Report on "Domain-Adversarial Training of Neural Networks" by Ganin et al

Ronak Dedhiya Dinesh (ronakdedhiya@iisc.ac.in; SR No: 06-18-05-19-52-21-1-20116)
Prashanth N Bhat(prashanthbn@iisc.ac.in; SR No: 04-03-06-19-52-21-1-20082)

September 28, 2024

1 Introduction

The paper [1] proposes an interesting approach to domain adaptation.

To improve domain adaptation tasks, the paper proposes a new representation learning approach. In a typical learning scenario, the data for train and test samples come from same distribution. The authors propose that we can have the datasets for training and test can come from similar but slightly different distributions. This leads to a Neural Network that can (i) Efficiently learn to be discriminative for the main task at hand (ii) Be indiscriminate to shifts in domains.

It proposes the addition of a few standard layer and a new gradient reversal layer to a feed forward neural network. These new layers are trained by back propagation and stochastic gradient descent. These kinds of neural networks are called Domain Adaptive Neural Network. (DANN).

The two datasets described here are the source and the target dataset. To train the DANN, class labels from the source dataset is used. The target dataset is also used without its class labels. An additional label is used with the dataset to indicate whether it is from source or target.

2 DANN architecture

The Neural network architecture uses a gradient reversal layer as shown in Figure 1 (Figure copied from [1]).

The layers in green perform the task of feature extraction and forms the base of the neural network $\mathcal{G}_f$. The layers in blue form a label predictor $\mathcal{G}_y$ and can be thought of as the head of the neural network which take the feature vectors from the base and assign a class label to it.

The layers in red are a new addition in DANN $\mathcal{G}_d$. The output of these layers is a domain label. It is connected to the base and the head through a gradient reversal layer(GRL) reverses the gradient to make the feature extractor immune to changes in domain.

2.1 Training the DANN

1. DANN training uses both the source and target datasets.

2. The label predictor is trained using only the source dataset along with class labels.

3. The domain classifier uses both the source and the target dataset. It is supposed to predict the domain the dataset belongs to.

3 Gradient reversal layer

The GRL is inserted between the feature extractor G_f and the domain classifier G_d. During backpropogation, the partial derivative of the loss downstream of GRL, $\mathcal{L}_d$, with respect to upstream layer parameters, θ_f, gets multiplied by -1. Mathematically, GRL represents a $pseudo - function$ $\mathcal{R}(x)$ whose forward behaviour is represented as

$$\mathcal{R}(x) = x \tag{1}$$

and backpropogation behaviour is represented as

$$\frac{d\mathcal{R}(x)}{dx} = -\mathbb{I} \tag{2}$$

where $\mathbb{I}$ is the identity matrix.

Then the objective of the DANN can be defined as a combination of loss of the classifier and the domain predictor. It can be defined mathematically as:

$$\begin{aligned}\mathbb{E}(\theta_f, \theta_y, \theta_d) = &\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y(\mathcal{G}_y(\mathcal{G}_f(x_i;\theta_f);\theta_y), y_i) \\ &-\lambda(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y(\mathcal{G}_y(\mathcal{G}_f(x_i;\theta_f);\theta_y), d_i) \\ &+\frac{1}{n'}\sum_{i=n+1}^{N}\mathcal{L}_y(\mathcal{G}_y(\mathcal{G}_f(x_i;\theta_f);\theta_y), d_i))\end{aligned} \tag{3}$$

Here λ is a hyperparameter that can be optimized for best performance.

This objective function can be optimized using Stochastic Gradient Descent. The resultant network learns features that are domain-invariant and discriminative at the same time.

4 Conclusion

The paper presents a simple method to add domain adaptation to any Neural Network Architecture that uses back propagation for training. It is not limited to classification tasks, it can be extended to other feed forward neural networks such as descriptor learning for person re-identification.

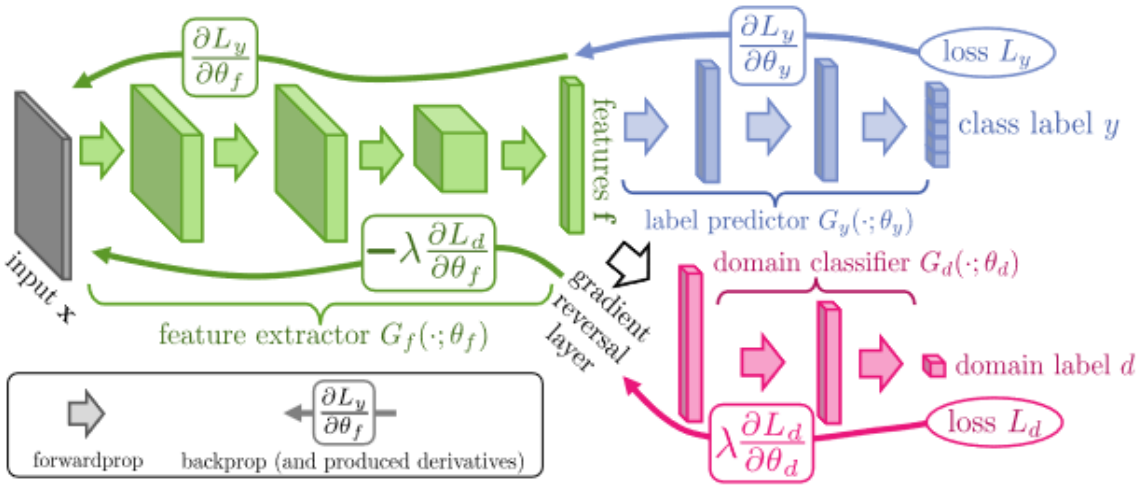

Figure 1: DANN architecture. Image copied from [1]

This method achieves state of the art domain adaptation performance on standard benchmarks.

References

[1] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. 2015.