COGS 260 Spring 2018: Assignment 2

Qiaojun Feng
University of California, San Diego
qjfeng@ucsd.edu

Abstract

In this assignment we compared the performance of different methods of image recognition on MNIST dataset[5].

1. Method

Here we compared five categories of methods, including Convolutional Neural Network[5, 8], 1-Nearest Neighbor, Support Vector Machines, Spatial Pyramid Matching[4] and Deep Belief Nets[3].

1.1. Convolutional Neural Network

Convolutional Neural Network is the most popular machine learning algorithm nowadays. It can work because its complex architecture enhance the ability of representation. Also the multi-layer structure can combine both local and global features. With enough training data, a good classifier can be generalized to many other test data.

1.2. 1-Nearest Neighbor

Nearest Neighbor method simply memorizes all the training dataset and uses the distance metric to assign label to test dataset. The learning stage doesn't cost any time actually, merely loading all the data. But the prediction stage will be expensive as the training set increases. 1-Nearest Neighbor picks the nearest neighbor point and assigns its label.

1.3. Support Vector Machines

Support Vector Machines(SVM) are popular discriminative classification algorithms. Also it is a non-probabilistic classifier. The main idea is to use different kernel to project the feature vectors to a different space and use a linear classifier to distinguish them as much as possible. The origin version is a binary classifier. When dealing with more than two categories, multiple binary classifiers can be combined and each of them classifier (1)one class against all the others or (2)every two classes.

1.4. Spatial Pyramid Matching

Spatial Pyramid Matching is more like a feature engineering work, i.e., the feature extraction. It leverages predefined spatial structure so captures holistic information to some extent. After the features being extracted, SVM is used for classification

1.5. Deep Belief Nets

Deep Belief Nets is a generative model. It first learns some feature distribution of the training data without specific supervision. After the feature detector is learned, a classifier can be trained accordingly.

2. Experiments

2.1. Dataset

We used the whole MNIST dataset with a training set of 60,000 examples, and a test set of 10,000 examples. Each one is a 28×28 size gray-scale image with a label among 0-9.

2.2. LeNet

The given LeNet model was directly used. The architecture was not changed, with 2 convolution layers, 2 maxpooling layers and 2 dense connected layers. Adadelta optimizer was used.

2.3. VGG16

We did transfer learning on the imagenet-pretrained VGG16 model provided by keras. Adadelta optimizer was used. The pretrained model gave the feature map output. We concatenated one GlobalAveragePooling layer and two dense connected layers. We first kept the origin VGG16's layers and only tuned the parameters of the newly-added layers. Afterwards we also trained the last 4 layers together with the final 3 layers together.

2.4. 1-Nearest Neighbor

We used the implementation of scikit-learn [7]. We tested with euclidean distance and manhattan distance. The former(0.969) is slightly better than the latter(0.963).

2.5. Support Vector Machines

We used the implementation of scikit-learn [6]. We tested with Radial basis function(RBF) kernel and linear kernel. The former(0.945) is slightly better than the latter(0.918).

2.6. Spatial Pyramid Matching

We used the codes from [2]. The default parameters setting is vocabulary size VOC_SIZE = 100, pyramid structure PYRAMID_LEVEL = 1, sift sampling stepsize DSIFT_STEP_SIZE = 4. We didn't tune these parameters since it took a long time to run once.

2.7. Deep Belief Nets

We used the implementation of Deep Belief Nets from [1]. The parameters setting is listed below.

hidden_layers_structure	[256, 256]
learning_rate_rbm	0.05
learning_rate	0.1
n_epochs_rbm	10
n_iter_backprop	100
batch_size	32
activation_function	ReLU
dropout_p	0.2

Table 1. Parameters setting

3. Discussion

We only reported the classification accuracy in the report Table 2. For the confusion matrices they could be found in the supplementary file COGS 260 Assignment 2.pdf. Also the training and validation loss(using test set) were reported in Figure 1.

LeNet	VGG16	1-NN	SVM	SPM	DBN
0.993	0.990	0.969	0.945	0.868	0.985

Table 2. Classification Accuracy

We can classify these methods into two categories based on its input data format. Basically Nearest Neighbor(NN) and Support Vector Machine(SVM) are the general classifiers and they don't capture any spatial information of the 2D image, simply take it as a long vector. Therefore usually they are fast. And since this is a simple dataset, the results are acceptable. The other category takes the image matrix as the input and process on it. Spatial Pyramid Matching(SPM) leverages the spatial information and extract features with different spatial information. However here we didn't tune the hyper-parameters well so it appeared to be not as good as even simple methods. The main problem

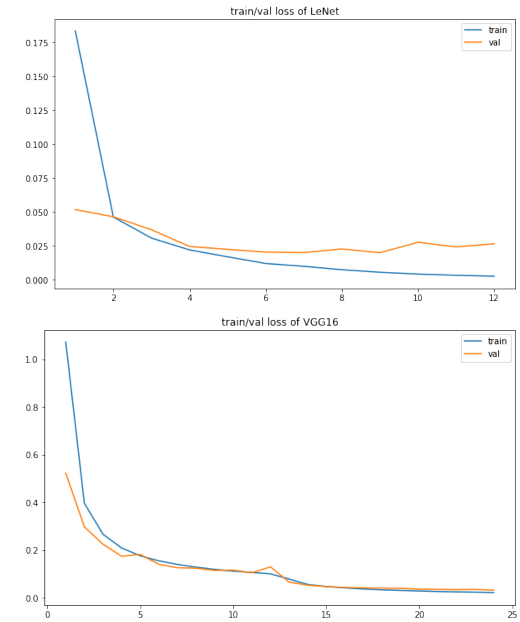

Figure 1. training and validation loss using LeNet and VGG16

here is that the features are still hand-crafted. Deep Belief Nets(DBN) and other Convolutional Neural Network(CNN) methods can learn the feature on their own. And we can see their performance are better than the others. While the DBN is still a shallow model, LeNet and VGG16 have more layers so that they can performance even better.

References

[1] albertbup. A python implementation of deep belief networks built upon numpy and tensorflow with scikit-learn compatibility. https://github.com/albertbup/deep-belief-network. 2

[2] CyrusChiu. Image recognition methods from bag of words (bow), spatial pyramid matching (spm), sparse codeing spm (scspm) to convolutional neural networks (cnn). https://github.com/CyrusChiu/Image-recognition. 2

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 1

[4] 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*, volume 2, pages 2169–2178. IEEE, 2006. 1

[5] 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. 1

[6] scikit learn. 1.4. support vector machines scikit-learn 0.19.1 documentation. http://scikit-learn.org/stable/modules/svm.html. 2

[7] scikit learn. 1.6. nearest neighbors scikit-learn 0.19.1 documentation. http://scikit-learn.org/stable/modules/neighbors.html. 1

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1