The 38th Annual AAAI Conference on Artificial Intelligence
FEBRUARY 20-27, 2024 | VANCOUVER, CANADA
VANCOUVER CONVENTION CENTRE – WEST BUILDING

Progressively Knowledge Distillation via Re-parameterizing Diffusion Reverse Process

Xufeng Yao, Fanbin Lu, Yuechen Zhang,
Xinyun Zhang, Wenqian Zhao, Bei Yu

Department of Computer Science and Engineering
The Chinese University of Hong Kong

Feb. 08, 2024

Outline

1 Background and Motivation
 1.1 Introduction
 1.2 New Findings
 1.3 Solutions

2 Method
 2.1 Problem Review
 2.2 KDiffusion
 2.3 Other Training Strategies

3 Results
 3.1 Cifar100
 3.2 ImageNet100

Background and Motivation

Introduction

Knowledge distillation (KD) is a model compression method in which a small model is trained to mimic a pre-trained, larger model (or ensemble of models)[1]

- This method was first proposed by[2] then generalized by[3]
- This training setting is sometimes referred to as "teacher-student", where the large model is the teacher and the small model is the student.
- In distillation, knowledge is transferred from the teacher model to the student by minimizing a loss function in which the target is the distribution of class probabilities predicted by the teacher model.

[1] **https://www.cse.cuhk.edu.hk/~byu/CMSC5743/2023Fall/slides/Mo5-KD.pdf**.

[2] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil (2006). “Model compression”. In: *SIGKDD*, pp. 535–541.

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean (2015). “Distilling the knowledge in a neural network”. In: *NeurIPS*.

New Findings and Pilot Experiments

However, it is observed that the distillation performance may be disrupted in the presence of significant distribution gaps.

Teacher	Swin	Swin	Swin
	94.48%	94.48%	94.48%
Student	MobileNetV2	ResNet18	ShuffleNetV2
	84.04%	84.42%	76.86%
CRD	83.72%	84.26%	77.88%
	-0.32	-0.16	+1.02

Table: Top-1 accuracies of teacher and student networks on ImageNet100.

- Table 1 shows that some conventional KD methods such as CRD yield only marginal distillation improvement in large distribution gap distillation scenario.

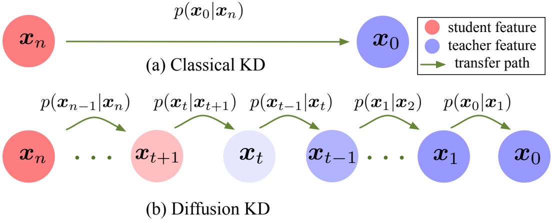
Mathematical insights

Feature-level KD mainly uses $\mathcal{L}_2$ distance as the loss function. This loss function is based on the assumption that the outputs conform to the normal distribution. This assumption may pose a significant challenge when confronting large distribution gaps.

$$\mathcal{L}_{trans} = -\log p(\boldsymbol{x}^T|\boldsymbol{x}^S) \propto \log \hat{\boldsymbol{\sigma}} + \frac{(\boldsymbol{x}^T - \hat{\boldsymbol{\mu}})^2}{2\hat{\boldsymbol{\sigma}}^2}. \tag{1}$$

- The objective is to predict the corresponding $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}$.
- In the standard $\mathcal{L}_2$ loss paradigm, variance is treated as a constant value.
- This assumption may pose a significant challenge when confronting large distribution gaps.

Motivations and Solutions



$p(\boldsymbol{x}_0|\boldsymbol{x}_n)$
$\boldsymbol{x}_n$
$\boldsymbol{x}_0$
(a) Classical KD
● student feature
● teacher feature
→ transfer path
$p(\boldsymbol{x}_{n-1}|\boldsymbol{x}_n)$
$p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$
$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$
$p(\boldsymbol{x}_1|\boldsymbol{x}_2)$
$p(\boldsymbol{x}_0|\boldsymbol{x}_1)$
$\boldsymbol{x}_n$
$\boldsymbol{x}_{t+1}$
$\boldsymbol{x}_t$
$\boldsymbol{x}_{t-1}$
$\boldsymbol{x}_1$
$\boldsymbol{x}_0$
(b) Diffusion KD

(a) Conventional feature-level distillation directly predicts teacher by student. (b) Our proposed diffusion KD decouples the objective into multiple timesteps and transfer step by step.

- To address the problem, we propose to decompose the transfer objective into small parts and optimize it progressively.
- Insipred by diffusion models, we aim to map student features to teachers features step by step.
- Directly using diffusion models is heavy, we adopt structural-reparameterization to overcome it.

Method

Problem Formulation

Generally, the objective of transfer learning is to align the teacher and student distributions. We define P and Q are corresponding distributions, then the conventional KL divergence between teacher and student distributions can be defined as :

$$\mathrm{KL}(P||Q) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}^T) \log(\frac{p(\boldsymbol{x}^T)}{q(\boldsymbol{x}^S)}), \tag{2}$$

With regard to the maximum likelihood estimation approach, the transfer objective can be defined as $-\log(q_\theta(\boldsymbol{x}^T|\boldsymbol{x}^S))$. By assuming the Markov chain for the intermediate steps between teacher and student, the transfer objective can be reformulated as:

$$-\log\left(q_\theta(\boldsymbol{x}_0^T|\boldsymbol{x}_1^T)\cdots q_\theta(\boldsymbol{x}_{t-1}^T|\boldsymbol{x}_t^T)\cdots q_\theta(\boldsymbol{x}_{n-1}^T|\boldsymbol{x}_n^S)\right). \tag{3}$$

Review Diffusion Process

Assume we have a series student features $\boldsymbol{x}_0^S, \boldsymbol{x}_1^S \cdots \boldsymbol{x}_n^S$ which are sampled independently from the standard Normal distribution. The diffusion forward process can be given by:

$$\boldsymbol{x}_t^T = \alpha_t \boldsymbol{x}_{t-1}^T + \beta_t \boldsymbol{x}_t^S = \hat{\alpha}_t \boldsymbol{x}_0^T + \hat{\beta}_t \boldsymbol{x}_0^S, \tag{4}$$

we can write down the density function of any intermediate features $\boldsymbol{x}_t^T$ by:

$$q(\boldsymbol{x}_t^T | \boldsymbol{x}_0^T) := \mathcal{N}(\boldsymbol{x}_t^T; \hat{\alpha}_t \boldsymbol{x}_0^T, \hat{\beta}_t^2 \mathbf{I}). \tag{5}$$

Assume we have a well-trained diffusion model u_θ, $\boldsymbol{x}_{t-1}^T$ can be recovered by:

$$\boldsymbol{x}_{t-1}^T = \frac{1}{\sqrt{\alpha_t}} (\boldsymbol{x}_t^T - \frac{1-\alpha_t}{\sqrt{1-\hat{\alpha}_t}} \mu_\theta(\boldsymbol{x}_t^T, t)) + \sigma_t \boldsymbol{x}_t^S. \tag{6}$$

Problems and Solutions

However, this basic design has several drawbacks.

- The conventional diffusion reverse process is time-consuming.
- Diffusion models rely on sampling multiple student features.

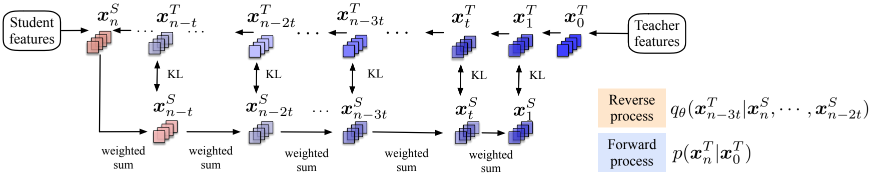
We introduce structural re-parameterization to overcome these issues. The insight is leveraging the linear properties of a set of linear modules $f_0, f_1, \cdots, f_n$ which can produce diverse outputs with a common input, i.e., $f_0(x), f_1(x), \ldots, f_n(x)$ without further inference cost:

$$\alpha_1 f_0(x) + \cdots + \alpha_n f_n(x) = (\alpha_1 f_0 + \cdots + \alpha_n f_n)(x). \tag{7}$$

By introducing structural re-parameterization techiniques, we solve problems by two sides:

- Introducing more student intermediate features.
- Combine them into inference stage without further inference cost.

KDiffusion



Proposed knowledge transfer via re-parameterizing diffusion reverse progress.

We construct both forward and reverse process like in diffusion models. The intermediate features and update rules are based on diffusion theories.

Diffusion Forward Process

We follow the same setting in[4] that assumes feature outputs follow normal distributions. In this work, given multiple student features after a batch normalization layer, we define they follow a complex normal distribution $\mathcal{N}(0, \sigma_S^2)$. We can obtain the probability distributions of each intermediate features $\boldsymbol{x}_t^T$ by:

$$q(\boldsymbol{x}_t^T | \boldsymbol{x}_0^T) := \mathcal{N}(\boldsymbol{x}_t^T; \hat{\alpha}_t \boldsymbol{x}_0^T, \hat{\beta}_t^2 \sigma_S^2). \tag{8}$$

[4] Sungsoo Ahn et al. (2019). "Variational information distillation for knowledge transfer". In: *CVPR*, pp. 9163–9171.

Diffusion Reverse Process

The diffusion reverse process can be formulated by:

$$q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_n^T, \boldsymbol{x}_0^T) = \frac{q(\boldsymbol{x}_n^T|\boldsymbol{x}_{n-t}^T)q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_0^T)}{q(\boldsymbol{x}_n^T|\boldsymbol{x}_0^T)}. \tag{9}$$

Equation (9) is also Gaussian, so the density function can be given as Equation (10).

$$\begin{aligned} q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_n^T, \boldsymbol{x}_0^T) &:= \mathcal{N}(\boldsymbol{x}_{n-t}^T; u(\boldsymbol{x}_n^T) + v(\boldsymbol{x}_0^T), w(\sigma_S^2)), \\ \text{where} \quad u(\boldsymbol{x}_n^T) &= \frac{\hat{\beta_{n-t}^2}\hat{\alpha_{n2t}}}{\hat{\beta_n^2}}\boldsymbol{x}_n^T, v(\boldsymbol{x}_0^T) = \frac{\hat{\beta_{n2t}^2}\hat{\alpha_{n-t}}}{\hat{\beta_n^2}}\boldsymbol{x}_0^T \\ w(\sigma_S^2) &= \frac{\hat{\beta_{n2t}^2}\hat{\beta_{n-t}^2}}{\hat{\beta_n^2}}\sigma_S^2, \hat{\alpha_{n2t}} = \frac{\hat{\alpha_n}}{\hat{\alpha_{n-t}}}, \hat{\beta_{n2t}^2} = 1 - \hat{\alpha_{n2t}^2}. \end{aligned} \tag{10}$$

Transfer Objective Construction

We take one intermediate step $\boldsymbol{x}_{n-3t}^{T}$ as an example. To reverse $\boldsymbol{x}_{n-3t}^{T}$, we need to predict $\boldsymbol{x}_{n-2t}^{T}$ and $\boldsymbol{x}_{0}^{T}$. The transfer objective of the intermediate step can be defined as:

$$D_{KL}(p(\boldsymbol{x}_{n-3t}^{T}|\boldsymbol{x}_{n-2t}^{T},\boldsymbol{x}_{0}^{T})||q_{\theta}(\boldsymbol{x}_{n-3t}^{T}|\boldsymbol{x}_{n-2t}^{S}\cdots\boldsymbol{x}_{n}^{S})). \tag{11}$$

By re-parameterization trick, we can eliminate the variance term, the loss can be given by:

$$\left\|(u(\boldsymbol{x}_{n-2t}^{T})+v(\boldsymbol{x}_{0}^{T}))-(u(f(\boldsymbol{x}_{n}^{S},\boldsymbol{x}_{n-t}^{S}))+v(\boldsymbol{x}_{n-2t}^{S}))\right\|^{2}, \tag{12}$$

Target guided diffusion training

Inspired by class guided diffusion[5], which offers a practical solution on conditional diffusion that considers class information (i.e., y), we can introduce y into our formulation:

$$\begin{aligned} \log p(\boldsymbol{x}_0^T|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S, y) &= \log p(\boldsymbol{x}_0^T|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S) \\ +(\log p(y|\boldsymbol{x}_0^T) &- \log p(y|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S)), \end{aligned} \tag{13}$$

Assume the weights of next teacher layer is $\boldsymbol{w}_t$, for $\boldsymbol{x}_0^T$ and predicted $\hat{\boldsymbol{x}_0^T}$, we simply use $\mathcal{L}_2$ loss, that is:

$$\mathcal{L}_{guided} = \left\|\boldsymbol{x}_0^T \boldsymbol{w}_t - \hat{\boldsymbol{x}_0^T} \boldsymbol{w}_t\right\|^2 . \tag{14}$$

[5] Prafulla Dhariwal and Alexander Nichol (2021). "Diffusion models beat gans on image synthesis". In: *Advances in Neural Information Processing Systems* 34, pp. 8780–8794.

Shuffle sampling strategy

For each training iteration, we randomly shuffle all student features such that all student features are forced to learn target features from different timesteps. The setting of uniform weights is not trivial, since we assume all student features are from the same complex normal distribution, the density function of uniformly weighted of all student features is:

$$p(\frac{1}{m}(\boldsymbol{x}_n^S + \cdots + \boldsymbol{x}_1^S)) = \mathcal{N}(0, \frac{1}{m}\sigma_S^2). \tag{15}$$

Results

Results on Cifar100

Distillation Manner	Teacher Acc	ResNet32x4 79.42	WRN40-2 75.61	VGG13 74.64	ResNet50 79.34	ResNet32x4 79.42
	Student Acc	ShuffleNetV1 70.50	ShuffleNetV1 70.50	MobileNetV2 64.6	MobileNetV2 64.6	ShuffleNetV2 71.82
Multiple Layers	AT	71.73	73.32	59.40	58.58	72.73
Multiple Layers	VID	73.38	73.61	65.56	67.57	73.40
Multiple Layers	OFD	75.98	75.85	69.48	69.04	76.82
Multiple Layers	Review	77.45	77.14	70.37	69.89	77.78
Single Layer	Avgerage	75.01	75.32	66.45	67.56	75.46
Single Layer	Kdiffusion	76.62	75.83	69.14	69.20	76.87
Multiple Layer	Kdiffusion	77.90	76.83	69.91	69.95	77.34
+ Target Guide	Kdiffusion	**78.14**	**77.26**	**70.49**	**71.14**	**77.84**

Table: Results on CIFAR-100 with the teacher and student having different architectures.

Results on ImageNet100

Distillation Manner	Teacher Acc	Swin 94.48	Swin 94.48	Swin 94.48	Swin 94.48	Swin 94.48
	Student Acc	MobileNetV2 84.04	MobileNetV3 84.98	ResNet18 84.42	ShuffleNetV1 74.74	ShuffleNetV2 76.86
Multiple Layers	AT	84.70	85.86	85.23	77.26	76.74
Multiple Layers	VID	85.42	86.46	85.12	77.56	79.46
Multiple Layers	Review	84.94	86.94	85.22	76.88	79.92
Single Layer	Kdiffusion	85.88	87.48	86.18	77.90	80.54
Multiple Layer	Kdiffusion	**86.20**	**87.88**	**86.30**	**78.04**	**80.68**

Table: Results on ImageNet-100 with the teacher and student having different architectures.

THANK YOU!