Zero-Shot Knowledge Distillation in Deep Networks

Gaurav Kumar Nayak [*1] **Konda Reddy Mopuri** [*2] **Vaisakh Shaj** [*3] **R. Venkatesh Babu** [1]
Anirban Chakraborty [1]

Abstract

Knowledge distillation deals with the problem of training a smaller model (*Student*) from a high capacity source model (*Teacher*) so as to retain most of its performance. Existing approaches use either the training data or meta-data extracted from it in order to train the *Student*. However, accessing the dataset on which the *Teacher* has been trained may not always be feasible if the dataset is very large or it poses privacy or safety concerns (e.g., bio-metric or medical data). Hence, in this paper, we propose a novel data-free method to train the *Student* from the *Teacher*. Without even using any meta-data, we synthesize the *Data Impressions* from the complex *Teacher* model and utilize these as surrogates for the original training data samples to transfer its learning to *Student* via knowledge distillation. We, therefore, dub our method "Zero-Shot Knowledge Distillation" and demonstrate that our framework results in competitive generalization performance as achieved by distillation using the actual training data samples on multiple benchmark datasets.

1. Introduction

Knowledge Distillation (Hinton et al., 2015) enables to transfer the complex mapping functions learned by cumbersome models to relatively simpler models. The cumbersome model can be an ensemble of multiple large models or a single model with large capacity and strong regualrizers such as Dropout (Srivastava et al., 2014), BatchNorm (Ioffe & Szegedy, 2015), etc. Typically the complex and small models are referred to as *Teacher (T)* and *Student (S)* models respectively. Generally the *Teacher* models deliver excellent performance, but they can be huge and computationally expensive. Hence, these models can not be deployed in limited resource environments or when real-time inference is expected. On the other hand, a *Student* model has substantially less memory footprint, requires less computation, and thereby often results in a much faster inference time than that of the much larger *Teacher* model.

The latent information hidden in the confidences assigned by the *Teacher* to the incorrect categories, referred to as 'dark knowledge' is transferred to the *Student* via the distillation process. It is this knowledge that helps the *Teacher* to generalize better and transfers to the *Student* via matching their soft-labels (output of the soft-max layer) instead of the one-hot vector encoded labels. Matching the soft-labels produced by the *Teacher* is the natural way to transfer its generalization ability. For performing the knowledge distillation, one can use the training data from the target distribution or an arbitrary data. Typically, the data used to perform the distillation is called 'Transfer set'. In order to maximize the information provided per sample, we can make the soft targets to have a high entropy (non-peaky). This is generally achieved by using a high temperature at the softmax layer (Hinton et al., 2015). Also, because of non-peaky soft-labels, the training gradients computed on the loss will have less variance and enable to use higher learning rates leading to quick convergence.

The existing approaches use natural data either from the target data distribution or a different transfer set to perform the distillation. It is found by (Hinton et al., 2015) that using original training data performs relatively better. They also suggest to have an additional term in the objective for the *Student* to predict correct labels on the training data along with matching the soft-labels from the *Teacher* (as shown in eq. (1)). However, accessing the samples over which the *Teacher* had been trained may not always be feasible. Often the training datasets are too large (e.g., ImageNet (Russakovsky et al., 2015)). However, more importantly, most datasets are proprietary and not shared publicly due to privacy or confidentiality concerns. Especially while dealing with biometric data of large population, healthcare data of patients etc. Also, quite often the corporate would not prefer its proprietary data to be potentially accessed by its competitors. In summary, data is more precious than anything else

[*]Equal contribution [1]Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India [2]School of Informatics, University of Edinburgh, United Kingdom [3]University of Lincoln, United Kingdom. Correspondence to: Gaurav Kumar Nayak <gauravnayak@iisc.ac.in>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

in the era of deep learning and hence access to premium data (used in training a model) may not always be realistic.

Therefore, in this paper, we present a novel data-free framework to perform knowledge distillation. Since we do not use any data samples (either from the target dataset or a different transfer set) to perform the knowledge transfer, we name our approach "Zero-Shot Knowledge Distillation" (ZSKD). With no prior knowledge about the target data, we perform pseudo data synthesis from the *Teacher* model that act as the transfer set to perform the distillation. Our approach obtains useful prior information about the underlying data distribution in the form of *Class Similarities* from the model parameters of the *Teacher*. Further, we successfully utilize this prior in the crafting process via modelling the output space of the *Teacher* model as a Dirichlet distribution. We name the crafted samples *Data Impressions* (DI) as these are the impressions of the training data as understood by the *Teacher* model. Thus, the contributions of this work can be listed as follows:

- Unlike the existing methods that use either data samples or the extracted meta-data to perform Knowledge Distillation, we present, for the first time, the idea of Zero-Shot Knowledge Distillation (ZSKD), with no data samples and no extracted prior information.
- In order to compose a transfer set for performing distillation, we present a sample extraction mechanism via modelling the softmax space as a Dirichlet distribution and craft *Data Impressions* (DI) from the parameters of a *Teacher* model.
- We present a simple, yet powerful procedure to extract useful prior in the form of *Class Similarities* (sec. 3.2) which enables better modelling of the data distribution and is utilized in the Dirichlet sampling based DI generation framework.
- We demonstrate the effectiveness of our ZSKD approach via an empirical evaluation over multiple benchmark datasets and model architectures (sec. 4).

The rest of the paper is organized as follows: section 2 presents a brief account of existing research that are related to this work, section 3 discusses the proposed framework in detail, section 4 demonstrates the empirical evaluation, and section 5 presents a discussion on the proposed method and concludes the paper.

2. Related Works

The teacher model generally has high complexity and are not preferred for real-time embedded platforms due to its large memory and computational requirements. In practice, networks of smaller size which are compact and deployable are required. Several techniques have been proposed in the past to transfer the knowledge from the teacher to the student model without much compromise in performance. We can categorize them broadly into three types based on the amount of data used for knowledge distillation:

- **Using entire training data or similar data**: In (Bucilu et al., 2006), model compression technique is used. The target network is trained using the pseudo labels obtained from the larger model with an objective to match the pre-softmax values (called logits). In (Hinton et al., 2015), the softmax distribution of classes produced by teacher model using high temperature in its softmax (called "soft targets") are used to train the student model as the knowledge contained in incorrect class probabilities tends to capture the teacher generalization ability better in comparison to hard labels. The matching of logits is a special case of this general method. In (Furlanello et al., 2018), knowledge transfer is done across several generations where the student of current generation learns from its previous generation. The final predictions are made from the ensemble of student models using the mean of the predictions from each student.
- **Using few samples of original data**: In (Kimura et al., 2018), knowledge distillation is performed using few original samples of training data which are augmented by "pseudo training examples". These pseudo examples are obtained using inducing point (Snelson & Ghahramani, 2006) method via iterative optimization technique in an adversarial manner which makes the training procedure complicated.
- **Using meta data**: In (Lopes et al., 2017), activation records are stored at each layer after the training of teacher model and used as meta data to reconstruct training samples and utilize them to train the student model. Although, this method does consider the case of knowledge distillation in the absence of training data but meta data is formed using the training data itself. So, meta data has dependency on training samples and hence it is not a complete data-free approach.

To the best of our knowledge, we are the first to demonstrate knowledge distillation in case where no training data is available in any form. It has been shown by (Mopuri et al., 2018) that the pretrained models have memory in terms of learned parameters and can be used to extract class representative samples. Although, it was used in the context of adversarial perturbation task, we argue that carefully synthesized samples can be used as pseudo training data for knowledge distillation.

3. Proposed Method

In this section, we briefly introduce the process of Knowledge Distillation (KD) and present the proposed framework for performing Zero-Shot Knowledge Distillation in detail.

3.1. Knowledge Distillation

Transferring the generalization ability of a large, complex *Teacher* (T) deep neural network to a less complex *Student* (S) network can be achieved using the class probabilities produced by a *Teacher* as "soft targets" (Hinton et al., 2015) for training the *Student*. For this transfer, existing approaches require access to the original training data consisting of tuples of input data and targets $(x, y) \in \mathbb{D}$. Let T be the *Teacher* network with learned parameters θ_T and S be the *Student* with parameters θ_S, note that in general $|\theta_S| \ll |\theta_T|$. Knowledge distillation methods train the *Student* via minimizing the following objective (L) with respect to the parameters θ_S over the training samples $(x, y) \in \mathbb{D}$

$$L = \sum_{(x,y)\in\mathbb{D}} L_{KD}(S(x, \theta_S, \tau), T(x, \theta_T, \tau)) + \lambda L_{CE}(\hat{y}_S, y) \tag{1}$$

L_{CE} is the cross-entropy loss computed on the labels $\hat{y}_S$ predicted by the *Student* and their corresponding ground truth labels y. L_{KD} is the distillation loss (e.g. cross-entropy or mean square error) comparing the soft labels (softmax outputs) predicted by the *Student* against the soft labels predicted by the *Teacher*. $T(x, \theta_T)$ represents the softmax output of the *Teacher* and $S(x, \theta_S)$ denotes the softmax output of the *Student*. Note that, unless it is mentioned, we use a softmax temperature of 1. If we use a temperature value (τ) different from 1, we represent it as $S(x, \theta_S, \tau)$ and $T(x, \theta_T, \tau)$ for the remainder of the paper. λ is the hyper-parameter to balance the two objectives.

3.2. Modelling the Data in Softmax Space

However, in this work, we deal with the scenario where we have no access to (i) any training data samples (either from the target distribution or different), or (ii) meta-data extracted from it (e.g. (Lopes et al., 2017)). In order to tackle this, our approach taps the memory (learned parameters) of the *Teacher* and synthesizes pseudo samples from the underlying data distribution on which it is trained. Since these are the impressions of the training data extracted from the trained model, we name these synthesized input representations as *Data Impressions*. We argue that these can serve as representative samples from the training data distribution, which can then be used as a transfer set in order to perform the knowledge distillation to a desired *Student* model.

Thus, in order to craft the *Data Impressions*, we model the output (softmax) space of the *Teacher* model. Let $\boldsymbol{s} \sim p(\boldsymbol{s})$, be the random vector that represents the neural softmax outputs of the *Teacher*, $T(x, \theta_T)$. We model $p(\boldsymbol{s}^k)$ belonging to each class k, using a Dirichlet distribution which is a distribution over vectors whose components are in $[0, 1]$ range and their sum is 1. Thus, the distribution to represent the softmax outputs $\boldsymbol{s}^k$ of class k would be modelled as, $Dir(K, \boldsymbol{\alpha}^k)$, where $k \in \{1 \dots K\}$ is the class index, K is the dimension of the output probability vector (number of categories in the recognition problem) and $\boldsymbol{\alpha}^k$ is the concentration parameter of the distribution modelling class k. The concentration parameter $\boldsymbol{\alpha}^k$ is a K dimensional positive real vector, i.e, $\boldsymbol{\alpha}^k = [\alpha_1^k, \alpha_2^k, \dots, \alpha_K^k]$, and $\alpha_i^k > 0, \forall i$.

Concentration Parameter $(\boldsymbol{\alpha})$: Since the sample space of the Dirichlet distribution is interpreted as a discrete probability distribution (over the labels), intuitively, the concentration parameter $(\boldsymbol{\alpha})$ can be thought of as determining how "concentrated" the probability mass of a sample from a Dirichlet distribution is likely to be. With a value much less than 1, the mass will be highly concentrated in only a few components, and all the rest will have almost zero mass. On the other hand, with a value much greater than 1, the mass will be dispersed almost equally among all the components.

Obtaining prior information for the concentration parameter is not straightforward. The parameter cannot be the same for all components since this results in all sets of probabilities being equally likely, which is not a realistic scenario. For instance, in case of CIFAR-10 dataset, it would not be meaningful to have a softmax output in which the *dog* class and *plane* class have the same confidence (since they are visually dissimilar). Also, same α_i values denote the lack of any prior information to favour one component of sampled softmax vector over the other. Hence, the concentration parameters should be assigned in order to reflect the similarities across the components in the softmax vector. Since these components denote the underlying categories in the recognition problem, $\boldsymbol{\alpha}$ should reflect the *visual* similarities among them.

Thus, we resort to the *Teacher* network for extracting this information. We compute a normalized class similarity matrix (C) using the weights W connecting the final (softmax) and the pre-final layers. The element $C(i, j)$ of this matrix denotes the visual similarity between the categories i and j in $[0, 1]$. Thus, a row $\boldsymbol{c}_k$ of the class similarity matrix (C) gives the similarity of class k with each of the K categories (including itself). Each row $\boldsymbol{c}_k$ can be treated as the concentration parameter $(\boldsymbol{\alpha})$ of the Dirichlet distribution (Dir), which models the distribution of output probability vectors belonging to class k.

Class Similarity Matrix: The class similarity matrix C is calculated as follows. The final layer of a typical recognition model will be a fully connected layer with a softmax non-linearity. Each neuron in this layer corresponds to a class (k) and its activation is treated as the probability pre-

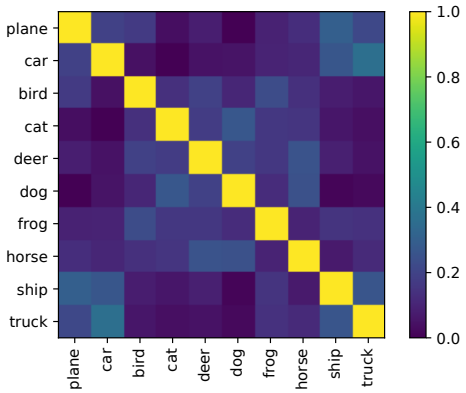

Figure 1. Class similarity matrix computed for the *Teacher* model trained over CIFAR-10 dataset. Note that the class labels are mentioned and the learned similarities are meaningful.

dicted by the model for that class. The weights connecting the previous layer to this neuron $(\boldsymbol{w}_k)$ can be considered as the template of the class k learned by the *Teacher* network. This is because the predicted class probability is proportional to the alignment of the pre-final layer's output with the template $(\boldsymbol{w}_k)$. The predicted probability peaks when the pre-final layer's output is a positive scaled version of this template $(\boldsymbol{w}_k)$. On the other hand, if the output of the pre-final layer is misaligned with the template $\boldsymbol{w}_k$, the confidence predicted for class k is reduced. Therefore, we treat the weights $\boldsymbol{w}_k$ as the class template for class k and compute the similarity between classes i and j as:

$$C(i,j) = \frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|} \tag{2}$$

Since the elements of the concentration parameter have to be positive real numbers, we further perform a min-max normalization over each row of the class similarity matrix. The visualization of the class similarity matrix calculated from a CIFAR-10 trained model is shown in Figure 1.

3.3. Crafting *Data Impressions* via Dirichlet Sampling

Once the parameters K and $\boldsymbol{\alpha}^k$ of the Dirichlet distribution are obtained for each class k, we can sample class probability (softmax) vectors, which respect the class similarities as learned by the *Teacher* network. Using the optimization procedure in eq. (3) we obtain the input representations corresponding to these sampled output class probabilities. Let $Y^k = [\boldsymbol{y}_1^k, \boldsymbol{y}_2^k, \ldots, \boldsymbol{y}_N^k] \in \mathbb{R}^{K \times N}$, be the N softmax vectors corresponding to class k, sampled from $Dir(K, \boldsymbol{\alpha}^k)$ distribution. Corresponding to each sampled softmax vector $\boldsymbol{y}_i^k$, we can craft a *Data Impression* $\bar{x}_i{}^k$, for which the *Teacher* predicts a similar softmax output. We achieve this by optimizing the objective shown in eq. (3). We initialize $\bar{x}_i^k$ as a random noisy image and update it over multiple iterations till the cross-entropy loss between the sampled softmax vector $(\boldsymbol{y}_i^k)$ and the softmax output predicted by the *Teacher* is minimized.

$$\bar{x}_i{}^k = \underset{x}{\operatorname{argmin}}\ L_{CE}(\boldsymbol{y}_i^k, T(x, \theta_T, \tau)) \tag{3}$$

where τ is the temperature used in the softmax layer. The process is repeated for each of the N sampled softmax probability vectors in Y^k, $k \in \{1 \ldots K\}$.

Scaling Factor (β): The probability density function of the Dirichlet distribution for K random variables is a $K-1$ dimensional probability simplex that exists on a K dimensional space. In addition to parameters K and $\boldsymbol{\alpha}$ as discussed in section 3.2, it is important to discuss the significance of the range of $\alpha_i \in \boldsymbol{\alpha}$, in controlling the density of the distribution. When $\alpha_i < 1,\ \forall i\ \epsilon[1, K]$, the density congregates at the edges of the simplex (Balakrishnan & Nevzorov, 2004; Lin, 2016). As their values increase (when $\alpha_i > 1, \forall i \in [1, K]$), the density becomes more concentrated on the center of the simplex (Balakrishnan & Nevzorov, 2004; Lin, 2016). Thus, we define a scaling factor $(\boldsymbol{\beta})$ which can control the range of the individual elements of the concentration parameter, which in turn decides regions in the simplex from which sampling is performed. This becomes a hyper-parameter for the algorithm. Thus the actual sampling of the probability vectors happen from $p(\boldsymbol{s}) = Dir(K, \beta \times \boldsymbol{\alpha})$. β intuitively models the spread of the Dirichlet distribution and acts as a scaling parameter atop $\boldsymbol{\alpha}$ to yield the final concentration parameter (prior). β controls the l_1-norm of the final concentration parameter which, in turn, is inversely related to the variance of the distribution. Variance of the sampled simplexes is high for smaller values of β . However very low values for β (e.g. 0.01), in conjunction with the chosen $\boldsymbol{\alpha}$, result in highly sparse softmax vectors concentrated on the extreme corners of the simplex, which is equivalent to generating class impressions (see Fig. 3). As per the ablation studies, β values of 0.1, 1.0 or a mix of these are in general favorable since they encourage higher diversity (variance) and at the same time does not result in highly sparse vectors.

3.4. Zero-Shot Knowledge Distillation

Once we craft the Data Impressions (DI) $(\bar{X})$ from the *Teacher* model, we treat them as the 'Transfer set' and perform the knowledge distillation. Note that we use only the distillation loss L_{KD} as shown in eq. (4). We ignore the cross-entropy loss from the general Distillation objective (eq. (1)) since there is only minor to no improvement in the performance and it reduces the burden of hyper-parameter λ. The proposed ZSKD approach is detailed in Algorithm 1.

$$\theta_S = \underset{\theta_S}{\operatorname{argmin}} \sum_{\bar{x} \in \bar{X}} L_{KD}(T(\bar{x}, \theta_T, \tau), S(\bar{x}, \theta_S, \tau)) \tag{4}$$

Algorithm 1 Zero-Shot Knowledge Distillation

Input : *Teacher* model T
N: number of DIs crafted per category,
$[\beta_1, \beta_2, ..., \beta_B]$: B scaling factors,
τ: Temperature for distillation
Output : Learned *Student* model $S(\theta_S)$,
$\bar{X}$: *Data Impressions*

Obtain K: number of categories from T
Compute the class similarity matrix $C = [\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_K^T]$ as in eq. (2)
$\bar{X} \leftarrow \emptyset$
for *k=1:K* **do**
  Set the concentration parameter $\boldsymbol{\alpha}^k = \mathbf{c}_k$
  for *b=1:B* **do**
    for *n=1:*$\lfloor N/B \rfloor$ **do**
      Sample $\boldsymbol{y}_n^k \sim Dir(K, \beta_b \times \boldsymbol{\alpha}^k)$
      Initialize $\bar{x}_n^k$ to random noise and craft $\bar{x}_n^k = \underset{x}{\operatorname{argmin}}\, L_{CE}(\boldsymbol{y}_n^k, T(x, \theta_T, \tau))$
      $\bar{X} \leftarrow \bar{X} \cup \bar{x}_n^k$
    end
  end
end
Transfer the *Teacher*'s knowledge to *Student* using the *DIs* via $\theta_S = \underset{\theta_S}{\operatorname{argmin}} \sum_{\bar{x} \in \bar{X}} L_{KD}(T(\bar{x}, \theta_T, \tau), S(\bar{x}, \theta_S, \tau))$

Thus we generate a diverse set of pseudo training examples that can provide with enough information to train the *Student* model via Dirichlet sampling. Some of the *Data Impressions* are presented in Figure 4 for CIFAR-10 dataset. Note that the figures show 3 *DI*s per category. Also, note that the top-2 confidences in the sampled softmax corresponding to each DI are mentioned on top. We observe that the *DI*s are visually far away from the actual data samples of the dataset. However, some of the *DI*s synthesized from peaky softmax vectors (e.g. the bird, cat, car, and deer in the first row) contain clearly visible patterns of the corresponding objects. The observation that the *DI*s being visually far away from the actual data samples is understandable, since the objective to synthesize them (eq. (3)) pays no explicit attention to visual detail.

4. Experiments

In this section, we discuss the experimental evaluation of the proposed data-free knowledge transfer framework over a set of benchmark object recognition datasets: MNIST (LeCun et al., 1998), Fashion MNIST (FMNIST) (Xiao et al., 2017), and CIFAR-10 (Krizhevsky & Hinton, 2009). As all the experiments in these three datasets are dealing with classification problems with 10 categories each, value of the parameter K in all our experiments is 10. For each dataset, we first train the *Teacher* model over the available training data using the cross-entropy loss. Then we extract a set of *Data Impressions* (DI) from it via modelling its softmax output space as explained in sections 3.2 and 3.3. Finally, we choose a (light weight) *Student* model and train over the transfer set (DI) using eq. (4).

Table 1. Performance of the proposed ZSKD framework on the MNIST dataset.

Model	Performance
Teacher-CE	99.34
Student-CE	98.92
Student-KD (Hinton et al., 2015) 60K original data	99.25
(Kimura et al., 2018) 200 original data	86.70
(Lopes et al., 2017) (uses meta data)	92.47
ZSKD (Ours) (24000 *DI*s, and no original data)	98.77

We consider two ($B = 2$) scaling factors, $\beta_1 = 1.0$ and $\beta_2 = 0.1$ across all the datasets, i.e., for each dataset, half the *Data Impressions* are generated with β_1 and the other with β_2. However we observed that one can get a fairly decent performance with a choice of beta equal to either 0.1 or 1 (even without using the mixture of Dirichlet) across the datasets. A temperature value (τ) of 20 is used across all the datasets. We investigate (in sec. 4.4) the effect of transfer set size, i.e., the number of *Data Impressions* on the performance of the *Student* model. Also, since the proposed approach aims to achieve better generalization, it is a natural choice to augment the crafted *Data Impressions* while performing the distillation. We augment the samples using regular operations such as scaling, translation, rotation, flipping etc. which has proven useful in further boosting the model performance (Dao et al., 2018).

4.1. MNIST

The MNIST dataset has 60000 training images and 10000 test images of handwritten digits. We consider Lenet-5 for the *Teacher* model and Lenet-5-Half for *Student* model similar to (Lopes et al., 2017). The Lenet-5 Model contains 2 convolution layers and pooling which is followed by three fully connected layers. Lenet-5 is modified to make Lenet-5-Half by taking half the number of filters in each of the convolutional layers. The *Teacher* and *Student* models have 61706 and 35820 parameters respectively. Input images are resized from 28×28 to 32×32 and the pixel values are normalized to be in $[0, 1]$ before feeding into the models.

The performance of our Zero-Shot Knowledge Distillation for MNIST dataset is presented in Table 1. Note that, in order to understand the effectiveness of the proposed ZSKD, the table also shows the performance of the

Table 2. Performance of the proposed ZSKD framework on the Fashion MNIST dataset.

Model	Performance
Teacher-CE	90.84
Student-CE	89.43
Student-KD (Hinton et al., 2015) 60K original data	89.66
(Kimura et al., 2018) 200 original data	72.50
ZSKD (Ours) (48000 *DI*s, and no original data)	79.62

Teacher and *Student* models trained over actual data samples along with a comparison against existing distillation approaches. Teacher-CE denotes the classification accuracy of the *Teacher* model trained using the cross-entropy (CE) loss, Student-CE denotes the performance of the *Student* model trained with all the training samples and their ground truth labels using cross-entropy loss. Student-KD denotes the accuracy of the *Student* model trained using the actual training samples through Knowledge Distillation (KD) from *Teacher*. Note that this result can act as a vague upper bound for the data-free distillation approaches.

It is clear that the proposed Zero-Shot Knowledge Distillation (ZSKD) outperforms the existing few data (Kimura et al., 2018) and data-free counterparts (Lopes et al., 2017) by a great margin. Also, it performs close to the full data (classical) Knowledge Distillation while using only 24000 *DI*s, i.e., 40% of the the original training set size.

4.2. Fashion MNIST

In comparison to MNIST, this dataset is more challenging and contains images of fashion products. The training and testing set has 60000 and 10000 images respectively. Similar to MNIST, we consider Lenet-5 and Lenet-5-Half as *Teacher* and *Student* model respectively where each input image is resized from dimension 28×28 to 32×32.

Table 2 presents our results and compares with the existing approaches. Similar to MNIST, ZSKD outperforms the existing few data knowledge distillation approach (Kimura et al., 2018) by a large margin, and performs close to the classical knowledge distillation scenario (Hinton et al., 2015) with all the training samples.

4.3. CIFAR-10

Unlike MNIST and Fashion MNIST, this dataset contains RGB images of dimension $32 \times 32 \times 3$. The dataset contains 60000 images from 10 classes, where each class has 6000 images. Among them, 50000 images are form the training set and rest of the 10000 images compose the test set. We take AlexNet (Krizhevsky et al., 2012) as *Teacher* model which is relatively large in comparison to LeNet-5. Since the standard AlexNet model is designed to process input of dimension $227 \times 227 \times 3$, we need to resize the input image to this large dimension. To avoid that, we have modified the standard AlexNet to accept $32 \times 32 \times 3$ input images. The modified AlexNet contains 5 convolution layers with BatchNorm (Ioffe & Szegedy, 2015) regularization. Pooling is also applied on convolution layers $1, 2$, and 5. The deepest three layers are fully connected. AlexNet-Half is derived from the AlexNet by taking half of convolutional filters and half of the neurons in the fully connected layers except in the classification layer which has number of neurons equal to number of classes. The AlexNet-Half architecture is used as the *Student* model. The *Teacher* and *Student* models have 1.65×10^6 and 7.23×10^5 parameters respectively. For architectural details of the teacher and the student nets, please refer to the supplementary document.

Table 3. Performance of the proposed ZSKD framework on the CIFAR-10 dataset.

Model	Performance
Teacher-CE	83.03
Student-CE	80.04
Student-KD (Hinton et al., 2015) 50K original data	80.08
ZSKD (Ours) (40000 *DI*s, and no original data)	69.56

Table 3 presents the results on the CIFAR-10 dataset. It can be observed that the proposed ZSKD approach can achieve knowledge distillation with the *Data Impressions* that results in performance competitive to that realized using the actual data samples. Since the underlying target dataset is relatively more complex, we use a bigger transfer set containing 40000 *DI*s. However, the size of this transfer set containing *DI*s is still 20% smaller than that of the original training set size used for the classical knowledge distillation (Hinton et al., 2015).

4.4. Size of the Transfer Set

In this subsection, we investigate the effect of transfer set size on the performance of the distilled *Student* model. We perform the distillation with different number of *Data Impressions* such as $\{1\%, 5\%, 10\%, \ldots, 80\%\}$ of the training set size. Figure 2 shows the performance of the resulting *Student* model on the test set for all the datasets. For comparison, the plots present performance of the models distilled with the equal number of actual training samples from the dataset. It is observed that, as one can expect, the performance increases with size of the transfer set. Interestingly, even a small number of *Data Impressions* (e.g. 20% of the training set size) are sufficient to provide a competitive performance, though the improvement in performance gets quickly saturated. Also, note that the initial performance

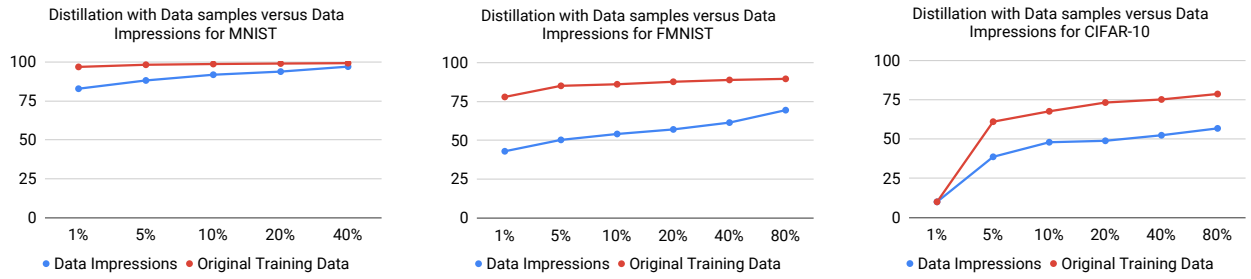

Figure 2. Performance (Test Accuracy) comparison of Data samples versus Data Impressions (without augmentation). Note that the x-axis denotes the number of *DI*s or original training samples (in %) used for performing Knowledge Distillation with respect to the size of the training data.

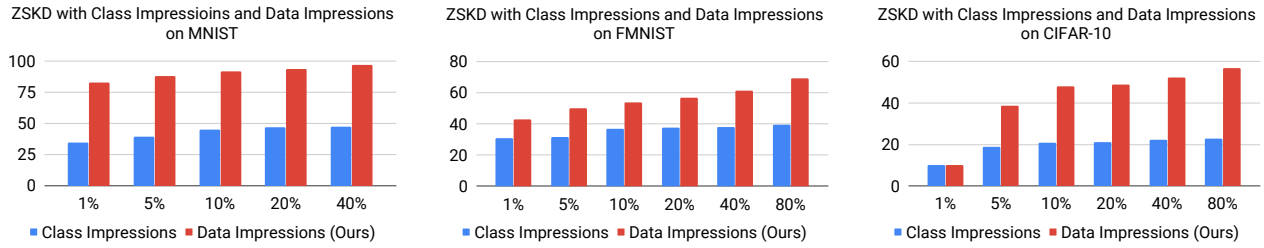

Figure 3. Performance (Test Accuracy) comparison of the ZSKD with Class Impressions (Mopuri et al., 2018) and proposed Data Impressions (without augmentation). Note that the x-axis denotes the number of *DI*s or CIs (in %) used for performing Knowledge Distillation with respect to the training data size.

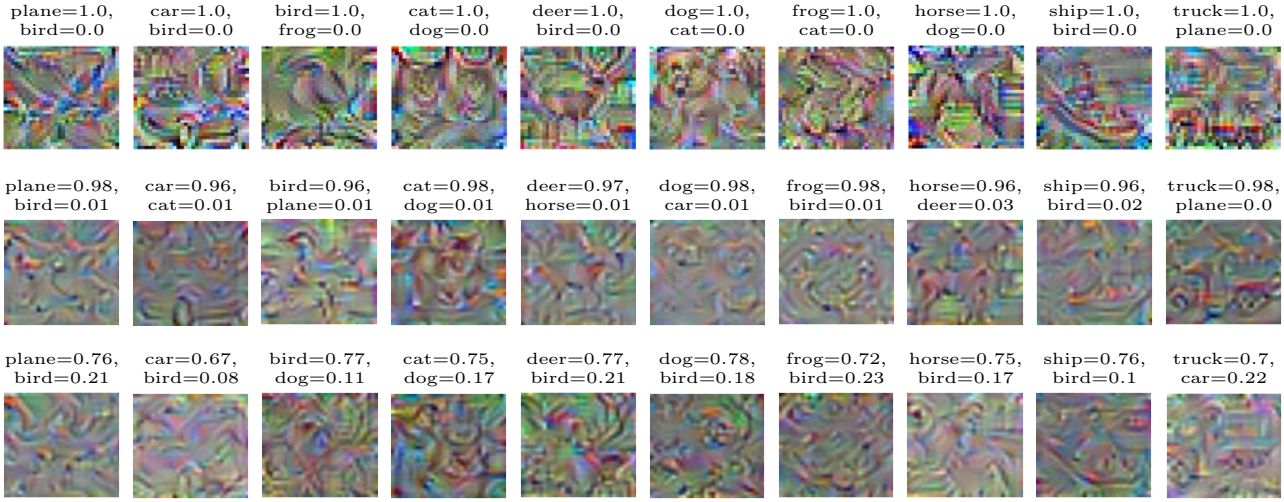

Figure 4. Visualizing the *DI*s synthesized from the *Teacher* model trained on the CIFAR-10 dataset for different choices of output softmax vectors (i.e., output class probabilities). Note that the figure shows 3 *DI*s per class in each column, each having a different spread over the labels. However, only the top-2 confidences in the sampled softmax corresponding to each DI are mentioned on top for clarity. Please note that there is no explicit objective for encouraging these pseudo samples to be visually closer to the actual training data samples, and yet some of the samples show striking patterns visually very similar to actual object shapes (e.g., bird, car, cat, dog, and deer in the first/second rows).

(with smaller transfer set) reflects the complexity of the task (dataset). For simpler datasets such as MNIST, smaller transfer sets are sufficient to achieve competitive performance. In other words, small number of *Data Impressions* can do the job of representing the patterns in the dataset. As the dataset becomes complex, more number of *Data Impressions* need to be generated to capture the underlying patterns in the dataset. Note that similar trends are observed in the distillation with the actual training samples as well.

4.5. Class Versus Data Impressions

Feature visualization works such as (Simonyan et al., 2014; Springenberg et al., 2015; Olah et al., 2017; Mordvintsev et al., 2015) attempt to understand the patterns learned by the deep neural networks in order to recognize the objects. These works reconstruct a chosen neural activation in the input space as one way to explain away the model's inference.

One of the recent works by (Mopuri et al., 2018) reconstructs samples of a given class for a downstream task of adversarial fooling. They optimize a random noise in the input space till it results in a one-hot vector (softmax) output. This means, their optimization to craft the representative samples would expect a one-hot vector in the output space. Hence, they call the reconstructions *Class Impressions*. Our reconstruction (eq. (3)) is inspired from this, though we model the output space utilizing the class similarities perceived by the *Teacher* model. Because of this, we argue that our modelling is closer to the original distribution and results in better patterns in the reconstructions, calling them *Data Impressions* of the *Teacher* model.

In this subsection, we compare these two varieties of reconstructions for the application of distillation. Figure 3 demonstrates the effectiveness of *Class* and *Data Impressions* over three datasets. It is observed that the proposed Dirichlet modelling of the output space and the reconstructed impressions consistently outperform their class counterparts by a large margin. Also, in case of *Class Impressions*, the increment in the performance due to increased transfer set size is relatively small compared to that of *Data Impressions*. Note that for better understanding, the results are shown without any data augmentation while conducting the distillation.

5. Discussion and Conclusion

Knowledge Distillation (Hinton et al., 2015) and few-shot learning hold a great deal of potential in terms of both the challenges they pose and the applications that can be realised. Data-free learning presented in recent works such as (Lopes et al., 2017; Mopuri et al., 2018) can be treated as a type of zero-shot learning, where the aim is to extract or reconstruct samples of the underlying data distribution from a trained model in order to realize a target application. It is easy to see that this line of research has significant practical implications. For instance, a deep learned model can be obtained (i) from commercial products with deployed models (e.g., mobile phone or autonomous driving vehicle), or (ii) via hacking a deployment setup. In such cases only trained model is available without training data. Also, it can help us to mitigate the absence of training data in scenarios such as medical diagnosis, where, it is often the case that patients' privacy prohibits distribution of the training data. In those cases only the trained models can be made available. Further, given (i) the cost of annotating the data, and (ii) competitive advantage leveraged with more training data, it is quite a possibility that the trained models will be made available but not the actual training data. For example, models trained by Google and Facebook might utilize proprietary data such as JFT-300M, SFC.

In this work, we presented for the first time, a complete framework called Zero-Shot Knowledge Distillation (ZSKD) to perform knowledge distillation without utilizing any data samples or meta data extracted from it. We proposed a sample extraction mechanism via modelling the data distribution in the softmax space. As a useful prior, our model utilizes class similarity information extracted from the learned model and attempts to synthesize the underlying data samples. Further, we have investigated the effectiveness of the these synthesized samples, named *Data Impressions* for a downstream task of training a substitute model via distillation.

A set of recent works that attempt to extract the training data from a learned model, drive a downstream task such as crafting adversarial perturbations or training a substitute model. However, in the current setup, the extracted samples are not influenced by the target task so as to call them task driven. Besides, it is not observed that these aforementioned extractions utilize any strong prior about the data distribution during the reconstruction. In that sense, our Dirichlet modelling of the output space that inculcates the visual similarity prior among the categories can be considered as a step towards a faithful extraction of the underlying patterns in the distribution. However, we believe that there is a lot of scope for imbibing additional and better priors particularly in the task driven scenario. For instance, utilizing multiple *Teacher* models trained on different tasks can enable better extraction of the data patterns. Also, while estimating the impressions, we can formulate objectives that can explicitly encourage diversity in the extracted samples. Some of these ideas will be considered as our future research directions.

References

Balakrishnan, N. and Nevzorov, V. B. *A primer on statistical distributions*. John Wiley & Sons, 2004.

Bucilu, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *International Conference on Knowledge discovery and Data mining*. ACM, 2006.

Dao, T., Gu, A., Ratner, A. J., Smith, V., De Sa, C., and Ré, C. A kernel theory of modern data augmentation. *arXiv preprint arXiv:1803.06084*, 2018.

Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

Kimura, A., Ghahramani, Z., Takeuchi, K., Iwata, T., and Ueda, N. Few-shot learning of neural networks from scratch by pseudo example optimization. In *British Machine Vision Conference (BMVC)*, 2018.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research, 2009.

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

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lin, J. On the dirichlet distribution. Master's thesis, Department of Mathematics and Statistics, Queens University, Kingston, Ontario, Canada, 2016.

Lopes, R. G., Fenu, S., and Starner, T. Data-free knowledge distillation for deep neural networks. In *LLD Workshop at Neural Information Processing Systems (NIPS)*, 2017.

Mopuri, K. R., Krishna, P., and Babu, R. V. Ask, acquire, and attack: Data-free uap generation using class impressions. In *European Conference on Computer Vision (ECCV)*, 2018.

Mordvintsev, A., Tyka, M., and Olah, C. Google deep dream. 2015. URL https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2017. URL https://distill.pub/2017/feature-visualization.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations (ICLR) Workshops*, 2014.

Snelson, E. and Ghahramani, Z. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

Springenberg, J., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations (ICLR) workshops*, 2015.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.