Knowledge Distillation

Armaan Khetarpaul, Aditya Gupta, Suhas Vundavilli, Shankradithyaa Venkateswaran

UG BTech 2nd year

April 12, 2024

Outline

Overview of Knowledge Distillation

- Knowledge distillation is a powerful tool whose need arises when dealing with large and computationally intense models.

Overview of Knowledge Distillation

- Knowledge distillation is a powerful tool whose need arises when dealing with large and computationally intense models.
- These large models are slow and take a lot of storage space, but are greatly accurate in what they do.

Overview of Knowledge Distillation

- Knowledge distillation is a powerful tool whose need arises when dealing with large and computationally intense models.
- These large models are slow and take a lot of storage space, but are greatly accurate in what they do.
- What do we need to do if we need a highly accurate model on an edge device with limited memory, storage space, and computational power?

Overview of Knowledge Distillation

- Knowledge distillation is a powerful tool whose need arises when dealing with large and computationally intense models.
- These large models are slow and take a lot of storage space, but are greatly accurate in what they do.
- What do we need to do if we need a highly accurate model on an edge device with limited memory, storage space, and computational power?
- Usage of new techniques such as Knowledge Distillation come into play.

Framework:

- **Response Based Knowledge:**
  - Simple and Concise knowledge.
  - Take the output from the last layer of the teacher and try to mimic this output using a smaller model.

Framework:

- **Response Based Knowledge:**
  - Simple and Concise knowledge.
  - Take the output from the last layer of the teacher and try to mimic this output using a smaller model.
- **Offline Distillation:**
  - Take a pre-trained teacher and distill its knowledge to the student.

Algorithms

- **Adversarial :** The teacher model is trained to obtain ground truth while the student is trained on the training set, with outputs from the teacher.

Algorithms

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

Loss Function

- MNIST data we used the softmax cross-entropy, or the log-loss with softmax activation:

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

Loss Function

- MNIST data we 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).

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

Loss Function

- MNIST data we 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.

Bayes' Knows Best

Lemma 1: $Var_{S\sim\mathbb{P}^N}[\hat{R}_*(f;S)] \leq Var_{S\sim\mathbb{P}^N}[\hat{R}(f;S)]$ i.e., Bayes's risk is lesser than empirical risk.

Bayes' Knows Best

Lemma 1: $Var_{S\sim\mathbb{P}^N}[\hat{R}_*(f;S)] \leq Var_{S\sim\mathbb{P}^N}[\hat{R}(f;S)]$ i.e., Bayes's risk is lesser than empirical risk.
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.

Bayes' Knows Best

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' Knows Best

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:

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

Bayes' Knows Best

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

Bayes' Knows Best

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:

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

Bayes' Knows Best

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
$(\forall x \in X)(\forall y, y' \in support(p^*(x)))\; l(y,f(x)) = l(y',f(x))$

Bias Variance Bound:

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

Bias Variance Bound:

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

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

Bias Variance Bound:

Also,

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

Thus

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

Important Results

Does the training sample S for the student, come from the training sample of teacher? Or can it be something which the teacher is seeing for the first time?

Important Results

Does the training sample S for the student, come from the training sample of teacher? Or can it be something which the teacher is seeing for the first time?
No, it doesn't have to be the same. To train on a dataset unseen by the teacher, the student model can use cls_loss of the new dataset to account for it, and the div_loss for the teacher model to learn from the teacher, and produce weights for itself.

Important Results

If the student is trained on extremely random data, will the results be meaningful?

Important Results

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.

Overview

- Distilled Knowledge from large Neural Networks to relatively smaller ones.
- Parameters that were measured:

Overview

- Distilled Knowledge from large Neural Networks to relatively smaller ones.
- Parameters that were measured:
    - Accuracy (correctness of the student).

Overview

- Distilled Knowledge from large Neural Networks to relatively smaller ones.
- Parameters that were measured:
  - Accuracy (correctness of the student).
  - Teaching time (time taken for the student to learn the weights from the teacher).

Overview

- Distilled Knowledge from large Neural Networks to relatively smaller ones.
- Parameters that were measured:
  - Accuracy (correctness of the student).
  - Teaching time (time taken for the student to learn the weights from the teacher).
  - Inference times (time taken by the student to classify a given data point).

Overview

- Distilled Knowledge from large Neural Networks to relatively smaller ones.
- Parameters that were measured:
    - Accuracy (correctness of the student).
    - Teaching time (time taken for the student to learn the weights from the teacher).
    - Inference times (time taken by the student to classify a given data point).
- Worked on two datasets: MNIST and CIFAR-10.

Overview

- Distilled Knowledge from large Neural Networks to relatively smaller ones.
- Parameters that were measured:
  - Accuracy (correctness of the student).
  - Teaching time (time taken for the student to learn the weights from the teacher).
  - Inference times (time taken by the student to classify a given data point).
- Worked on two datasets: MNIST and CIFAR-10.
- Aim was to have a student with an inference time that is 10-100 times faster than the teacher.

MNIST Dataset

- Contains data on handwritten digits, of 10 classes, where each data is represented as a grayscale image of size 28×28.

MNIST Dataset

- Used Response-Based Knowledge, through offline distillation and the Adversarial Algorithm to train the student.

MNIST Dataset

- Used Response-Based Knowledge, through offline distillation and the Adversarial Algorithm to train the student.
- Teacher model is 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.

MNIST Dataset

- Used Response-Based Knowledge, through offline distillation and the Adversarial Algorithm to train the student.
- Teacher model is 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%.

MNIST Dataset

- Used Response-Based Knowledge, through offline distillation and the Adversarial Algorithm to train the student.
- Teacher model is 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%.
- Average teacher inference time is 1.339×10^{-3}.

MNIST Dataset

- Used Response-Based Knowledge, through offline distillation and the Adversarial Algorithm to train the student.
- Teacher model is 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%.
- Average teacher inference time is 1.339×10^{-3}.
- Varying the student's structure, epochs, temperature, and batch size.

MNIST Dataset

- Used Response-Based Knowledge, through offline distillation and the Adversarial Algorithm to train the student.
- Teacher model is 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%.
- Average teacher inference time is 1.339×10^{-3}.
- Varying the student's structure, epochs, temperature, and batch size.
- Throughout, the student will be trained on 3 epochs. Each student will have a single dense, fully connected hidden layer, of variable sizes.

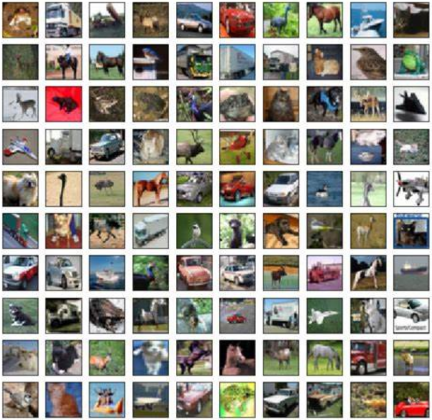
MNIST Dataset

Table: 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

CIFAR-10

- It contains data on 10 classes namely: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks



CIFAR-10

- 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.

CIFAR-10

- 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.

CIFAR-10

- 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's structure, blocks per layer, the stride of the Base Block, and loss function.

CIFAR-10

- 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's structure, blocks per layer, the stride of the Base Block, and loss function.
- Here we have a feature-based knowledge, offline distillation using a Quantized algorithm.

CIFAR-10

Table: 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

CIFAR-10 - Notation:

- *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

CIFAR-10 - Notation:

- *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).

Conclusion

- Knowledge distillation is a useful tool to produce smaller models with faster inference times.

Conclusion

- Knowledge distillation is a useful tool to produce smaller models with faster inference times.
- Teacher model can be used to produce labels in the absence of labeled data, as seen in the MNIST experiment.

Conclusion

- Knowledge distillation is a useful tool to produce smaller models with faster inference times.
- Teacher model can be used to produce labels in the absence of labeled data, as seen in the MNIST experiment.
- 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.

References

[1] 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.
[2] Gou, J., Yu, B., Maybank, S. J., & Tao, D. Knowledge Distillation: A Survey. In *International Journal of Computer Vision (2021)*, 2020.
[3] Hinton, G. E., Vinyals, O., & Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.
[4] 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.