Transfer learning

Mohammed Fayiz, Juntao Jiang, Marfa Zakirova, Zuhaib Khaki, Nazila Eshghi
Tempora pod, NMA 2022

Abstract

Machine learning algorithms often assume that training and test samples contain the same features and belong to the same distribution. However, it happens that it is necessary to build a complex model for a specific task, but there is not enough data for its proper training. Then the application of transfer learning is used.
In our project, we consider the task of classifying images on datasets with different numbers of classes.
Then we observe catastrophic forgetting problem. A neural network loses the information learned in a previous task after training on subsequent tasks.

Introduction

Idea: to use the knowledge gained by other networks on similar tasks.
We can draw from real-world non-technical experiences to understand why transfer learning is possible. Consider an example of two people who want to learn to play the piano. One person has no previous experience playing music, and the other person has extensive music knowledge through playing the guitar. The person with an extensive music background will be able to learn the piano in a more efficient manner by transferring previously learned music knowledge to the task of learning to play the piano [1]. One person is able to take information from a previously learned task and use it in a beneficial way to learn a related task.

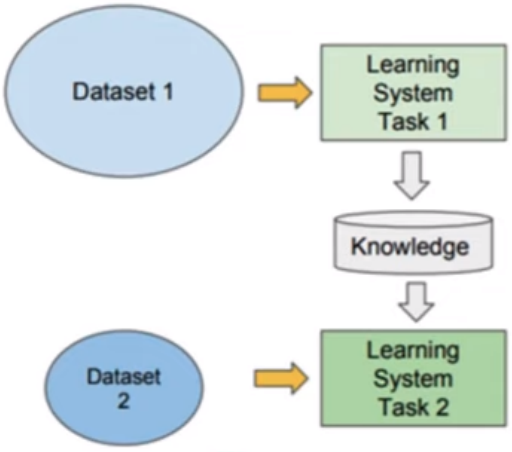
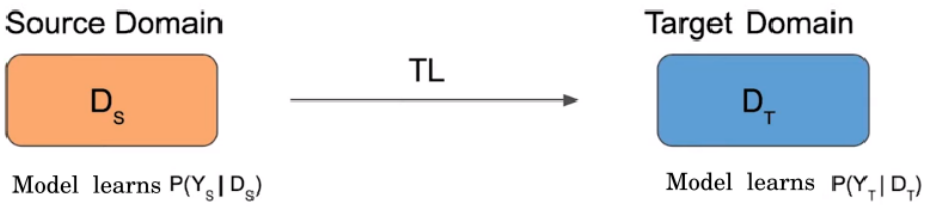


Figure 1: Transfer learning scheme

Types of transfer learning tasks



Source domain, D_S - domain with enough data, net can train there from scratch. The domain from which we will transfer information to another network.

The model, when trained on this domain, must learn the distribution to labels, provided data from this domain, $P(Y_S|D_S)$.

Target domain, D_T - a domain that contains little training data. To train a network on this dataset, you need to use some information from networks that have been trained on the source domain.

The model must also learn the distribution to labels provided this dataset,$P(Y_T|D_T)$.

1. $D_S \neq D_T$
   The feature spaces of the two domains are different.

   Example: Two datasets of texts in two different languages.

2. $P(D_S) \neq P(D_T)$
   The distributions of features in the two domains are different. Such a TL is called Domain Adaptation.

   Example: Dataset review for movies on Netflix and dataset review for mobile apps on Google Play. Both there and there the task is to predict the review score (from 0 to 5)

3. $Y_S \neq Y_T : P(Y_S \mid D_S) \neq P(Y_T \mid D_T)$
   The distributions of label spaces in the two domains are different.

   Example: Dataset of people's faces.
   Task 1: classification of persons by mood Task 2: Classification of persons by race

In our project, just the third case is considered. We consider the task of classifying images on datasets with different numbers of classes: ImageNet, CIFAR - 10.

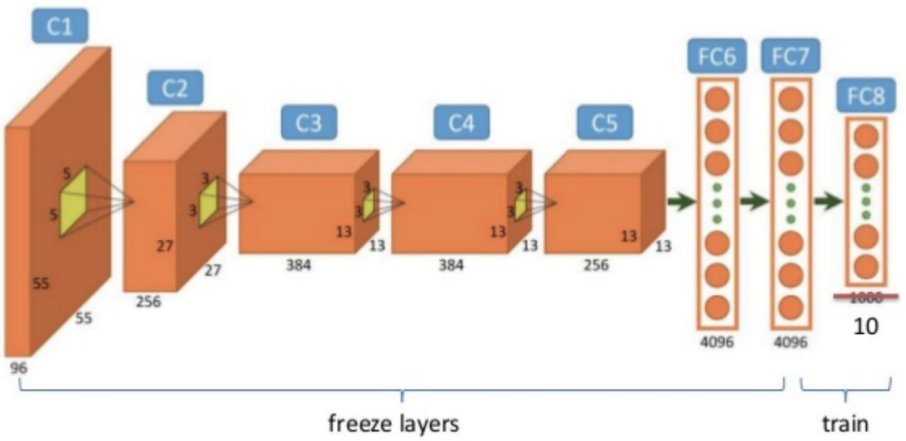
Methods

Ideas for solving TL tasks:

1. Fine - tuning

   (a) Replace the last layer
   Change the layer before the softmax, so as to switch from the output classes of the training domain to the classes of the target domain

   (b) Freezing layers

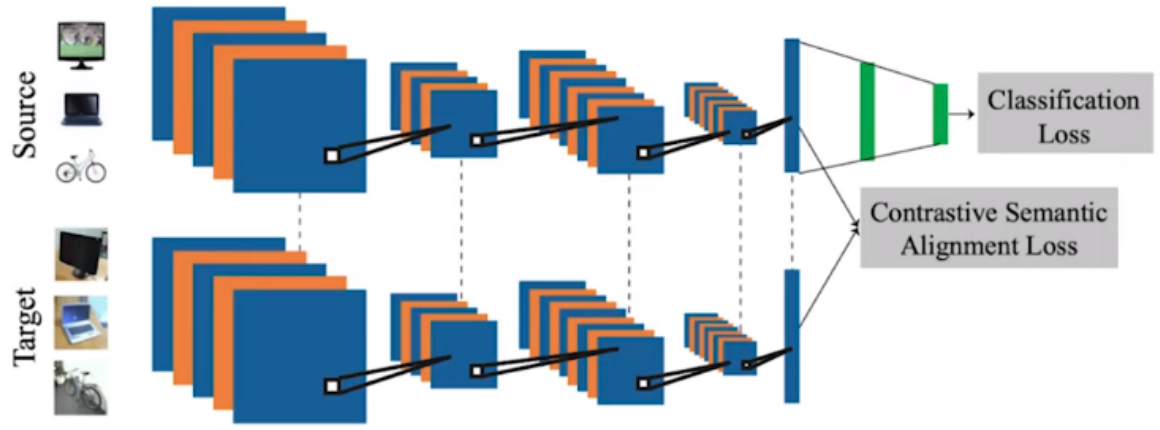


   i. The upper(first) layers of the network are frozen, the lower ones are being retrained. The upper layers highlight low-level information, and they have learned to do it well during retraining. The lower layers extract task-specific information from the information received from the upper layers, so they need to be retrained.
   Thus, when training the network, the layers specific to this task will be trained and the retraining of the network will be minimal (if any)

   ii. How many layers to freeze depends on the difference between datasets and tasks, the complexity of the task and the volume of the dataset for further training.
   One can choose which layers to freeze and which layers to replace. You can freeze the first 10 layers, train the remaining 5, among which 2 will be new layers.

   iii. The more training data in target tasks and the more source and target tasks differ from each other, the more layers need to be retrained: the network does not have to retrain so much when updating more parameters

2. Next idea is suitable for unsupervised TL: no labels on target domain.

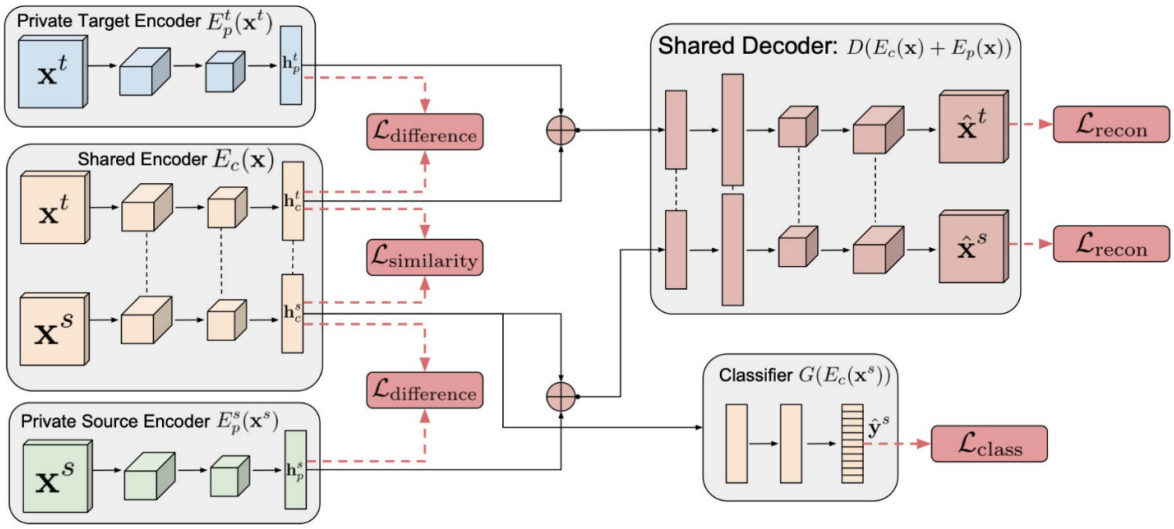


With the help of the loss function, the distribution of the output after some layers of the network for the image from the source and target domains was the same.

In detail, the model is trained on source domain; then a picture is taken from the source domain, another from target, both are run through the network and the outputs of its layers are obtained. Let it turn out that, for example, the outputs of the third layer after the image from the source domain and after the image from the target domain are distributed differently. In order for the model to work well both in the pictures from the source domain and from the target, the distributions must be the same.

One of the ways to achieve this is described here: [2]

3. The network is architecturally divided into 2 parts, one part learns from information specific to a particular domain, the other part of the network learns from information specific to both domains. The necessary training is achieved due to different loss functions. More details here: [3]



Results

For implementation, we use the Resnet-18 network pre-trained on ImageNet. Afterward, we tune it for applying to the target domain, CIFAR – 10; the model also learns the distribution to labels provided with this dataset. We change the number of classes in the pre-trained model's last layer before the softmax, freeze the remaining weights and gradually improve the model: layer by layer from the end.

The result is exhaustive: when all the layers were frozen, the accuracy in training only the last layer reached 77% in only 15 epochs. Furthermore, when defrosting only two layers, the accuracy reaches 97% in 20 more epochs.

Then we observe catastrophic forgetting problem. We train the previously obtained model on CIFAR-10 subclasses and evaluate the accuracy of the final model. Step 1: training and testing of the 1st and 2nd classes. Step 2: training on 3rd and 4th, checking 1,2,3,4. And so on. As predicted, the accuracy is falling.

In this paper, we give an example of qualitative models' extensive learning opportunities on a small number of data. Of course, it is worth considering the consequences of forgetting, which motivates to study specialized models to clarify this drawback.

References

[1] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, 2010.

[2] Motiian, Saeid, et al. "Unified deep supervised domain adaptation and generalization." Proceedings of the IEEE international conference on computer vision. 2017.

[3] Bousmalis, Konstantinos, et al. "Domain separation networks." Advances in neural information processing systems 29 (2016).

Gratitude for the open access to the materials of the Deep Learning School.