Report on Learning Deep Features for One-Class Classification

Valliappan

1 Introduction

1.1 Problem Statement

- The paper address the problems similar to one shot learning, where they just have information/data of one class.
- Instances of only a single object class are available during training.
- In the context of this paper, all other classes except the class given for training is called alien classes. This can be connected to abnormality classsification.
- The aim of the paper is to improvise the results of abnormality classification, by introducing novel loss function.

1.2 Conventional Approach

- The feature extraction network from a pre-trained model is used. Only a new classification (Fully connected) network is trained/fine tuned in the case of low training samples.

1.3 Proposed Approach

- Strategy used in this paper uses deep-features extracted from a pre-trained model g_s (refer figure-1), where training is carried out on a different dataset (complement of the one class instance), to perform one class classification.
- The paper proposes a fine tuning framework g_l which produces deep features specialized for one class classification.
- Classification network for "one-class" is defined as h_c. (refer figure-1)
- Based on the output of the classification networks h_c, two loss function is defined.
- The loss function defiined will helps the architecture fine tune so that it can differentiate the one class instance from the alien classes

1.4 Key Idea

- The two loss function are compactness (l_C) and descriptiveness loss(l_D).

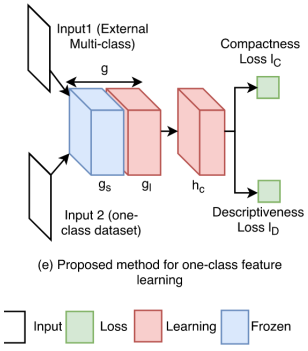

Figure 1: Proposed one-class classification network

2 Network Architecture

2.1 Notation

- g_s - pre-trained weigths
- g_l - fine tuning network
- h_c - classification network
- l_C - Compactness loss
- l_D - Descriptiveness loss
- R - Reference network (Alexnet)
- r - Reference Dataset (Imagenet)
- S - Secondary Network (similar to R)
- t - Target Dataset
- W - Model weights initialised with pretrained weights

2.2 Loss function

Compactness Loss: A desired quality of a feature is to have a similar feature representation for different images of the same class. Hence, a collection of features extracted from a set of images of a given class will be compactly placed in the feature space. This quality is desired even in features used for multi-class classification. In such cases, compactness is quantified using the intra-class distance; a compact representation would have a lower intra-class distance.

Forward pass: This paper considers the Euclidian distance as the measure to compute the compactness loss. So lets consider a batch of N samples, represented as $X = \{x_1, x_2, ..., x_n\}$. So for every i^{th} sample in X where $1 \leq i \leq n$. Each x_i is of the dimension $1 \times k$, which means $x_i = \{x_{i1}, x_{i2}, ..., x_{ik}\}$. Hence l_C is defined as average euclidian distance for a batch of n in Eq-3

$$Z_i = m_i - x_i \tag{1}$$

$$m_i = \frac{1}{n-1} \sum_{j \neq i} x_j \qquad 1 \leq j \leq n \tag{2}$$

$$l_C = \frac{1}{nk} \sum_{j \neq i} z_i^T z_i \qquad 1 \leq j \leq n \tag{3}$$

Backpropogation: It is the tricky part, but easy to understand from the appendix of the paper[1]. Eq-3 can be re-written as Eq-4. We need to differentiate the loss w.r.t x_{ij}, which is represented in Eq-5.

$$l_C = \frac{1}{nk} \sum_{i=1}^{i=n} \sum_{l=1}^{l=k} (x_{il} - m_{il})^2 \tag{4}$$

$$\frac{dl_C}{dx_{ij}} = \frac{2}{nk} \sum_{l=1}^{l=k} (x_{il} - m_{il}) \times \frac{d(x_{il} - m_{il})}{dx_{ij}} \tag{5}$$

Remember $\frac{d(x_{il} - m_{il})}{dx_{ij}}$ when $j = l$ it is 1 and rest all values of j it's $\frac{-1}{(n-1)}$ (m_i is shown in Eq-2). After substituting this in Eq-5.

$$\frac{dl_C}{dx_{ij}} = \frac{2}{nk} \Big[(x_{ij} - m_{ij}) - \frac{1}{n-1} \sum_{l \neq k} (x_{il} - m_{il}) \Big] \tag{6}$$

$$\frac{dl_C}{dx_{ij}} = \frac{2}{nk} \Big[(x_{ij} - m_{ij}) + \frac{1}{n-1} (x_{ij} - m_{ij}) \ + \ (-\frac{1}{n-1} (x_{ij} - m_{ij}) - \frac{1}{n-1} \sum_{l \neq k} (x_{il} - m_{il})) \Big] \tag{7}$$

$$\frac{dl_C}{dx_{ij}} = \frac{2}{nk(n-1)} \Big[\frac{n}{n-1} (x_{ij} - m_{ij}) - \frac{1}{n-1} \sum_{l=0}^{l=k} (x_{il} - m_{il}) \Big] \tag{8}$$

$$\frac{dl_C}{dx_{ij}} = \frac{2}{nk(n-1)}\left[n(x_{ij} - m_{ij}) - \sum_{l=0}^{l=k}(x_{il} - m_{il})\right] \tag{9}$$

Descriptiveness loss: The given feature should produce distinct representations for images of different classes. Ideally, each class will have a distinct feature representation from each other. Descriptiveness in the feature is also a desired quality in multi-class classification. There, a descriptive feature would have large inter-class distance.

To classify the one class instance properly, it is important to satisfy both these character collectively. The optimisation problem statement is as follows:

$$loss = \max_t(D(g(t)) + \lambda C(g(t))) \tag{10}$$

where t is the training data, g is feature representation and λ is a positive constant.

In this work, the variance of the feature distribution approximate it to the variance of each feature batch. This quantity as the compactness loss (l_C). On the other hand, descriptiveness of the learned feature cannot be assessed using a single class training data. However, if there exists a reference dataset with multiple classes, even with random object classes unrelated to the problem at hand, it can beused to assess the descriptiveness of the engineered feature. The learned feature to perform classification on an external multi-class dataset, and consider classification loss there as an indicator of the descriptiveness of the learned feature. So, the paper use cross-entropy loss calculated in this fashion as the descriptiveness loss (l_D)

So with this formulation, the optimization function can be writen as :

$$loss = \min_g(l_D(r) + \lambda l_C(t)) \tag{11}$$

Here, the r is the reference data and the optimization works with the hyper parameter as g_t the learning network.

2.3 Architecture

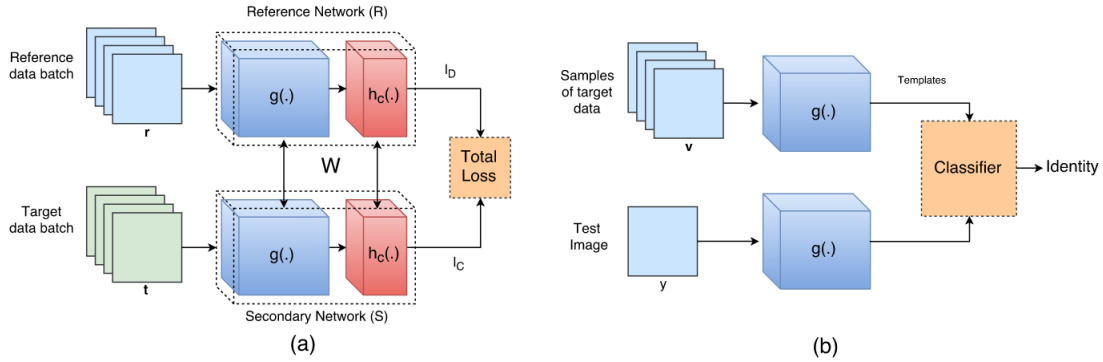

Figure 2: Network Architecture (a) training frame work (b) Testing frame work

The proposed training architecture is shown in Figure-2a. The architecture consists of two CNNs, the reference network (R) and the secondary network (S) as introduced in the previous sub-section. Here, weights of the reference network and secondary network are tied across each corresponding counterparts.

2.4 Training

Both the reference (R) and secondary (S) net is initialized with the same weights. Just that the loss functions are different for each network. During the training phase we consider 2 set of images per batch, i.e reference batch and target batch. So for a target batch the loss function is l_C and refrence batch bach the loss is l_D. The weights are fine tunned based on the loss function for each network.

Hence its just one network, that has separate loss function based on the training data, which I felt was denoted in the paper, in an unusual way.

The λ in eq-11 determines how well both model is trained, for a both model to be trained which can discriminate the abnormal class, should have more weightage to the l_D hence the $\lambda = 0.1$. So the λ governs the importance of each loss, our aim is to have good discriminative features hence the λ is set to 0.1.

2.5 Testing

The test phase is divided into 2-phases. **Template generations** and **matching** phase. To be simple, **Template generation** is generating features from $g(.)$ on the sample from the training data. So these features act as the representation of the how the one-class features will look like. These features are used for learning the classifier. Now for the test set of image, features are extracted from the $g(.)$ fine tuned network. These features can be used to predict from the model trained from the **template generation** phase

The classifier used to learn the templates can be a simple SVM, KNN.

3 Reference

1. Perera, Pramuditha, and Vishal M. Patel. "Learning Deep Features for One-Class Classification." arXiv preprint arXiv:1801.05365 (2018).