. Our method is motivated by the following two arguments. First, modeling the output distribution of the teacher is important for knowledge transfer as it provides additional information for student. For regression tasks, we explicitly model the output distribution of the teacher as a Gaussian and transfer it to the student model. Note that, for classification, the output is already encoded as a categorical distribution. Second, if student exactly matches the teacher’s output on the entire input domain (e.g., all possible images), we are guaranteed to bridge this gap. This is not possible in practice since student has limited capacity and optimization is not perfect. Instead, we focus the optimization to high density regions of data distribution to match student with the teacher. Specifically, we use random convex combination of data points to generate new examples.

The main contributions of our work are:

- We show that modeling teacher and student uncertainties is important for KD, and introduce a new KD formulation for regression.
- We introduce XCL, a new KD method, that models uncertainties of the teacher, and distills teacher’s knowledge utilizing samples from an approximate data distribution in addition to the empirical distribution. XCL does not require additional unlabeled samples and does not require hyper-parameter tuning.
- XCL reduces the gap between student and teacher generalization accuracies by 46% to 68% compared to the original KD. Compared to the supervised learning baselines, XCL leads to 7.9% regression error reduction on MPIIGaze dataset, +3.4% (PyramidNet), +5.1% (ResNet), +9.1% (BinaryNet) for top-1 image classification accuracy on CIFAR100, and +2.9% (ResNet) for top-1 image classification accuracy on ImageNet.

2 Preliminaries

In supervised learning, we seek a set of parameters \( \theta \) of a parametric function \( f_{\theta} \) (e.g., weights of a neural network) to minimize the expected risk:

\[
\min_{\theta} \mathbb{E}_{(x,y) \sim p}[l(f_{\theta}(x), y)],
\]

where \( p(x, y) \) is the joint distribution of (example, label) pairs and \( l(\cdot) \) is the loss function determining how close \( f_{\theta}(x) \) and \( y \) are. For almost every practical problem, \( p \) is not available, yet a finite set of training data points \( D = \{(x_i, y_i)\}_{i=1}^{n} \) is given. The empirical risk approximation of (1) is to substitute \( p \) with empirical distribution \( p_{\text{emp}} = \frac{1}{n} \sum_{i=1}^{n} \delta(x = x_i, y = y_i) \), where \( \delta(x = x_i, y = y_i) \) is a Dirac mass function located at \((x_i, y_i)\). This leads to the Empirical Risk Minimization (ERM):

\[
\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} l(f_{\theta}(x_i), y_i)
\]
In knowledge distillation [25], the student model $f_\theta$ is encouraged to match the output of the teacher $\tau$ at training data points, i.e.:

$$\min_\theta \frac{1}{n} \sum_i l(f_\theta(x_i), \tau(x_i)) \quad (3)$$

$\tau$ in (3) can be a single more powerful model or an ensemble of several models. In the original KD [25] an average of losses in (2) and (3) is used.

KD is widely studied for the classification task, where $y_i$ is a one-hot vector that indicates the true class of $x_i$. The teacher output $\tau(x_i)$, however, is a soft label. Components of $\tau(x_i)$ encode similarity of $x_i$ to other classes [25], which encapsulates additional information compared to $y_i$. Hence, training a model with soft labels from a stronger teacher instead of one-hot labels leads to accuracy gain.

Here, we consider a different interpretation of $\tau(x_i)$ for classification that allow us to generalize KD to regression tasks as well. In classification, $\tau(x_i)$ (softmax probabilities) is a categorical distribution capturing the conditional probability of the correct class given $x_i$, while $y_i$ is only the maximum a posteriori point estimate. The full label distribution provides the student model with the uncertainty associated to the data point. For example, when objects from different classes are present in an image, or when there is ambiguity in correct target class due to occlusion. The student model is trained to minimize the average Kullback-Leibler (KL) divergence from its predicted categorical distribution to that of the teacher.

3 Knowledge Distillation for Regression with Uncertainty Estimation

In this section, we propose a new KD algorithm for regression by modeling the uncertainties of the teacher and the student. KD is often discussed in the context of classification, where the model output is a distribution that naturally captures the uncertainties. Uncertainty estimation is important for effective knowledge transfer since: (1) it provides student with not only a point estimate of teacher’s output, but also the full distribution; (2) it prevents over-penalizing student on samples that teacher is not confident. In the regression task, the model regresses the expected output signal over the empirical distribution, for example by reducing the L2 loss. It is not clear if the conventional knowledge distillation (3) would necessarily train a better student compared to the vanilla ERM (2) for a regression task. Recently, Saputra et al. [50] explored some variations of KD for regression. However, their proposed methods lack uncertainty modeling, which we show is a key property for an effective KD.

We estimate the heteroscedastic aleatoric uncertainties (uncertainties that depend on each example $x_i$) for regression tasks, similar to [41, 27]. Specifically, for a data point $x_i$ we assume the model outputs $f_\theta(x_i) = (\mu_i, \sigma_i^2)$ approximating the conditional probability $p(y|x_i)$ with a Gaussian $\mathcal{N}(\mu_i, \sigma_i^2)$. Hence, $\mu_i$’s regress $y_i$’s and $\sigma_i$’s indicate the uncertainties. We can learn $\sigma_i$’s without having access to “uncertainty labels” by minimizing the Negative Log Likelihood (NLL) loss:

$$l(f_\theta(x_i), y_i) = \frac{1}{2\sigma_i^2} ||\mu_i - y_i||_2^2 + \frac{1}{2} \log \sigma_i^2 = \frac{1}{2} \exp(-s_i) ||\mu_i - y_i||_2^2 + \frac{1}{2} s_i \quad (4)$$

In practice, for numerical stability the model predicts log variance $s_i = \log \sigma_i^2$. This can be simply implemented by adding an additional output to the last layer of a neural network. The computational overhead for uncertainty estimation is negligible.

To this point, we can train a model by ERM (2) using the NLL loss in (4) that predicts the uncertainties as well. We now use this framework and introduce a new KD algorithm for regression. We define the loss in (3) to be the KL divergence between two Gaussians $\mathcal{N}(\mu_i, \sigma_i^2)$ and $\mathcal{N}(\mu_i^*, \sigma_i^{*2})$ determined by the outputs of student and teacher, respectively:

$$l(f_\theta(x_i), \tau(x_i)) = D_{KL}(\mathcal{N}(\mu_i^*, \sigma_i^{*2}) \parallel \mathcal{N}(\mu_i, \sigma_i^2))$$

$$= \frac{1}{2} \left[ \exp(s_i^* - s_i) - \exp(-s_i) ||\mu_i^* - \mu_i||_2^2 - (s_i^* - s_i) - 1 \right] \quad (5)$$

In Section 5.2, we show KD for regression using the loss in (5) improves student accuracy significantly compared to the alternatives not accounting for uncertainties.
4 Extracurricular Learning (XCL)

In Extracurricular Learning (XCL), we also extend the knowledge transfer to data points beyond the empirical distribution. We use a pre-trained teacher model $\tau$ to annotate examples drawn from a distribution $q(x)$, which approximates the underlying marginal distribution $p(x) = \int_y p(x, y)dy$. The target (compressed) model is trained over $(x, \tau(x))$ pairs where $x \sim q$:

$$\min_{\theta} E_{x \sim q}|l(f_\theta(x), \tau(x))|$$  \hfill (6)

The loss function $l(\cdot)$ in (6) is the KL divergence from the label conditional distribution predicted by student to that of teacher. In classification, $l(\cdot)$ is the KL divergence between two categorical distributions, and in regression it is between two Gaussians as in (5).

XCL approximates the expected risk (1) more accurately compared to the ERM (2) by deploying a more accurate approximation of the data distribution $p(x, y) = p(x)p(y|x)$. If in (1) we approximate the marginal distribution $p(x)$ by a density estimator $q(x)$, and the label conditional distribution $p(y|x)$ by teacher’s output with uncertainty estimation (Gaussian in regression and categorical in classification), and define the loss function $l(f_\theta(x), y)$ to be the negative log likelihood of student observing $y$, we obtain XCL as in (6). $q(x)$ can be one or a combination of functions that approximate the marginal distribution, for example: unlabeled data, generative models [42, 59, 29, 19], data augmentations [52, 31, 14], data mixing methods [66, 60, 65, 23, 18], vicinal distribution [8], etc.

Compared to KD on the empirical distribution (3), in XCL we match student and teacher on much more data points. Specifically, when $q(x)$ is a good data density estimator, we encourage the student to imitate teacher’s output on high density regions, which helps transferring knowledge of the teacher to the student. The additional data points sampled from $q$ (the extra curriculum), compared to the original labeled dataset may be more ambiguous, yet are useful for knowledge transfer. Therefore, uncertainty estimation, i.e., approximating $p(y|x)$ instead of a single point estimate, is even more crucial in XCL.

5 Experiments

5.1 Approximating the Marginal Distribution: $q(x)$

We implemented a data mixing method as in [66] to approximate the marginal data distribution. Data mixing allows us to extend the empirical distribution by considering random convex combinations of data. We denote this by MixUp($\hat{\delta}$):

$$(x_i, y_i), (x_j, y_j) \sim \hat{\delta}, \quad \lambda \sim \text{Beta}(\alpha, \alpha) \quad \rightarrow \quad x = \lambda x_i + (1 - \lambda)x_j$$  \hfill (7)

$\hat{\delta}$ refers to the empirical distribution with standard augmentations and normalization. Note that the original MixUp paper [66], additionally, uses linear interpolation to associate a label to $x$ as: $y = \lambda y_i + (1 - \lambda)y_j$. In XCL, we only use $q(x) = \text{MixUp}(\hat{\delta})$ as an approximation for the marginal distribution and use teacher’s output for the labels. Similar to [66], we mix examples within a mini-batch. We use $\alpha = 1.0$ (uniform distribution) in (7) for mixing. Other choices of $q$ for XCL are discussed in Section 7.1.

5.2 Regression: Gaze Estimation

We evaluate XCL on a regression task, human eye-gaze estimation, that is to predict the 2D gaze orientation vector given the image of an eye. We used the MPIIGaze dataset [67, 68] that contains 45,000 annotated eye images of 15 persons. We followed the leave-one-person-out training and evaluation as in the original works [67, 68]. We used LeNet [32] model as student and PreAct-ResNet [21] model as teacher. Our training setup matches the accuracies reported in the original works [67, 68]. Note that, implementation details (e.g., learning rate schedule, batch size, etc.) of all of our experiments are provided in the Appendix.

In Table 1 we report the estimated angle error (in degrees). Each reported value is averaged over 15 persons with 3 replicas per person using different random initializations. Compared to the original KD, the proposed KD method in Section 3 with uncertainty matching achieves significant reduction (28%) in student-teacher accuracy gap. This shows the importance of matching student with teacher’s
output distribution (rather than its point estimate). In addition, when the student is trained to match teacher’s output distribution over an extended training set with XCL, we observe an even larger reduction (62%) in accuracy gap compared to the original KD. This results in 7.9% reduction in regression error compared to the ERM training. We also report the results of uncertainty estimation with NLL loss, and MixUp with linear interpolation, in isolation. In addition, we report results using Attentive Imitation Loss (AIL), a recent method that controls extent of knowledge transfer at each data point based on teacher’s error. For all models, we used the same teacher with an average angle error of 5.63 degrees. Note that, all methods shown in Table 1 are re-implemented, trained, and tested with identical setups.

In Figure 1, we show average predicted uncertainty by the teacher and two student models (the models in the last two rows in Table 1) as a function of mixing coefficient $\lambda$ defined in (7). First, we observe that models predict higher uncertainty as $x$ deviates from the empirical distribution ($\lambda$ close to 0.5). Note that, this intuitive prediction is obtained without an explicit supervision for uncertainties. In addition, when we use XCL the student model imitates teacher on more data points, and therefore has closer uncertainty estimation to the teacher on average.

5.3 Classification: CIFAR100

We evaluate performance of XCL for image classification task on the CIFAR100 dataset that contains 100 classes with 50,000 and 10,000 images in the training and test sets, respectively. For fair comparison, we reimplemented all methods and trained with identical training setups. To compute accuracies, we first compute the median over the last 10 epochs, and then average the results over 8 independent runs with different random initializations. The standard-deviation of accuracy with respect to different initializations is denoted by $\pm$ std.

**ResNet-18:** We followed the same setup as to train the ResNet-18 model, and obtained closely matching accuracies. Training over an extended distribution requires a longer training, hence, we considered $\times 2$ longer experiments as well.

To obtain an accurate teacher $\tau$, we use ensemble method. We trained a committee that consists of 8 models trained for 400 epochs using CutMix. The teacher’s output is the ensemble average of the committee members’ outputs. The ensemble model has top-1 test accuracy of 84.6%. Ensemble averaging results in a better generalization by reducing the model variance. It is a simple, yet effective way to acquire a good teacher for various applications. We explore other choices for the teacher in Section 7.

The results in Table 2 demonstrate that XCL significantly (by 67%) reduces the teacher-student accuracy gap compared to the regular KD with the same teacher (both trained for 400 epochs). This leads to major improvements over the ERM training (+5.1%) and the data mixing methods (MixUp and CutMix) that use linear interpolation for labels (+4%).

**Table 1:** Gaze angle estimation using LeNet model. Teacher angle error is 5.63. All results are reproduced.

| method               | angle error | gap  |
|----------------------|-------------|------|
| ERM                  | 6.23±0.01   |      |
| +MixUp [66]          | 6.02±0.01   | N/A  |
| ERM+uncer.           | 6.48±0.01   |      |
| KD [25]              | 5.92±0.01   | 0.29 (-) |
| KD+AIL [50]          | 5.92±0.06   | 0.29 (0%) |
| KD+uncer. (ours)     | 5.84±0.01   | 0.21 (-28%) |
| XCL (ours)           | 5.74±0.03   | 0.11 (-62%) |

**Figure 1:** Uncertainty predictions for regression task as a function of data mixing coefficient.
An alternative is to artificially increase \( \hat{H}(y) \) by applying Label Smoothing (LS) \([55]\). To match \( \hat{H} \) of XCL we apply LS with \( \varepsilon \approx 0.18 \), that is to use \( y^j = 1 - \varepsilon \) if \( j \) is the correct class, and use \( y^j = \varepsilon / (c - 1) \) otherwise. In Table 2 we see LS improves the baseline accuracy by 1%. However, LS is worse than XCL by more than 4% while having the same average entropy. We also applied smoothing by using a temperature parameter in KD \([25]\). KD with temperature and LS required exhaustive hyper-parameter tuning. We found that using temperature can improve performance of KD by 1%, which is still 2% worse than XCL without any parameter tuning. See Sections 7.2 and 7.3 for additional results.

**PyramidNet-200:** We evaluate performance of XCL on a higher capacity architecture, PyramidNet-200 \([20]\), which obtains the state-of-the-art results on CIFAR100 dataset. We used the same training setup as in \([65]\), and obtained close accuracies. The teacher is an ensemble of 8 models trained with CutMix, having a top-1 test accuracy of 87.5%. Results are shown in Table 3. Compared to the regular KD with the same teacher, XCL significantly (by 68%) reduces teacher-student accuracy gap.

**Quantized Networks:** We evaluate performance of XCL to train an extremely compressed student, a Binary-Weight \([11, 46]\) ResNet-18. This network has \( \sim 20 \times \) smaller size compared to the full-precision (32-bits) model. We use the training setup as described in \([36]\). Teacher is an ensemble of 8

---

1 As shown in the Table2, the average entropy of the teacher over the empirical distribution is 10.5%. To analyze the overfitting, we computed the same measure over the test set, which is 27.8%.
full-precision ResNet-18 models trained with CutMix, having a top-1 accuracy of 84.6%. Results are shown in Table 5. Compared to the regular KD with the same teacher, XCL significantly (by 52%) reduces teacher-student accuracy gap.

5.4 Classification: ImageNet

The ImageNet 2012 dataset [49] consists of ~1.3 million training examples and a validation set with 50,000 images from 1,000 classes. We followed the training setup as in [22] and used 300 epochs for all ImageNet experiments [65]. The model is a regular ResNet-101 architecture [21]. We use an ensemble of 4 ResNet-152-D [22] models trained with CutMix, having a top-1 validation accuracy of 83.3%. In addition to the regular validation set of the ImageNet dataset, we evaluated the performance of the models on three recently introduced test sets for ImageNet, called ImageNetV2 [47] that are collected with different sampling strategies: Threshold-0.7 (V2-A), Matched-Frequency (V2-B), and Top-Images (V2-C). Results are shown in Table 5.

Compared to the regular KD with the same teacher, XCL reduces student-teacher validation accuracy gap by 46%. Similarly, on all other test sets, XCL obtains significant improvements compared to the ERM, data mixing methods, and the regular KD. We also report the training accuracies here which, in contrast to the CIFAR100 experiments, are not perfect. To reduce under-fitting for XCL, we used 10× smaller weight decay in this experiment. Note that using a reduced weight decay did not help for other methods. We also report results for ResNet-50 training in Table 5 which shows the same trend.

6 XCL as a Nonlinear Interpolation

Samples from MixUp [66] and CutMix [65] lie on image manifolds connecting pairs of examples $(x_i, x_j)$. In MixUp, the manifold is a straight line between $x_i$ and $x_j$ characterized by (7), and in CutMix it is a discrete sequence of images constructed by replacing a block with area $(1 - \lambda)$ in $x_i$ with that of $x_j$. The original works, in addition to this mixing strategies, approximate the labels using a linear interpolation: $y = \lambda y_i + (1 - \lambda)y_j$. In this section, we discuss shortcomings of using linear interpolation to obtain labels, and analyze XCL as a non-linear interpolation method.

The majority of modern neural network architectures can obtain close to zero empirical loss using a variation of SGD optimizer. This makes the neural network models exact over the empirical distribution (e.g., ResNet-18 achieves 100% training accuracy on the CIFAR100). They can be used
We compare alternative choices of teacher in Table 8. Each teacher is an ensemble of 8 instances shown in each row. We observe that using XCL, a teacher trained with LS is both more accurate and transfers less knowledge to the student. We observe that using XCL, a teacher trained with LS is both more accurate and transfers less knowledge to the student.

In this section, we explore alternative choices of distribution approximations \( q \) and teacher \( \tau \).

### 7.1 Choices of Distribution Approximation \( q \) and Teacher \( \tau \)

In this section, we explore alternative choices of distribution approximations \( q \) and teacher models \( \tau \). We conduct these experiments on the CIFAR100 dataset using ResNet-18 model. The analyzed choices of \( q \) are: MixUp(\( \hat{p}_\delta \)), CutMix(\( \hat{p}_\delta \)), and MixUp/CutMix(\( \hat{p}_\delta \)) which is a combination of both mixing methods (each with 50\% probability), pixel-wise Gaussian, and pixel-wise Gaussian noise added to \( \hat{p}_\delta \). Table 7 shows test accuracy results. Better approximations to the data distribution, such as the data mixing methods, result in better knowledge transfer compared to uninformative distributions such as the pixel-wise Gaussian.

We compare alternative choices of teacher in Table 8. Each teacher is an ensemble of 8 instances of the given model, trained with different initializations. We observe a general trend that a more accurate teacher results in a more accurate student. [39] observed that when teacher is trained with Label Smoothing (LS), it is more accurate, but can transfer less knowledge to the student. We observe that using XCL, a teacher trained with LS is both more accurate and transfers more accurate student.
7.2 Label Smoothing

Label Smoothing (LS) [55] with a parameter $\varepsilon$ replaces ground truth labels with:

$$y_j^l = 1 - \varepsilon \quad \text{if} \quad j \text{ is the correct class} \quad \text{else} \quad \frac{\varepsilon}{c-1} \quad (8)$$

This method artificially increases label entropy. The results of ERM training with LS are reported in Table 9. All results are ResNet-18 model trained on CIFAR100 dataset for 400 epochs. Using $\varepsilon = 0.1$, LS achieves 1.5% improvement over the baseline. Note that, finding an optimal $\varepsilon$ requires extensive hyper-parameter tuning. XCL naturally obtains smooth labels, and without hyper-parameter tuning obtains significant accuracy improvement (by 3.8%) compared to the best LS.

7.3 Knowledge Distillation with Temperature Scaling

In KD [25], logits of the student and the teacher are inversely scaled by a temperature parameter $T$ before softmax probabilities are computed. This smoothing strategy can slightly improve the knowledge distillation accuracy (+1.2% compared to KD without temperature scaling). Results are reported in Table 10. Student is ResNet-18 and teacher is an ensemble of 8 ResNet-18 models trained with CutMix, having a top-1 accuracy of 84.6% on the CIFAR100 dataset.

We observe that XCL is not sensitive to temperature (Table 11). Note that finding an optimal $T$ requires extensive hyper-parameter tuning. XCL does not require hyper-parameter tuning, and compared to the best KD with temperature scaling reduces the accuracy gap by 59%.

8 Other Related Works and Conclusion

In several works, multi-stage KD was proposed to improve both teacher and student by training a sequence of models [16, 37, 3]. Our work is complementary to these approaches. There are several recent semi supervised learning methods [45, 63, 5, 53, 4, 2, 7] that produce pseudo labels for unlabeled data using a model trained on a limited labeled set. The extended dataset is then used to train the target model. These methods are similar to XCL in that they utilize unlabeled samples from data distribution to improve the model generalization. However, these methods utilize real samples, whereas XCL does not require an additional dataset, and samples from an approximate data distribution.

We introduced XCL, a framework for KD that estimate teacher and student uncertainties and match their output distributions over high density regions of the data distribution. Our method does not require additional dataset or hyper-parameter tuning, and is applicable to both regression and classification tasks. Experiments on MPII Gaze, CIFAR100, and ImageNet datasets showed that XCL achieves state-of-the-art accuracies for KD.
References

[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In Advances in neural information processing systems, pages 6155–6166, 2019.

[2] Eric Arazo, Diego Ortega, Paul Albert, Noel O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. arXiv preprint arXiv:1908.02983, 2019.

[3] Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi. Label refinery: Improving imagenet classification through label progression. arXiv preprint arXiv:1805.02641, 2018.

[4] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv preprint arXiv:1911.09785, 2019.

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In Advances in Neural Information Processing Systems, pages 5050–5060, 2019.

[6] Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 535–541, 2006.

[7] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Self-paced pseudo-labeling for semi-supervised learning. arXiv preprint arXiv:2001.06001, 2020.

[8] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In Advances in neural information processing systems, pages 416–422, 2001.

[9] Yevgen Chebotar and Austin Waters. Distilling knowledge from ensembles of neural networks for speech recognition. In Interspeech, pages 3439–3443, 2016.

[10] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Mannmohan Chandraker. Learning efficient object detection models with knowledge distillation. In Advances in Neural Information Processing Systems, pages 742–751, 2017.

[11] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In Advances in neural information processing systems, pages 3123–3131, 2015.

[12] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 113–123, 2019.

[13] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. arXiv preprint arXiv:1909.13719, 2019.

[14] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

[15] Thomas G Dietterich. Ensemble methods in machine learning. In International workshop on multiple classifier systems, pages 1–15. Springer, 2000.

[16] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. arXiv preprint arXiv:1805.04770, 2018.

[17] Xavier Gastaldi. Shake-shake regularization. arXiv preprint arXiv:1705.07485, 2017.

[18] Chengyue Gong, Tongzheng Ren, Mao Ye, and Qiang Liu. Mixup: A simple way to improve generalization of neural network training. arXiv preprint arXiv:2002.09024, 2020.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[20] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5927–5935, 2017.

[21] 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, pages 770–778, 2016.

[22] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 558–567, 2019.

[23] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781, 2019.

[24] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6):82–97, 2012.

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.
[53] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685, 2020.

[54] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. arXiv preprint arXiv:1908.09355, 2019.

[55] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[56] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. arXiv preprint arXiv:1908.09355, 2019.

[57] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[58] Ryoichi Takashima, Sheng Li, and Hisashi Kawai. An investigation of a knowledge distillation method for ctc acoustic models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5809–5813. IEEE, 2018.

[59] Ryoichi Takashima, Sheng Li, and Hisashi Kawai. An investigation of a knowledge distillation method for ctc acoustic models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5809–5813. IEEE, 2018.

[60] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[61] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[62] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[63] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[64] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[65] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[66] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[67] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[68] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[69] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.

[70] Raphael Tang, Yao Lu, Linqiang Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. arXiv preprint arXiv:1903.12136, 2019.
Appendix: Training details

Gaze estimation

We used the MPIIGaze dataset [67, 68] that contains 45,000 annotated eye images of 15 persons (3,000 images per person divided equally between left and right eyes). We followed the leave-one-person-out evaluation process as in the original works [67, 68]. For each experiment, 15 models trained using one person’s data as test set, and the rest 14 persons’ data as training set. We do the same process to train the teacher models as well (one leave-one-person-out per student model). To report the error, we average the error of all 15 models. Additionally, each experiment is repeated 3 times with different initializations and the average error is reported. We followed the implementation of original works [67, 68] for training, and used two existing architectures for student and teacher: the student is a 4 layers LeNet [32] and the teacher is a 9 layers PreAct-ResNet [21] trained with MixUp. The models output a two-dimensional vector that predicts the gaze vector. When we also estimate the uncertainty, we use an isotropic Gaussian $N(\mu, \sigma^2)$ to model the output distribution. Therefore, the network output is three-dimensional. We set weight decay to $10^{-4}$, learning rate to $10^{-4}$ for LeNet and $10^{-3}$ for ResNet that is decayed by a factor of 10 after 30 and 36 epochs. All experiments are trained using ADAM optimizer [28] for 40 epochs with 0.9 momentum and batch-size of 32.

ResNet-18 on CIFAR100

We followed the same setup as [14] to train the ResNet-18 model [21]. Weight decay is $5 \cdot 10^{-4}$, learning rate is 0.1, and is decayed by a factor of 5 after 60, 120, and 160 epochs when trained for 200 epochs, and after 120, 240, and 320 epochs when trained for 400 epochs. Training over an extended distribution requires a longer training, hence, we considered $\times 2$ longer experiments as well. For all experiments, we use the standard random cropping and horizontal flipping augmentations, and train with Nesterov [40] accelerated SGD with 0.9 momentum and batch-size of 128.

PyramidNet-200 on CIFAR100

We use the same training setup as in [65], namely, PyramidNet [20] is initialized with depth 200 and $\tilde{\alpha} = 240$, weight decay is $10^{-4}$, learning rate is 0.25 that is decayed by a factor of 10 after 150 and 225 epochs. For all experiments we use standard random cropping and horizontal flipping augmentations and train with Nesterov accelerated SGD for 300 epochs with 0.9 momentum and batch-size of 64.

BinaryNet on CIFAR100

We implemented Binary-Weight [11, 46] ResNet-18 architecture, where all weights (with the exception of the first and the last layers) are represented with 1-bit. We use the binary architecture as in [46], and training setup as in [56], namely, weight decay is zero, learning rate is $2 \cdot 10^{-4}$ that is decayed by a factor of 10 after 150 and 250 epochs. For all experiments, we use standard random cropping and horizontal flipping augmentations and train with ADAM optimizer [28] for 350 epochs with 0.9 momentum and batch-size of 128. The implementation is the same as [44].

ResNet on ImageNet

We use the training setup introduced in [22]. The weight decay is $10^{-4}$, learning rate is linearly warmed-up during the first 5 epochs from 0.1 to 0.4, and then decayed to 0 by a cosine function. For all experiments, we use SGD with Nesterov with batch-size of 1024, and apply standard data augmentations: random crop and resize to 224×224, random horizontal flipping, color jittering, and lightening during training, and resize the images to 256×256 followed by a center cropping to 224×224 during test. We use 300 epochs to train all models similar to [65]. We use regular ResNet-50 and ResNet-101 architectures [21] (not the D variant introduced in [22]). For XCL, we use a smaller weight decay $(10^{-5})$. 