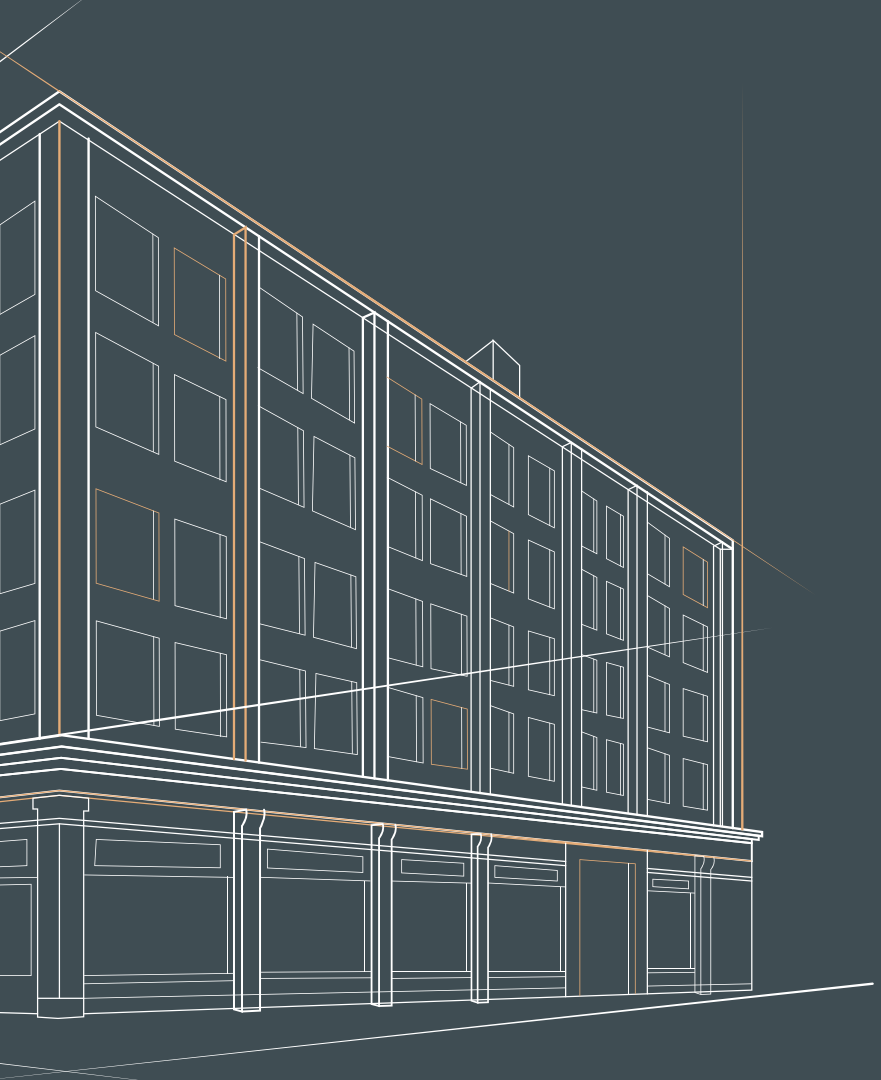
Triplet Loss for Knowledge Distillation

Team No 7:
Sashank S (2021701038)



01

Introduction & Proposal

Problem Statement

Knowledge distillation - technique for transferring knowledge of deep ensemble models with many parameters (teacher model) to smaller shallow models (student model)

Purpose of knowledge distillation is to increase the similarity between the teacher model and the student model

The Task : Make the student model closer to the teacher model using pairs or triplets of the training samples.

Goals

Goal 1

Build a model that can increase the similarity of outputs for similar samples.

Metric learning aims at reducing the distance between similar and increasing the distance between dissimilar.

Goal 2

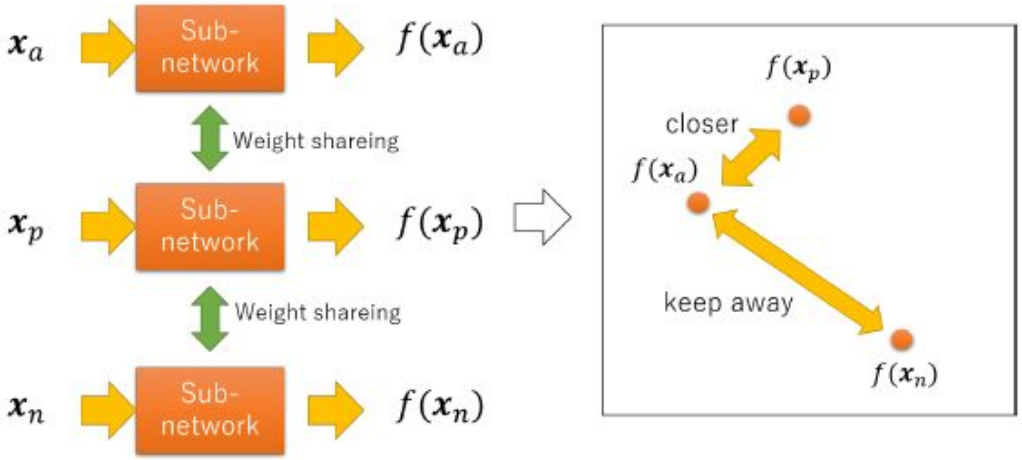
The functionality of the metric learning to reduce the differences between similar outputs can be used for the knowledge distillation to reduce the differences between the outputs of the teacher model and the student model.

Goal 3

Use metric learning to clarify the difference between the different outputs, and improve the performance of the student model.

Triplet Network

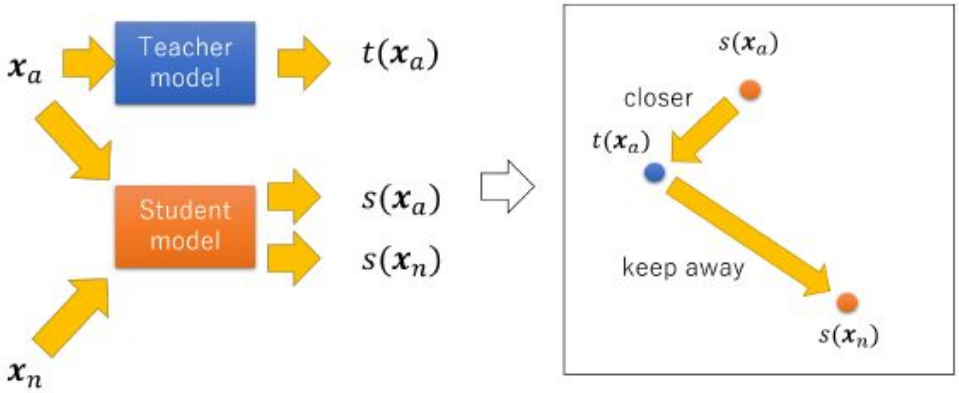
- Triplet Network is designed to learn embedding from a triplet of samples called ”anchor”, ”positive” and ”negative”.



The triplet network learns embedding such that the distance between ”anchor” and ”positive” is smaller than the distance between ”anchor” and ”negative”.

$$E = \sum_{(a,p,n)\in\Theta} max(0, m + ||f(x_a) - f(x_p)||_2^2 - ||f(x_a) - f(x_n)||_2^2)$$

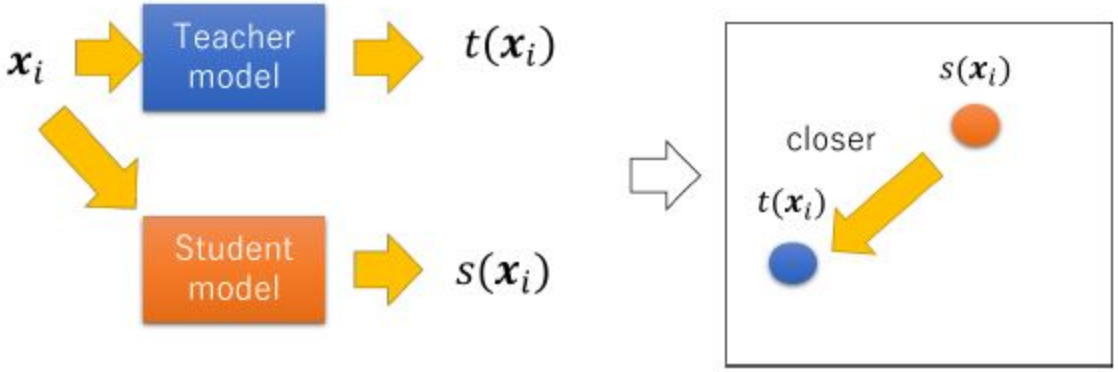
Architecture



$$E_{ourKD} = \sum_{(a,n)\in\Omega} max(0, m + ||t(\boldsymbol{x}_a) - s(\boldsymbol{x}_a)||_2^2 - ||t(\boldsymbol{x}_a) - s(\boldsymbol{x}_n)||_2^2) ,$$

$$E_{KD} = E_{hard} + \lambda_{soft} E_{soft}$$

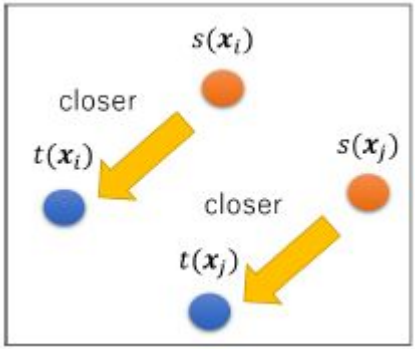
Knowledge Distillation



- Knowledge distillation is a technique for transferring knowledge of deep or ensemble model with many parameters (teacher model) to smaller shallow model (student model).

Hinton's KD

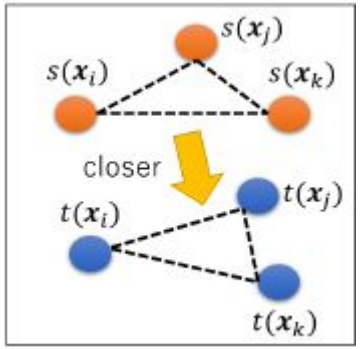
- Hinton et al. [2] proposed training the student model so that the softmax outputs of the teacher model and the softmax outputs (probability) of the student model are close (HKD).
- They used the KL-divergence of the softmax outputs of both models as the loss for training the student model.



$$E_{HKD} = \sum_{i \in \chi} KL(\text{softmax}(\frac{t(x_i)}{T}), \text{softmax}(\frac{s(x_i)}{T}))$$

$$KL(p, q) = \sum_i p_i log(\frac{p_i}{q_i})$$

Park's KD



- Park et al. [3] expressed the relationship between the outputs of the teacher model as the Euclidean distance between the two outputs, and transferred it to the student model (RKD-D).

$$\psi_D(t(\boldsymbol{x}_i), t(\boldsymbol{x}_j)) = \frac{||t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)||_2}{\sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \chi^2} ||t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)||_2},$$

- Optimized the student model by Huber loss so that the similarity between the outputs of the student model and the similarity of the outputs of the teacher model got closer.

$$E_{RKD-D} = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \chi^2} l(\psi_D(t(\boldsymbol{x}_i), t(\boldsymbol{x}_j)), \psi_D(s(\boldsymbol{x}_i), s(\boldsymbol{x}_j)))$$

$$l(p, q) = \begin{cases} \frac{1}{2}(p-q)^2 & (|p-q| \leq 1) \\ |p-q| - \frac{1}{2} & (otherwise) \end{cases}$$

Park's KD

They also used the cosine of the angle formed by the three outputs as the similarity of the model outputs (RKD-A).

$$\psi_A^{(t_{ijk})} = \cos \angle t(\boldsymbol{x}_i)t(\boldsymbol{x}_j)t(\boldsymbol{x}_k) \qquad \cos \angle t(\boldsymbol{x}_i)t(\boldsymbol{x}_j)t(\boldsymbol{x}_k) = \langle \boldsymbol{e}^{(t_{ij})}, \boldsymbol{e}^{(t_{kj})} \rangle$$

- Optimized the student model by Huber loss so that the similarity between the outputs of the student model and the similarity of the outputs of the teacher model got closer.

$$\boldsymbol{e}^{(t_{ij})} = \frac{t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)}{||t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)||_2},$$

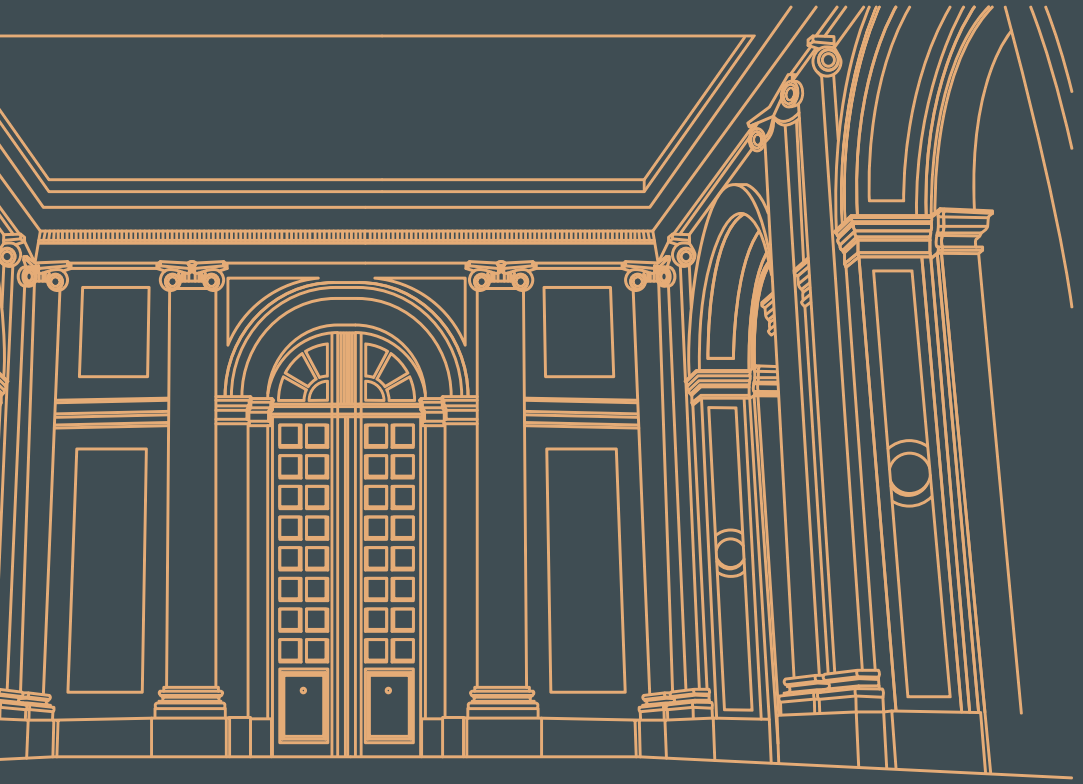
$$\boldsymbol{e}^{(t_{kj})} = \frac{t(\boldsymbol{x}_k) - t(\boldsymbol{x}_j)}{||t(\boldsymbol{x}_k) - t(\boldsymbol{x}_j)||_2}.$$

$$E_{RKD-A} = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in \chi^3} l(\psi_A^{(t_{ijk})}, \psi_A^{(s_{ijk})})$$

Park's KD

- They also argued that using both angles and Euclidean distance for knowledge transfer would further improve the performance of the student model (RKD-DA).

$$E_{RKD-DA} = \lambda_{RKD-D} E_{RKD-D} + \lambda_{RKD-A} E_{RKD-A}$$

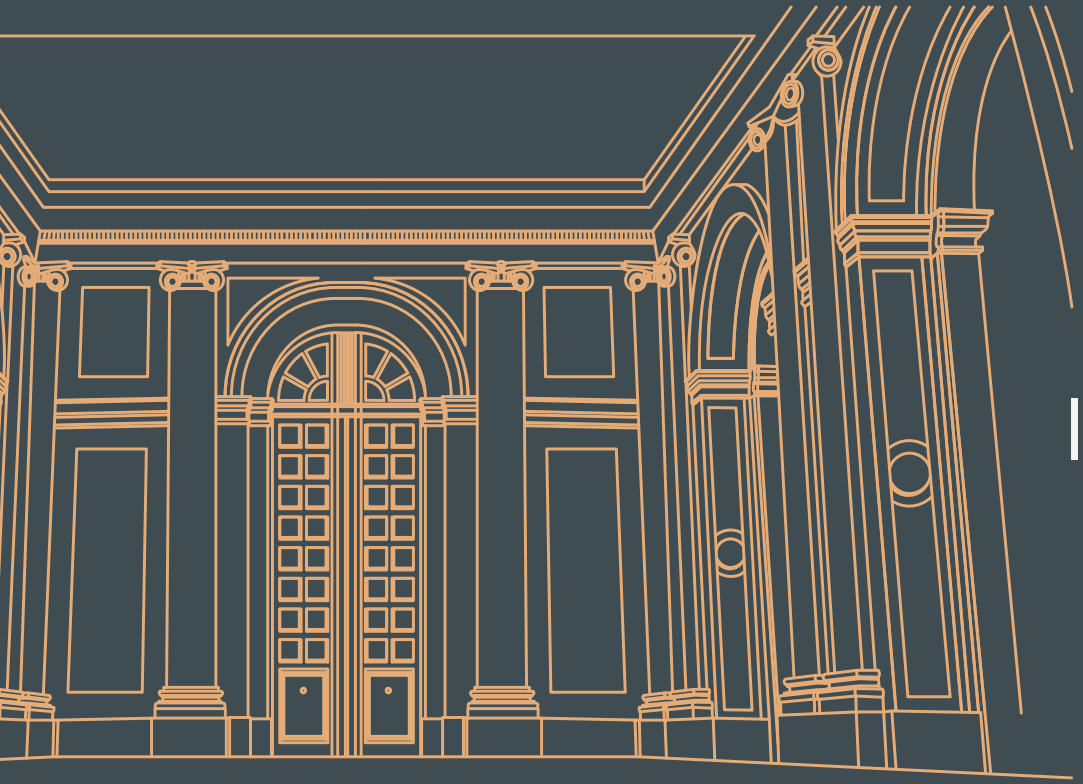


02

Project Expectations

Expected Deliverables

- Analysing and Creating Dataloaders for different Datasets (MNIST, FashionMNIST, CIFAR10)
- Build and Train the Teacher Model
- Build the Student Model and Distill the Knowledge from the Teacher
- Analyse the effect of using Triplet Loss along with Hinton's KL Divergence and Park's RKD-DA Losses



03

Implementation Details

CIFAR 10 Model

CIFAR-10 is a dataset that includes color images of 10 kinds of objects such as "automobile" and "dog."
The size of each image in CIFAR-10 is **32 x 32**.
The numbers of the training samples and the test samples are **50,000 and 10,000**.

```
Net_teacher(
  (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv2): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv4): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (fc1): Linear(in_features=2048, out_features=512, bias=True)
  (fc2): Linear(in_features=512, out_features=128, bias=True)
  (fc3): Linear(in_features=128, out_features=10, bias=True)
  (batchnorm1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (batchnorm2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (batchnorm3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (batchnorm4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (batchnorm5): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (dropout): Dropout(p=0.5, inplace=False)
  (relu): ReLU()
)
```

```
Net_student(
  (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv2): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (fc1): Linear(in_features=1024, out_features=128, bias=True)
  (fc2): Linear(in_features=128, out_features=10, bias=True)
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (dropout): Dropout(p=0.5, inplace=False)
  (relu): ReLU()
)
```

MNIST Model

MNIST is a dataset of handwritten digits.
The size of each image in CIFAR-10 is **28 x 28**.
The numbers of the training samples and the test samples are **60,000 and 10,000**.

```
TeacherNetwork(
  (fc1): Linear(in_features=784, out_features=1200, bias=True)
  (fc2): Linear(in_features=1200, out_features=1200, bias=True)
  (fc3): Linear(in_features=1200, out_features=10, bias=True)
)
```

```
StudentNetwork(
  (fc1): Linear(in_features=784, out_features=400, bias=True)
  (fc2): Linear(in_features=400, out_features=10, bias=True)
)
```

Fashion MNIST Model

Fashion MNIST is a dataset of Zalando's article images
The size of each image in CIFAR-10 is **28 x 28**.
The numbers of the training samples and the test samples are **60,000 and 10,000**.

```
TeacherNetwork(
  (fc1): Linear(in_features=784, out_features=1200, bias=True)
  (fc2): Linear(in_features=1200, out_features=1200, bias=True)
  (fc3): Linear(in_features=1200, out_features=10, bias=True)
)
```

```
StudentNetwork(
  (fc1): Linear(in_features=784, out_features=400, bias=True)
  (fc2): Linear(in_features=400, out_features=10, bias=True)
)
```

Evaluation

Loss : Distillation loss uses the soft targets to minimize the squared difference between the logits produced by the teacher model and the logits produced by the student model.

$$E_{KD} = E_{hard} + \lambda_{soft} E_{soft}$$

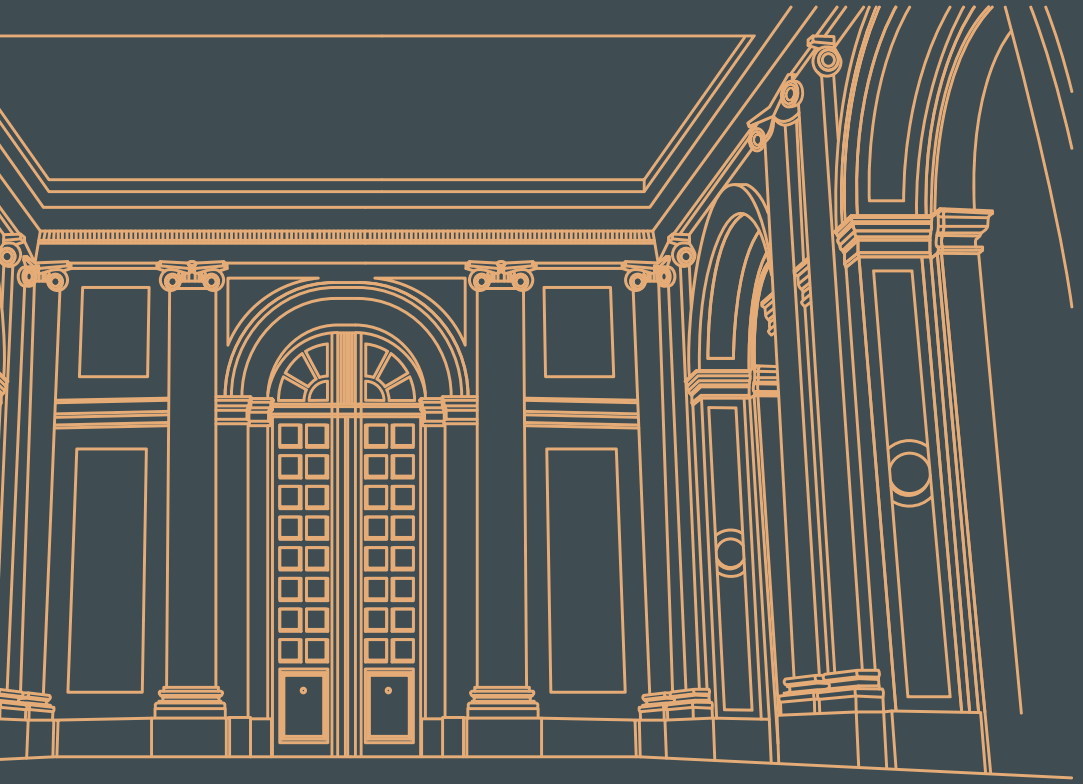
$$E = E_{hard} + \lambda_{soft} E_{soft} + \lambda_{HKD} E_{HKD}$$

$$E = E_{hard} + \lambda_{soft} E_{soft} + \lambda_{HKD} E_{HKD} + \lambda_{RKD-DA} E_{RKD-DA}$$

Accuracy : Accuracy is a metric that generally describes how the model performs across all classes. It is calculated as the ratio between the number of correct predictions to the total number of predictions.

Implementation Details

1. **Network Implemented using Pytorch**
   Both Teacher and Student Models are implemented. Teacher model is first trained and then distillation performed onto the student.
2. **Distillation Loss Calculation**
   Triplet Loss + HKD Loss + RKD-DA Loss implemented from scratch
3. **Data Loader Built for Triplets**
   Each Triplet has an Anchor, Positive and Negative sample
4. **Student Model Distilled with Early Stopping**
   All losses and metrics are logged and visualized

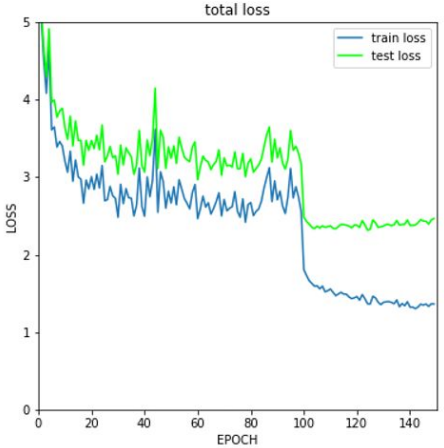


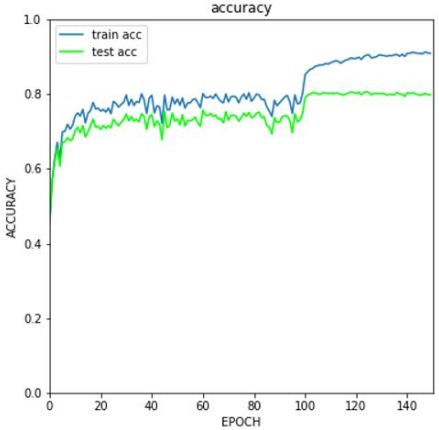
04

Results

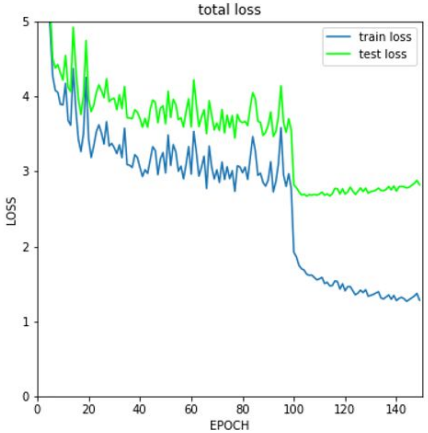
Loss Curves - CIFAR 10

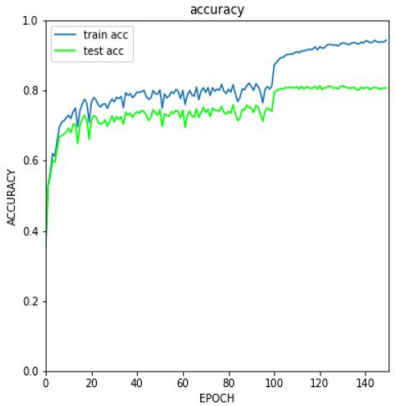
Triplet Loss



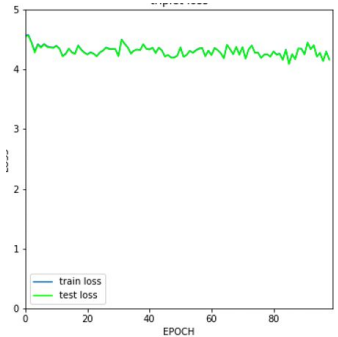


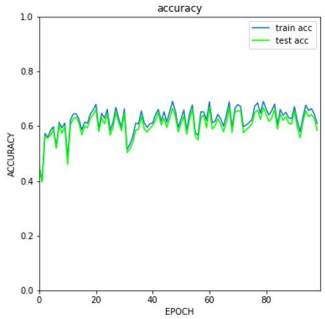
Triplet Loss + HKD Loss





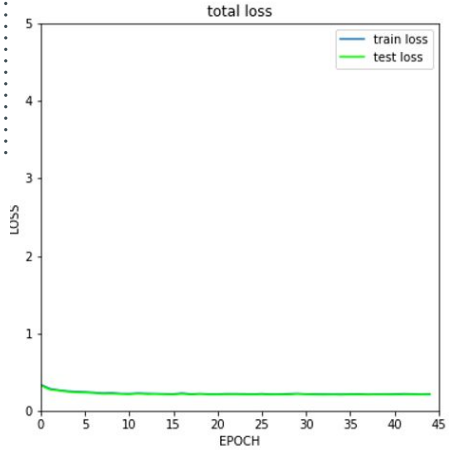
Triplet Loss + HKD Loss + RKD-DA Loss

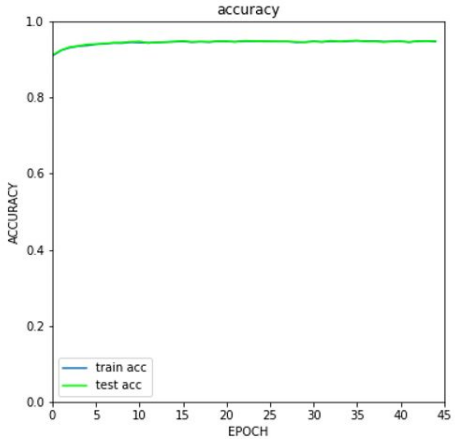




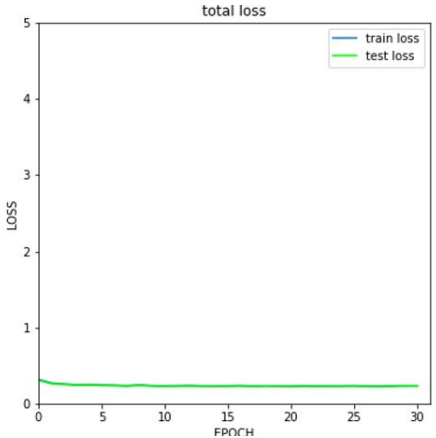
Loss Curves - MNIST

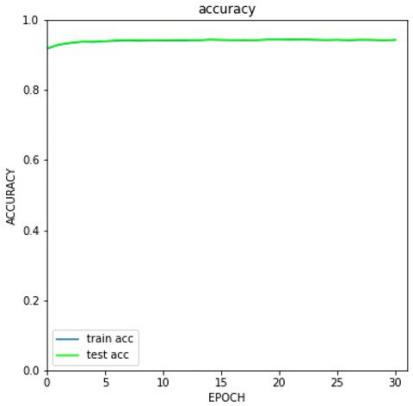
Triplet Loss



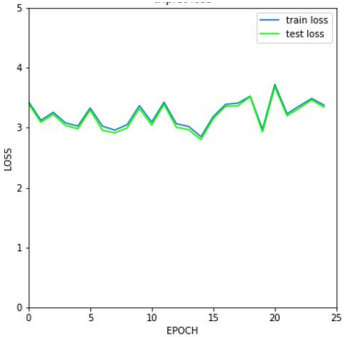


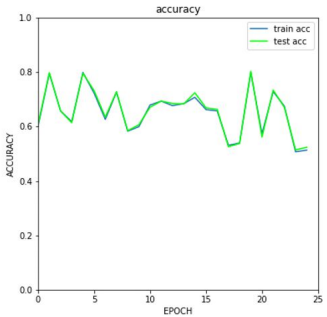
Triplet Loss + HKD Loss





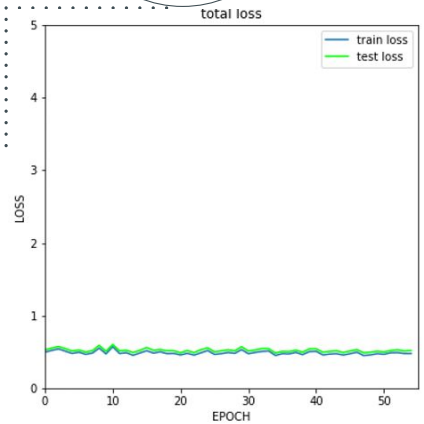
Triplet Loss + HKD Loss + RKD-DA Loss

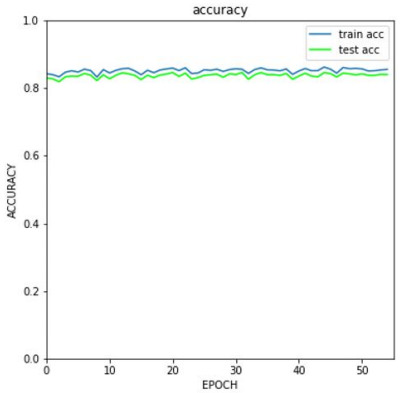




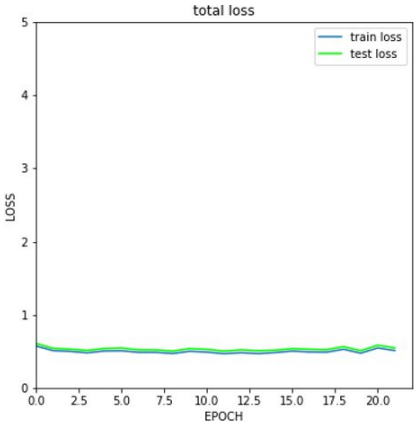
Loss Curves - Fashion MNIST

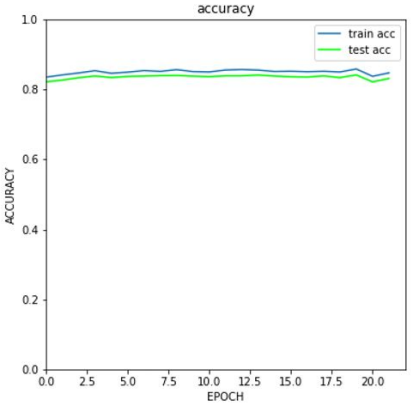
Triplet Loss



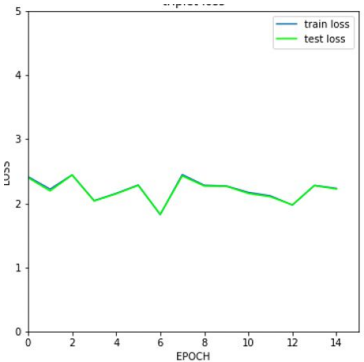


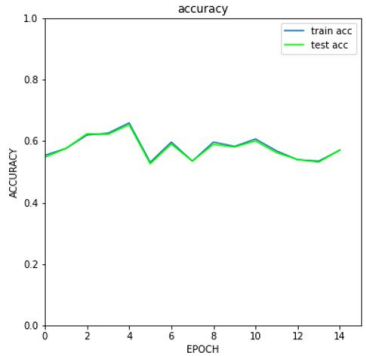
Triplet Loss + HKD Loss





Triplet Loss + HKD Loss + RKD-DA Loss



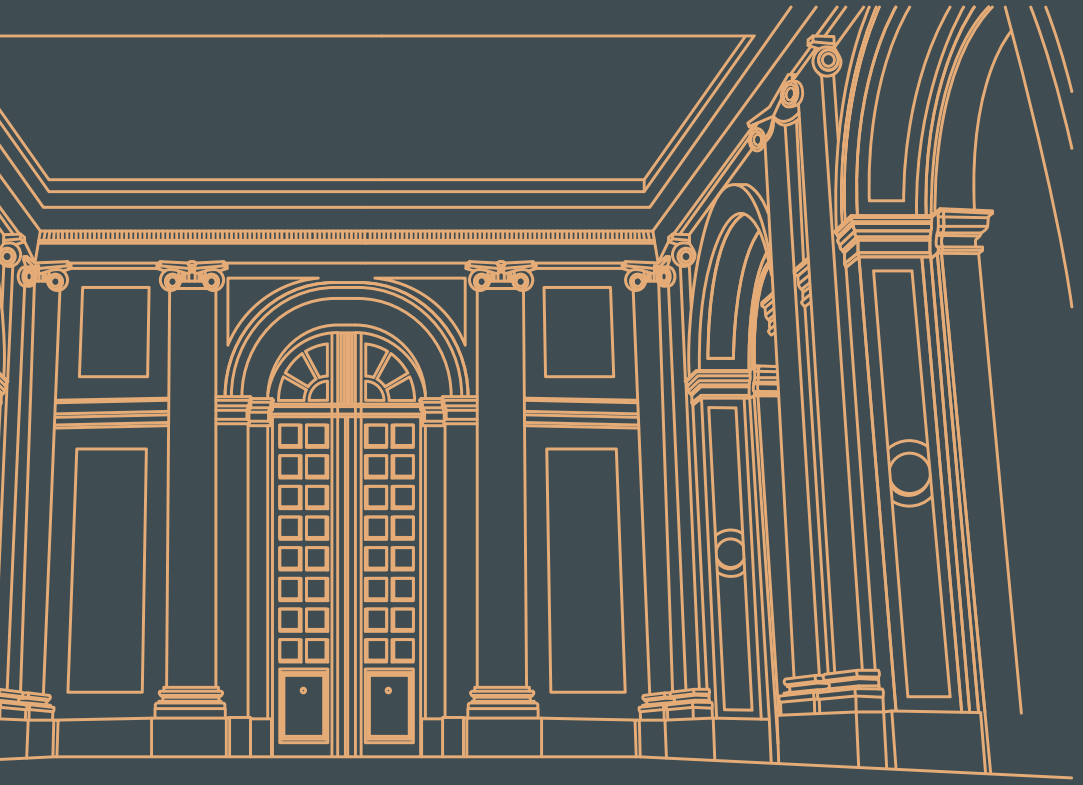


Results

Dataset	Loss Used	Test Loss	Test Accuracy
CIFAR10	Triplet	2.464	0.7979
	Triplet + HKD	2.8216	0.8068
	Triplet + HKD + RKD-DA	15.1264	0.585
MNIST	Triplet	0.2108	0.9481
	Triplet + HKD	0.2331	0.9434
	Triplet + HKD + RKD-DA	13.5119	0.5237
Fashion MNIST	Triplet	0.5199	0.8396
	Triplet + HKD	0.5486	0.831
	Triplet + HKD + RKD-DA	10.0567	0.571

Achievements

- Train Teacher-Student network with Knowledge Distillation over the datasets - CIFAR 10, MNIST, Fashion MNIST.
- Set up 3 different losses for KD - Triplet Loss, Hinton's KL-divergence and Park's Relational KD with both Distance and Angle.
- Set up Early Stopping to Prevent Overfitting.
- Train the Teacher Model and freeze the parameters. Distill the knowledge onto the Student Model using the Distillation Losses implemented.
- Understand the working of Knowledge Distillation in Neural Networks.

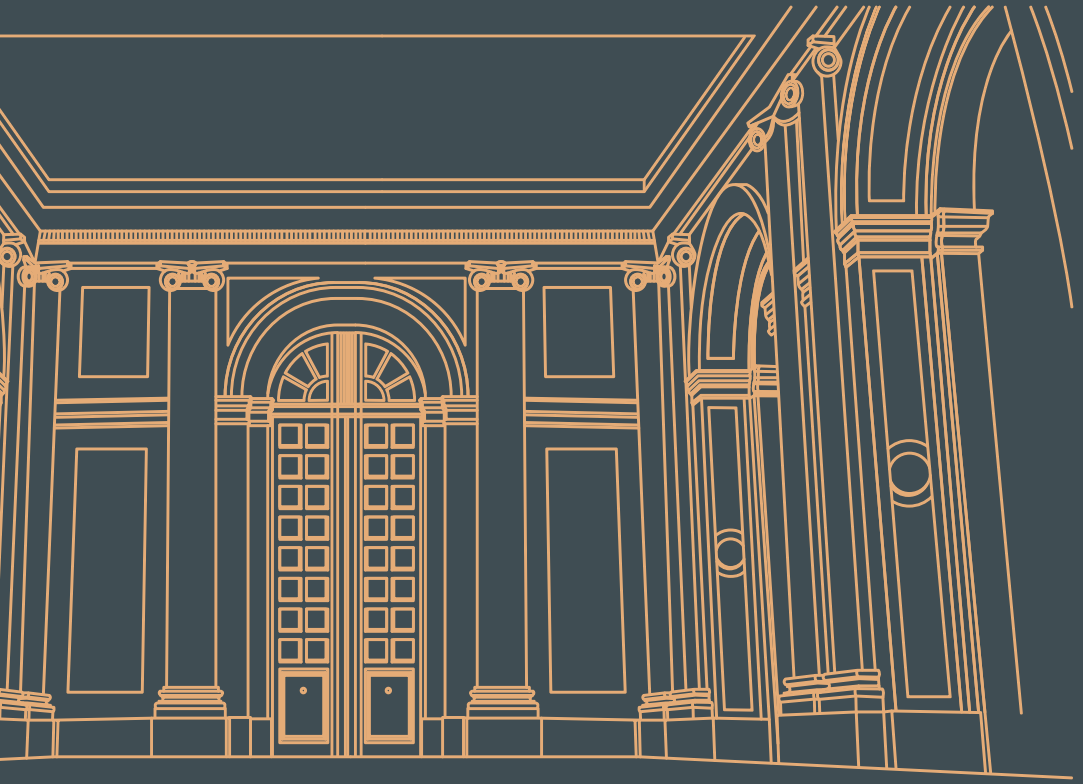


05

Future Steps

Next Steps

- Study the effect of Distillation Losses when the networks are deeper.
- Use hard sampling and select negative samples that are difficult for the model to learn instead of random sampling.
- Explore larger datasets to verify performance of the distillation loss.



06

References

References

1. H. Oki, M. Abe, J. Miyao and T. Kurita, "**Triplet Loss for Knowledge Distillation**," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-7, doi: 10.1109/IJCNN48605.2020.9207148.
2. Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. **Distilling the knowledge in a neural network**. arXiv preprint arXiv:1503.02531, 2015.
3. Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. **Relational knowledge distillation**. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3967–3976, 2019.

Thanks!

...