2016 International Conference on Applied Mathematics and Mechanics (ICAMM 2016)
ISBN: 978-1-60595-399-1

Adapting Convolutional Neural Network to Multi-label Image Classification

Xiao-Bo JIN[1,*], Sheng WANG[2], Guicai WANG[1], Junwei YU[1] and Feng WANG[1]

[1]Henan University of Technology, China

[2]Chengdu Economic Information Center, China

*Corresponding author

Keywords: Multi-label, Pre-training, Fine-tuning, CNN, Image classification.

Abstract. Previous Multi-label image classification is largely limited by the representation power of the hand-crafted features. The convolutional neural network (CNN) has achieved successes in many computer vision tasks. In this work, we adapt the CNN to the multi-label image classification, where three approaches are used including end-to-end training on the target dataset, pre-training on Image Net and fine-tuning on the target dataset, CNN features extracted from Image Net for the AdaBoost.MH classifier. The experimental results on two datasets show that CNN model can boost large margin on the object dataset in contrast with the hand-crafted features methods, which achieves at most 98% on MSRC compared to 92% by the state-of-art algorithms, but benefit a little in the scene dataset. The primary discrepancies between the object and the scene classification tasks lies in that the former need to only focus on the foreground part of the image but the latter requires paying attention to the entire image. Source code upon Torch7 toolkit to reproduce the experiments in the paper is made publicly available.

Introduction

Many real-world applications can be formulated as the multi-label classification where some instances have more than one labels or no labels. For example, a document can belong to multiple topics in the text categorization [10]. Multi-label algorithm can be categorized as problem transformation and algorithm adaptation [12]. The algorithm adaptation methods often adapt the multi-class version of the algorithm to handle the one-vs-rest binary problem at one time including AdBoost, neural networks, support vector machine(SVM) and the nearest-neighbor classifier [12]. AdaBoost.MH [9] and AdaBoost.MR [10] assign to each pair of the instance-label a weight and learn multiple binary classifiers in one round, which can take advantage of zero threshold without the post-processing step.

Convolutional Neural Networks (CNNs) have achieved the big success in various visual tasks such as image classification [6], object detection [3] and image segmentation [8]. Specially, CNN obtained the state-of-the-art performance with 10% gain over the previous methods based on handcrafted features in the large-scale single-label object recognition dataset ImageNet [5] with more than one million images from 1,000 object categories. One of the greatest advantages of convolutional neural networks is that they are trained in an end-to-end fashion, thus removing the need for manual feature engineering and greatly reducing the need for adapting to new tasks and domains. In the multi-label image classification, a image can be categorized as multiple semantic categories [1]. Cabral et al. [2] solved the multi-label image classification problems with matrix completion. Li et al. [7] used informative label combination pairs to augment the original labels to enhance the individual label prediction. However, these multilabel image classification methods are largely limited by the representation power of the hand-crafted features. In this paper, we will focus on the multi-label image classification with CNN and adapt its single-label version to handle the multi-label classification tasks.

In our work, we adopt the following three fashions to solve the multi-label image classification under CNN: (1) The end-to-end training on the target dataset; (2) CNN pre-training on ImageNet and fine-tuning on the target dataset; (3) CNN feature extracted from ImageNet used for the multi-label AdaBoost.MH. The experiments shows that CNN model in ImageNet will be helpful greatly for the target problem when the target dataset is similar to ImageNet dataset, which gains 6% with the AUROC measure on MSRC dataset than the state-of-the-art work in our experiments. However, in the scene dataset, CNN has not attained the same level of success. This phenomenon shows that the scene recognitions do not benefit from the deep features trained from ImageNet due to their intrinsic differences of the feature representation.

The rest of the paper is organized as follows: Section 2 gives the framework of the multi-label classification; Section 3 introduces three multi-label image classification approaches under the CNN model; Section 4 details some experiments and the conclusions are given in Section 5.

Framework of Multi-label Classification

In this section, firstly we give a formal description of the multi-label classification and then introduce the classical multi-label classification algorithm AdaBoost.MH.

Formulation of Multi-label Classification

Let us consider a labeled dataset $\mathcal{D} = \{(\boldsymbol{x}_n, t_n) | n = 1, 2, \cdots, N\}$, where $\boldsymbol{x}_n \in \mathcal{R}^d$ and t_n is the subset of $\mathcal{L} = \{1, 2, \cdots, L\}$. The successful multi-label classification will produce a ranking of the possible labels to rank the outputs in the label set t on the top of those not in t or make the difference between the predict label set and the true label set as small as possible.

Multi-label AdaBoost.MH

Multi-label AdaBoost.MH algorithm maintains a distribution D_t that passed to the weak learner. It finds a set of weak hypotheses by calling the weak learner repeatedly in a series of rounds to minimize the average Hamming loss

$$R = \frac{1}{NL} \sum_{n=1}^{N} \sum_{l=1}^{L} |f(\boldsymbol{x}_n, l) \triangle t_n|,$$

where $\triangle$ denotes symmetric difference and $f(\boldsymbol{x}_n, l)$ is the output of the l-th class for the n-th example.

Before the training of the multi-label AdaBoost.MH, it initializes a set of weights over training examples and labels. As the iteration progresses, the training examples and their corresponding labels that are hard to predict correctly get incrementally higher weights while examples and labels that are easy to classify get lower weights.

Multi-label Image Classification with CNN

We introduce three approaches for multi-label image classification with the CNN model: (1) The end-to-end training on the target dataset; (2) CNN pre-training on ImageNet and fine-tuning on the target dataset; (3) CNN feature extracted from ImageNet for the multi-label AdaBoost.MH. We firstly describe the common CNN architecture shared by three approaches and then give their implementation details.

Common Architecture and Training

The shared CNN architecture by three approaches as shown in Table 1 is inspired by overfeat [11] similar to AlexNet and it has the following chacteristics: (i) use ReLUs (Rectified Linear Units) instead of contrast normalization; (ii) pooling regions are non-overlapping and the stride is smaller. The front part of network contains 6 convolutional layers and 3 full-connected layers, where the number of neuron in the last layer is equal to the number of the labels. ReLUs allow the network learn non-linear decision boundaries and affect the convergence rate and the quality of the obtained solution. Each image is cropped into a square image around the center of the image and then downsampled to 221x221 pixels.

Table 1. CNN architecture for multi-label image classification.

Stage	D	W	S	P	Stage	D	W	S	P
(1) conv + relu	96	7	2	0	(7) conv + relu	1024	3	1	1
(2) max		3	3	0	(8) conv + relu	1024	3	1	1
(3) conv + relu	256	7	1	0	(9) max		3	3	0
(4) max		2	2	0	(10) conv + relu	4096	5	1	0
(5) conv + relu	512	3	1	1	(11) conv + relu	4096	1	1	0
(6) conv + relu	512	3	1	1	(12) conv + tanh	#classes	1	1	0

D, the number of the neurons
W, the width of kernel
S, the sliding stride
P, the amount of the zero-padding
#classes, the label number of the dataset
conv, the convolutional layer
relu, the ReLU layer
max, the max-pooling layer

End-to-End Training

In topmost of the network, we add a tanh layer to constrain the output into $[-1,+1]$. It will helpful for the mean squared error criterion used for the multi-label learning

$$\mathcal{R} = \sum_{n=1}^{N} \|f(\boldsymbol{x}_n;\Theta) - t_n\|^2,$$

where t_n is the target label of the n-th example with each element in ±1, $f(\boldsymbol{x}_n;\Theta)$ is the output of the n-th example from the tanh layer and Θ is the network parameter. It is possible using sigmoid function but tanh has wider range space than sigmoid function, which can avoid falling early into the local stationary point. Back-propagation framework is used to compute the gradient. A CNN can be represented as the composition of the function of the each layer

$$f(\boldsymbol{x};\Theta) = g_K(\cdots g_k(g_{k-1}(\cdots g_1(\boldsymbol{x})\cdots)))$$

where $g_k(\cdot)$ is the forward transfer function of the k-th layer. We can iteratively compute $\frac{\partial f(\cdot)}{\partial g_k}$ with $\frac{\partial f(\cdot)}{\partial g_{k+1}} \times \frac{\partial g_{k+1}}{\partial g_k}$ and then obtain the gradient by

$$\frac{\partial f(\cdot)}{\partial \theta_k} = \frac{\partial f(\cdot)}{\partial g_k} \times \frac{\partial g_k}{\partial \theta_k}$$

Where Θ is in the stacked form of θ_k, e.g. $\Theta = [\theta_1^T, \theta_2^T, \cdots, \theta_K^T]^T$.

Pretraining and Finetuning

The CNN features are trained on ImageNet dataset and applied to the target dataset. Generally, it was demonstrated that fine-tuning a pre-trained CNN can significantly improve the learning performance. Before the training of the CNN, we initialize the weights with the trained CNN on ImageNet in all layers except the last layer. Then, the weights in all layers will be fine-tune by continuing the backpropagation. It is noted that CNN features are more generic in early layers and more dataset-specific in later layers. In order to distort the pre-trained weights too quickly and too much, the pre-training process maybe stop earlier than the end-to-end training on the target dataset.

Multi-label AdaBoost.MH with CNN Features

We take the CNN model pretrained on ImageNet and remove the last output layer (1000 neurons for different classes). The rest CNN can be regarded as a fixed feature extractor for the target dataset. The last but one layer in the network contains 4096 neurons. We use the output feature vector in combination with AdaBoost.MH to solve the multi-label classification problems. It is possible using other multi-label classifier such as one-vs-rest SVM or ML-LVQ [4]. However, AdaBoost.MH can be less susceptible to the overfitting problem than other learning algorithms and be referred as the best out-of-box classifier with only one hyperparameter.

Experiments

In this section, we evaluated three approaches on two multi-label image classification tasks including MSRC and 15-Scene dataset. We compared AUROC results among MC-Pos, MC-Simplex [2] with three approaches: Endto-End Training (EET), Pre-training and Fine-tuning (PF) and AdaBoost.MH with CNN features (ACNN). Three approaches ran on 'GeForce GTX TITAN X' GPU platform with Torch7 toolkit except that AdaBoost.MH was implemented by Java.

Datasets

The MSRC dataset consists of 591 images with 21 classes and each image has 3 classes on average. The measures were evaluated via five cross-fold validation and the process were repeated with 5 times by splitting randomly. The 15-scene dataset is composed of 4485 images with fifteen scene classes. In this dataset, we repeated splitting randomly each class of the dataset for ten times, where 100 examples in per class were selected as the training examples and the remain as the test ones. It is noted that all discussed algorithms will run on the datasets with the same split fashion.

Training Protocol

In the validation of the hyper-parameters (the epoches of iteration), about 1/3 of the training dataset were used to optimize AUROC. The difference of the loss values between the back epoch and the forth one is used to control the overfitting during the training stage, which eventually will be small enough (less than 10^{-3}) to lead to the end of the training. The parameters of the network were iteratively trained by stochastic gradient descent (SGD) with shuffled mini-batches.

We observed the larger batchsize does not influence the performance of the algorithm but can accelerate the convergence of the algorithm and result in smoother convergence, so we set the batchsize to 16 restricted to the capacity of GPU. We set the weight decay to 10^{-4} and the momentum to 0.9 by default. The examples are not large enough so that the learning rate was fixed to 10^{-3} for both of MSRC and 15-Scene.

Before training AdaBoost.MH, we normalized the dataset into the distribution with zero means and unit standard deviations for each attribute and then selected the best number of the stumps from $\{200, 400, 600, 800, 1000\}$ for the classifier model.

Results Analysis and Discussions

Table 2 shows that ACNN approach on MCRC dataset outperforms the state-of-the-art algorithm including MC-Pos and MC-Simplex methods by a significant 6% ~ 8% margin. Compared with EET, the fine-tuning is able to adjust the learned deep representation from ImageNet to better suit the target dataset with 6% improvement (from 84% to 90%). However, on Table 2, our three approaches obtains the inferior results than previous methods and only achieve at most 59% results. Evenif we used data augmentation such as the cropping and flipping to generate additional examples for training and test, it is no obvious difference in contrast with no data augmentation.

Table 2. AUROC measures on the MCRC dataset.

Avg. AUROC (%)	MCRC	15-Scene
MC-Pos	92	91
MC-Simplex	90	94
EET	84	54
PF	90	59
ACNN	98	54

Conclusions

We adapt a single-label CNN to handle the multi-label image classification problems by three approaches: end-to-end training on the target dataset, pre-training on ImageNet and fine-tuning on the target dataset, CNN features extracted from ImageNet for the AdaBoost.MH classifier. The architecture of CNN is kept as same for three approaches and the settings of the dataset as same for all algorithms. The experimental results on two different type of multi-label datasets show that CNN model or CNN features from ImageNet can achieve large gain in contrast with the classical hand-crafted features methods when handling the objection recognition problem, but it reaches the performance inferior to the baseline on the scene recognition problem because the CNN model (or features) grasps the object parts but drops the remain. Source code (https://github.com/xbjin) upon Torch7 toolkit to reproduce the experiments in the paper is made publicly available.

In future work, we will try the model pretrained on other large-scale scene datasets. In another direction, we also may use the output of the middle layers for the scene recognition.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under the grants No. 61103138, U1204617, 61300123 and the Fundamental Research Funds for the Henan Provincial Colleges and Universities in Henan University of Technology.

Reference

[1] M Boutell, J Luo, X Shen, and C Brown. Learning multi-label scene classification. Pattern Recognition, 37(9):1757–1771, 2004.

[2] Ricardo S. Cabral, Fernando Torre, Joao P. Costeira, and Alexandre Bernardino. Matrix Completion for Multi-label Image Classification. In NIPS 2011, pages 190–198, 2011.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR 2014, 2014.

[4] Xiao-Bo Jin, Guang-Gang Geng, Junwei Yu, and Dexian Zhang. Multi-label learning vector quantization algorithm. In 2012 21st International Conference on Pattern Recognition (ICPR), pages 2140–2143, 2012.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In NIPS 2012, pages 1106–1114, 2012.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[7] Xin Li, Feipeng Zhao, and Yuhong Guo. Multi-label image classification with a probabilistic label enhancement model. In UAI 2014, 2014.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In CVPR 2015, 2015. arXiv: 1411.4038.

[9] Robert E. Schapire and Yoram Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning, 37:297–336, 1999.

[10] Robert E Schapire and Yoram Singer. BoosTexter: A Boosting-based System for Text Categorization. Machine Learning, 39(2/3):135–168, 2000.

[11] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In ICLR 2014, 2014.

[12] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining Multi-label Data. Data Mining and Knowledge Discovery Handbook, pages 667–685, 2010.