Exploring Knowledge Distillation

Aditya Gupta* **Armaan Khetarpaul*** **Shankaradithyaa Venkateswaran*** **Suhas Vundavilli***
Undergraduate B.Tech., Indian Institute of Science, Bengaluru, Karnataka-560012
{adityapg,armaank,shankaradith,suhasv}@iisc.ac.in

Abstract

Knowledge distillation, a potent technique in deep learning, transfers knowledge from a complex (teacher) to a simpler (student) model. This report explores its framework, methodologies, and applications, covering adversarial, offline, and response-based techniques. It details the distillation process, transferring knowledge from teacher to student networks. Experimental investigations on ResNet50 and ResNet18, using pre-trained models and datasets like CIFAR-10 and MNIST, evaluate their efficacy in model compression, computational efficiency, and performance preservation. Through comprehensive analysis, we assess knowledge distillation's impact on neural network architectures, offering insights into its benefits for optimizing model size and computational resources.

1 A Brief Overview of Knowledge Distillation

Knowledge distillation, vital for resource-constrained edge devices, addresses the challenge of deploying accurate models with limited resources. Larger and more accurate models demand significant computational power and storage. When faced with memory and processing constraints, the solution lies in techniques like knowledge distillation (KD).

1.1 Framework

We will use the **Response Based Knowledge** in this paper. In this, we take the output from the last layer of the teacher and try to mimic this output using a smaller model. Complementing this we focus on **Offline Distillation** . This is where we take a pre-trained teacher and distill its knowledge to the student.
We focus on the following two algorithms:

1. **Adversarial :** The teacher model is trained to obtain ground truth while the student is trained on the training set, with outputs from the teacher.
2. **Quantized :** Here we use a high precision teacher, quantized on feature maps and which transfers knowledge to a quantized student.

1.2 Loss Function

Every distillation/transfer process requires a loss function. For this report, we've implemented two loss functions. For MNIST data, we've used the softmax cross-entropy, or the log-loss with softmax activation:

$$l(y, \mathbf{f}(x)) = -f_y(x) + \log \left[\sum_{y' \in [L]} e^{f_{y'}(x)} \right]$$

For the CIFAR-10 dataset, we've used a combination of two losses: cls_loss (cross loss) and div_loss (for distillation).

$$\text{cls\_loss} = -\sum_{x \in X} a(x) \log(s(x)) \quad \text{div\_loss} = KL\_div(s, t)$$

where a is the one-shot true probability distribution of training data, s is the distribution obtained by applying softmax to the student logits, and t is the distribution obtained by applying softmax to the teacher logits

2 Interesting Findings:

Bayes' Knows the Best: We have found that the Bayes' Risk is less than the empirical risk. Empirical risk approximates the distilled class probabilities based on one label. On the other hand, the Bayes' teacher considers all alternate label realizations, weighted by their likelihood, rather than keeping them discrete. Although both Empirical and Bayes' are unbiased estimates of the true risk, we have that:

$$\underset{\text{Bayes' Variance}}{Var_{S \sim \mathbb{P}^N}[\hat{R}_*(f; S)]} \leq \underset{\text{Empirical Variance}}{Var_{S \sim \mathbb{P}^N}[\hat{R}(f; S)]} \quad (\textbf{Lemma 1})$$

Refer to Appendix A for further explanation.

Bias-Variance Bound for Distillation: We observe that the teacher predictor is imperfect. A bound can be established on the bias-variance trade-off depending on how "good" is the teacher predictor.

$$\mathbb{E}(\Delta^2) \leq \frac{1}{N} Var[p^t(x)^T l(\mathbf{f}(x))] + \mathcal{O}(||\mathbb{E}[p^t(x)] - p^*(x))||_2^2 + Var[p^t(x)]) \quad (\textbf{Lemma 2})$$

where $\Delta := \tilde{R}(f; S) - R(f)$. Refer to Appendix A for further explanation.

If the student is trained on extremely random data, will the results be meaningful? No, that may not be the case. Suppose the data has different contexts with respect to the teacher and student. In that case, misleading feedback from the div_loss term will lead to generalization failure and noise amplification from the train data, which gives us garbage results.

3 Experiments

We performed experiments in Knowledge distillation, by trying to distill Knowledge from ginormous Neural Networks to relatively smaller ones. The parameters that were measured were accuracy (correctness of the student), teaching time (time taken for the student to learn the weights from the teacher), and most importantly, the inference times (time taken by the student to classify a given data point). We've worked on two datasets: MNIST and CIFAR-10. Our primary aim was to have a student with an inference time that is 10-100 times faster than the teacher.

3.1 MNIST Dataset

It contains data on handwritten digits, of 10 classes, where each data is represented as a grayscale image of size 28×28. We used response-based knowledge, through offline distillation and the adversarial algorithm to train the student. The teacher model is a CNN with a temperature of 3.5, with two fully connected, dense, hidden layers of size 1200, trained using 1 epoch on a batch size of 32. The teacher had an accuracy of 97.41%. We'll be varying the student's structure, epochs, temperature, and batch size. The average teacher inference time is 1.339×10^{-3} seconds. Each student will have a single dense, fully connected hidden layer, of variable sizes. The results are presented in Table 1.

Table 1: Experiments on MNIST dataset

Student Layer Size	**Temperature**	**No. of Epochs**	**Batch Size**	**Accuracy**	**Training Time** (seconds)	**Inference Time Ratio** (Teacher/Student)
50	3.5	3	32	95.97%	30.998	**315.332**
300	3.5	3	16	97.24%	95.715	138.753
300	3.5	3	64	97.12%	48.421	129.564
300	3.5	2	32	96.17%	**25.108**	126.174
300	10	3	32	96.87%	64.573	123.326
300	1	3	32	96.98%	65.171	123.264
300	3.5	3	32	**97.26%**	64.854	119.127
300	3.5	5	32	97.23%	247.926	116.654
600	3.5	3	32	97.21%	103.010	38.005

From the table above, we can see that the best inference time was obtained by the simplest model, which had a layer size of 50. The accuracy of all the student models was very high since MNIST is a simple dataset. To get a better idea of the usefulness of knowledge distillation, we now try to work with a more complex dataset.

3.2 CIFAR-10

It contains data on 10 classes namely: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each data point is a 32×32 matrix, with each entry as a tuple of 3 elements to signify the RGB values of each cell. We chose ResNet50 as our teacher with the number of blocks in each layer as $[3, 4, 6, 3]$ respectively. We'll be varying the student by changing the blocks per layer, the stride of the base block, and the loss function. Here we have a feature-based knowledge, offline distillation using a quantized algorithm. The results are presented in Table 2.

Table 2: Experiments on CIFAR-10 dataset

Experiment	**Accuracy**	**Training Time** (seconds)	**Inference Time Ratio** (Teacher / Student)
Baseline	79.3%	400	6.58
Blocks = $[1, 1, 1, 1]$	79.1%	275	8.53
Blocks = $[2, 2, 0, 0]$	78.6%	370	7.81
Blocks = $[1, 1, 0, 0]$	79.4%	340	9.85
Blocks = $[1, 1, 1, 1]$, Stride = 2	77.6%	280	**11.60**
Blocks = $[1, 1, 1, 1]$, Stride = 4	**79.9%**	405	9.94
Ideal with no KD loss	75.8%	330	10.84
Ideal on reduced train set	61.2%	**60**	10.33
Ideal on reduced train set with no KD loss	56.0%	**60**	9.79

Ideal here refers to the student model with blocks $[1, 1, 1, 1]$ and stride = 2 since it had the greatest Inference Time Ratio without compromising much on accuracy. Reduced Train Set refers to the CIFAR-10 training set but with only the first 500 images of each class (instead of 5000).

For the CIFAR-10 dataset, we see a lower average accuracy across all models. The best inference time was obtained by simplifying the block structure and increasing the stride to 2. We can also compare the accuracy obtained with and without the KD loss for the ideal student on both the normal and reduced train sets. We see a significant increase in the accuracy on the reduced train set, which shows how KD is helpful when enough training data is not available.

We conclude that knowledge distillation is a useful tool to produce smaller models with faster inference times. The teacher model can be used to produce labels in the absence of labeled data, as seen in the MNIST experiment. The teacher can also be used to make up for the lack of training data, as shown in the experiment with the reduced CIFAR-10 dataset.

Acknowledgements

We thank our mentor **Dhruva Kashyap** for his constant assistance and guidance throughout this term paper. We also thank **Dr. Chiranjib Bhattacharyya** for giving us a chance to present Knowledge Distillation.

References

[1] Hinton, G. E., Vinyals, O., & Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[2] Menon, A. K., Rawat, A. S., Reddi S. J., Kim, S., & Kumar, S. A Statistical Perspective on Distillation. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[3] Gou, J., Yu, B., Maybank, S. J., & Tao, D. Knowledge Distillation: A Survey. In *International Journal of Computer Vision (2021)*, 2020.

[4] Bucila, C., Caruana, R., & Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 535–541, New York, NY, USA, 2006. ACM.

Source Code

All the source code can be found in this **GitHub repository**.

A Appendix

A.1 Proof of Lemma 1:

We observe that for population risk:

$$R(\mathbf{f}) = \mathbb{E}_x[p^*(x)^T l(f(x))]$$

where $p^*(x) = [\mathbb{P}(y|x)]_{y\in[L]}$ is the Bayes class probability distribution over the labels.
Empirical risk is given by:

$$\hat{R}(\mathbf{f}; S) = \frac{1}{N}\sum_{n\in[N]} e_{y_n}^T l(f(x_n))$$

Bayes' risk is given by:

$$\hat{R}_*(\mathbf{f}; S) = \frac{1}{N}\sum_{n\in[N]} p^*(x_n)^T l(f(x_n))$$

Claim: $Var_{S\sim\mathbb{P}^N}[\hat{R}_*(\mathbf{f}; S)] \leq Var_{S\sim\mathbb{P}^N}[\hat{R}(\mathbf{f}; S)]$

Proof.

$$\begin{aligned}
Var_{S\sim\mathbb{P}^N}[\hat{R}_*(\mathbf{f}; S)] &= \frac{1}{N} Var[\mathbb{E}_{y|x} l(y, \mathbf{f}(x))] \\
&= \frac{1}{N}\mathbb{E}_x[\mathbb{E}_{y|x}[l(y, \mathbf{f}(x))]^2] - \frac{1}{N}\mathbb{E}_x[\mathbb{E}_{y|x}[l(y, \mathbf{f}(x))]]^2 \\
&\leq \frac{1}{N}\mathbb{E}_x[\mathbb{E}_{y|x}[l(y, \mathbf{f}(x))^2]] - \frac{1}{N}\mathbb{E}_x[\mathbb{E}_{y|x}[l(y, \mathbf{f}(x))]]^2 \\
&= Var_{S\sim\mathbb{P}^N}[\hat{R}(\mathbf{f}; S)]
\end{aligned}$$

□

Here equality holds iff $(\forall x \in X)(\forall y, y' \in support(p^*(x)))\, l(y, f(x)) = l(y', f(x))$

A.2 Proof of Lemma 2:

For a given teacher predictor p^t, with corresponding distilled risk, for any predictor $\mathbf{f}$, we produce a bound on $\tilde{R}(\mathbf{f}; S) - R(\mathbf{f})$:

$$\begin{aligned}
\Delta &:= \tilde{R}(\mathbf{f}; S) - R(\mathbf{f}) \\
\mathbb{E}(\Delta^2) &= Var(\Delta) + E(\Delta)^2 \\
\mathbb{E}(\Delta) &= \mathbb{E}_x[(p^t(x) - p^*(x))^t l(\mathbf{f}(x))] \\
&\leq \mathbb{E}_x[||(p^t(x) - p^*(x))^t||_2 \cdot ||l(\mathbf{f}(x))||_2] \\
&\leq c \cdot \mathbb{E}_x[||p^t(x) - p^*(x))^t||_2]
\end{aligned}$$

Also,

$$Var(\Delta) = Var(\tilde{R}(\mathbf{f}; S)) = \frac{1}{N} Var[p^t(x)^T l(\mathbf{f}(x))]$$

Thus

$$\mathbb{E}(\Delta^2) \leq \frac{1}{N} Var[p^t(x)^T l(\mathbf{f}(x))] + \mathcal{O}(||\mathbb{E}[p^t(x)] - p^*(x))||_2^2 + Var[p^t(x)])$$