Feature Encoding for Image Classification

Haowei Huang
516021910491
Shanghai Jiao Tong University
Shanghai, China
1270927224@qq.com

Zhixin Lin
516021910495
Shanghai Jiao Tong University
Shanghai, China
1069066484@qq.com

Yaojie Ding
516021910430
Shanghai Jiao Tong University
Shanghai, China
416914846@qq.com

***Abstract*—Abstract of Project4.**

I. Introduction

Introduction of Project4.

II. Approaches

In this section, we present our methods applied for classification. And we will emphasize differences of our approaches compared with standard ones.

A. ADDA

Eric Tzeng, et al. summarized a generalized architecture for adversarial domain adaptation and introduced a method of domain adaptation, Adversarial Discriminative Domain Adaptation, in his work [1]. Using their summarization, ADDA is a combination of generative and discriminative neural network model that uses untied weight sharing between source mapping and target mapping and a GAN loss.

The general ADDA approach is presented in figure 1. There are overall four relatively separated subnetworks within the ADDA model:

1) Source encoder network, M_s. Source encoder take source data set as input and output the encoded source features.
2) Target encoder network, M_t. Target encoder take source data set as input and output the encoded target features.
3) Discriminator network, D. Discriminator take encoded source features and encoded target features and tries to identify which come from source dataset and target dataset.
4) Classifier network, C. Classifier network take encoded features, from either source or target domain, as input and output the class prediction.

According to work of Eric Tzeng, et al., we can divide the way the model runs into three stages:

1) Pre-training. In this stage, we feed source training data, X_s for source encoder network and use the output features, $M_s(X_s)$, to feed classifier network and use cross entropy as classification loss , L_c:

$$L_c(X_s, Y_s) = - \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} \sum_{k=1}^{K} \mathbb{I}_{[k=y_s]} \log C(M_s(X_s)) \quad (1)$$

to train the source network and classifier network. After that, both source network and classifier network are fixed. K is the total number of classes.
2) Adversarial adaptation. In this stage, we use the idea of GAN to train M_t to generate features, $M_t(X_t)$, to be similarly distributed as $M_s(X_s)$. We feed X_s and X_t for M_s and M_t respectively and combination of $M_s(X_s)$ and $M_t(X_t)$ for D. We in turn optimize D to minimize L_D:

$$\begin{aligned} L_D(X_s, X_t, M_s, M_t) = \\ - \mathbb{E}_{(x_s)\sim(X_s)} \log D(M_s(X_s)) \\ - \mathbb{E}_{(x_t)\sim(X_t)} \log(1 - D(M_t(X_t)) \end{aligned} \quad (2)$$

and optimize M_t to minimize L_t:

$$L_t(X_s, X_t, D) = -\mathbb{E}_{(x_t)\sim(X_t)} \log D(M_t(X_t)) \quad (3)$$

D tries to distinguish $M_s(X_s)$ and $M_t(X_t)$ while M_t want to deceive D.
Our approach is not always quite standard as Eric Tzeng's work. Except that we use fully connected layers instead of CNN for our X_s and X_t, we also initialized parameters of M_t using pre-trained M_s's. This is not mentioned in the paper and it is not likely to be feasible in most cases. We can do so because our structure of M_s is designed to be the same as M_t. And the measure really help a lot.
3) Testing. In this stage, we feed $M_t(X_t)$ for C and evaluate the classification accuracy.

During our practice, we combined the last two stage into one.

B. DANN

Yaroslav Ganin, et al. proposed a representation learning approach for domain adaptation in their work [2]. DANN can also be viewed as an instance of Eric Tzeng, et al.'s summarization of domain adaptation method. Compared with ADDA, in general, the only difference of DANN is that DANN is an architecture of tied weight sharing. The general architecture is presented in figure 2. Domain classifier G of DANN performs similar function as discriminator D of ADDA does. DANN is similar to ADDA but target encoder and source encoder share the same weight weights. In other words, only one feature encoder, M, is used within DANN. Thus, we do not go to details of components of DANN and just simply

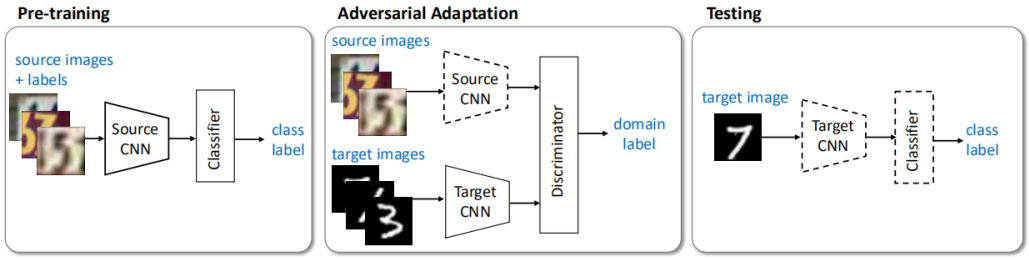

Fig. 1. ADDA Overview: An overview of standard ADDA architecture. Dashed lines indicate fixed network parameters in the indicated stage.

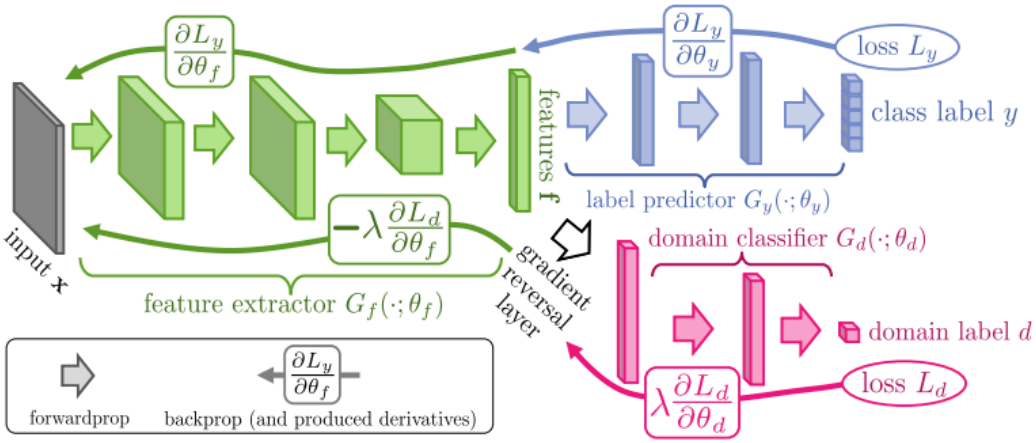

Fig. 2. DANN Overview: An overview of standard DANN architecture.

present the way the DANN model work. We will use similar representations as ADDA to make things easy to understand (although they may not be Yaroslav Ganin et al.'s symbols).

1) Pre-training. In this stage, we optimize M and C to minimize L_C.
2) Adversarial adaptation. In this stage, we feed both X_s and X_t for M. By fixing C, we optimize M and G to minimize L_{da}:

$$\begin{aligned} L_{da}(X_s, X_t, M) = & \\ & -(\mathbb{E}_{(x_s)\sim(X_s)} \log D(M(X_s)) \\ & + \mathbb{E}_{(x_t)\sim(X_t)} \log(1 - D(M(X_t))) \times c_{da} \\ & - \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} \sum_{k=1}^{K} \mathbb{I}_{[k=y_s]} \log C(M(X_s)) \end{aligned} \tag{4}$$

   Unlike ADDA, parameters of M and G are updated at the same time. And the loss, L_{da} of DANN is combination of both L_C and L_D of ADDA. There is subtle difference between ADDA and DANN in this stage. C_{da} is a balance factor for classification loss and discriminator loss. c_{da} is not mentioned in the paper, but we use it and find it helps improve our model.
3) Testing. In this stage, we feed $M(X_t)$ for C and evaluate the classification accuracy.

Also, we combined stage of adversarial adaptation and testing together.

III. Experiments And Results

Experiments.

IV. Discussions

Discussions.

V. Conclusion

Conclusions.

References

Please number citations consecutively within brackets [**?**]. The sentence punctuation follows the bracket [**?**].

References

[1] Eric Tzeng, Judy Hoffman, Kate Saenko and Trevor Darrel, "Adversarial Discriminative Domain Adaptation".
[2] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Francois Laviolette, Mario Marchand, Victor Lempitsky, "Domain-Adversarial Training of Neural Networks".

IEEE conference templates contain guidance text for composing and formatting