A Basic Look into Knowledge Distillation

Group Name: Group Go Go Go
Group Member and ID:

Name	Student ID
Puquan Chen	z5405329
Song Lin	z5362555
Tianyu Pan	z5150836
Zheyuan Shao	z5334189
Haoyu Zang	z5326339

Date: 30/ 07/ 2022

Abstract

In simple terms, knowledge distillation is the process of extracting knowledge from a bloated teacher model and condensing it into a small student model. We can deploy the student model obtained from knowledge distillation to a mobile terminal with limited computing power.

In this project, we will first conduct operational experiments using the model and dataset mentioned in the paper we studied. After that, we perform the experimental study again using the same kind of model and different datasets. Finally, we do a final experiment using the same dataset as the second test and a model with a different architecture. With the three experiments, we came to some conclusions.

1. Introduction

1.1 Knowledge Distillation

Knowledge distillation is the process where we apply any kind of training to transfer the knowledge from a cumbersome model to a smaller model more suitable for deployment when the cumbersome model has been trained [1]. In other words, it's where we distil the extracts from a large, specific model (which we call the teacher model) and condense them into a smaller model (which we call the student model).

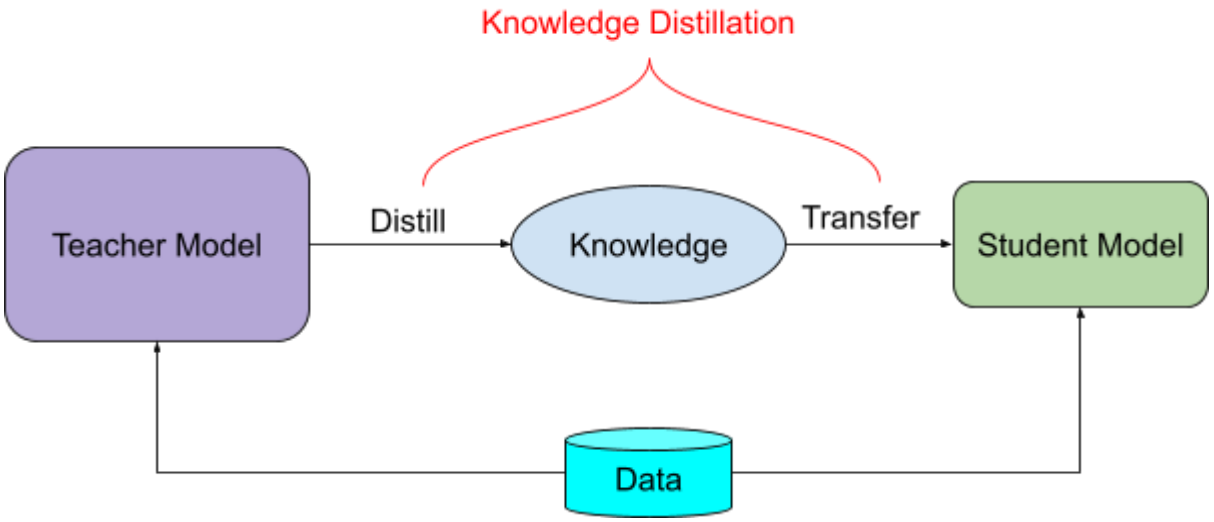


Figure 1. Knowledge Distillation Concept Map

The teacher model is often relatively large and may be an integration of many models. Although the teacher model holds a lot of knowledge and information and has a high performance, it is very unsuitable for use in some situations. The resulting student model after distillation is lightweight and simple compared to the teacher model. There are several reasons for the emergence of knowledge distillation:

a. In our current technical environment, the volume of various AI algorithms (e.g., computer vision, speech recognition, etc.) is very large, but the computing power of various terminals such as smartphones, computers, smart watches, etc. are very limited. The teacher model is often generated in a data centre by spending massive amounts of time with equally massive amounts of computing power, which is obviously not suitable for deployment on terminals with smaller computing power. Therefore, we need to transform the larger models into smaller models before deploying them to mobile devices.
b. Although large models tend to have better prediction results, they also result in longer inference time delays, which is undoubtedly unacceptable to us in many applications, such as self-driving cars.

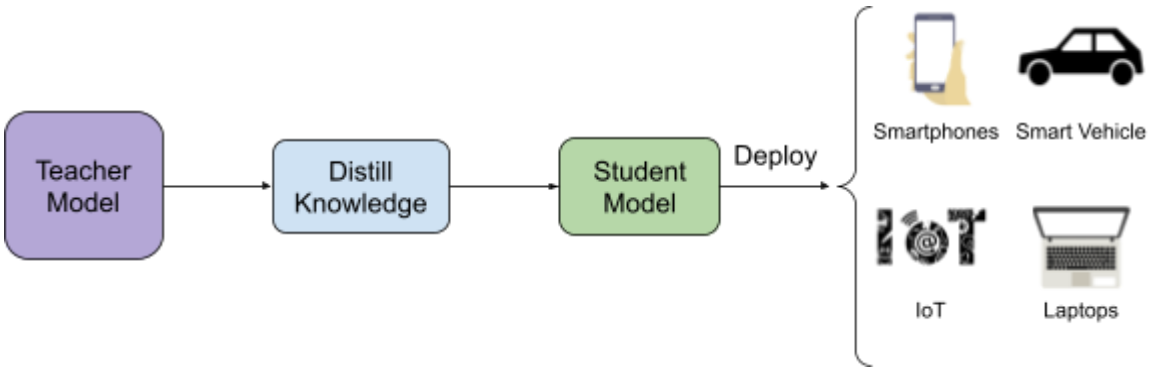


Figure 2. Application Areas of Knowledge Distillation

With the rapid development of large training pre-trained model technology, AI algorithmic models will become more and more powerful, and the parameter size of models will increase much faster than the growth of terminal computing power. As computational cost continues to increase, knowledge distillation will become a crucial research area, and hence our group has chosen this topic as the project content.

In this report, we will follow Geoffrey Hinton et al. 's paper *Distilling the Knowledge in a Neural Network*, summarise their conclusions and findings, implement their proposed algorithm on a new or simulated dataset, and try to make some improvements.

1.2 Types of Knowledge Distillation

There are three major types of knowledge distillation [2] :

1. Response-based distillation
2. Feature-based distillation
3. relation-based distillation

Response-based distillation:
Response-based knowledge distillation is to make the student model focus on the teacher's final output layer so the student model will learn the predictions from the teacher model mimicry, according to the hypothesis [1].
We can use a loss function as the distillation loss to capture the differences between these two models to make the student model more accurate in making predictions than the teacher model.

In this project, we focus our main distillation types on Response-based Distillation, because *Distilling the Knowledge in a Neural Network* uses this method and our project is based on this essay.

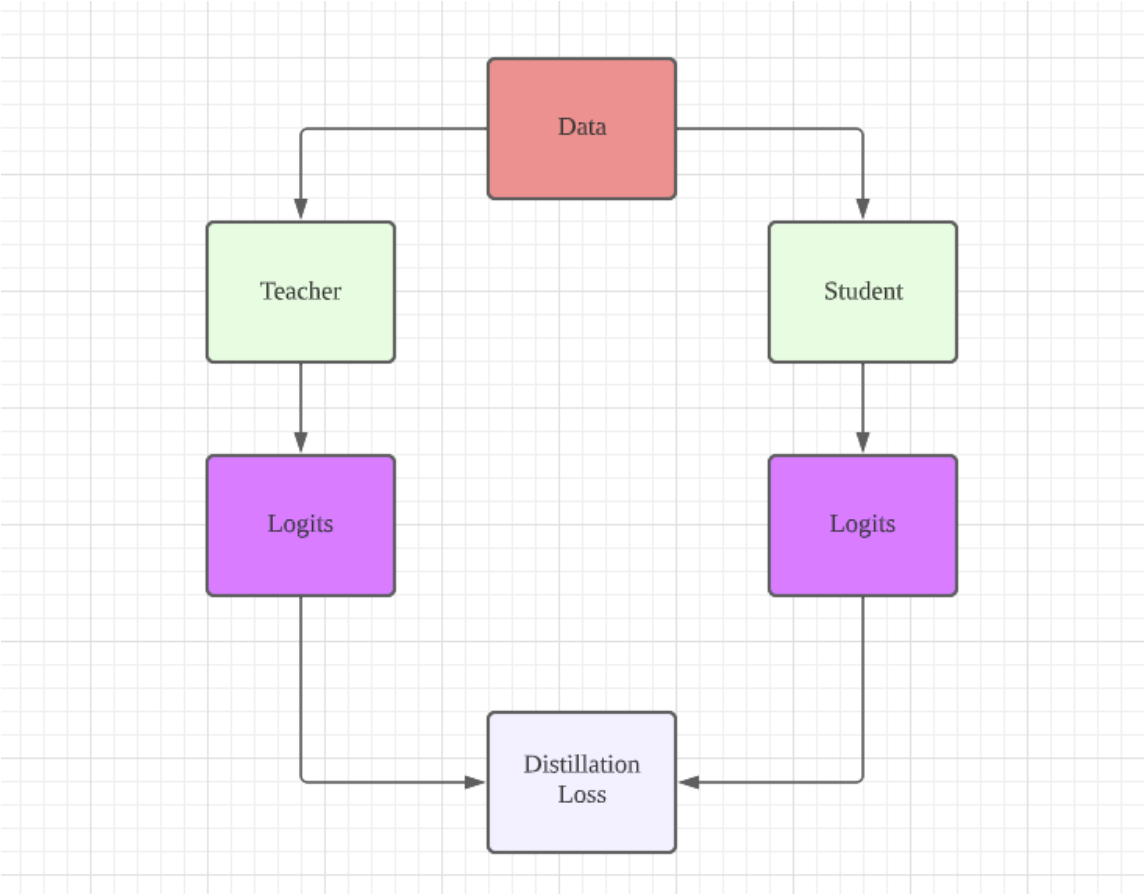


Figure 3. Response-Based Distillation

Feature-based distillation:[2]

A well-trained teacher model is capable of capturing data knowledge in the intermediate layer which is significant for networks. The intermediate layer will learn the distribution of specific features and then use this feature to teach a student model.

As shown below, the teacher model will train the student model by minimising the difference between teacher model feature activations and student feature activations.

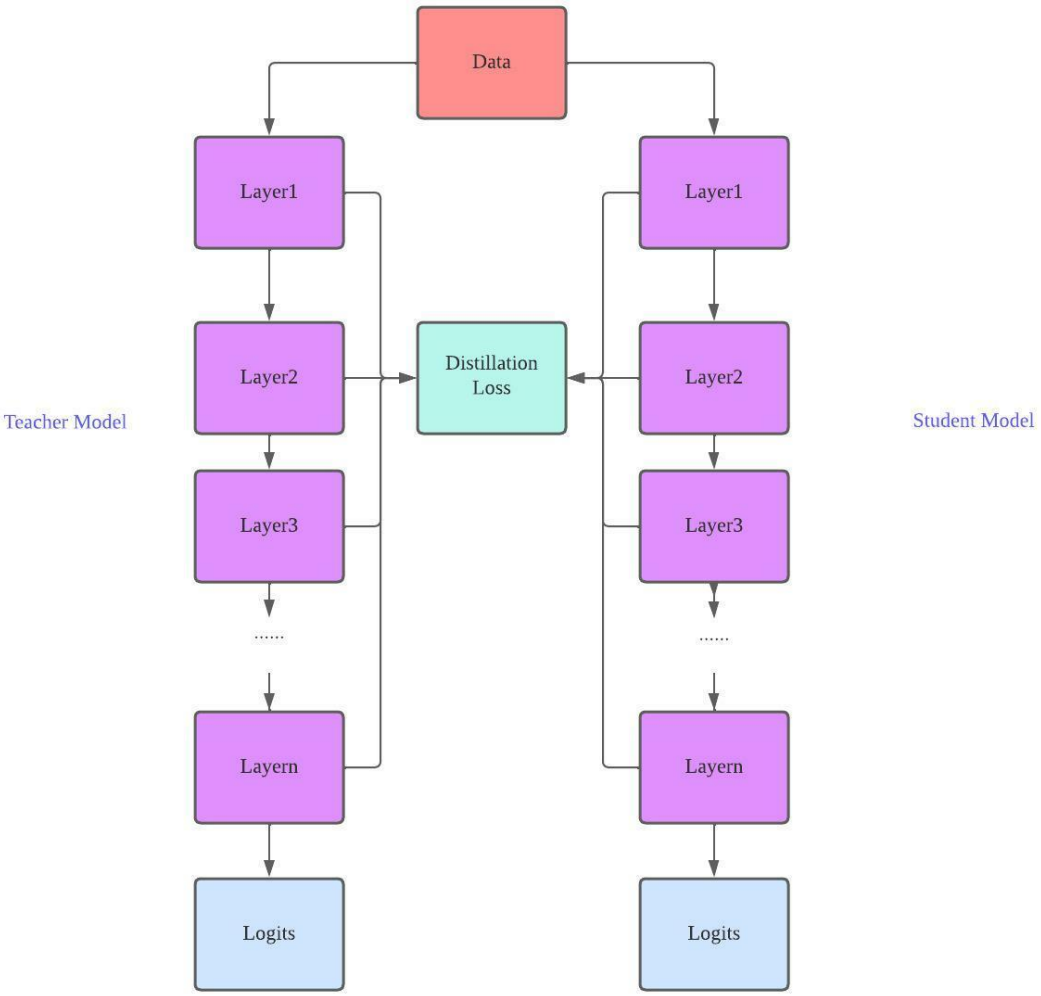


Figure 4. Feature-Based Distillation

Relation-based distillation:

Relation-based knowledge distillation uses the outputs of a teacher model based on some specific layers. Relation-based knowledge distillation focuses on the connections of different layers. Gram matrix defined FSP(flow of solution process) to find out the relationship between different featured maps [2].

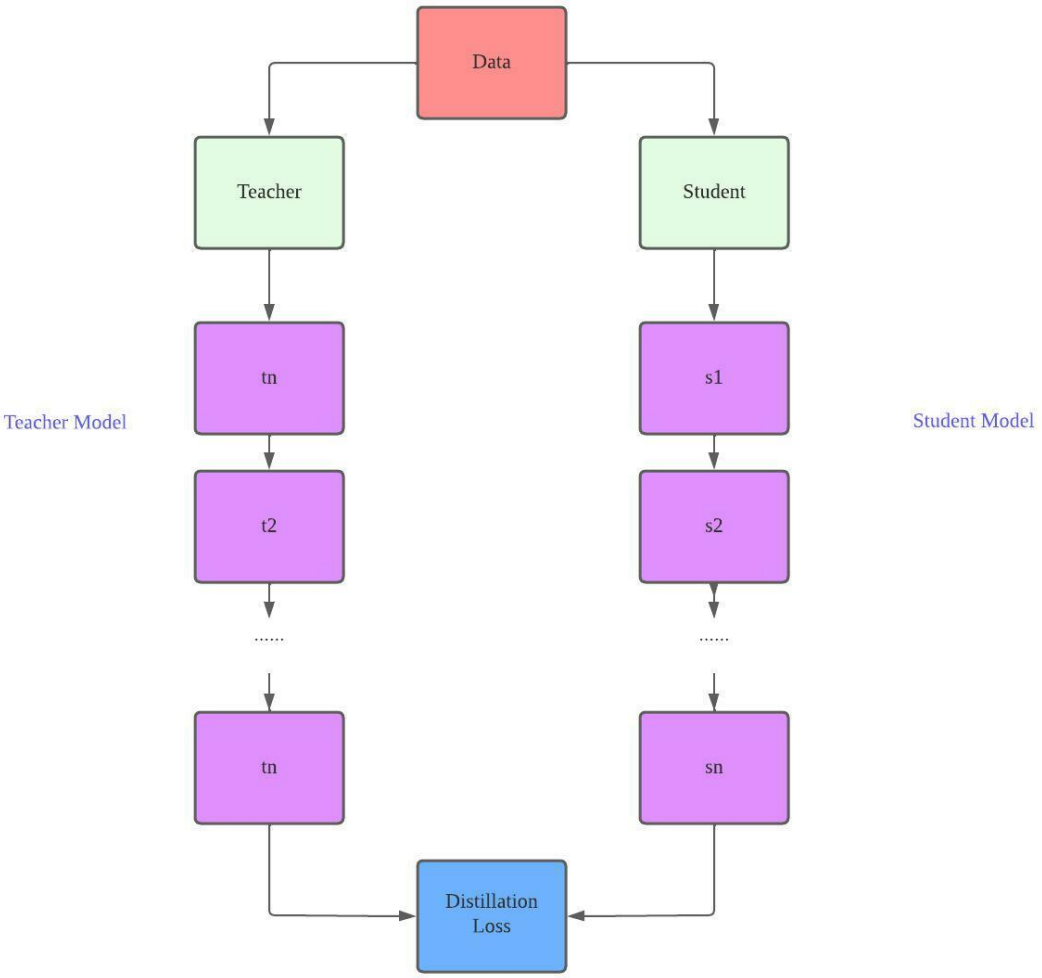


Figure 5. Relation-Based Distillation

1.3 Modes of Knowledge Distillation

There are basically three modes of distillation that we are usually using in our daily lives, which are offline distillation, online distillation, and self distillation.

Offline Distillation:[3]
Offline distillation is a traditional type of knowledge distillation where the user pre-trains a teacher model based on the dataset that the user currently has, then manually supervises the student model's training to complete the distillation progress.
The higher the difference in training accuracy between the teacher model and the student model will result in the better the distillation will be. Usually, the parameters of the teacher model will remain the same. The loss of distillation calculation will be based on the difference of predicted output values and do gradient update to student loss to get a higher accuracy student model.
Offline distillation is a well-established technique and its advantageous in that it is technically easy to implement while training the teacher network is usually complicated and takes a lot of training time.

In this project, we focus our main distillation modes on offline distillation, because our data is not too large and our operation is not complicated. Therefore Offline Distillation is the best way to make our project work.

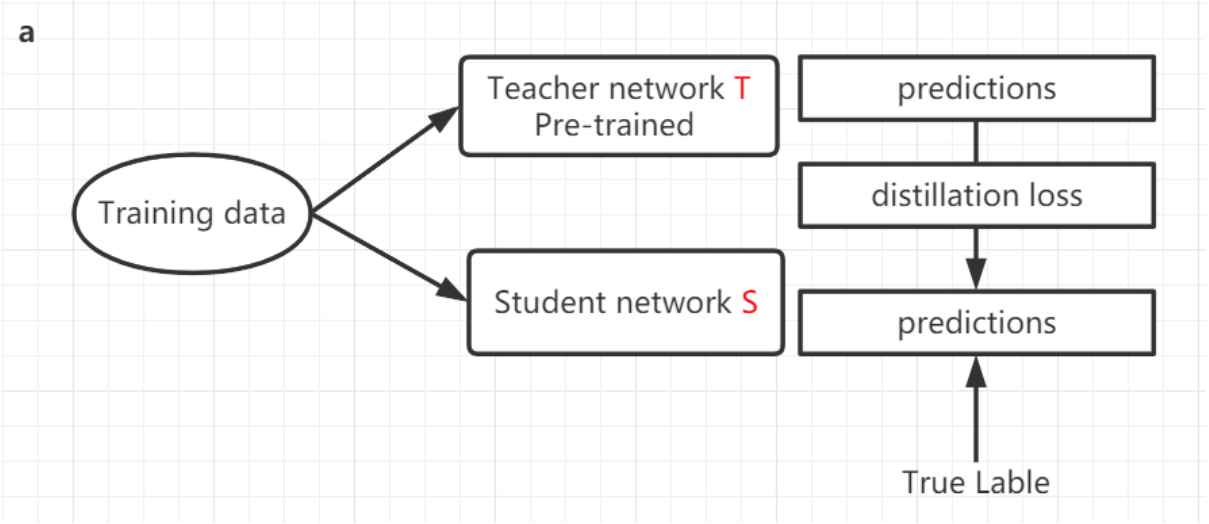


Figure 6. Offline Distillation

Online Distillation:

Online distillation uses the teacher model as a prediction tag to supervise the training process of the student model. Based on the different offline distillation, before training the student model, online distillation inputs part of the data that has not been tagged, using the teacher network output tag as the supervising information and then inputting it to the student network to make the distillation work.

Online distillation can help improve the student model's performance and the improvement will be better when the teacher model with large-capacity is not available.

Online distillation is advantageous in that it can use fewer datasets to improve the accuracy. It can also be used when the teacher network model is not completely finished. Update both teacher model and student model at the same time and the whole progress is trainable from start to end [4].

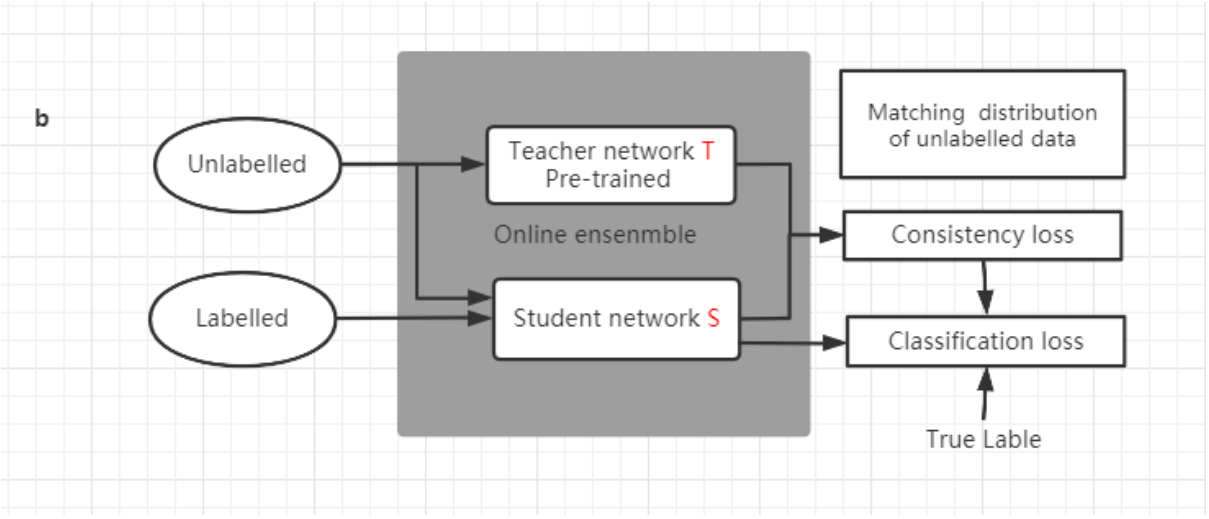


Figure 7. Online Distillation

Self Distillation:
Unlike traditional distillation techniques, self-distillation does not require a teacher network model, instead it has a student model that distils itself. There are many ways to execute self distillation and one of them is to train the student model and leave the final server epochs alone, by using the previous training process as a supervised model to train the last epochs.
Self-distillation is a kind of online distillation where knowledge is distilled into the shallow layer of a network [4].

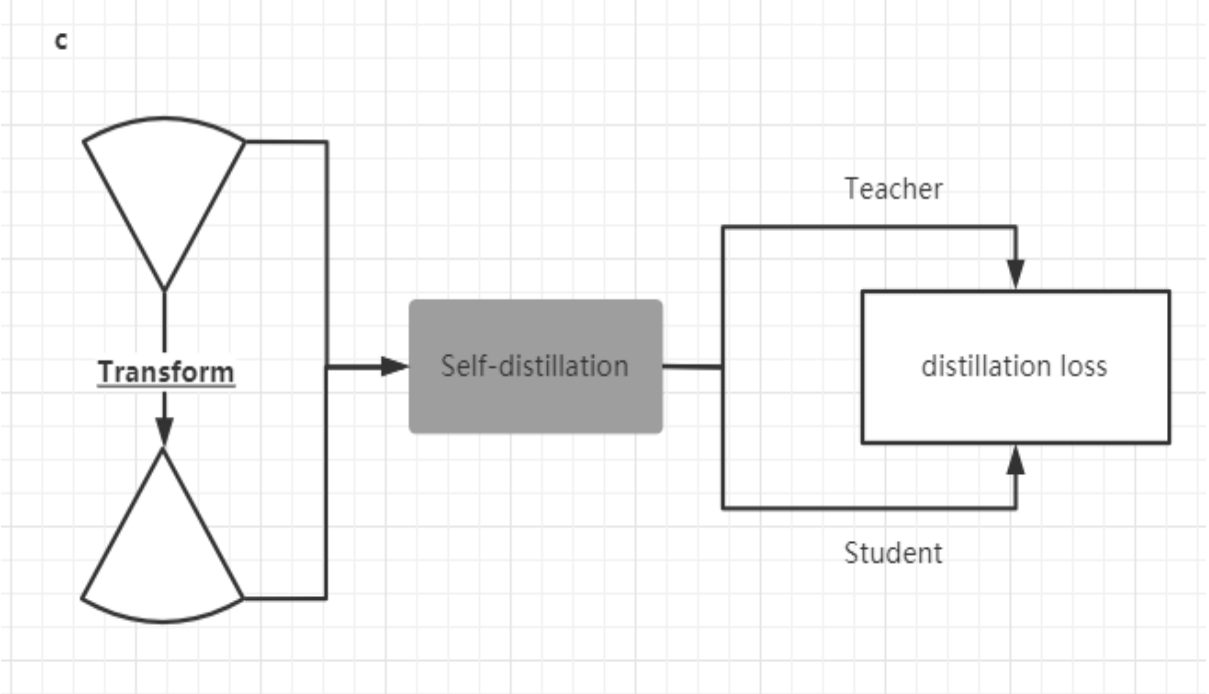


Figure 8. Self Distillation

2. Explanation of Key Terminology

2.1 Hard targets and soft targets

For example, if we give a picture of a horse to a neural network or to a model for image classification, this image may have very many categories (like a cat, dog, car, donkey, etc.), and each category may be identified with a probability (such as the probability that the content of this picture is a cat, the probability that the content of this picture is a dog, etc.)

For a picture of a horse, usually, when we train the network, we only tell the network that the content in this picture is a horse and not something else. This corresponds to a probability of 1 that the content of this picture is a horse and a probability of 0 that it is any other category, and this training method is called hard targets. However, the hard target is not reasonable. Once the picture of a horse is labelled a hard target, this kind of label is the same as telling the network that the content of this picture is just a horse, and it is completely impossible to be similar to other categories of things.

But in real life, there are similarities between horses and donkeys in various characteristics, and there are almost no similarities between horses and cars in various characteristics. In hard targets, the probability that a picture of a horse is 1, but the probabilities that it is a donkey or a car are both 0.

This means that a horse is as similar to a donkey as a horse is to a car, which defies our common sense of categorization in real life.

If we give the picture of the horse to an image classification neural network that has been trained, the network may give another result (such as the probability of this picture is a horse is 0.7, the probability of it being a donkey is 0.25, and the probability of it being a car is 0.05), which is called the soft target. The result given by soft targets is more realistic, it not only shows that the content of this picture has the highest probability of being a horse but also shows the probability that horses and donkeys are more similar, the similarity between horses and cars and the similarity between donkeys and cars are very low. It follows that soft targets can convey more information.

Hard targets		Soft targets	
Horses	1	Horses	0.7
Donkeys	0	Donkeys	0.25
Cars	0	Cars	0.05

Figure 9. Processing of Image Information by Hard Targets and Soft Targets

The soft targets do contain more information. As shown in the figure below, on the identification of the first number, soft targets contain information such as the number has the highest probability of being 2 but also has a higher probability of being 3, and the number has a very small probability of being a number other than 2 or 3. But the only information in the hard targets is that this number is 2. For the identification of the second number, soft targets contain information that the number is most likely to be 2, but also has a higher probability of being 7, and that the number has a very small probability of being a number other than 2 or 7. But the only information contained in hard targets is the number 2.

It can be seen that soft targets cover more knowledge and information, if we simply use hard targets we will erase significant amounts of information, such as what else the image looks like, what the image does not look like, and how similar the image is to something else and the relative magnitude of the probability of a non-correct category (such as the similarity between donkeys and cars mentioned above).

Therefore, we need to train the teacher network with hard targets first, and after training the teacher network, the network is able to convey more information about the prediction results of images (i.e. soft targets). And we can use the soft targets trained by the teacher network as labels to train the student network more efficiently.

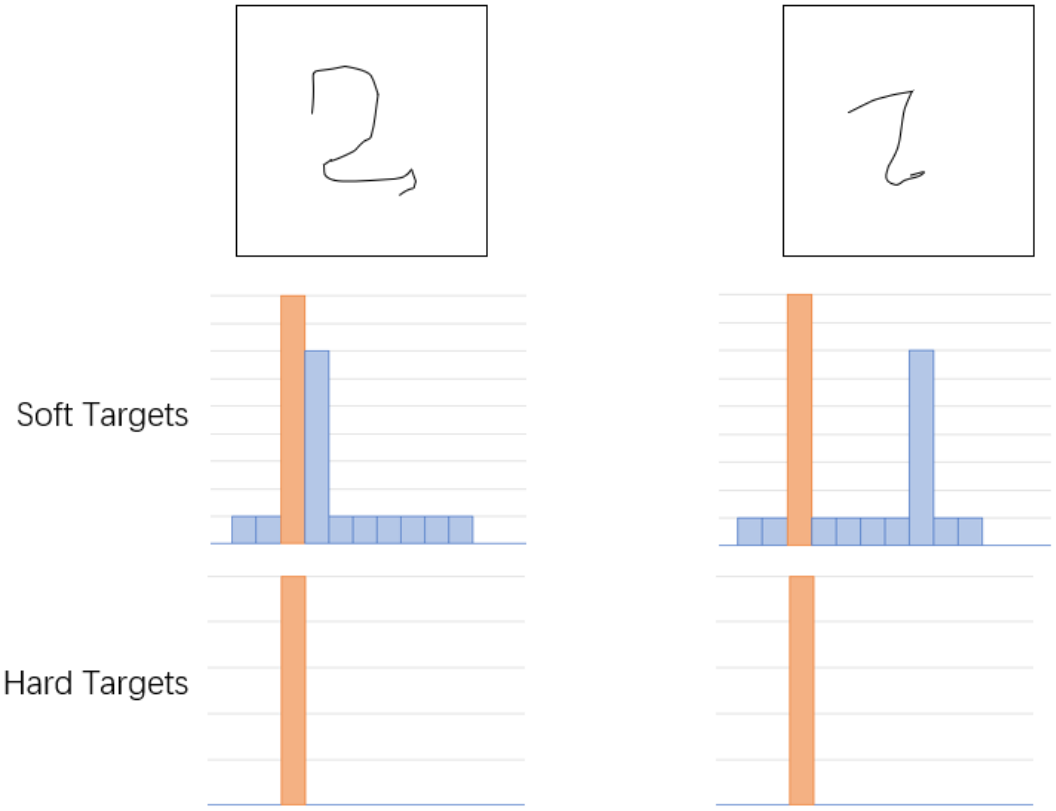


Figure 10. Information Contained in Soft Targets and Hard Targets in Digital Recognition

2.2 Distillation temperature

In order to better train the student network, we need to perform one more operation. We need to amplify the probability of the other categories in the soft label to allow the student network to decouple the information of non-correct categories more clearly. For this purpose we need to introduce the distillation temperature T.

It was mentioned in the paper that neural networks typically produce class probabilities by using a "softmax" output layer that converts the logit, z_i, computed for each class into a probability, q_i, by comparing z_i with the other logits [1].

$$q_i = \frac{exp(z_i/T)}{\Sigma_j exp(z_i/T)} \quad (1)$$

T in the above equation is the distillation temperature T. When T = 1, the equation represents the "softmax". As T becomes larger, the information in soft targets becomes 'softer', i.e. the relative size of the non-correct category of information becomes larger.

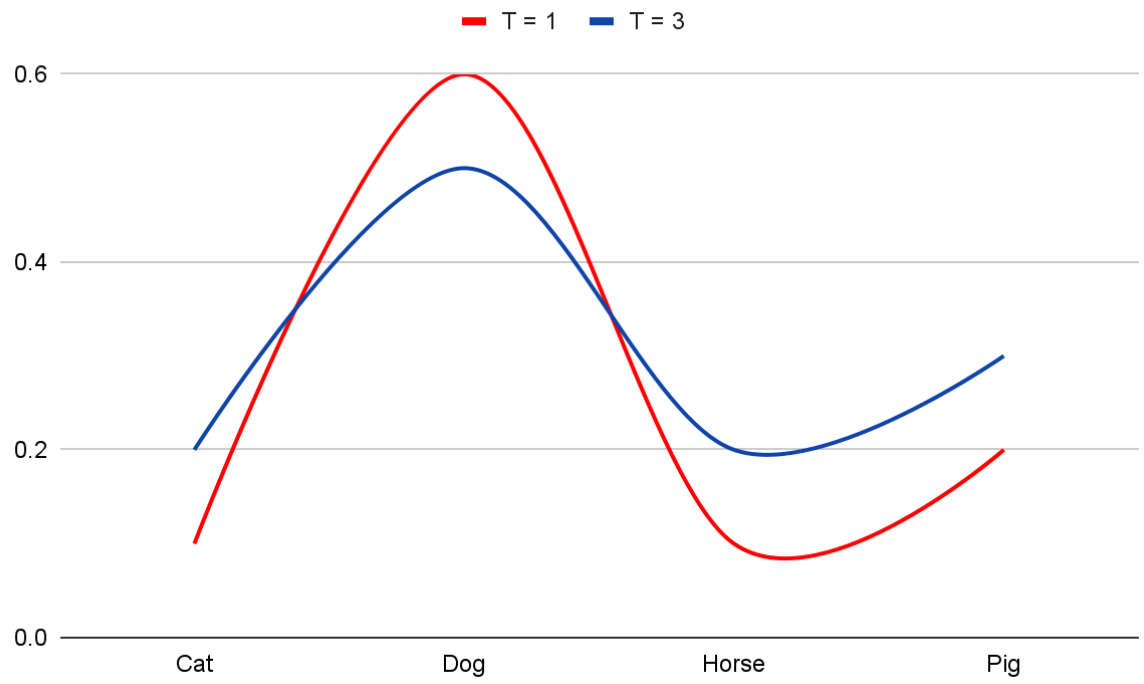


Figure 11. The Probability of Different Information in Soft Targets when T is Different

The distillation temperature T can control the size of the difference in probability between the messages, the smaller T can highlight the difference in probability of each message, but the larger T can not highlight the difference. However, when T becomes very large, the difference in probability between individual messages is almost non-existent.

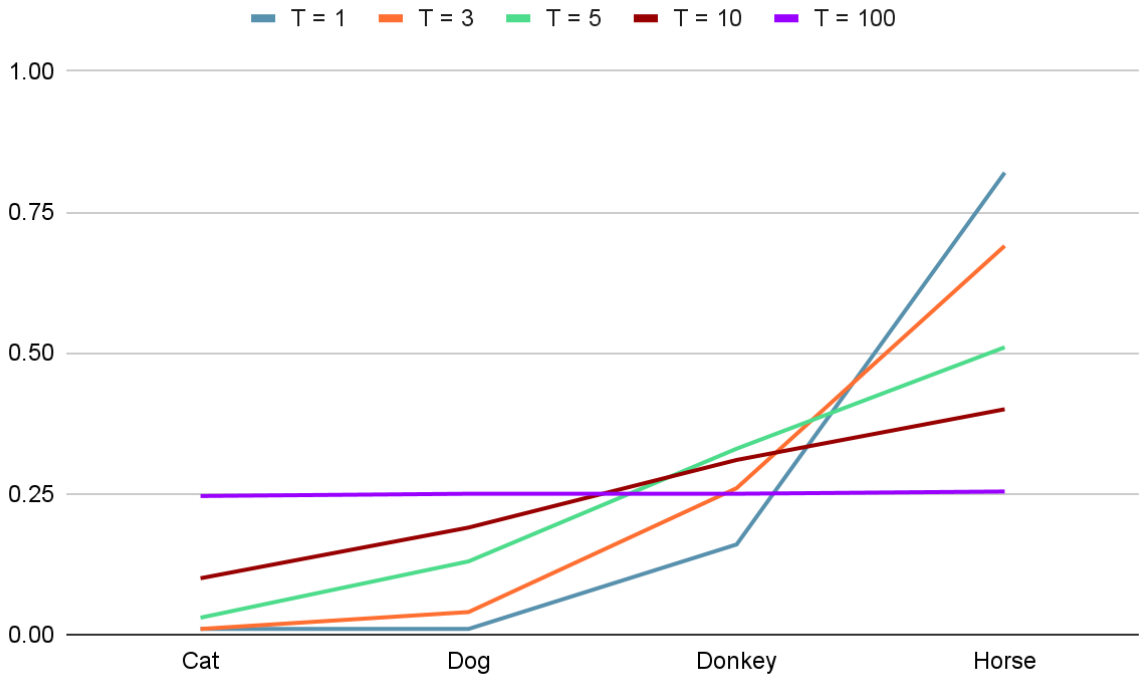


Figure 12. The Distribution of Information Probabilities of Soft targets for Different Values of T

3. Implementation

3.1 The Process of Distilling the Knowledge

First, we have a trained teacher model where a large amount of data has been imported into the model. The teacher model generates a softmax with distillation temperature t for each data. By softmax, we will get soft labels.

Then we feed this data into the student model, which may not have been trained or is in the process of being trained. Two softmax are obtained based on the distillation temperature t and the distillation temperature 1, and the soft predictions and hard predictions are obtained by these two softmax respectively.

After this, we use soft labels and soft predictions to find a loss function called distillation loss. We expect the values of soft labels and soft predictions to be as close as possible. We then use hard prediction and hard label y (i.e. ground truth) to generate another loss function called student loss. We also expect the value of hard prediction and the value of hard label y to be as close as possible.

After we get the distillation loss and student loss, we multiply the distillation loss by the weight β and multiply the student loss by the weight α, and we add these two items to get the total loss. The total is a weighted loss, which is combined with distillation loss and student loss. It should be pointed out that the sum of β and α is 1 (i.e. $\beta + \alpha = 1$). Our ultimate goal is to minimise the total loss by fine-tuning the weights in the student model through gradient descent and backpropagation.

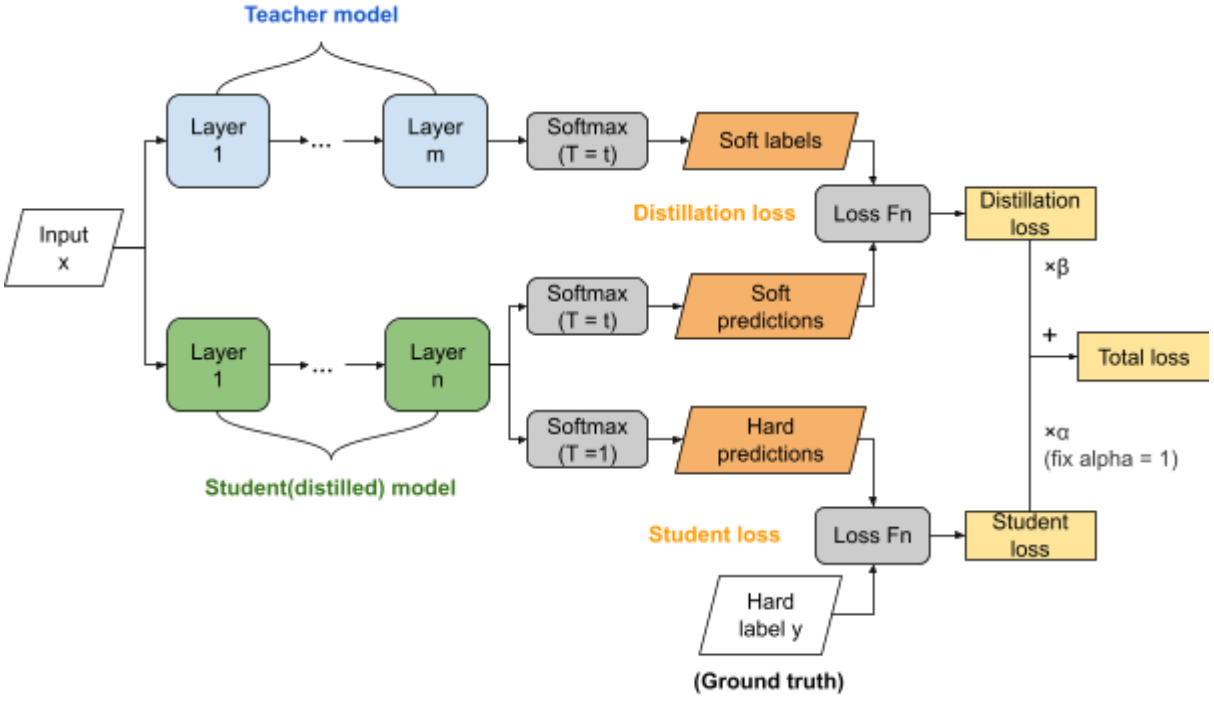


Figure 13. The Process of Distilling the Knowledge

When we finish the training process of knowledge distillation mentioned above, i.e., when the student model is trained, we can directly put the vector into the student model to complete the prediction results. The student models we end up with are small and useful and can be applied to various scenarios.

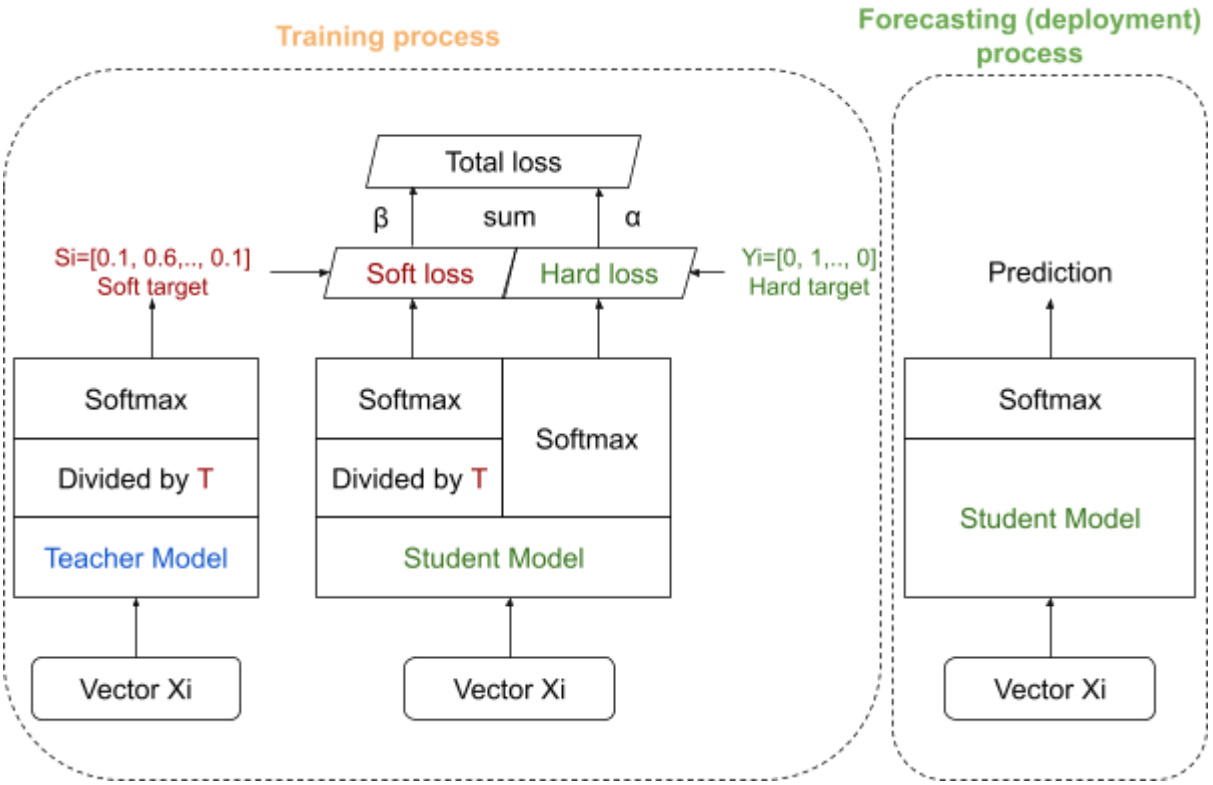


Figure 14. The Complete Process of Knowledge Distillation

3.2 Python Implementation

We implemented the algorithm of knowledge distillation as mentioned above. We used Pytorch to do all the training and evaluating steps. Our code is also available on Github.

Please follow the instructions of the README.md file in the root folder to get the result of the experiments below. For some training steps, we tried to run on Google Colab, so some directories may be strange. Google Colab did not allow us to use too much time, so we could not store the whole experiment results on the jupyter notebook. Some variables were not automatically tested on the code, like temperature and β. These two variables are edited by hand, such as the code we use for section 4.1.

4. Experimentation

4.1 MNIST - Image Classification - Fully Connected Neural Network

Firstly, we tried to achieve knowledge distillation on the MNIST dataset as in the original paper in order to verify whether our implementation is correct. We use two different neural networks for the teacher model and the student model respectively.

Also, for comparison, we tried to use different distilling temperatures. In our first experiment, we just set the soft targets and hard targets equally weighted, which means that the value of β is equal to 0.5.

We found that the different OS (Puquan tested code on Ubuntu, while Tianyu tested code on Windows) will result in different accuracy even if using the manual seed. Thus we save the model with temperature=10 and soft loss(beta) =0.5, the screenshot for training results could be found in the appendix B (Figure B.1, Figure B.2, Figure B.3).

Method

To start, we created two fully connected neural networks, of which we selected the more complex network as the teacher model, and the less complex network as the student model. The teacher model featured three levels of fully connected neural networks, with 2 million trainable parameters and a dropout layer. The student model only had two layers of neural networks, with 11 thousand trainable parameters and no dropout layer.
The experimental methodology is as follows：

1. Load the MNIST dataset with 60000 training set examples, and 10000 testing set examples.
2. Set the global optimizer as the Adam optimizer, using a learning rate of 0.0001.
3. Train the teacher model (6 epochs).
4. Train the student model (6 epochs) without knowledge distillation for result comparison.
5. Create a new student model
6. Use the teacher model to train the student model using knowledge distillation (6 epochs)
7. Test with different distillation temperatures and beta values as independent variables

Results:

The results comparison is as follows:

	Teacher model (6 epochs)	Student model (6 epochs)
Accuracy	98.03%	91.69%

Table 1. Test Accuracy for Teacher Model and Student Model without Distillation

Student model with distillation (after 6 epochs):

	T = 5	T = 7	T = 10
β = 0.5	91.71%	91.90%	92.04%

Table 2. Test Accuracy with Distillation

Analysis:

The result shows that the distillation gives a slight improvement to the student network, the accuracy is higher when applying a higher temperature.

After training the teacher model and student model with distillation after 6 epochs, we found that when the soft loss (beta) is equal to 0.5 with temperature >=5 the student model with distillation is slightly higher than the student model without distillation.

4.2 CIFAR10 - Image Classification - ResNet(Convolutional Neural Network)

Next, we attempted to test with the CIFAR10 dataset. Unlike the previous experiment, we substituted fully connected neural networks with convolutional neural networks. In this experiment, we selected the ResNet networks as our models. To go deeper, we tried to use more values of temperature and β than in the last experiment.

Method:

We selected the ResNet50 network as the teacher model, with 25 million trainable parameters, while ResNet18 was used as the student model with 11 million trainable parameters. The number of trainable parameters were calculated by the torch-summary package. We used the Adam optimizer during the training process.

The experimental methodology is as follows:

1. Load the CIFAR10 dataset with 50000 training set examples and 10000 testing set examples.
2. Train the ResNet18 (30 epochs) without pre-trained weight, using a learning rate of 0.0002.
3. Train the ResNet50 (30 epochs) with pre-trained weight, using a learning rate of 0.0002.
4. Create a new student model ResNet18 which is without pre-trained weight.
5. Use the teacher model to train (85 epochs) the student model using knowledge distillation, using a learning rate of 0.002.
6. Repeat step 5, testing with different distillation temperatures and beta values.
7. Save all trained models.

Results:

The experimental results are as follows,

	ResNet18 from Scratch	ResNet50 with Pretrained
Test Accuracy	59.11%	80.96%

Table 3. Test Accuracy without Knowledge Distillation

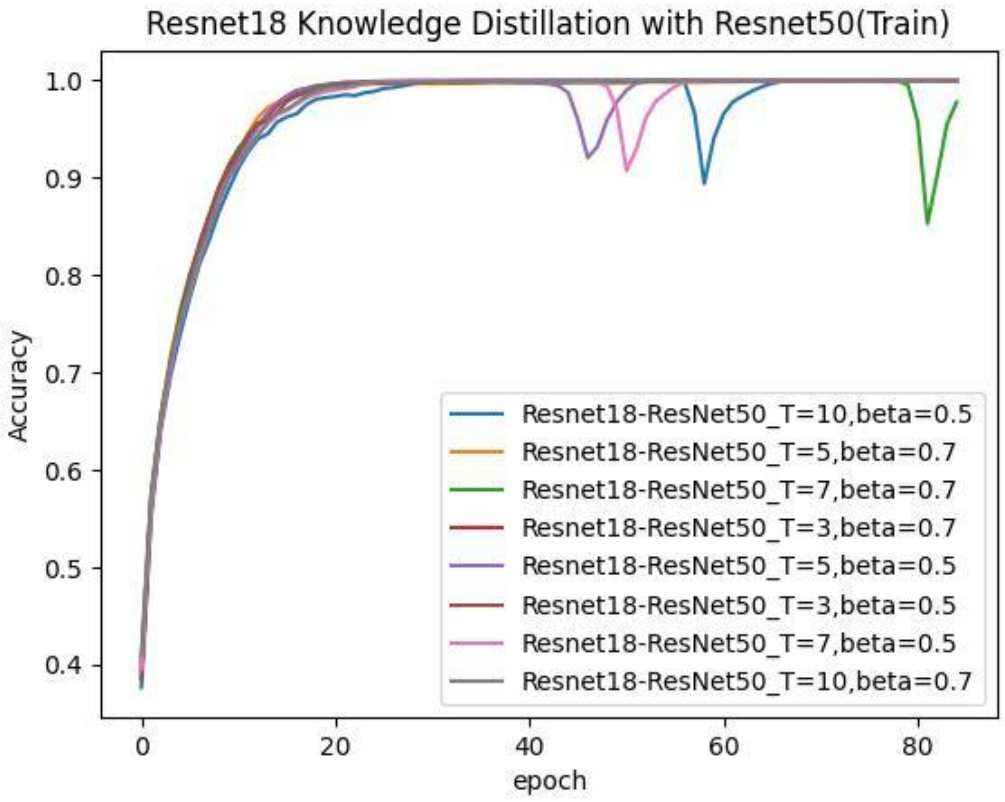
	T = 3	T = 5	T = 7	T = 10
β = 0.7	75.49%	75.62%	76.48%	76.34%
β = 0.5	75.82%	76.64%	77.09%	77.60%

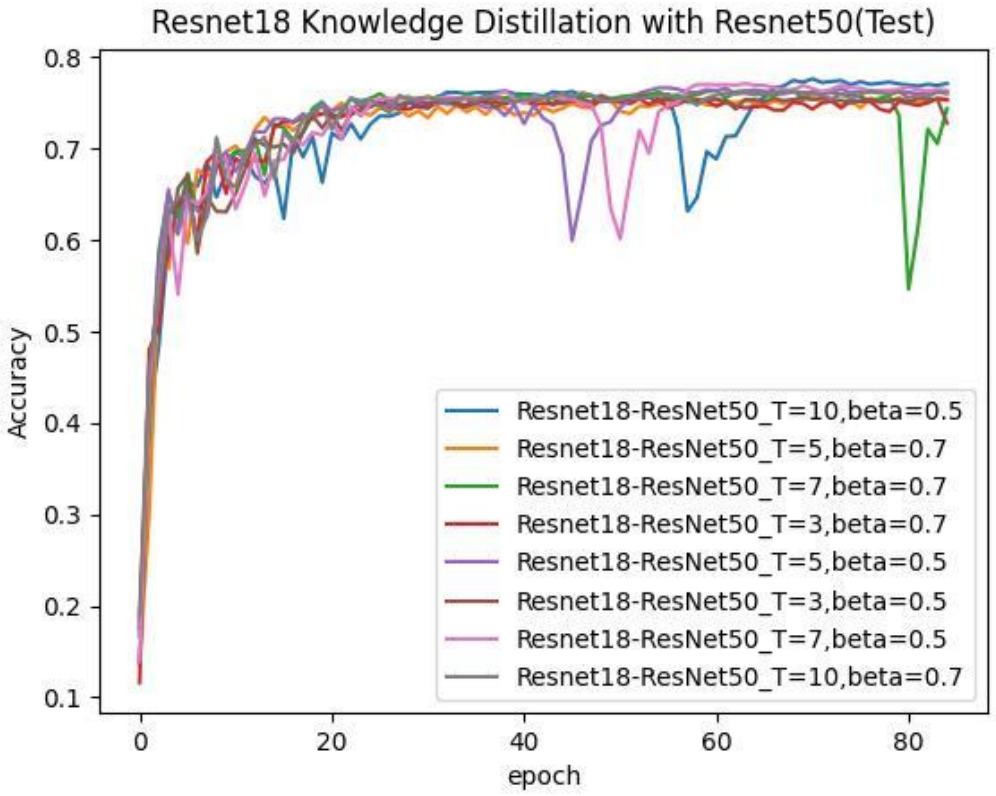
Table 4. Test Accuracy of ResNet18 with ResNet50 Knowledge Distillation

Analysis:

From the results, we can see that knowledge distillation did a great job. The performance of student models ResNet18 can almost reach the same test accuracy as the teacher model ResNet50. From the table above, we can see that a higher distilling temperature will lead to better performance in this experiment. However, when using the same temperature, the lower values of β (the weight of soft loss) will cause better test accuracy. According to the figures below, we can find that student models converge quickly when we use higher distilling temperature, see Figure 15.

According to Figure 16 and Figure 17, we can find that around the 25th epoch, every student model can reach 99% training accuracy. These student models are overfitting. This situation is similar as we train the original ResNet18 model from scratch, achieving around 98% training accuracy, and only 55% testing accuracy. Though knowledge distillation cannot effectively avoid overfitting, we can see that after around the 40th epoch, the accuracy curves (blue, pink, purple, green) have big drops, then increase, resulting in better accuracy, especially the blue curve which is the best result.





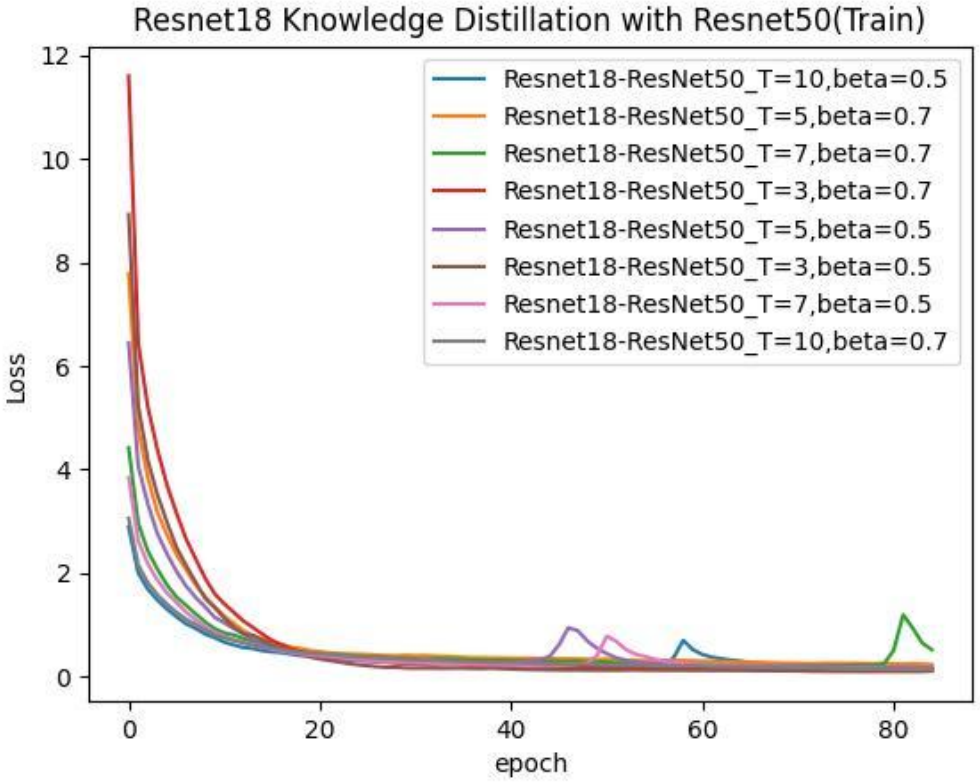


Figure 15,16,17. The Complete Process of Knowledge Distillation

4.3 CIFAR10 - Image Classification - VGG16(Convolutional Neural Network)

We also conducted the final experiment on the CIFAR10 dataset. In the previous experiment, we used neural network models of the same design but different depths, ResNet50 and ResNet18, where the difference in trainable parameters was by about a factor of two.

In this experiment, we selected different designs and depths of neural network models, and a far greater difference in trainable parameters to demonstrate the difference in the effectiveness of knowledge distillation.

Method:

For the final experiment, we selected the vgg16 network as the teacher model. This model hosts 138 million trainable parameters, while the ResNet18 network remains the student model. This results in the difference in trainable parameters between teacher and student models being over a factor of ten as opposed to just two in the previous experiment.

The experimental methodology is as follows:

1. Load the CIFAR10 dataset with 50000 training set examples and 10000 testing set examples.
2. Train the VGG16 (30 epochs) with pre-trained weight, using a learning rate of 0.0001.
3. Create a new student model ResNet18 which is without pre-trained weight.
4. Use the teacher model to train (85 epochs) the student model using knowledge distillation, using a learning rate of 0.002.
5. Repeat step 5, testing with different distillation temperatures and beta values,

The experimental results are as follows:

	ResNet18 from Scratch	VGG16 with pre-trained
Test Accuracy	59.11%	85.93%

Table 5. Model Accuracy without Knowledge Distillation

	T = 3	T = 5	T = 7	T = 10
β = 0.7	76.57%	76.19%	75.90%	75.14%
β = 0.5	77.14%	77.22%	76.10%	76.27%

Table 6. ResNet18 Model Accuracy with VGG16 Knowledge Distillation

Numbers of Training Parameters in Millions		
ResNet18	ResNet50	VGG16
11.4	23.9	134.7

Table 7. Numbers of Training Parameters for VGG, ResNet

Analysis:

First, we do not get the same conclusions as the experiment of 4.2. From the table above, we can see that higher distilling temperature will not lead to better performance of student models. However, the lower values of β still work better when we use the same distilling temperature.

Secondly, we find that although VGG16 has better performance than ResNet50, it does not teach as well as ResNet50 during knowledge distillation. In terms of generalization, we can compare the best results from VGG16-distillation(T=5, β=0.5) and ResNet50-distillation(T=10, β=0.5). According to the figures below, when using ResNet50 as the teacher model, ResNet18 has a much better generalization ability to the test datasets than using VGG16 as the teacher model, since there is a gap between the two loss curves below.

To see other training and testing plots in detail, please see Appendix A.

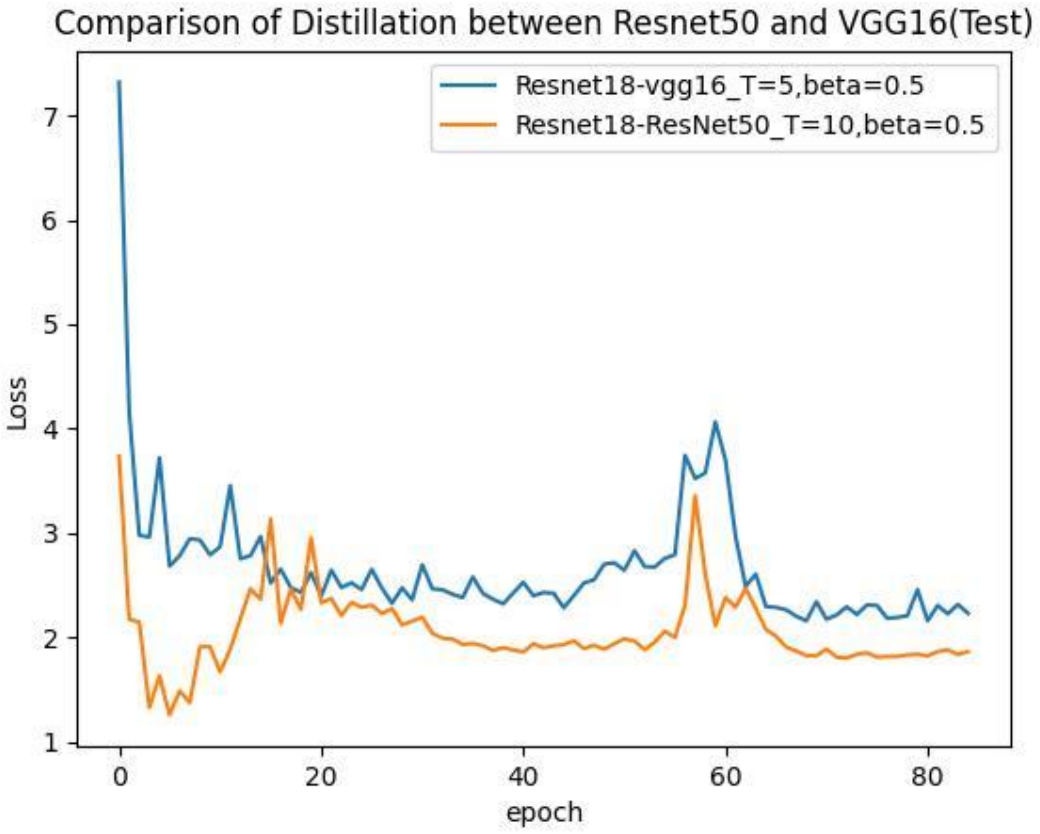


Figure 18. Loss Comparison between VGG16, ResNet50 Knowledge Distillation

5. Conclusions

Knowledge Distillation Result				
Teacher(Pretrained)	Student(Scratch)	Accuracies	T	β
ResNet50 (80.96%)	ResNet18 (59.11%)	77.60%	10	0.5
VGG16 (85.93%)	ResNet18 (59.11%)	77.22%	5	0.5

Table 8. Result Summary of Knowledge Distillation

Knowledge distillation is an innovative and effective method that can be applied to a dataset. It allows the compression of a very large and effective model into a more robust, smaller model that carries out the same tasks at a much more optimised speed while retaining much effectiveness. The training process is optimised such that overfitting is prevented even in very large models with millions of trainable parameters. The teacher-guided student convergence park will be more accurate and distillation makes the model focus more to improve accuracy.

As demonstrated by our experiments, we verified the validity of our knowledge distillation model by replicating a successful increase in accuracy within the first experiment. From there, we observed a trend that the distillation temperature was proportionally increasing with the resulting accuracy specifically when both models were of the same type of network. The final experiment also demonstrated that previously observed trends do not necessarily hold up when the network structure of the teacher and student models are not the same. The ResNet18 student trained with distillation from VGG16 showed that significantly more trainable parameters in the teacher model do not guarantee a higher accuracy and that the highest temperature does not always mean the highest student accuracy either.

Overall, we have demonstrated the effectiveness of knowledge distillation in transferring knowledge and testing accuracy from a large teacher model into a smaller student model. In doing so, we observed many trends within our results that would take further testing to have concrete reliability. In every experiment, knowledge distillation caused a significant increase in the accuracy of the student model when compared to the same model trained from scratch.

6. Further work

Knowledge distillation is currently used on multi-modes which includes visionary, text, sound data analysis, knowledge mapping and pre-trained big models. As of now, the main area of use for knowledge distillation is in the area of classification, specifically a conventional 2d image. As the need for similar prediction algorithms extends to such areas as 180 or 360 degree vision, knowledge distillation is sure to extend its area of research into new imaging formats [5].

In our experiments, each distillation with new parameters took upwards of 30 minutes per test, and thus we were not able to test in large quantities. Hence, we could not gather concrete evidence for some of the smaller trends observed in our results. Further areas of research would include more testing to see if the increase in distillation temperature stopped increasing the accuracy of the student model at a certain point when both student and teacher are of the same network structure. Another point of interest would be to test if increasing the trainable parameter count in the teacher model would cause the student model to be more accurate if they were of the same network structure. It would also be interesting to see if the same trends hold true in other types of distillation.

The current method of knowledge distillation is to train the student model based on the teacher model by using the teacher model to "teach" the student model and nothing further. As techniques of knowledge distillation improve, the student model may extend to benefit the teacher model after being trained by the teacher model. And as our need for optimised models grows with machine learning advancements, the technique of knowledge distillation will follow and improve into the future.

Bibliography

[1] G. Hinton, O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network", *arXiv.org*, 2022. [Online]. Available: https://arxiv.org/abs/1503.02531. [Accessed: 18- Jul- 2022].

[2] V. Lendave, "A beginner's guide to Knowledge Distillation in Deep Learning", *Analytics India Magazine*, 2022. [Online]. Available: https://analyticsindiamag.com/a-beginners-guide-to-knowledge-distillation-in-deep-learning/#:~:text=According%20to%20Knowledge%20Distillation%3A%20A,Let%27s%20discuss%20them%20briefly. [Accessed: 31- Jul- 2022]

[3] "DeepLearning, model compress of knowledge distillation_tensorflow", *Blog.csdn.net*, 2022. [Online]. Available: https://blog.csdn.net/u012655441/article/details/120835715. [Accessed: 31- Jul- 2022]

[4] "Summary of Knowledge Distillation：Mechanism of Distillation - Baidu Library", *Wenku.baidu.com*, 2022. [Online]. Available: https://wenku.baidu.com/view/30773901a6e9856a561252d380eb6294dd88223f.html. [Accessed: 31- Jul- 2022]

[5]L. Wang and K. Yoon, "Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks", *arXiv.org*, 2022. [Online]. Available: https://arxiv.org/abs/2004.05937. [Accessed: 31- Jul- 2022].

Appendix A

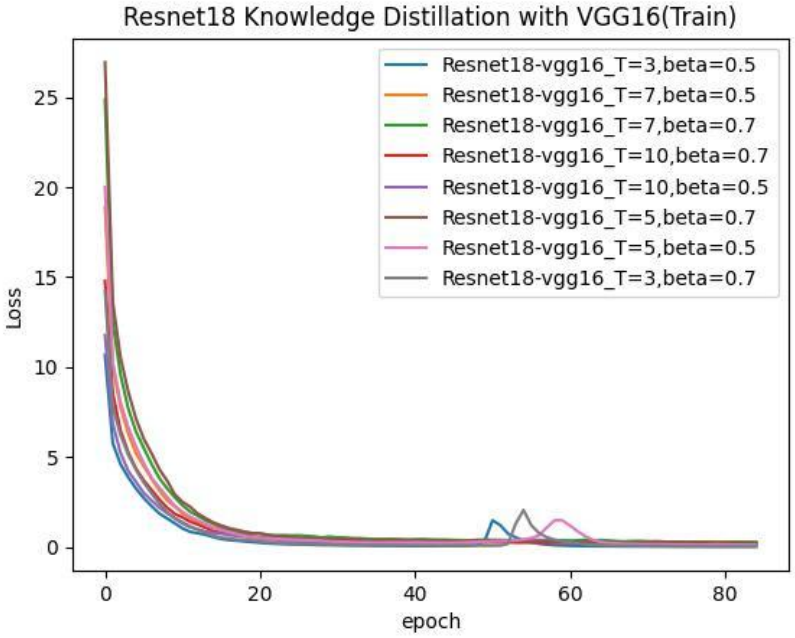


Figure A.1 Training Loss

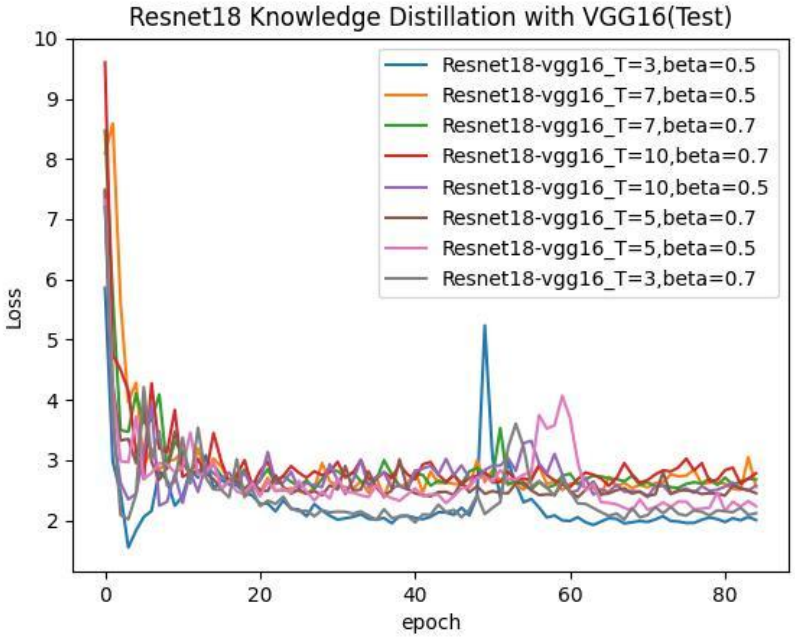


Figure A.2 Testing Loss

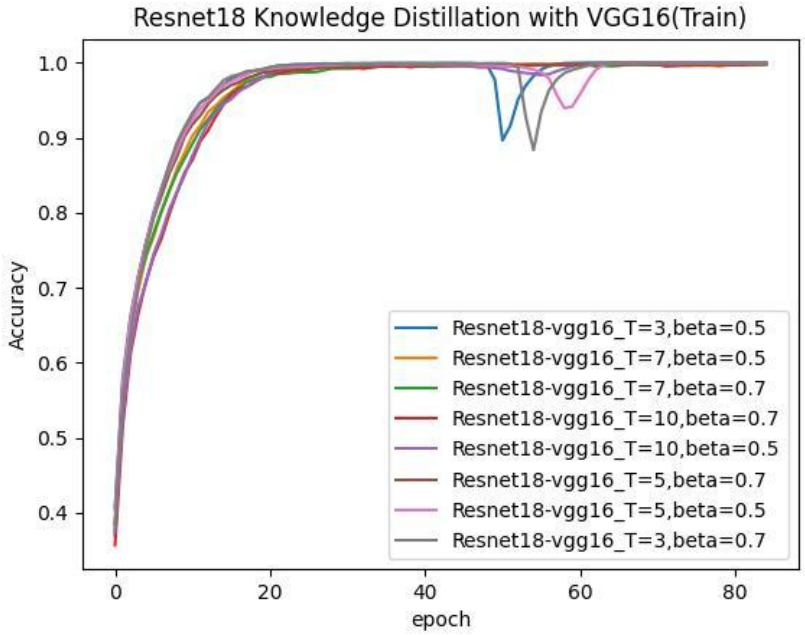


Figure A.3 Training Accuracy

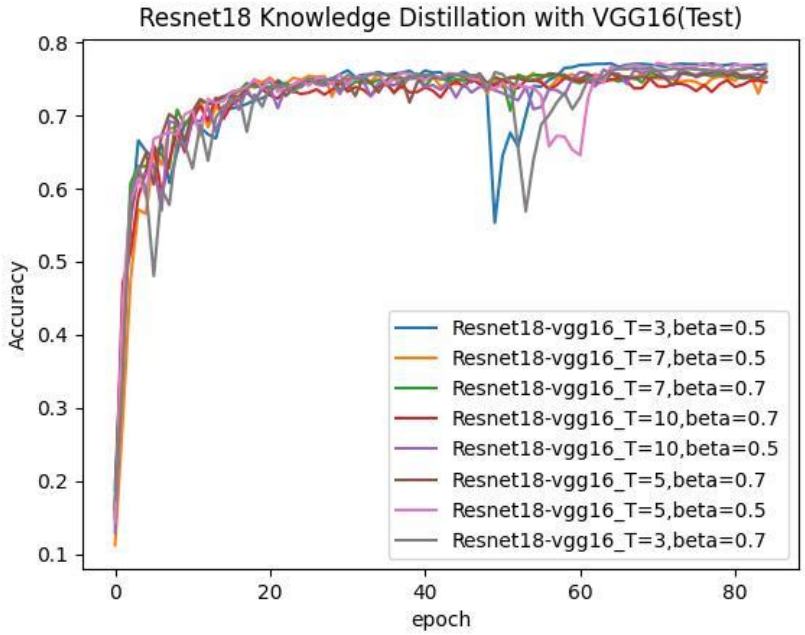


Figure A.4 Testing Accuracy

Appendix B

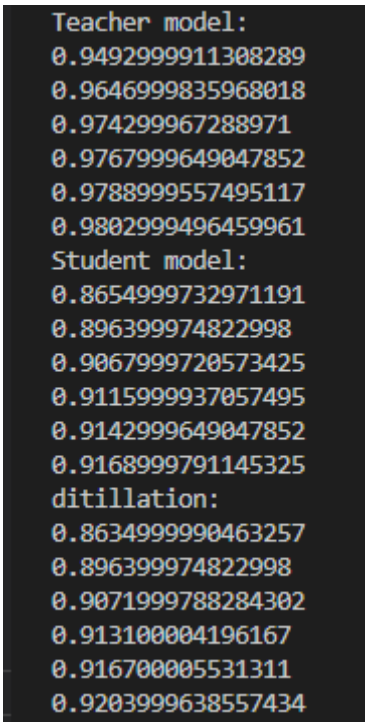


Figure B.1 MNIST distillation temperature=10 result

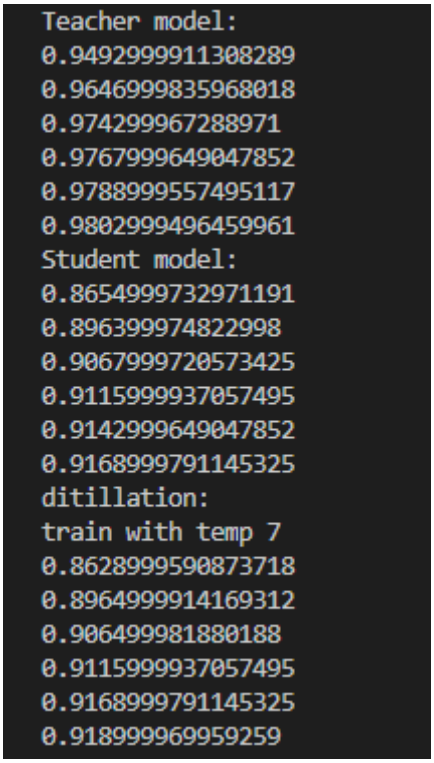


Figure B.2 MNIST distillation temperature=7 result

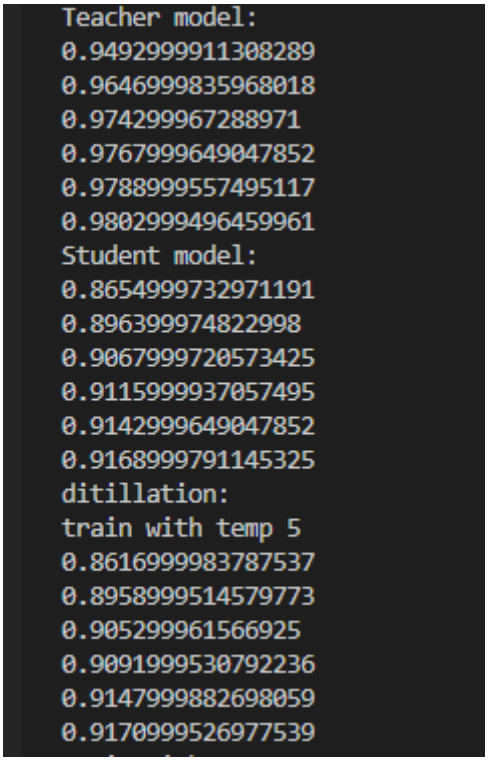


Figure B.3 MNIST distillation temperature=5 result