Data Challenge for "Kernel Methods for Machine Learning"
Team Name: Crackernel

Dexiong Chen
master MVA
dexiong.chen@polytechnique.edu

Huiqin Chen
master MVA
huiqin.chen@telecom-paristech.fr

Pierre-Cyril Aubin
master MVA
pierre-cyril.aubin@polytechnique.edu

1 Introduction

In this challenge of multi-class classification of images, we have investigated and reviewed various kernel methods, including what we have seen in the course such as SVM, large scale learning with kernel methods and what we have not yet seen in the course such as "deep" kernel machines. Our method is based on the convolutional kernel networks (CKN) [4, 3], which builds kernel hierarchies via the composition of feature spaces and approximation of Gaussian kernels. Each image is represented by the last map of the CKN. In order to classify them, we have also implemented a dual coordinate descent method for linear SVM [2], which is much more efficient compared to the general quadratic programming softwares in this case. The final score of our challenge is 0.716 situated in the first place of the public leaderboard.

The report is organized as follows. Firstly, we present the principles of CKNs and the choice of parameters used in the challenge. Then, we review the dual coordinate descent for linear SVM [2] and the implementation details. Finally, we concentrate on the data challenge results and particularly illustrate our improvement track.

2 Convolutional Kernel Network

The key idea of the CKN [4] is to define a convolutional kernel on the patch feature space and approximate this kernel by the Euclidean inner product of two mapping vectors through the exploitation of an integral form of the Gaussian kernel. Then we can apply successively this procedure and construct a hierarchical structure representing larger and larger image neighborhoods. The original CKN [4] solves a non-convex optimization problem in order to approximate the true data points in a RKHS by vectors that may not live in the RKHS.

Lately, we have improved our method based on the new CKN [3] using only the unsupervised approach. In fact, using the variant of Nyström approximation, we can project data points onto a subspace of the RKHS with finite dimension. The paper proposed a spherical k-means method to efficiently construct the bases of this subspace. We have implemented both methods. Results and comparisons will be illustrated in the last section.

3 Dual Coordinate Descent Method for Linear SVM

Each image is represented by the last map of CKN. We have then implemented a dual coordinate descent method [2] to solve the large-scale linear SVM. This method is based on the coordinate

Model	Method	num. features	Accuracy
CKN [4]	GM	3200	56.64
	PM	3200	61.18
	CO1	8200	64.22
new CKN [3]	GM	3200	65.88
	PM	3200	65.62
	SM	3200	65.06
	CO1	8200	68.58
	CO2	12300	69.4
	CO3	82000	71.3

Table 1: Cross validation accuracy in % on the training set for different initial maps: gradient map (GM), patch map (PM), shape map (SM) and combination of the first two models and a single-layer model (CO). All models have only two layers. For the different combined models, the number of filters are chosen to be larger and larger: $(100, 200)$, $(200, 300)$ and $(1000, 2000)$.

descent with random permutation for coordinates and applies the shrinking technique to accelerate the optimization. Concretely, the shrinking technique consists of using an active-set and removing the coordinates from the set that satisfies no more the shrunken condition during the optimization, defined by the equation (16) in [2]. This method can generally accomplish the training within 1000 iterations in the challenge.

To extend the SVM to a multi-label classifier, we have used the one-versus-rest strategy. For the prediction phase, we choose the label that is farthest from the decision boundary.

4 Data Challenge Results

As presented previously, we have implemented the original CKN [4] and the new one [3] using Python and Cython. We have also implemented the dual coordinate descent method in Cython, which leads the training of SVMs to a few minutes. We have used several architectures with 2 layers. For the first layer (initial map), we have tried gradient map (GM) and patch map (PM) [4] with different numbers of filters. In addition, we have also tried a variant shape map for the first layer, defined in [1], but in a continuous manner. Specifically, we only compute the value differences around the center without binarization and then perform the variant Nyström approximation to project onto a finite dimensional subspace. The representations of images obtained by CKNs are centered and rescaled to have unit ℓ_2-norm on average. Different representations obtained by different architectures can also be combined to higher dimensional vectors. These representations are then used to train a linear SVM, whose regularization parameter is selected by 5-fold cross validation on the training set.

The comparisons of cross validation accuracies for the different architectures are displayed in Table 1. Using only features learned by shape map gives an accuracy of 65.06%. However, we notice that adding these representations to the combined model gains no improvement in terms of accuracy, which may be due to the fact that the features extracted by the shape map are similar to that extracted by other maps such as gradient map. Another observation is that the new CKN [3] pronouncedly outperforms the original CKN [3] for all architectures. Furthermore, we have also noticed that for the new CKN [3], wide architectures lead to better performance. Regarding the running time of the implementation, the new CKN is very efficient to learn and it takes only about 20 minutes on a 2,3 GHz Intel Core i7 CPU to finish the training of the new CKN-CO1 model and the cross validation for SVM, which achieves a classification accuracy of 70.6% on the testing set according to the public leaderboard.

5 Conclusion

In this data challenge, we have reviewed and reimplemented the two CKN models [4, 3] and compared performances of different architectures. We have shown that wide shallow CKN model gives the best result, 71.3% in terms of the cross validation accuracy.

References

[1] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1729–1736. IEEE, 2011.

[2] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415. ACM, 2008.

[3] J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in Neural Information Processing Systems*, pages 1399–1407, 2016.

[4] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *Advances in Neural Information Processing Systems*, pages 2627–2635, 2014.