Domain Specific Transfer Learning Using Image Mixing and Stochastic Image Selection

Sheldon Coup[1], Varvara Vetrova[1], Eibe Frank[2], Rachael Tappenden[1]
[1] *School of Mathematics and Statistics, University of Canterbury*
[2] *Department of Computer Science, University of Waikato*
New Zealand
sheldon.coup@pg.canterbury.ac.nz, varvara.vetrova@canterbury.ac.nz
eibe@cs.waikato.ac.nz, rachael.tappenden@canterbury.ac.nz

Abstract

Can a gradual transition from the source to the target dataset improve knowledge transfer when fine-tuning a convolutional neural network to a new domain? Can we use training examples from general image datasets to improve classification on fine-grained datasets? We present two image similarity metrics and two methods for progressively transitioning from the source dataset to the target dataset when fine-tuning to a new domain. Preliminary results, using the Flowers 102 dataset, show that the first proposed method, stochastic domain subset training, gives an improvement in classification accuracy compared to standard fine-tuning, for one of the two similarity metrics. However, the second method, continuous domain subset training, results in a reduction in classification performance.

1. Introduction

For the purpose of training a convolutional neural network (CNN) classifier to perform on a new *target* domain, a standard approach is to fine-tune a CNN that has been pre-trained on some *source dataset* of general images. This means that the transition from source to target dataset is abrupt, with the relationship between the target and source classes not taken into consideration. In this paper, we investigate whether a progressive transition aids the training process when fine-tuning from the source dataset to the target dataset. Our methods associate classes in the source dataset with classes in the target dataset. The aim of the progressive transition is to perform source class to target class knowledge transfer. To enable such a transition, we propose two metrics for evaluating the visual similarity of source and target classes.

The type of transfer learning we consider in this paper concerns the use of knowledge gained about a task from one domain to perform that task on a new domain. [1] proposed a method of transfer learning where the CNN is first pre-trained on a selected subset of the source dataset. This subset consists of the source images that are found to be most similar to the target domain. [1] found that this method outperforms classifiers that were fine-tuned from an ImageNet initialization. [1] approximate domain similarity using Earth Mover's Distance ([5]). [2] proposes a method in which a CNN is jointly optimized to perform classification on two different tasks. One task is to classify the target data and the other is to classify a subset of the source data. [2] shows that this can improve classification accuracy on small datasets by 2% - 10% when compared to using standard fine-tuning.

The first approach we consider in this paper performs image mixing to enable a gradual transition from source to target. This is inspired by recent regularization and data augmentation methods ([7] [9] [4]) that apply image mixing, thus indicating that CNNs can learn effectively from image mixtures. Firstly, we propose two methods for measuring the similarity of two image classes. This is followed by an outline of a *dataset pairing*, a relationship between the source and target classes that utilizes the class similarity measures. Next, Section 3 introduces our transfer learning methods, which enable a progressive transition between source and target dataset, and provides a description of the experimental setup used for evaluation. In Section 4, the experimental results are given. Finally, in Section 5, conclusions are drawn and future work is proposed.

2. Class Similarity Metrics and Dataset Pairings

We propose two algorithms for calculating the similarity of two image classes. In what follows, a pre-trained CNN feature extractor $\mathcal{C} : \mathbb{R}^m \to \mathbb{R}^n$ is considered to be a continuous function from the image space to the output feature space, i.e., $\mathcal{C}(a)$ is the feature embedding of an image a. In practice, this is a CNN with the final dense layers removed and the output of the final convolutional layer being average-pooled to create a feature vector representation for each input image.

2.1. Average Location

The *Average Location* (AL) algorithm measures class similarity by calculating the distance between class centroids in the convolutional feature space. Given a set of images $\mathcal{A}$ and a convolutional neural network $C : \mathbb{R}^m \to \mathbb{R}^n$, the *centroid* of $\mathcal{A}$ is defined as,

$$\bar{\mathcal{A}} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} C(a). \tag{1}$$

Therefore, given two sets of images $\mathcal{A}, \mathcal{B}$, the AL distance between these sets is defined as follows:

$$AL(\mathcal{A}, \mathcal{B}) = d(\bar{\mathcal{A}}, \bar{\mathcal{B}}) \tag{2}$$

where d is the Euclidean distance.

2.2. Maximum of Distance Curve

The *maximum of distance curve* (MDC) algorithm is our second method of calculating the similarity of two image sets. Given two sets of images, $\mathcal{A}, \mathcal{B}$, and a pre-trained CNN C, we first calculate the centroid of each set, $\bar{\mathcal{A}}, \bar{\mathcal{B}}$, as defined by Equation (1). Next, we find the image in each set whose corresponding feature embedding is closest to the set centroid. These images are called the set ”representatives”. Formally, the set representative a_r for the set $\mathcal{A}$ is,

$$a_r = \underset{a \in \mathcal{A}}{\arg\min}\, d(\bar{\mathcal{A}}, C(a)) \tag{3}$$

where d is the Euclidean distance. Using these set representatives, the pre-trained CNN $\mathcal{C}$ and some integer k we calculate the MDC distance using Algorithm 1.

2.3. Dataset Pairings

A *pairing* $\mathcal{P}$ is an injective mapping from the target classes to a set of the source classes. Pairings are used in our transfer learning methods to exchange target images for appropriate source images during training. Pairings can be found by creating a *inter-class distance matrix* M where the i, jth entry $m_{i,j}$, represents the dissimilarity between the ith target class and the jth source class. Using the Hungarian algorithm [3] provides a set of source/target class pairs that minimises the sum of the inter-class distances. Using these pairs of classes as an injective mapping defines a *minimum cost pairing*.

Algorithm 1 Maximum of Distance Curve algorithm, for approximating the maximum value of the distance curve between two image sets.

Input: C is a CNN with pre-trained weights, a, b are the class representatives of two image sets and k is non-negative.

Output: y_{max} Maximum value of distance curve

```
procedure MDC($C, a, b, k$)
    $a' \leftarrow C(a)$
    $b' \leftarrow C(b)$
    $\delta_0 \leftarrow d(a', b')$
    $i \leftarrow 0$
    $y_{\max} \leftarrow 0$
    while $i \leq k$ do
        $\lambda \leftarrow i/k$
        $c \leftarrow (1 - \lambda)a + \lambda b$
        $c' \leftarrow C(c)$
        $\delta_a \leftarrow d(a', c')$
        $\delta_b \leftarrow d(b', c')$
        $x \leftarrow (\delta_a^2 - \delta_b^2 + \delta_0^2)/(2\delta_0)$
        $y \leftarrow \sqrt{\delta_a^2 - x^2}$
        if $y > y_{\max}$ then
            $y_{\max} \leftarrow y$
        $i \leftarrow i + 1$
    return $y_{\max}$
```

3. Transfer Learning Methods

[1] proposes a method of transfer learning whereby a subset of the source dataset that is similar to the target dataset is selected and a CNN is trained on it prior to being trained on the target dataset. Thus, the overall training process is split into two distinct steps, and the transition between the source and target datasets is immediate between the steps. Here, we propose two methods of transfer learning, in which the transition between the source and target data is a gradual process. Continuous Domain Subset Training (CDST) uses image mixing to transition from source to target data. Stochastic Domain Subset Training (SDST) starts by training on source data and progressively increases the number of target images appearing in each batch until only target images are being used.

3.1. Continuous Domain Subset Training

CDST performs training using convex combinations of source and target images, mixed in a ratio λ that is determined by how many training epochs have occurred. Assume that we have a source domain $\mathcal{S}$, a CNN $\mathcal{C}$ that has been pre-trained on $\mathcal{S}$ and a target domain $\mathcal{T}$. Additionally, assume that we have a pairing $\mathcal{P}$ between the classes of $\mathcal{T}$ and $\mathcal{S}$. Given a target image t, the composite input image c is calculated using,

$$c = (1 - \lambda)s + \lambda t \tag{4}$$

where s is a random source image such that $\text{class}(s) = \mathcal{P}(\text{class}(t))$,

$$\lambda = \max\left\{0, \min\left\{\frac{e_{\text{current}}}{e_{\text{total}}} + \beta, 1\right\}\right\}, \tag{5}$$

e_{current} is the current epoch number, e_{total} is a hyperparameter determining the number of epochs the transition between the source and target domains requires, and $\beta \sim \mathcal{N}(0, 0.1)$ is a normally distributed random variable.

The pseudo-code for this method for each epoch is as follows:

1. For each target image $t \in \mathcal{T}$, calculate the composite image c using Equations (4) and (5). Note that β is resampled for each image in the dataset, so the mixing ratio λ may be different for each image.

2. Train the network $\mathcal{C}$ on each of the composite images.

3. Increment e_{current} by one until $e_{\text{current}} = e_{\text{total}}$, then train as normal on the target dataset $\mathcal{T}$.

3.2. Stochastic Domain Subset Training

Assume that we have a source domain $\mathcal{S}$, a CNN $\mathcal{C}$ that has been pre-trained on $\mathcal{S}$ and a target domain $\mathcal{T}$. Additionally, assume that we have a pairing $\mathcal{P}$ between $\mathcal{T}$ and $\mathcal{S}$.

We can write the pseudo-code for this method for each epoch as follows:

1. For each target image $t \in \mathcal{T}$, sample $\gamma \sim \mathcal{U}(0, 1)$. If $\gamma > \frac{e_{\text{current}}}{e_{\text{total}}}$ then substitute the target image t for a random source image $s \in \mathcal{S}$ such that $\text{class}(s) = \mathcal{P}(\text{class}(t))$.

2. Train the network $\mathcal{C}$ on the mixture of target and source images.

3. Increment e_{current} by one until $e_{\text{current}} = e_{\text{total}}$, then train as normal on the target dataset $\mathcal{T}$.

This means that the CNN is being trained primarily on source images in early stages of training. As training progresses, the number of target images per epoch increases until e_{total} training epochs have elapsed, by which point only target images are being trained on.

3.3. Experiments

The performance of the proposed transfer learning methods was evaluated using the Flowers 102 dataset as the target dataset. Training was done using only the 1020 image training set, the validation set was not used for training. The source dataset used for all experiments is the 1000 class ImageNet/ILSVRC2012 dataset. Pairings were calculated using both the AL and MDC metrics. An InceptionV3 network [6], pre-trained on ImageNet, was used as $\mathcal{C}$ for feature extraction for the AL and MDC algorithms. For each proposed transfer learning method, the pre-trained InceptionV3 network was fine tuned on each generated pairing.

For CDST and SDST, once the number of mixing epochs e_{total} has been reached, the learning rate of the optimizer is reset to its initial value and the CNN is then trained using only the target data $\mathcal{T}$ for 50 epochs. Additionally, we have also included a delay of 10 before the transition starts during which training only uses source dataset images. For these experiments $e_{\text{total}} = 90$. The Adadelta optimizer [8] with default hyperparameter values was used for all experiments.

4. Results

The final test accuracies for the Flowers 102 can be found in Table 1. The MDC metric does not appear to be able to be approximating image feature similarity as well as the AL metric. Using the pairing calculated with the AL metric in conjunction with the SDST method results in a 0.8 % increase in accuracy when compared to training using plain fine-tuning. However, use of the CDST method shows a drop in classification accuracy when compared to normal fine-tuning.

	AL Pairing	MDC Pairing
CDST	75.439	73.844
SDST	89.415	81.152
Baseline	88.648	

Table 1. Final test accuracies on the Flowers 102 dataset using CDST, SDST and normal fine-tuning.

5. Conclusions

We have proposed two methods for performing transfer learning from a known domain to a new domain by performing gradual transition from the training data of the source domain to the training data of the target domain. To enable this transition, classes from the source and target domain are paired using appropriate similarity metrics. Our two transfer learning methods, CDST and SDST, were evaluated on the Flowers 102 dataset in conjunction with two metrics, AL and MDC, for measuring the similarity of classes. In our preliminary results, the SDST method performs significantly better than CDST using both the AL and MDC class pairings. The pairing produced when measuring class similarity using the AL metric resulted in better performing classification models than the MDC metric. This indicates that the AL metric might be more suitable for measuring the similarity of two image classes. Additionally, the AL metric is less computationally expensive to calculate than the MDC metric.

References

[1] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4109–4118, 2018. 1, 2

[2] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1086–1095, 2017. 1

[3] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2

[4] Shigeru Maya and Ken Ueno. Dadil: Data augmentation for domain-invariant learning. 2016. 1

[5] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000. 1

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 3

[7] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018. 1

[8] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 3

[9] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1