A New Method of Multi-Scale Receptive Fields Learning

Shaorong Feng

Department of Computer Science, Xiamen University, 361005 Xiamen, China

shaorong@xmu.edu.cn

Abstract

Deep learning architecture has been applied in computer vision to learn features in an unsupervised manner. Thousands of features can be achieved in such manner. Furthermore, in some modified architectures, multi-scale features which contain middle layer features and output layer features, can connect to classifier. The classifier is trained using these features to predict the label of input image. The multi-scale can provide both global structures and local details, but it is prone to cause overfitting due to the expansion of features, which will make the performance degrade. In this paper, we propose a method to limit the number of features by multi-scale receptive fields (MSRF) learning. With this method, we can choose the most effective receptive fields in multiple scales. It will improve classification performance in the object recognition task. In our experiments, we compare several pre-define pooling strategies and receptive fields learning algorithm. The MSRF learning achieves the best performance among the results.

Keywords: multi-scale; receptive fields; features; overfitting; learning.

Introduction

Recently, a great number of computer vision researches is in the field of deep learning. Deep learning has drawn much attention due to its hierarchical unsupervised learning properties. Break through of neural networks happened in 2006, Hinton, et al. proposed deep learning architecture for handwritten digits recognition [1]. The network can be trained layer by layer in a unsupervised manner. Then, these trained layers are stacked to be a network. In the following years, layer-wise training [2] become extremely popular. Numerous algorithms based on deep learning architecture were resented.

Deep learning offers an effective way to find structures hidden in the data. With this method, we can use a multilayer pipeline to find good image features. Some deep learning networks start with extracting patches from raw images or descriptor maps [3] (such as HOG or SIFT). An overcomplete dictionary is learned using these patches with *k-means* or sparse coding algorithm. Then, a representation is achieved by encoding with the overcomplete dictionary. Using a smaller labeled training dateset, fine-tuning is implemented in the networks to its optimal state.

The terminology of receptive field is borrowed here from biology. It consists of simple cells in mammalian primary visual cortex and it can be characterized as being spatially localized, oriented and bandpass. In [4], the authors apply the concept in constructing image representation, proposing sparse coding to simulate working manners of V1 region in human brain that only response to specific characteristics. The researches of receptive fields in visual nervous system inspire the idea of spatial pooling which combines the responses of features at nearby locations by some statistic operations, such as average or maximum. Pooling is an important component in many computer vision architectures, for instance, convolutional neural networks. Some literatures systematically make analysis and comparison on pooling strategies [5]. The common used spatial pooling is on regular grid. In [6], [7], a type of pooling region called spatial pyramid is proposed, which provides a reasonable cover with scale information. Jia et al. designs overcomplete receptive fields to improve networks performance [8]. The overcomplete receptive fields consist of all rectangle candidates on the raw image or feature map. Coates et al. uses receptive field learning to limit connections between layers, which greatly reduces the parameters in deep networks.

In this paper, we investigate that how to sufficiently utilize features in various scales. In the classification pipeline we adopted, local details and global representation are used to predict labels in final classifier. However, not all local and global features are fed into the classifier. An incremental feature selecting algorithm is implemented to decide which feature can be connected. To further improve the performance, the idea of overcomplete receptive fields is adopted in the final pooling stage for learning a group of overcomplete multi-scale pooling regions. We integrate overcomplete receptive fields with multiscale features in our experiments. The experiments with our algorithm on the dataset CIFAR-10 show an improvement in accuracy which is better than results in [8]. The overcomplete multi-scale receptive fields show the best performance with middle scale dictionary, though, to our surprise, the learning algorithm on overcomplete MSRF dose not obtain the top accuracy in the experiments.

The Network Architecture

The architecture adopted in our research is a two stage networks and each stage has two layers, a coding layer and a pooling layer. We achieve different scale features in each stage. Finally, a classifier uses all of these multi-scale features to make prediction.

A. Coding

Before we begin to code raw images or lower features, it is necessary to learn an overcomplete dictionary. An overcomplete dictionary means that the columns in the dictionary are far greater than the dimension of input data. Normalization and ZCA whitening should be performed on raw images or lower features before training and coding since that we hope the training algorithm see the pixels or features less redundant and same variance.

There are two simple and effective unsupervised learning algorithms, *k-means* and OMP-k, shown us a quite good performance in [9]. *k-means* clustering is a fast training algorithm and easy to implement. It computes the centroid of size k by minimizing the squared distance:

$$\min_{D,s(i)} \sum_i \| x^{(i)} - s^{(i)} \|_2^2 \tag{1}$$

where $x^{(i)}$ is a vector that is reshaped from patches sampled randomly from raw images or lower features and $s^{(i)}$ is the centroid should be learnt.

OMP-k is also an unsupervised learning algorithm. It aims to minimize the reconstruction error with constraint and can be described as follows:

$$\min_{D,s(i)} \sum_i \| Ds^{(i)} - s^{(i)} \|_2^2 \quad \text{subject to} \quad \| D^{(i)} \|_2^2 = 1 \quad \forall j \quad \textit{and} \ \| s^{(i)} \|_{\mathbf{0}} \leq m \,, \quad \forall i \tag{2}$$

where $\|s^{(i)}\|_0$ denotes the number of non-zero elements in s(i). That means each patch can be represented by a combination of at most k codes. $D^{(i)}$ stands for a column in dictionary *D*. Then, employing Orthogonal Matching Pursuit (OMP) algorithm [10], [11], we can obtain a dictionary with at most *k* codes. When we choose *k*=1 in OMP-*k*, the learning method became similar to *k-means* which dictionary elements $D^{(j)}$ all have unit length.

Obtained the dictionary using above learning method, we then need to choose encoder that converts low-layer features to high-layer. Soft thresholding is a type of encoder. The encoder calculates the inner product of the $x^{(i)}$ and each elements of the dictionary. Then, we select a maximum value between zero and the result subtracts a fixed threshold:

$$f_j = \max\{0, D^{(j)T} x - \alpha\} \tag{3}$$

where α is the threshold specified manually. Another commonly used technique, triangle coding, is proposed in [12]. It can be described as:

$$f_j = \max\{0, \mu(z) - z_k\} \tag{4}$$

where $z_k = \|x^{(i)} - c^{(k)}\|_2$ and $\mu(z)$ is the mean of the z. The activation function means that when the distance to the centroid c^k is farther than the average $\mu(z)$, the function outputs 0. The two encoding approaches take advantage of fast and efficient for feature coding.

In our experiments below, for compromise speed and performance, we will perform *k-means* to training dictionary and the triangle encoder is adopted for coding.

B. Pooling

The idea of pooling operation originates in research on complex cell in visual cortex [13]. The region of pooling is called receptive fields in neuroscience. Pooling operation is to aggregate statistics of features over a local neighborhood and pooling will generate a certain degree of translation invariant and reduce dimensions to avoid overfitting.

In details, after obtaining coded features as described above, we can take average or max operation on the neighborhood feature activations, such as 2×2 and 3×3. A set of values is work out to represent these adjacent regions. These pooled features are smaller representation than input features maps.

The pooling regions have many kinds of types. It is usually to define pooling regions on regular grids [7] or spatial pyramids [6]. However, Jia et al. [8] have proposed a novel overcomplete receptive fields that can effectively improve classification accuracy. The overcomplete receptive fields consist of all rectangular regions on a $n \times n$ grid.

Applying overcomplete receptive fields will make the number of receptive field candidates rapidly increase. There will be 100 overcomplete regions more than 16 regular pooling regions on a 4×4 regular grid. Hence, the number of pooled features is easy to become unacceptable when there are enormous feature maps. It is hard to train a classifier well in this situation.

Multi-Scale Receptive Fields Learning

Inspired by the idea of the multiple path approaches in convolutional neural networks that can fuses local and global information [14], [15], we propose an algorithm to learn receptive fields in multiple scales for improving classification performance. The most effective receptive fields can be selected from a large number of candidates. Using the small part of receptive field candidates will avoid overfitting and forms a multi-scale representation containing effective local and global information.

A. Multi-Scale Features

Traditional convolutional neural networks (CNN) are constructed in strict feed-forward layered structure that lower layer’s output is taken as input of the layer above. Instead, a type of CNN that each layer has a global branch and a local branch is proposed in [14]. The global branch can enlarge the region of receptive fields, and the local branch can capture more local details of image and is directly connected to classifier. There are some unsupervised versions described in [15]. These networks are constructed based on convolutional sparse coding with layer-wised pre-training at each stage.

This architecture with multi-scale receptive fields may makes parameters of classifier blow up. The blowup leads difficulty in training and to cause overfitting easily. In reality, only a small portion of receptive fields have contribution and a majority of receptive fields have no effect and time-consuming for training. The problem can be settled by finding which are the most effective features from multi-scale receptive fields.

B. Receptive Fields Learning

Given multi-scale receptive fields, we can achieve pooled features from different layers after pooling operation on these fields. Each pooled feature relates to a receptive field. Contribution of each pooled feature reflects that the importance of corresponding receptive field. Hence, the goal of learning algorithm is to discriminate importance of these pooled features and to remove redundance.

Feature selection algorithms can meet this goal and there are various methods to choose. Here, we adopt an incremental approach [16] which scales linearly with the number of samples and most quadratically with the number of features. Features are gradual selected to a feature set while training a classifier using gradient Descent. It is able to balance speed and performance well.

Before we begin, we introduce a multiple-class linear classifier with structured sparsity constraint. The multiple-class linear classifier brings two benefits: reducing feature computation and sharing features among classifiers [8]. Utilizing global and local features, we train the classifier *y*=*f*(*x*)=*Wx*+*b* by optimizing energy function below:

$$\min_{\boldsymbol{W},\boldsymbol{b}} \frac{1}{N} \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} l(W^T x_n + b, y_n) + \lambda \| W \|_{Fro}^2 \qquad (5)$$

where *W*, *b* and *N* are weight matrix, bias and the number of samples, x_n notates input features and y_n stands for the labels. There are also approaches using additional term $\lambda_2 s$, where *s* is the number of selected features, or sparsity regularization $\lambda_2\|\mathrm{W}\|_{1,\infty}$. We do not adopt these ideas, because the number of learned features is fixed in our experiments. In addition, feature selection algorithm implicitly introduces sparsity in the optimization. Hence, we drop the structure sparsity constraint.

The first term is a multi-class extension of the binomial negative log likelihood (BNLL) loss function [16]:

$$l(W^T x + b, y) = \sum_{n=1}^{N} ln(1 + e^{-\boldsymbol{y_i}(w_{.,i}^T + bi)}) \qquad (6)$$

The second term $\lambda \| W \|_{Fro}^2$ is squared Frobenius norm of weight matirx. It helps the classifier minimize the structure loss and avoids overfitting in the optimization.

Given objective function eq.(5), we can implement grafting algorithm to select features. With this method, a set of pooled features will be obtained. Receptive fields corresponding to these features are the most effective regions for improving performance. The algorithm can be described as following steps:

Step 1. Establish a empty selected feature set *S* and a candidate feature set S_c which contains all the features;

Step 2. Add features to the set *S* and remove the features from S_c;

Step 3. Retrain the model when new features are added.

Repeat Step 2 and Step 3 until a fixed number of features are obtained.

In details, we can assume that there are a set S_c that contains pooled feature candidates and a set *S* that keeps the features which are selected. In each iteration, we calculate a score for each unselected feature:

$$score(j) = \left|\left| \frac{\partial L(W,b)}{\partial W_{j,\cdot}} \right|\right|_{Fro}^{2} \tag{7}$$

where $L(W, b)$ is the objective function. We choose the feature with maximum score, add it to the set S and remove it from the set S_c. Then, the classification model should be retrained. Take the parameter trained in last iteration as initial value in next iteration.

Experiments

We have implemented our method on the CIFAR-10 dataset which contains 60000 32×32 color image (50000 training images and 10000 testing images) from 10 categories.

The experimental pipeline is described in Section II and we employ *k-means* training algorithm and triangle encoder in the experiment. For comparative purposes, some experiment parameters are the same as experiments in literature [8], [9]. When implementing overcomplete receptive fields or selecting algorithm, λ is fixed to 0.01 in eq.(5). For pre-define pooling regions, SVM regularization term is chosen to train the classifier with 5-fold cross validation on the training data.

In first stage, we use 6×6 receptive fields to learn dictionary and extract features. After coding step, a 32×32 color image becames 200 27×27 feature maps where we have a dictionary of size 200. Then pooling operation is implemented on these feature maps. We use average or maximum pooling over adjacent, disjoint 3×3 spatial blocks. 9×9×200 pooled features are achieved as first stage representation. For obtaining second stage features, we connect each feature extractor to a rectangle area within feature maps, rather than connecting extractor to all features in the first stage. By this means, the number of parameters is dramatically reduced. That will make the networks achieve higher performance. We take 2×2×50 receptive fields on each first stage feature maps to extract 25 feature maps. It means that we randomly choose 50 maps of first layer as a group and extract patches from 2×2 regions on these first layer maps. We have 8 groups to achieve 8×8×200 feature maps in the stage. Then 4×4×200 features are done with 2×2 pooling operation.

In the experiments, the overcomplete receptive fields are defined on 4 regular grid. Hence, an extra-pooling is needed. For example, in first stage we implement an extra-pooling on 8×8×200 pooled features to have 4×4×200 features with 2×2 pooling regions. Overcomplete regions then will be constructed on these features.

TABLE 2 COMPARISON OF MULTI-SCALE RECEPTIVE FIELDS

Metho d	Area	Stage-1	Stage-2	MSRF	MSRF Learned
MAX	2×2	65.73	65.77	69.21	
AVE	2×2	69.76	68.12	72.71	
MAX	4×4	70.63	70.38	71.90	
AVE	4×4	73.25	71.23	74.09	
MAX	OC	76.01	73.90	77.93	76.50
AVE	OC	73.59	71.16	75.39	73.44

TABLE 1 COMPARISON OF DIFFERENT POOLING STRATEGIES

Architecture	Pooling Area	Metho d	Features	Accuracy
1 Layer	2× 2	AVE	800	69.76
1 Layer	2 ×2	MAX	800	65.73
1 Layer	4 ×4	AVE	3200	70.63
1 Layer	4 × 4	MAX	3200	73.25
1 Layer	OC, all features	AVE	20000	73.59
1 Layer	OC, all features[10]	MAX	20000	76.44
1 Layer	OC, feat select[10]	MAX	6400	76.72
2 Layer	4× 4	AVE	3200	70.38
2 Layer	4× 4	MAX	3200	71.23
2 Layer	MSRF, OC, all feat	AVE	40000	75.39
2 Layer	MSRF, OC, all feat	MAX	40000	77.93
2 Layer	MSRF, OC, feat select	AVE	6400	73.44
2 Layer	MSRF, OC, feat select	MAX	6400	76.50

A. Comparison of Different Pooling Strategies

Results from Table 1 show us that overcomplete receptive fields is more effective than regular grid pooling. Here, we refer to Jia et al.'s results [8] as baseline in Table I and, in fact, our result, 76.01% (in Table 2), is slight lower than theirs, 76.44%, with all overcomplete features. Max pooling always has better result than average pooling when we choose overcomple as our pooling strategies. We can find more discussions about overcomplete receptive fields in literature [8], but the effect of pooling operator has not been involved.

In our experiments, we take overcomplete receptive fields in first stage and second stage as a MSRF. We compare our multiscale receptive field (MSRF) to other pooling regions. The overcomplete pooling always has better results than regular grid and the MSRF can improve the performance based on regular grid or overcomplete receptive fields. Though our result 76.50% with MSRF learning is lower than the accuracy 76.72% in [8], but it is better than the results using all overcomplete receptive fields, 76.01%, in our implementation, which is 76.44% in [8]. To our surprise, using overcomplete multi-scale receptive fields without learning, we obtain the best performance 77.68% among all results in the experiments, which has an improvement of 1.21%.

B. Comparison of Multi-Scale Receptive Fields

Table 2 shows us the performance with various receptive fields.We can see that the accuracy is hard to increase too much even using higher layer features with regular grid pooling or overcomplete receptive fields. In their experiments, the accuracy based on first layer features is slight better than second layer features without any other constraints. Similar results can be seen in our experiments.

Comparing to use receptive fields in the stage-1, the performance of stage-2 decrease by 2% on average. Only maximum pooling makes the accuracy increase by 0.04% with 2×2 pooling regions. However, our MSRF can improve the performance to 76.50% with learning. Intuitively, MSRF learning algorithm should have the top accuracy in the experiments, but we have the best performance 77.68% by means of MSRF without learning. The results may suggest that the incremental feature learning algorithm cannot capture the most useful features well, when the MSRF is implemented in the classification pipeline.

Conclusion

In this paper, we have investigated the effect of multiscale receptive fields on the classification task. While using a denser grid or overcomplete can obtain performance increase, it is hard to achieve a better accuracy employing higher layer feature. To address the problem, we have proposed to combine the multi-scale features with overcomplete receptive fields in hierarchical unsupervised learning architecture. Then, we implement learning algorithm on the overcomplete receptive fields. With MSRF learning, we hope to take full advantage of local details and global information to enhance the performance of classification pipeline. To our surprise, using overcomplete multi-scale receptive field, we obtain the best performance among all results in the experiments without learning. With this method, an improvement has shown in our experiments compare to overcomplete receptive fields. Our further work is to find explanations: why does the learning algorithm not work as our expectation and how to design a powerful method to address the problem on multi-scale features.

Acknowledgement

In this paper, the research was supported by Major Program of National Social Science Foundation of China(13&ZD148), National Natural Science Foundation of China(61303004), and Natural Science Foundation of Fujian Province of China(2013J05099).

References

[1] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks.” *Science (New York, N.Y.)*, Vol. 313, No. 5786, 7, Jul. 2006: 504.

[2] Y. Bengio and P. Lamblin, “Greedy layer-wise training of deep networks,” *Neural Information Processing Systems*, 2007.

[3] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, Vol. 60, No. 2, 2004: 91-110.

[4] B. Olshausen, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, Vol. 381, No.13, 1996: 607-609.

[5] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, “Learning mid-level features for recognition,” in *Computer Vision and Pattern Recognition* (*CVPR*), 2010 *IEEE Conference on*. IEEE, 2010: 2559-2566.

[6] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Computer Vision and Pattern Recognition,* 2006 *IEEE Computer Society Conference on*, Vol. 2. IEEE, 2006: 2169-2178.

[7] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *Computer Vision and Pattern Recognition*, 2009*. CVPR* 2009*. IEEE Conference on*. IEEE, 2009: 1794-1801.

[8] Y. Jia, C. Huang, and T. Darrell, “Beyond spatial pyramids: Receptive field learning for pooled image features,” *Computer Vision and Pattern Recognition*, 2012: 3370-3377.

[9] A. Coates and A. Ng, “The importance of encoding versus training with sparse coding and vector quantization,” *Proceedings of the* 28*th* *International Con-ference on Machine Learning,* 2011.

[10] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Signals, Systems and Computers,* 1993. 1993 *Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993: 40-44.

[11] T. Blumensath and M. E. Davies, “On the difference between orthogonal matching pursuit and orthogonal least squares,” 2007.

[12] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” *Proceedings of the 14th International Con-ference on Artificial Intelligence and Statistics*, 2011.

[13] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of physiology*, Vol. 160, No.1, 1962: 106.

[14] J. Fan, W. Xu, Y. Wu, and Y. Gong, “Human tracking using convolutional neural networks,” *Neural Networks, IEEE Transactions on*, Vol. 21, No.10, 2010: 1610-1623.

[15] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning,” in *Computer Vision and Pattern Recognition* (*CVPR*), 2013 *IEEE Conference on*. IEEE, 2013: 3626-3633.

[16] S. Perkins, K. Lacker, and J. Theiler, “Grafting: Fast, incremental feature selection by gradient descent in function space,” *The Journal of Machine Learning Research*, Vol.3, 2003: 1333-1356.