KNOWLEDGE DISTILLATION

Ruchika Chavhan
GNR638: Course Project

Abstract: This is a detailed report on a set of experiments done in the domain of Knowledge Distillation. Due to the increasing number of model parameters and the hardware required to train them, it is required to decrease the model complexity and hardware by shifting to smaller compact models that provide the same degree of precision in their tasks. Knowledge Distillation is a method to transfer knowledge from the bigger cumbersome model to the smaller compact network. This report contains the experiments on the CIFAR dataset using Knowledge distillation using soft targets and hard targets. Self Distillation is a method in which the network teaches itself using the features extracted by deeper layers as teachers. The report then contains an experiment on semantic segmentation on the CityScapes Dataset using self distillation on an autoencoder model. Results of the self-teaching auto-encoder network have been compared to previous experiments.

INTRODUCTION

Knowledge Distillation is a state-of-the-art way to improve the performance of a network using the knowledge learned by another network. The network learning to improve the performance is called the student network and is much smaller and compact than the teacher network. The teacher network is a much larger network in terms of parameters and model complexity. The whole training procedure is the transfer of knowledge from the expert teacher network and the compact student network.

Knowledge Distillation using Soft Targets: The teacher network is first pre-trained to learn the distribution of the data. The transferring of the generalization ability of the teacher model to a small model can be done by the use of class probabilities produced by the cumbersome model as “soft targets” for training the small model. When the soft targets have high entropy, they provide much more information per training case than hard targets and much less variance in the gradient between training cases, so the small model can often be trained on much less data than the original cumbersome

model while using a much higher learning rate. For the student network to learn, it should receive knowledge from both the expert teacher and the ground truth data. Therefore, the objective should consider both reducing the difference in the distribution of the predictions of the teacher and the student and the ground truth distribution. Therefore, we minimize the KL divergence of the teacher's predictions and the cross-entropy between the student's predictions and the ground truth labels during classification.

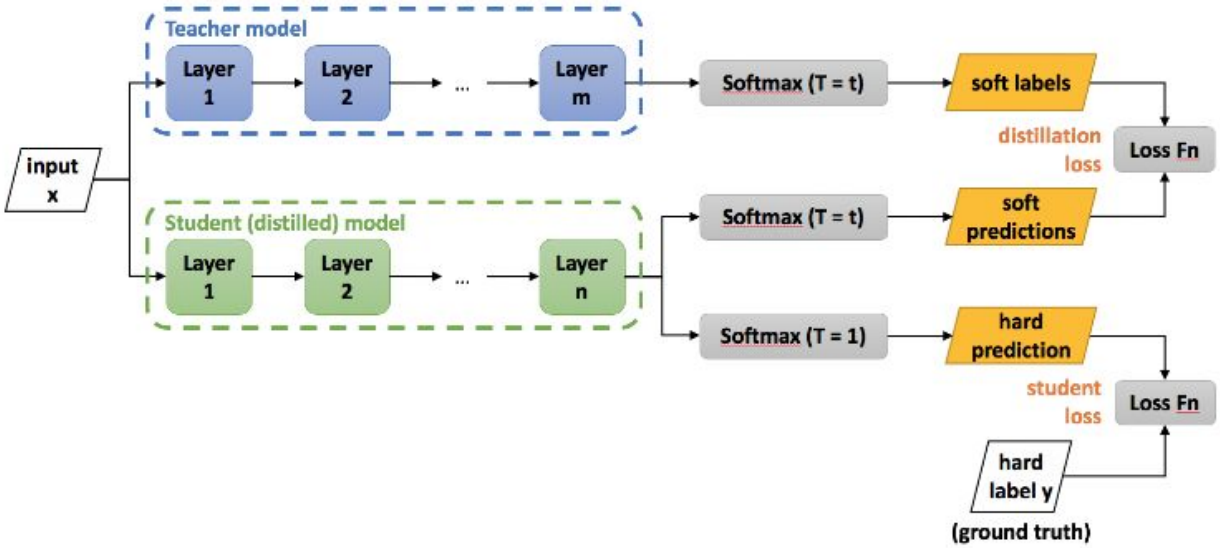


Fig 1: Working principle of Teacher and Student networks sharing knowledge

Self Distillation: Self Distillation involves a network self-teaching itself using the deeper layers of the convolutional network. Since the deeper layers learn more abstract and high-level features, we encourage the relatively shallow layers of the network to learn the features extracted by the deeper layers in order to learn more high-level features of the dataset. Self Distillation was first performed by [], on the task of classification. They designed a network with four ResNet Blocks and a fully connected layer for classification. For distilling knowledge from the deeper layers, the distance between the first three ResNet blocks and the last ResNet block is minimized. To make the sizes of the features maps, the bottleneck of different dimensions are added. The outputs of all the intermediate layers of the network are passed through their respective bottleneck layers followed by fully connected layers.

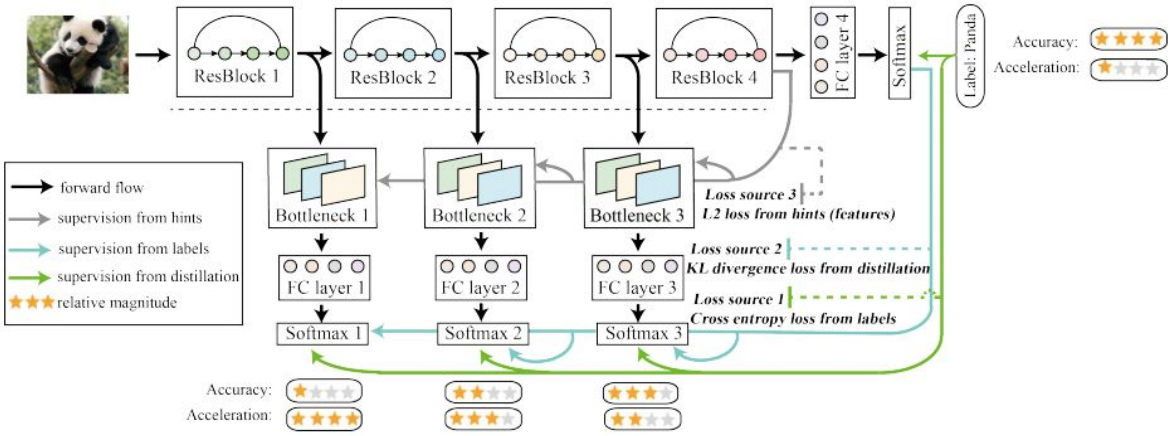


Fig 2: Working Principle of self-distillation for classification

The objective function is as follows:

$$
\begin{aligned}
loss &= \sum_{i}^{C} loss_i \\
&= \sum_{i}^{C} \Big((1-\alpha) \cdot CrossEntropy(q^i, y) \\
&+ \alpha \cdot KL(q^i, q^C) + \lambda \cdot ||F_i - F_C||_2^2 \Big)
\end{aligned}
$$

Here, F_c is the feature map extracted by the fourth ResNet Block. F_i is the feature map extracted by the i^{th} ResNet Block before the fourth block. q^i is the output of the i^{th} fully connected layer. q^C is the output of the fourth fully connected layer.

EXPERIMENTS

Proposed Experiment: This experiment is aimed at improving the performance of an autoencoder network using self distillation. A novel approach at self distillation is considered for the task. There are two domains of images in the CityScapes Dataset, namely X and Y. The task is to convert X to Y and Y to X using two different autoencoders. The network is symmetrical, implying that there is an equal number of encoders and decoders. Intuitively, The features upsampled by the fourth decoder in the autoencoder converting image in X to image in Y are matched with the features

downsampled by the first decoder in the autoencoder converting image in Y to image in X. Subsequently, experiments on extracting knowledge from different layers of both the auto-encoders have been performed and qualitative results have been presented.

Fig 3: Working principle of self distillation in the auto-encoder network.

The objective function minimized is as follows:

$$L = (1 - \alpha)(||O_1 - Y_{true}|| + ||O_2 - X_{true}||) + \alpha(||O_{enc1} - O_{dec4}||)$$

Here, O_1 is the output of the first network (X to Y) and O_2 is the output of the second network (Y to X). O_{enc1} is the output of the first encoder in the first network (X to Y) and O_{dec4} is the output of the fourth decoder in the second network (Y to X).

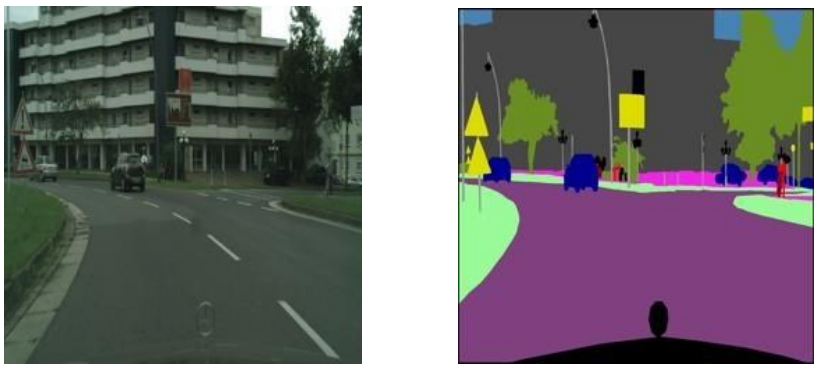


Fig 4: Image from domain X and Y respectively

Different experiments have been performed:

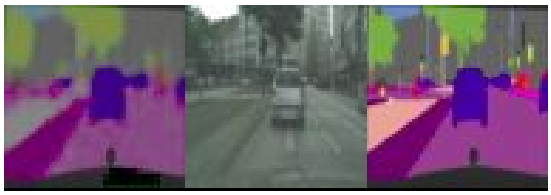
1. Adding connections between 1 and 4
2. Adding connections between 2 and 3
3. Since the shapes of feature maps extracted by the encoder and decoder are different by one row and one column, max pool has been added to reduce size.
4. The max pool replaced by convolution to reduce the size by 1.

RESULTS

The image below is in the order: Predicted output, True image in X domain, True image in domain Y.

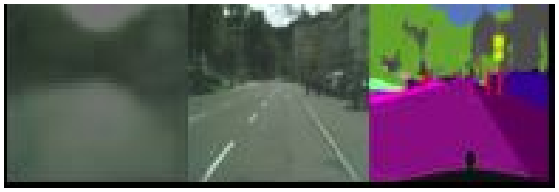
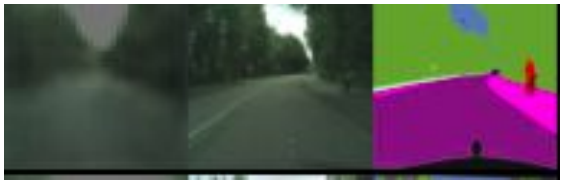
Experiment 1: Adding self distillation between encoder 1 and decoder 4 with max pool to reduce the size of the decoder output by 1.

Conversion from X to Y:




Conversion from Y to X:



Observations:

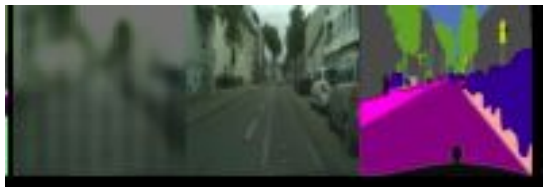
- Conversion from Y to X is not proper.
- The output images are blurry.
- The colors purple and pink are not appearing on the conversion from X to Y.

Experiment 2: Adding self distillation between (encoder 1 and decoder 4) and (encoder 2 and decoder 3) with max pool to reduce the size of the decoder output by 1.

Conversion from X to Y:




Conversion from Y to X:

Observations:

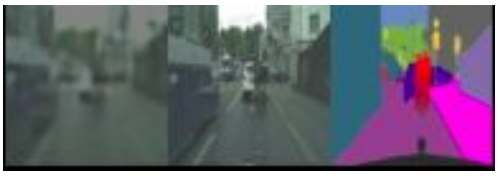
- Conversion from Y to X is not proper.
- The output images are very blurry.
- The colors purple and pink are not appearing on the conversion from X to Y.
- The performance of Y to X autoencoder has decreased compared to the previous model. The images are not only blurry but also grainy.

Experiment 3: Adding self distillation between (encoder 1 and decoder 4) with a convolutional layer for output of decoder 4 to reduce its size by 1. The convolution layer has kernel size, stride, and padding as 2, 1, and 0 respectively.

Conversion from X to Y:




Conversion from Y to X:

Observations:

- The conversion of Y to X has improved. Images are much sharper and more objects are being predicted by the network.
- Conversion of X to Y has also improved as compared to previous models.

CONCLUSIONS

Knowledge Distillation has been applied in traditional autoencoders. By applying self distillation, two networks converting images from one domain to another have been trained jointly where they act as teachers for one another. The autoencoders have been designed to be symmetrical in order to match features within the two networks. Results of different variants of the network and self distillation training setup have been provided and analyzed.